# Motor Algebra Approach for Computing the Kinematics of Robot Manipulators

**Eduardo Bayro-Corrochano***
*Centro de Investigación en Matemáticas, A.C.*
*Apartado Postal 402*
*36000, Guanajuato, Gto., Mexico*
*e-mail: edb@fractal.cimat.mx*

**Detlef Kähler**
*Computer Science Institute*
*Christian Albrechts University*
*Preußerstraße 1-9, 24105*
*Kiel, Germany*
*e-mail: dek@ks.informatik.uni-kiel.de*

Received December 11, 1999; accepted June 5, 2000

This article presents the formulation of the robot manipulator kinematics in the geometric algebra framework. In this algebraic system the three-dimensional Euclidean motion of points, lines, and planes can be advantageously represented using the algebra of motors. The computational complexity of the direct and indirect kinematics and other problems concerning robot manipulators depend on their degrees of freedom as well as on their geometric characteristics. Our approach makes possible a direct algebraic formulation of the problem in such a way that it reflects the underlying geometric structure. This is achieved by switching where necessary to a description of parts of the problem based on motor representations of points, lines, and planes. This article presents the formulation and computation of closed-form solutions of the direct and indirect kinematics of standard robot manipulators and a simple example of a grasping task. The flexible method presented here is new, and it widens the current standard point or line representation-based approaches for the treatment of problems related to robot manipulators. 

* To whom correspondence should be addressed.

Journal of Robotic Systems 17(9), 495–516 (2000)

## 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[1] introduced the kinematic notation used most for lower pair mechanisms based on matrix algebra, Walker[2] used the epsilon algebra for the treatment of the manipulator kinematics, Gu and Luh[3] utilized dual-matrices for computing the Jacobians useful for kinematics and robot dynamics, and Pennock and Yang[4] derived closed-form solutions for the inverse kinematics problem for various types of robot manipulators employing dual-matrices. The dual form of the Jacobian for the analysis of multilinks was used similarly by McCarthy.[5] Finally, Funda and Paul[6] 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 for dealing with the kinematics of robot chains than the unit quaternion formalism. Using dual quaternions Kim and Kumar[7] computed a closed-form solution of the inverse kinematics of a 6-degree of freedom robot manipulator in terms of line transformations. Aspragathos and Dimitros[8] confirmed once again that the uses of dual quaternion and Lie algebra in robotics were overseen so far and that their use helps to reduce the number of representation parameters.

In all these mathematical approaches 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 article we are concerned with the extension of the representation capabilities of the dual numbers, considering particularly the case of using the motor algebra beside the point and line representation to enable the modeling of the motion of planes. This widens the possibilities for the modeling of the motion of the basic geometric objects, that are referred to frames attached to the robot manipulator which, according to the circumstances, simplifies the complexity of the problem, preserving the underlying geometry. After giving the modeling 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 article clearly shows the advantages of the use of representations in motor algebra for solving problems related to robot manipulators.

The organization of the article is as follows: section 2 gives an introduction to the geometric algebra and section 3 to the motor algebra; section 4 is dedicated to the representation of points, lines, and planes; section 5 involves the modeling of the motion of points, lines, and planes; section 6 describes the prismatic and revolute transformations of robot manipulators in the motor algebra framework; section 7 deals with the computation of the direct kinematics of robot manipulators; section 8 is dedicated to the solution of the inverse kinematics of one standard robot manipulator, and finally, section 9 presents the conclusions.

In this article multivectors are presented in bold italic type, scalars in italic, and matrices in bold. Lower case letters denote vectors in $\mathscr{E}^3$ and upper case letters denote bivectors in $\mathbb{R}^4$.

## 2. GEOMETRIC ALGEBRA

Clifford algebra is a coordinate-free approach to geometry based on the algebras of Grassmann[9] and Clifford.[10] The approach to Clifford algebra we adopt here was pioneered in the 1960s by David Hestenes[11] and later, with Garret Sobczyk, was developed into a unified language for mathematics and physics.[12,13] Some preliminary applications of geometric algebra to the field of computer vision and neural computing have already been given.[14–16]

### 2.1. Basic Definitions

Let $\mathscr{G}_n$ denote the geometric algebra of $n$-dimensions. The *geometric* or *Clifford product* of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is written $\boldsymbol{ab}$ and can be expressed as a sum of its symmetric and antisymmetric parts

$$\boldsymbol{ab} = \boldsymbol{a}\cdot\boldsymbol{b} + \boldsymbol{a}\wedge\boldsymbol{b}, \tag{1}$$

where the inner product $\boldsymbol{a}\cdot\boldsymbol{b}$ and the outer product $\boldsymbol{a}\wedge\boldsymbol{b}$ are defined by

$$\boldsymbol{a}\cdot\boldsymbol{b} = \tfrac{1}{2}(\boldsymbol{ab} + \boldsymbol{ba}) \tag{2}$$

$$\boldsymbol{a}\wedge\boldsymbol{b} = \tfrac{1}{2}(\boldsymbol{ab} - \boldsymbol{ba}) \tag{3}$$

The inner product of two vectors is the standard *scalar* or *dot* product and produces a scalar. The outer or wedge product of two vectors is a new quantity which we call a *bivector*. We think of a

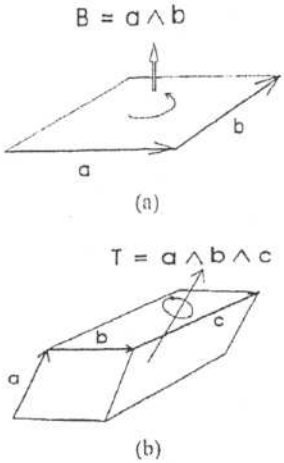


Figure 1. The directed area, or bivector, $a \wedge b$. (b) The oriented volume, or trivector, $a \wedge b \wedge c$.

bivector as an oriented area in the plane containing $a$ and $b$, formed by sweeping $a$ along $b$ (Fig. 1a).

The bivector $b \wedge a$ has the opposite orientation and is antisymmetric as given in Eq. (3). The outer product is immediately generalizable to higher dimensions. The outer product of $k$ vectors is a $k$-vector or $k$-blade and is said to have *grade k* (Fig. 1b). A *multivector* (a linear combination of objects of different grades) is said to be *homogeneous* if it contains terms of only a single grade.

In a space of 3 dimensions we can construct a trivector $a \wedge b \wedge c$, but no 4-vectors exist since there is no possibility of sweeping the volume element $a \wedge b \wedge c$ over a 4th dimension. The highest grade element in a space is called the *pseudoscalar*. The unit pseudoscalar is denoted by $I$ and is crucial when discussing duality.

### 2.2. The Geometric Algebra of *n*-Dimensional Space

In an $n$-dimensional space we can introduce an orthonormal basis of vectors $\{\sigma_i\}$ $i = 1, \ldots, n$, such that $\sigma_i \cdot \sigma_j = \delta_{ij}$. This leads to a basis for the entire algebra:

$$1, \{\sigma_i\}, \{\sigma_i \wedge \sigma_j\}, \{\sigma_i \wedge \sigma_j \wedge \sigma_k\}, \ldots, \sigma_1 \wedge \sigma_2 \wedge \cdots \wedge \sigma_n. \quad (4)$$

Note that the basis vectors are not represented by bold symbols. Any multivector can be expressed in terms of this basis. In this article, a geometric algebra $\mathcal{G}_n$ is of the form $\mathcal{G}_{p,q,r}$, where $p$, $q$, and $r$ stand for the number of basis vectors which square to 1, $-1$, and 0, respectively, where $n = p + q + r$. Its even subalgebra will be denoted by $\mathcal{G}^+_{p,q,r}$.

In the $n$-D space there are multivectors of grade 0 (scalars), grade 1 (vectors), grade 2 (bivectors), grade 3 (trivectors), etc., up to grade $n$. Any two such multivectors can be multiplied using the geometric product. Consider two multivectors $A_r$ and $B_s$ of grades $r$ and $s$, respectively. The geometric product of $A_r$ and $B_s$ can be written as

$$A_r B_s = \langle AB \rangle_{r+s} + \langle AB \rangle_{r+s-2} + \cdots + \langle AB \rangle_{|r-s|} \quad (5)$$

where $\langle M \rangle_t$ is used to denote the $t$-grade part of multivector $M$, e.g., consider the geometric product of two vectors $ab = \langle ab \rangle_0 + \langle ab \rangle_2 = a \cdot b + a \wedge b$.

### 2.3. The Geometric Algebra of 3D Space

The basis for the geometric algebra $\mathcal{G}_{3,0,0}$ of the 3D space has $2^3 = 8$ elements and is given by

$$\underbrace{1}_{\text{scalar}}, \underbrace{\{\sigma_1, \sigma_2, \sigma_3\}}_{\text{vectors}}, \underbrace{\{\sigma_1\sigma_2, \sigma_2\sigma_3, \sigma_3\sigma_1\}}_{\text{bivectors}}, \underbrace{\{\sigma_1\sigma_2\sigma_3\} \equiv I}_{\text{trivector}}. \quad (6)$$

It can easily be verified that the trivector or pseudoscalar $\sigma_1\sigma_2\sigma_3$ squares to $-1$ and commutes with all multivectors in the 3-D algebra. We therefore give it the symbol $I$, noting that this is not the uninterpreted commutative scalar imaginary $j$ used in quantum mechanics and engineering.

### 2.4. Rotors

Multiplication of the three basis vectors $\sigma_1$, $\sigma_2$, and $\sigma_3$ by $I$ results in the three basis bivectors $\sigma_1\sigma_2 = I\sigma_3$, $\sigma_2\sigma_3 = I\sigma_1$, and $\sigma_3\sigma_1 = I\sigma_2$. These simple bivectors rotate vectors in their own plane by 90°, e.g., $(\sigma_1\sigma_2)\sigma_2 = \sigma_1$, $(\sigma_2\sigma_3)\sigma_2 = -\sigma_3$, etc. Identifying $i$, $j$, $k$ of the quaternion algebra with $I\sigma_1$, $-I\sigma_2$, $I\sigma_3$, the famous Hamilton relations $i^2 = j^2 = k^2 = ijk = -1$ can be recovered. Since $i$, $j$, $k$ are bivectors, it comes as no surprise that they represent 90° rotations in orthogonal directions and provide a well-suited system for the representation of general 3D rotations (Fig. 2).

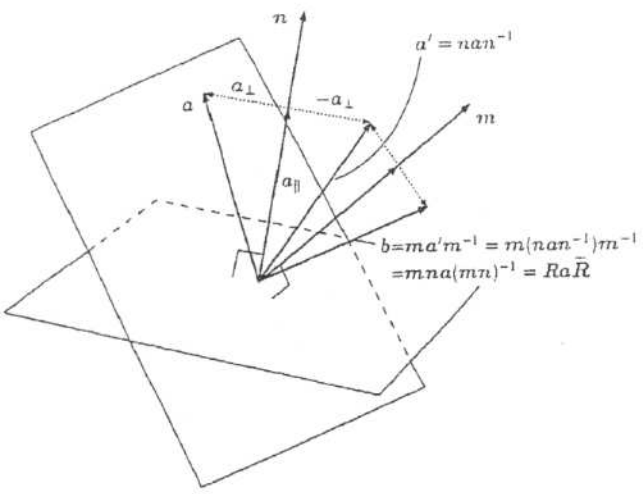


Figure 2. The rotor in the 3D space formed by a pair of reflections.

In geometric algebra a rotor (short name for rotator), $R$, is an even-grade element of the algebra which satisfies $R\tilde{R} = 1$, where $\tilde{R}$ stands for the conjugate of $R$. If $\mathcal{A} = \{a_0, a_1, a_2, a_3\} \in \mathcal{G}_{3,0,0}$ represents a unit quaternion, then the rotor which performs the same rotation is simply given by

$$R = \underbrace{a_0}_{\text{scalar}} + \underbrace{a_1(I\sigma_1) - a_2(I\sigma_2) + a_3(I\sigma_3)}_{\text{bivectors}}. \quad (7)$$

The quaternion algebra is therefore seen to be a subset of the geometric algebra of 3-space.

A rotation can be performed by a pair of reflections (Fig. 2). It can easily be shown that the result of reflecting a vector $a$ in the plane perpendicular to a unit vector $n$ is $a_\perp - a_\parallel = a' = -nan^{-1}$, where $a_\perp$ and $a_\parallel$, respectively, denote projections of $a$ perpendicular and parallel to $n$. Thus, a reflection of $a$ in the plane perpendicular to $n$, followed by a reflection in the plane perpendicular to another unit vector $m$, gives the new vector $b = -m(-nan^{-1})m^{-1} = (mn)a(mn)^{-1} = Ra\tilde{R}$. Using the geometric product, we can show that the rotor $R$ of Eq. (7) is a multivector consisting of both a scalar part and a bivector part, i.e., $R = mn = m \cdot n + m \wedge n$. These components correspond to the scalar and vector parts of an equivalent unit quaternion in $\mathcal{G}_{3,0,0}$. Considering the scalar and the bivector parts, we can further write the Euler representation of a rotor as follows

$$R = e^{n(\theta/2)} = \cos\frac{\theta}{2} + n\sin\frac{\theta}{2}, \quad (8)$$

where the rotation axis $n = n_1\sigma_2\sigma_3 + n_2\sigma_3\sigma_1 + n_3\sigma_1\sigma_2$ is spanned by the bivector basis.

The transformation in terms of a rotor $a \mapsto Ra\tilde{R} = b$ is a very general way of handling rotations; it works for multivectors of any grade and in spaces of any dimension, in contrast to quaternion calculus. Rotors combine in a straightforward manner, i.e., a rotor $R_1$ followed by a rotor $R_2$ is equivalent to a total rotor $R$ where $R = R_2 R_1$.

## 3. THE MOTOR ALGEBRA

Clifford introduced the motors with the name biquaternions.[17] Motor is the abbreviation of "moment and rotor." Motors are the dual numbers for 3D kinematics with the necessary condition of $I^2 = 0$. They can be found in the special 4D even subalgebra of $\mathcal{G}_{3,0,1}$ or motor algebra. This even subalgebra denoted by $\mathcal{G}^+_{3,0,1}$ is spanned by a basis of scalars, bivectors, and pseudoscalars,

$$\underbrace{1}_{\text{scalar}}, \underbrace{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2, \gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3}_{\text{6 bivectors}}, \underbrace{I}_{\text{unitpseudoscalar}}. \quad (9)$$

Note that the bivectors in the basis correspond to the same basis for spanning 3D lines. Also note that the dual of a scalar is the pseudoscalar $P$ and the duals of the first three basis bivectors are the next three; for example, the dual of $\gamma_2\gamma_3$ is $I\gamma_2\gamma_3$ or $\gamma_4\gamma_1$.

### 3.1. Motors, Rotors, and Translators in $\mathcal{G}^+_{3,0,1}$

Since a rigid motion consists of a rotation and a translation, it should be possible to express a motor in terms of individual operators like rotors and translators. The motor action can be described basically in terms of two steps: rotate one axis until its direction is parallel to another axis and then shift it to overlap the another one (Fig. 3). Note that the vectors indicated in the figure will be represented in next sections as bivectors.

Let us now express this procedure algebraically. First, let us consider a simple rotor in its Euler representation for a rotation with angle $\theta$. The rotor of a screw motion should be represented in terms of

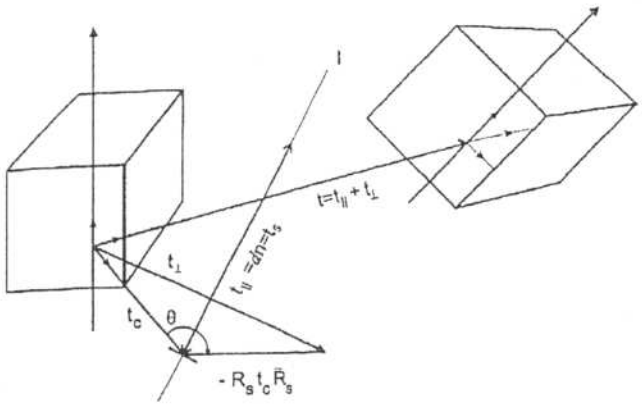


Figure 3. Screw motion of an object about the line axis $l$ with $t_s$ longitudinal displacement by $d$ and rotation $R_s$ with angle $\theta$.

screw axis line as follows

$$\begin{aligned} R_s &= a_0 + a_s n + I a_s n \wedge t_c \\ &= a_c + a_s(n + Im) = \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right)(n + Im) \\ &= \cos\left(\frac{\theta}{2}\right) + \sin\left(\frac{\theta}{2}\right) l. \end{aligned} \tag{10}$$

Note that the line is expressed using a bivector for direction $n$ and dual bivector for the momentum $m = n \wedge t_c$.

The motor is defined by sliding along the rotation axis line $l$ the distance $t_s = dn$. Since a motor is applied from the left and its conjugated form from the right we use the half of $t_s$ when we define the motor

$$\begin{aligned} M = T_s R_s &= \left(1 + I\frac{t_s}{2}\right)(a_0 + a + Ia \wedge t_c) \\ &= \left(1 + I\frac{dn}{2}\right)(a_c + a_s n + I a_s n \wedge t_c) \\ &= a_c + a_s n + I a_s n \wedge t_c + I\frac{d}{2} a_c n - I\frac{d}{2} a_s nn \\ &= \left(a_c - I\frac{d}{2} a_s\right) + \left(a_s + I a_c \frac{d}{2}\right)(n + In \wedge t_c) \\ &= \left(a_c - I a_s \frac{d}{2}\right) + \left(a_s + I a_c \frac{d}{2}\right) l. \end{aligned} \tag{11}$$

Note that this expression of the motor makes explicit the unitary screw axis line $l$. Now let us express a motor as an Euler representation. Substituting the constants $a_c = \cos(\theta/2)$ and $a_s = \sin(\theta/2)$ in the motor Eq. (11) and using the property of scalar functions with dual argument we get

$$\begin{aligned} M = T_s R_s &= \left(\cos\left(\frac{\theta}{2}\right) - I \sin\left(\frac{\theta}{2}\right)\frac{d}{2}\right) \\ &\quad + \left(\sin\left(\frac{\theta}{2}\right) + I \cos\left(\frac{\theta}{2}\right)\frac{d}{2}\right) l \\ &= \cos\left(\frac{\theta}{2} + I\frac{d}{2}\right) + \sin\left(\frac{\theta}{2} + I\frac{d}{2}\right) l. \end{aligned} \tag{12}$$

If we want to express the motor using only a rotor and its conjugated form given by

$$\tilde{R} = r_0 - r_1\sigma_2\sigma_3 - r_2\sigma_3\sigma_1 - r_3\sigma_1\sigma_2 = r_0 - r, \tag{13}$$

we proceed as follows

$$M = T_s R_s = \left(1 + I\frac{t_s}{2}\right) R_s = R_s + I\frac{t_s}{2} R_s = R_s + IR'_s. \tag{14}$$

We can now express the bivector $t_s$ in terms of the rotors

$$R'_s \tilde{R}_s = \left(\frac{t_s}{2} R_s\right) \tilde{R}_s \tag{15}$$

so that

$$t_s = 2 R'_s \tilde{R}. \tag{16}$$

Figure 3 shows that t is a 3D distance vector referred to the coordinate system of the rotation axis of an object and $t_\parallel$ or $t_s$ is a bivector along the motor axis line. The distance $t$, considered here as a bivector, can be computed in terms of the bivectors $t_c$ and $t_s$ as follows:

$$\begin{aligned} t &= t_\perp + t_\parallel = t_\perp + t_s \\ &= \left(t_c - R_s t_c \tilde{R}_s\right) + (t \cdot n) n \\ &= \left(t_c - R_s t_c \tilde{R}_s\right) + dn \\ &= t_c - R_s t_c \tilde{R}_s + t_s \\ &= \left(t_c - R_s t_c \tilde{R}_s\right) + \left(2 R'_s \tilde{R}_s\right). \end{aligned} \tag{17}$$

### 3.2. Properties of Motors

A general motor can be expressed as

$$M_\alpha = \alpha M \tag{18}$$

where $\alpha \in R$ and $M$ is a unit motor as in the previous sections. The norm of a unit motor $M$ is defined by

$$\begin{aligned} |M| = M\tilde{M} &= T_s R_s \tilde{R}_s \tilde{T}_s \\ &= \left(1 + I\frac{t_s}{2}\right) R_s \tilde{R}_s \left(1 - I\frac{t_s}{2}\right) \\ &= 1 + I\frac{t_s}{2} - I\frac{t_s}{2} = 1, \end{aligned} \tag{19}$$

where $\tilde{M}$ is the conjugate of the motor, defined by

$$\tilde{M} = \widetilde{TR} = \tilde{R}\tilde{T} \tag{20}$$

and 1 is the identity of the motor multiplication. Now using Eq. (14) and considering the unit motor magnitude, we find two useful properties:

$$\begin{aligned} |M| = M\tilde{M} &= (R_s + IR'_s)\left(\tilde{R}_s + I\tilde{R}'_s\right) \\ &= R_s\tilde{R}_s + I\left(\tilde{R}_s R'_s + \tilde{R}_s R_s\right) = 1. \end{aligned} \tag{21}$$

This requires that

$$R_s\tilde{R}_s = 1 \tag{22}$$

$$\tilde{R}_s R'_s + \tilde{R}_s R_s = 2(r_0 r'_0 - \mathbf{r}\cdot\mathbf{r}') = 0. \tag{23}$$

The combination of two rigid motions can be expressed using two motors. The resultant motor describes the overall displacement, namely,

$$\begin{aligned} M_c = M_a M_b &= (R_{s_a} + IR'_{s_a})(R_{s_b} + IR'_{s_b}) \\ &= R_{s_a}R_{s_b} + I(R_{s_a}R'_{s_b} + R'_{s_a}R_{s_b}) \\ &= R_{s_c} + IR'_{s_c}. \end{aligned} \tag{24}$$

Note that pure rotations combine multiplicatively, whereas the dual parts containing the translation combine additively.

Using Eq. (14), let us express a motor in terms of a scalar, bivector, dual scalar, and dual bivector

$$\begin{aligned} M = T_s R_s &= R_s + IR'_s \\ &= (a_0 + a_1\gamma_2\gamma_3 + a_2\gamma_3\gamma_2 + a_3\gamma_2\gamma_1) \\ &\quad + I(b_0 + b_1\gamma_2\gamma_3 + b_2\gamma_3\gamma_2 + b_3\gamma_2\gamma_1) \\ &= (a_0 + a) + I(b_0 + b). \end{aligned} \tag{25}$$

We can use another notation to enhance the components of the real and dual parts of the motor as follows:

$$M = (a_0, a) + I(b_0, b), \tag{26}$$

where each term within the parentheses consists of a scalar part and a 3D bivector.

## 4. Representation of Points, Lines, and Planes

In this section we will model points, lines, and planes in the 4D space. For that we choose the special algebra of the motors $\mathcal{G}^+_{3,0,1}$, which using a bivector basis spans in 4D the line space.

For the case of the point representation, we proceed by embedding a 3D point on the hyperplane $X_4 = 1$; thus the equation of the point $X \in \mathcal{G}^+_{3,0,1}$ reads

$$\begin{aligned} X &= 1 + x_1\gamma_4\gamma_1 + x_2\gamma_4\gamma_2 + x_3\gamma_4\gamma_3 \\ &= 1 + I(x_1\gamma_2\gamma_3 + x_2\gamma_3\gamma_1 + x_3\gamma_1\gamma_2) \\ &= 1 + Ix \equiv (1,0) + I(0,x). \end{aligned} \tag{27}$$

We can see that in this expression the real part consists of the scalar 1 and the dual part of only a 3D bivector.

Since we are working in the algebra $\mathcal{G}^+_{3,0,1}$ spanned only by bivectors and scalars, we can see this special geometric algebra as the appropriate system for line modeling. Opposite to the line representation, the point and the plane are in some sense unsymmetric representations with respect to the scalar and bivector parts. Let us now write the line equation in the degenerated geometric algebra $\mathcal{G}^+_{3,0,1}$. Since the product of the unit pseudoscalar $I = \gamma_1\gamma_2\gamma_3\gamma_4$ with any dual bivectors built from the basis $\{\gamma_4\gamma_1, \gamma_4\gamma_2, \gamma_4\gamma_3\}$ results in zero, we have to select the bivector basis $\{\gamma_2\gamma_3, \gamma_3\gamma_1, \gamma_1\gamma_2\}$ for representing the line

$$L = n + Im, \tag{28}$$

where the bivectors for the line direction and the moment are computed using two bivector points $x_1$ and $x_2$ lying on the line as follows

$$\begin{aligned} n &= (x_2 - x_1) = (x_{21} - x_{11})\gamma_2\gamma_3 \\ &\quad + (x_{22} - x_{12})\gamma_3\gamma_1 + (x_{23} - x_{13})\gamma_1\gamma_2 \\ &= L_{n_1}\gamma_2\gamma_3 + L_{n_1}\gamma_3\gamma_1 + L_{n_3}\gamma_1\gamma_2 \\ m &= x_1 \times x_2 \\ &= (x_{12}x_{23} - x_{13}x_{22})\gamma_2\gamma_3 + (x_{13}x_{21} - x_{11}x_{23})\gamma_3\gamma_1 \\ &\quad + (x_{11}x_{22} - x_{12}x_{21})\gamma_1\gamma_2 \\ &= L_{m_1}\gamma_2\gamma_3 + L_{m_2}\gamma_3\gamma_1 + L_{m_3}\gamma_1\gamma_2 \end{aligned} \tag{29}$$

This line representation using dual numbers is easy to understand and to manipulate algebraically, and it is fully equivalent to the one in terms of Plücker coordinates. Using the notation with parentheses the line equation reads

$$L \equiv (0, n) + I(0, m), \tag{30}$$

where the $n$ and $m$ are spanned with a 3D bivector basis. Fig. 4 depicts the line in terms of dual bivector basis.

For the equation of the plane, we represent its orientation via the bivector $n$ and the outer product between a bivector touching the plane and its orientation $n$. Since this outer product results in a quatrivector, we can express it as the Hesse distance $d = (x \cdot n)$ multiplied by the unit pseudoscalar

$$\begin{aligned} H &= n + x \wedge n = n + I(x \cdot n) = n + Id \\ &\equiv (0, n) + I(d, 0). \end{aligned} \tag{31}$$

Note that the plane equation is the dual of the point equation

$$H = (d + In)^* = (In)^* + (d)^* = n + Id, \tag{32}$$

where we consider instead for the plane orientation the unit bivector $n$ and for the scalar 1 the Hesse distance $d$.

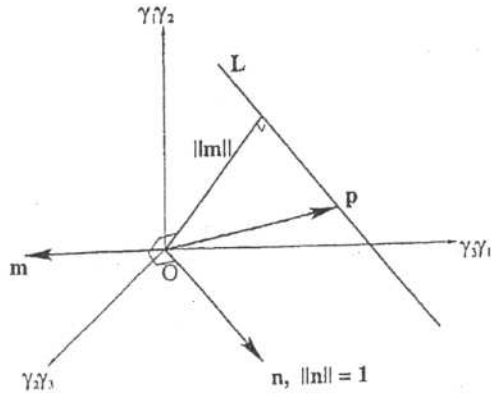


Figure 4. The moment and the direction of the line using dual bivector basis.

## 5. MODELING THE MOTION OF POINTS, LINES, AND PLANES

The modeling of the 3D motion of the geometric primitives using the motor algebra $\mathcal{G}^+_{3,0,1}$ takes place in a 4D space where rotation and translation are operators which are applied multiplicatively, as a result the 3D general motion becomes linear.

### 5.1. Point Motion

For the modeling of the point motion we use the point representation of Eq. (27) and the motor properties with $I^2 = 0$:

$$\begin{aligned} X' &= 1 + Ix' = MX\tilde{M} = M(1 + Ix)\tilde{M} \\ &= T_s R_s (1 + Ix)\tilde{R}_s T_s \\ &= \left(1 + I\frac{t_s}{2}\right) R_s (1 + Ix)\tilde{R}_s \left(1 + I\frac{t_s}{2}\right) \\ &= \left(1 + I\frac{t_s}{2}\right)\left(1 + IR_s x\tilde{R}_s\right)\left(1 + I\frac{t_s}{2}\right) \\ &= 1 + I\frac{t_s}{2} + IR_s x\tilde{R}_s + I\frac{t_s}{2} \\ &= 1 + I\left(R_s x\tilde{R}_s + t_s\right). \end{aligned} \tag{33}$$

Note that the dual part of this equation in the 4D space is fully equivalent to the equation formulated in the geometric algebra of the 3D space[16] $\mathcal{G}_{3,0,0}$.

### 5.2. Line Motion

Using the line Eq. (28) we can express the motion of a line as follows

$$
\begin{aligned}
L' &= n' + Im' = MLM\tilde{} = M(n + Im)M\tilde{} \\
&= T_s R_s (n + Im)\tilde{R}_s \tilde{T}_s \\
&= \left(1 + I\frac{t_s}{2}\right) R_s (n + Im)\tilde{R}_s \left(1 - I\frac{t_s}{2}\right) \\
&= \left(1 + I\frac{t_s}{2}\right)\left(R_s n\tilde{R}_s + IR_s m\tilde{R}_s - IR_s n\tilde{R}_s \frac{t_s}{2}\right) \\
&= R_s n\tilde{R}_s + I\left(R_s n\frac{t_s}{2}\tilde{R}_s + \frac{t_s}{2}R_s n\tilde{R}_s + R_s m\tilde{R}_s\right) \\
&= R_s n\tilde{R}_s + I\left(R_s n\tilde{R}'_s + R'_s n\tilde{R}_s + R_s m\tilde{R}_s\right) \qquad (34)
\end{aligned}
$$

Note that in Eq. (34) before we merge the bivector $t_s/2$ with the rotor $R_s$ or $\tilde{R}_s$, the real and dual parts are fully equivalent with the elements of the line equation[16] formulated in $\mathcal{G}_{3,0,0}$. Figure 5 illustrates the screw motion of the line.

### 5.3. Plane Motion

The motion of a plane in $\mathcal{G}^+_{3,0,1}$ can be seen as the motion of the dual of the point; thus, using the expression of Eq. (31) the motion equation of the plane is

$$
\begin{aligned}
H' &= n' + Id' = MH\tilde{M} = M(n + Id)\tilde{M} \\
&= T_s R_s (n + Id)\tilde{R}_s T_s \\
&= \left(1 + I\frac{t_s}{2}\right)\left(R_s n\tilde{R}_s + Id\right)\left(1 + I\frac{t_s}{2}\right) \\
&= R_s n\tilde{R}_s + I\left(R_s n\tilde{R}_s \frac{t_s}{2} + \frac{t_s}{2}R_s n\tilde{R}_s + d\right) \\
&= R_s n\tilde{R}_s + I\left(t_s \cdot \left(R_s n\tilde{R}_s\right) + d\right). \qquad (35)
\end{aligned}
$$

The real part and the dual part of this expression are equivalent in a higher dimension to the bivector and trivector parts of the equivalent equation[16] formulated in $\mathcal{G}_{3,0,0}$.

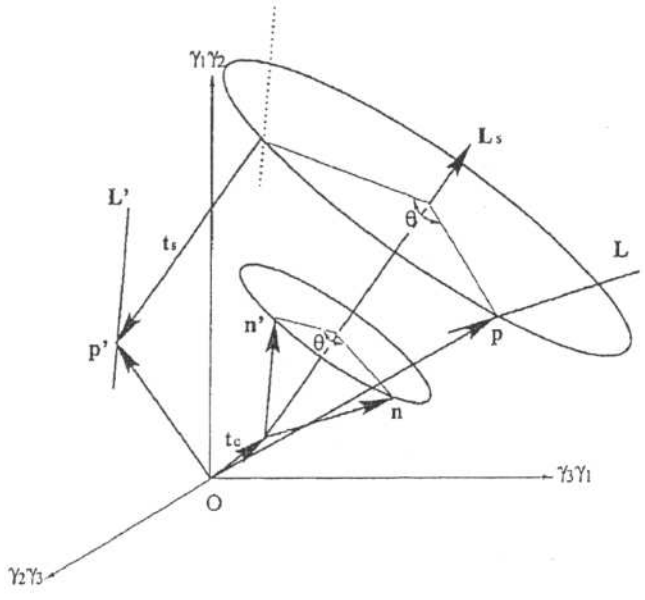


Figure 5. The screw motion of a line.

### 5.4. Dual Quaternions or Motor Algebra

Blaschke[18] was the first to introduce the modeling of points, lines, and planes using purely dual quaternions. The notion of motors was introduced by Clifford,[17] but he did not show the modeling of the motion of points, lines, and planes in terms of motors. Dual quaternions are isomorph with motors; however, due to the geometric meaning of bivectors as rotating planes, we can easily formulate screws using motors from a purely geometric point of view. As a result, the motor algebra modeling of problems involving the algebraic manipulation of points, lines, and planes is simpler than the modeling with the help of dual quaternions. Since motor algebra is a geometric algebra, we can also make use of the concept of duality for the selection and interpretation of 2-blades, as well as the formulation of geometric constraints or flags.

In the last decades many authors[2,4,5,7,19] used dual matrices or dual quaternions for various tasks in robotics. We believe that the use of motor algebra offers one major advantage: it does not make us lose geometric insight of the problem through the computations. Opposite to dual quaternions, the modeling of the motion of points, lines, and planes based on motors is geometrically motivated. The motor algebra is a subalgebra of higher dimension Clifford algebras; thus, we can profit from this fact when we want to access other facilities like the algebra of incidence in the geometric algebra of the 3D affine plane $\mathcal{G}_{4,1,0}$ (here we get a null vector similar to the pseudoscalar of the motor algebra). The computa-

tion with dual matrices adds extra redundant numerical computations (dual matrices have 18 coefficients and a motor only 8) and obscures the geometry of the problem.

In this article we show what many authors have neglected: the use of the plane representation to alleviate the computational burden. Computing the inverse kinematic in section 8, we illustrate clearly that in some parts of the backward computing we can simplify the computations, resorting to a line or a plane representation instead of a point. This fact is a novel and very useful aspect in our approach which can be used for computations in more complex multilink mechanisms or in parallel manipulators.

## 6. ELEMENTARY TRANSFORMATIONS OF ROBOT MANIPULATORS

The study of the rigid motion of objects in 3D space plays an important role in robotics. To linearize the rigid motion of the Euclidean space homogeneous coordinates are normally utilized. That is why we chose 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. 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 we will illustrate an end-effector grasping task.

### 6.1. 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 description approach used most is known as Denavit–Hartenberg procedure.[1] This is based on the uniform description of the position of the reference

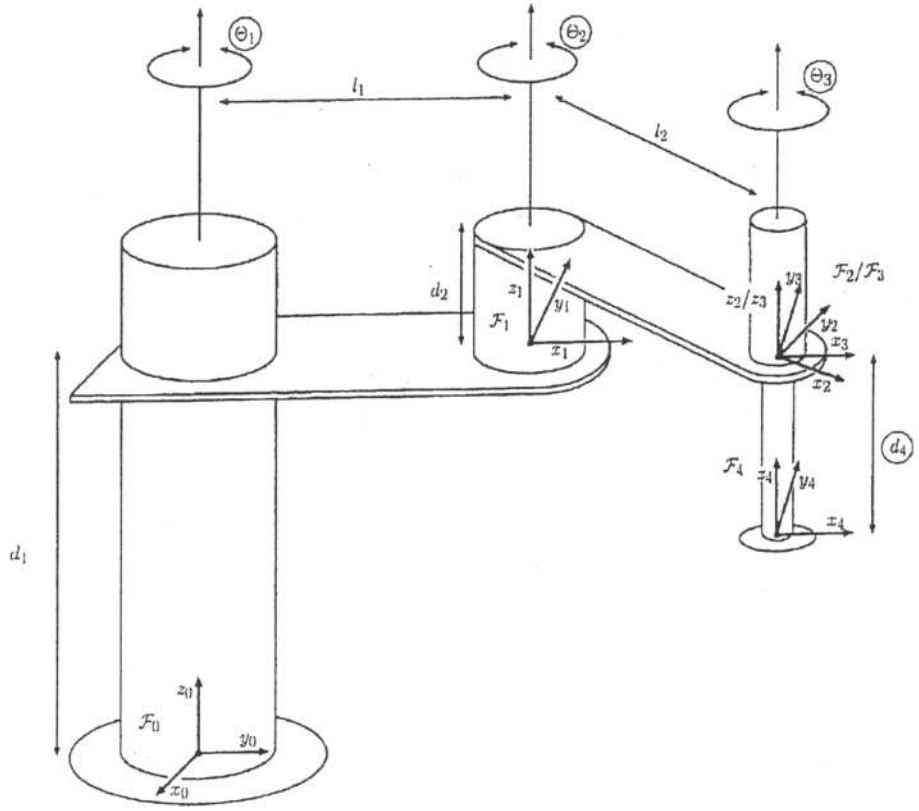


**Figure 6.** SCARA type manipulator according to the DH parameters in Table I. Variable parameters are encircled.

coordinate system of a joint relative to the next one in consideration. Figure 8a shows how coordinate frames are attached to a joint of a robot manipulator. Table I presents the specifications of two robot manipulators: the SCARA and the Stanford as shown in Figures 6 and 7, respectively.

In Table I 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 8b shows the screws involved 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 way we get the direct kinematics straightforwardly.

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

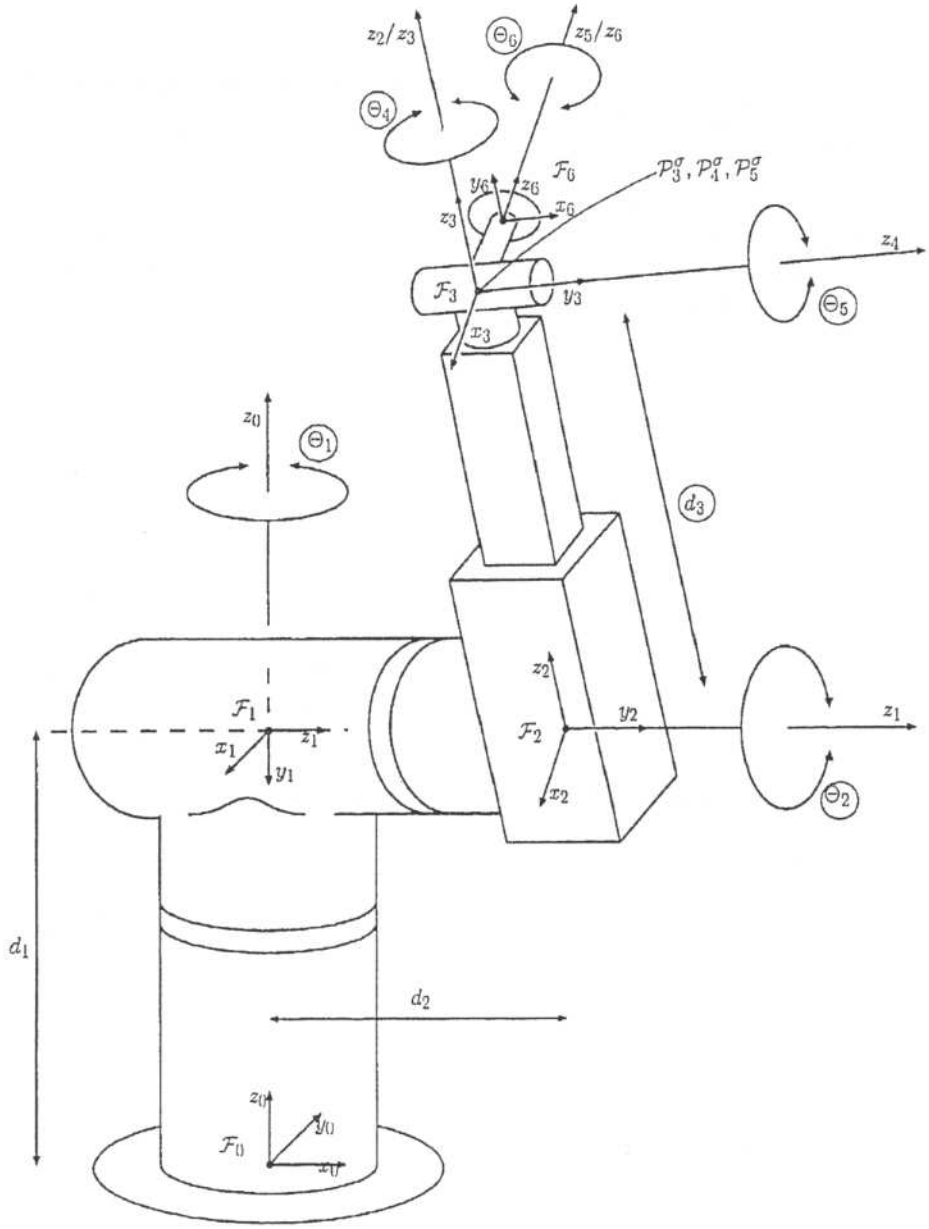


Figure 7. Stanford type manipulator according to the DH parameters in Table I. Variable parameters are encircled.

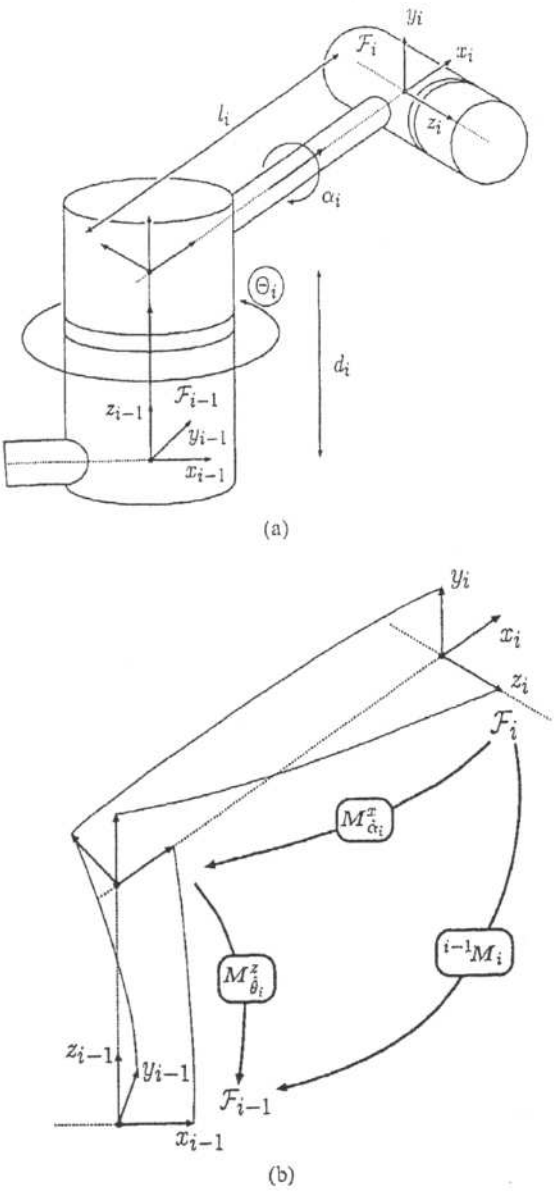


**Figure 8.** (a) $i$th joint of a robot manipulator and the attached coordinate frames according to the Denavit–Hartenberg procedure. Here the encircled $\theta_i$ is the variable parameter. (b) The transformation from frame $\mathcal{F}_i$ to $\mathcal{F}_{i-1}$ is represented by $^{i-1}M_i$. The motor $^{i-1}M_i$ consists of two screw transformations $M^x_{\hat{\alpha}_i}$ and $M^z_{\hat{\theta}_i}$.

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.

### 6.2. Representations of Prismatic and Revolute Transformations

The transformation of any point, lines, 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 $\theta_i$ and a prismatic one when the degree of freedom is only a variable length $d_i$. The transformation motor $^{i-1}M_i$ between $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$ consists of a sequence of two screw transformations, one fixed, i.e., $M^x_{\hat{\alpha}_i}$, and another variable, i.e., $M^z_{\hat{\theta}_i}$ (Fig. 8b). Note that we use dual angles $\hat{\theta}_i = \theta_i + Id_i$ and $\hat{\alpha}_i = \alpha_i + Il_i$. In the revolute case the latter has as a variable parameter the angle $\theta_i$ and in the prismatic case the displacement $d_i$. The transformation reads

$$
\begin{aligned}
^{i-1}M_i &= M^z_{\hat{\theta}_i} M^x_{\hat{\alpha}_i} = T^z_{d_i} R^z_{\theta_i} T^x_{l_i} R^x_{\alpha_i} \\
&= \left(1 + \frac{I}{2}\begin{pmatrix} 0 \\ 0 \\ d_i \end{pmatrix}\right) R^z_{\theta_i} \left(1 + \frac{I}{2}\begin{pmatrix} l_i \\ 0 \\ 0 \end{pmatrix}\right) R^x_{\alpha_i}. \qquad (36)
\end{aligned}
$$

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

Since $^{i-1}M_i\,^{i-1}\tilde{M}_i = 1$, we obtain

$$
^{i}M_{i-1} = \tilde{M}^x_{\hat{\alpha}_i} \tilde{M}^z_{\hat{\theta}_i} = \tilde{T}^x_{l_i} \tilde{R}^x_{\alpha_i} \tilde{T}^z_{d_i} \tilde{R}^z_{\theta_i}. \qquad (37)
$$

For the rest of the paper $^{j}M_i$ 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 $\theta_i$ and $d_i$, respectively, as a variable and with two fixed parameters $\alpha_i$ and $l_i$. In the joint depicted in Figure 8b a revolute transformation will take place only when $\theta_i$ varies and a prismatic transformation only when $d_i$ varies. Now taking a point $X$ 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 with either $\theta_i$ or $d_i$ as a variable parameter. We will call this transformation a *forward transformation.*

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

$$
\begin{aligned}
^{i}X &= {}^{i}M_{i-1}\,{}^{i-1}X\,{}^{i}\bar{\tilde{M}}_{i-1} = \tilde{M}^x_{\hat{\alpha}_i} \tilde{M}^z_{\hat{\theta}_i}\,{}^{i-1}X \overline{M}^z_{\hat{\theta}_i} \overline{M}^x_{\hat{\alpha}_i} \\
&= \tilde{T}^x_{l_i} \tilde{R}^x_{\alpha_i} \tilde{T}^z_{d_i} \tilde{R}^z_{\theta_i}\,{}^{i-1}X R^z_{\theta_i} \tilde{T}^z_{d_i} R^x_{\alpha_i} \tilde{T}^x_{l_i} \\
&= 1 + I\,{}^{i}x, \qquad (38)
\end{aligned}
$$

where $^{i}x$ is a bivector representing the 3D position of $X$ referred to $\mathcal{F}_i$. Thinking in a transformation in

**Table I.** Kinematic configuration of two robot manipulators.

| Robot type | Link | Revolute $\theta_i$ | v/c | Prismatic $d_i$ | v/c | Twist angle $\alpha_i$ | Link length $l_i$ |
|---|---|---|---|---|---|---|---|
| SCARA | 1 | $\theta_1$ | v | $d_1$ | c | 0 | $l_1$ |
| | 2 | $\theta_2$ | v | $d_2$ | c | 0 | $l_2$ |
| | 3 | $\theta_3$ | v | 0 | | 0 | 0 |
| | 4 | 0 | | $d_4$ | v | 0 | 0 |
| Stanford | 1 | $\theta_1$ | v | $d_1$ | c | $-90°$ | 0 |
| | 2 | $\theta_2$ | v | $d_2$ | c | $90°$ | 0 |
| | 3 | 0 | | $d_3$ | v | 0 | 0 |
| | 4 | $\theta_4$ | v | 0 | | $-90°$ | 0 |
| | 5 | $\theta_5$ | v | 0 | | $90°$ | 0 |
| | 6 | $\theta_6$ | v | $d_6$ | c | 0 | 0 |

the reverse sense we call it a *backward transformation* which transforms a point $X$ represented in the frame $\mathcal{F}_i$ to the frame $\mathcal{F}_{i-1}$ as follows

$$
\begin{aligned}
{}^{i-1}X &= {}^{i-1}M_i\,{}^{i}X^{i-1}\bar{M}_i = M_{\theta_i}^{z} M_{\alpha_i}^{x}\,{}^{i}X\bar{M}_{\alpha_i}^{x}\bar{M}_{\theta_i}^{z} \\
&= 1 + I^{i-1}x. \qquad (39)
\end{aligned}
$$

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 the plane.

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

$$
\begin{aligned}
{}^{i}L &= {}^{i}M_{i-1}\,{}^{i-1}L\,{}^{i}\tilde{M}_{i-1} = \tilde{M}_{\alpha_i}^{x}\tilde{M}_{\theta_i}^{z}\,{}^{i-1}L M_{\theta_i}^{z} M_{\alpha_i}^{x} \\
&= {}^{i}n + I^{i}m. \qquad (40)
\end{aligned}
$$

Its backward transformation reads

$$
\begin{aligned}
{}^{i-1}L &= {}^{i-1}M_i\,{}^{i}L^{i-1}\tilde{M}_i = M_{\theta_i}^{z} M_{\alpha_i}^{x}\,{}^{i}L\tilde{M}_{\alpha_i}^{x}\tilde{M}_{\theta_i}^{z} \\
&= {}^{i-1}n + I^{i-1}m. \qquad (41)
\end{aligned}
$$

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

$$
\begin{aligned}
{}^{i}H &= {}^{i}M_{i-1}\,{}^{i-1}H\,{}^{i}\bar{M}_{i-1} = \tilde{M}_{\alpha_i}^{x}\tilde{M}_{\theta_i}^{z}\,{}^{i-1}H\bar{M}_{\theta_i}^{z}\bar{M}_{\alpha_i}^{x} \\
&= {}^{i}n + I^{i}d_H, \qquad (42)
\end{aligned}
$$

**Table II.** Rendezvous equations obtained for $P_3^o$ regarding frames $\mathcal{F}_0$, $\mathcal{F}_1$, $\mathcal{F}_2$, and $\mathcal{F}_3$.

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

and similarly as above, its backward transformation equation is

$$
\begin{aligned}
{}^{i-1}H &= {}^{i-1}M_i\,{}^{i}H^{i-1}\bar{\bar{M}}_i = M_{\theta_i}^{z}\,M_{\alpha_i}^{x}\,{}^{i}H\bar{\bar{M}}_{\alpha_i}^{x}\bar{\bar{M}}_{\theta_i}^{z} \\
&= {}^{i-1}n + I^{i-1}d_H . \qquad (43)
\end{aligned}
$$

### 6.3. Grasping by Using Constraint Equations

In this subsection we will illustrate a manipulation related task of a very simplified grasping operation. This task involves the positioning of a two-finger grasper in front of a static object. Figure 9 shows the grasper and the considered object $O$. The manipulator moves the grasper near to the object and together they should fulfill some conditions to grasp the object firmly. To determine the overall transformation ${}^{0}M_n$, which moves the grasper to an appropriate grasping position, we claim that ${}^{0}M_n$ has to fulfill three constraints. For the formulation of these constraints we can take advantage 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 $O$ in frame $\mathcal{F}_0$ are known.

#### Attitude Condition

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

$$
{}^{0}M_n\,{}^{n}H_n^{yz}\,{}^{0}\bar{\bar{M}}_n - {}^{0}H_O \approx 0, \qquad (44)
$$

where ${}^{n}H_n^{yz} = (1,0,0)^T + I\ 0 = (1,0,0)^T$ (Fig. 9).

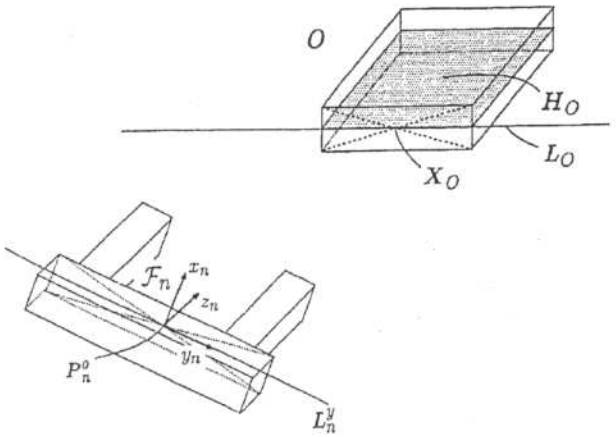


Figure 9. Two finger grasper approaching an object.

#### Alignment Condition

The grasper and object should be aligned parallel after the application of the motor ${}^{0}M_n$. That is, the direction of the $y$ axis and the line $L_O$ should be the same. This condition can be simply expressed in terms of a line equation

$$
\langle {}^{0}M_n\,{}^{n}L_n^{y}\,{}^{0}\bar{M}_n\rangle_d - \langle {}^{0}L_O\rangle_d \approx 0, \qquad (45)
$$

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

#### Touching condition

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

$$
{}^{0}M_n\,{}^{n}P_n^{o}\,{}^{0}\bar{\bar{M}}_n - {}^{0}\mathrm{X}_O \approx 0. \qquad (46)
$$

By these three conditions we get constraints for the components of ${}^{0}M_n$, and we can determine ${}^{0}M_n$ 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 ${}^{0}M_n$. This problem is called the inverse kinematics problem of robot manipulators and will be treated in section 8.

## 7. 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 (Fig. 10).

In this section we will show how the direct kinematics can be computed when we use a point, line, or plane as geometric object. The notation for points, lines, and planes we will use in the next section is illustrated in Figure 11. The direct kinematics for the general case of a manipulator with $n$

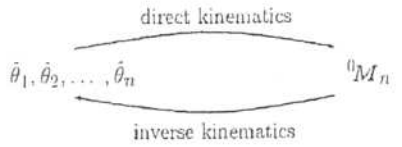


Figure 10. Direct and inverse kinematics.

joints can be written as follows

$$ {}^0M_n = {}^0M_1\,{}^1M_2\,{}^2M_3 \cdots {}^{n-1}M_n = \prod_{i=1}^{n} {}^{i-1}M_i. \quad (47) $$

Now we can straightforwardly formulate the direct kinematics in terms of point, line, or plane representations as follows

$$ \begin{aligned} {}^0X &= {}^0M_n\,{}^nX\,{}^0\tilde{M}_n = \prod_{i=1}^{n} {}^{i-1}M_i\,{}^nX \prod_{i=1}^{n} {}^{n-i}\tilde{M}_{n+1-i}, \\ {}^0L &= \prod_{i=1}^{n} {}^{i-1}M_i\,{}^nL \prod_{i=1}^{n} {}^{n-i}\tilde{M}_{n+1-i}, \\ {}^0H &= \prod_{i=1}^{n} {}^{i-1}M_i\,{}^nH \prod_{i=1}^{n} {}^{n-i}\tilde{M}_{n+1-i}. \end{aligned} \quad (48) $$

Let us now write the motor ${}^0M_4$ for the direct kinematics for points, lines, and planes like Eq. (48) for the SCARA manipulator specified by the Denavit–Hartenberg parameters of Table I. First, using Eq. (47) with $n = 4$, we can write down straightforwardly the required motor ${}^0M_4$ as follows

$$ \begin{aligned} {}^0M_4 &= {}^0M_1\,{}^1M_2\,{}^2M_3\,{}^3M_4 = \left(M_{\theta_1}^{z} M_{\alpha_1}^{x}\right) \cdots \left(M_{\theta_4}^{z} M_{\alpha_4}^{x}\right) \\ &= \left(T_{d_1}^{z} R_{\theta_1}^{z} T_{l_1}^{x} R_{\alpha_1}^{x}\right) \cdots \left(T_{d_4}^{z} R_{\theta_4}^{z} T_{l_4}^{x} R_{\alpha_4}^{x}\right) \\ &= \left(1 + \frac{I}{2}\begin{pmatrix} 0 \\ 0 \\ d_1 \end{pmatrix}\right) R_{\theta_1}^{z} \left(1 + \frac{I}{2}\begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix}\right) \left(1 + \frac{I}{2}\begin{pmatrix} 0 \\ 0 \\ d_2 \end{pmatrix}\right) \\ &\quad R_{\theta_2}^{z} \left(1 + \frac{I}{2}\begin{pmatrix} l_2 \\ 0 \\ 0 \end{pmatrix}\right) R_{\theta_3}^{z} \left(1 + \frac{I}{2}\begin{pmatrix} 0 \\ 0 \\ d_4 \end{pmatrix}\right). \end{aligned} \quad (49) $$

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

Applying the motor ${}^0M_4$ from the left and ${}^0\tilde{M}_4$ from the right for point and plane equations and the motor ${}^0M_4$ from the left and ${}^0\tilde{M}_4$ from the right for line equations as indicated by Eqs. (48), we get the direct kinematics equations of points, lines, and planes for the SCARA robot manipulator.

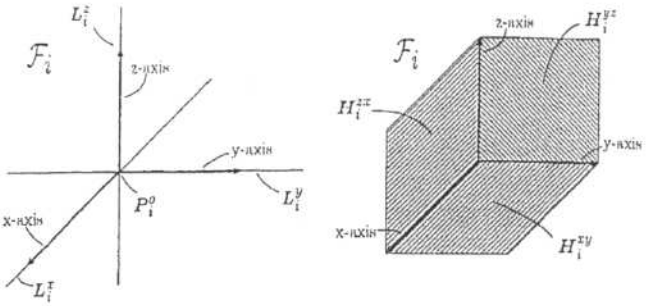


Figure 11. Notations for frame specific entities as the origin, the coordinate axis, and coordinate planes.

### 7.1. 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 $P_4^o$ of $\mathcal{F}_4$ for the SCARA manipulator specified by the Denavit–Hartenberg parameters of Table I. Figure 12 shows the frames and the point $P_4^o$ referred to $\mathcal{F}_0$. The final result is

$$ \begin{aligned} {}^0P_4^o &= {}^0M_4\,{}^4P_4^o\,{}^0\tilde{M}_4 = {}^0M_4\left(1 + I\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}\right){}^0\tilde{M}_4 \\ &= 1 + I\begin{pmatrix} l_2\cos(\theta_1+\theta_2) + l_1\cos(\theta_1) \\ l_2\sin(\theta_1+\theta_2) + l_1\sin(\theta_1) \\ d_1 + d_2 + d_4 \end{pmatrix}. \end{aligned} \quad (50) $$

## 8. 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

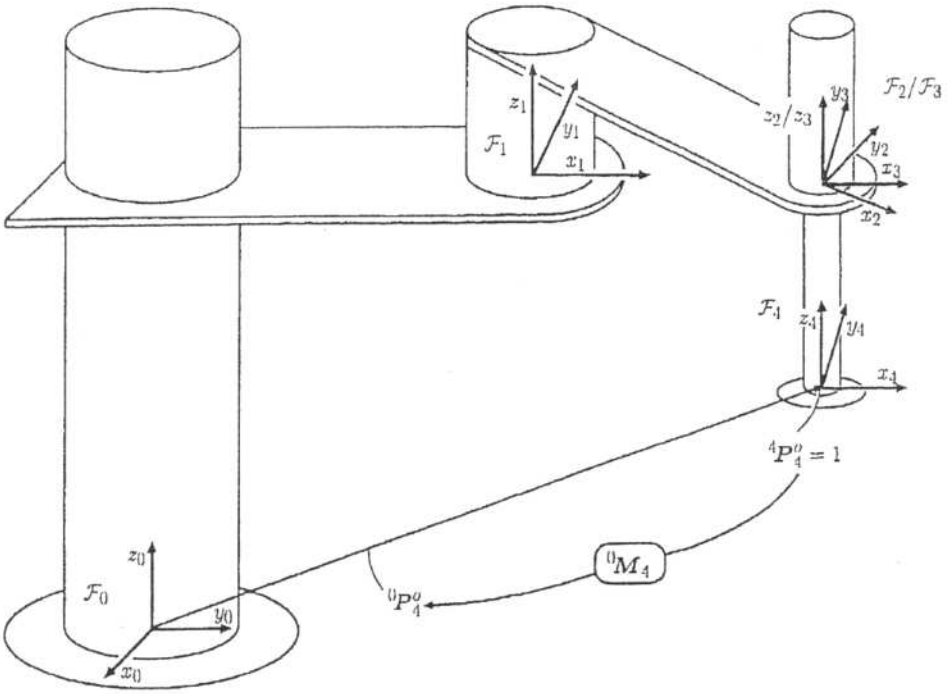


**Figure 12.** The representation ${}^0P_4^o$ of $P_4^o$ in frame $\mathcal{F}_0$ is computed using ${}^0M_4$.

formulation as in the case of the direct kinematics Eq. (47). That is why we better choose a real robot manipulator and compute its inverse kinematics 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 I the variable parameters to be computed are $\theta_1$, $\theta_2$, $\theta_4$, $\theta_5$, $\theta_6$, 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 4–6. Since the philosophy of our approach relies on the application of point, line, or plane representation where it is needed, we should first 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 the 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.

### 8.1. 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 is depicted in Figure 13. 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*. Slightly similar procedures have been presented in the past,[7,19] however, either using simply points or only lines. In contrast, our approach uses equations of points, lines, and planes. The simple idea of the rendezvous method has proved to be very useful as a strategy for the solution of the inverse kinematics. The

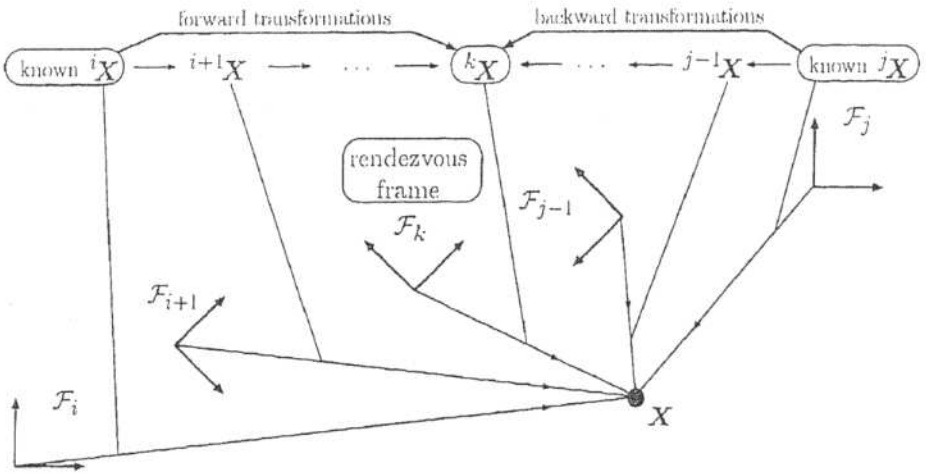


**Figure 13.** Rendezvous method: If ${}^{i}X$ and ${}^{j}X$ are known, we can compute ${}^{k}X$ for each $i \leq k \leq j$ in two different ways: by successive forward transformations of ${}^{i}X$ and by successive backward transformation of ${}^{j}X$.

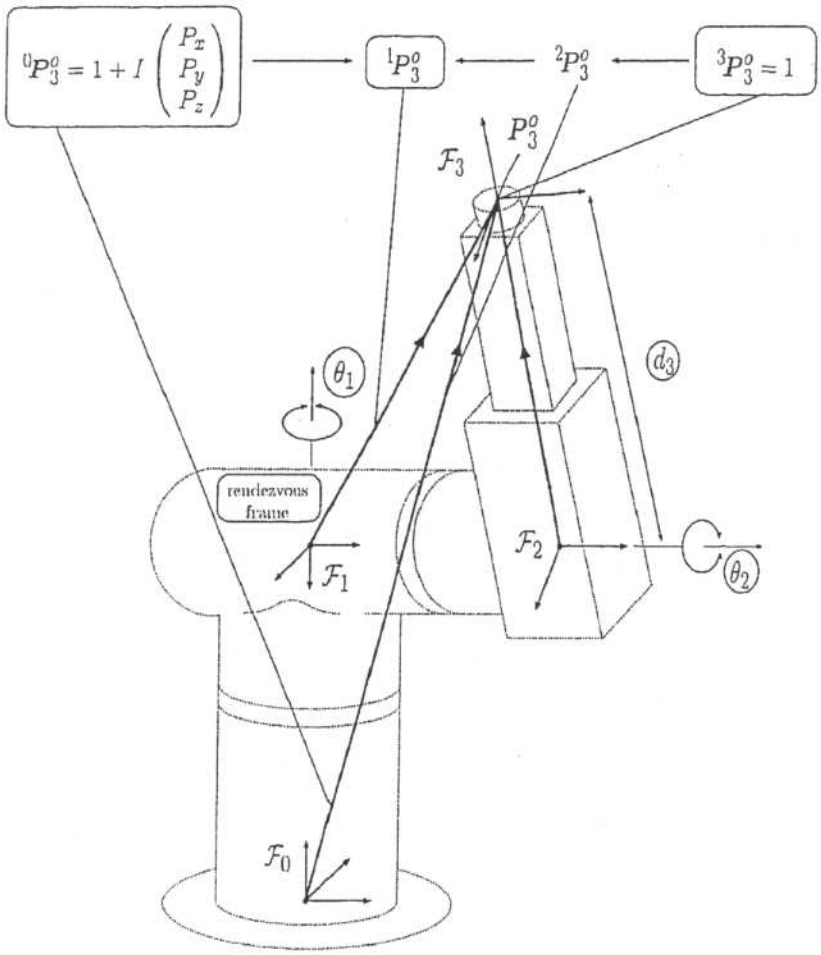


**Figure 14.** The rendezvous method applied to $P_3^o$ to determine the equations shown in Table II. The equations of rendezvous frame $\mathcal{F}_1$ are chosen to compute the variable parameters $\theta_1$, $\theta_2$, and $d_3$.

searching of meeting points helps to reduce the complexity of equations systems. This approach can be extended for dealing with complex linked mechanisms.

### 8.2. Computing $\theta_1$, $\theta_2$, and $d_3$ Using a Point Representation

In the case of the Stanford manipulator the orientation and position of frame $\mathcal{F}_6$ uniquely determine 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 ${}^0P_3^o$ of $P_3^o$ in $\mathcal{F}_0$. By successive forward transformation applied on ${}^3P_3^o = 1$ we get the representation ${}^6P_3^o$ of $P_3^o$ in $\mathcal{F}_6$ by

$$ {}^6P_3^o = {}^6M_3\,{}^3P_3^o\,{}^6\bar{\bar{M}}_3 = 1 - I\begin{pmatrix} 0 \\ 0 \\ d_6 \end{pmatrix}. \tag{51} $$

Now we can compute ${}^0P_3^o$ by

$$ {}^0P_3^o = {}^0M_6\,{}^6P_3^o\,{}^0\bar{\bar{M}}_6 = {}^0M_6\left(1 - I\begin{pmatrix} 0 \\ 0 \\ d_6 \end{pmatrix}\right){}^0\bar{\bar{M}}_6 $$

$$ = 1 + I\begin{pmatrix} P_x \\ P_y \\ P_z \end{pmatrix}. \tag{52} $$

Note that ${}^0M_6$ is given. The vector $(P_x, P_y, P_z)^T$ describes the position of the origin $P_3^o$ of frame $\mathcal{F}_3$ in frame $\mathcal{F}_0$ for a given overall transformation ${}^0M_6$. Now we can apply the rendezvous method since we know the representation of $P_3^o$ in the two different frames $\mathcal{F}_0$ and $\mathcal{F}_3$ (Fig. 14).

Applying successive forward transformations we obtain

$$ \begin{aligned} {}^1P_3^o &= {}^1M_0\,{}^0P_3^o\,{}^1\bar{\bar{M}}_0, \\ {}^2P_3^o &= {}^2M_1\,{}^1P_3^o\,{}^2\bar{\bar{M}}_1, \\ {}^3P_3^o &= {}^3M_2\,{}^2P_3^o\,{}^3\bar{\bar{M}}_2. \end{aligned} \tag{53} $$

These computations were carried out with our Maple program getting the left-hand sides of the four groups of equations of Table II.

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

$$ {}^3P_3^o = 1 + I\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 1, \tag{54} $$

we get

$$ \begin{aligned} {}^2P_3^o &= {}^2M_3\,{}^3P_3^o\,{}^2\bar{\bar{M}}_3 = 1 + I\begin{pmatrix} 0 \\ 0 \\ d_3 \end{pmatrix}, \\ {}^1P_3^o &= {}^1M_2\,{}^2P_3^o\,{}^1\bar{\bar{M}}_2 = 1 + I\begin{pmatrix} d_3\sin(\theta_2) \\ -d_3\cos(\theta_2) \\ d_2 \end{pmatrix}, \\ {}^0P_3^o &= {}^0M_1\,{}^1P_3^o\,{}^0\bar{\bar{M}}_1 \\ &= 1 + I\begin{pmatrix} d_3\sin(\theta_2)\cos(\theta_1) - d_2\sin(\theta_1) \\ d_3\sin(\theta_2)\sin(\theta_1) + d_2\cos(\theta_1) \\ d_3\cos(\theta_2) + d_1 \end{pmatrix}. \end{aligned} \tag{55} $$

These equations correspond to the right-hand sides of the four groups of equations of Table II. 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 compute

$$ \theta_1 = \arctan_2(x_{1/2}, y_{1/2}), \tag{56} $$

where

$$ \begin{aligned} x_{1/2} &= \frac{d_2 - P_y y_{1/2}}{-P_x}, \\ y_{1/2} &= \frac{P_y d_2 \pm P_x\sqrt{P_x^2 + P_y^2 - d_2^2}}{P_x^2 + P_y^2} \end{aligned} \tag{57} $$

and

$$ \arctan_2(x, y) = \begin{cases} \arctan(\frac{x}{y}) & : \ y > 0 \\ \frac{\pi}{2} & : \ y = 0 \text{ and } x > 0 \\ \text{undefined} & : \ y = 0 \text{ and } x = 0 \\ -\frac{\pi}{2} & : \ y = 0 \text{ and } x < 0 \\ \arctan(\frac{x}{y}) + \pi & : \ y < 0. \end{cases} \tag{58} $$

This gives two values for $\theta_1$. Now let us look for $d_3$ and $\theta_2$. 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

$$d_{3_{1/2}} = \sqrt{a_{1/2}^2 + b^2}\,. \tag{59}$$

Using this value in the first and second equations of Table II, we compute straightforwardly

$$\theta_2 = \arctan_2\left(\frac{a_{1/2}}{d_{3_{1/2}}}, \frac{b}{d_{3_{1/2}}}\right). \tag{60}$$

### 8.3. Computing $\theta_4$ and $\theta_5$ 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 ${}^0L_6^z$ of the line $L_6^z$ in frame $\mathcal{F}_0$ can be computed using ${}^0M_6$

$$\begin{aligned} {}^0L_6^z &= {}^0M_6\,{}^6L_6^z\,{}^0\tilde{M}_6 \\ &= {}^0M_6\left(\begin{pmatrix}0\\0\\1\end{pmatrix} + I\begin{pmatrix}0\\0\\0\end{pmatrix}\right){}^0\tilde{M}_6. \end{aligned} \tag{61}$$

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 the $\mathcal{F}_3$ frame is

$${}^3L_6^z = {}^3M_0\,{}^0L_6^z\,{}^3\tilde{M}_0 = \begin{pmatrix}A_x\\A_y\\A_z\end{pmatrix} + I\begin{pmatrix}0\\0\\0\end{pmatrix}. \tag{62}$$

Note that ${}^3M_0$ is known since we have already computed $\theta_1$, $\theta_2$, and $d_3$.

Now applying successively forward transformations as follows

$$\begin{aligned} {}^4L_6^z &= {}^4M_3\,{}^3L_6^z\,{}^4\tilde{M}_3, \\ {}^5L_6^z &= {}^5M_4\,{}^4L_6^z\,{}^5\tilde{M}_4, \\ {}^6L_6^z &= {}^6M_5\,{}^5L_6^z\,{}^6\tilde{M}_5, \end{aligned} \tag{63}$$

we get the left-hand sides of the four groups of equations of Table III. The $z$ axis line $L_6^z$ of $\mathcal{F}_6$ represented in $\mathcal{F}_6$ has zero moment; thus, it can be expressed as

$${}^6L_6^z = \begin{pmatrix}0\\0\\1\end{pmatrix} + I\begin{pmatrix}0\\0\\0\end{pmatrix}. \tag{64}$$

Now applying successive backward transformations, we have

$$\begin{aligned} {}^5L_6^z &= {}^5M_6\,{}^6L_6^z\,{}^5\tilde{M}_6, \\ {}^4L_6^z &= {}^4M_5\,{}^5L_6^z\,{}^4\tilde{M}_5, \\ {}^3L_6^z &= {}^3M_4\,{}^4L_6^z\,{}^3\tilde{M}_4, \end{aligned} \tag{65}$$

Using our Maple program, we compute the right-hand sides of the four groups of equations of Table III.

We will consider the equations of rendezvous frame $\mathcal{F}_4$. Using the third equation, we compute

$$\theta_4 = \arctan_2(x_{1/2}, y_{1/2}), \tag{66}$$

where

$$\begin{aligned} x_{1/2} &= -\frac{A_y y_{1/2}}{-Ax} = \pm\frac{A_y}{\sqrt{A_x^2 + A_y^2}}, \\ y_{1/2} &= \pm\frac{A_x}{\sqrt{A_x^2 + A_y^2}}. \end{aligned} \tag{67}$$

This results in two values for $\theta_4$, which substituted in the first and second equations of Table III, help us to find two solutions for $\theta_5$

$$\begin{aligned} \theta_5 &= \arctan_2(s_5, c_5) \\ &= \arctan_2\big((-A_y s_4 - A_x c_4), -A_z\big). \end{aligned} \tag{68}$$

### 8.4. Computing $\theta_6$ Using a Plane Representation

Since $\theta_1$, $\theta_2$, $d_3$, $\theta_4$, and $\theta_5$ are now known, we can compute the motor ${}^5M_0$. The $yz$ plane $H_6^{yz}$ represented in $\mathcal{F}_6$ has the Hesse distance 0; thus,

$${}^6H_6^{yz} = \begin{pmatrix}1\\0\\0\end{pmatrix} + I0 = \begin{pmatrix}1\\0\\0\end{pmatrix}. \tag{69}$$

Table III. Rendezvous equations obtained for $L_6^z$ 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 |

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

$$ {}^0H_6^{yz} = {}^0M_6\, {}^6H_6^{yz}\, {}^0\bar{\bar{M}}_6 = {}^0M_6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} {}^0\bar{\bar{M}}_6. \quad (70) $$

Now we compute ${}^5H_6^{yz}$ by

$$ {}^5H_6^{yz} = {}^5M_0\, {}^0H_6^{yz}\, {}^5\bar{\bar{M}}_0 = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} + I^5 d_{H_6^{yz}}. \quad (71) $$

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

$$ {}^6H_6^{yz} = {}^6M_5\, {}^5H_6^{yz}\, {}^6\bar{\bar{M}}_5. \quad (72) $$

Table IV. Rendezvous equations obtained for $H_6^{yz}$ regarding frames $\mathcal{F}_5$ and $\mathcal{F}_6$.

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

Using our Maple program, we get the left hand sides of the two groups of equations of Table IV. Since the values for $\theta_1$, $\theta_2$, $d_3$, $\theta_4$, and $\theta_5$ are not unique, we will get different values for the equations. Applying ${}^5M_6$ to ${}^6H_6^{yz}$ we get the right hand sides of the two groups of equations of Table IV by

$$ {}^5H_6^{yz} = {}^5M_6\, {}^6H_6^{yz}\, {}^5\bar{\bar{M}}_6 = {}^5M_6 \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} {}^5\bar{\bar{M}}_6 = \begin{pmatrix} \sin(\theta_6) \\ \cos(\theta_6) \\ 0 \end{pmatrix}. \quad (73) $$

We will consider the equations of the rendezvous frame $\mathcal{F}_5$. Using the first and second equations, we can compute $\theta_6$ by

$$ \theta_6 = \arctan_2(s_6, c_6) = \arctan_2(N_x, N_y). \quad (74) $$

Note that since we had two values for $\theta_4$ and two values for $\theta_5$, there is more than one solution for $\theta_6$.

### 8.5. Multiple-Link and Cooperative Mechanisms

In this subsection we will briefly give some comments related to the possible use of the motor algebra for the understanding and handle of multiple-link mechanisms and cooperative operation of multiple manipulators. The use of motor algebra for such applications is, in fact, an important issue for future research. In that regard, the importance of this paper is twofold: it gives to the novice the basics of motor algebra and shows to everybody how we can exploit the computational advantages

of the motor algebra without losing the geometry of the problem. This aspect is crucial for high level reasoning utilizing geometric objects. Researchers can now start to apply motor algebra to handle more complicated robot mechanisms.

Motor algebra is an even 4D geometric algebra equipped with the geometric product, geometric interpretation of rotors and translators, and a useful concept of duality. The fact that the 3D rigid motion is linearized using multiplicative rotors and translators helps to reduce the complexity of the computation. As the architecture and dynamics of parallel manipulators are in a certain sense related to serial-link manipulators, most of the theoretical problems can be reconsidered using the motor algebra in the light of the duality concept.

Cooperative operation schemes of multiple manipulators have become popular in industrial automation, outer space, and hazardous terrestrial applications. Requirements for increased speeds of operation and light-weight design of robot manipulators show that *structural flexibility* becomes a dominant factor in the design and control of cooperating manipulator systems. In this regard, we believe that the motor algebra gives the approach flexibility due to the alternative use of points, lines, and planes. Since motor algebra is a coordinate-free system, we can attach observers to deal locally with the dynamics of points, lines and planes. Furthermore, extending the signature from $\mathcal{G}^+_{3,0,1}$ to the geometric algebra of the 3D affine plane $\mathcal{G}_{4,1,0}$,[20] we can use rotors and translators effectively the same as motors and the algebra of incidence operations of *join* and *meet* for computing the join and the intersection of points, lines, and planes. These algebra of incidence operations help enormously for the formulation of geometric constraints and flags.

In this article we have not discussed the dynamics of rigid objects; however, its formulation is straightforward[21] and briefly follows. In the motor algebra the motion of a point of a rigid body given by Eq. (33) obeys the kinematic equation

$$\dot{M} = \tfrac{1}{2} V M \tag{75}$$

with $V = -iw + vI$, where $i = \sigma_1\sigma_2\sigma_3$ (pseudoscalar of $\mathcal{G}^3$), $w$ is the angular velocity of the body, and $v$ is assumed as the velocity of the center of mass. Accordingly it follows that $\dot{x} = V \cdot x$ and $\dot{x} = w \times x + v$. We are now ready to define basic equations for dynamics.

The *co-momentum* is defined as

$$P = M_I(V) = iI_I(w) + Imv = iu + Ip, \tag{76}$$

where for simple notation we introduce a "mass transformation" $M_I(\ )$ in terms of the"inertia transformation" $I_I(\ )$ and the body mass $m$.

The *coforce* or *wrench* $W$ acting on a rigid body is defined in terms of the torque $\Gamma$ and a net force $f$ as follows:

$$W = i\Gamma + If. \tag{77}$$

Thus, the compact dynamical equation for rotational and translational motion is given by the time derivative

$$\dot{P} = \frac{dP}{dt} = W \tag{78}$$

and the conservation of energy law is given by

$$\dot{K} = V \cdot W = w \cdot \Gamma + v \cdot f \tag{79}$$

for the kinetic energy

$$K = \tfrac{1}{2} V \cdot P = \tfrac{1}{2}(w \cdot u + v \cdot p). \tag{80}$$

Equation (33) is related to a particular reference frame; any displacement of the frame changes the velocity and co-momentum of the object as follows:

$$V' = MV\bar{M} \tag{81}$$

$$P = \bar{M}P'M. \tag{82}$$

Note that since $V$ is covariant and $P$ is contravariant, the dot product remains invariant:

$$\begin{aligned} P' \cdot V' &= (MV\bar{M}) \cdot (MP\bar{M}) \\ &= M(P \cdot V)\bar{M} = P \cdot V. \end{aligned} \tag{83}$$

The formulae presented above should be useful for the treatment of problems involving actuator torques in a closed-loop chain structure as in the case of multiple manipulators acting cooperatively on an rigid object. Essentially, redundant actuation is one of the inherent characteristics of such systems. The computation of actuator torques, necessary to achieve a prescribed object motion, is known as the *inverse dynamics process*. Due to the presence of redundant actuators, inverse dynamics torques for cooperating manipulator systems admit, unfortu-

nately, an infinite number of solutions. Furthermore, the consideration of flexibility in the links of manipulators restricted to the applications space not only complicates the modeling of the dynamics of the system but also introduces instability in the inverse dynamics solution.

It seems natural to partition the problem into two separate parts: the force distribution problem and the inverse dynamics problem for serial flexible-link manipulators. Since motor algebra linearizes by means of an economic representation of the displacement transformation, we could formulate the force distribution problem as a linearly constrained local optimization problem. Note that all the entities given in section 5 are multivectors with only 8 coefficients. The traditional use of matrices with unnecessary redundant coefficients is a burden for the computation. On the other hand, an approach based on distributed force with coordinate-free observers guarantees the stable behavior of the internal dynamics system. One way to reduce the number of elastic coordinates of the system is the search for redundant actuators or the use within the computation duals of the serial manipulators where the alternative use of points, lines, or planes and their constraints gained by *join* and *meet* operations helps to simplify the problem. As a result we gain systems of equations; here, the rendezvous method can be helpful as a natural strategy for solving the problem.

## 9. CONCLUSION

This article presents 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 nonlinear 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 the geometric language for the treatment of robotic problems.

The article 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 for solving its inverse kinematics, according to the need, we can resort either to a point, a line, or a plane representation. Thus, the main contribution of this article is to show that motor algebra is indeed a language for high level reasoning by means of which our approach gains more flexibility, preserving the geometric insight during the computation. We believe that the increasing complexity of future multiple-link and cooperative mechanisms will profit from the versatility of the motor algebra framework.

## REFERENCES


1. J. Denavit and R.S. Hartenberg, A kinematic notation for the lower-pair mechanism based on matrices, ASME J Appl Mech June (1955), 215–221.
2. M.W. Walker, Manipulator kinematics and the epsilon algebra, IEEE J Robotics Automat 4(2) (1988), 186–192.
3. Y.L. Gu and J.Y.S. Luh, Dual-number transformation and its applications to robotics, IEEE J Robotics Automat RA-3(6) (1987), 615–623.
4. G.R. Pennock and A.T. Yang, Application of dual-number matrices to the inverse kinematics problem of robot manipulators, J Mechanisms Transmissions Automat Design 107 (1985), 201–208.
5. J.M. McCarthy, Dual orthogonal matrices in manipulator kinematics, Int J Robotics Res 5(2) (1986), 45–51.
6. J. Funda and R.P. Paul, A computational analysis of screw transformation in robotics, IEEE Trans Robotics Automat 6(3) (1990), 348–356.
7. J.H. Kim and V.R. Kumar, Kinematics of robot manipulators via line transformations, J Robotic Syst 7(4), 649–674.
8. N.A. Aspragathos and J.K. Dimitros, A comparative study of three methods for robot kinematics, IEEE Trans Systems, Man, Cybernetics Part B: Cybernetics 28(2) (1998), 135–145.
9. H. Grassmann, Der Ort der Hamilton'schen Quaternionen in der Ausdehnungslehre, Math Ann 12 (1877), 375–395.
10. W.K. Clifford, Applications of Grassmann's extensive algebra, Am J Math 1 (1878), 350–358.
11. D. Hestenes, Space-time algebra, Gordon and Breach, London, 1966.
12. D. Hestenes, New foundations for classical mechanics, D. Reidel, Dordrecht, 1986.
13. D. Hestenes and G. Sobczyk, Clifford algebra to geometric calculus: A unified language for mathematics and physics, D. Reidel, Dordrecht, 1984.
14. J. Laseby and E. Bayro-Corrochano, Analysis and computation of projective invariants from multiple views in the geometric algebra framework, Int J Pattern Recognition Artif Intell 13(8) (1999), 1105–1121.
15. E. Bayro-Corrochano, S. Buchholz, and G. Sommer, Self-organizing clifford neural network, Proc Int Conf Neural Networks, ICNN'96, Washington, DC, June 3–6, 1996, Vol. 1, pp. 120–125.
16. E. Bayro-Corrochano, K. Daniilidis, and G. Sommer, Motor Algebra for 3D kinematics. The case of the hand-eye calibration. To appear in Journal of Mathematical Imaging and Vision, 2000.

17. W.K. Clifford, Preliminary sketch of bi-quaternions, Proc London Math Soc 4 (1873), 381–395.
18. W. Blaschke, Kinematik und quaternionen, VEB Deutsch Verlag der Wissenschaften, Berlin, 1960.
19. A.T. Yang and F. Freudenstein, Application of dual-number quaternion algebra to the analysis of spatial mechanisms, ASME Trans J Appl Mechanics (1964), 300–306.
20. E. Bayro-Corrochano and G. Sobczyk, "Applications of Lie algebras and the algebra of incidence," Advances in geometric algebra with applications in science and engineering: Automatic theorem proving, computer vision, quantum and neural computing and robotics, E. Bayro-Corrochano and G. Sobczyk (Editors), Birkhäuser, New York, Chap. 13, 2000.
21. D. Hestenes, "Old wine in new bottles: A new algebraic framework for computational geometry," Advances in geometric algebra with applications in science and engineering: Automatic theorem proving, computer vision, quantum and neural computing and robotics, E. Bayro-Corrochano and G. Sobczyk (Editors), Birkhäuser, New York, Chap. 1, 2000.