Contents

Chapter 1

The Clifford Algebra of Double Quaternions and the Optimization of TS Robot Design

Shawn G. Ahlers
J. Michael McCarthy

1.1 Introduction

The goal of this chapter is a computer aided design environment that assists the inventor to formulate a task and evaluate candidate devices. The task trajectory of a robot is specified as a set of homogeneous transforms that define key frames for a desired end-effector trajectory. These key frames are converted to double quaternions and interpolated by generalizing well known techniques for Bezier interpolation of quaternions. The result is an efficient interpolation algorithm.

Our focus here is the design of a five degree of freedom TS robot that reaches the given task trajectory. The TS robot is constructed by connecting a pair of revolute joints perpendicular to each other as the base pivot to a spherical (S) joint by a fixed distance, see Figure 1.1. The pair of revolute joints is also known as a gimbal (T) or universal joint. The set of reachable positions and orientations of this device is its workspace which may not include the entire specified trajectory. Our goal is to find the TS robot minimizes the local error between its workspace and this task trajectory.

1.2 Literature Review

Bezier interpolation is used in computer drawing systems to generate curves through specified points (Farin [5]). Shoemake [13] shows that this technique can be used to interpolate rotation key frames specified by quaternion coordinates (Hamilton [9]); the result is an efficient animation algorithm. Ge and Ravani [8] generalize Shoemake's results to spatial displacements using double quaternions (Clifford [1]). These results were refined to ensure smooth transitions at each key frame by Ge and Kang [7].

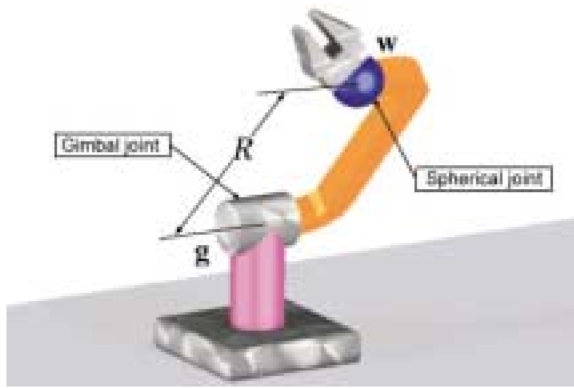

FIGURE 1.1. TS Robot

In this chapter, we apply the results of Ge and Kang to double quaternion interpolation. Etzel and McCarthy [4] show how spatial displacements can be transformed to 4×4 rotations in E^4, and then to double quaternions. A benefit of this approach is that the interpolation algorithm can be applied to the quaternion components separately.

The robot design problem seeks the dimensions of the device that satisfy geometric constraints (Suh and Radcliffe [14]). The structure of the TS robot requires the wrist **w** to lie on a sphere about the fixed gimbal joint **g**. Innocenti [10] presents a design algorithm that yields as many as 20 TS chains that reach seven arbitrary positions. Our goal is to find the TS robot that fits our end-effector trajectory with arbitrarily many positions.

1.3 Overview of the Design Algorithm

The design algorithm begins with the specification of the task. The task is defined by the N+1 user-specified key frames. These key frames are converted from their representation as homogeneous transforms to double quaternions. These double quaternions are interpolated to define the task trajectory of a desired robot. To compute a TS robot, the frames of this task trajectory are converted back to their homogeneous transforms. By using four position synthesis, the parameters of a TS robot are computed from four frames of the task trajectory. The synthesis procedure is repeated for all combinations of four frames of the task trajectory. The optimization procedure begins by calculating the closest positions and orientations reachable by a designed TS robot to the remaining frames of the trajectory. These new reachable frames are converted to double quaternions. The local error between a frame from task trajectory and the reachable frame is calculated as the magnitude of the difference of these double quaternions. This local error is summed for each frame on the task trajectory and divided by the total number of frames to obtain the error. The optimization procedure is repeated for each TS robot obtained from the synthesis procedure. The TS robot with the minimum error is the optimum fit of the robot to the task trajectory, see Figure 1.2. If the optimum TS robot is not satisfactory, the user may alter the key frames and the design process is repeated to obtain another robot candidate.

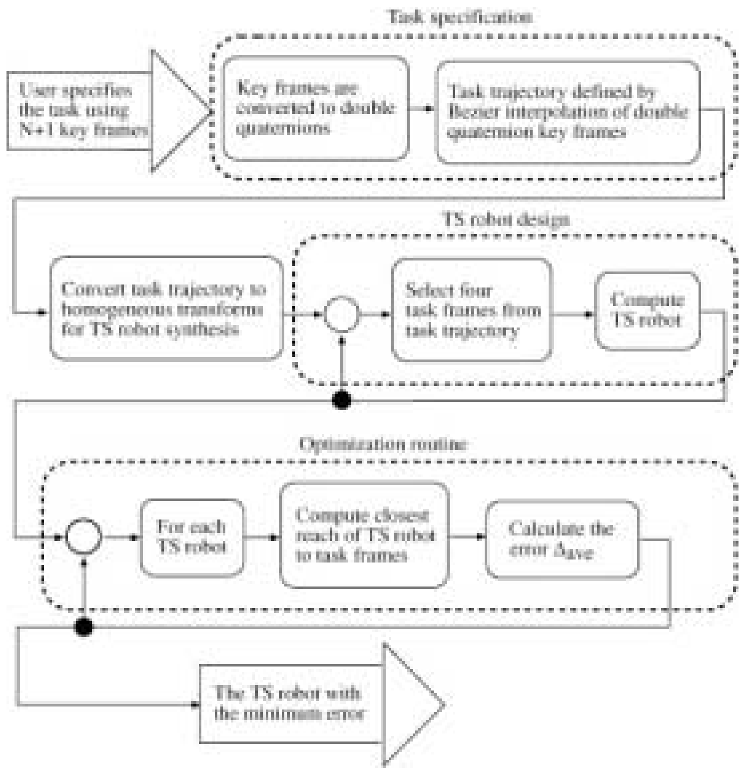

FIGURE 1.2. TS Robot Design Flowchart

1.4 Double Quaternions

1.4.1 homogeneous transforms

The transformation equation for a spatial displacement is not a linear transformation. A spatial displacement consists of a 3×3 rotation matrix and a 3×1 displacement vector. A standard strategy to adjust for this inhomogeneity is to add a fourth component to our position vectors that will always equal 1, then we can introduce the 4×4 *homogeneous transform*

$$\left\{ \begin{array}{c} \mathbf{y} \\ 1 \end{array} \right\} = \begin{bmatrix} A & \mathbf{d} \\ 0\,0\,0 & 1 \end{bmatrix} \left\{ \begin{array}{c} \mathbf{x} \\ 1 \end{array} \right\} \tag{1.1}$$

which we write as

$$\mathbf{y} = [H]\mathbf{x}. \tag{1.2}$$

The rotation-translation pairs $[H] = [A, \mathbf{d}]$ represent all the spatial positions of M relative to F, known as the special Euclidean group, $SE(3)$.

1.4.2 The Clifford Algebra on E^4

A set of hypercomplex numbers called *double quaternions* may be obtained from the even Clifford Algebra of four dimensional Euclidean space E^4. Let $\mathbf{e}_i, i = 1, \ldots, 4$ be the natural coordinate vectors of E^4, then we can construct the multilinear algebra of points in E^4. Introduce the Clifford product

$$\mathbf{e}_i\mathbf{e}_j + \mathbf{e}_j\mathbf{e}_i = -2\mathbf{e}_i \cdot \mathbf{e}_j, \tag{1.3}$$

where the dot denotes the usual Euclidean scalar product. The even subalgebra $C^+(E^4)$ is of the rank 8, and a typical element can be written as

$$\tilde{\mathbf{Q}} = \mathbf{G} + \omega\mathbf{H}, \tag{1.4}$$

where $\mathbf{G}$ and $\mathbf{H}$ are Hamilton's quaternions and $\omega = \mathbf{e}_1\mathbf{e}_2\mathbf{e}_3\mathbf{e}_4$ satisfies the identity $\omega^2 = 1$.

Clifford shows that depending on the definition of the scalar product in equation (1.3) we can also obtain dual quaternions, $\omega^2 = 0$, and complex quaternions, $\omega^2 = -1$. Using the double quaternion algebra, $\omega^2 = 1$, we now introduce the symbols $\xi = (1-\omega)/2$ and $\eta = (1+\omega)/2$, and construct the double quaternion

$$\tilde{\mathbf{Q}} = (\mathbf{G} - \mathbf{H})\xi + (\mathbf{G} + \mathbf{H})\eta \tag{1.5}$$

Notice that $\xi^2 = \xi$, $\eta^2 = \eta$, and $\xi\eta = 0$. These identities provide a complete separation of the operations on the quaternions $(\mathbf{G} - \mathbf{H})$ and $(\mathbf{G} + \mathbf{H})$. For example, for any two double quaternions $\tilde{\mathbf{P}} = \mathbf{P}_1\xi + \mathbf{P}_2\eta$ and $\tilde{\mathbf{R}} = \mathbf{R}_1\xi + \mathbf{R}_2\eta$, we have

$$\tilde{\mathbf{P}}\tilde{\mathbf{R}} = \mathbf{P}_1\mathbf{R}_1\xi + \mathbf{P}_2\mathbf{R}_2\eta. \tag{1.6}$$

Since operations on the quaternions may be done independently, the interpolation technique defined for a single quaternion may be utilized for the individual quaternions of the double quaternions. As we will show in a later section, this will allow us to interpolate the quaternions independently.

1.4.3 Homogeneous transformations as a Rotations in E^4

The general 4×4 homogeneous transform for a spatial displacement can be written as

$$[H] = [A, \mathbf{d}] = \begin{bmatrix} c\theta c\psi - s\theta s\phi s\psi & -c\theta s\phi - s\theta s\phi c\psi & s\theta c\phi & d_x \\ c\phi s\psi & c\phi c\psi & s\phi & d_y \\ -s\theta c\psi - c\theta s\phi s\psi & s\theta s\psi - c\theta s\phi c\psi & c\theta c\phi & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1.7)$$

where the angles θ, ϕ, and ψ are the longitude, latitude, and roll angles defining the orientation of the displaced frame, respectively, and c and s represent the cosine and sine functions.

Now define the angles α, β, and γ be defined such that

$$\alpha = \frac{d_x}{R}, \quad \beta = \frac{d_y}{R}, \quad \text{and} \quad \gamma = \frac{d_z}{R}, \quad (1.8)$$

where R is the radius of the hypersphere to which the translational elements are computed. We can compute the 4×4 rotation matrix [J] composed of successive rotations of α in the W-X plane, β in the W-Y plane, and γ in the W-Z plane to obtain

$$[J] = \begin{bmatrix} c\alpha & 0 & 0 & s\alpha \\ -s\beta s\alpha & c\beta & 0 & s\beta c\alpha \\ -s\gamma c\beta s\alpha & -s\gamma s\beta & c\gamma & s\gamma c\beta c\alpha \\ -c\gamma c\beta s\alpha & -s\beta c\gamma & -s\gamma & c\gamma c\beta c\alpha \end{bmatrix}. \quad (1.9)$$

If we let A(θ, ϕ, ψ) be the upper left 3×3 submatrix of the 4×4 matrix [K] and keep a 1 in the fourth diagonal location, we may express a general rotation in four dimensional space, E^4, as the product of two 4×4 rotation matrices [D]=[J(α,β,γ)][K(θ,ϕ,ψ)]. Explicitly written

$$[D] = \begin{bmatrix} c\alpha & 0 & 0 & s\alpha \\ -s\beta s\alpha & c\beta & 0 & s\beta c\alpha \\ -s\gamma c\beta s\alpha & -s\gamma s\beta & c\gamma & s\gamma c\beta c\alpha \\ -c\gamma c\beta s\alpha & -s\beta c\gamma & -s\gamma & c\gamma c\beta c\alpha \end{bmatrix} \begin{bmatrix} c\theta c\psi - s\theta s\phi s\psi & -c\theta s\phi - s\theta s\phi c\psi & s\theta c\phi & 0 \\ c\phi s\psi & c\phi c\psi & s\phi & 0 \\ -s\theta c\psi - c\theta s\phi s\psi & s\theta s\psi - c\theta s\phi c\psi & c\theta c\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.10)$$

If we assume that the angles α, β, and γ are small, $\cos\alpha = \cos\beta = \cos\gamma = 1$, and

$$\sin\alpha = \frac{d_x}{R}, \quad \sin\beta = \frac{d_y}{R}, \quad \text{and} \quad \sin\gamma = \frac{d_z}{R}. \quad (1.11)$$

Then the 4×4 rotation matrix becomes

$$[D] = \begin{bmatrix} 1 & 0 & 0 & \frac{d_x}{R} \\ 0 & 1 & 0 & \frac{d_y}{R} \\ 0 & 0 & 1 & \frac{d_z}{R} \\ -\frac{d_x}{R} & -\frac{d_y}{R} & -\frac{d_z}{R} & 1 \end{bmatrix}$$
$$\begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi - \mathrm{s}\theta\mathrm{s}\phi\mathrm{s}\psi & -\mathrm{c}\theta\mathrm{s}\phi - \mathrm{s}\theta\mathrm{s}\phi\mathrm{c}\psi & \mathrm{s}\theta\mathrm{c}\phi & 0 \\ \mathrm{c}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\phi & 0 \\ -\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\theta\mathrm{s}\phi\mathrm{s}\psi & \mathrm{s}\theta\mathrm{s}\psi - \mathrm{c}\theta\mathrm{s}\phi\mathrm{c}\psi & \mathrm{c}\theta\mathrm{c}\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{1.12}$$

If we shift the coordinate frame to $W = R$ to cancel the $1/R$ terms, the result is an approximation to a spatial displacement of order $O(1/R^2)$.

The parameter R is identified by specifying a maximum length L for the problem, then the error of this approximation is $\varepsilon \leq (L/R)^2$. Specify ε and solve for R in order to define the rotation in E^4 that approximates a given spatial displacement.

1.4.4 Double quaternion for a spatial displacement

In this subsection, we will reformulate $[H]$ in terms of double quaternions. After we have converted a spatial displacement $[H] = [A, \mathbf{d}]$ to a 4×4 rotation $[D]$, we may use Cayley's formula (Bottema and Roth, 1979), to obtain the skew symmetric matrix

$$[B] = [D - I][D + I]^{-1} = \begin{bmatrix} 0 & -u_3 & u_2 & v_1 \\ u_3 & 0 & -u_1 & v_2 \\ -u_2 & u_1 & 0 & v_3 \\ -v_1 & -v_2 & -v_3 & 0 \end{bmatrix}. \tag{1.13}$$

We now define the matrix $[B']$ by interchanging the u_i and v_i terms, in order to obtain the matrices (Etzel and McCarthy, [4])

$$k_1[S] = \frac{[B] + [B']}{2} = k_1 \begin{bmatrix} 0 & -s_3 & s_2 & s_1 \\ s_3 & 0 & -s_1 & s_2 \\ -s_2 & s_1 & 0 & s_3 \\ -s_1 & -s_2 & -s_3 & 0 \end{bmatrix} \tag{1.14}$$

and

$$k_2[T] = \frac{[B] - [B']}{2} = k_2 \begin{bmatrix} 0 & -t_3 & t_2 & -t_1 \\ t_3 & 0 & -t_1 & -t_2 \\ -t_2 & t_1 & 0 & -t_3 \\ t_1 & t_2 & t_3 & 0 \end{bmatrix} \tag{1.15}$$

where $[B] = k_1[S] + k_2[T]$ and $\sum s_i^2 = \sum t_i^2 = 1$. We can then compute μ and ν by the equations

$$\begin{aligned} \mu &= \arctan(k_1 + k_2) + \arctan(k_1 - k_2) \\ \nu &= \arctan(k_1 + k_2) - \arctan(k_1 - k_2). \end{aligned} \tag{1.16}$$

The double quaternion is now given by $\tilde{\mathbf{G}} = \mathbf{G}_1\xi + \mathbf{G}_2\eta$, where

$$\mathbf{G}_1 = \left\{ \begin{array}{c} \sin\mu\mathbf{s} \\ \cos\mu \end{array} \right\} \quad \text{and} \quad \mathbf{G}_2 = \left\{ \begin{array}{c} \sin\nu\mathbf{t} \\ \cos\nu \end{array} \right\}. \tag{1.17}$$

where $\mathbf{s} = (s_1, s_2, s_3)^T$ and $\mathbf{t} = (t_1, t_2, t_3)^T$ define the axes to which the angles μ and ν are to be rotated, respectively. Again, notice that each 4×4 rotation matrix defines a double quaternion, that can be separated into a pair of quaternions that multiply separately.

1.4.5 Spatial displacement from a double quaternion

Assuming we have a double quaternion $\tilde{\mathbf{G}}$ of the form of equation (1.17), we compute the associated spatial displacement as follows. Note that a 4×4 rotation matrix can be written in exponential form

$$[D] = e^{[M]} \tag{1.18}$$

where $[M]$ is a 4×4 skew symmetric matrix. The matrix $[M]$ has the form $[M] = \mu[S] + \nu[T]$ (Ge, [6]). Thus

$$[D] = e^{(\mu[S] + \nu[T])} = e^{\mu[S]}e^{\nu[T]}. \tag{1.19}$$

The series expansion of $e^{\mu[S]}$ and $e^{\nu[T]}$ and the identities $[S]^2 = [T]^2 = -[I]$ yield the formulas

$$e^{\mu[S]} = \begin{bmatrix} \cos\mu & -s_3\sin\mu & s_2\sin\mu & s_1\sin\mu \\ s_3\sin\mu & \cos\mu & -s_1\sin\mu & s_2\sin\mu \\ -s_2\sin\mu & s_1\sin\mu & \cos\mu & s_3\sin\mu \\ -s_1\sin\mu & -s_2\sin\mu & -s_3\sin\mu & \cos\mu \end{bmatrix}, \tag{1.20}$$

and

$$e^{\nu[T]} = \begin{bmatrix} \cos\nu & -t_3\sin\nu & t_2\sin\nu & -t_1\sin\nu \\ t_3\sin\nu & \cos\nu & -t_1\sin\nu & -t_2\sin\nu \\ -t_2\sin\nu & t_1\sin\nu & \cos\nu & -t_3\sin\nu \\ t_1\sin\nu & t_2\sin\nu & t_3\sin\nu & \cos\nu \end{bmatrix}. \tag{1.21}$$

The result is the 4×4 rotation matrix $[D]$ defined by $\tilde{\mathbf{G}} = \mathbf{G}_1\xi + \mathbf{G}_2\eta$.

The 4×4 homogeneous transform approximating the rotation is $[H] = [A, \mathbf{d}]$, where A is the upper left 3×3 rotation matrix and the translation vector $\mathbf{d} = (d_x, d_y, d_z)$ is given by

$$d_x = d_{14}R, \quad d_y = d_{24}R, \quad d_z = d_{34}R, \tag{1.22}$$

where d_{ij} is the ijth element of the $[D]$ matrix. The longitude, latitude, and roll angles are

$$\begin{aligned}\theta &= \arctan(d_{13}/d_{33}),\\ \phi &= \arctan(d_{23}\cos\theta/d_{33}) = \arctan(d_{23}\sin\theta/d_{13}),\\ \psi &= \arctan(d_{21}/d_{22}).\end{aligned} \tag{1.23}$$

Thus, a homogeneous transformation may be computed from a double quaternion.

1.5 The Task Trajectory

The task of the robot is defined in terms of the trajectory of the end-effector. In order to define this task, we specify a set of key frames and use Bezier interpolation to generate the trajectory. Consider $N+1$ key frames defined by the homogenous transforms $[H_k], k = 0, \dots, N$. The double quaternions associated with each key frame are denoted as $\tilde{\mathbf{P}}_k = \mathbf{P}_{k,1}\xi + \mathbf{P}_{k,2}\eta$. We use the Bezier interpolation of double quaternions developed by Ge and Kang [7] to create the task trajectory.

Bezier interpolation for double quaternions follows the principles of Bezier interpolation for curves, see Farin [5]. There are two main features the generation of a curve segment between two key frames using the deCasteljau [2, 3] algorithm, and the joining of these segments together to maintain G^1 and G^2 continuity. These continuity conditions ensure a smooth movement of the body along the trajectory.

1.5.1 The DeCasteljau algorithm

In order to generate a trajectory segment between the twokey frames $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{P}}_{i+1}$, we need a Bezier polygon $\tilde{\mathbf{B}}_{3i}, \tilde{\mathbf{B}}_{3i+1}, \tilde{\mathbf{B}}_{3i+2}$, and $\tilde{\mathbf{B}}_{3i+3}$. The first and last double quaternions of the Bezier polygon are identified with the two key frames,

$$\tilde{\mathbf{B}}_{3i} = \tilde{\mathbf{P}}_i \quad \text{and} \quad \tilde{\mathbf{B}}_{3i+3} = \tilde{\mathbf{P}}_{i+1}. \tag{1.24}$$

The intermediate Bezier double quaternions $\tilde{\mathbf{B}}_{3i+1}$ and $\tilde{\mathbf{B}}_{3i+2}$ are calculated to provide the desired continuity conditions when the complete trajectory is assembled. We show in the next section how this is done.

Here we show how the DeCasteljau algorithm is used to generate positions along the trajectory between two key frames for a given Bezier polygon. The central feature of the algorithm is an interpolation formula between two double quaternions $\tilde{\mathbf{P}}_i$ and $\tilde{\mathbf{P}}_{i+1}$, which is a generalization of Shoemake's [13] original results. Let $\tilde{\mathbf{V}} = \xi\mathbf{V}_1 + \eta\mathbf{V}_2$ and $\tilde{\mathbf{W}} = \xi\mathbf{W}_1 + \eta\mathbf{W}_2$

be two unit double quaternions, then the great circular arc $\tilde{\mathbf{L}}(t)$ between them is defined by the formula

$$\tilde{\mathbf{L}}(t) = \frac{\sin(1-t)\tilde{\rho}}{\sin\tilde{\rho}}\tilde{\mathbf{V}} + \frac{\sin t\tilde{\rho}}{\sin\tilde{\rho}}\tilde{\mathbf{W}} \tag{1.25}$$

where $\cos\tilde{\rho} = \tilde{\mathbf{V}}\cdot\tilde{\mathbf{W}}$.

Expanding the double quaternions in this equation, we obtain

$$\tilde{\mathbf{L}}(t) = \xi(\frac{\sin(1-t)\rho_1}{\sin\rho_1}\mathbf{V}_1 + \frac{\sin t\rho_1}{\sin\rho_1}\mathbf{W}_1) + \eta(\frac{\sin(1-t)\rho_2}{\sin\rho_2}\mathbf{V}_2 + \frac{\sin t\rho_2}{\sin\rho_2}\mathbf{W}_2). \tag{1.26}$$

Notice that this equation separates to define the interpolation of the quaternion components of $\tilde{\mathbf{V}}$ and $\tilde{\mathbf{W}}$, individually. Thus our formalism simply requires us to apply Shoemake's interpolation formula twice. In fact, all of our algorithms handle the components of the double quaternions independently.

For a particular value of the parameter t we now seek the double quaternion $\tilde{\mathbf{D}}(t)$ along the Bezier curve segment. The DeCasteljau algorithm uses equation (1.25) to generate circular arcs between each of the Bezier double quaternions $\tilde{\mathbf{B}}_{3i}, \tilde{\mathbf{B}}_{3i+1}, \tilde{\mathbf{B}}_{3i+2}$, and $\tilde{\mathbf{B}}_{3i+3}$ associated with this ith segment. To do this we first compute the double quaternions $\tilde{\mathbf{X}}_0$, $\tilde{\mathbf{X}}_1$, and $\tilde{\mathbf{X}}_2$ on each of these arcs by the formula

$$\tilde{\mathbf{X}}_m(t) = \frac{\sin(1-t)\tilde{\rho}_m}{\sin\tilde{\rho}_m}\tilde{\mathbf{B}}_{3i+m} + \frac{\sin t\tilde{\rho}_m}{\sin\tilde{\rho}_m}\tilde{\mathbf{B}}_{3i+(m+1)}, m = 0,1,2. \tag{1.27}$$

where m denotes the arc connecting the Bezier double quaternions $\tilde{\mathbf{B}}_{3i+m}$ and $\tilde{\mathbf{B}}_{3i+m+1}$. Next repeat this process in order to define the double quaternions $\tilde{\mathbf{Y}}_0$ and $\tilde{\mathbf{Y}}_1$ on the arcs joining $\tilde{\mathbf{X}}_0$, $\tilde{\mathbf{X}}_1$ and $\tilde{\mathbf{X}}_1$, $\tilde{\mathbf{X}}_2$, defined by the formula

$$\tilde{\mathbf{Y}}_n(t) = \frac{\sin(1-t)\tilde{\sigma}_n}{\sin\tilde{\sigma}_n}\tilde{\mathbf{X}}_n + \frac{\sin t\tilde{\sigma}_n}{\sin\tilde{\sigma}_n}\tilde{\mathbf{X}}_{n+1}, n = 0,1. \tag{1.28}$$

Finally, we obtain the frame $\tilde{\mathbf{D}}(t)$ as

$$\tilde{\mathbf{D}}_i(t) = \frac{\sin(1-t)\tilde{\tau}}{\sin\tilde{\tau}}\tilde{\mathbf{Y}}_0 + \frac{\sin t\tilde{\tau}}{\sin\tilde{\tau}}\tilde{\mathbf{Y}}_1. \tag{1.29}$$

As the parameter t varies from 0 to 1, $\tilde{\mathbf{D}}(t)$ will define the trajectory from $\tilde{\mathbf{P}}_i$ to $\tilde{\mathbf{P}}_{i+1}$. This procedure can be generalized for Bezier polygons with more intermediate vertices, see Farin [5].

1.5.2 Bezier interpolation

To define the entire task trajectory, we must compute the Bezier polygon for each of the N segments. The intermediate Bezier double quaternions, $\tilde{\mathbf{B}}_{3i+1}$ and $\tilde{\mathbf{B}}_{3i+2}$ are determined to ensure continuity at each junction.

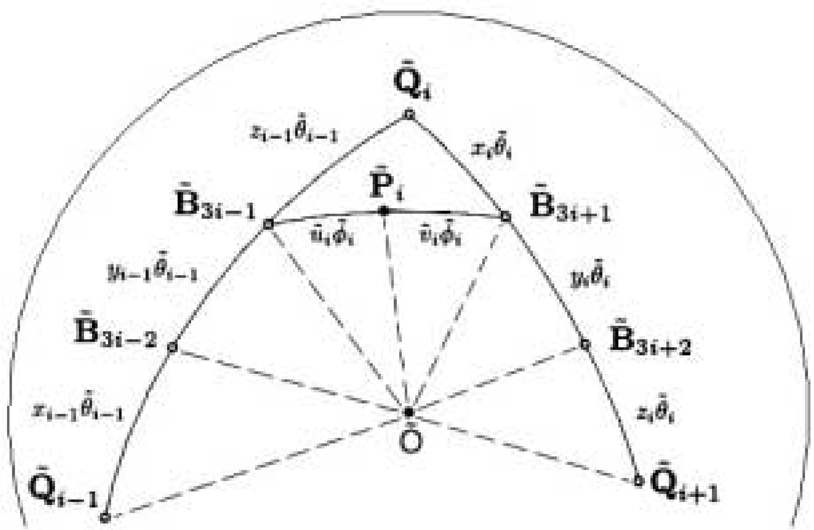

FIGURE 1.3. Construction of Bezier Double Quaternions

To ensure G^1 continuity, $\tilde{\mathbf{B}}_{3i-1}$ and $\tilde{\mathbf{B}}_{3i+1}$ and the key frame $\tilde{\mathbf{P}}_i$ must lie the same arc, see Figure 1.3. Therefore, $\tilde{\mathbf{P}}_i$ is related to $\tilde{\mathbf{B}}_{3i+1}$ and $\tilde{\mathbf{B}}_{3i-1}$ by

$$\sin\tilde{\phi}_i\tilde{\mathbf{P}}_i = \sin\tilde{v}_i\tilde{\phi}_i\tilde{\mathbf{B}}_{3i-1} + \sin\tilde{u}_i\tilde{\phi}_i\tilde{\mathbf{B}}_{3i+1}. \tag{1.30}$$

The parameter $\tilde{\phi}_i = \arccos(\tilde{\mathbf{B}}_{3i-1}\cdot\tilde{\mathbf{B}}_{3i+1})$ is the double arclength between $\tilde{\mathbf{B}}_{3i-1}$ and $\tilde{\mathbf{B}}_{3i+1}$. The parameters $\tilde{u}_i$ and $\tilde{v}_i$ locate $\tilde{\mathbf{P}}_i$ on this arc, such that $\tilde{u}_i\tilde{\phi}_i = \arccos(\tilde{\mathbf{B}}_{3i-1}\cdot\tilde{\mathbf{P}}_i), \tilde{v}_i\tilde{\phi}_i = \arccos(\tilde{\mathbf{P}}_i\cdot\tilde{\mathbf{B}}_{3i+1})$. These parameters satisfy the relation

$$\tilde{u}_i + \tilde{v}_i = 1. \tag{1.31}$$

To ensure G^2 continuity, the five double quaternions $\tilde{\mathbf{B}}_{3i-2}$, $\tilde{\mathbf{B}}_{3i-1}$, $\tilde{\mathbf{P}}_i$, $\tilde{\mathbf{B}}_{3i+1}$, and $\tilde{\mathbf{B}}_{3i+2}$ must lie on the same great sphere (see Ge and Kang [7]). To do this, we introduce the control double quaternion $\tilde{\mathbf{Q}}_i$ that is defined to be the intersection of the arcs through by $\tilde{\mathbf{B}}_{3i-2}$, $\tilde{\mathbf{B}}_{3i-1}$, and $\tilde{\mathbf{B}}_{3i+1}$, $\tilde{\mathbf{B}}_{3i+1}$. These Bezier double quaternions lie on arcs through $\tilde{\mathbf{Q}}_{i-1}$ and $\tilde{\mathbf{Q}}_i$, and $\tilde{\mathbf{Q}}_i$ and $\tilde{\mathbf{Q}}_{i+1}$. They are located by the parameters x_i, y_i, z_i, so that

$$\begin{gathered}\sin\tilde{\theta}_{i-1}\tilde{\mathbf{B}}_{3i-1} = \sin z_{i-1}\tilde{\theta}_{i-1}\tilde{\mathbf{Q}}_{i-1} + \sin(x_{i-1}+y_{i-1})\tilde{\theta}_{i-1}\tilde{\mathbf{Q}}_i,\\ \text{for}\quad i = 2,\dots,N-1,\end{gathered} \tag{1.32}$$

and

$$\begin{gathered}\sin\tilde{\theta}_i\tilde{\mathbf{B}}_{3i+1} = \sin(y_i+z_i)\tilde{\theta}_i\tilde{\mathbf{Q}}_i + \sin x_i\tilde{\theta}_i\tilde{\mathbf{Q}}_{i+1},\\ \text{for}\quad i = 1,\dots,N-2,\end{gathered} \tag{1.33}$$

where the angle $\tilde{\theta}_i = \arccos(\tilde{\mathbf{Q}}_i \cdot \tilde{\mathbf{Q}}_{i+1})$. Note, the x_i, y_i, z_i are greater than zero and satisfy the constraint

$$x_i + y_i + z_i = 1 \quad \text{for} \quad 1 \leq \mathrm{i} \leq \mathrm{N} - 2. \tag{1.34}$$

At the endpoints of the trajectory $x_0 = 0$ and $z_{N-1} = 0$.

To complete the condition for G^2 continuity, we require the parameters $\tilde{u}_i$ and $\tilde{v}_i$ to satisfy the constraint derived in [7],

$$\frac{\tilde{v}_i \sin \tilde{v}_i \tilde{\phi}_i}{\tilde{u}_i \sin \tilde{u}_i \tilde{\phi}_i} = \frac{y_{i-1} \tilde{\theta}_{i-1} \sin x_i \tilde{\theta}_i}{y_i \tilde{\theta}_i \sin z_{i-1} \tilde{\theta}_{i-1}}. \tag{1.35}$$

This equation together with equation (1.31) can be solved to determine the parameters $\tilde{u}_i$ and $\tilde{v}_i$ in terms of the angles $\tilde{\theta}_i$ and $\tilde{\phi}_i$. These angles are computed from the locations of $\tilde{\mathbf{Q}}_i$ and $\tilde{\mathbf{B}}_{3i\pm1}$. The control double quaternions $\tilde{\mathbf{Q}}_i$ must be found to fit the user-specified $\tilde{\mathbf{P}}_i$. The parameters $\tilde{u}_i$ and $\tilde{v}_i$ are determined by solving equations (1.35) and (1.31) numerically.

Substitute equations (1.32) and (1.33) into (1.30) to define the key frame double quaternions $\tilde{\mathbf{P}}_i$ directly in terms of the control quaternions $\tilde{\mathbf{Q}}_i$

$$\tilde{\mathbf{P}}_i = a_i \tilde{\mathbf{Q}}_{i-1} + b_i \tilde{\mathbf{Q}}_i + c_i \tilde{\mathbf{Q}}_{i+1} \tag{1.36}$$

where

$$\begin{aligned} \tilde{a}_i &= \frac{\sin z_{i-1} \tilde{\theta}_{i-1} \sin \tilde{v}_i \tilde{\phi}_i}{\sin \tilde{\theta}_{i-1} \sin \tilde{\phi}_i}, \\ \tilde{b}_i &= \frac{\sin(x_{i-1} + y_{i-1}) \tilde{\theta}_{i-1} \sin \tilde{v}_i \tilde{\phi}_i}{\sin \tilde{\theta}_{i-1} \sin \tilde{\phi}_i} + \frac{\sin(y_i + z_i) \tilde{\theta}_i \sin \tilde{u}_i \tilde{\phi}_i}{\sin \tilde{\theta}_i \sin \tilde{\phi}_i}, \\ \tilde{c}_i &= \frac{\sin x_i \tilde{\theta}_i \sin \tilde{u}_i \tilde{\phi}_i}{\sin \tilde{\theta}_i \sin \tilde{\phi}_i}. \end{aligned} \tag{1.37}$$

This relationship can be written in matrix form as

$$\begin{bmatrix} \tilde{a}_1, & \tilde{b}_1, & \tilde{c}_1, & 0, & \cdots & 0, & 0, & 0 \\ 0, & \tilde{a}_2, & \tilde{b}_2, & \tilde{c}_2, & \cdots & 0, & 0, & 0 \\ \vdots & \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots \\ 0, & 0, & 0, & 0, & \cdots & \tilde{a}_{N-1}, & \tilde{b}_{N-1}, & \tilde{c}_{N-1} \end{bmatrix} \left\{ \begin{array}{c} \mathbf{Q}_1 \\ \vdots \\ \mathbf{Q}_{N-1} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{P}_1 \\ \vdots \\ \mathbf{P}_{N-1} \end{array} \right\}, \tag{1.38}$$

or

$$[\mathrm{M}]\vec{\mathbf{Q}} = \vec{\mathbf{P}}, \tag{1.39}$$

where $[M]$ is defined as the coefficient matrix, $\vec{\mathbf{Q}} = (\tilde{\mathbf{Q}}_1, \ldots, \tilde{\mathbf{Q}}_{N-1})^T$, and $\vec{\mathbf{P}} = (\tilde{\mathbf{P}}_1, \ldots, \tilde{\mathbf{P}}_{N-1})^T$. Note, the coefficients $\tilde{a}_i, \tilde{b}_i$, and $\tilde{c}_i$ are dependent on the angles $\tilde{\theta}_i$ and $\tilde{\phi}_i$ which, in turn, are dependent on the control quaternions $\tilde{\mathbf{Q}}_i$.

Given an estimate for the angles $\tilde{\theta}_i$ and $\tilde{\phi}_i$ and the variables $\tilde{u}_i$ and $\tilde{v}_i$, we can compute $[M^j]$ and solve equation (1.39) for $\vec{\mathbf{Q}}^{j+1}$ such that

$$\vec{\mathbf{Q}}^{j+1} = [\mathrm{M}^j]^{-1}\vec{\mathbf{P}}. \tag{1.40}$$

From $\vec{\mathbf{Q}}^{j+1}$, we calculate $\tilde{\mathbf{B}}_{3i-1}$ and $\tilde{\mathbf{B}}_{3i+1}$ from equations (1.32, 1.33). At this point, we correct the estimates for $\tilde{\theta}_i$, $\tilde{\phi}_i$, $\tilde{u}_i$, and $\tilde{v}_i$ and recompute $\tilde{\mathbf{Q}}_i$. The process stops when equation (1.30) is satisfied.

The Bezier interpolation procedure is as follows

Step 1. The special cases of the endpoints are handled by defining the control quaternions at the boundaries, that is, $\tilde{\mathbf{Q}}_{-1} = \tilde{\mathbf{B}}_0 = \tilde{\mathbf{P}}_0$ and $\tilde{\mathbf{Q}}_{N+1} = \tilde{\mathbf{B}}_{3N} = \tilde{\mathbf{P}}_N$. The adjacent Bezier double quaternions are defined as $\tilde{\mathbf{B}}_1 = \tilde{\mathbf{Q}}_0$ and $\tilde{\mathbf{B}}_{3N-1} = \tilde{\mathbf{Q}}_N$. However, the choices for $\tilde{\mathbf{Q}}_0$ and $\tilde{\mathbf{Q}}_N$ are arbitrary. We choose them to lie one-tenth ($t = 0.1$) of the way from the first and last key frames on the arc-segments passing through $\tilde{\mathbf{P}}_0$, $\tilde{\mathbf{P}}_1$, and $\tilde{\mathbf{P}}_N$, $\tilde{\mathbf{P}}_{N-1}$, such that

$$\tilde{\mathbf{Q}}_0 = \frac{\sin(1-0.1)\tilde{\rho}_0}{\sin\tilde{\rho}_0}\tilde{\mathbf{P}}_0 + \frac{\sin 0.1\tilde{\rho}_0}{\sin\tilde{\rho}_0}\tilde{\mathbf{P}}_1 \tag{1.41}$$

and

$$\tilde{\mathbf{Q}}_N = \frac{\sin(1-0.1)\tilde{\rho}_N}{\sin\tilde{\rho}_N}\tilde{\mathbf{P}}_N + \frac{\sin 0.1\tilde{\rho}_N}{\sin\tilde{\rho}_N}\tilde{\mathbf{P}}_{N-1} \tag{1.42}$$

where $\tilde{\rho}_0 = \arccos(\tilde{\mathbf{P}}_0 \cdot \tilde{\mathbf{P}}_1)$ and $\tilde{\rho}_N = \arccos(\tilde{\mathbf{P}}_N \cdot \tilde{\mathbf{P}}_{N-1})$.

We also set the variables $x_i = y_i = z_i = \frac{1}{3}$ for $i = 1, \dots, N-2$; and near the ends of the trajectory, we select $y_0 = y_{N-1} = 0.6$ and $z_0 = x_{N-1} = 0.4$ for equation (1.34). Recall, $x_0 = z_{N-1} = 0$.

Step 2. Determine the initial $[M^0]$ such that $j = 0$ in equation (1.40).

- Let the initial $\tilde{u}_i$ and $\tilde{v}_i$ be defined by

$$\tilde{u}_i = \frac{(x_i y_{i-1})^{1/2}}{(x_i y_{i-1})^{1/2} + (y_i z_{i-1})^{1/2}}, \tag{1.43}$$

$$\tilde{v}_i = \frac{(y_i z_{i-1})^{1/2}}{(x_i y_{i-1})^{1/2} + (y_i z_{i-1})^{1/2}}. \tag{1.44}$$

- Compute initial values for the matrix components $[M^0]$ by

$$\begin{aligned} \tilde{a}_i &= z_{i-1} v_i, \\ \tilde{b}_i &= (x_{i-1} + y_{i-1}) v_i + (y_i + z_i) u_i, \\ \tilde{c}_i &= x_i u_i. \quad \text{for} \quad i = 1, \dots, N-1. \end{aligned} \tag{1.45}$$

	x	y	z	θ	ϕ	ψ
M_1	0.0	0.0	0.0	90°	-45°	0°
M_2	2.0	0.0	2.0	0°	0°	0°
M_3	3.5	1.0	4.0	0°	45°	0°
M_4	5.0	3.0	3.0	20°	20°	22.5°
M_5	6.5	3.0	2.0	45°	0°	45°
M_6	8.0	2.0	0.0	90°	30°	0°

TABLE 1.1. The key frame data for end-effector trajectory

Step 3. Solve (1.40) to determine the control double quaternions $\vec{\tilde{\mathbf{Q}}}^{j+1}$. Normalize each $\tilde{\mathbf{Q}}_i$ and compute $\tilde{\theta}_i = \arccos(\tilde{\mathbf{Q}}_i \cdot \tilde{\mathbf{Q}}_{i+1})$.

Step 4. Compute the Bezier quaternions $\tilde{\mathbf{B}}_{3i\pm1}$ from equations (1.32) and (1.33). Determine $\tilde{\phi}_i = \arccos(\tilde{\mathbf{B}}_{3i-1} \cdot \tilde{\mathbf{B}}_{3i+1})$, then calculate $\tilde{\mathbf{P}}_i^j$ from the equation (1.30).

Step 5. Compare the computed key frames $\tilde{\mathbf{P}}_i^j$ to the actual key frames $\tilde{\mathbf{P}}_i$ by calculating the angle $\tilde{\omega}_i = \arccos(\tilde{\mathbf{P}}_i^j \cdot \tilde{\mathbf{P}}_i)$. We define the error E to be the sum

$$E = \sum_{i=0}^{N} \tilde{\omega}_i^2. \tag{1.46}$$

The iterative procedure stops when $E \leq \delta$, where δ is the tolerance for convergence, in our case $\delta = 10^{-5}$.

If $E > \delta$,

- Calculate parameters $\tilde{u}_i$ and $\tilde{v}_i$ from the G^2 continuity equations (1.35) and (1.31).

- Compute the new components of the matrix $[M^j]$ using equation (1.37) for the next iteration, and return to step 3.

The result of this procedure is the set of Bezier polygons for each segment of the entire trajectory. DeCasteljau's algorithm is used to determine the frames along each segment.

1.5.3 Example of double quaternion interpolation

To illustrate the double quaternion interpolation procedure, we interpolate the E^3 key frames listed in Table 1.1. The double quaternions associated with these key frames is listed in Table 1.2. The resulting interpolated trajectory is shown in Figure 1.4 where the white end-effectors correspond to the key frames and the black end-effectors are the interpolated frames. The interpolation procedure converged in four interations.

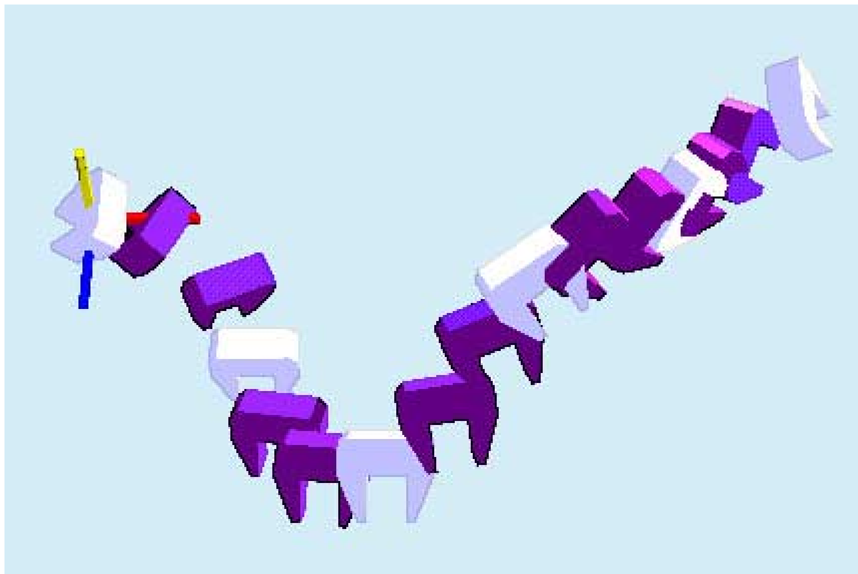

FIGURE 1.4. Double Quaternion Interpolated Path

	$\mathbf{P}_1$				$\mathbf{P}_2$			
M_1	0.271	0.653	-0.271	0.653	0.271	0.653	-0.271	0.653
M_2	0.006	0.000	0.006	0.999	-0.006	0.000	-0.006	0.999
M_3	-0.373	-0.002	0.013	0.928	-0.393	0.002	-0.013	0.920
M_4	-0.119	0.205	0.232	0.943	-0.149	0.197	0.205	0.947
M_5	0.165	0.355	0.364	0.845	0.128	0.352	0.342	0.862
M_6	-0.165	0.683	0.201	0.683	-0.201	0.683	0.165	0.683

TABLE 1.2. Double quaternion key displacements

1.6 The Design of the TS Robot

The TS robot is a five-degree of freedom mechanism that has as its base a gimbal joint and is connected to a spherical joint by a rigid link, see Figure 1.1.

Let $\mathbf{v}$ be the coordinate vector in E^3 of the wrist center in a frame M attached to a workpiece. The TS chain constrains a the point $\mathbf{v}$ to lie on a sphere of radius R about the shoulder joint $\mathbf{g}$. The point, $\mathbf{v}$, in the moving frame, M, takes the position $\mathbf{w}^i = [H_i]\mathbf{v}$ in the fixed frame, F. We have the constraint equation

$$(\mathbf{w} - \mathbf{g}) \cdot (\mathbf{w} - \mathbf{g}) = R^2. \tag{1.47}$$

For a given $\mathbf{w}$ and $\mathbf{g}$, the set of all transformations, $[H]$, that satisfies the equation (1.47) defines the workspace of the TS robot.

We use this constraint equation to define a TS robot for a given trajectory. Choose $v = (0, 0, 0)^T$ to be the origin of the moving frame, M. This reduces the non-linear design problem to a set of linear equations.

If we choose a reference position M_1 and subtract equation 1 from the rest of the equations of the form (1.47), we obtain

$$\mathbf{w}^i \cdot \mathbf{w}^i - \mathbf{w}^1 \cdot \mathbf{w}^1 - 2\mathbf{w}^i \cdot \mathbf{g} + 2\mathbf{w}^1 \cdot \mathbf{g}, \quad i = 2, \dots, n. \tag{1.48}$$

Because their are six unknown parameters, $\mathbf{g} = (x, y, z)^T$ and those of $\mathbf{w}^1 = (u, v, w)^T$, in general, we may obtain exact solutions for up to seven arbitrary positions (Innocenti [10] and McCarthy and Liao [11]).

For this chapter, we specify $\mathbf{w}^1$ and solve for the ground pivot $\mathbf{g}$. This yields a system of linear equations that has a unique solution for four specified spatial positions ($n = 4$). Writing the three constraint equations in matrix form, we have the system

$$\begin{bmatrix} (\mathbf{w}^2 - \mathbf{w}^1)^T \\ (\mathbf{w}^3 - \mathbf{w}^1)^T \\ (\mathbf{w}^4 - \mathbf{w}^1)^T \end{bmatrix} \mathbf{g} = \frac{1}{2} \left\{ \begin{array}{c} \mathbf{w}^2 \cdot \mathbf{w}^2 - \mathbf{w}^1 \cdot \mathbf{w}^1 \\ \mathbf{w}^3 \cdot \mathbf{w}^3 - \mathbf{w}^1 \cdot \mathbf{w}^1 \\ \mathbf{w}^4 \cdot \mathbf{w}^4 - \mathbf{w}^1 \cdot \mathbf{w}^1 \end{array} \right\} \tag{1.49}$$

or

$$[\mathrm{W}]\mathbf{g} = \mathbf{c}. \tag{1.50}$$

The solution $\mathbf{g}$ may be obtained by inverting the matrix [W]. This unique solution $\mathbf{g} = (x, y, z)^T$ is the center of the sphere that passes through the four points $\mathbf{w}^i, i = 1, 2, 3, 4$. The calculation of the armlength R of the TS chain can be computed from equation (1.47),

$$R = \sqrt{(x - u)^2 + (y - v)^2 + (z - w)^2}. \tag{1.51}$$

By calculating the base location $\mathbf{g}$ and the armlength R, the workspace for a specific TS chain is defined. Now, we design an optimimal TS chain such that the workspace of the chain attempts to satisfy the task trajectory.

1.7 The Optimum TS Robot

At this point, we are able to create a task trajectory from user-defined key positions and orientations. We also can determine a TS robot from four specified positions. The goal now is to find the best fit of a TS robot to our task trajectory.

Select four frames from the task trajectory. These four positions become $\mathbf{w}^i$, $i = 1, 2, 3, 4$ in the design equations (1.49) of the TS chain. The physical parameters of the robot $\mathbf{g}$ and R are then calculated. The TS robot will pass through four positions and orientations of our task trajectory and approximate the rest of the task frames. To get the closest point $\mathbf{w}$ of the robot's workspace to an arbitrary pose $\mathbf{a}$ from the task trajectory, we define the unit direction vector $\mathbf{v}$ from the fixed pivot $\mathbf{g} = (x, y, z)$ to the point $\mathbf{a} = (a, b, c)$ as

$$\mathbf{v} = \frac{\mathbf{a} - \mathbf{g}}{|\ \mathbf{a} - \mathbf{g}\ |}. \tag{1.52}$$

The point $\mathbf{w} = R\mathbf{v}$ is the closest point of the TS robot workspace to the task frame $\mathbf{a}$. The end-effector of the TS robot can attain the exact orientation of the task frame. This position and orientation is converted to the double quaternion $\tilde{\mathbf{W}} = \xi\mathbf{W}_1 + \eta\mathbf{W}_2$. Let the double quaternion $\tilde{\mathbf{A}} = \xi\mathbf{A}_1 + \eta\mathbf{A}_2$ define the task frame. The local error is defined as the magnitude of the eight-vector

$$\Delta(\tilde{\mathbf{A}}, \tilde{\mathbf{W}}) = \sqrt{(\mathbf{A}_1 - \mathbf{W}_1)^2 + (\mathbf{A}_2 - \mathbf{W}_2)^2}. \tag{1.53}$$

The total number of task frames of a task trajectory is given by

$$T = (N)(s + 1) + 1 \tag{1.54}$$

where $N + 1$ is the total number of key frames and s is the number of interpolations between any two key frames. The error is defined as the summation of the local errors between the TS robot workspace and each frame of the task trajectory divided by the total number of task frames for specific TS design

$$\Delta_{ave} = \frac{\sum_{k=0}^{T-1} \Delta(\tilde{\mathbf{A}}_k, \tilde{\mathbf{W}}_k)}{T}. \tag{1.55}$$

This error value is one value of the cost function which must be minimized. A new set of four task frames is selected and the process is repeated. An exhaustive search of all combinations of four task frames in the task trajectory is utilized. The TS robot with minimum error is the optimum fit to the task trajectory.

Best Fixed Pivot **g**	(4.49, 0.44, 0.10)
Arm Length R	3.84
Error Δ_{ave}	0.001

TABLE 1.3. TS robot design parameters

1.7.1 The optimum TS robot and the task trajectory

The TS robot synthesized to fit the task trajectory obtained from the previous example is shown in Figure 1.4. The fixed pivot **g** and the arm length R are listed in Table 1.3. Figure 1.5 shows the sphere reachable by the wrist of the TS robot. The grey end-effectors show the closest positions to the task frames. The center of the sphere is the location of fixed pivot, **g**.

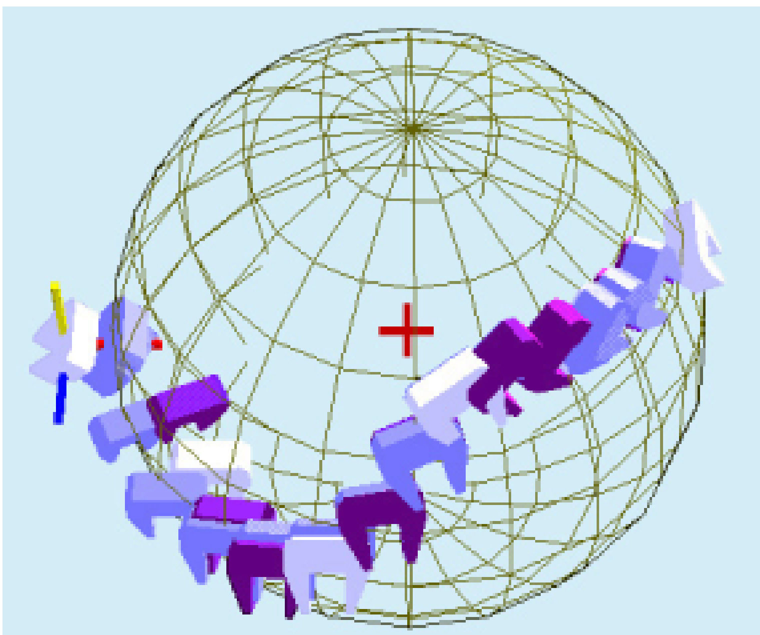

FIGURE 1.5. The TS Robot and the Task Trajecory

1.8 Conclusion

In this chapter, we present a method to design a TS robot that reaches a specified task trajectory. The task trajectory is defined by interpolating a set of key frames selected by the designer. The interpolation is done using a double quaternion representation of the specified key frames to obtain an efficient formulation. The TS robot that best fits this trajectory is determined by minimizing local error between the workspace and the task frames. An example of this design algorithm is presented.

This procedure allows a user to design a TS robot to accomplish a desired task. If the designed robot is not satisfactory, the user alters the task and the procedure is repeated. This interactive process, allows the user to formulate a task as a set of spatial positions and orientations, scan and evaluate candidate devices, including assessment of range of motion and mechanical advantage, and, finally, select a TS robot to achieve the desired performance.

References

[1] W. K. Clifford. Preliminary Sketch of Biquaternions (1873). *Mathematical Papers*, edited by R. Tucker, Macmillan, London, 1882.

[2] P. DeCasteljau. *Outillages methodes calcul.* Technical report, A. Citroen, Paris, 1959.

[3] P. DeCasteljau. *Courbes et surfaces a poles.* Technical report, A. Citroen, Paris, 1963.

[4] K. R. Etzel and J. M. McCarthy. Interpolation of Spatial Displacements Using the Clifford Algebra of E4. *ASME Journal of Mechanical Design.* 121:1-6, 1999.

[5] G. Farin. *Curves and Surfaces for CAGD.* Academic Press, Inc., Chestnut Hill, MA, 1997.

[6] Q. J. Ge. On the Matrix Algebra Realization of the Theory of Biquaternions. *Mechanism Synthesis and Analysis*, DE 70:425-432, ASME, 1994.

[7] Q. J. Ge and D. Kang. Motion Interpolation With G^2 Composite Bezier Motions. *ASME Journal of Mechanical Design*, 117:520-525, 1995.

[8] Q. J. Ge and B. Ravani. Geometric Construction of Bézier Motions. *ASME Journal of Mechanical Design*, 116:749-755, 1994.

[9] W. R. Hamilton. *Elements of Quaternions* (1860), Reprinted by Chelsea Pub., 1969, New York.

[10] C. Innocenti. Polynomial Solution of the Spatial Burmester Problem. *ASME Journal of Mechanical Design*, 117:64-68, 1995.

[11] J. M. McCarthy and Q. Liao. On the Seven Position Synthesis of a 5-SS Platform Linkage. *ASME Journal of Mechanical Design (in press)*.

[12] J. M. McCarthy. *An Introduction to Theoretical Kinematics*, MIT Press, Cambridge, MA, 1990.

[13] K. Shoemake. Animating Rotation with Quaternion Curves. *ACM SIGGRAPH*, 19(3):245-254, 1985.

[14] C. H. Suh and C. W. Radcliffe. *Kinematics and Mechanism Design.* John Wiley and Sons, New York, 1978.