Rational Motion Interpolation Under Kinematic Constraints of Spherical 6*R* Closed Chains

Anurag Purwar

Zhe Jin

Q. J. Ge

Department of Mechanical Engineering,
State University of New York,
Stony Brook, NY 11794-2300

The work reported in this paper brings together the kinematics of spherical closed chains and the recently developed free-form rational motions to study the problem of synthesizing rational interpolating motions under the kinematic constraints of spherical 6R closed chains. The results presented in this paper are an extension of our previous work on the synthesis of piecewise rational spherical motions for spherical open chains. The kinematic constraints under consideration are workspace related constraints that limit the position of the links of spherical closed chains in the Cartesian space. Quaternions are used to represent spherical displacements. The problem of synthesizing smooth piecewise rational motions is converted into that of designing smooth piecewise rational curves in the space of quaternions. The kinematic constraints are transformed into geometric constraints for the design of quaternion curves. An iterative algorithm for constrained motion interpolation is presented. It detects the violation of the kinematic constraints by searching for those extreme points of the quaternion curve that do not satisfy the constraints. Such extreme points are modified so that the constraints are satisfied, and the resulting new points are added to the ordered set of the initial positions to be interpolated. An example is presented to show how this algorithm produces smooth spherical rational spline motions that satisfy the kinematic constraints of a spherical 6R closed chain. The algorithm can also be used for the synthesis of rational interpolating motions that approximate the kinematic constraints of spherical 5R and 4R closed chains within a user-defined tolerance. [DOI: 10.1115/1.2898879]

1 INTRODUCTION

This paper deals with the problem of synthesizing smooth piecewise rational interpolating motions of spherical closed chains under kinematic constraints. More specifically, the problem we seek to solve in this paper is defined as follows: Given a set of positions of the coupler of a spherical 6*R* closed chain, synthesize a smooth piecewise rational motion of the coupler that passes through the given positions and satisfies the kinematic constraints of the spherical 6*R* closed chain. The rational part of the motion is characterized by the trajectory of any point on the coupler being a rational curve in the Cartesian space, and the smoothness is ensured by requiring the motion to be C^2 continuous. It is shown that the presented solution to this problem is readily extended to the spherical closed 4*R* and 5*R* chains as well.

The past two decades have witnessed a significant body of work emanating from the application of well-known curve and surface design algorithms from computer aided geometric design (CAGD) to the field of theoretical kinematics for the purpose of developing rational Bézier and *B*-spline motions of rigid bodies. The idea behind such a synergy is that the problem of designing rational motions of rigid bodies can be transformed into a problem of designing rational curves in a higher dimensional projective space via a special mapping. By choosing the quaternion representation of the spherical displacements, the problem is further reduced to designing curves in the space of quaternions. Rational motions, with applications spanning across areas such as motion animation in computer graphics, task specification in mechanism synthesis, and virtual reality systems as well as Cartesian motion planning in robotics, are an attractive proposition since they integrate well with the industry standard nonuniform rational *B*-spline (NURBS) based computer aided design/computer aided manufacturing (CAD/CAM) system. Furthermore, from a computational perspective, they can easily exploit fast and stable algorithms from CAGD.

The earliest instance of such a synergistic endeavor is found in Shoemake's [1] work, who extended the idea of linear interpolation in affine spaces to the spherical linear interpolation (*Slerp*) of quaternions. Since then, there has been quite some work on the synthesis of spherical motions by designing quaternion based splines by Kim et al. [2], Nielson [3,4], Wang and Joe [5], and Barr et al. [6]. Ge and Ravani [7,8] extended this technique of motion synthesis to the domain of spatial motion interpolation by using a dual quaternion approach. Their work was further refined by Jüttler and Wagner [9], Wagner [10], and Purwar and Ge [11]. A comprehensive list of references on the topic of rational motion design can be found in a survey paper by Röschel [12] and in Purwar and Ge [11].

Despite all the advances in the area of rational motion design, none of the aforementioned work deals with the design of rational motions under the kinematic constraints. The seemingly related work by Horsch and Jüttler [13] sought a direct application of rational motions to the Cartesian motion planning of robots and has not dealt with rational motions under kinematic constraints.

Ge and Larochelle [14] presented a preliminary framework for approximating algebraic motions of four-bar spherical mechanisms with rational *B* spline spherical motions. Recently, Purwar et al. [15] presented algorithms for synthesizing spherical rational motions under the kinematic constraints imposed by spherical 2*R* and 3*R* robot arms. The results presented in this paper are an extension of the results presented in the latter work to spherical closed chains. The scope of the present paper is restricted to spherical rational motions under the kinematic constraints imposed by spherical 4*R*, 5*R*, and 6*R* closed chains.

Contributed by the Mechanisms and Robotics Committee of ASME for publication in the Journal of Mechanical Design. Manuscript received April 15, 2007; final manuscript received October 11, 2007; published online April 14, 2008. Review conducted by José M. Rico. Paper presented at the ASME 2007 Design Engineering Technical Conferences and Computers and Information in Engineering Conferences (DETC2007), Las Vegas, NV, September 4–7, 2007.

In this paper, the spherical displacement of the coupler is represented by a quaternion (see Bottema and Roth [16] and McCarthy [17] for a quaternion representation of displacements). The kinematic constraints of the closed chains are transformed into geometric constraints for the design of quaternion curves. Thus, given a series of coupler positions in Cartesian space, the problem of synthesizing the smooth interpolating rational motion of a spherical 6*R* closed chain is reduced to that of designing a C^2 continuous rational spline constrained to lie on the constraint manifold of the 6*R* chain in the quaternion space. Although in the past, some work has been done on designing splines on quadrics and hypersurfaces (see Gfrerrer [18] and Wang and Joe [19]), the focus in this paper is on presenting a numerical algorithm that can be used to solve 4*R*, 5*R*, and 6*R* closed chain motions. To solve this problem, first, a free-form C^2 continuous *B*-spline quaternion curve interpolating through the given quaternions is designed in the quaternion space. An iterative algorithm detects the violation of the kinematic constraints by searching for those extreme points of the quaternion curve that do not satisfy the geometric constraints. Such extreme points are modified so that the constraints are satisfied and the resulting new points are added to the ordered set of the initial positions to be interpolated. An example is given to show how this algorithm produces smooth spherical rational spline motions that satisfy the kinematic constraints of a spherical 6*R* closed chain.

The organization of the paper is as follows. Section 2 reviews the concept of quaternions and the rational *B*-spline spherical motions. Section 3 reviews the kinematic constraints of spherical 4*R*, 5*R*, and 6*R* spherical closed chains using quaternions. Section 4 presents an algorithm and an example of the problem of rational motion planning for 6*R* spherical closed chains. Section 5 shows how the algorithm presented in the previous section can be used to do rational motion interpolation of 4*R* and 5*R* spherical closed chains, while the kinematic constraints are satisfied approximately within a user-defined tolerance.

2 Quaternion and Spherical Rational Motion

In this section, we review the concept of quaternions and the unconstrained rational spherical motion in so far as necessary for the development of the current paper.

2.1 Unit Quaternion as Rotation. Any rotation in three-dimensional space has a rotation axis and a rotation angle about this axis. Let $\mathbf{s}=(s_x,s_y,s_z)$ denote a unit vector along the axis and let θ denote the angle of rotation. They can be used to define the so-called *Euler–Rodrigues parameters*,

$$q_1 = s_x \sin(\theta/2), \quad q_2 = s_y \sin(\theta/2), \quad q_3 = s_z \sin(\theta/2),$$

$$q_4 = \cos(\theta/2) \tag{1}$$

The Euler–Rodrigues parameters and the quaternion units $1,\mathbf{i},\mathbf{j},\mathbf{k}$ can be combined to define a quaternion of rotation,

$$\mathbf{q} = q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} + q_4 \tag{2}$$

A quaternion $\mathbf{q}$ is, at times, also written as an ordered quadruple (q_1,q_2,q_3,q_4). Since $q_1^2+q_2^2+q_3^2+q_4^2=1$, $\mathbf{q}$ is also called a unit quaternion. See Bottema and Roth [16] and McCarthy [17] for more details on quaternions and the ensuing discussion. The components of a quaternion are related to the matrix form of a rotation given by

$$[R] = \frac{1}{S^2}\begin{bmatrix} q_4^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_4q_3) & 2(q_1q_3+q_4q_2) \\ 2(q_2q_1+q_4q_3) & q_4^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_4q_1) \\ 2(q_3q_1-q_4q_2) & 2(q_3q_2+q_4q_1) & q_4^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{3}$$

where $S^2=q_1^2+q_2^2+q_3^2+q_4^2$. Note that when q_i is replaced by $Q_i=wq_i(i=1,2,3,4)$, where w is a nonzero scalar, the matrix $[\mathbf{R}]$ is unchanged. Thus, the quaternion components of $\mathbf{q}$ can be considered as homogeneous coordinates of a rotation.

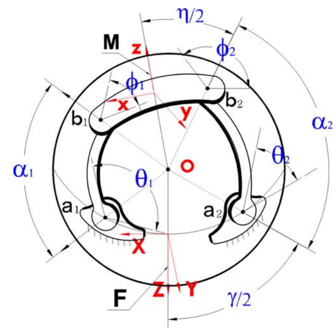

Fig. 1 A 4*R* spherical closed chain

2.2 Unconstrained Rational Spherical Motion. This section reviews the idea of a synthesis of unconstrained (also known as free-form) rational spherical motions. Ravani and Roth [20] considered the components of $\mathbf{q}$ as defining a point in a projective three space, called the *image space* of spherical kinematics. The image space of spherical kinematics is a three-dimensional Cayley–Klein space with an elliptic metric (see Müller [21])

A polynomial curve in the image space corresponds to a rational spherical motion in the Cartesian space. By applying CAGD techniques for designing curves in the image space, we obtain rational motions in the Cartesian space.

Let $\mathbf{Q}_i, i=0,\ldots,n$ be given quaternions; then, the following represents a *B*-spline quaternion curve in the space of quaternions:

$$\mathbf{Q}(u) = \sum_{i=0}^{n} N_{i,p}(u)\mathbf{Q}_i \tag{4}$$

where $N_{i,p}(u)$ are *p*th-degree basis functions. See Farin [22], Hoschek and Lasser [23], and Piegl and Tiller [24] for details on the *B*-spline curves.

A representation for the rational *B*-spline motion in the Cartesian space can be obtained by substituting the coordinates of $\mathbf{Q}(u)$ given by Eq. (4) into the homogeneous matrix $[R]$ (Eq. (3)). It is easy to see that if the *B*-spline curve $\mathbf{Q}(u)$ is expressed as a polynomial function of degree p with parameter u treated as time, then the matrix $[R]$ represents a rational *B*-spline motion of degree $2p$.

3 Kinematic Constraints of Spherical 4*R*, 5*R*, and 6*R* Closed Chains

In this section, we review the constraint manifolds of the spherical 4*R*, 5*R*, and 6*R* closed chains (see McCarthy [17] and Ge [25] for detailed information). The kinematic constraints of the spherical closed chains specify the positions and orientations obtainable by a certain link in the chain.

3.1 Spherical 4*R* Closed Chain. A 4*R* spherical closed chain can be assembled by joining the end links of two spherical 2*R* robot arms. Thus, the kinematic constraints of the 4*R* spherical closed chain can be obtained by, first, appropriately transforming the location of the fixed and the moving frame, writing out the kinematic constraints of the end link from the perspective of the two 2*R* robot arms, and then equating these two perspectives.

Consider a spherical 4*R* robot arm (see Fig. 1) with moving pivots at $\mathbf{b}_1$ and $\mathbf{b}_2$ and fixed pivots at $\mathbf{a}_1$ and $\mathbf{a}_2$. The joint axes at all the pivots intersect at a fixed point $\mathbf{O}$. The angle between the fixed pivots is denoted by γ, and the fixed frame $\mathbf{F}(\mathbf{XYZ})$ is positioned midway along this angle with its x axis normal to the plane defined by the fixed joints $\mathbf{a}_1$ and $\mathbf{a}_2$ in the direction $(\mathbf{a}_1-\mathbf{O})\times(\mathbf{a}_2-\mathbf{O})$. The z axis of the fixed frame $\mathbf{F}$ points in the direction of the bisector of the angle γ between vectors $(\mathbf{a}_1-\mathbf{O})$ and $(\mathbf{a}_2-\mathbf{O})$. The angle between the moving pivots is denoted by η, and the moving frame $\mathbf{M}(\mathbf{xyz})$ is positioned midway along this angle. Similar to the orientation of the fixed frame, its x axis is normal to the plane defined by the moving joints $\mathbf{b}_1$ and $\mathbf{b}_2$ in the direction $(\mathbf{b}_1-\mathbf{O})\times(\mathbf{b}_2-\mathbf{O})$ and the z axis bisects the angle η between the vectors $(\mathbf{b}_1-\mathbf{O})$ and $(\mathbf{b}_2-\mathbf{O})$. The joint angles for the first and second $2R$ robot arms are denoted by θ_1,ϕ_1 and θ_2,ϕ_2, respectively, while the angular spans of the two links joining the coupler are α_1 and α_2. For the first $2R$ robot arm, the moving frame $\mathbf{M}$ can be positioned with respect to the fixed frame $\mathbf{F}$ by the following quaternion product (see Bottema and Roth [16] for quaternion products):

$$\mathbf{Y}_1(\theta_1,\phi_1)=\mathbf{X}(-\gamma/2)\mathbf{D}(\theta_1,\phi_1)\mathbf{X}(\eta/2) \tag{5}$$

where $\mathbf{X}(-\gamma/2)$ and $\mathbf{X}(\eta/2)$ represent rotations about the $\mathbf{X}$ axis by angles $-\gamma/2$ and $\eta/2$, respectively. Using a local coordinate transformation approach to building transformations for robotic chains, these two transformations bring the origin of the fixed frame $\mathbf{F}$ to the fixed pivot $\mathbf{a}_1$ and bring the moving pivot $\mathbf{b}_1$ to the origin of the moving frame $\mathbf{M}$, while $\mathbf{D}(\theta_1,\phi_1)$ represents the transformation of a spherical $2R$ chain. Equation (5) is given in the matrix form by

$$\mathbf{Y}_1(\theta_1,\phi_1)=[\mathbf{C}_1]\mathbf{D}(\theta_1,\phi_1) \tag{6}$$

where $[\mathbf{C}_1]$ is a 4×4 matrix and $\mathbf{D}(\theta_1,\phi_1)$ is written as a 4×1 column vector. We note here that for brevity, we use the same symbol to denote both a quaternion and a column vector and rely on the context to reveal which use is intended. Thus, for example, $\mathbf{D}(\theta_1,\phi_1)$ is written in the quaternion form in the quaternion product given by Eq. (5), while the same symbol in Eq. (6) involving multiplication with a matrix $[\mathbf{C}_1]$ denotes a 4×1 column vector.

Similarly, for the second $2R$ robot arm, the moving frame $\mathbf{M}$ can be positioned with respect to the fixed frame $\mathbf{F}$ by the following transformation:

$$\mathbf{Y}_2(\theta_2,\phi_2)=\mathbf{X}(\gamma/2)\mathbf{D}(\theta_2,\phi_2)\mathbf{X}(-\eta/2) \tag{7}$$

Equation (7) is given in the matrix form by

$$\mathbf{Y}_2(\theta_2,\phi_2)=[\mathbf{C}_2]\mathbf{D}(\theta_2,\phi_2) \tag{8}$$

where $\mathbf{D}(\theta_i,\phi_i)$; $i=1,2$ have been derived in Purwar et al. [15], $\mathbf{X}(\pm\gamma/2)=[\pm\sin(\gamma/4),0,0,\cos(\gamma/4)]$, and $\mathbf{X}(\pm\eta/2)=[\pm\sin(\eta/4),0,0,\cos(\eta/4)]$.

Matrices $[C_1]$ and $[C_2]$ in Eqs. (6) and (8) are orthogonal matrices, given by

$$[\mathbf{C}_1]=[\mathbf{X}(-\gamma/2)^+][\mathbf{X}(\eta/2)^-]=\begin{bmatrix}\cos\tau & 0 & 0 & -\sin\tau\\ 0 & \cos\sigma & \sin\sigma & 0\\ 0 & -\sin\sigma & \cos\sigma & 0\\ \sin\tau & 0 & 0 & \cos\tau\end{bmatrix} \tag{9}$$

$$[\mathbf{C}_2]=[\mathbf{X}(\gamma/2)^+][\mathbf{X}(-\eta/2)^-]=\begin{bmatrix}\cos\tau & 0 & 0 & \sin\tau\\ 0 & \cos\sigma & -\sin\sigma & 0\\ 0 & \sin\sigma & \cos\sigma & 0\\ -\sin\tau & 0 & 0 & \cos\tau\end{bmatrix} \tag{10}$$

where $[\mathbf{X}(\pm\gamma/2)^+]$ and $[\mathbf{X}(\pm\eta/2)^-]$ are the matrix representations of their respective quaternion form (see McCarthy [17] for a matrix representation of quaternion products) and

$$\sigma=(\gamma+\eta)/4,\quad \tau=(\gamma-\eta)/4 \tag{11}$$

Equations (6) and (8) describe the constraint surfaces of the two spherical $2R$ robot arms. Their intersection gives us the kinematic constraints of the $4R$ spherical closed chain, that is, $\mathbf{Y}_1=\mathbf{Y}_2=\mathbf{Y}=(Y_1,Y_2,Y_3,Y_4)$. The algebraic equations for the intersection are obtained by inverting Eqs. (6) and (8) and by using the fact that $[\mathbf{C}_1][\mathbf{C}_2]=[\mathbf{I}]$,

$$\mathbf{D}(\theta_1,\phi_1)=[\mathbf{C}_2]\mathbf{Y}_1(\theta_1,\phi_1),\mathbf{D}(\theta_2,\phi_2)=[\mathbf{C}_1]\mathbf{Y}_2(\theta_2,\phi_2) \tag{12}$$

Substituting Eq. (12) into the equation for the constraint surface of the spherical $2R$ robot arm chain (see Purwar et al. [15]), we obtain the following equations for two quadrics:

$$\mathbf{Y}^T[Q_1]\mathbf{Y}=0$$

$$\mathbf{Y}^T[Q_2]\mathbf{Y}=0 \tag{13}$$

where $[Q_1]$ and $[Q_2]$ are

$$[Q_1]=\begin{bmatrix} c^2\frac{\alpha_1}{2}-s^2\tau & 0 & 0 & s\tau c\tau\\ 0 & c^2\frac{\alpha_1}{2}-s^2\sigma & -s\sigma c\sigma & 0\\ 0 & -s\sigma c\sigma & -s^2\frac{\alpha_1}{2}+s^2\sigma & 0\\ s\tau c\tau & 0 & 0 & -s^2\frac{\alpha_1}{2}+s^2\tau\end{bmatrix} \tag{14}$$

and

$$[Q_2]=\begin{bmatrix} c^2\frac{\alpha_2}{2}-s^2\tau & 0 & 0 & -s\tau c\tau\\ 0 & c^2\frac{\alpha_2}{2}-s^2\sigma & s\sigma c\sigma & 0\\ 0 & s\sigma c\sigma & -s^2\frac{\alpha_2}{2}+s^2\sigma & 0\\ -s\tau c\tau & 0 & 0 & -s^2\frac{\alpha_2}{2}+s^2\tau\end{bmatrix} \tag{15}$$

In the above matrices, c and s represent the cosine and sine functions, respectively. We expand the equation of the two quadrics given by Eq. (13) to

$$F_1(Y_1,Y_2,Y_3,Y_4)=\cos^2\frac{\alpha_1}{2} \tag{16}$$

$$F_2(Y_1,Y_2,Y_3,Y_4)=\cos^2\frac{\alpha_2}{2} \tag{17}$$

where $F_1(Y_1,Y_2,Y_3,Y_4)$ and $F_2(Y_1,Y_2,Y_3,Y_4)$ are given by

$$F_1(Y_1,Y_2,Y_3,Y_4)=\frac{(Y_1\sin\tau-Y_4\cos\tau)^2+(Y_2\sin\sigma+Y_3\cos\sigma)^2}{Y_1^2+Y_2^2+Y_3^2+Y_4^2} \tag{18}$$

and

$$F_2(Y_1,Y_2,Y_3,Y_4)=\frac{(Y_1\sin\tau+Y_4\cos\tau)^2+(Y_2\sin\sigma-Y_3\cos\sigma)^2}{Y_1^2+Y_2^2+Y_3^2+Y_4^2} \tag{19}$$

Equations (16) and (17) characterize the kinematic constraints of a spherical $4R$ closed chain (McCarthy [17]) and are said to define Clifford quadrics (Müller [21]). These kinematic constraints are homogeneous in nature since multiplying $Y_i(i=1,\ldots,4)$ with a common scalar does not change these equations.

3.2 Spherical 5*R* Closed Chain. Consider a spherical $5R$ closed chain (see Fig. 2). The constraint manifold for this spheri-

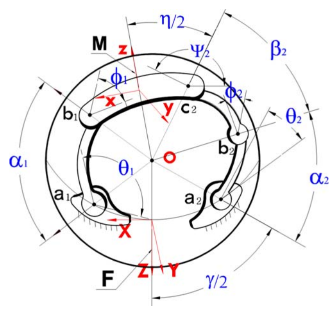

Fig. 2 A 5R spherical closed chain

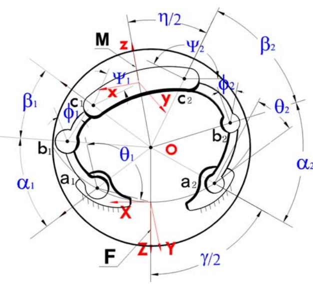

Fig. 3 A 6R spherical closed chain

cal 5R closed chain is the intersection of the constraint surface of a spherical 2R robot arm with the constraint solid of a spherical 3R robot arm.

The constraint surface of a spherical 2R open chain is given by Eq. (6). The constraint solid of a spherical 3R robot arm is given by transforming $\mathbf{D}(\theta_2,\phi_2,\psi_2)$ (see Purwar et al. [15]) by matrix $[\mathbf{C}_2]$

$$\mathbf{Y}_1(\theta_1,\phi_1)=[\mathbf{C}_1]\mathbf{D}(\theta_1,\phi_1)$$

$$\mathbf{Y}_2(\theta_2,\phi_2,\psi_2)=[\mathbf{C}_2]\mathbf{D}(\theta_2,\phi_2,\psi_2) \tag{20}$$

The algebraic equations for the intersection of the above manifolds are

$$\mathbf{Y}^T[Q_1]\mathbf{Y}=0, \quad \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y}=0 \tag{21}$$

where $[Q_1]$ is given by Eq. (14) and $[Q_2(\phi_2)]$ is obtained by substituting α_2 in Eq. (15) with ρ_2. The angle $\rho_2=\rho_2(\phi_2)$ defines the distance between the fixed pivot $\mathbf{a}_2$ and the moving pivot $\mathbf{c}_2$ of the second 3R spherical robot arm and varies according to joint angle ϕ_2 and the angular spans of the links of the 3R arm as follows:

$$\cos\rho_2=\cos\alpha_2\cos\beta_2-\sin\alpha_2\sin\beta_2\cos\phi_2$$

$$\cos(\alpha_2+\beta_2)\leqslant\cos\rho_2\leqslant\cos(\alpha_2-\beta_2) \tag{22}$$

The above inequality is always true since the link lengths given by α_2 and β_2 are never more than π. We can get the kinematic constraints of a spherical 5R closed chain by expanding Eq. (21),

$$F_1(Y_1,Y_2,Y_3,Y_4)=\cos^2\frac{\alpha_1}{2} \tag{23}$$

$$\cos^2\left(\frac{\alpha_2+\beta_2}{2}\right)\leqslant F_2(Y_1,Y_2,Y_3,Y_4)\leqslant\cos^2\left(\frac{\beta_2-\alpha_2}{2}\right) \tag{24}$$

where $F_1(Y_1,Y_2,Y_3,Y_4)$ and $F_2(Y_1,Y_2,Y_3,Y_4)$ are given by Eqs. (18) and (19). Equations (23) and (24) characterize the kinematic constraints of a spherical 5R closed chain and are said to define the constraint manifold for the spherical 5R closed chain (McCarthy [17]).

3.3 Spherical 6*R* Closed Chain. Consider a spherical 6R closed chain (see Fig. 3). The constraint manifold for this spherical 6R closed chain is the intersection of constraint solids of two spherical 3R robot arms. In the figure, angles θ_1,ϕ_1,ψ_1 and θ_2,ϕ_2,ψ_2 denote the joint angles for the first and second spherical 3R robot arms, respectively.

The constraint solids of two spherical 3R closed chains can be obtained by transforming $\mathbf{D}(\theta_1,\phi_1,\psi_1)$ by matrix $[\mathbf{C}_1]$ and transforming $\mathbf{D}(\theta_2,\phi_2,\psi_2)$ by matrix $[\mathbf{C}_2]$ (see Purwar et al. [15] for expressions of the parametrized solids given by $\mathbf{D}(\theta_i,\phi_i,\psi_i);i=1,2$):

$$\mathbf{Y}_1(\theta_1,\phi_1,\psi_1)=[\mathbf{C}_1]\mathbf{D}(\theta_1,\phi_1,\psi_1)$$

$$\mathbf{Y}_2(\theta_2,\phi_2,\psi_2)=[\mathbf{C}_2]\mathbf{D}(\theta_2,\phi_2,\psi_2) \tag{25}$$

The algebraic equations for the intersection of the above manifolds are

$$\mathbf{Y}^T[Q_1(\phi_1)]\mathbf{Y}=0, \quad \mathbf{Y}^T[Q_2(\phi_2)]\mathbf{Y}=0 \tag{26}$$

where $[Q_2(\phi_2)]$ has been already defined and $[Q_1(\phi_1)]$ is obtained by substituting α_1 in Eq. (14) with $\rho_1(\phi_1)$. The angle $\rho_1=\rho_1(\phi_1)$ defines the distance between the fixed pivot $\mathbf{a}_1$ and the moving pivot $\mathbf{c}_1$ of the first 3R spherical robot arm and varies according to the joint angle ϕ_1 and the angular spans of the links of the first 3R robot arm as follows:

$$\cos\rho_1=\cos\alpha_1\cos\beta_1-\sin\alpha_1\sin\beta_1\cos\phi_1 \tag{27}$$

We can derive the kinematic constraints of a spherical 6R closed chain from Eq. (26),

$$\cos^2\left(\frac{\alpha_1+\beta_1}{2}\right)\leqslant F_1(Y_1,Y_2,Y_3,Y_4)\leqslant\cos^2\left(\frac{\beta_1-\alpha_1}{2}\right) \tag{28}$$

$$\cos^2\left(\frac{\alpha_2+\beta_2}{2}\right)\leqslant F_2(Y_1,Y_2,Y_3,Y_4)\leqslant\cos^2\left(\frac{\beta_2-\alpha_2}{2}\right) \tag{29}$$

where $F_1(Y_1,Y_2,Y_3,Y_4)$ and $F_2(Y_1,Y_2,Y_3,Y_4)$ are given by Eqs. (18) and (19).

Equations (28) and (29) characterize the kinematic constraints of a spherical 6R closed chain and are said to define the constraint manifold for the spherical 6R closed chain (McCarthy [17]).

4 Rational Motions of Spherical 6*R* Closed Chain

In this section, we present an algorithm for synthesizing C^2 continuous piecewise rational motions of spherical 6R closed chain under the kinematic constraints derived in the previous section. We will show that the same algorithm can be applied for the synthesis of interpolating rational motions of spherical 4R and 5R closed chains with the kinematic constraints being satisfied only approximately, but within a user-defined tolerance.

The positions of the coupler of a spherical 6R closed chain can be given by either Cartesian coordinates $(s_{xi},s_{yi},s_{zi},\theta_i)$ or joint coordinates (θ_i,ϕ_i,ψ_i). Equation (1) can be used to convert Cartesian coordinates into quaternions, while Eq. (25) can be used to convert joint coordinates into quaternions. If the given positions of the coupler are not connectable with a continuous trajectory, then the mechanism has to be disassembled and reassembled to interpolate the positions. This is commonly known as an "assembly mode defect." Recently, Schröker et al. [26] and Schröcker and Husty [27] presented fast numerical algorithms for determining if two task positions of planar and spherical four-bar mechanisms suffer from this defect. The algorithm presented below assumes that the given positions are on the same branch.

4.1 C^2 Interpolating Rational Motion for Spherical 6*R* Closed Chain. Given: a set of positions of the coupler link of a spherical 6R closed chain in its workspace, the corresponding parameter values $u_i(i=1,\ldots,n)$, angular spans of the links given by

$\alpha_i, \beta_i (i=1,2)$, and the angular distance between the two moving pivots and between the two fixed pivots given by η and γ, respectively. Find: a C^2 rational motion of the coupler link that interpolates the given positions at the parameter values subject to the kinematic constraints of the spherical $6R$ closed chain.

In the ensuing discussion over a general description of the algorithm, we assume that the given positions of the coupler link are represented by quaternion coordinates given by $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$, $i=1,\ldots,n$. These quaternions map into points in the space of quaternions.

4.1.1 Detection of Kinematic Constraints Violation. Finding a motion that satisfies the requirements stated earlier in this section begins with the construction of a C^2 cubic B-spline image curve[1] $\mathbf{Y}(u)$ that interpolates $\mathbf{Y}_i$ at parameter values u_i. Such a curve does not automatically satisfy the kinematic constraints given by Eqs. (28) and (29). Specifically, kinematic constraints are considered violated at those points of the synthesized image curve $\mathbf{Y}(u)$, where the following functions (rewritten from Eqs. (18) and (19)) do not satisfy the inequality given in Eqs. (28) and (29):

$$F_1(u) = \frac{(Y_1(u)\sin\tau - Y_4(u)\cos\tau)^2 + (Y_2(u)\sin\sigma + Y_3(u)\cos\sigma)^2}{Y_1^2(u) + Y_2^2(u) + Y_3^2(u) + Y_4^2(u)} \tag{30}$$

$$F_2(u) = \frac{(Y_1(u)\sin\tau + Y_4(u)\cos\tau)^2 + (Y_2(u)\sin\sigma - Y_3(u)\cos\sigma)^2}{Y_1^2(u) + Y_2^2(u) + Y_3^2(u) + Y_4^2(u)} \tag{31}$$

To detect violations of the kinematic constraints, extrema of functions $F_1(u)$ and $F_2(u)$ are calculated using the following equations:

$$\frac{dF_1(u)}{du} = 0, \quad \frac{dF_2(u)}{du} = 0 \tag{32}$$

If an extremum falls outside the limits in any of the inequalities (Eqs. (28) and (29)), the constraints are considered violated. We call a point on the image curve corresponding to an extremum of $F_1(u)$ or $F_2(u)$ an extreme point. A C^2 cubic B-spline curve can be easily converted into a piecewise Bézier form using standard methods (see Farin [22]), where each Bézier segment has an algebraic form. The piecewise Bézier form allows easy evaluation of Eq. (32) for each of the L Bézier segments, $\mathbf{Y}_i(u)$, $i = 0,1,\ldots,L-1$, that make up the B-spline curve $\mathbf{Y}(u)$.

4.1.2 Modifying the Curve to Satisfy Kinematic Constraints. By restricting to operate in the image space, we have transformed the kinematic constraints into geometric constraints. Thus, we can take a purely geometric approach to modify the synthesized image curve in such a way that the kinematic constraints are satisfied. Our approach is to replace the extreme point that violates the kinematic constraints with a new point that satisfies kinematic constraints, thereby dragging the curve to the inside of the constraint manifold. To ensure that the image curve changes minimally, the new point should be closest possible to the extreme point. This approach is realized in a two step iterative process: First, a new point is generated that satisfies the kinematic constraints and is minimally away from the extreme point; then, this new point is added (at the parameter value corresponding to the extreme point) to the initial set of given points, and a new C^2 cubic B-spline image curve that interpolates the new point is generated as well. If the new image curve detects any further violation of the kinematic constraints as outlined in the previous subsection, this two-step process is repeated until no extreme points that violate the kinematic constraints are found. This process may yield many new points before the kinematic constraints are satisfied completely. A visual interpretation of this approach is that the image curve is repeatedly "bent in" or "bent out" in the vicinity of the constraints violating extreme points in its image space while still interpolating through the given positions, until it fits in the constraint manifold defined by the kinematic constraints.

[1] Designing a C^2 B-spline curve is a standard scheme in CAGD (see Farin [22], Hoschek and Lasser [23], and Piegl and Tiller [24]).

4.1.3 Generating New Points. Since we require the new points to be minimally away from the extreme points, the issue of finding new points can be seen as a normal distance minimization problem in the image space subject to certain constraints. Ravani and Roth [28] proposed a general algebraic method for approximate normal distance calculation between the image curve and a given position in the image space. Later on, Bodduluri and McCarthy [29] used their method for a finite position synthesis of a spherical four-bar motion. In what follows, we outline this method and deal with the issue of finding new points within the framework of this method for a solution.

We first rewrite the kinematic constraint equations from Eqs. (18) and (19) and Eqs. (28) and (29) as follows:

$$(Y_1\sin\tau - Y_4\cos\tau)^2 + (Y_2\sin\sigma + Y_3\cos\sigma)^2 = r_1^2\sum_{i=1}^{4} Y_i^2 \tag{33}$$

$$(Y_1\sin\tau + Y_4\cos\tau)^2 + (Y_2\sin\sigma - Y_3\cos\sigma)^2 = r_2^2\sum_{i=1}^{4} Y_i^2 \tag{34}$$

Here, r_1 and r_2 are variables that should satisfy the following inequalities:

$$\left|\cos\left(\frac{\alpha_1+\beta_1}{2}\right)\right| \leq r_1 \leq \left|\cos\left(\frac{\alpha_1-\beta_1}{2}\right)\right|$$
$$\left|\cos\left(\frac{\alpha_2+\beta_2}{2}\right)\right| \leq r_2 \leq \left|\cos\left(\frac{\alpha_2-\beta_2}{2}\right)\right| \tag{35}$$

The new point should satisfy Eqs. (33) and (34) as well as the following:

$$Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2 = w^2 \tag{36}$$

where w is a nonzero scalar (or weight) for the quaternion coordinates of the new point. Here, without any loss of generality, we choose $w=1$ since as we showed before in Sec. 2, the coordinates of $\mathbf{Y}$ are homogeneous coordinates.

In the context of a spherical four-bar mechanism synthesis, the method used by Bodduluri and McCarthy [29] solves the design problem of determining an image curve that passes through or near a set of given points. The image curve is algebraically given by the intersection of two quadric hypersurfaces (constraint surfaces) and is expressed as functions of the design parameters, such as link lengths and pivot locations. They seek the values of design parameters, which reduce the error between the image curve and the given points. The error at each given position is defined as the normal distance to the image curve. They define the total error as the sum of squares of each position error. The method is approximate in the sense that the constraint surfaces are approximated by their tangent hyperplane in the vicinity of the desired position. We use the same approach to find a new point $\mathbf{Y}$ that satisfies the constraints given by Eqs. (33), (34), and (36) and is also at the minimum distance from the extreme point ($\mathbf{Y}^*$) that violates the kinematic constraints. In our case, we define $\mathbf{e}=\mathbf{Y}-\mathbf{Y}^*$ as the normal error vector at the position $\mathbf{Y}^*$. Minimization of the square of the magnitude of $\mathbf{e}$ subject to Eq. (35) would yield r_1 and r_2. This, in turn, would give the new point $\mathbf{Y}=\mathbf{e}^*+\mathbf{Y}^*$, where $\mathbf{e}^*$ is the optimized normal distance. This new point $\mathbf{Y}$ may not satisfy the kinematic constraints because the constraint surfaces are only approximated in this approach. In that case, this new point $\mathbf{Y}$ is set as the new extreme point $\mathbf{Y}^*$, and the process described above is repeated.

The optimization procedure outlined above involves the tangent hyperplane approximation to the hypersurfaces. For that, we first rewrite Eqs. (33), (34), and (36) as follows:

$$H_1(\mathbf{Y}):(Y_1 \sin \tau - Y_4 \cos \tau)^2 + (Y_2 \sin \sigma + Y_3 \cos \sigma)^2 - r_1^2 = h_1 \tag{37}$$

$$H_2(\mathbf{Y}):(Y_1 \sin \tau + Y_4 \cos \tau)^2 + (Y_2 \sin \sigma - Y_3 \cos \sigma)^2 - r_2^2 = h_2 \tag{38}$$

$$G(\mathbf{Y}):Y_1^2 + Y_2^2 + Y_3^2 + Y_4^2 - 1 = g \tag{39}$$

where $h_1 = h_2 = g = 0$.

These surfaces can be approximated by the tangent hyperplanes at a point $\mathbf{Y}^* = (Y_1^*, Y_2^*, Y_3^*, Y_4^*)$ as follows:

$$0 = H_1(\mathbf{Y}^*) + \sum_{i=1}^{4} \frac{\partial H_1(\mathbf{Y}^*)}{\partial Y_i} \Delta Y_i$$

$$0 = H_2(\mathbf{Y}^*) + \sum_{i=1}^{4} \frac{\partial H_2(\mathbf{Y}^*)}{\partial Y_i} \Delta Y_i$$

$$0 = \sum_{i=1}^{4} 2Y_i^* \Delta Y_i \tag{40}$$

These equations can be assembled as follows:

$$\begin{bmatrix} \frac{\partial H_1}{\partial Y_1} & \frac{\partial H_1}{\partial Y_2} & \frac{\partial H_1}{\partial Y_3} & \frac{\partial H_1}{\partial Y_4} \\ \frac{\partial H_2}{\partial Y_1} & \frac{\partial H_2}{\partial Y_2} & \frac{\partial H_2}{\partial Y_3} & \frac{\partial H_2}{\partial Y_4} \\ 2Y_1^* & 2Y_2^* & 2Y_3^* & 2Y_4^* \end{bmatrix} \begin{bmatrix} \Delta Y_1 \\ \Delta Y_2 \\ \Delta Y_3 \\ \Delta Y_4 \end{bmatrix} = \begin{bmatrix} -H_1 \\ -H_2 \\ 0 \end{bmatrix} \tag{41}$$

The above can also be written as

$$[J]\mathbf{e} = \mathbf{v} \tag{42}$$

Bodduluri and McCarthy [29] solved for the normal error vector $\mathbf{e}$ by minimizing the Lagrangian function given as follows:

$$L(\mathbf{e}, \mathbf{A}) = \mathbf{e}^T \mathbf{e} + \mathbf{A}^T([J]\mathbf{e} - \mathbf{v}) \tag{43}$$

where $\mathbf{A} = (A_1, A_2, A_3)$ is a vector of Lagrange multipliers, $\mathbf{e} = (Y_1 - Y_1^*, Y_2 - Y_2^*, Y_3 - Y_3^*, Y_4 - Y_4^*)$, $\mathbf{v} = (-H_1(\mathbf{Y}^*), -H_2(\mathbf{Y}^*), 0)$. Putting the condition for a minimum as

$$\frac{\partial L}{\partial e_i} = 0, \quad i = 1,2,3,4 \tag{44}$$

where (e_1, e_2, e_3, e_4) are the coordinates of the vector $\mathbf{e}$, and assembling the solution equations, we obtain

$$2\mathbf{e}^T + \mathbf{A}^T [J] = 0 \tag{45}$$

Thus, if $\mathbf{e}^*$ designates the solution to the error vector (or the normal distance), then it should satisfy the equations

$$[J]\mathbf{e}^* = \mathbf{v}$$

$$2\mathbf{e}^* + [J]^T \mathbf{A} = 0 \tag{46}$$

Equation (46) gives an explicit formula for the solution error vector $\mathbf{e}^*$ in terms of variables r_1 and r_2 (see Bodduluri and McCarthy [29]),

$$\mathbf{e}^* = [J]^T([J][J]^T)^{-1}\mathbf{v} \tag{47}$$

Thus, we can determine the variables r_1 and r_2 by optimizing the function

$$E(\mathbf{Y}, r_1, r_2) = (\mathbf{e}^*)^T \mathbf{e}^* \tag{48}$$

subject to constraints given by Eq. (35).

With normal error vector $\mathbf{e}^*$ known, the new point is given by

$$\mathbf{Y} = \mathbf{e}^* + \mathbf{Y}^* \tag{49}$$

Since the constraint surfaces have been approximated by their tangent hyperplanes in the vicinity of the extreme point $\mathbf{Y}^*$, this new point $\mathbf{Y}$ may or may not lie inside the constraint solids given by Eqs. (28) and (29). If the new point does not satisfy the constraints, the newly obtained point $\mathbf{Y}$ is set as a new $\mathbf{Y}^*$ and the procedure described above is repeated from Eq. (40) until a new point $\mathbf{Y}$ that satisfies the kinematic constraints is obtained. This new point not only satisfies the kinematic constraints but is also at the shortest distance from the original extreme point $\mathbf{Y}^*$. With this new point added to the set of initial points in the quaternion space, a new C^2 *B*-spline is interpolated to the points. If the new image curve detects any further violation of the kinematic constraints, the optimization process is repeated until no further extreme points that violate the kinematic constraints are found. This process at the end gives an interpolating motion that satisfies the kinematic constraints. Now, we present the following algorithm:

(1) Convert given positions of the coupler link into unit quaternions (points in image space) $\mathbf{Y}_i = (Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$ using either Eq. (1) or Eq. (25).
(2) Current list of points to be interpolated = given points ($\mathbf{Y}_i, i = 1, \ldots, n$).
(3) Construct a C^2 cubic *B*-spline curve $\mathbf{Y}(u)$ that interpolates a current list of points.
    (a) Evaluate the extrema of $F_1(u)$ and $F_2(u)$ using Eq. (32). Say, the extrema are found at $u = u_1^*$ and at $u = u_2^*$ for $F_1(u)$ and $F_2(u)$, respectively. Designate extreme points on the curve as $\mathbf{Y}_1^*$ and $\mathbf{Y}_2^*$.
    (b) Check if $\mathbf{Y}_1^*$ and $\mathbf{Y}_2^*$ satisfy the kinematic constraints (Eqs. (28) and (29)).
    (c) If the curve is constrained, continue to step 5.
    (d) Otherwise, find new points $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ (Eqs. (40)–(49)).
    (e) Check if the new points $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ satisfy kinematic constraints (Eqs. (28) and (29)).
        (i) If yes, continue to (f)
        (ii) Otherwise, set $\mathbf{Y}_1^* = \mathbf{Y}(u_1^*)$ and $\mathbf{Y}_2^* = \mathbf{Y}(u_2^*)$, and repeat from (d).
    (f) Add $\mathbf{Y}(u_1^*)$ and $\mathbf{Y}(u_2^*)$ to the current list of points to be interpolated and go to step 3.
(4) The image curve $\mathbf{Y}(u)$ defines a C^2 interpolating piecewise rational motion of degree 6 after the substitution into Eq. (3).

We have observed that this algorithm always converges as long as the interpolating points are on the same branch. In this algorithm, the *B*-spline curve is generated using a global interpolation scheme (Piegl and Tiller [24]). In this scheme, although moving one of the interpolating points changes the curve globally, the change in the curve diminishes away from the modification point. Next, we show via an example that the algorithm converges fast as well.

Table 1 gives the quaternion coordinates $(Y_{i1}, Y_{i2}, Y_{i3}, Y_{i4})$ for five positions of the coupler link of a spherical 6*R* closed chain along with their parameter values. The table also gives the link lengths and the angular distance between moving and fixed pivots. The range of the inequality in the kinematic constraints given by Eqs. (28) and (29) is shown in Table 2.

To visualize the constraint solids in three-dimensional Euclidean space, we recast the constraint equations (Eqs. (28) and (29)) in terms of Rodrigues parameters (see Bottema and Roth [16])

Table 1 Quaternion coordinates of the given positions of the coupler link of spherical 6*R* closed chain ($\alpha_1=\pi/4$, $\beta_1=\pi/6$, $\alpha_2=\pi/3$, $\beta_2=\pi/6$, $\eta=\pi/3$, and $\gamma=\pi/2$)

i	$\mathbf{Y}_i=(Y_{i1},Y_{i2},Y_{i3},Y_{i4})$	u_i
1	(−0.2636, 0, 0, 0.9646)	0.0
2	(−0.0039, −0.2949, −0.0814, 0.9520)	2.0
3	(−0.0110, −0.5860, 0.0150, 0.8101)	5.0
4	(−0.4910, −0.4170, −0.0005, 0.7649)	7.0
5	(−0.5620, 0, 0, 0.8271)	10.0

given by $(Y_1/Y_4, Y_2/Y_4, Y_3/Y_4)$. Geometrically, this is equivalent to observing the intersection of constraint solids with hyperplane $Y_4=1$. Figure 4 shows such an intersection of the constraint surfaces of the four-dimensional solids described by Eqs. (28) and (29) with the hyperplane $Y_4=1$. Various surfaces are indicated by the limits of the inequalities (Table 2). These surfaces are hyperboloid of one sheet, and we note here that contrary to what the reader might expect, the inner and outer hyperboloids indicate the upper and the lower, respectively, range of the kinematic constraint inequalities (Eqs. (28) and (29)). The volume bounded by the surfaces is the valid region in which the image curve should remain for the constraints to be satisfied.

Figure 5 shows the kinematic constraint surfaces and the unconstrained interpolation of the given positions from an angle different from that shown in Fig. 4. The extreme points that violate the kinematic constraints are shown by the star points (★). Given positions interpolated by the image curve are indicated by (■). This figure shows that there are two extreme points, each of which violate one of the kinematic constraints. One of the extreme point is seen to be at $u^*=0.92$, where $F_1=1.0$. This point is outside the bounds of $F_1(u): 0.63 \leq F_1 \leq 0.98$. It is not difficult to see from the figure that this extreme point does not violate the other kinematic constraint given by $0.5 \leq F_2 \leq 0.93$. This point can be clearly seen between the two kinematic constraint surfaces given by the limits of $F_2(u)$. Similarly, the other extreme point is seen to be at $u^*=8.74$, where $F_2=0.38$. This point violates the lower limit of the inequality $0.5 \leq F_2$. It can also be seen from the figure that this extreme point does not violate the other kinematic constraint.

Figures 6 and 7 focus on the violation of different constraints and visually demonstrate the application of the algorithm in constraining the image curve. Figure 6, which is obtained by rotating and zooming in on Fig. 5, focuses on the kinematic constraint $0.5 \leq F_2 \leq 0.93$ and the extreme point that violates this constraint. It shows the unconstrained curve with one of the extreme points that violates the constraint $0.5 \leq F_2$. This point is clearly seen to

Table 2 Kinematic constraints of a given spherical 6*R* closed chain

i	Kinematic constraints F_i
1	$0.63 \leq F_1 \leq 0.98$
2	$0.5 \leq F_2 \leq 0.93$

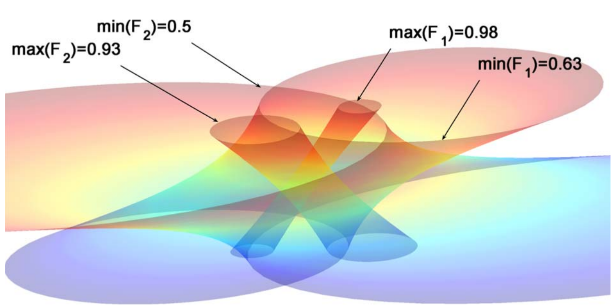

Fig. 4 Kinematic constraint surfaces obtained by intersection with the hyperplane $Y_4=1$

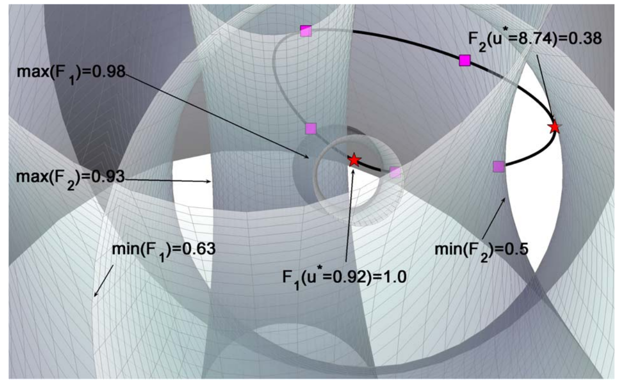

Fig. 5 Unconstrained interpolation and the kinematic constraint surfaces obtained by intersection with the hyperplane $Y_4=1$. The image curve is shown as a continuous curve, the two extreme points that violate kinematic constraints as (★) and the given positions as (■).

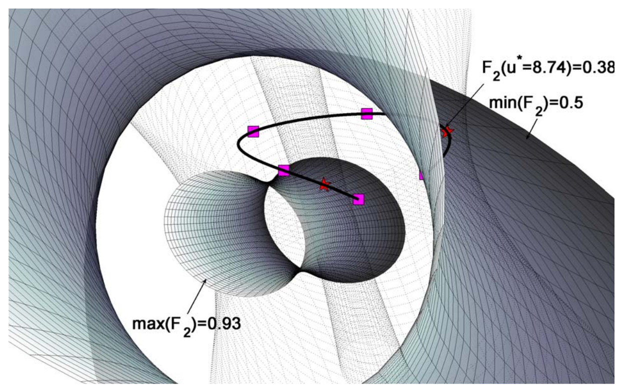

Fig. 6 A zoomed-in and rotated view of Fig. 5: One extreme point with $F_2(8.74)=0.38$ violates the kinematic constraint: $0.5 \leq F_2$. Surfaces given by the limits of the other constraint are shown in light broken lines.

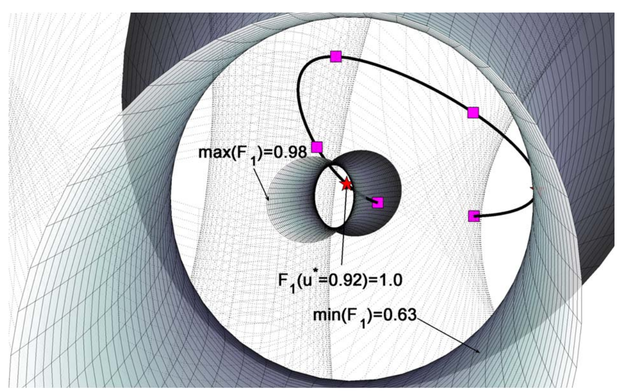

Fig. 7 A zoomed-in and rotated view of Fig. 5: One extreme point with $F_1(0.92)=1.0$ violates the kinematic constraint: $F_1 \leq 0.98$. Surfaces given by the limits of the other constraint are shown in light broken lines.

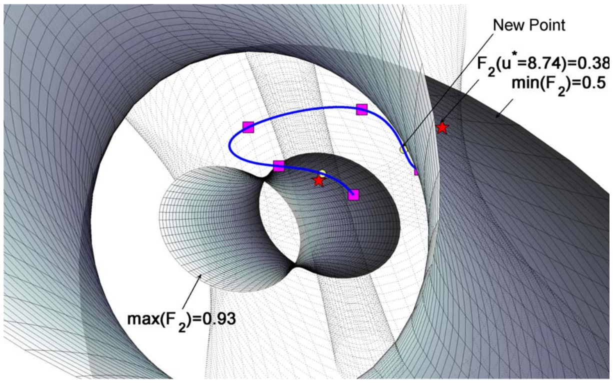

Fig. 8 Constrained interpolation (compare to Fig. 6). The labeled new point (o) is the point that replaces the labeled extreme point.

be outside the volume defined by the constraint surfaces. Surfaces given by the limits of the other constraint $F_1(u)$ (Eq. (28)), which is not violated by this extreme point, are shown in light broken lines. It should also be noted that the other unlabeled extreme point does not violate the constraint $0.5 \leqslant F_2 \leqslant 0.93$.

Similarly, Fig. 7 shows from another angle the kinematic constraint surfaces given by the limits of $F_1(u)$ and the unconstrained curve. This figure shows the other extreme point that violates the upper limit ($F_1 \leqslant 0.98$) of the kinematic constraints given by the inequality in Eq. (28). Also shown are the constraint surfaces given by the limits of $F_2(u)$ in light broken lines.

The application of the algorithm presented earlier is illustrated in Figs. 8 and 9, where in two iterations, the algorithm produces a constrained curve. On a Pentium 4 2.0 GHz system with 512 Mbytes of primary memory, the algorithm ran for less than 5 s. Two new points that replace the extreme points are added in the process, which are indicated by o . It might seem from Figs. 8 and 9 that the two new points added to satisfy the limits on $F_2(u)$ and $F_1(u)$ are not minimally away from their corresponding extreme point, respectively. However, this is not true. A different view of the locations of the two new points verifies this assertion visually in Figs. 8 and 9; e.g., the new point labeled in Fig. 8 is seen to be very close to the outer constraint surface shown in light broken lines in Fig. 9.

The final interpolating curve satisfies all the constraints, which—in the Cartesian space—translates into a C^2 continuous B-spline rational motion of the spherical 6R closed chain.

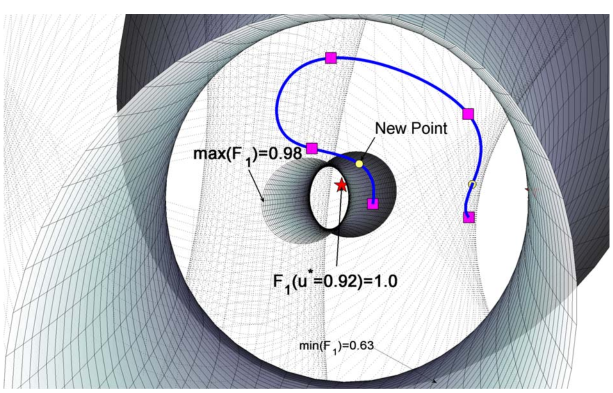

Fig. 9 Constrained interpolation (compare to Fig. 7). The labeled new point (o) is the point that replaces the labeled extreme point.

5 Rational Motions of Spherical 4*R* and 5*R* Closed Chain

Constraint equations of spherical 4R and 5R closed chains can be seen as a special case of the constraint equations of spherical 6R closed chain. By turning the equality equations of 4R and 5R closed chains (Eqs. (16), (17), and (23)) into inequalities, we can apply the same algorithm as presented before to do constrained interpolation for 4R and 5R closed chains as well.

For spherical 4R closed chains, we modify the kinematic constraints (Eqs. (16) and (17)) as follows:

$$\cos^2\left(\frac{\alpha_1+\varepsilon_1}{2}\right) \leqslant F_1(Y_1,Y_2,Y_3,Y_4) \leqslant \cos^2\left(\frac{\alpha_1-\varepsilon_1}{2}\right) \quad (50)$$

$$\cos^2\left(\frac{\alpha_2+\varepsilon_2}{2}\right) \leqslant F_2(Y_1,Y_2,Y_3,Y_4) \leqslant \cos^2\left(\frac{\alpha_2-\varepsilon_2}{2}\right) \quad (51)$$

where ε_1 and ε_2 are user-defined tolerances. Thus, by choosing these values to be as small as possible, the user can use the same algorithm to do constraint interpolation to a desired degree of satisfaction. The addition of new points is handled slightly differently; we use the following equations in place of Eq. (35):

$$r_1 = \cos\frac{\alpha_1}{2}, \quad r_2 = \cos\frac{\alpha_2}{2} \quad (52)$$

In the case of a spherical 5R closed chain, we modify the kinematic constraint in Eq. (23) to be the same as the kinematic constraint in Eq. (50). This puts the first kinematic constraint in the inequality form, while we use the second constraint equation unchanged, i.e., Eq. (24). The new points are added in the same way as illustrated before; however, the value of r_1 in Eq. (35) is replaced with the value given in Eq. (52).

6 Conclusions

In this paper, we developed a method for synthesizing piecewise rational motions subject to the kinematic constraints of the spherical 6R closed chain. We presented an algorithm for synthesizing C^2 continuous piecewise rational motions for a spherical 6R closed chain. It was shown that the algorithm was general enough to handle the spherical 4R and 5R closed chains as well. An example and visualizations were presented to illustrate the effectiveness and the working of the algorithm.

Acknowledgment

The support of National Science Foundation under Grant No. DMI-0500064 is gratefully acknowledged. We also thank the reviewers for their suggestions and comments in helping us improve this paper.

References

[1] Shoemake, K., 1985, "Animating Rotation With Quaternion Curves," *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, pp. 245–254.
[2] Kim, M.-J., Kim, M.-S., and Shin, S. Y., 1995, "A C^2 Continuous B-Spline Quaternion Curve Interpolating a Given Sequence of Solid Orientations," *Proceedings of the Computer Animation, Proceedings of the Computer Animation*, IEEE Computer Society, pp. 19–21.
[3] Nielson, G., 1993, "Smooth Interpolation of Orientations," *Computer Animation, Models and Techniques in Computer Animation*, Springer, New York, pp. 75–93.
[4] Nielson, G. M., 2004, "Nu-Quaternion Splines for the Smooth Interpolation of Orientations." IEEE Trans. Vis. Comput. Graph., **10**(2), pp. 224–229.
[5] Wang, W., and Joe, B., 1993, "Orientation Interpolation in Quaternion Space Using Spherical Biarcs," *Graphics Interface '93*, Canadian Information Processing Society, pp. 24–32.
[6] Barr, A. H., Currin, B., Gabriel, S., and Hughes, J. F., 1992, "Smooth Interpolation of Orientations With Angular Velocity Constraints Using Quaternions," Comput. Graph., **26**(2), pp. 313–320.
[7] Ge, Q. J., and Ravani, B., 1994, "Computer-Aided Geometric Design of Motion Interpolants," ASME J. Mech. Des., **116**(3), pp. 756–762.
[8] Ge, Q. J., and Ravani, B., 1994, "Geometric Construction of Bezier Motions," ASME J. Mech. Des., **116**(3), pp. 749–755.

[9] Juttler, B., and Wagner, M. G., 1996, "Computer-Aided Design With Spatial Rational B-Spline Motions," ASME J. Mech. Des., **118**(2), pp. 193–201.
[10] Wagner, M. G., 1994, "A Geometric Approach to Motion Design," Ph.D. thesis, Technische Universität Wien.
[11] Purwar, A., and Ge, Q. J., 2005, "On the Effect of Dual Weights in Computer Aided Design of Rational Motions," ASME J. Mech. Des., **127**(5), pp. 967–972.
[12] Röschel, O., 1998, "Rational Motion Design: A Survey," Comput.-Aided Des., **30**(3), pp. 169–178.
[13] Horsch, T., and Juttler, B., 1998, "Cartesian Spline Interpolation for Industrial Robots," Comput.-Aided Des., **30**(3), pp. 217–224.
[14] Ge, Q. J., and Larochelle, P. M., 1999, "Algebraic Motion Approximation With Nurbs Motions and Its Application to Spherical Mechanism Synthesis," ASME J. Mech. Des., **121**(4), pp. 529–532.
[15] Purwar, A., Zhe, J., and Ge, Q. J., 2006, "Computer Aided Synthesis of Piecewise Rational Motions for Spherical 2R And 3R Robot Arms," *Proceedings of IDETC/CIE 2006, ASME 2006 International Design Engineering Technical Conferences*, Paper No. DETC2006-99650.
[16] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, North-Holland, Amsterdam.
[17] McCarthy, J. M., 1990, *Introduction to Theoretical Kinematics*, MIT, Cambridge, MA.
[18] Gfrerrer, A., 1999, "Rational Interpolation on a Hypersphere," Comput. Aided Geom. Des., **16**(1), pp. 21–37.
[19] Wang, W., and Joe, B., 1997, "Interpolation on Quadric Surfaces With Rational Quadratic Spline Curves," Comput. Aided Geom. Des., **14**(3), pp. 207–230.
[20] Ravani, B., and Roth, B., 1984, "Mappings of Spatial Kinematics," ASME J. Mech., Transm., Autom. Des., **106**(3), pp. 341–347.
[21] Müller, H. R., 1962, *Sphärische Kinematik*, Deutscher Ver-lag der Wissenschaften, Berlin.
[22] Farin, G., 1996, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed., Academic, New York.
[23] Hoschek, J., and Lasser, D., 1993, *Fundamentals of Computer Aided Geometric Design*, A. K. Peters, Wellesley, MA.
[24] Piegl, L., and Tiller, W., 1995, *The Nurbs Book*, Springer, Berlin.
[25] Ge, Q. J., 1990, "Kinematics Constraints as Algebraic Manifolds in the Clifford Algebra of Projective Three Space," Ph.D. thesis, University of California, Irvine.
[26] Schröcker, H.-P., Husty, M. L., and McCarthy, J. M., 2007, "Kinematic Mapping Based Assembly Mode Evaluation of Planar Four-Bar Mechanisms," ASME J. Mech. Des., **129**(9), pp. 924–929.
[27] Schröcker, H.-P., and Husty, M. L., 2007, "Kinematic Mapping Based Assembly Mode Evaluation of Spherical Four-Bar Mechanisms," *IFToMM 2007, the 12th World Congress in Mechanism and Machine Science*, J.-P. Merlet and M. Dahan, eds., Vol. 129, pp. 924–929.
[28] Ravani, B., and Roth, B., 1983, "Motion Synthesis Using Kinematic Mappings," ASME J. Mech., Transm., Autom. Des., **105**(3), pp. 460–467.
[29] Bodduluri, R. M. C., and McCarthy, J. M., 1992, "Finite Position Synthesis Using Image Curve of a Spherical Four-Bar Motion," ASME J. Mech. Des., **114**(1), pp. 55–60.