Hai-Jun Su
e-mail: suh@eng.uci.edu
Robotics and Automation Laboratory,
University of California, Irvine,
Irvine, CA 92697

Peter Dietmaier
e-mail: dietmaier@mech.tu-graz.ac.at
Institute of General Mechanics, TU-Graz

J. Michael McCarthy
e-mail: jmmccart@uci.edu
Robotics and Automation Laboratory,
University of California, Irvine,
Irvine, CA 92697

Trajectory Planning for Constrained Parallel Manipulators

This paper presents an algorithm for generating trajectories for multi-degree of freedom spatial linkages, termed constrained parallel manipulators. These articulated systems are formed by supporting a workpiece, or end-effector, with a set of serial chains, each of which imposes a constraint on the end-effector. Our goal is to plan trajectories for systems that have workspaces ranging from two through five degrees-of-freedom. This is done by specifying a goal trajectory and finding the system trajectory that comes closest to it using a dual quaternion metric. We enumerate these parallel mechanisms and formulate a general numerical approach for their analysis and trajectory planning. Examples are provided to illustrate the results. [DOI: 10.1115/1.1623187]

1 Introduction

In this paper we formulate a trajectory planning algorithm for parallel manipulators that have less than six degrees-of-freedom. For our purposes, we assume that each supporting serial chain of the system imposes a constraint on the movement of the workpiece, or end-effector. Thus, no supporting chain has six-degrees-of-freedom. The constraints imposed by each chain can be designed to provide structural resistance to forces in one or more directions, while allowing the system to move in other directions.

There are six basic joints used in the construction of these supporting chains, and we enumerate their various combinations. This allows us to count the large number of assemblies available for constrained parallel manipulators.

We then formulate a general algorithm for the analysis of these systems which uses the Jacobians the supporting chains. Of importance is the ability to compute a trajectory for the end-effector of the system that approximates a specified trajectory while maintaining its kinematic constraints. The algorithm has been integrated into SYNTHETICA [1], a Java based kinematic synthesis software. And examples are provided to illustrate the results.

2 Literature Review

This research arises in the context of efforts to develop a software system for the kinematic synthesis of spatial linkages [1]. Kinematic synthesis theory yields designs for serial chains that guide a workpiece through a finite set of positions and orientations, see McCarthy [2]. These chains necessarily have less than six degrees-of-freedom, and are often termed "constrained robotic systems." Also see [2–5].

The design process yields multiple serial chains that can reach the prescribed goal positions and provides the opportunity to assemble systems with parallel architecture. Analysis and simulation allows interactive evaluation these candidate designs. This framework for linkage design was introduced by Rubel and Kaufman [6], Erdman and Gustafson [7] and Waldron and Song [8], and later followed by Ruth and McCarthy [9] and Larochelle [10].

Our focus is the challenge of animating the broad range of linkage systems that are not constrained to one degree-of-freedom, but do not have full six degrees-of-freedom of the usual parallel manipulator. Joshi and Tsai [11] call these systems "limited DOF parallel manipulators." Examples are the recent study of the 3-RPS system by Huang et al. [12], the 3-PSP by Gregorio and Parenti-Castelli [13], and the double tripod by Hertz and Hughes [14].

Our work has a goal similar to that of Merlet's [15] "Trajectory Verifier" in that we define a trajectory and determine whether the system can reach it. However, we also determine the closest approaching system movement. This closest approaching trajectory is also described in Fluckiger's [16] CINEGEN, however, we focus on constrained systems with parallel structure. Our approach is to solve for the constrained parallel manipulator configuration that comes closest at each frame in a specified trajectory. The key-frame interpolation scheme we use is presented in [17], and based on double quaternion formulation of Etzel and McCarthy [18] and Ge et al. [19]. See also [20].

3 Kinematics of Constrained Robots

The kinematic analysis of a constrained robot begins with the kinematics equations of its supporting serial chains. Each chain can be modelled using 4×4 homogeneous transformations and the Denavit-Hartenberg convention [21] to obtain the kinematic equation

$$[K(\vec{\theta})]=[Z(\theta_1,d_1)][X(\alpha_{12},a_{12})] \times[Z(\theta_2,d_2)]\dots[X(\alpha_{n-1,n},a_{n-1,n})][Z(\theta_n,d_n)], \quad (1)$$

where $2\leqslant n\leqslant 5$. $[Z(\cdot,\cdot)]$ and $[X(\cdot,\cdot)]$ denote screw displacements about the z and x-axes, respectively. The parameters (θ,d) define the movement at each joint and (α,a) are the twist angle and length of each link, collectively known as the Denavit-Hartenberg parameters.

Notice that a serial chain robot is usually defined in terms of revolute (R) and prismatic (P) joints which have the kinematics equations,

$$\text{revolute: } [R(\theta)]=[Z(\theta,-)], \quad \text{prismatic: } [P(d)]=[Z(-,d)]. \quad (2)$$

The hyphen denotes parameters that are constant. For our purposes, we include four additional joints that are special assemblies of R and P joints, Table 1. They are:

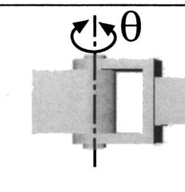
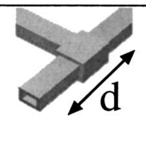
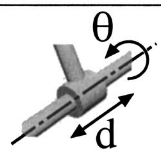
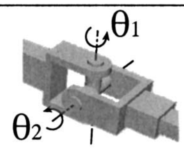
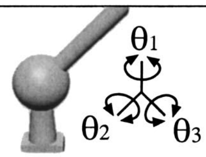
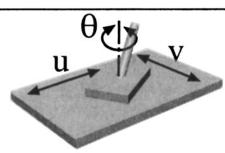
i. the cylindric joint, denoted by C, which is a PR chain with parallel axes, such that

$$[C(\theta,d)]=[Z(\theta,d)]; \quad (3)$$

ii. the universal joint, denoted by T, which consists of two revolute joints with axes that intersect in a right angle, that is

Contributed by the Mechanisms and Robotics Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received Aug. 2002; rev. April 2003. Associate Editor: M. Raghavan.

Table 1 The six basic joints

Joint	Diagram	Symbol	DOF
Revolute		R	1
Prismatic		P	1
Cylindric		C	2
Universal		T	2
Spherical		S	3
Planar		E	3

$$[T(\theta_1,\theta_2)]=[Z(\theta_1,-)]\left[X\left(0,\frac{\pi}{2}\right)\right][Z(\theta_2,-)]; \qquad (4)$$

iii. the spherical joint, denoted by S, which is constructed from 3R chain with axes that are intersect in a single point, given by

$$[S(\theta_1,\theta_2,\theta_3)]=[Z(\theta_1,-)]\left[X\left(0,\frac{\pi}{2}\right)\right][Z(\theta_2,0)]\left[X\left(0,-\frac{\pi}{2}\right)\right] \times[Z(\theta_3,-)]; \qquad (5)$$

iv. the planar joint, denoted by E, has several constructions but is equivalent to a $2PR$ chain constructed so the axis of the revolute joint is perpendicular to the plane defined by the two prismatic joints, given by

$$[E(u,v,\theta)]=[Z(-,u)]\left[X\left(0,\frac{\pi}{2}\right)\right][Z(0,v)]\left[X\left(0,\frac{\pi}{2}\right)\right][Z(\theta,-)]. \qquad (6)$$

The transformation along each link is $[X(\alpha_{j,j+1},a_{j,j+1})]$, where $(\alpha_{j,j+1},a_{j,j+1})$ define the dimensions of the chain.

If $[K(\vec{\theta}_i)]$ represents the kinematics equation of the ith supporting chain of a constrained robotic system, then the kinematics equations for a system with k supporting chains are given by

$$[D(\vec{\theta}_i)]=[G_i][K(\vec{\theta}_i)][H_i], i=1,\cdots,k, \qquad (7)$$

where $[G_i]$ locates the base of the ith chain in the fixed frame $\mathcal{F}$, and $[H_i]$ locates the workpiece $\mathcal{M}$ relative to the last link frame of chain i.

Each of the kinematics equations of a constrained robot imposes a constraint on the movement of the workpiece. Our goal is to provide a numerical solution to these equations (7) in order to simulate this movement.

4 Enumeration of Constrained Parallel Manipulators

Using the basic joints R, P, C, T, S and E, it is possible enumerate all serial chains that constrain an end-effector, and therefore all of the constrained robots that our kinematics equations define. Tsai [22] presents enumeration theory for the design of mechanical systems. Also see [23].

Recall that the freedom F of the end-effector of a serial chain is the sum of the degree-of-freedom f_j at each joint. We define degree-of-constraint $U=6-F$ as the number of constraints on the end-effector imposed by the serial chain, that is a serial chain with n joints has

$$U=6-\sum_{j=1}^{n} f_j. \qquad (8)$$

If a single R or P joint supports the end-effector, then $U=5$, and the system cannot have another supporting serial chain because an additional constraint eliminates movement of the end-effector. These two cases are considered to be trivial constrained robots. On the other hand, a serial chain with six or more degrees-of-freedom has $U\leqslant 0$, and does not impose any constraint on the end effector. Our interest is in serial chains with degree-of-constraint $1\leqslant U\leqslant 4$, that is those that have freedom $2\leqslant F\leqslant 5$.

4.1 Supporting Serial Chains. We first categorize the supporting serial chains by their degree-of-constraint U. Within each category, we define the class of serial chains based only on the presence of various joints, independent of the order of the joints along the chain. Note that a serial chain can have no more than three prismatic joints. Furthermore, the C and E joints contain the equivalent of one and two prismatic joints respectively, which must be accommodated in the enumeration.

After determining all the classes of chains, we permute the joints to obtain various chains. The number of permutations for each class can be evaluated using the formula

$$\frac{n!}{n_R!n_P!n_C!n_T!n_S!n_E!}, \qquad (9)$$

where n is the number joints in the chain, and n_R denotes the number of revolute joints, and so on for the other joints, as well. The result is that we have six serial chains in category IV, 18 in category III, 51 in category II, and 139 in category I; see Table 2.

4.2 Parallel Assemblies. Consider a constrained parallel manipulator with k chains, each of which imposes U_i constraints, then the mobility M of the end-effector is

$$M=6-\sum_{i=1}^{k} U_i. \qquad (10)$$

Notice that the degree-of-constraint $C=6-M$ of the end-effector can be computed as the sum of the constraints imposed by the individual chains.

Figures 1 and 2 show examples of the 2TPR and 3RPS constrained parallel manipulators. Each of the TPR chain imposes two constraints therefore the parallel system 2TPR has two degrees-of-freedom. Similarly, the individual RPS chains impose one constraint, which means the system 3RPS has three degrees-of-freedom.

Table 2 Serial chains with four, three, two and one degrees-of-constraint

Category	Class	Chains
IV	2R	RR
(6)	RP	RP, PR
	2P	PP
	C	C
	T	T
III	3R	RRR
(18)	2RP	RRP, RPR, PRP
	R2P	RPP, PRP, PPR
	RC	RC, CR
	RT	RT, TR
	3P	PPP
	PC	PC, CP
	PT	PT, TP
	S	S
	E	E
II	4R	RRRR
(51)	3RP	RRRP, RRPR, RPRR, PRRR
	2R2P	RRPP, RPPR, PPRR, PRRP, PRPR, RPRP
	2RC	RRC, RCR, CRR
	2RT	RRT, RTR, TRR
	R3P	RPPP, PRPP, PPRP, PPPR
	RPC	RPC, RCP, PRC, PCR, CRP, CPR
	RPT	RPT, RTP, PRT, PTR, TRP, TPR
	RS	RS, SR
	RE	RE, ER
	2PC	PPC, PCP, CPP
	2PT	PPT, PTP, TPP
	PS	PS, SP
	PE	PE, EP
	2C	CC
	CT	CT, TC
	2T	TT
I	4RP	RRRRP, RRRPR, RRPRR, RPRRR, PRRRR
(139)	3R2P	RRRPP, RRPPR, RPPRR, PPRRR, PRPRR
		PRRPR, PRRRP, RPRPR, RPRRP, RRPRP
	3RC	RRRC, RRCR, RCRR, CRRR
	3RT	RRRT, RRTR, RTRR, TRRR
	2R3P	PPPRR, PPRRP, PRRPP, RRPPP, RPRPP
		RPPRP, RPPPR, PRPRP, PRPPR, PPRPR
	2RPC	RRPC, RRCP, RPRC, RPCR, RCRP, RCPR
		PRRC, PRCR, PCRR, CRRP, CRPR, CPRR
	2RPT	RRPT, RRTP, RPRT, RPTR, RTRP, RTPR
		PRRT, PRTR, PTRR, TRRP, TRPR, TPRR
	2RS	RRS, RSR, SRR
	2RE	RRE, RER, ERR
	R2PC	PPRC, PPCR, PRPC, PRCP, PCPR, PCRP
		RPPC, RPCP, RCPP, CPPR, CPRP, CRPP
	R2PT	PPRT, PPTR, PRPT, PRTP, PTPR, PTRP
		RPPT, RPTP, RTPP, TPPR, TPRP, TRPP
	RPS	RPS, RSP, PRS, PSR, SRP, SPR
	RPE	RPE, REP, PRE, PER, ERP, EPR
	R2C	RCC, CRC, CCR
	RCT	RCT, RTC, CRT, CTR, TRC, TCR
	R2T	RTT, TRT, TTR
	3PT	PPPT, PPTP, PTPP, TPPP
	2PS	PPS, PSP, SPP
	P2C	PCC, CPC, CCP
	PCT	PCT, PTC, CPT, CTP, TPC, TCP
	P2T	PTT, TPT, TTP
	CS	CS, SC
	CE	CE, EC
	TS	TS, ST
	TE	TE, ET

We can list the parallel assemblies that impose a particular degree-of-constraint by combining chains from the various categories. Lee and Tsai [24] enumerate parallel structures for mechanical hands, but with identical supporting chains. For example, 4I denotes four supporting chains that impose one degree-of-constraint each. Similarly 2I-1II is a parallel assembly consisting of two chains that impose one constraint each, combined with a third chain that imposes two constraints. Both systems provide the workpiece with two degrees-of-freedom of movement. Table 3 provides an exhaustive list of the various assembly categories.

The number of combinations of chains within each assembly category is easily determined. Consider the assembly category 4I consisting of four serial chains each of which imposes one constraint. Recall that the number of serial chains in category I is 139. Therefore, the number of assemblies is the combination of $k=4$ chains taken from the list of $n=139$ with repetitions allowed, which is given by the formula

$$C_k^{n+k-1}=C_4^{139+4-1}=\frac{(139+4-1)!}{4!(139-1)!}=16{,}234{,}505. \quad (11)$$

Applying this formula to each of the various assembly categories we can compute the total number of 1–5 degree-of-freedom constrained robotic systems. See Table 3.

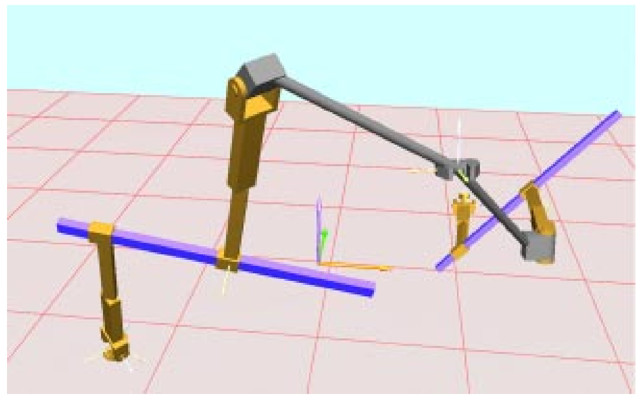


Fig. 1 A TPR chain imposes two constraints, so a 2TPR spatial linkage has two degrees-of-freedom

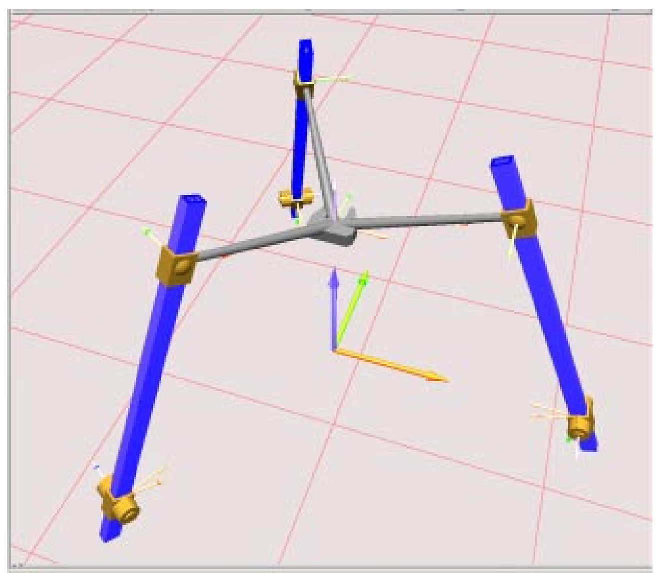


Fig. 2 An RPS chain imposes one constraint, so a 3RPS spatial linkage has three degrees-of-freedom

Table 3 Enumeration of constrained parallel manipulators

Constraints	Assembly Categories	Total
5	5I, 3I-1II, 2I-1III, 1I-2II, 1I-1IV	487,990,859
4	4I, 2I-1II, 2II, 1I-1III, 1IV	16,734,569
3	3I, 1I-1II, 1III	464,417
2	2I, 1II	9,781
1	1I	139

5 Differential Kinematics

The trajectory of a point $\mathbf{p}$ in the workpiece of a constrained robot supported by k serial chains can be computed using the kinematics equations of the ith supporting chain as

$$\mathbf{P}(t)=[G_i][K(\vec{\theta}_i)][H_i]\mathbf{p} \tag{12}$$

The velocity $\dot{\mathbf{P}}$ of this point is given by

$$\dot{\mathbf{P}}=\mathbf{v}+\vec{\omega}\times(\mathbf{P}-\mathbf{d}), \tag{13}$$

where $\mathbf{d}$ is the origin of the end-effector frame $\mathbf{M}$, $\mathbf{v}$ is its velocity, and $\vec{\omega}$ is the angular velocity of this frame. The six-vector $\mathbf{V}=(\mathbf{v},\vec{\omega})^T$ is related to the joint rates of each supporting chain by the equation

$$\mathbf{V}=[J_1]\dot{\vec{\theta}}_1=\cdots=[J_i]\dot{\vec{\theta}}_i=\cdots=[J_k]\dot{\vec{\theta}}_k, \tag{14}$$

where $[J_i]$ is known as the Jacobian of the ith chain. See [25] for a similar approach to formulating the differential closure equations by cutting a closed spatial chain at various links.

It is possible to show that the Jacobian for ith serial chain takes form

$$[J_i]=\begin{bmatrix}\mathbf{S}_1\times(\mathbf{d}-\mathbf{C}_1) & \cdots & \mathbf{S}_j\times(\mathbf{d}-\mathbf{C}_j) & \cdots & \mathbf{S}_{n_i}\times(\mathbf{d}-\mathbf{C}_{n_i})\\ \mathbf{S}_1 & \cdots & \mathbf{S}_j & \cdots & \mathbf{S}_{n_i}\end{bmatrix}, \tag{15}$$

where n_i is the number of joint variables in the ith supporting chain, $\mathbf{S}_j$ is the direction of the jth revolute joint axis and $\mathbf{C}_j$ is a point on this axis, measured in the base frame Φ. Furthermore, if the jth revolute joint is replaced by a prismatic joint, then the column vector $(\mathbf{S}_j\times(\mathbf{d}-\mathbf{C}_j),\mathbf{S}_j)^T$ is replaced by $(\mathbf{S}_j,\mathbf{O})^T$. See [20,2].

The joints C, T, S, and E are assemblies of revolute and prismatic joints, therefore the Jacobians for serial chains with these joints have the same structure as Eq. (15). In our system, we have integrated these higher order joints to facilitate simulation of more complex system.

We now introduce dual quaternion parameters formulated as an eight dimensional vector $\mathbf{q}=(q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8)$ to represent the position of the end-effector [26,27]. These parameters provide a convenient formulation for key-frame interpolation, as well as for measurement of position error [17,18]. Let $[D(\mathbf{q})]=[R(\mathbf{q}),\mathbf{d}(\mathbf{q})]$ be the 4×4 homogeneous transform which locates the end-effector, written in terms of the components of the dual quaternion $\mathbf{q}$, such that the rotation matrix $[R(\mathbf{q})]$ and translation vector $\mathbf{d}(\mathbf{q})$ are given by,

$$[R(\mathbf{q})]=\begin{bmatrix} q_1^2-q_2^2-q_3^2+q_4^2 & 2(q_1q_2-q_3q_4) & 2(q_1q_3+q_2q_4)\\ 2(q_1q_2+q_3q_4) & -q_1^2+q_2^2-q_3^2+q_4^2 & 2(q_2q_3-q_1q_4)\\ 2(q_1q_3-q_2q_4) & 2(q_2q_3+q_1q_4) & -q_1^2-q_2^2+q_3^2+q_4^2\end{bmatrix} \tag{16}$$

and

$$\mathbf{d}(\mathbf{q})=\begin{Bmatrix}2(-q_1q_8+q_2q_7-q_3q_6+q_4q_5)\\ 2(-q_1q_7-q_2q_8+q_3q_5+q_4q_6)\\ 2(q_1q_6-q_2q_5-q_3q_8+q_4q_7)\end{Bmatrix}. \tag{17}$$

The velocity $\mathbf{V}$ of the end-effector is given by

$$\mathbf{V}=[J_E]\dot{\mathbf{q}}, \tag{18}$$

where $[J_E]$ is the 6×8 matrix

$$[J_E]=\begin{bmatrix} -2q_8 & 2q_7 & -2q_6 & 2q_5 & 2q_4 & -2q_3 & 2q_2 & -2q_1\\ -2q_7 & -2q_8 & 2q_5 & 2q_6 & 2q_3 & 2q_4 & -2q_1 & -2q_2\\ 2q_6 & -2q_5 & -2q_8 & 2q_7 & -2q_2 & 2q_1 & 2q_4 & -2q_3\\ 2q_4 & -2q_3 & 2q_2 & -2q_1 & 0 & 0 & 0 & 0\\ 2q_3 & 2q_4 & -2q_1 & -2q_2 & 0 & 0 & 0 & 0\\ -2q_2 & 2q_1 & 2q_4 & -2q_3 & 0 & 0 & 0 & 0\end{bmatrix} \tag{19}$$

These equations are augmented by the derivatives of the two dual quaternion constraint equations $q_1^2+q_2^2+q_3^2+q_4^2=1$ and $q_1q_5+q_2q_6+q_3q_7+q_4q_8=0$, which becomes

$$\begin{Bmatrix}0\\0\end{Bmatrix}=\begin{Bmatrix}2q_1 & 2q_2 & 2q_3 & 2q_4 & 0 & 0 & 0 & 0\\ q_5 & q_6 & q_7 & q_8 & q_1 & q_2 & q_3 & q_4\end{Bmatrix}\dot{\mathbf{q}}=[C]\dot{\mathbf{q}}. \tag{20}$$

We now use the Eq. (14) and Eq. (18) together with the differential quaternion constraints Eq. (20) to define the differential kinematics equations for the constrained robot. Approximate the derivatives in these equations to obtain a linearized set of closure equations

$$\begin{bmatrix}[J_1] & \mathbf{0} & \cdots & \mathbf{0} & -[J_E]\\ & [J_2] & \cdots & \mathbf{0} & -[J_E]\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \mathbf{0} & \mathbf{0} & \cdots & [J_k] & -[J_E]\\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & [C]\end{bmatrix}\begin{Bmatrix}\Delta\vec{\theta}_1\\ \Delta\vec{\theta}_2\\ \vdots\\ \Delta\vec{\theta}_k\\ \Delta\mathbf{q}\end{Bmatrix}=\begin{Bmatrix}\vec{E}_1\\ \vec{E}_2\\ \vdots\\ \vec{E}_k\\ \vec{E}_c\end{Bmatrix}, \tag{21}$$

which we write as

$$[A]\Delta\mathbf{r}=\mathbf{E}. \tag{22}$$

The vectors $\vec{E}_i$, $i=1,\cdots k$, define the difference between the position of the end-effector defined by $\mathbf{q}$ and its position defined by joint parameters $\vec{\theta}_i$ of the ith chain. We compute $\vec{E}_i$ from the difference of the 4×4 matrices $[D(\vec{\theta}_i)]=[R(\vec{\theta}_i),\mathbf{d}(\vec{\theta}_i)]$ and $[D(\mathbf{q})]=[R(\mathbf{q}),\mathbf{d}(\mathbf{q})]$. In particular, we have

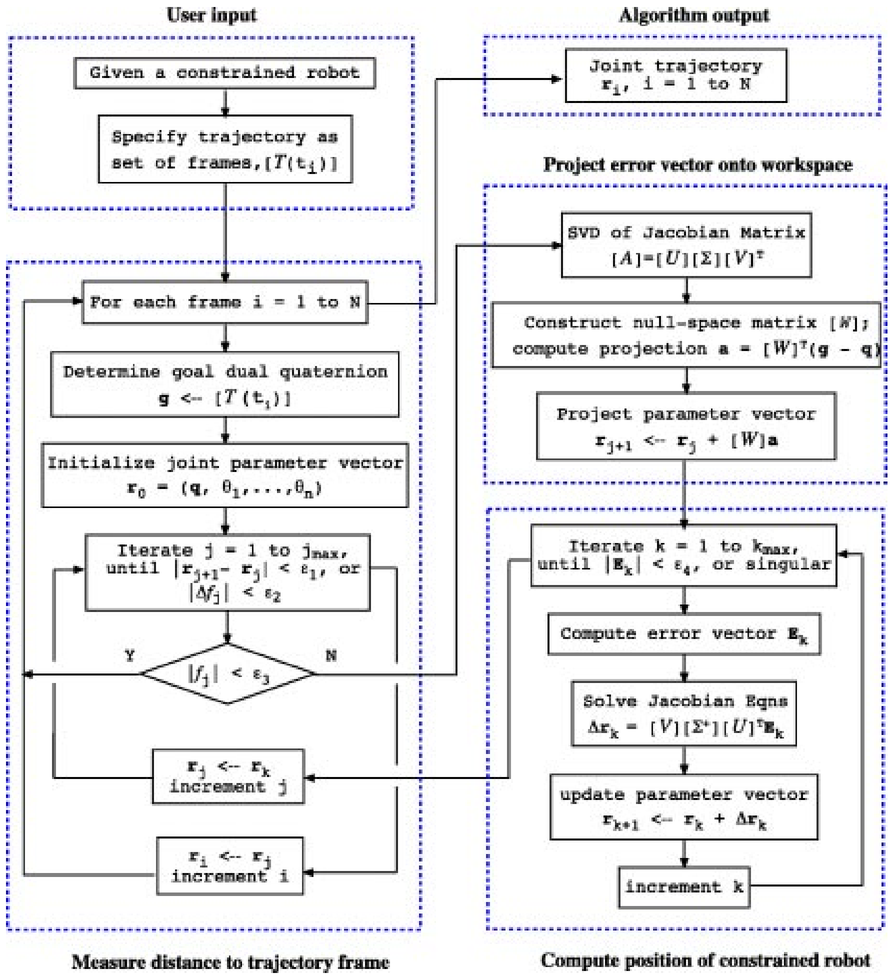


Fig. 3 Flowchart of the trajectory planning algorithm

$$\vec{E}_i=\begin{Bmatrix}\mathbf{d}(\vec{\theta}_i)-\mathbf{d}(\mathbf{q})\\ \Delta\vec{\phi}_i\end{Bmatrix}, \tag{23}$$

where $\Delta\vec{\phi}_i$ is the 3×1 vector constructed from the skew-symmetric part of the matrix

$$[\Delta\Phi_i]=[R(\vec{\theta}_i)-R(\mathbf{q})][R(\mathbf{q})]^T. \tag{24}$$

The term $\vec{E}_c$ is the difference between the dual quaternion constraint equations and their required values, given by

$$\vec{E}_c=\begin{Bmatrix}q_1^2+q_2^2+q_3^2+q_4^2-1\\ q_1q_5+q_2q_6+q_3q_7+q_4q_8\end{Bmatrix}. \tag{25}$$

The coefficient matrix $[A]$ in Eq. (22) consists of $m=6k+2$ rows and $n=8+\Sigma\, n_i$ columns, where n_i is the number of joint variables in the ith supporting chain as stated before. The difference between these two is the mobility M of the system, that is

$$M=n-m=\left(8+\sum_{i=1}^{k} n_i\right)-(6k+2)=6-\sum_{i=1}^{k}(6-n_i). \tag{26}$$

The definition of a constrained robot requires that $1\leqslant M\leqslant5$.

6 Inverse Kinematics

The linearized closure Eq. (21) can be used to find the joint parameters for a constrained robot that positions its end-effector at or near a specified goal position, depending on whether this position is inside or outside the workspace of the system. This gives us the flexibility to specify a trajectory for the end-effector of the constrained robot that may pass outside its workspace.

We follow Dietmaier and Pavlin [28] and use the singular value decomposition in the solution of these equations. Also see [16]. The singular value decomposition of $[A]$ is given by

$$[A]=[U][\Sigma][V]^T, \tag{27}$$

where $[\Sigma]$ is an $m\times n$ matrix with the square roots of the eigenvalues of $[AA^T]$, denoted σ_i, along its main diagonal and $n-m$ columns of zeros. The associated eigenvectors form the $m\times m$ orthogonal matrix $[U]$. Let $r\leqslant m$ be the rank of $[A]$, then the first r columns of the $n\times n$ matrix $[V]$ are obtained by normalizing the vectors $[A]^Tu_i$, where u_i is the ith column of $[U]$. The remaining $n-r$ columns are constructed by Gram-Schmidt orthogonalization to span R^n and define the null-space of $[A]$.

For a given location $\mathbf{q}$ of the end-effector and an estimate of the joint parameters, denoted $\mathbf{r}_i$, we can solve the closure equations to determine an updated parameter vector, $\mathbf{r}_{i+1}$. This is done by computing the pseudoinverse of $[A]^+$ using the singular value decomposition

$$\mathbf{r}_{i+1}=\mathbf{r}_i+[V][\Sigma]^+[U]^T\mathbf{E}, \tag{28}$$

where $[\Sigma]^+$ is the $n\times m$ matrix with $1/\sigma_i$ along its main diagonal and $n-r$ rows of zeros. This provides the minimum norm solution vector $\mathbf{r}_{i+1}$, which we iterate until the error vector $\mathbf{E}$ becomes less than a prescribed tolerance.

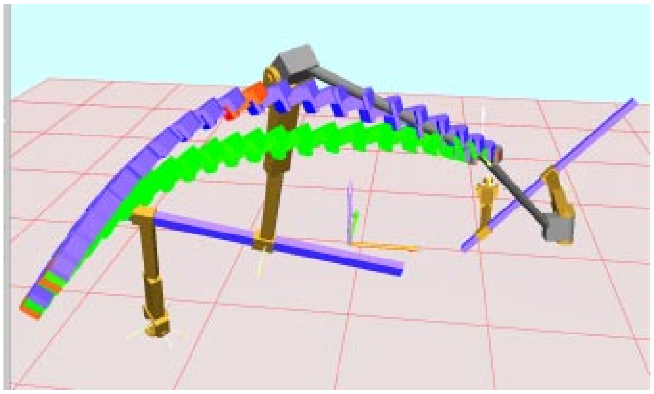


Fig. 4 Example of the trajectory of a 2TPR linkage that approximates a specified trajectory

Table 4 The Denavit-Hartenberg parameters for the TPR chain

Joint	α	a	θ	d
1	⋯	⋯	θ_1	0
2	−90°	0	θ_2	0
3	45°	1	0	d_3
4	45°	1	θ_4	0

Table 5 The base and gripper frames of a 2TPR system

	x	y	z	Longitude	Latitude	Roll
G_1	−1.000	−1.000	0.000	30.00°	0.00°	0.00°
H_1	−0.103	0.550	0.666	69.58°	−57.08°	−55.21°
G_2	1.000	1.000	0.000	0.00°	45.00°	0.00°
H_2	0.500	0.500	0.500	0.00°	0.00°	0.00°

7 Trajectory Planning

In order to plan the movement of a constrained robotic system, the user specifies a trajectory $[T(t_i)]$ consisting of a set of positions obtained by key frame interpolation, where $i=1,\cdots N$ denotes the interpolation frame count. Our goal is to have the end-effector of the constrained robot follow this trajectory as closely as possible.

For each frame $[T(t_i)]$, let $\mathbf{g}_i$ be the associated dual quaternion. We seek the end-effector position $\mathbf{q}_i$ that minimizes $|\mathbf{g}_i-\mathbf{q}_i|$, which is the same as minimizing the objective function,

$$f(\mathbf{q}_i)=(\mathbf{q}_i-\mathbf{g}_i)\cdot(\mathbf{q}_i-\mathbf{g}_i)/2, \quad (29)$$

subject to constraints imposed by the kinematics equations (7).

The gradient of the objective function $f(\mathbf{q}_i)$ yields

$$\nabla f(\mathbf{q}_i)=(\mathbf{q}_i-\mathbf{g}_i), \quad (30)$$

which we project onto the null space of the constraint equations defined by $[A]$ in Eq. (21). The negative of the gradient ∇f is the direction in which the objective function $f(\mathbf{q}_i)$ decreases.

The null space of $[A]$ defines the feasible directions for changes to the parameter vector $\mathbf{r}$. This null space is obtained from the singular value decomposition Eq. (27) as the last $n-r$ vectors of the matrix $[V]$. Denote these orthonormal vectors as $\vec{w}_j$, $j=1,\cdots,n-r$, so we have the $n\times(n-r)$ matrix $[W]$. The projection of $-\nabla f$ onto this null space is

$$\mathbf{a}=[W]^T(\mathbf{g}_i-\mathbf{q}_i), \quad (31)$$

where $\mathbf{g}_i$ and $\mathbf{q}_i$ are augmented by $n-8$ zeros. The result is the incremental parameter vector $\Delta\mathbf{r}=[W]\mathbf{a}$. We then use the numerical inverse kinematics Eq. (28) to update the parameter vector $\mathbf{r}$ so that it satisfies the constraint equations.

This procedure is iterated until the objective function reaches a local minimum. See Fig. 3.

8 Numerical Examples

8.1 2TPR System. To demonstrate this algorithm consider 2TPR constrained parallel manipulator shown in Fig. 4. Table 4 defines the Denavit-Hartenberg parameters for the two TPR chains. The locations of the base and gripper frames for these two chains are specified by the translation vector and three angles given in Table 5.

The darker trajectory is specified by Bezier interpolation of key-frames at the beginning and the end and one in the middle. Notice that initial and final positions are reachable by the 2TPR manipulator, but that the remainder of the trajectory is outside the workspace of this system. The lighter color trajectory denotes the approximation to this path as determined by the planning algorithm.

8.1 3RPS System. Another example of a constrained parallel manipulator is defined by the 3RPS system shown in Fig. 5. The Denavit-Hartenberg parameters of the three supporting serial chains are given in Table 6. The locations of the base and gripper frames for these three chains are specified by the translation vector and three angles given in Table 7.

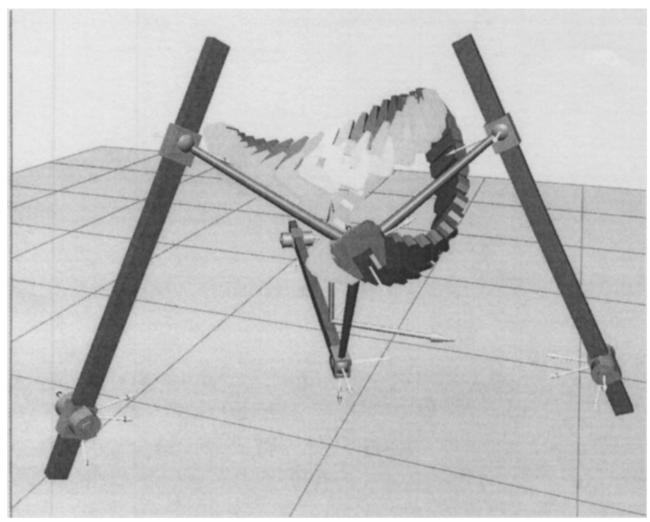


Fig. 5 Example of the trajectory of a 3RPS linkage that approximates a specified movement

Table 6 The Denavit-Hartenberg parameters for the RPS chain

Joint	α	a	θ	d
1	⋯	⋯	θ_1	0
2	90°	0	0	d_2
3	0	0	θ_3	0
4	90°	0	θ_4	0
5	−90°	0	θ_5	0

Table 7 The base and gripper frames of a 3RPS system

	x	y	z	Longitude	Latitude	Roll
G_1	−0.500	−0.866	0.000	−90.00	30.00	90.00
H_1	0.571	0.000	0.223	−41.00	0.00	−60.00
G_2	1.003	0.000	0.000	−180.00	−90.00	0.00
H_2	0.571	0.000	0.223	−41.00	0.00	180.00
G_3	−0.500	0.866	0.000	90.00	30.00	−90.00
H_3	0.571	0.000	0.223	−41.00	0.00	0.00

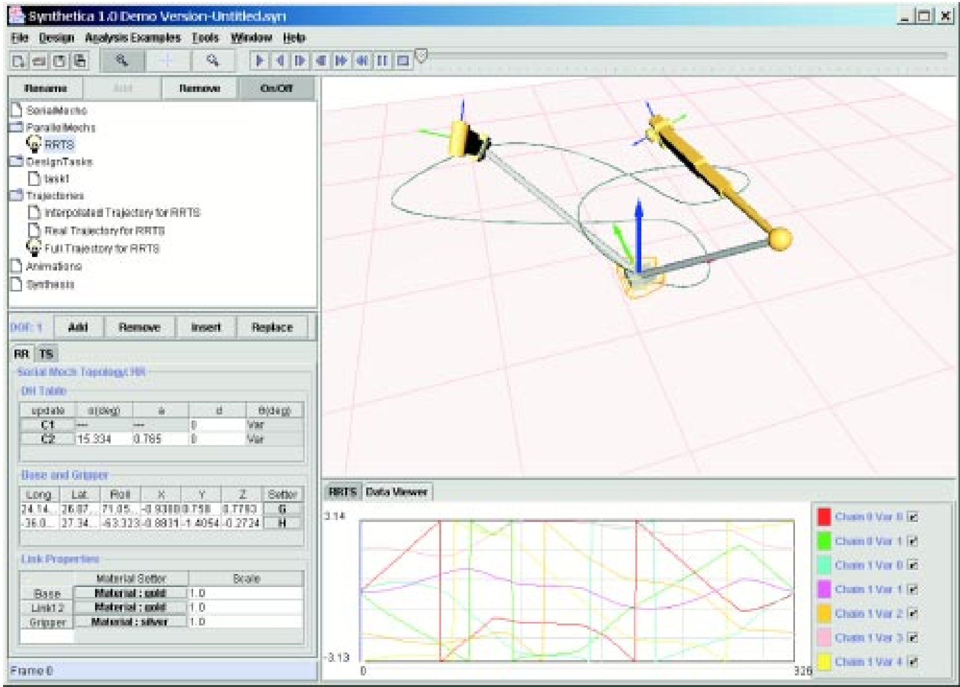


Fig. 6 The user interface for the SYNTHETICA software system

As in the previous example the darker trajectory is specified by Bezier interpolation of a set of key-frames. The lighter trajectory is the approximation that is reachable by the 3RPS manipulator.

This algorithm is integrated into SYNTHETICA, a Java-based kinematic synthesis software under development at UCI, Fig. 6. The software is available online at http://synthetica.eng.uci.edu/~mccarthy/. These examples were run on a 1.5GHz Pentium 4 and required less than 10 sec to compute a 33 frame trajectory and display the results.

9 Conclusions

This paper presents an algorithm for the analysis of constrained parallel manipulators for use in kinematic synthesis software. An elementary enumeration shows that there is a wide range of these devices, and our formulation is tailored to accommodate this variety. An important feature is the ability to specify a trajectory for the end-effector without considering the constrained workspace of the system. The algorithm finds a sequence of end-effector positions and joint parameter values that satisfy the kinematic constraints and approximate the specified trajectory. This result highlights the unique challenges of robotic systems with less than six degrees-of-freedom.

10 Conclusions

The authors gratefully acknowledge contributions Curtis Collins and Alba Perez to the SYNTHETICA project and the support of the National Science Foundation.

References

[1] Collins, C. L., McCarthy, J. M., Perez, A., and Su, H.-J., 2002, ''The Structure of an Extensible Java Applet for Spatial Linkage Synthesis,'' ASME J. Comput. Inf. Sci. Eng., **2**(1), pp. 45–49.

[2] McCarthy, J. M., 2000, *Geometric Design of Linkages*, Springer-Verlag, New York, NY.

[3] McCarthy, J. M., 2000, ''Mechanisms Synthesis Theory and the Design of Robots,'' *Proc. Int. Conf. Robotics and Automation*, San Francisco, CA, April 24–28.

[4] Lee, E., and Mavroidis, C., 2002, ''Solving the Geometric Design Problem of Spatial 3R Robot Manipulators Using Polynomial Continuation,'' ASME J. Mech. Des., **124**(4), pp. 652–661.

[5] Perez, A., and McCarthy, J. M., 2002, ''Dual Quaternion Synthesis of a 2-TPR Constrained Parallel Robot,'' *Proc. of the WORKSHOP on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Clement M. Gosselin and Imme Ebert-Uphoff eds., Quebec City, Quebec, Canada, Oct. 3–4.

[6] Rubel, A. J., and Kaufman, R., 1977, ''KINSYN III: A New Human-Engineered System for Interactive Computer-Aided Design of Planar Linkages,'' ASME J. Eng. Ind., **99B**(2), pp. 440–448.

[7] Erdman, A., and Gustafson, J., 1977, ''LINCAGES: Linkage INteractive Computer Analysis and Graphically Enhanced Synthesis Packages,'' *Technical Report 77-DET-5, American Society of Mechanical Engineers*.

[8] Waldron, K. J., and Song, S. M., 1981, ''Theoretical and Numerical Improvements to Interactively Linkage Design Program, RECSYN,'' *Proc. of the Seventh Applied Mechanisms Conference*, Kansas City, MO.

[9] Ruth, D. A., and McCarthy, J. M., 1997, ''SphinxPC: An Implementation of Four Position Synthesis for Planar and Spherical 4R Linkages,'' *CD-ROM Proc. of the ASME DETC'97*, paper no. DETC97/DAC-3860, Sept. 14–17, Sacramento, CA.

[10] Larochelle, P. M., 1998, ''Spades: Software for Synthesizing Spatial 4C Linkages,'' *CD-ROM Proc. of the ASME DETC'98*, paper no. DETC98/Mech-5889, Sept. 13–16, Atlanta, GA.

[11] Joshi, S. A., and Tsai, L. W., 2002, ''Jacobian Analysis of Limited-DOF Parallel Manipulators,'' ASME J. Mech. Des., **124**(2), pp. 254–258.

[12] Huang, Z., Wang, J., and Fang, Y. F., 2002, ''Analysis of Instantaneous Motions of Deficient-Rank 3-RPS Parallel Manipulators,'' Mech. Mach. Theory, **37**(2), pp. 229–240.

[13] Gregorio, R. D., and Parenti-Castelli, V., 2001, ''Position Analysis in Analytical Form of the 3-PSP Mechanism,'' ASME J. Mech. Des., **123**(1), pp. 51–57.

[14] Hertz, R. B., and Hughes, P. C., 1998, ''Kinematic Analysis of a General Double-Tripod Parallel Manipulator,'' Mech. Mach. Theory, **33**(6), pp. 683–696.

[15] Merlet, J. P., 2001, ''A Generic Trajectory Verifier for the Motion Planning of Parallel Robots,'' ASME J. Mech. Des., **123**(4), pp. 510–515.

[16] Fluckiger, L., Baur, C., and Clavel, Reymond, 1998, ''CINEGEN: A Rapid Prototyping Tool for Robot Manipulators,'' *Proc. of the 4th International Conference on Motion and Vibration Control*(MOVIC'98), Vol. 1, pp. 129–134, Zurich(CH), August.

[17] Ahlers, S. G., and McCarthy, J. M., 2000, ''The Clifford Algebra of Double Quaternions and the Optimization of TS Robot Design,'' *Applications of Clifford Algebras in Computer Science and Engineering*, E. Bayro and G. Sobczyk, eds., Birkhauser.

[18] Etzel, K., and McCarthy, J. M., 1996, ''Spatial Motion Interpolation in an Image Space of SO(4),'' *CD-ROM Proc. 1996 ASME Design Engineering Technical Conference*, paper no. 96-DETC/MECH-1164, August 18–22, Irvine, CA.

[19] Ge, Q. J., Varshney, A., Menon, J. P., and Chang, C.-F., 1998, ''Double Quaternions for Motion Interpolation,'' *CD-ROM Proc. of the ASME DETC'98*, paper no. DETC98/DFM-5755, Sept. 13–16, Atlanta, GA.

[20] Srinivasan, L., and Ge, Q. J., 1998, ''Fine Tuning of Rational B-Spline Motions,'' ASME J. Mech. Des., **120**(1), pp. 46–51.

[21] Craig, J. J., 1989, *Introduction to Robotics: Mechanics and Control. Second Edition*, Addison-Wesley.
[22] Tsai, L. W., 2001, *Mechanism Design, Enumeration of Kinematic Structures According to Function*, CRC Press, Boca Raton.
[23] Tuttle, E. R., Peterson, S. W., and Titus, J. E., 1989, ''Enumeration of Basic Kinematic Chains Using the Theory of Finite Groups,'' ASME J. Mech., Transm., Autom. Des., **111**, pp. 498–503.
[24] Lee, J. J., and Tsai, L. W., 2002, ''Structural Synthesis of Multi-Fingered Hands,'' ASME J. Mech. Des., **124**(2), pp. 272–276.
[25] Zou, H. L., Abdel-Malek, K. A., and Wang, J. Y., 1997, ''Design Propagation in Mechanical Systems: Kinematic Analysis,'' ASME J. Mech. Des., **119**(3), pp. 338–345.
[26] Bottema, O., and Roth, B., 1979, *Theoretical Kinematics*, North Holland. (reprinted Dover Publications 1990).
[27] McCarthy, J. M., 1990, *An Introduction to Theoretical Kinematics*, MIT Press, Boston, MA.
[28] Dietmaier, P., and Pavlin, G., 1995: Automatic Computation of Direct and Inverse Kinematics of General Spatial Mechanisms and Structures, *Proc. of 9th World Congress IFToMM*, Vol. 1, 80–84, Milano.