Robotica (2007) volume 25, pp. 121–128. © 2006 Cambridge University Press
doi:10.1017/S0263574706003080 Printed in the United Kingdom

A direct numerical solution to forward kinematics of general Stewart–Gough platforms

Yunfeng Wang

Department of Mechanical Engineering, The College of New Jersey, Ewing, NJ 08628, USA. E-mail: jwang@tcnj.edu.

(Received in Final Form: July 28, 2006, First published online: September 4, 2006)

SUMMARY
This paper presents a simple numerical method for forward kinematics of general Stewart–Gough platforms, which can generate a unique solution directly. This method utilizes the trivial nature of the inverse kinematics of parallel manipulators, and derives a straightforward linear relationship between the small change in joint variables (leg lengths) and the resulting small motion of the platform. The solution to the forward kinematics is then achieved through a series of small changes in joint variables. Numerical examples validate and confirm the efficiency of the method.

KEYWORDS: Parallel manipulators; Forward kinematics.

1. Introduction

Stewart–Gough platforms (parallel manipulators) consist of two bodies connected by several prismatic legs or kinematic chains in parallel. They offer higher rigidity, faster dynamic response, and larger load/weight ratios over their serial counterparts. They have been broadly applied in flight simulation systems, manufacturing, and medical robots. Extensive research has been conducted on platform design, kinematics, dynamics, singularity analysis, workspace analysis, and calibration, since the concept was first introduced by Gough[1] and further developed by Stewart.[2] Owing to their coupled and closed kinematic loops, forward kinematics is one of the most challenging problems in the area of parallel manipulators. Recall that this problem is to determine the pose (the position and orientation of the moving body relative to the fixed body) of a platform from a given set of leg lengths.

Although the forward kinematics problem has been addressed in numerous works, majority of them focus on finding all the possible solutions to the forward kinematics of certain kinds of parallel manipulators.[3–6] These approaches usually use algebraic formulations to generate a high degree of polynomial or set of nonlinear equations. Then, methods such as algebraic elimination,[7, 8] interval analysis,[6] and continuation[9, 10] are used to find the roots of the polynomial. The forward kinematics problem is not fully solved just by finding all the possible solutions. Schemes are further needed to find a unique actual pose of the platform from among all the possible solutions. Use of an iterative numerical procedure,[11–14] and auxiliary sensors,[15–18] are the two commonly adopted schemes to further lead to a unique solution. Numerical iteration is usually sensitive to the choice of initial values and the nature of the resulting constraint equations. The auxiliary sensors approach has practical limitations, such as cost and measurement errors. Some researchers have also tried using neural networks for solving the forward kinematics problem.[19–21] No matter how the forward kinematics problem may be solved, the direct determination of a unique solution is still a challenging problem.

The complexity of the forward kinematics problem depends widely on the manipulator architecture, geometry, and joint sensor layouts. Though promising results have been achieved for certain simplified configurations, the pursuit of practical algorithms for the general Stewart–Gough platform has continued. This paper is a contribution of this type of algorithm, and presents a simple numerical method to generate the unique pose of a general Stewart–Gough platform directly. Some of the initial results of the proposed method have been presented at a conference.[22]

The proposed incremental method utilizes the trivial nature of the inverse kinematics of parallel manipulators, and derives a straightforward linear relationship between the small change in joint variables (leg lengths) and the resulting small motion of the platform. The solution to the forward kinematics is then achieved through a series of small changes in joint variables. The continuous small changes in joint variables result in a unique forward kinematics solution.

The rest of the paper is organized as follows. Notation from the basic theory of Lie groups used later in the paper are reviewed in Section 2. In Section 3, a simple and efficient method for solving the forward kinematics of general Stewart–Gough platforms is presented. In Section 4, numerical simulations are performed to validate the proposed method. This numerical method is further demonstrated by solving the inverse kinematics of hybrid serial-parallel manipulators, in Section 5.

2. Notation and Terminology

Throughout this paper, terminology and concepts from the theory of Lie groups will be used. Basic ideas, terminology, and notation are reviewed here. For more detailed treatments, see ref. 23 and 26.

The Euclidean motion group (or "special Euclidean" group), $SE(3)$, is the semidirect product of $\mathbb{R}^3$ with the special orthogonal group, $SO(3)$. We denote elements of

$SE(3)$ as $g = (\mathbf{a}, A) \in SE(3)$ where $A \in SO(3)$ and $\mathbf{a} \in I\!R^3$. The group law is written as $g_1 \circ g_2 = (\mathbf{a}_1 + A_1\mathbf{a}_2, A_1A_2)$, and $g^{-1} = (-A^T\mathbf{a}, A^T)$. Any element of $SE(3)$ can be written as the product of pure translation and pure rotation as $(\mathbf{a}, A) = (\mathbf{a}, I) \circ (\mathbf{0}, A)$.

Any element of $SE(3)$ can be represented as a 4×4 homogeneous transformation matrix of the form

$$g(\mathbf{a}, A) = \begin{pmatrix} A & \mathbf{a} \\ \mathbf{0^T} & 1 \end{pmatrix}.$$

Associated with $SE(3)$ is a set of matrices

$$\tilde{E}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{E}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\tilde{E}_3 = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{E}_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\tilde{E}_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad \tilde{E}_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

which, when linearly combined and exponentiated, produce elements of $SE(3)$.

The space spanned by all possible linear combinations of the matrices $\{\tilde{E}_i\}$ (together with the matrix commutator and an inner product, such that $(\tilde{E}_i, \tilde{E}_j) = \delta_{ij}$ is called $SE(3)$. It is the Lie algebra associated with the Lie group $SE(3)$. The commutator relations are written all together as

$$[\tilde{E}_i, \tilde{E}_j] = \sum_{k=1}^{6} C_{ij}^k \tilde{E}_k \tag{1}$$

where C_{ij}^k's are called the structure constants of the Lie algebra and $[\tilde{E}_i, \tilde{E}_j] = \tilde{E}_i\tilde{E}_j - \tilde{E}_j\tilde{E}_i$.

Given a rigid-body motion $g(t)$, the quantity

$$g^{-1}\dot{g} = \begin{pmatrix} A^T\dot{A} & A^T\dot{a} \\ 0^T & 0 \end{pmatrix} \tag{2}$$

is a spatial velocity as seen in the body-fixed frame of reference. This velocity can be described as the six-dimensional (6-D) vector

$$\boldsymbol{\xi} = (\boldsymbol{g}^{-1}\dot{\boldsymbol{g}})^\vee = \sum_{\boldsymbol{i=1}}^{\boldsymbol{6}} (\boldsymbol{g}^{-1}\dot{\boldsymbol{g}}, \tilde{\boldsymbol{E}}_i)\boldsymbol{e}_i = \begin{pmatrix} (\boldsymbol{A^T\dot{A}})^\vee \\ \boldsymbol{A^T\dot{a}} \end{pmatrix}. \tag{3}$$

The $\vee$ operator converts a 4×4 screw matrix into a 6×1 vector.[25] The vector ξ contains both the angular and translational velocity of the motion $g(t)$ as seen in the body-fixed frame of reference.

If an element of $SE(3)$ is parameterized as $g(q_1, \ldots, q_6)$, then the associated "right" (or spatial) Jacobian is of the form

$$\boldsymbol{J_R(q)} = \left[\left(\boldsymbol{g}^{-1}\frac{\boldsymbol{\partial g}}{\boldsymbol{\partial q}_1} \right)^\vee, \ldots, \left(\boldsymbol{g}^{-1}\frac{\boldsymbol{\partial g}}{\boldsymbol{\partial q}_6} \right)^\vee \right]. \tag{4}$$

This definition relates twists to small changes in $\boldsymbol{q}$ as

$$\boldsymbol{\xi} = \boldsymbol{J}_{\mathrm{R}}(\boldsymbol{q})\dot{\boldsymbol{q}}. \tag{5}$$

In the context of rigid-body motion as a function of time, the dot represents the derivative with respect to time.

3. Numerical Forward Kinematics Method

Given a general Stewart–Gough platform, frames of reference are affixed to the upper and lower bodies. Let the connection points, where the legs of the platform meet the lower body at universal or ball joints at positions $x_1, \ldots, x_6$, as measured in the frame of reference, be affixed to the lower body (base). Likewise, let the positions, where the distal ends of the legs meet the joints in the upper body (moving platform), be denoted as $y_1, \ldots, y_6$ in the frame of reference affixed in the upper body. Letting the relative motion between the frame of reference fixed in the upper body and that in the lower body, be denoted as $g \in SE(3)$, the leg lengths are then computed as

$$\boldsymbol{L_i} = \|\boldsymbol{g} \cdot \boldsymbol{y_i} - \boldsymbol{x_i}\| \tag{6}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector, and for any $y \in I\!R^3$, the action of $g = (a, A) \in SE(3)$ is defined as $g \cdot y = Ay + a$. Equation (6) is an expression for the *inverse kinematics* of the platform, $L_i = L_i(g)$, and reflects the trivial nature of such computations. In contrast, obtaining truly-closed-form solutions (i.e., solutions written in terms of algebraic and trigonometric functions, square roots, etc.) for the forward kinematics of the form $g = g(L_1, \ldots, L_6)$, is generally not possible.

To the best of the author's knowledge, the following simple formulation has not been developed by any of the prior authors. Let us imagine making a small change to the leg lengths in Eq. (6). Then, there will be a corresponding small change in the motion, relating the frames of reference in the upper and lower bodies. This is described as

$$\boldsymbol{L_i} + \boldsymbol{\epsilon_i} = \|\boldsymbol{g} \circ \boldsymbol{\gamma} \cdot \boldsymbol{y_i} - \boldsymbol{x_i}\| \tag{7}$$

where ϵ_i is a small change in the length of leg i, and

$$\gamma = \begin{pmatrix} I_3 + \Omega & \boldsymbol{v} \\ \boldsymbol{0}^T & 1 \end{pmatrix}$$

is a small motion such that $\|\boldsymbol{\xi}\| \ll 1$ where

$$\boldsymbol{\xi} = (\gamma - I_4)^\vee. \tag{8}$$

I_n is an $n \times n$ identity matrix and Ω a skew-symmetric matrix. Note that due to the fact that $(g \circ \gamma) \cdot y = g \cdot (\gamma \cdot y)$, there is no need to retain parenthesis.

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

If we subtract Eq. (6) from Eq. (7), the result relates the incremental changes in leg lengths to the incremental changes in end pose as

$$\epsilon_i = \|g \circ \gamma \cdot y_i - x_i\| - \|g \cdot y_i - x_i\| = \|\gamma \cdot y_i - g^{-1} \cdot x_i\| - \|y_i - g^{-1} \cdot x_i\|.$$

Since γ is a small motion, $\gamma \cdot y_i = \omega \times y_i + v$, where $\Omega^{\vee} = \omega$. Applying the multidimensional first-order Taylor series approximation

$$\|b + \delta\| \approx \|b\| + b^T \delta / \|b\|,$$

to $\|\gamma \cdot y_i - g^{-1} \cdot x_i\|$, allows us to write

$$\epsilon_i = m_i^T \xi$$

where

$$m_i^T = (g^{-1} \cdot x_i - y_i)^T (y_i \times, -I_3) / \|g^{-1} \cdot x_i - y_i\|,$$

and $y\times$ denotes the 3×3 skew-symmetric matrix corresponding to y. Stacking these equations for $i = 1, \ldots, 6$, results in a linear system of equations of the form

$$\boldsymbol{\epsilon} = M\boldsymbol{\xi} \tag{9}$$

which can be inverted to find ξ. Then, the small motion γ corresponding to ξ can be obtained from Eq. (8). The platform configuration now becomes $g \circ \gamma$. This process is iterated to follow a trajectory in the leg-length space and compute the corresponding forward kinematic path. Due to the continuous small change in the leg lengths, this numerical method directly leads to a unique foward kinematics solution.

4. Numerical Simulations of Forward Kinematics

4.1. Distance metric

A distance metric is first selected to check the closeness of the platform pose obtained from the proposed numerical forward kinematics method, g_{num}, and the desired pose, g_{des}. The distance metric for $SE(3)$ given by ref. 27 is chosen for the numerical simulations, though others exist.[23, 28]. This distance metric is defined as

$$D(g_1(\mathbf{a}_1, A_1), g_2(\mathbf{a}_2, A_2)) = \sqrt{\|\mathbf{a}_1 - \mathbf{a}_2\|^2 + \alpha \|\log A_1^T A_2\|^2} \tag{10}$$

where g_1 and g_2 are elements of $SE(3)$, $\|\cdot\|$ is the Euclidean norm, and α is a parameter used to balance the position error and orientation error. Here, we choose the value of α so that the position error and the orientation error have the same order.

4.2. Numerical simulations of forward kinematics

To simulate and verify the proposed forward kinematics method, we first perform theinverse kinematics from Eq. (6) to get the leg lengths for a given pose (configuration), g_{des}. This given pose, g_{des}, will serve as the desired pose for the accuracy check. Then, using these leg lengths, we conduct the proposed forward kinematics to find the pose, g_{num}. The correctness and accuracy of the proposed incremental method can be checked by comparing g_{des} and g_{num} with the distance metric given in Eq. (10).

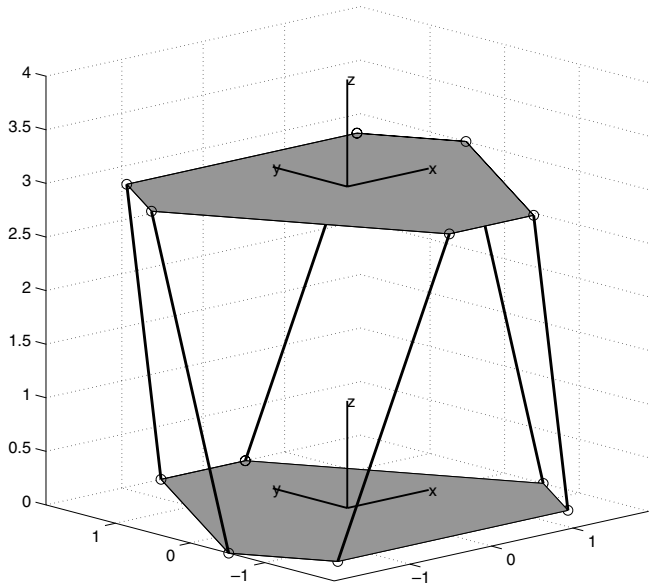

Fig. 1. The example general Steward–Gough platform.

A general Stewart–Gough platform is selected as an example parallel manipulator for numerical simulations. It is shown in Fig. 1. The coordinates of the six connection points at the lower body and the upper body are chosen as

$$\begin{pmatrix} 2\sin(2(i-1)\pi/3 \pm \pi/12) \\ 2\cos(2(i-1)\pi/3 \pm \pi/12) \\ 0 \end{pmatrix}$$

and

$$\begin{pmatrix} 2\sin(2(i-1)\pi/3 \pm \pi 11/12) \\ 2\cos(2(i-1)\pi/3 \pm \pi 11/12) \\ 0 \end{pmatrix}$$

where, $i = 1, \ldots, 3$, respectively. The initial pose for this platform is

$$g_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

The six leg lengths for this initial setup are calculated from Eq. (6) as

$$\mathbf{L0} = [3.1736,\ 3.1736,\ 3.1736,\ 3.1736,\ 3.1736,\ 3.1736].$$

In the numerical simulations, the orientation part of the frame of reference is parameterized with the *Z–Y–X*

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

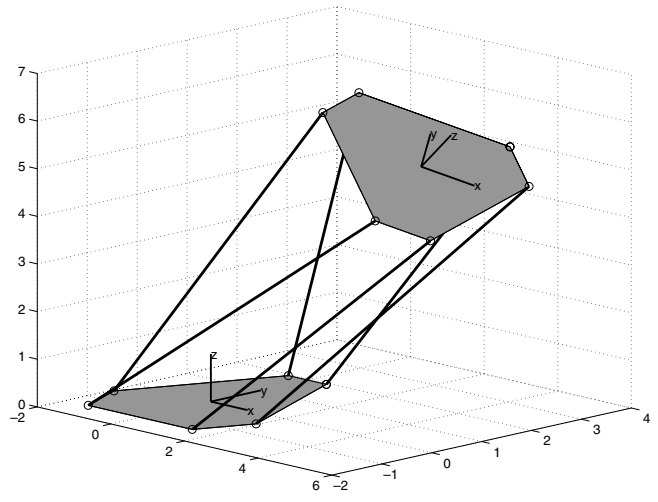

Fig. 2. The platform configuration corresponding to the leg lengths **L1**.

Euler angles.[23] For an orientation with $(\pi/6, \pi/6, \pi/3)$, and position with $\mathbf{a} = [3, 2, 5]^T$, the desired pose is written as

$$g_{\text{des}} = \begin{pmatrix} 0.4330 & -0.6250 & 0.6495 & 3.0000 \\ 0.7500 & 0.6495 & 0.1250 & 2.0000 \\ -0.5000 & 0.4330 & 0.7500 & 5.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}.$$

The leg lengths of the example platform corresponding to the pose g_{des} are calculated from Eq. (6) as

$$\mathbf{L1} = [5.7568,\ 6.6353,\ 7.3836,\ 7.1991,\ 5.5535,\ 6.2567].$$

For this example, the initial leg lengths of the manipulator are calculated as **L0** and the final leg lengths as **L1**. The small change in the leg lengths ϵ_i is chosen as 0.01. This is treated as a positive number in case the initial leg length is shorter than the final leg length, and a negative number in case of vice versa. Using the proposed forward kinematics method, the platform configuration corresponding to the final leg lengths computed within 0.4 s, using Matlab with a 1.0-GHz and 516-MB RAM computer, is

$$g_{\text{num}} = \begin{pmatrix} 0.4331 & -0.6250 & 0.6495 & 3.0003 \\ 0.7500 & 0.6496 & 0.1250 & 2.0001 \\ -0.5001 & 0.4330 & 0.7500 & 5.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}.$$

The error between the desired frame and the numerically obtained frame is $D(g_{\text{des}}, g_{\text{num}}) = 0.001$. The parameter α in the distance metric [Eq. (10)] is chosen as 0.01. The error can be further reduced with smaller ϵ but at the expense of computation time. Figure 2 shows the configuration of the Stewart platform corresponding to the leg lengths **L1**.

The effects of the small change in leg length ϵ, on accuracy and computation time are further studied below. The accuracy is evaluated with the distance metric given in Eq. (10), and the computation time with the CPU time. With the same setup

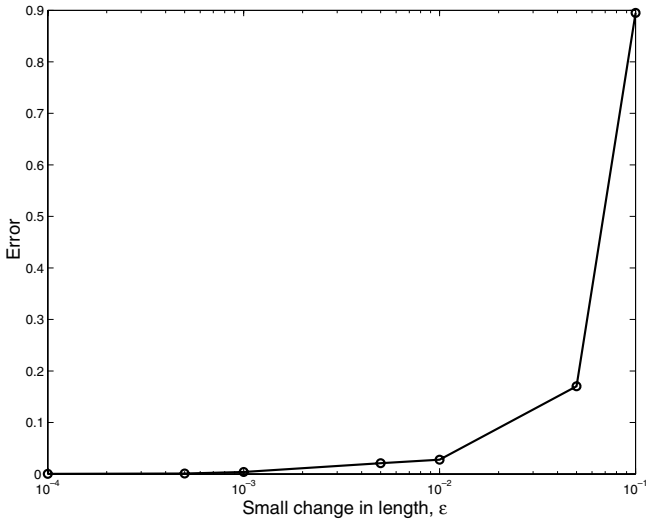

Fig. 3. The effect of ϵ on accuracy.

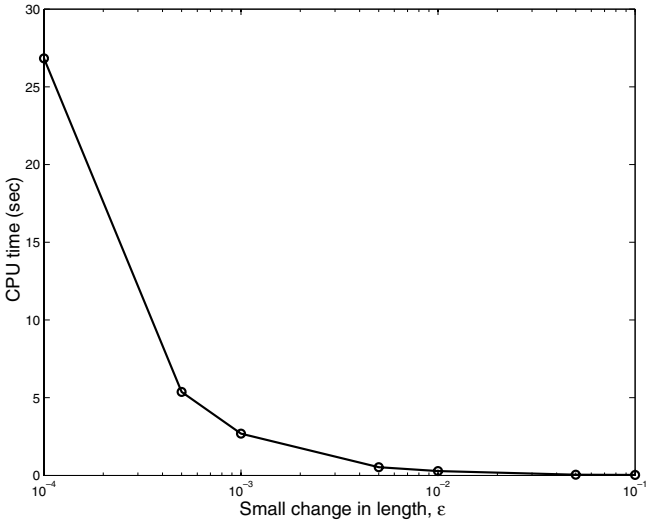

Fig. 4. The effect of ϵ on computation time.

of the previous example, the error and computation time as a function of ϵ are plotted in semi-logarithmic graphs in Figs. 3 and 4, respectively. In these graphs, ϵ is changed from 10^{-1} to 10^{-4}. It can be observed that, as ϵ decreases, the error decreases as rapidly as the increase in computation time. When ϵ reduces from 10^{-3} to 10^{-4}, the error is almost at the same level, but the computation time increases dramatically. From these observations, it is easy to choose a suitable value of ϵ that balances the error and the computation time. These graphs are generated using Matlab with a 1.0-GHz and 516-MB RAM computer.

5. Demonstration to the Inverse Kinematics of Hybrid Manipulators

The proposed numerical forward kinematics is now demonstrated by applying it to the inverse kinematics of hybrid manipulators, in which the forward kinematics solution of

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

each platform is required. The inverse kinematics method uses Jacobian-based redundancy resolution techniques, such as those in ref. 29 and 30, and is stated in detail as follows.

Let the numerically determined forward kinematics for the ith module in a hybrid manipulator be denoted as $g_i^{i-1}(q_i)$, where q_i, is the vector of joint variables for this module, and g_i^{i-1} is the homogeneous transformation describing the position and orientation of the ith frame relative to the $(i-1)$st. Then, the numerical forward kinematics for the whole hybrid manipulator will be

$$\boldsymbol{g}_{\mathrm{ee}}(\boldsymbol{q}) = \boldsymbol{g}_n^0(\boldsymbol{q}) = \boldsymbol{g}_1^0(\boldsymbol{q}_1)\cdots\boldsymbol{g}_i^{i-1}(\boldsymbol{q}_i)\cdots\boldsymbol{g}_n^{n-1}(\boldsymbol{q}_n) \quad (11)$$

where $q = [q_1^T, q_2^T, \ldots, q_n^T]^T$ is the composite vector of all joint variables, and g_{ee} is the desired end-effector position and orientation. Given the forward kinematics in Eq. (11), we can always obtain numerical inverse kinematics by iterating velocity kinematics with the given initial configurations as follows. For the initial configurations of a hybrid manipulator $q(0)$ and $g_{\mathrm{ee}}(0) = g(q(0))$, and the trajectory for the end-effector $g_{\mathrm{trj}}(t)$, the corresponding joint variables $q(t)$ can be numerically found with the recursive formula

$$\boldsymbol{q}(t_n) = \boldsymbol{q}(\boldsymbol{t}_{n-1}) + \boldsymbol{\Delta t}\,\dot{\boldsymbol{q}}(\boldsymbol{t}_{n-1}) \quad (12)$$

where Δt is the small division for numerical calculation, $t_n = n\,\Delta t$, and $n = 1, \ldots, t/\Delta t$. The velocity (derivative) term in Eq. (12) can be obtained from the definition of the spatial Jacobian matrix (see Section 2)

$$\boldsymbol{\xi}_{\mathrm{ee}} = \left(\boldsymbol{g}_{\mathrm{ee}}^{-1}\dot{\boldsymbol{g}}_{\mathrm{ee}}\right)^{\vee} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}, \quad (13)$$

that is,

$$J = \left[\left(\left(g_n^0\right)^{-1}\frac{\partial g_n^0}{\partial q_{11}}\right)^{\vee}, \ldots, \left(\left(g_n^0\right)^{-1}\frac{\partial g_n^0}{\partial q_{ij}}\right)^{\vee}, \ldots, \times\left(\left(g_n^0\right)^{-1}\frac{\partial g_n^0}{\partial q_{nm}}\right)^{\vee}\right]$$

where q_{ij} stands for the jth joint variable of the ith module of the manipulator.

5.1. Numerical demonstration

A hybrid manipulator, consisting of four modules of a general Stewart–Gough platform shown in Fig. 1, is constructed for the numerical demonstration of inverse kinematics. This example hybrid manipulator has a total of 24 joint variables with 6 joint variables for each module. Define $\mathbf{q}_n$ as the vector of joint variables of the nth module. The configuration of the ith module is indicated by g_i^{i-1}, which is the homogeneous transformation describing the position and orientation of the ith frame, relative to the $(i-1)$st. Here, the module number is labeled from 1 to n, starting from the proximal end of the manipulator. The frame of reference affixed to the upper body of the ith module is labeled with i, and that to the lower body is labeled with $i-1$.

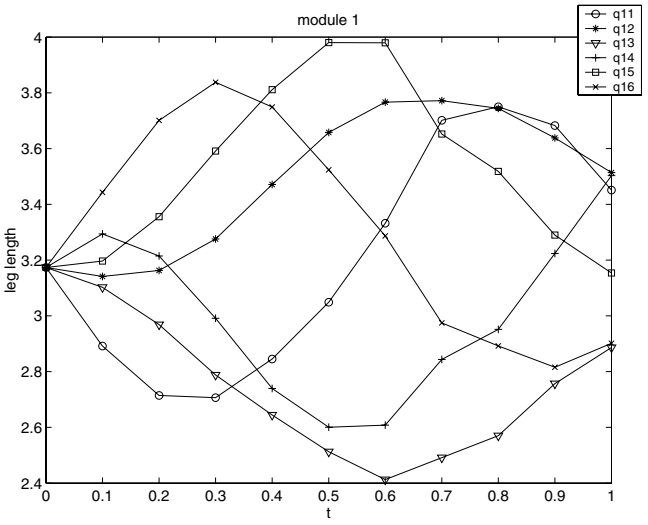

Fig. 5. The computed joint variables of module 1.

A circular trajectory with a radius of 4 units is planned. The orientation of the frame attached to the distal end of the manipulator moves sinusoidally with respect to the frame attached to the base. This trajectory is parameterized with a parameter t, and $t \in [0, 1]$. The orientation part is expressed by Z–Y–X Euler angles with $(\alpha, \beta, \gamma)_{\mathrm{trj}} = (0, 0, 2\pi\sin(2\pi t)/10)$, and the position part by $\mathbf{a}_{\mathrm{trj}} = [4\cos(2\pi t),\ 4\sin(2\pi t),\ 12]^T$.

The initial configurations (at $t = 0$) of this hybrid manipulator are assumed as

$$g_1^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad g_2^1 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$g_3^2 = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \qquad g_4^3 = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Hence, the initial joint variables (leg lengths) for each module are

$$\mathbf{q}_1 = [3.1736,\ 3.1736,\ 3.1736,\ 3.1736,\ 3.1736,\ 3.1736],$$

$$\mathbf{q}_2 = [3.0461,\ 3.5868,\ 3.3274,\ 3.0461,\ 3.5868,\ 3.3274],$$

$$\mathbf{q}_3 = [3.5868,\ 3.0461,\ 3.3274,\ 3.5868,\ 3.0461,\ 3.3274],$$

$$\mathbf{q}_4 = [3.2381,\ 4.2022,\ 3.7512,\ 3.2381,\ 4.2022,\ 3.7512].$$

Using the inverse kinematics method stated above, the joint variables along the planned circular trajectory and the manipulator configurations (frame of references) can be easily calculated. Figures 5–8 illustrate the calculated joint variables of each module at 11 points along the trajectory by taking t from 0 to 1 with an increment of 0.1.

The configurations of the hybrid manipulator at $t = 0, 0.25$, 0.5, 0.75, 1 are shown in Fig. 9. The desired and computed end-effector frame at the values above t are listed in Table I.

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

Table I. The desired and computed frames at illustrated t values and their errors.

t	g_{des}	g_{num}	Error
0.25	$\begin{pmatrix} 0.8090 & -0.5878 & 0 & 0.0000 \\ 0.5878 & 0.8090 & 0 & 4.0000 \\ 0 & 0 & 1.0000 & 12.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\begin{pmatrix} 0.8089 & -0.5884 & 0.0006 & -0.0119 \\ 0.5884 & 0.8089 & -0.0002 & 4.0002 \\ -0.0004 & 0.0005 & 1.0001 & 11.9997 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	0.0119
0.5	$\begin{pmatrix} 1.0000 & -0.0000 & 0 & -4.0000 \\ 0.0000 & 1.0000 & 0 & 0.0000 \\ 0 & 0 & 1.0000 & 12.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\begin{pmatrix} 1.0005 & 0.0034 & -0.0001 & -3.9974 \\ -0.0034 & 1.0005 & 0.0002 & -0.0145 \\ 0.0001 & -0.0003 & 1.0002 & 12.0014 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	0.0152
0.75	$\begin{pmatrix} 0.8090 & 0.5878 & 0 & -0.0000 \\ -0.5878 & 0.8090 & 0 & -4.0000 \\ 0 & 0 & 1.0000 & 12.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\begin{pmatrix} 0.8090 & 0.5890 & 0.0005 & 0.0191 \\ -0.5890 & 0.8090 & 0.0002 & -3.9980 \\ -0.0004 & -0.0004 & 1.0004 & 12.0003 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	0.0192
1	$\begin{pmatrix} 1.0000 & 0.0000 & 0.0000 & 4.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 \\ 0 & 0 & 1.0000 & 12.0000 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	$\begin{pmatrix} 1.0009 & 0.0030 & 0.0001 & 3.9963 \\ 0.0030 & 1.0010 & 0.0004 & 0.0123 \\ 0.0001 & 0.0004 & 1.0005 & 12.0021 \\ 0 & 0 & 0 & 1.0000 \end{pmatrix}$	0.0134

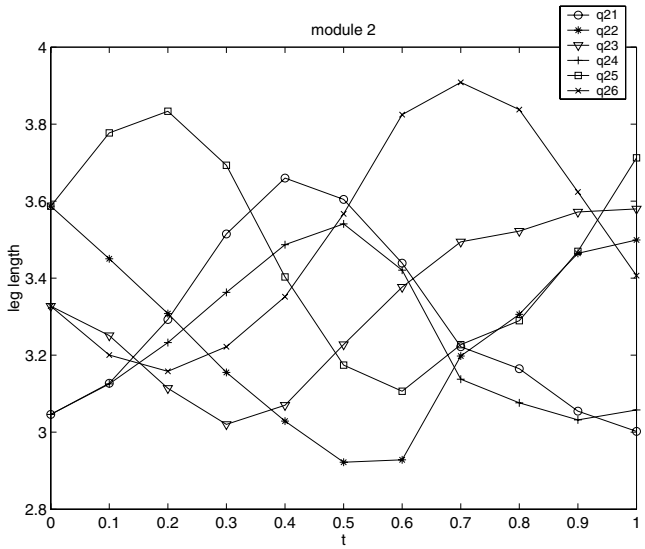

Fig. 6. The computed joint variables of module 2.

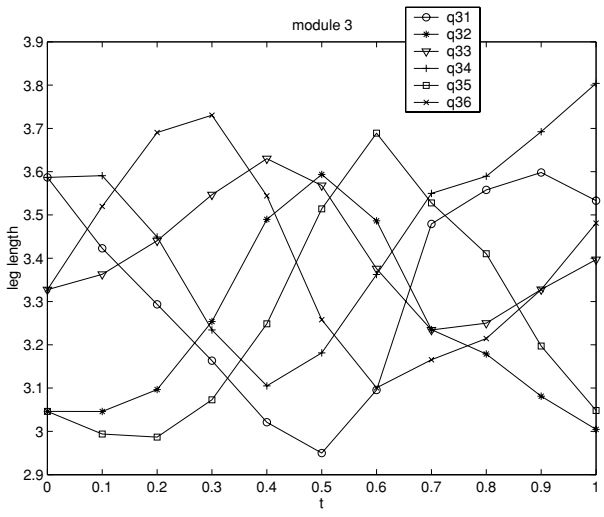

Fig. 7. The computed joint variables of module 3.

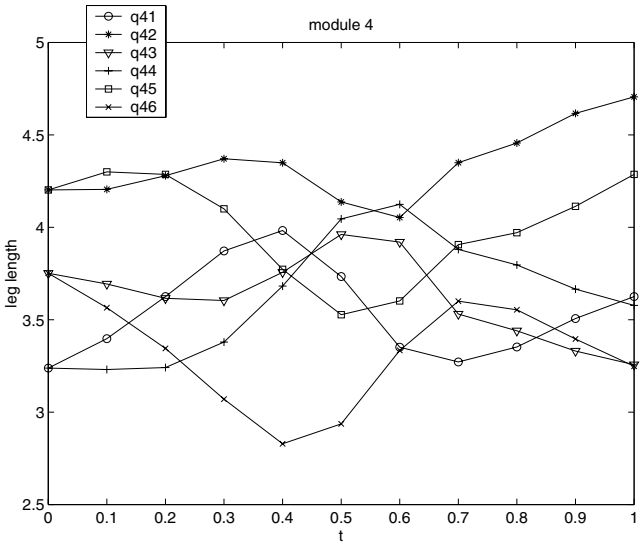

Fig. 8. The computed joint variables of module 4.

It can be seen that the errors are very small and to the order of 10^{-2}.

For the above calculations, we take $\epsilon = 0.001$, and $\Delta t = 0.01$. It takes less than 3 min to get the solutions, using Matlab with a 1.0 GHz and 516 MB RAM computer.

6. Conclusions

A simple numerical method that can directly generate a unique forward kinematics solution of general Stewart–Gough platforms is developed. This method is numerically stable and computationally efficient. Compared with the other numerical methods, this method is more straightforward and systematic, and independent of manipulator type. The effectiveness of this method is demonstrated by numerical simulations and its application to solve the inverse kinematics of hybrid manipulators.

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

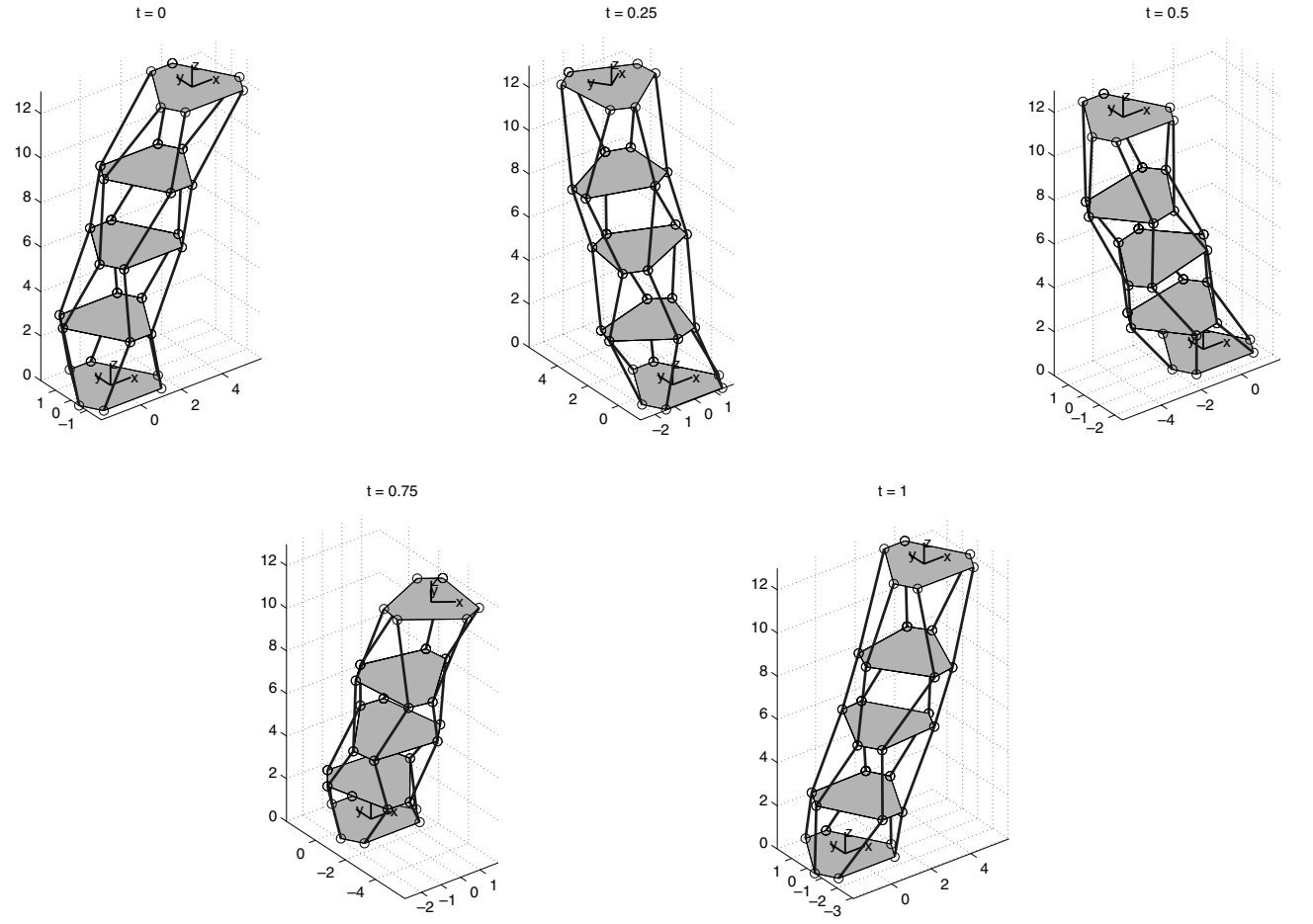

Fig. 9. The computed manipulator configurations along the trajectory at different t.

References

1. V. E. Gough, "Contribution to discussion to papers on research in automobile stability on control and in tyre performance," *Proceedings of the Automotive Division Instrument Engineering* (1956).
2. D. Stewart, "A platform with 6 degrees of freedom," *Proceedings of the Institute of Mechanical Engineering* **180**(1), 371–386 (1965).
3. P. Dietmaier, "The Stewart–Gough platform of general geometry can have 40 real postures," *Proceedings of ARK*, Strobl, Austria (Jun 1998) pp. 7–16.
4. M. L. Husty, "An algorithm for solving the direct kinematics of general Stewart–Gough platforms," *Mech. Mach. Theory* **31**(4), 365–379 (1996).
5. P. Ji. and H. Wu, "A closed-form forward kinematics solution for the 6-6P stewart platform," *IEEE Trans. Robot. Autom.* **17**(4), 522–526 (2001).
6. J. P. Merlet, "Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis," *Int. J. Robot. Res.* **23**(3), 221–235 (2004).
7. C. Innocenti, "Forward kinematics in polynomial form of the general Stewart platform," *J. Mech. Des.* **123**(2), 254–260 (2001).
8. T. Y. Lee and J. K. Shim, "Forward kinematics for the general 6-6 Stewart platform using algebraic elimination," *Mech. Mach. Theory* **36**, 1073–1085.
9. M. Raghavan, "The Stewart platform of general geometry has 40 configurations," *Proceedings of the ASME Design and Automation Conference*, Chicago, IL, (Sep. 1991). Vol. 32-2, pp. 397–402.
10. C. W. Wampler, "Forward displacement analysis of general six-in-parallel SPS (Stewart) platform manipulators using soma coordinates," *Mech. Mach. Theory* **31**(3), 331–337 (1996).
11. C. Innocenti, "A novel numerical approach to the closure of the 6-6 Stewart platform mechanism," *Adv. Robot.* 852–855 (1991).
12. D. M. Ku, "Direct displacement analysis of a Stewart plat–form mechanism," *Mech. Mach. Theory* **34**, 453–465-(1999).
13. J. P. Merlet, "Direct kinematics of parallel manipulators," *IEEE Trans. Robot. Autom.* **9**(6), 842–845 (1993).
14. L. C. T. Wang and C. C. Chen, "On the numerical kinematic analysis of general parallel robotic manipulators," *IEEE Trans. Robot. Autom.* **9**(3), 272–285 (1993).
15. L. Baron and J. Angeles, "The direct kinematics of parallel manipulators under joint-sensor redundancy," *IEEE Trans. Robot. Autom.* **16**(1) (2000).
16. Y. J. Chiu and M. H. Perng, "Forward kinematics of a general fully parallel manipulator with auxiliary sensors," *Int. J. Robot. Res.* **20**(5) (2001).
17. K. R. Han, W. K. Chung and Y. Youm, "New resolution scheme of the forward kinematics of parallel manipulators using extra sensors," *ASME J. Mech. Des.* 214–219 (1996).
18. V. Parenti-Castelli and R. D. Gregorio, "A new algorithm based on two extra-sensors for real-time computation for the actual configuration of the generalized Stewart–Gough manipulator," *ASME J. Mech. Des.* **112**, 294–298 (2000).
19. Z. Geng and L. S. Haynes, "Neural network solution for the forward kinematics problem of a Stewart platform," *Robot. Comput. Integrat. Manuf.* **9**(6), 485–495 (1992).
20. C. S. Yee, "Forward kinematics solution of Stewart platform using neural networks," *Neurocomputing* **16**(4), 333–349 (1997).

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press

21. P. J. Parikh and S. Y. Lam, "A hybrid strategy to solve the forward kinematics problem in parallel manipulators," *IEEE Trans. Robot.* **21**(1), 18–25 (2005).
22. Y. Wang, "An incremental method for forward kinematics of parallel manipulators," *Proceedings of the IEEE International Conference on Robotics, Automation and Mechantronics*, Bangkok, Thailand, (Jun 2006). pp. 243–247.
23. G. S. Chirikjian and A. B. Kyatkin, *Engineering Applications of Noncommutative Harmonic Analysis* (CRC: Boca Raton, FL, 2001).
24. A. Karger and J. Novák, *Space Kinematics and Lie Groups* (Gordon and Breach Science Publishers, New York, 1985).
25. R. M. Murray, Z. Li and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, Boca Raton 1994).
26. J. M. Selig, *Geometrical Methods in Robotics* (Springer, New York, 1996).
27. F. C. Park, "Distance metrics on the rigid-body motions with applications to mechanism design," *Trans. ASME, Mech. Des.* **117**, 58–54 (1995).
28. G. S. Chirikjian and S. Zhou, "Metrics on motion and deformation of solid models," *ASME J. Mech. Des.* **120**(2), 252–261 (1998).
29. C. A. Klein and C. H. Huang, "Review of pseudoinverse control for use with kinematically redundant manipulators," *IEEE Trans. Syst. Man and Cybern.* **13**(2), 245–250 (1983).
30. D. E. Whitney, "Resolved motion-rate control of manipulators and human prostheses," *IEEE Trans. Man-Mach. Syst.* **10**(2), 47–53 (1969).

https://doi.org/10.1017/S0263574706003080 Published online by Cambridge University Press