

A Geometric Newton-Raphson Method for Gough-Stewart Platforms

J.M. Selig and Hui Li

Abstract A geometric version of the well known Newton-Raphson methods is introduced. This root finding method is adapted to find the zero of a function defined on the group of rigid body displacements. At each step of the algorithm a rigid displacement is found that approximates the solution. The method is applied to the forward kinematics problem of the Gough-Stewart platform.

1 Introduction

The forward kinematics of parallel manipulators is a well known ‘hard’ problem. It requires that the rigid-body displacement undergone by the platform is found given the lengths of the six legs. For a general Gough-Stewart platform, there are up to 40 possible solutions, [10, 6, 9, 11, 1, 14]. Many workers have used numerical methods for this problem, for example [4, 15, 2, 7, 3, 8]. Most of this work concerns finding all solutions and uses general numerical techniques which do not take account of the geometry of the group of rigid-body displacements.

Here we present a practical, fast numerical algorithm that finds a single solution given the solution at a nearby position. The idea is that the algorithm might be used in real-time with the solution at the last time-step as initial guess for the current configuration. Many different root-finding techniques could be used here but notice that the solution required is a rigid displacement. So it seems sensible to use a method that respects this structure. That is a geometric numerical method, a method designed to work on a smooth manifold or Lie group. Such methods are becoming more popular though most current work seems to be on geometric integration, see [5] for example.

J.M. Selig and Hui Li
Faculty of Business, Computing and Information Systems, London South Bank University, London
SE1 0AA, U.K. e-mail: seligjm@lsbu.ac.uk

The idea of a geometric root finding method for this problem seem to have been first suggested by Wang [16]. The method proposed here has several differences to Wang's. Our method is a geometric version of the well-known Newton-Raphson method and hence can be expected to be fast and reasonably robust. Also our method of updating the approximation for the rigid displacement is based on the exponential map, this is more accurate than the method used by Wang. Thirdly, our condition for termination of the algorithm is based on the difference between the desired leg-lengths and the computed lengths rather than an *ad hoc* metric in the group. There are some other minor differences too.

We begin with some background material to establish a convenient notation.

2 Rigid Body Displacements and Screws

Rigid body displacements can be written as 4×4 matrices,

$$M = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix}$$

with R a 3×3 rotation matrix and \mathbf{t} a translation vector. We denote a point in space by a vector $\mathbf{p} = (x, y, z)^T$, then we extend this vector by adding an extra 1. After the rigid transformation the new position of the point will be given by,

$$\tilde{\mathbf{p}}' = \begin{pmatrix} \mathbf{p}' \\ 1 \end{pmatrix} = \begin{pmatrix} R & \mathbf{t} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ 1 \end{pmatrix} = \begin{pmatrix} R\mathbf{p} + \mathbf{t} \\ 1 \end{pmatrix} = M\tilde{\mathbf{p}}.$$

Notice that we use the tilde to denote the 4-dimensional position vector.

Elements of the Lie algebra of the group correspond to velocities or small displacements, they are called twists or screws. A motion is a path in the group given by a continuously parameterised sequence of group elements $M(t)$. At any parameter value there is a Lie algebra element given by,

$$S = \left(\frac{d}{dt} M(t) \right) M(t)^{-1} = \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix},$$

where Ω is a 3×3 anti-symmetric matrix corresponding to the angular velocity of the motion. The vector \mathbf{v} is the linear velocity of the origin.

It is often convenient to write screws as 6-dimensional vectors, corresponding to a screw $S = \begin{pmatrix} \Omega & \mathbf{v} \\ 0 & 0 \end{pmatrix}$ we can write a vector, $\mathbf{s} = \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix}$, where the vector $\boldsymbol{\omega}$ corresponds to the anti-symmetric matrix Ω in the following way, for any vector \mathbf{u} we have $\Omega\mathbf{u} = \boldsymbol{\omega} \times \mathbf{u}$.

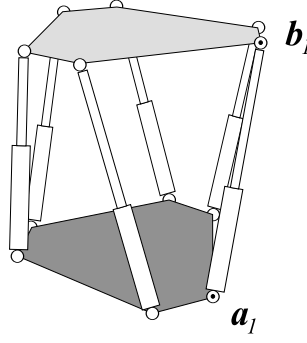


Fig. 1 The General Gough-Stewart Platform.

Elements of the dual space to the Lie algebra are called wrenches and written, $\mathcal{W} = \begin{pmatrix} \tau \\ \mathbf{F} \end{pmatrix}$, where \mathbf{F} is a force and τ is a moment. These could also be linear and angular momenta, if we were considering dynamics.

If a rigid body is moving with instantaneous velocity given by the screw \mathbf{s} and is acted on by a total force and moment given by the wrench \mathcal{W} then the instantaneous power exerted on the body is given by,

$$\text{power} = \mathcal{W}^T \mathbf{s} = \tau \cdot \omega + \mathbf{F} \cdot \mathbf{v}.$$

A more comprehensive account of this view of screw theory can be found in [12].

3 The Gough-Stewart Platform

Assume the position and orientation of the platform is given, the leg-lengths are simple to find. Let \mathbf{a}_i be the position of the centre of the spherical joint on the i -th leg at the base. In the home configuration the corresponding position of the joint centre on the platform will be, \mathbf{b}_i . The length of the i -th leg, or rather its square, is,

$$l_i^2 = (\tilde{\mathbf{a}}_i - M\tilde{\mathbf{b}}_i)^T (\tilde{\mathbf{a}}_i - M\tilde{\mathbf{b}}_i) \quad i = 1, \dots, 6. \quad (1)$$

Here M is a rigid transformation written as a 4 matrix and $\tilde{\mathbf{b}}_i$ is the 4-vector corresponding to the point \mathbf{b}_i , see the previous section.

In equation (1), the matrix M represents the displacement that takes the platform from home to the current position. Think of the leg lengths as components of a mapping from the group to the space of leg-lengths, $SE(3) \rightarrow \mathbb{R}^6$. A point in \mathbb{R}^6 is given in coordinates as (l_1, l_2, \dots, l_6) . We will need the Jacobian of this mapping, and its inverse later. To find the Jacobian the derivatives of the leg-lengths can be taken,

$$\left. \frac{dl_i^2}{dt} \right|_{t=0} = 2l_i \dot{l}_i = -2(\tilde{\mathbf{a}}_i - \tilde{\mathbf{b}}_i)^T S \tilde{\mathbf{b}}_i. \quad (2)$$

The matrix S here is the velocity screw of the rigid transformation, $S = (\dot{M})M^{-1}$.

Rearranging (2) using the cyclic property of the scalar triple product, gives,

$$\dot{l}_i = \frac{1}{l_i} (\tilde{\mathbf{b}}_i - \tilde{\mathbf{a}}_i)^T S \tilde{\mathbf{b}}_i = \frac{1}{l_i} ((\mathbf{a}_i \times \mathbf{b}_i)^T, (\mathbf{b}_i - \mathbf{a}_i)^T) \begin{pmatrix} \omega \\ \mathbf{v} \end{pmatrix}.$$

This gives the joint rate of each leg as a linear function of the velocity screw of the platform. The Jacobian J , is the matrix satisfying the formula,

$$\begin{pmatrix} \dot{l}_1 \\ \vdots \\ \dot{l}_6 \end{pmatrix} = J \begin{pmatrix} \omega \\ \mathbf{v} \end{pmatrix}.$$

So it can be seen that the rows of this Jacobian matrix are simply,

$$\mathcal{W}_i^T = \frac{1}{l_i} ((\mathbf{a}_i \times \mathbf{b}_i)^T, (\mathbf{b}_i - \mathbf{a}_i)^T), \quad i = 1, \dots, 6.$$

This is the wrench given by a unit force directed along the i -th leg.

4 A Geometric Newton-Raphson Method

In general we are trying to find a root of the vector function $\mathbf{F}(M)$, in particular this vector function is given by the inverse kinematics of the platform. That is we set,

$$L_i = (\tilde{\mathbf{a}}_i - M\tilde{\mathbf{b}}_i)^T (\tilde{\mathbf{a}}_i - M\tilde{\mathbf{b}}_i) - l_i^2, \quad i = 1, \dots, 6,$$

and then,

$$\mathbf{F}(M) = (L_1, L_2, L_3, L_4, L_5, L_6)^T.$$

So, given the six leg-lengths l_1, \dots, l_6 we seek the rigid transformation M which satisfies $\mathbf{F}(M) = \mathbf{0}$.

The key idea here is to represent an error in position and orientation by an exponential of the form e^S , where S represents the error in the Lie algebra. We will refer to S as the *error screw*. So the recursion for the rigid-body displacement we are trying to find will have the form,

$$M^{(i+1)} = e^{S^{(i)}} M^{(i)}$$

where $M^{(i)}$ is the i -th iterate for the solution and $S^{(i)}$ is the i -th error screw. The error screw is found as follows. Consider the Taylor series approximation for the function $\mathbf{F}(e^{tS}M)$ about the root M ,

$$\mathbf{F}(e^{tS}M) \approx \mathbf{F}(M) + t \frac{d}{dt} \mathbf{F}(e^{tS}M)_{t=0}.$$

Since M is a root of \mathbf{F} , $\mathbf{F}(M) = \mathbf{0}$. To compute the derivative of \mathbf{F} we can look at the component functions and as in the previous section,

$$\left. \frac{dL_i}{dt} \right|_{t=0} = -2(\tilde{\mathbf{a}}_i - M\tilde{\mathbf{b}}_i)^T S M \tilde{\mathbf{b}}_i = 2((\mathbf{a}_i \times \mathbf{b}'_i)^T, (\mathbf{b}'_i - \mathbf{a}_i)^T) \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{v} \end{pmatrix}, \quad (3)$$

where $\mathbf{b}'_i = R\mathbf{b}_i + \mathbf{t}$ is the position of the point \mathbf{b}_i at the solution. The Taylor expansion can now be written,

$$\mathbf{F}(e^{tS}M) \approx K(M)\mathbf{s}t,$$

where the matrix $K(M) = 2 \text{diag}(l_1, l_2, \dots, l_6)J(M)$, with $J(M)$ the Jacobian of the platform. The error screw \mathbf{s} , is found by solving the above equation with $t = 1$, so $\mathbf{s} = -K^{-1}(M)\mathbf{F}(e^S M)$.

As usual with the Newton-Raphson method, we don't know the value of the inverse Jacobian at the solution M so we approximate it by $K^{-1}(M^{(i)})$. This justifies our use of the following recurrence relation for \mathbf{s} ,

$$\mathbf{s}^{(i)} = -K^{-1}(M^{(i)})\mathbf{F}(M^{(i)}).$$

This recurrence relation, together with the update relation for $M^{(i)}$:

$$M^{(i+1)} = e^{S^{(i)}} M^{(i)},$$

forms our numerical method. A sensible choice for the condition for iteration to terminate is that the quantity $|\mathbf{F}(M^{(i)})|^2$ be smaller than some predetermined threshold. Notice that this quantity is the sum of the squares of the errors, $L_1^2 + \dots + L_6^2$. In practical situations the threshold value should be determined by the accuracy to which the leg-lengths can be measured.

5 Summary of the Algorithm

For convenience we summarise the algorithm sketched above here:

Inputs:

- Home position of passive joints $\mathbf{a}_1, \dots, \mathbf{a}_6, \mathbf{b}_1, \dots, \mathbf{b}_6$,
- Current position and orientation $M^{(0)}$,
- Desired leg-lengths, l_1, \dots, l_6 ,
- Accuracy threshold, δ .

Outputs:

- Position and orientation for desired leg-lengths, M .

Method:

- Compute $\mathbf{F}(M^{(0)})$,

Compute $|\mathbf{F}(M^{(0)})|^2$,
 While $\delta > |\mathbf{F}(M^{(i)})|^2$ Repeat:
 Evaluate the Jacobian $K(M^{(i)})$,
 Compute the error screw,
 $\mathbf{s}^{(i+1)} = -K^{-1}(M^{(i)})\mathbf{F}(M^{(i)})$,
 Update the position and orientation estimate,
 $M^{(i+1)} = e^{\mathbf{s}^{(i)}} M^{(i)}$,
 Compute $\mathbf{F}(M^{(i+1)})$,
 Compute $|\mathbf{F}(M^{(i+1)})|^2$,
 Output $M = M^{(i+1)}$.

6 Implementation Details

The proposed algorithm was implemented in *Mathematica* for simplicity and little attention was paid to optimising the code. The following remarks are relevant whatever language is used.

When computing the rows of the Jacobian equation (3) should be used rather than the relation $K(M) = 2 \text{diag}(l_1, l_2, \dots, l_6)J(M)$.

The error screw \mathbf{s} , can be found using standard linear algebra libraries. Of course this will fail when the Jacobian is singular or nearly singular and these exceptions should be caught.

A fundamental question is whether rigid displacements should be represented by matrices or quaternions. The action of matrices on points is simpler than the corresponding action using quaternions. On the other hand the product of two rigid displacements is simpler using quaternions. Especially if one considers that the result of a matrix multiplication must be subject to a Gram-Schmidt orthogonalisation to ensure that it is still an orthogonal matrix. By contrast, quaternions only need to be normalised.

The exponential of a screw S can be computed using a degree 3 polynomial in the 4×4 matrix S , similar to the Rodrigues formula for rotations, see [12, p.67], similar relations can be found for quaternions. To minimise the number of arithmetic operations, nested multiplication should be used to evaluate these polynomials.

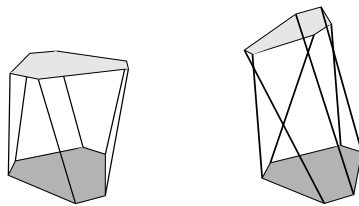


Fig. 2 Initial and final pose of the Gough-Stewart Platform for the first example in the text.

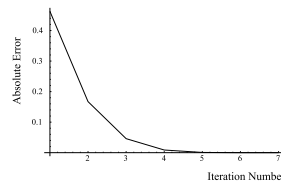


Fig. 3 Plot of the absolute error in the length leg 1 against iteration number for the second example in the text.

7 Numerical Examples

Here we study the example presented by Wang [16]. The positions of the passive spherical joints on the base and the home positions of the spherical joints on the platform are,

$$\begin{aligned} \mathbf{a}_1 &= (2 \sin(\pi/12), 2 \cos(\pi/12), 0)^T, & \mathbf{b}_1 &= (2 \sin(3\pi/12), 2 \cos(3\pi/12), 3)^T, \\ \mathbf{a}_2 &= (2 \sin(7\pi/12), 2 \cos(7\pi/12), 0)^T, & \mathbf{b}_2 &= (2 \sin(5\pi/12), 2 \cos(5\pi/12), 3)^T, \\ \mathbf{a}_3 &= (2 \sin(9\pi/12), 2 \cos(9\pi/12), 0)^T, & \mathbf{b}_3 &= (2 \sin(11\pi/12), 2 \cos(11\pi/12), 3)^T, \\ \mathbf{a}_4 &= (2 \sin(15\pi/12), 2 \cos(15\pi/12), 0)^T, & \mathbf{b}_4 &= (2 \sin(13\pi/12), 2 \cos(13\pi/12), 3)^T, \\ \mathbf{a}_5 &= (2 \sin(17\pi/12), 2 \cos(17\pi/12), 0)^T, & \mathbf{b}_5 &= (2 \sin(19\pi/12), 2 \cos(19\pi/12), 3)^T, \\ \mathbf{a}_6 &= (2 \sin(23\pi/12), 2 \cos(23\pi/12), 0)^T. & \mathbf{b}_6 &= (2 \sin(21\pi/12), 2 \cos(21\pi/12), 3)^T. \end{aligned}$$

We seek a rigid displacement for which the leg-lengths are

$$l_1 = 5.7568, l_2 = 6.6353, l_3 = 7.3836, l_4 = 7.1991, l_5 = 5.5535, l_6 = 6.2567.$$

After 5 iteration of the algorithm, using the identity as the initial value $M^{(0)}$, the result is,

$$M = \begin{pmatrix} 0.4329 & 0.6250 & -0.6495 & -1.0514 \\ -0.7500 & 0.6495 & 0.1250 & 1.6250 \\ 0.5000 & 0.4331 & 0.7500 & 2.7500 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

This position is illustrated in Fig.2. The leg-length errors, that is the difference between the desired and computed leg-lengths are, -3.5×10^{-9} , -8.8×10^{-9} , -2.9×10^{-9} , 5.6×10^{-10} , 1.2×10^{-8} , 1.8×10^{-9} respectively for the legs 1 to 6.

This computation took 0.01s running on a 2GHz Pentium 4 processor with 496MB of RAM.

We can make the algorithm work harder by setting the desired leg-lengths closer to a singularity. The manipulator will be singular if platform and base are coplanar. Setting the desired leg-lengths to,

$$l_1 = 2.84429, l_2 = 1.07787, l_3 = 2.84429, l_4 = 1.07787, l_5 = 2.84429, l_6 = 1.07787,$$

gives the rigid displacement from the home position,

$$M = \begin{pmatrix} 0.5000 & -0.8660 & 0.0000 & 0.0000 \\ 0.8660 & 0.50000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 1.0000 & -2.7000 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

after 7 iterations. A plot of the error in the length of leg 1 against the number of iterations is shown in Fig. 3, this shows the quadratic improvement in error expected of the Newton-Raphson method. Plots of the errors in the other leg-lengths are very similar. After the seven iterations the errors in all the leg-lengths 1 to 6 are respectively, 1.3×10^{-9} , 3.4×10^{-9} , 1.3×10^{-9} , 3.4×10^{-9} , 1.3×10^{-9} and 3.4×10^{-9} .

8 Conclusions

The algorithm presented here appears to be fast and reasonably robust. However some improvements may be possible.

For some designs of Gough-Stewart platform, notably the 6-3 platform, it is possible to use a semi-numerical routine to find the inverse Jacobian of the manipulator.

Rather than use the exponential map to represent errors, it might be better to use a Cayley map instead, see[13]. This is because the exponential map involves division by the rotational error, this can be small if the error is almost a pure translation.

References

1. P. Dietmaier, "The Stewart-Gough platform of general geometry can have 40 real postures", in *Advances in Robot Kinematics: Analysis and Control*, Kluwer Academic Publishers, 1998, pp. 1–10.
2. S. Egner, "Semi-Numerical Solution to 6/6-Stewart-Platform Kinematics Based on Symmetry", *Applicable Algebra in Engineering, Communication and Computing* **7**(6):449–468, 1996.
3. M.L. Husty, "An algorithm for solving the direct kinematics of general Stewart-Gough platforms", *Mechanism and Machine Theory*, **31**(4):365-379, 1996.
4. C. Innocenti and V. Parenti-Castelli. "A Novel Numerical Approach to the Closure of the 6-6 Stewart PlatformMechanism". *IEEE Fifth International Conference on Advanced Robotics*, pp. 851–855, 1991.
5. A. Iserles, H.Z. Munthe-Kaas, S.P. Nørset, and A. Zanna, "Lie-group methods", *Acta Numerica* **9**:215–365, 2000.
6. D. Lazard, "Generalized Stewart platform: how to compute with rigid motions?", in *IMACS-SC'93 Lille, France*, G. Jacob, N.E. Oussous, and S. Steinberg, editors, 1993.
7. P.R. McAree and R.W. Daniel, "A fast, robust solution to the Stewart platform forward kinematics", *J. of Robotic Systems*, **13**(7):407–427, 1996.
8. J-P. Merlet, "Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis", *Int. J. of Robotics Research*, **23**(3):221–236, 2004.
9. B. Mourain, "The 40 generic positions of a parallel robot", in *Proceedings of the 1993 International Symposium on Symbolic and Algebraic Computation, ISSAC '93, Kiev, Ukraine*, ACM Press, pp. 173–182, 1993.
10. M. Raghavan, "The Stewart platform of general geometry has 40 configurations", in *ASME Design and Automation Conf.*, **32**(2):397–402.
11. F. Ronga, and T. Vust, "Stewart platforms without computer?", in *Real Analytic and Algebraic Geometry, Proceedings of the International Conference, (Trento, 1992)*, Walter de Gruyter, pp. 196–212, 1995.
12. J.M. Selig, *Geometric Fundamentals of Robotics*. Springer Verlag, New York, 2005.
13. J.M. Selig, "Cayley Maps for SE(3)", in *Proceedings of the International Federation of Theory of Machines and Mechanisms 12th World Congress, Besançon, France*, 2007.
14. C.W. Wampler, "Forward displacement analysis of general six-in-parallel SPS (stewart) platform manipulators using soma coordinates", *Mechanism and Machine Theory*, **31**(3):331–337, 1996.
15. Li-Chun Tommy Wang and Chih Cheng Chen. "On the Numerical Kinematic Analysis of General Parallel Robotic Manipulators". *IEEE Trans. Robotics and Automation.*, **9**(3):272–285, 1993.
16. Yenfeng Wang. "A direct numerical solution to forward kinematics of general Stewart-Gough platforms" *Robotica* **25**(1):121–128, 2007.