Identifying Singularity-Free Spheres in the Position Workspace of Semi-regular Stewart Platform Manipulators

Anirban Nag, Vikranth Reddy, Saurav Agarwal and Sandipan Bandyopadhyay

Abstract This paper presents a method to compute the largest sphere inside the position-workspace of a semi-regular Stewart platform manipulator, that is free of gain-type singularities. The sphere is specific to a given orientation of the moving platform, and is centred at a designated point of interest. The computation is performed in two parts; in the first part, a Computer Algebra System (CAS) is used to derive a set of exact symbolic expressions, which are then used further in a purely numerical manner for faster computation. The method thus affords high computation speed, while retaining the exactness and generic nature of the results. The numerical results are validated against those obtained from an established numerical algebraic geometry tool, namely, `Bertini`, and are illustrated via an example.

1 Introduction

This paper presents a method for finding a sphere inside the position workspace of a semi-regular Stewart platform manipulator (SRSPM), which is free of *gain-type*[1] singularities. The singularity-free sphere (SFS) is derived for a given orientation of the moving platform, and is centred at a designated point of interest. The choice of this point is typically motivated by the intended applications of the manipulator.

[1]Gain-type singularities (also known as *type-II singularities*) occur when the forward kinematic solutions of a manipulator merge. See [1] and the references therein for more details.

A. Nag · S. Agarwal · S. Bandyopadhyay (✉)
Department of Engineering Design, Indian Institute of Technology Madras,
Chennai 600036, India
e-mail: nag.anirban16@gmail.com

S. Agarwal
e-mail: agr.saurav1@gmail.com

S. Bandyopadhyay
e-mail: sandipan@iitm.ac.in

V. Reddy
Systemantics India Pvt. Ltd., Bengaluru, India
e-mail: vikranthiit@gmail.com

J. Lenarčič and J.-P. Merlet (eds.), *Advances in Robot Kinematics 2016*,
Springer Proceedings in Advanced Robotics 4, DOI 10.1007/978-3-319-56802-7_44

The identification of such an SFS facilitates several aspects of path-planning and design of such manipulators. As the sphere describes a convex region in $\mathbb{R}^3$, it is obvious that any path consisting of line segments is free of (gain-type) singularities, so long as the end-points of the segments are *inside* the SFS. For any given manipulator, such a calculation needs to be done *only once*, for any given orientation. If such an SFS is to be identified for a range of orientations of the moving platform, then one can scan the said range (up to some desired resolution), and identify the smallest SFS, which would be free of singularities for the entire range of orientations. Identifying such an SFS forms an important part of computing the “Safe Working Zone (SWZ)” of such a manipulator, where the manipulator can operate without encountering singularities and other issues, as explained in [11]. It is, therefore, possible to think of a design algorithm, to invert the problem, and identify the geometric parameters which would allow the manipulator to be free of singularities over a desired range of orientations, and a spherical region in $\mathbb{R}^3$.

Motivated by such utilities, several attempts have been made in the recent times to obtain such an SFS, or variants of the same. Determination of the maximal SFS in the orientation workspace, parametrised by Euler angles for the Minimal Simplified Symmetric Manipulator (MSSM) has been presented by Jiang et al. [7]. Li et al. [8] have tried to solve this problem in the six-dimensional space of rigid body motions, $\mathbb{SE}(3)$, by finding a sphere that is tangential to the gain-singularity manifold in this space. The formulation, however, seems to lack mathematical rigour for several reasons, as explained below. It is well-known that $\mathbb{SE}(3)$ does not admit a bi-invariant Riemannian metric (see, e.g., [9], Corollary A.5.1, pp. 427), and hence the notion of “distance” or length in $\mathbb{SE}(3)$ is *non-unique*. Thus, the application of the Euclidean metric to define a sphere in $\mathbb{SE}(3)$ is mathematically inaccurate, given that the Euclidean metric is a bi-invariant one. Furthermore, because of the non-existence of a unique “natural/characteristic length” in $\mathbb{SE}(3)$, the results obtained by the application of this method are always subject to the choice of the assumed *characteristic length*, and have therefore limited value in any generic problem. Also, it is not clear as to how the eliminations were implemented to solve the system of equations, and the corresponding computational efforts involved are not mentioned. Finally, in the process of solution, the number of solutions is stated to be 81, which is much higher than the total-degree Bézout’s number of 27.

In this paper, the formulation adheres to the standard definition of a sphere in $\mathbb{R}^3$, and accordingly, the SFS is computed only in the position space. Thus, there is an SFS for each point in $\mathbb{SO}(3)$ which is accessible to the moving platform. The formulation is therefore free of any mathematical inaccuracies, and it renders the problem to be solvable analytically. The analytical description of the singularity manifold of an SRSPM is available in [2], which is used in this work. The formulation leads to three cubic equations in the coordinates of the point of tangency between the SFS and the singularity surface. By elimination of two of the coordinates, a univariate polynomial of degree 48 is obtained in the remaining one. It may be noted that the degree of the final polynomial is still higher than the theoretical limit of 27, but is closer to the same. The coefficients of this polynomial are computed *exactly*, via a series of intermediate expressions which are evaluated numerically in the end. Thus, the entire formulation is implemented symbolically and the final univariate polynomial expression obtained in a manner, which can be ported to any numerical programming environment like `C` or `C++`, thereby making the steps performed inside CAS a one-time procedure. The roots obtained are validated numerically, as well as against the numerical algebraic geometry (NAG) tool `Bertini` [3], and the solution are illustrated geometrically.

It may be noted that a complementary formulation of the problem is feasible, i.e., a singularity-free sphere could be identified in $\mathbb{SO}(3)$, for a given position of the end-effector. It is mathematically consistent, when the Euclidean distance is used in conjunction with the quaternion-based representation of $\mathbb{SO}(3)$ [6, 10]. However, the computations required are very demanding in this case, as the problem is defined in terms of four polynomials, one of total degree 2, and the rest of total degree 6 each, resulting in a Bézout number of 432, which puts this problem out of the scope of the present work.

The rest of the paper is organised as follows: in Sect. 2 the mathematical formulation of the problem is described, followed by the solution of the resulting equations. The results are described in Sect. 3. Finally, the paper is concluded in Sect. 4.

2 Mathematical Formulation

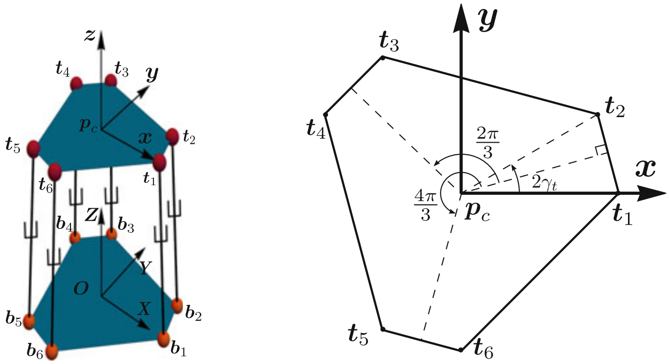
This section describes the geometry of the manipulator and the derivation of the equations describing the SFS in the position workspace of the SRSPM.

2.1 Geometry of the Manipulator

The SRSPM has *semi-regular* hexagonal top and bottom platforms, with alternate sides in each platform having equal lengths. The angular spacings between the adjacent pairs of legs are denoted by $2\gamma_t$ and $2\gamma_b$ for the top (see Fig. 1a, b) and the bottom platforms, respectively. Without any loss of generality, the radius of the circum-circle of the bottom platform is scaled to unity, thus rendering all the linear dimensions unit-less in this work. The circum-radius of the top platform is denoted by r_t. The orientation of the top platform is represented by the Rodrigue's parametrisation (see, e.g., [5], pp. 31) of $\mathbb{SO}(3)$, namely, $\{c_1, c_2, c_3\}$.

2.2 Derivation of the SFS Equations

The objective of this work is to find the largest sphere in $\mathbb{R}^3$, centred at a given point of practical interest, say, $\mathbf{p}_0 = \{x_0, y_0, z_0\}^\top$. The formulation is motivated by the observation that such a sphere would be the *smallest* among those tangential to the singularity surface in $\mathbb{R}^3$. Thus, the first and the main task is to find *all* the



(a) Schematic diagram of SRSPM. (b) Geometry of the top platform.

Fig. 1 Architecture of the SRSPM manipulator

spheres centred at $\mathbf{p}_0$, which are tangential to the singularity surface. In this case, the singularity surface is given by $f(x, y, z) = 0$, where:

$$f(x, y, z) = a_1x^2z + a_2x^2 + a_3xyz + a_4xy + a_5xz^2 + a_6xz + a_7x + a_8y^2z + a_9y^2 + a_{10}yz^2 + a_{11}yz + a_{12}y + a_{13}z^3 + a_{14}z^2 + a_{15}z + a_{16}. \quad (1)$$

The coefficients $a_i \in \mathbb{R}$ depend only on the orientation parameters c_1, c_2, c_3, and the architecture parameters γ_b, γ_t, and r_t [2]. The equation of the sphere is given by:

$$g(x, y, z) = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 - r^2 = 0, \quad (2)$$

where r is the radius of the sphere, and $\mathbf{p} = \{x, y, z\}^\top$ is the point of tangency between the sphere and the singularity surface. Therefore, at $\mathbf{p}$, the normals to these two surfaces should align (see Fig. 2), giving rise to the tangency conditions:

$$\nabla \mathbf{f} \times \nabla \mathbf{g} = \mathbf{0} \Rightarrow h_i(x, y, z) = 0, \; i = 1, 2, 3. \quad (3)$$

As only two of the equations $h_i = 0$ are linearly independent, any two of the three can be taken in combination with the equation defining the singularity surface, namely, Eq. (1), to complete the set of three equations in the three unknowns, x, y, z. Each real root of these equations leads to a sphere that is tangent to the singularity surface. The one with the smallest value of r among these is the SFS.

2.3 Solution Procedure

The degrees of h_1, h_2, h_3 in x, y, z individually are found to be {2, 3, 3}, {3, 2, 3} and {2, 2, 2}, respectively, while the total degree in x, y, and z equals 3 in each case. In

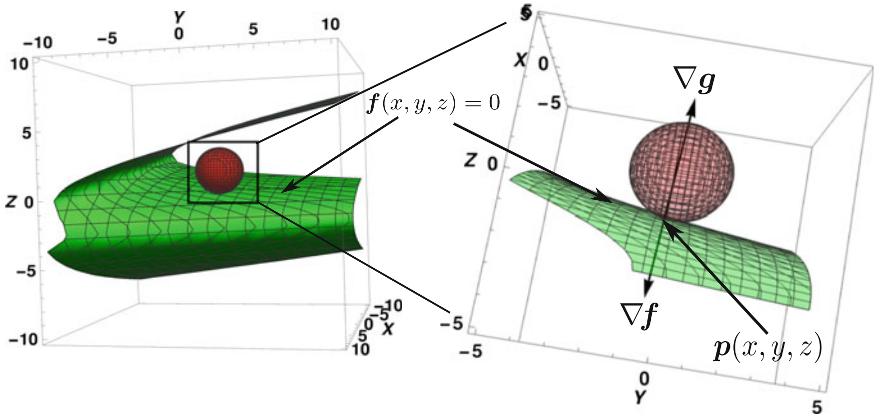


Fig. 2 Tangency of sphere with the singularity surface at the point $\mathbf{p}(x, y, z)$

view of these, $h_1 = 0$, $h_3 = 0$ are chosen for the solution process, alongside $f = 0$. From these equations, x, y are eliminated sequentially,[2] as shown schematically in Eq. (4):

$$\left.\begin{array}{r} f(x, y, z) = 0 \\ h_1(x, y, z) = 0 \\ h_3(x, y, z) = 0 \end{array}\right) \xrightarrow{\times x} \left.\begin{array}{r} g_1(y, z) = 0 \\ g_2(y, z) = 0 \end{array}\right) \xrightarrow{\times y} g_3(z) = 0. \tag{4}$$

In the above, "$\xrightarrow{\times x}$" denotes the elimination of the variable x from two or more equations in x, via computation of resultants with respect to x. The functions g_1 and g_2 have degrees $\{4, 8\}$ and $\{6, 7\}$ in y and z, respectively. However, g_2 is of the form yg_2', i.e., g_2' is of degree 5 in y. The variable y is eliminated between $g_1 = 0$ and $g_2' = 0$ (under the assumption $y \neq 0$; the case $y = 0$ is treated separately) using Bézout's method, leading to a Bézout matrix of size 5×5. Direct expansion of the determinant of this matrix leads to a polynomial in the only remaining unknown, z. However, the *size*[3] of the resulting symbolic expression is huge (about 29 GB). The time taken for expanding the determinant symbolically is about 17 min. The time taken for evaluating the determinant and the complexity of the resulting expression, makes this method computationally inefficient and practically eliminates the chance of it being used to find the SFS for a range of orientations, as a part of a larger but more relevant analysis/design problem.

In order to overcome the above-mentioned drawbacks, *a cascaded approach* was adopted to evaluate the 5×5 determinant, wherein it is expanded first in terms of five 4×4 sub-determinants, which, in turn are expanded in terms of 20 (of which

---

[2]It may be noted that many different elimination sequences are possible. The one presented here resulted in relatively smaller degrees of the intermediate and final polynomials.

[3]The "size" of an expression in this context indicates the amount of memory required to store the expression in the internal format of the computer algebra system (CAS) used, namely, `Mathematica`.

only 10 are distinct) sub-determinants of size 3 × 3. Thus the coefficients of the final univariate polynomial, $g_3(z)$, are obtained in terms of two stages of intermediate expressions. Firstly, each of the 3 × 3 determinants are obtained in closed-form, in terms of the coefficients a_j (defined in Eq. (1)):

$$\Delta_{3i} = \sum_{k=1}^{29} b_{ik}(a_j) z^{k-1}, \; i = 1, \ldots, 20; \; j = 1, \ldots, 16. \tag{5}$$

The new sets of coefficients, b_{ik}, are obtained as closed-form expressions in terms of the original coefficients, a_j. In the next step, the five 4 × 4 determinants are obtained in a similar manner, leading to the new set of coefficients c_{ik}:

$$\Delta_{4i} = \sum_{k=1}^{40} c_{ik}(b_{lm}) z^{k-1}, \; i = 1, \ldots, 5; \; l = 1, \ldots, 20; \; m = 1, \ldots, 29. \tag{6}$$

Finally, the required 5 × 5 determinant is computed in terms of the 4 × 4 determinants, and is cast as a polynomial in z:

$$\Delta_5 = \sum_{i=1}^{49} d_i(c_{jk})\, z^{i-1}, \; j = 1, \ldots, 5; \; k = 1, \ldots, 40. \tag{7}$$

Therefore, the final univariate equation in z is obtained as:

$$g_3(z) = \Delta_5 = 0. \tag{8}$$

Equation (8) is solved to find all the 48 solutions of z. The real solutions of z are used to find the values of x and y, and the radius of the desired sphere is obtained. These steps are explained with the help of a numerical example in the next section.

Symbolic expansion of the determinant of each of the 3 × 3 matrices takes an average of 0.5 s, and their original size is about 30 MB each. However, after symbolic simplification using the built-in `Mathematica` routine `Simplify`, the sizes of these determinants vary from 6.897 to 12.791 MB, with a total size of 93.158 MB (for the ten *unique* determinants). The actual coefficients of the 3 × 3 determinants are then replaced by the intermediate dummy variables (see Eq. (5)). Proceeding further, the sizes of the five 4 × 4 determinants (defined as Δ_{4i} in Eq. (6)) are found to be (in MB): 1.083, 0.971, 0.892, 1.067, and 1.267, respectively. The final determinant, Δ_5, is obtained in a similar manner.

These steps of computing the final set of coefficients d_i starting from the inputs a_l allow much faster computation (i.e., 11 s), and also leads to simpler final expressions. The univariate equation, $g_3(z) = 0$, consists of a total of 49 terms, with a cumulative size of 1.842 MB, while the largest term among these is only 160 KB in size. The comparison between the symbolic and the numeric computations of $g_3(z)$, in terms of the computational efforts and sizes of the expressions involved, are presented in Table 1.

Table 1 Comparison between the direct symbolic evaluation of the 5×5 determinant and the proposed approach. CPU specifications of the computer used: Intel(R) Core(TM) i7-4790 CPU running at a clock speed of 3.6 GHz, with 16 GB RAM

	Symbolic evaluation	Numerical evaluation
Software used	`Mathematica`, Symbolic mode	`Mathematica`, Numeric mode, with default working precision
Size of expressions	Final univariate polynomial, $g_3(z)$, (obtained by direct expansion of the 5×5 determinant): 29 GB	Final univariate polynomial $g_3(z)$ (computed following Eqs. (5)–(7)): 1.842, 100.280 MB inclusive of all intermediate expressions
Time taken	17 min and 33 s	11 s

It is important to note, that the expressions lead to the *exact* values of the final coefficients, subject only to the working precision of the numerical computation environment used. More importantly, it allows for a purely numerical implementation of the solution process (e.g., in `C` or `C++`) without either impacting the exact nature of the computation of the coefficients or restricting the computation to the symbolic computation environment of a CAS. Another point worth noting is that once the coefficients are obtained till the last level, the process need not be repeated, when the point of interest (centre of the sphere) or the architectural parameters of the SRSPM is changed. It also paves the way for computationally efficient scanning of the orientation workspace of the manipulator for finding the smallest SFS.

3 Results

A sample problem was solved in CAS `Mathematica` [12] version 10.4 using the default working precision of the system. The values of the architecture parameters are adopted from [4]: $\gamma_t = 0.0863$ rad, $\gamma_b = 0.0835$ rad, and $r_t = 0.8479$ (after scaling the base circum-radius r_b to 1). The fixed centre of the SFS is taken to be at $\mathbf{p}_0 = \{0, 0, 1.9500\}^\top$. The orientation parameters were taken to be $c_1 = 0.1013$, $c_2 = 0.0368$, and $c_3 = 0.2962$. The monic form of Eq. (8) for these inputs is given below (Fig. 3):

$$z^{48} + 4.4567 \times 10^{15} z^{47} + 3.9157 \times 10^{19} z^{46} + 8.4802 \times 10^{21} z^{45} + 8.8816 \times 10^{23} z^{44} - \cdots$$
$$+ 4.0056 \times 10^{64} z^3 + 3.2054 \times 10^{64} z^2 - 7.7684 \times 10^{64} z + 1.2071 \times 10^{64} = 0. \quad (9)$$

Bézout's limit for the number of solutions in this case was $3 \times 3 \times 3 = 27$. The higher degree of Eq. (9) indicates introduction of spurious solutions in the process of elimination of variables. Therefore, after completing the solutions with the corresponding values of x, y, the original set of equations (i.e., Eq. (4)) are used to filter out any such

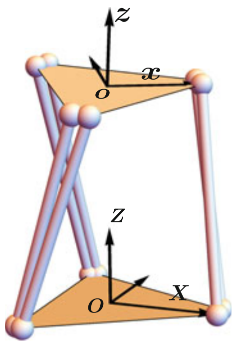
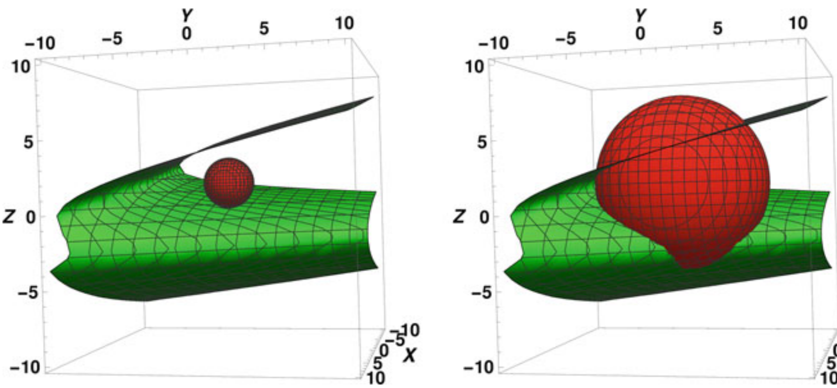


Fig. 3 Manipulator pose for the given input parameters

solutions. Only three sets of real solutions survive this test, producing residues of the order of 10^{-23}: $\{x, y, z\} = \{-0.4384, -0.3125, 0.1696\}$, $\{-0.3996, -6.4295, 2.2232\}$, $\{0.3653, -0.7859, 4.7318\}$. The corresponding values of r are: 1.8599, 6.4477, and 2.9137. Therefore, the SFS has a radius of 1.8599 for the given inputs. The actual tangency is depicted in Fig. 2. For spheres with radii greater than the minimum radius, the sphere may be tangential to the singularity surface at one point, and intersect the surface at another point, thus making them irrelevant for the purpose at hand. Figure 4a depicts the sphere with minimal radius that is tangential to the singularity surface. Figure 4b shows the sphere with radius 2.9137, which, though tangential at one point, actually cuts the singularity surface at several places.

The above solutions were obtained for the case $y \neq 0$. For $y = 0$, the obtained solutions were $\mathbf{p} = \{-66.3514, 0, 5.6010\}^\top$, and the corresponding $r = 66.4518$, which is more than the minimum radius already obtained. Hence, the above-reported



(a) $r = 1.8599$ (b) $r = 2.9137$

Fig. 4 Tangency of the minimal sphere with the singularity surface radius of the SFS holds. The residues obtained on substituting this solution set in the original set of equations was found to be of the order of 10^{-17}.

The results obtained above are prone to numerical errors, due to the high degree of the final equation in z and the huge variations in the order of magnitude of the values of the coefficients in Eq. (9). Thus, it is desirable to solve the system in Eq. (4) using another method, in order to assess the correctness of the solutions obtained. For that purpose, the NAG tool, `Bertini` [3] is used, which is well-known for its capability to compute *all* the solutions of a given polynomial system to a desired level of accuracy. As expected, `Bertini` finds only 27 solutions, of which 22 are finite, and the others escape to infinity. The real solutions match the solutions obtained above up to 10 digits after the decimal point, establishing the correctness of the solutions obtained.

4 Conclusion

A method for computing the largest gain-type singularity-free sphere inside the workspace of the SRSPM has been presented in this paper. The said sphere is a subset of the position workspace of the manipulator, and is derived for a given orientation of the moving platform. The formulation leads to three cubic equations in the coordinates of the point of tangency between the sphere and the singularity surface. A method is presented to derive a univariate equation of degree 48 from these three equations, such that all the coefficients of the intermediate as well as the final polynomials are computed *exactly*, albeit in a numerical manner. This is the main contribution of the paper, which allows, perhaps for the first time, fast computation of these spheres inside a purely numerical computation environment, without losing the accuracy of the solutions obtained. Although `Mathematica` was used to perform the numerical computations, none of the symbolic capabilities of `Mathematica` were made use of in the numerical evaluation of the coefficients.

There are existing numerical techniques, which allow the problem to be solved in a completely numerical framework. For example, Sylvester's dialytic method, leads to a matrix which has polynomial entries in a single variable. This matrix can be used to solve a generalised eigenproblem, where the eigenvalues of the system are the same as the roots obtained by solving the univariate polynomial after expanding the determinant. There exist efficient eigensolvers, which are capable of solving the problem. However, the methods being purely numerical, they have difficulties of their own. It is hard to ensure the numerical accuracy of the solutions, in particular, when fixed precision computational environments are used.

The cascaded approach presented in this paper produces the coefficients of the final univariate in their exact forms, thereby allowing accurate computations of these coefficients in a purely numerical environment. Work is in progress to implement the method presented in `C++`, so as to speed up the computations even more. Furthermore, it is intended to use this method in the computation of the SWZ of SRSPM and more general Stewart platform manipulators, as a part of their design process.

References

1. Agarwal, A., Nasa, C., Bandyopadhyay, S.: Dynamic singularity avoidance for parallel manipulators using a task-priority based control scheme. Mech. Mach. Theory **96, Part 1**, 107–126 (2016). doi:10.1016/j.mechmachtheory.2015.07.013
2. Bandyopadhyay, S., Ghosal, A.: Geometric characterization and parametric representation of the singularity manifold of a 6-6 Stewart platform manipulator. Mech. Mach. Theory **41**(11), 1377–1400 (2006)
3. Bates, D.J., Hauenstein, J.D., Sommese, A.J., Wampler, C.W.: Bertini: software for numerical algebraic geometry. http://bertini.nd.edu. doi:10.7274/R0H41PB5
4. Egner, S.: Semi-numerical solution to 6/6-Stewart-platform kinematics based on symmetry. Appl. Algebra Eng. Commun. Comput. **7**(6), 449–468 (1996). doi:10.1007/BF01293263
5. Ghosal, A.: Robotics: Fundamental Concepts and Analysis, 10th edn. Oxford University Press, New Delhi (2014)
6. Huynh, D.Q.: Metrics for 3D rotations: comparison and analysis. J. Math. Imaging Vis. **35**(2), 155–164 (2009)
7. Jiang, Q., Gosselin, C.M.: Determination of the maximal singularity-free orientation workspace for the Gough-Stewart platform. Mech. Mach. Theory **44**(6), 1281–1293 (2009)
8. Li, H., Gosselin, C.M., Richard, M.J.: Determination of the maximal singularity-free zones in the six-dimensional workspace of the general Gough-Stewart platform. Mech. Mach. Theory **42**(4), 497–511 (2007). doi:10.1016/j.mechmachtheory.2006.04.006
9. Murray, R.M., Li, Z., Sastry, S.S.: A Mathematical Introduction to Robotic Manipulation. CRC Press, Boca Raton (1994)
10. Ravani, B., Roth, B.: Motion synthesis using kinematic mappings. J. Mech. Transm. Autom. Design **105**(3), 460–467 (1983)
11. Srivatsan, R.A., Bandyopadhyay, S.: Determination of the safe working zone of a parallel manipulator. In: Thomas, F., Perez Gracia, A. (eds.) Computational Kinematics: Proceedings of the 6th International Workshop on Computational Kinematics (CK2013), pp. 201–208. Springer, Netherlands (2014). doi:10.1007/978-94-007-7214-4
12. Wolfram, Research: Mathematica, version 10.4 edn. Wolfram Research, Inc., Champaign, Illinois (2016)