A comparison of algebraic and iterative procedures for the generation of the workspace of parallel robots

Matteo RUSSO[1][0000-0002-8825-8983] and Marco CECCARELLI[2][0000-0001-9388-4391]

[1] Faculty of Engineering, University of Nottingham, Nottingham, United Kingdom
matteo.russo@nottingham.ac.uk
[2] LARM[2]: Laboratory of Robot Mechatronics, University of Rome "Tor Vergata", Rome, Italy
marco.ceccarelli@uniroma2.it

Abstract. Computing the workspace of parallel manipulators is critical to characterize their behavior. Usually, workspace is obtained by iterating the robot's forward kinematics for a discrete range of poses, resulting in a cloud of reachable points. Here we propose an alternative algebraic formulation that is based on a recursive generation of volumes for solids of revolution to evaluate each limb's workspace. The robot's workspace is then obtained as the intersection of the limbs' workspaces. With a practical example on the CaPaMan design and on a 3-UPR architecture, both the numerical and algebraic methods are discussed with their features and limits.

Keywords: Kinematics, mechanism analysis, workspace, parallel manipulators

1 Introduction

The architecture of parallel robots consists of two platforms—one fixed, one mobile—that are connected by multiple kinematic chains, called limbs, that determine their relative motion [1-2]. When compared to serial robotic architectures, which are made of a single kinematic chain from base to end-effector, parallel robots usually perform better in speed and repeatability. Furthermore, as each limb is usually actuated by a single motor, the moving masses can be limited to move with a lower inertia and higher efficiency than in the case of serial robots. However, the complex kinematics and the multiple physical constraints on the mobile platform result in a smaller workspace that can be further split into distinct working volumes by singular poses and assembly configurations [1]. These issues make the workspace of a parallel robot difficult to be represented with its geometrical shape, so that, rather than using the full reachable workspace, many researchers opt to operate a parallel robot in a subset of it that corresponds to suitable geometrical shapes, such as cylinders or spheres [3].

In general, the workspace is calculated in a recursive way, by iterating the forward kinematic problem of the robot with discretized pose parameters ranging from their minimum to maximum values. This workspace generation procedure results in a point cloud that includes the reachable poses of the robot, and as such it may be difficult for an easy geometrical interpretation unless a subset with a "familiar", easy-to-use shape (e.g. cylinder, cube, sphere) is used. Furthermore, while a good approximation of the workspace can only be obtained with a fine discretization interval, large intervals are needed for efficient computation with shorter times to solution [1-2].

An algebraic approach for the generation of the workspace of robots has been often proposed as an alternative to the popular iterative solution. This approach provides an exact solution for workspace volume and topology by generating the workspace geometrically as a solid of revolution when considering revolute motion generation [4-5]. A rich literature on the topic can be found for serial manipulators, but the application of this method to parallel robots has been limited because of the complexity in defining and modelling their boundary conditions and constraints [1-2]. Most of the applications of algebraic approaches for the determination of the workspace of parallel robots are related to planar parallel manipulators, as a generation procedure can be simplified from volumes to areas [6-8]. A solution has been also provided for a few parallel structures, such as Delta Robots [9-10], spherical manipulators [11-12], and others [13-14]. However, a general approach has not been proposed yet, as only structure-specific formulations have given solutions that are available in literature.

In this paper, the characteristics of iterative and algebraic approaches are discussed and compared. First, a general algorithm is presented to generate the workspace of a parallel robot for each of those methods. Then, two examples are reported as referring to the CaPaMan design [15-16] and a 3-UPR design [17-20] to compare the solutions and to discuss advantages and disadvantages of each of them.

2 Workspace determination

In this section, two main approaches to generate the workspace of a parallel manipulator are described: an iterative method and an algebraic one. The iterative method evaluates workspace boundaries as a point cloud of reachable configurations by discretizing the motion parameters and computing the position of the end-effector for each parameter combination. Conversely, the algebraic approach evaluates the workspace volume by generating a volume of reachable points for each limb of the robot and then obtaining the overall workspace of the robot as the intersection of such volumes.

2.1 Workspace discretization

A discrete approach for workspace evaluation can be obtained from the forward kinematic problem of a robot. The continuous range of motion $[q_{i,min}; q_{i,max}]$ of each active joint parameter q_i of actuation vector $\boldsymbol{q}$ can be discretized into a set of *n* points with an interval of Δq_i from each other, as

$$[q_{i,min}; q_{i,max}] \to \{q_{i,1}, q_{i,2}, \dots, q_{i,n}, \}, \tag{1}$$

$$q_{i,j} = q_{i,min} + \sum_{j=1}^{n}(j-1)\Delta q_i, \quad \Delta q_i = \frac{q_{i,max}-q_{i,min}}{n} \tag{2}$$

The forward kinematic problem (FKP) $\boldsymbol{x} = f_{\mathrm{FKP}}(\boldsymbol{q})$ can be used to compute the end-effector pose $\boldsymbol{x}$ corresponding to a given combination of the *m* active joint parameters. Thus, by considering all the possible point combinations given by the discretization (2) of each joint parameter, the workspace can be generated as a point cloud of n^m poses.

The exponential nature of the number of resulting poses means that the determination of a workspace with the discretization method can be extremely taxing from a computational point of view. Out of the two parameters that influence the number of loops, m is an intrinsic parameter that is defined by the robot architecture and it cannot be modified, whereas *n* is a function of the chosen discretization interval, as formulated in (2), and it can only be reduced at the expense of point density, which reduces the resolution of the workspace and risks missing critical points where the manipulator behaves singularly. However, this approach is widely used because of its simplicity as in [2], as it only requires knowledge of the forward kinematic problem of the robot.

2.2 Algebraic geometrical approach

In the algebraic geometrical approach, the workspace of a parallel manipulator can be computed by intersecting the workspace volumes that is generated by each of its limbs. This method does not require any *a priori* modelling of the manipulator but only its topological layout and constraints, in contrast with the previously described approach, which requires the FKP. The algebraic approach starts by determining the workspace Λ_i of each limb, which is obtained as a volume of revolution by considering the rotation of each link of the limb's kinematic chain around the axes of its revolute joints. For clarity, a practical example is here reported in Fig. 1 for a UPS kinematic chain (universal-prismatic-spherical joints, with the prismatic joint q_i actuated by a linear motor as per Gough-Stewart platform [1-2]). The universal and spherical joints are decomposed into a series of revolute joints (i.e., two consecutive revolute joints ϑ_{i1} and ϑ_{i2} along the x- and y-axes for the universal joint, and three consecutive revolute joints ϑ_{i3}, ϑ_{i4}, and ϑ_{i5} along the x-, y-, and z-axes respectively for the spherical joint). Thus, the limb workspace can be obtained as due to the sequence of the joint motions as

$$\Lambda_i = \mathrm{Rot_x}(\vartheta_{i1}) \cdot \mathrm{Rot_y}(\vartheta_{i2}) \cdot \mathrm{Tr_z}(q_i) \cdot \mathrm{Rot_x}(\vartheta_{i3}) \cdot \mathrm{Rot_y}(\vartheta_{i4}) \cdot \mathrm{Rot_z}(\vartheta_{i5}) \tag{3}$$

where $\mathrm{Rot_k}$ is a rotation matrix around the k-axis and $\mathrm{Tr_k}$ is a translation matrix along the k-axis. The resulting workspace is a family of spheres *S* that can be expressed as

$$S_i\colon (x - x_{0i})^2 + (y - y_{0i})^2 + (z - z_{0i})^2 = {r_i}^2 \tag{4}$$

These spheres are all centered on the extremity of the limb fixed onto the base in (x_{0i}, y_{0i}, z_{0i}) and they are characterized by variable radius r_i as family parameter, which corresponds to the length of the linear actuator in the limb and is thus related to its motion parameter and its lower and upper boundaries. By considering the actuation constraints, the limb workspace can be determined as the union of these spheres as

$$\Lambda_i = \bigcup_{r_i=r_{min}}^{r_{max}} S_i \tag{5}$$

Once the limb workspaces Λ_i have been defined, the overall workspace of the parallel manipulator Λ can be computed as their intersection [21] as

$$\Lambda = \bigcap_{i=1}^{m} \Lambda_i \tag{6}$$

When compared to the previous method, the algebraic geometrical approach is computationally efficient and provides an exact solution rather than an approximated one. However, the highly coupled behavior of different limbs of a parallel manipulator makes the implementation of self-collision constraints difficult and can result in unexpected operational failure. A first solution can represent a theoretical reachable workspace of the robot, without considering limb collisions.

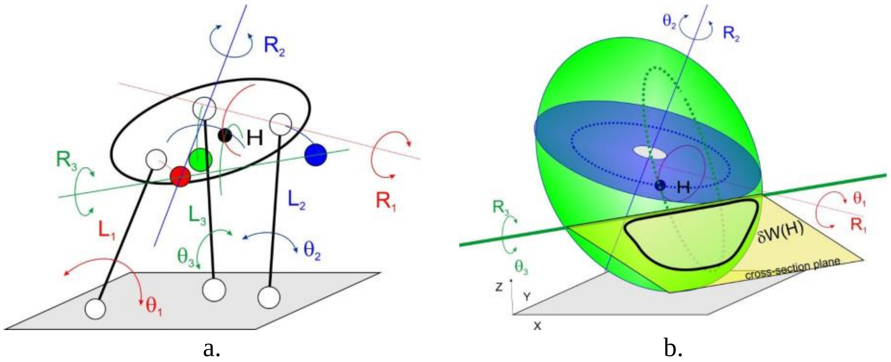


Fig. 1. Algebraic geometrical generation of workspace volume: a. A kinematic scheme with axes of rotation of the joints of each limb; b. An example workspace Λ of a parallel manipulator obtained as a solid of revolution by following the procedure in Eqs. (3-5).

3 Numerical examples

In this section, the above-mentioned methods are compared by referring to two examples of parallel manipulators: a lower-mobility mechanism that was designed for earthquake simulations, CaPaMan [15-16], and a robotic leg mechanism [17-20].

3.1 CaPaMan

The CaPaMan parallel robot has been conceived to simulate the 3D motion modes of earthquakes [16]. As shown in Fig. 2, CaPaMan is a lower-mobility manipulator with three limbs with a PaPS kinematic chain based on a four-bar parallelogram (Pa) [15]. With reference to the diagram in Fig. 2a and the formulation in [15], CaPaMan's kinematics can be written as

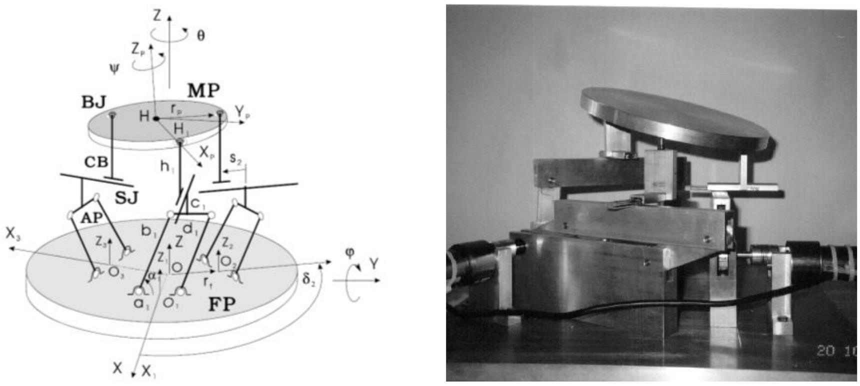
$$x = \frac{y_3-y_2}{\sqrt{3}} - \frac{r_p}{2}(1-\sin\varphi)\cos(\psi-\vartheta)\,,\ z = \frac{z_1+z_2+z_3}{3},$$
$$y = y_1 - r_p(\sin\psi\cos\vartheta + \cos\psi\sin\varphi\sin\vartheta) \qquad (7)$$

where

$$\varphi = \arccos\frac{2\sqrt{z_1^2+z_2^2+z_3^2-z_1z_2-z_2z_3-z_3z_1}}{3r_p}, \psi = \arctan\frac{\sqrt{3}(z_3-z_2)}{2z_1-z_2-z_3},$$
$$\vartheta = \arcsin\frac{2(y_1+y_2+y_3)}{3r_p(1+\sin\varphi)} - \psi, \qquad (8)$$

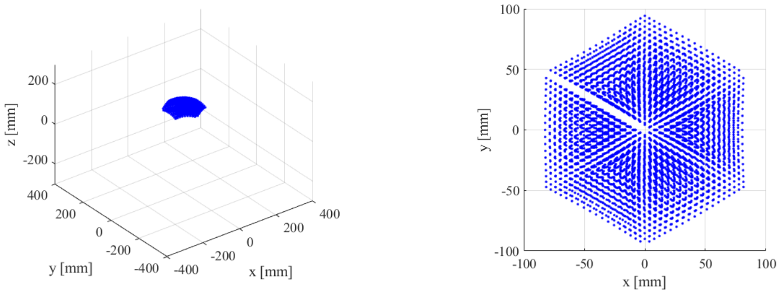
where $y_k = b_k\cos q_k$ and $z_k = b_k\sin q_k + h_k$ are function of motion variable q_k and geometrical parameters b_k and h_k.

By using (7), the workspace of CaPaMan with $b_1 = b_2 = b_3 = 100\,mm, h_1 = h_2 = h_3 = 100\,mm, r_p = 50\,mm$ has been computed by discretizing actuation variables $\{q_1, q_2, q_3\}$ with a step $\Delta q = \pi/36\,rad$, with the results shown in Fig. 3; and by using the proposed algebraic geometrical method, with results shown in Fig. 4.



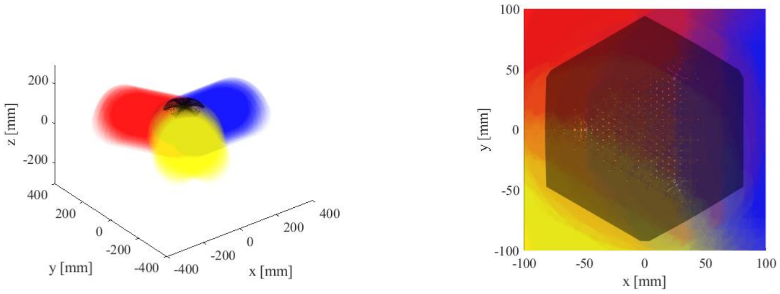
a. b.

Fig. 2. CaPaMan, Cassino Parallel Manipulator [15]: a. A kinematic scheme with main parameters; b. A prototype at LARM laboratory of University of Cassino.



a. b.

Fig. 3. Computed discretized workspace Λ of CaPaMan in Fig. 2: a. side view; b. top view.



a. b.

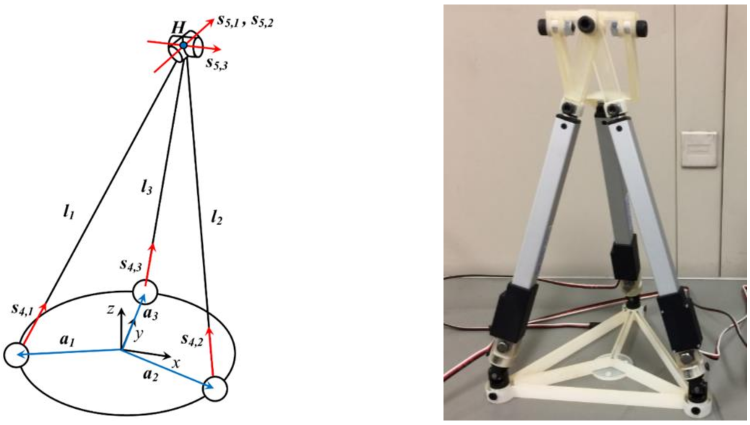
Fig. 4. Computed algebraic workspace Λ of CaPaMan in Fig. 2: a. side view; b. top view.

3.2 3-UPR mechanism

The 3-UPR parallel manipulator in this example has been designed for the locomotion system of service robots [17-20]. The mechanism design is shown in Fig. 5, and its performance is analyzed in [2, 20]. With reference to the diagram in Fig. 5a, the FKP of the system can be formulated as

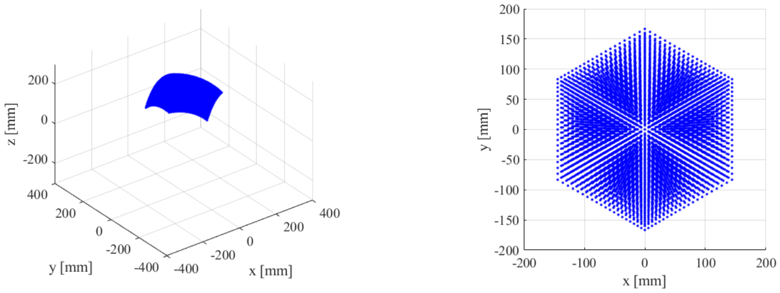
$$x = \frac{1}{2\sqrt{3}a}\left({l_1}^2 - {l_2}^2\right),\ y = \frac{1}{6a}\left({l_1}^2 + {l_2}^2 - 2{l_3}^2\right),\ z = \sqrt{{l_3}^2 - x^2 - (y-a)^2} \quad (9)$$

By using (9), the workspace of a 3-UPR mechanism with $a = 100\,mm, l_{i,min} = 200\,mm, l_{i,max} = 300\,mm$ has been computed by discretizing actuation variables $\{l_1, l_2, l_3\}$ with a step $\Delta l = 10\,mm$, with the results shown in Fig. 6; and by using the proposed algebraic geometrical method, with results shown in Fig. 7.



a. b.

Fig. 5. A 3-UPR parallel mechanism for robotic legs [20]: a. A kinematic scheme with main parameters; b. A prototype at LARM² laboratory of University of Rome Tor Vergata.



a. b.

Fig. 6. Computed discretized workspace Λ of the mechanism in Fig. 5: a. side view; b. top view.

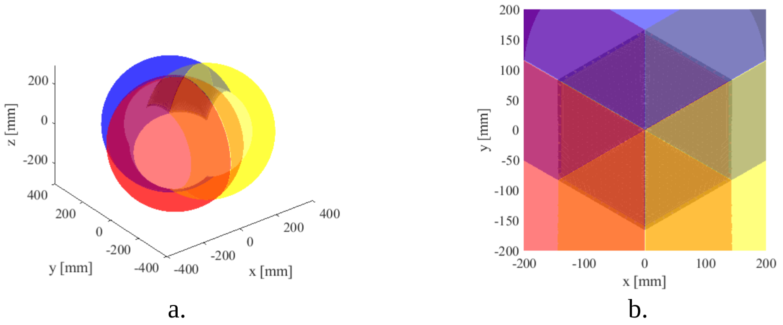


Fig. 7. Computed algebraic workspace Λ of the mechanism in Fig. 5: a. side view; b. top view.

3.3 Results and discussion

Both the proposed algebraic geometrical method and the discretized approach in the examples above have been computed in MATLAB 2021a on a Windows 10 computer with a 3.70 GHz 6-core CPU (AMD Ryzen 5 5600X). In both cases, the algebraic approach computes a general geometrical solution in less than 0.01 s, whereas the discretized workspace is obtained after 3.20 s for the CaPaMan and 0.40 s for the UPR mechanism.

Thus, the algebraic geometrical method is computationally more efficient than the discretized one and provides a solution rather than an approximated one as from the other procedure. However, the discretized workspace can include self-collision constraints that have not be implemented in the algebraic formulation up to now. Furthermore, the geometrical approach only results in the position workspace of the manipulator, without giving information on the orientation. In conclusion, the algebraic method is more precise and efficient for an evaluation and representation of the workspace, whereas the discretized workspace enables the inclusion of self-collision and can provide further information on the manipulator behavior that cannot be seen with the algebraic geometrical approach.

4 Conclusions

In this paper, two different methods for the determination of the workspace of parallel manipulators are compared: an iterative method based on the discretization of pose parameters, and a geometrical approach with an algebraic formulation. After briefly introducing the methods, advantages and disadvantages of both approaches are outlined through two examples on lower-mobility parallel manipulators with three degrees of freedom. Even if the discretized approach provides more information on the workspace, the algebraic formulation provides a faster exact solution that can be used for a fast representation of the workspace without the need of a kinematic model, as well as for an exact determination of the workspace boundaries.

References

1. Tsai L.W. (1999). *Robot analysis*. John Wiley & Sons.
2. Ceccarelli M. (2004). *Fundamentals of Mechanics of Robotic Manipulation*. Springer.
3. Russo M., Herrero S., Altuzarra O., Ceccarelli M. (2018). Kinematic analysis and multi-objective optimization of a 3-UPR parallel mechanism for a robotic leg. *Mechanism and Machine Theory 120:*192-202.
4. Ceccarelli M. (1996). A formulation for the workspace boundary of general N-revolute manipulators. *Mechanism and Machine Theory, 31*(5):637-646.
5. Gu H., & Ceccarelli M. (2011). Ring Workspace Topology of Three-Revolute Manipulators. *International Journal of Modelling and Simulation*, *31*(2):143-153.
6. Li T., & Ceccarelli M. (2011). A Geometric Approach for Workspace Determination of Planar n-DOF Parallel Manipulators. 4th Int.Congress Design Model. Mech. Systems.
7. Yahya S., Moghavvemi M., & Mohamed H.A. (2011). Geometrical approach of planar hyper-redundant manipulators: Inverse kinematics, path planning and workspace. *Simulation Modelling Practice and Theory*, *19*(1):406-422.
8. Kim D.J., Chung W.K., & Youm, Y. (1997). Geometrical approach for the workspace of 6-dof parallel manipulators. In Proc. of Int. Conf. on Robotics and Automation 4:2986-2991.
9. Liu X. J., Wang J., Oh K. K., & Kim J. (2004). A New Approach to the Design of a DELTA Robot with a Desired Workspace. *Jnl. of Intelligent and Robotic Systems*, *39*(2):209-225.
10. Di Gregorio R., & Zanforlin R. (2003). Workspace analytic determination of two similar translational parallel manipulators. *Robotica, 21*(5):555.
11. Bulca F., Angeles J., & Zsombor-Murray P.J. (1999). On the workspace determination of spherical serial and platform mechanisms. *Mechanism and Machine Theory*, *34*(3):497-512.
12. Alamdar A., Farahmand F., Behzadipour S., & Mirbagheri A. (2020). A geometrical approach for configuration and singularity analysis of a new non-symmetric 2DOF 5R Spherical Parallel Manipulator. *Mechanism and Machine Theory*, *147*:103747.
13. Hu B., Cui H., Shi D., Zhang D., Wang A., Wang Y., & Zhang Q. (2019). Reachable workspace determination for a spatial hyper-redundant manipulator formed by several parallel manipulators. *Journal of Mechanical Science and Technology, 33*(2):869-877.
14. Aboulissane B., El Haiek D., El Bakkali L., & El Bahaoui J. (2019). On the workspace optimization of parallel robots based on CAD approach. *Procedia Manuf., 32*:1085-1092.
15. Ceccarelli, M. (2013). Historical development of CaPaMan, Cassino parallel manipulator. In *New Trends in Mechanism and Machine Science* (pp. 749-757). Springer, Dordrecht.
16. Selvi Ö., & Ceccarelli M. (2012). Interpretation of earthquake effects on mechanism operation: an experimental approach. *J Naval Sci Eng*, 8(2), 31-45.
17. Ceccarelli M., Cafolla D., Russo M., & Carbone G. (2018). HeritageBot platform for service in Cultural Heritage frames. *Int. Jnl. of Adv. Robotic Systems*, *15*(4): 1729881418790692.
18. Ceccarelli, M. et al. (2017). HeritageBot service robot assisting in cultural heritage. In *2017 First IEEE International Conference on Robotic Computing (IRC)* (pp. 440-445). IEEE.
19. Russo M., Cafolla D., & Ceccarelli M. (2018). Design and experiments of a novel humanoid robot with parallel architectures. *Robotics 7(4):*79.
20. Russo M., Ceccarelli M., & Takeda Y. (2017). Force transmission and constraint analysis of a 3-SPR parallel manipulator. *Proceedings of the Institution of Mechanical Engineers Part C: Journal of Mechanical Engineering Science 232(23):*4399-4409.
21. Chkhartishvili, L. S. (2001). Volume of the intersection of three spheres. *Mathematical Notes*, *69*(3-4):421-428.