PERGAMON

Mechanism and Machine Theory 36 (2001) 15–28

MECHANISM AND MACHINE THEORY

www.elsevier.com/locate/mechmt

A new approach to orientation workspace analysis of 6-DOF parallel manipulators

Ilian A. Bonev, Jeha Ryu [*]

Department of Mechatronics, Kwangju Institute of Science and Technology, 1 Oryong-dong, Buk-ku, Kwangju 500-712, South Korea

Received 28 July 1999; accepted 29 September 1999

Abstract

This paper presents a new discretization method for the computation of the *orientation workspace* of 6-DOF parallel manipulators, defined as the set of all attainable orientations of the mobile platform about a fixed point. The method is based on the use of a modified set of Euler angles and a particular representation of the orientation workspace. In addition, the *projected orientation workspace* is introduced for use in 5-axis applications, defined as the set of possible directions of the approach vector of the mobile platform. Alternative ways of computing these two types of workspaces are also discussed. © 2000 Elsevier Science Ltd. All rights reserved.

1. Introduction

A 6-DOF fully parallel manipulator, also called a *hexapod*, consists of a *mobile platform* connected by six *links* to a *base* through, respectively, spherical and universal joints. Most commonly, the base joints are fixed on the base while the links are of variable length [1,2]. This typical design with 6 *RR$\underline{P}$S* kinematic chains will be referred to as the general parallel manipulator (GPM). There exist various other architectures of 6-DOF parallel manipulators [3].

In evaluating the performance of a parallel manipulator, much concern is given to the workspace factor. As the *complete workspace* of a 6-DOF parallel manipulator is in a six-dimensional (6-D) space for which no human representation exists, different subsets of it are usually determined. The most commonly determined subsets are the *constant-orientation*

[*] Corresponding author. Present address: Departement de Genie Mecanique, University of Laval, Que., Canada G1K 7P4. Tel.: +82-62-970-2389; fax: +82-62-970-2384.

E-mail addresses: ryu@kjist.ac.kr (J. Ryu), bonev@gmc.ulaval.ca (I.A. Bonev).

PII: S0094-114X(00)00032-X

workspace [3–5], the *reachable workspace*, and the *dexterous workspace* [6]. All of them are defined in the 3-D *position space* and are therefore easily depicted in a spatial Cartesian coordinate system. The main subset of the complete workspace that is defined in the 3-D *rotation space* is the *orientation workspace*, which is defined as the set of all attainable orientations of the mobile platform about a fixed point.

The 3-D orientation workspace is probably the most difficult one to determine and represent. Fortunately, many of the 6-DOF parallel manipulators are used for 5-axis machining operations, and thus, the user is only interested in the set of attainable directions of the *approach vector* of the mobile platform, which is the unit vector along the axisymmetric tool. We define this 2-D workspace as the *projected orientation workspace*.

Very few works exist on the topic of orientation workspace computation. The most relevant one in this area has been presented by Merlet [7], where a hybrid method is proposed for the determination of a 2-D subset of the orientation workspace of GPMs. In that method, the possible directions of a unit vector attached to the mobile platform are mapped on a unit sphere. To do so, the mobile platform is first rotated in *discrete* angles about a fixed vector $\mathbf{X}_1$. Then, the possible rotations of the mobile platform about a fixed vector $\mathbf{X}_2$ are *geometrically* investigated and subsequently mapped as circular arcs on the unit sphere. Thus, a general intersection of the orientation workspace is found. It remains unclear, however, whether there exists a choice of such vectors for which the computed intersection is exactly the projected orientation workspace. Furthermore, the method cannot be easily extended to other types of parallel manipulators as it is strongly dependent on the simplicity of the GPMs serial chains.

In this paper, we present a new discretization method for computing the orientation workspace of any parallel manipulator. The method is based on the use of a modified set of Euler angles and the particular representation of the orientation workspace in a cylindrical coordinate system as this guarantees that the orientation workspace is a single volume. Furthermore, in the case of *axisymmetric parallel manipulators*, we show that a close approximation of the projected orientation workspace can be found directly by fixing one of the Euler angles and finding an intersection of the orientation workspace.

The organization of this paper is as follows. In Section 2, we describe the kinematic constraints that limit the workspace of a GPM. Then, in Section 3, we discuss on the complex issue of representing the orientation of the mobile platform and present a modified set of Euler angles. Section 4 presents in detail the proposed discretization method used for computing the orientation workspace. Examples are provided to demonstrate the usefulness of the proposed modified set of Euler angles and the particular workspace representation. Based on the shape of the orientation workspace, Section 5 presents a simple discretization algorithm for computing an approximation of the projected orientation workspace by fixing the value of one of the Euler angles. Examples are again given to illustrate the easy interpretation of that 2-D workspace in a simple polar plot. Conclusions are made in Section 6.

2. Kinematic constraints

In this paper, we will deal only with the GPM, though the same methodology can be applied to other types of parallel manipulators [8]. An example of a GPM's is given in Fig. 1. The centers of

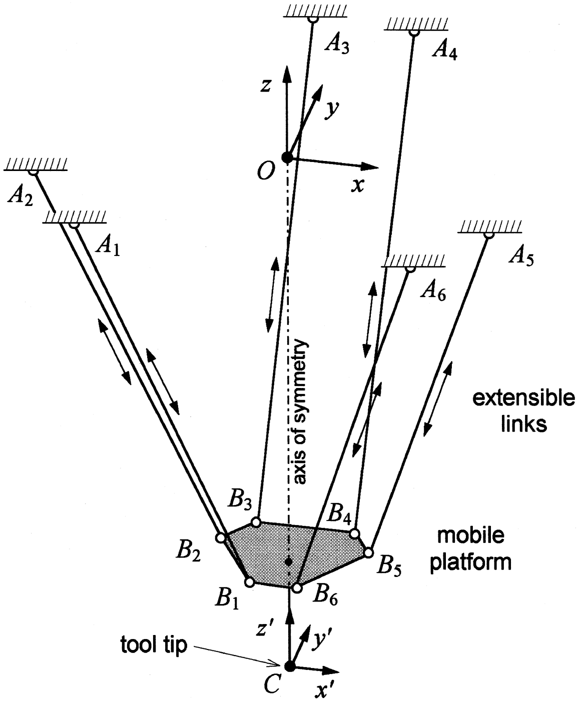


Fig. 1. A schematic of an axisymmetric GPM and the notation used.

the base universal joints are denoted by A_i, and the centers of the platform spherical joints by B_i $(i = 1..6)$. A base reference frame is selected fixed to the base at point O, with axes x, y, z, such that the base z-axis coincides with the axis of symmetry (if such exists). A mobile frame is chosen fixed to the tool tip of the mobile platform at point C, with axes x', y', z', such that the mobile z'-axis coincides with the tool axis. At the *reference orientation* of the mobile platform, the orientation of the mobile platform coincides with that of the base frame. Finally, we define the *approach vector* as the unit vector along the z'-axis of the mobile frame.

Let the orientation of the mobile platform be represented by the 3 × 3 orthogonal rotation matrix $\mathbf{R}$. For a given position (vector $\mathbf{OC}$) and orientation (matrix $\mathbf{R}$) of the mobile platform, we may compute the necessary link lengths, denoted by ℓ_i, using the following relation:

$$\ell_i = \left\|\mathbf{OC} + \mathbf{R}\mathbf{CB}'_i - \mathbf{OA}_i\right\| \quad \text{for } i = 1..6, \tag{1}$$

where vector $\mathbf{CB}'_i$ represents the coordinates of point B_i with respect to the mobile frame, and vector $\mathbf{OA}_i$ represents the coordinates of point A_i with respect to the base frame. Eq. (1) is the solution of the so-called inverse kinematics problem.

There are three main mechanical constraints that limit the workspace of a GPM: (i) the actuators' stroke, (ii) the range of the passive joints, and (iii) the link interference.

2.1. Actuators' stroke

The limited stroke of actuator i imposes a length constraint on link i, such that

$$\ell_{i,\min} \leqslant \ell_i \leqslant \ell_{i,\max} \quad \text{for } i = 1..6, \tag{2}$$

where $\ell_{i,\min}$ and $\ell_{i,\max}$ are, respectively, the minimum and maximum lengths of link i.

2.2. Range of the passive joints

Each passive joint has a limited range of motion. Let $\mathbf{j}_{A_i}$ be the unit vector with respect to the base frame that is along the axis of symmetry of the universal joint at point A_i. Let the maximum misalignment angle of that joint be α_i. Let also the unit vector along link i be denoted by $\mathbf{n}_i$, i.e., $\mathbf{n}_i = \mathbf{A}_i\mathbf{B}_i/\ell_i$. Then, the limits on base joint i impose a constraint, such that

$$\cos^{-1}\left(\mathbf{j}_{A_i}^{\mathrm{T}}\mathbf{n}_i\right) \leqslant \alpha_i \quad \text{for } i = 1..6. \tag{3}$$

Similarly, let $\mathbf{j}'_{B_i}$ be the unit vector with respect to the mobile frame that is along the axis of symmetry of the spherical joint at point B_i. Let vector $\mathbf{j}_{B_i}$ be the opposite vector, and with respect to the base frame, i.e., $\mathbf{j}_{B_i} = -\mathbf{R}\mathbf{j}'_{B_i}$. Let the maximum misalignment angle of that joint be β_i. Then, the limits on mobile platform joint i impose a constraint, such that

$$\cos^{-1}\left(-\mathbf{j}_{B_i}^{\mathrm{T}}\mathbf{n}_i\right) \leqslant \beta_i \quad \text{for } i = 1..6. \tag{4}$$

2.3. Link interference

Let us assume that the links can be approximated by cylinders of diameter D. This imposes a constraint on the relative position of all pairs of links, such that

$$\text{distance}(A_iB_i, A_jB_j) \geqslant D \quad \text{for } i = 1..6, \quad j = (i+1)..6 \tag{5}$$

or the minimum distance between every two line segments corresponding to the links of the parallel manipulator should be greater than or equal to D. The minimum distance between two line segments is not given by a simple formula but can be obtained through the application of a multi-step algorithm. Due to space limitations, we will not present that algorithm here but refer the reader to the well-detailed one given in [2].

2.4. Compatibility constraint

In general, a given configuration of the parallel manipulator may satisfy all the constraints given by Eqs. (2)–(5) and still be unattainable from the initial assembly of the manipulator. In other words, the configuration may be *incompatible with the initial assembly configuration*. A configuration is *compatible* if and only if it can be reached through a continuous motion starting from the initial assembly configuration, satisfying the constraints given by Eqs. (2)–(5). To the best of our knowledge, no direct concern has been given to this compatibility constraint by authors applying discretization methods for workspace evaluation (e.g. [1,2]).

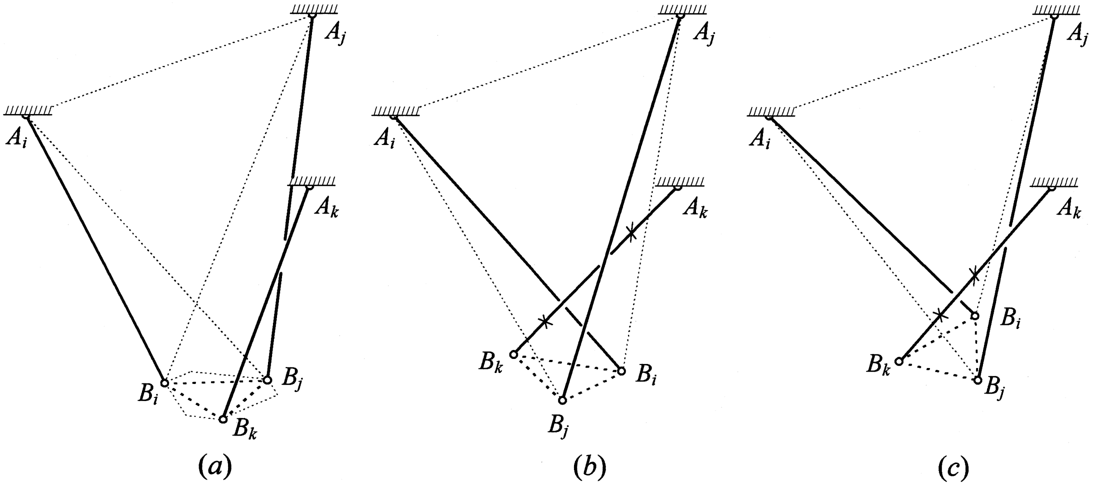


Fig. 2. A triplet of links in (a) a configuration compatible with the initial assembly, (b) an incompatible configuration (A_kB_k intersects $\Delta A_jB_iB_j$ and $\Delta A_iB_iB_j$), and again (c) a compatible configuration (A_kB_k intersects $\Delta A_iB_iB_j$ and $\Delta A_iA_jB_j$).

Unfortunately, it is not easy to devise a general algorithm for performing the compatibility check for any type of parallel manipulator. However, to illustrate the problem, we propose a simple geometric algorithm for the compatibility check of axisymmetric parallel manipulators. The algorithm works by performing a simple geometrical check on each triplet of links.

Fig. 2(a) shows a triplet of links of an axisymmetric parallel manipulator in a configuration, compatible with the initial assembly. If we assume that the mobile platform cannot attain tilt angles greater than 90°, then it is easy to see that the configuration of the sub-mechanism shown in Fig. 2(b) is incompatible with the initial assembly. The sub-mechanism is in an incompatible configuration if and only if one of the line segments, say A_kB_k, intersects two of the four planar triangular regions formed by the other four joint centers A_i B_i A_j and B_j ($\Delta A_iB_iB_j$ $\Delta A_jB_iB_j$ $\Delta A_iB_iA_j$ $\Delta A_iA_jB_j$). Indeed, in Fig. 2(b), A_kB_k intersects $\Delta A_jB_iB_j$ and $\Delta A_iB_iB_j$. Note, however, that the configuration in Fig. 2(c) is not incompatible, though A_kB_k intersects $\Delta A_iB_iB_j$ and $\Delta A_iA_jB_j$. Thus, for an incompatible configuration, the common line to the two intersected triangular regions should not be A_iB_j or A_jB_i.

Although, the basic check (intersection of line segment with planar triangular region) is not time consuming, this check has to be performed for each of the 4 planes with each of the three links in all 20 sub-mechanisms, or a total of 240 times. Therefore, an optimum discretization algorithm should eliminate the need for the application of the compatibility check.

3. Orientation representation

One of the basic problems in finding the 3-D orientation workspace is the choice of coordinates to describe the orientation of the mobile platform. Various redundant sets of orientation coordinates exist, such as Euler parameters [9], direction cosines, etc. While they provide a global parameterization of the orientation, they call for a representation in at least a 4-D space. To overcome this drawback, three Euler angles can be used to represent the mobile platform orientation. These angles correspond to three or more successive rotations about the base frame axes. Their main disadvantage is the existence of singularities at which the one-to-one correspondence between the actual orientation and the Euler angles does not hold.

Various types of Euler angles exist but they are all difficult to interpret in the general case. Let the two Euler angles, ϕ and θ, determine the direction of the approach vector while the third Euler angle, ψ, referred to as the *roll angle*, correspond to the last rotation about the mobile z'-axis. Probably the most intuitive choice for the Euler angles ϕ and θ is the one corresponding to the *azimuth* and *zenith* angles that define the ray direction in a spherical coordinate system. Further, in machining operations, the angle θ will also correspond to the *tilt* (*swivel*) *angle*, which is the angle between the tool approach vector and the base z-axis.

Such Euler angles are the *standard Euler angles*, defined by first rotating the mobile frame about the base z-axis by an angle ϕ, then about the mobile y'-axis by an angle θ, and finally about the mobile z'-axis by an angle ψ [10]. For this choice of Euler angles, the singularity occurs at $\theta = 0°$ and the rotation matrix is defined as

$$\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_{y'}(\theta)\mathbf{R}_{z'}(\psi) = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi), \tag{6}$$

where $\mathbf{R}_z(\cdot)$ and $\mathbf{R}_y(\cdot)$ are the basic rotation matrices. Note that the standard Euler angles can also be defined by first rotating the mobile frame about the base z-axis by an angle ψ, then about the base y-axis by an angle θ, and finally about the base z-axis by an angle ϕ.

Imagine now that the tool that is along the mobile z'-axis is asymmetric (e.g. a gripper or a carving tool) and thus, the user should be able to interpret easily not only the ϕ and θ angles but also the ψ angle. Let the *sliding vector* be the vector along the direction of the finger motion as the gripper closes or opens (in the case of a gripper), or the vector along the tool cutting edge (in the case of a carving tool), and let this vector be parallel to the mobile y'-axis. Then with the standard Euler angles, the user will know that for any approach vector, a zero roll angle means that the sliding vector is parallel to the base xy-plane. Therefore, in such 6-axis applications, the standard Euler angles will be the most suitable for interpretation.

However, the standard Euler angles have a serious representational degeneracy at the singularity $\theta = 0°$, where the range of the attainable roll angles cannot be evaluated. Indeed, at $\theta = 0°$, if at least one orientation exists represented by the rotation matrix $\mathbf{R}(\phi_0, 0°, \psi_0)$, then it would seem that all roll angles can be attained, since $\mathbf{R}(\phi_0 - \Delta\psi, 0°, \psi_0 + \Delta\psi) \equiv \mathbf{R}(\phi_0, 0°, \psi_0)$, where $\Delta\psi \in [-180°, 180°)$. Thus, although the standard Euler angles are easier to interpret when $\theta \neq 0°$, they are not suitable for representing the orientation workspace.

To overcome this problem, we introduce a *modified set of Euler angles*, which to the best of our knowledge has never been used in relation to parallel manipulators. In this new orientation representation, we first rotate the mobile platform about the base z-axis by an angle $-\phi$, then about the base y-axis by an angle θ, then about the base z-axis by an angle ϕ, and finally about the mobile z'-axis by an angle ψ (Fig. 3). The rotation matrix in this case is:

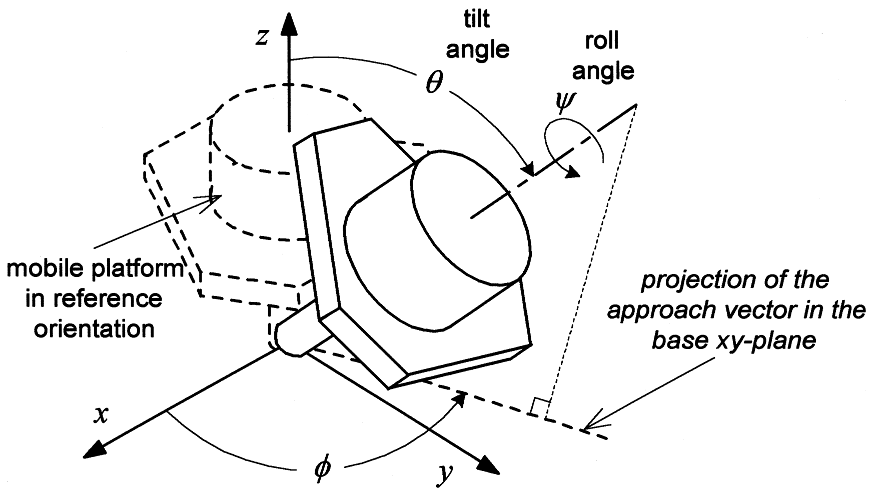

z
tilt angle
θ
roll angle
ψ
mobile platform in reference orientation
projection of the approach vector in the base xy-plane
x
φ
y

Fig. 3. The modified Euler angles defining the orientation of the mobile platform.

$$\begin{aligned}
\mathbf{R} &= \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(-\phi)\mathbf{R}_{z'}(\psi) = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi-\phi) \\
&= \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos(\psi-\phi) & -\sin(\psi-\phi) & 0 \\ \sin(\psi-\phi) & \cos(\psi-\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\phi\cos\theta\cos(\psi-\phi) - \sin\phi\sin(\psi-\phi) & -\cos\phi\cos\theta\sin(\psi-\phi) - \sin\phi\cos(\psi-\phi) & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos(\psi-\phi) + \cos\phi\cos(\psi-\phi) & -\sin\phi\cos\theta\sin(\psi-\phi) + \cos\phi\cos(\psi-\phi) & \sin\phi\sin\theta \\ -\sin\theta\cos(\psi-\phi) & \sin\theta\sin(\psi-\phi) & \cos\theta \end{bmatrix}.
\end{aligned} \tag{7}$$

Note that for a zero roll angle, the mobile platform is simply tilted (rotated) about an axis passing through the mobile frame center, parallel to the base xy-frame, and making an angle ϕ with the base y-axis. Also, at the singularity $\theta = 0°, \mathbf{R}(\phi_0 + \Delta\phi, 0°, \psi_0) \equiv \mathbf{R}(\phi_0, 0°, \psi_0)$, where $\Delta\phi \in [0°, 360°)$, and therefore, the range of attainable roll angles at $\theta = 0°$ can be correctly represented. As we see from Eq. (7), the relationship between the modified Euler angles and the standard ones is very simple – if the triplet (ϕ, θ, ψ) defines a given orientation in the modified Euler angles, then the same orientation is defined in the standard Euler angles by $(\phi, \theta, \psi - \phi)$. As we will see in the next section, the modified Euler angles allow us to represent the orientation workspace of most parallel manipulators as a simple shaped volume.

4. Orientation workspace

Now that we have selected the set of Euler angles for representing the orientation of the mobile platform, it remains to determine the way to represent the orientation workspace. With the selected set of Euler angles, the maximum range of orientations will be $\phi \in [-180°, 180°)$, $\theta \in [0°, 180°]$, and $\psi \in [-180°, 180°)$, since $\mathbf{R}(\phi, -\theta, \psi) \equiv \mathbf{R}(\phi, \theta, \psi \pm 180°)$. Three alternatives exist for representing the orientation workspace. The first one is to represent it in a Cartesian

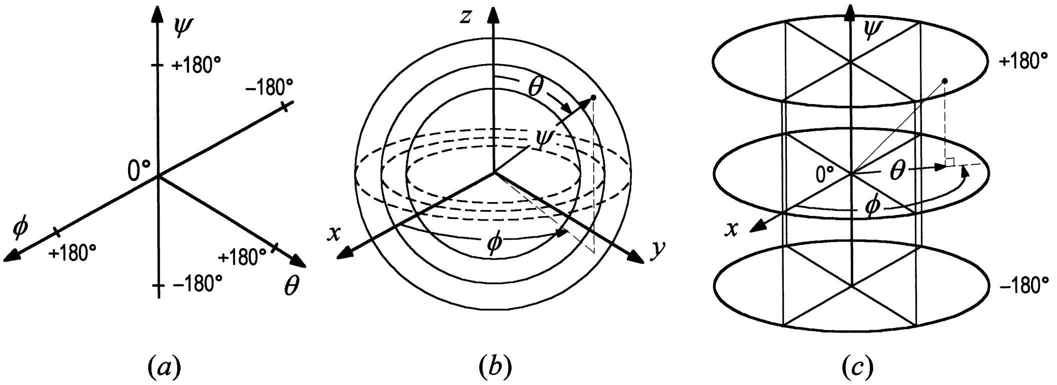


Fig. 4. The three possible representations of the orientation workspace.

coordinate system whose axes are the three Euler angles, Fig. 4(a). Such a representation is difficult to interpret and is degenerate at the plane $\theta = 0°$, corresponding to a singularity. The second alternative is to represent the orientation workspace in a spherical coordinate system where ϕ and θ are exactly the azimuth and zenith angles, Fig. 4(b). The length of the ray will correspond to ψ, being $\rho_0 - \Delta\rho$ for $\psi = -180°$, and $\rho_0 + \Delta\rho$ for $\psi = +180°$. Thus, the orientation workspace will be inside the spherical shell enclosed between the two spheres with radii $\rho_0 - \Delta\rho$ and $\rho_0 + \Delta\rho$, centered at the coordinate system origin. The final alternative is to represent the orientation workspace in a cylindrical coordinate system, where ϕ and θ are exactly the polar coordinates and ψ is the z-coordinate, Fig. 4(c).

Both the second and the third representations do not pose any problems at the singularity $\theta = 0°$ and are relatively easy to interpret. This is because both representations have the same representational singularity as the modified Euler angles – at $\theta = 0°$, the value of ϕ is irrelevant.

The second representation is easier to implement. Simply, discretize the range of ϕ and θ, and for each pair, start to increment the ψ angle from $-180°$ to $+180°$. At each step, solve the inverse kinematics by applying Eq. (1) and check all constraints defined by Eqs. (2)–(5). The first orientation for which all constraints are satisfied is stored in a double array and the next orientation for which a constraint becomes violated is stored in a second double array. The first array will define the inner boundary of the workspace while the second array will define the outer boundary of the workspace. The problem with this representation is that the inner boundary of the orientation workspace becomes hidden if the maximum tilt angles are close to or more than 90°. Furthermore, as we already discussed about the compatibility constraint, it will not be certain that for a given pair of angles ϕ and θ, there will be only one change from violated to all-satisfied constraints and one from all-satisfied to violated constraints.

To avoid these shortcomings, we choose the third type of representation shown in Fig. 4(c). In this representation, the orientation workspace is a single volume with no voids and no hidden regions. In addition, the projection of the orientation workspace onto a plane $\psi = \text{const}$ is exactly the projected orientation workspace. The most important property, however, is that the need for the compatibility check is eliminated since we always start the search from a configuration, which is compatible with the initial assembly. We propose the following discretization algorithm for determining and representing the orientation workspace.

4.1. Algorithm for the orientation workspace

4.1.1. Phase I: upper part of the orientation workspace

S1. Initialize double arrays $\mathbf{W}_{\phi,u}$ and $\mathbf{W}_{\theta,u}$, with dimensions $(n_\psi/2+1)\times n_\phi$, where $(n_\psi+1)$ is the (odd) number of equally spaced planes $\psi = \text{const}$ between $\psi=-180°$ and $\psi=+180°$ at which the workspace will be computed, and n_ϕ is the number of points to be computed at each plane $\psi = \text{const}$. These arrays will store, respectively, the values of ϕ and θ for the points defining the upper part of the workspace boundary.

S2. Set $\psi = 0°$. Assume that $(\phi_c, \theta_c) = (0°, 0°)$ is the center of the horizontal cross-section of the workspace for $\psi = 0°$.

S3. For the current ψ, construct a polar coordinate system at (ϕ_c, θ_c). Starting at n_ϕ equally spaced angles, increment the polar ray, solve the inverse kinematics, and apply the constraint checks defined by Eqs. (2)–(5) until a constraint is violated. The values for ϕ and θ at the point of constraint violation are written into the two double arrays.

S4. Compute the geometric center (ϕ_c, θ_c) of the workspace cross-section, which will serve as the assumed center for the next cross-section. If $\psi = 0°$, store the geometric center and repeat only once step 3 with that new geometric center and then jump to step 5.

S5. Set $\psi = \psi + 360°/n_\psi$.

S6. Repeat steps 3–5 until ψ becomes greater than 180° or the last horizontal cross-section of the orientation workspace is a single point (i.e., $\psi_{\max}$ is reached).

S7. Set $\psi_{\max} = \psi - 360°/n_\psi$.

4.1.2. Phase II: lower part of the orientation workspace

S8. Initialize double arrays $\mathbf{W}_{\phi,l}$ and $\mathbf{W}_{\theta,l}$, with dimensions $(n_\psi/2)\times n_\phi$.

S9. Set $\psi = -360°/n_\psi$. Assign to (ϕ_c, θ_c) the values that were stored in step 4 for $\psi = 0°$.

S10. Perform the same as in step 3.

S11. Compute the geometric center (ϕ_c, θ_c) of the workspace cross-section, which will serve as the assumed center for the next cross-section.

S12. Set $\psi = \psi - 360°/n_\psi$.

S13. Repeat steps 10–12 until ψ becomes less than −180° or the last horizontal cross-section of the orientation workspace is a single point (i.e., $\psi_{\min}$ is reached).

S14. Set $\psi_{\min} = \psi + 360°/n_\psi$.

4.1.3. Phase III: postprocessing and plotting

S15. Transfer the values from $\mathbf{W}_{\phi,u}$ and $\mathbf{W}_{\phi,l}$ to $\mathbf{W}_\phi$, and from $\mathbf{W}_{\theta,u}$ and $\mathbf{W}_{\theta,l}$ to $\mathbf{W}_\theta$, which are double arrays of dimension $N_\psi \times n_\phi$, where $N_\psi = (\psi_{\max} - \psi_{\min})/(360/n_\psi) + 1$.

S16. Transfer $\mathbf{W}_\phi$ and $\mathbf{W}_\theta$ into $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, so that $\mathbf{X}[i,j] = \mathbf{W}_\theta[i,j]\cos(\mathbf{W}_\phi[i,j])$, $\mathbf{Y}[i,j] = \mathbf{W}_\theta[i,j]\sin(\mathbf{W}_\phi[i,j])$, and $\mathbf{Z}[i,j] = \psi_{\max} - (i-1)(360/n_\psi)$, where $i = 1..N_\psi$ and $j = 1..n_\phi$, and $\mathbf{X}[i,j] = \mathbf{X}[i,1]$, $\mathbf{Y}[i,j] = \mathbf{Y}[i,1]$, and $\mathbf{Z}[i,j] = \mathbf{Z}[i,1]$, where $i = 1..N_\psi$ and $j = n_\phi + 1$.

S17. Plot the closed surface whose nodes are defined in the double arrays **X**, **Y**, and **Z**.

The proposed algorithm was implemented in MATLAB for the GPM whose data are given in Appendix A. In our implementation, $n_\psi = 180$ and $n_\phi = 120$. Two examples are presented here. In the first (Fig. 5), the orientation workspace is computed for a position at which point C lies on the axis of symmetry of the GPM. Consequently, we may observe in Fig. 5(b) the symmetrical shape

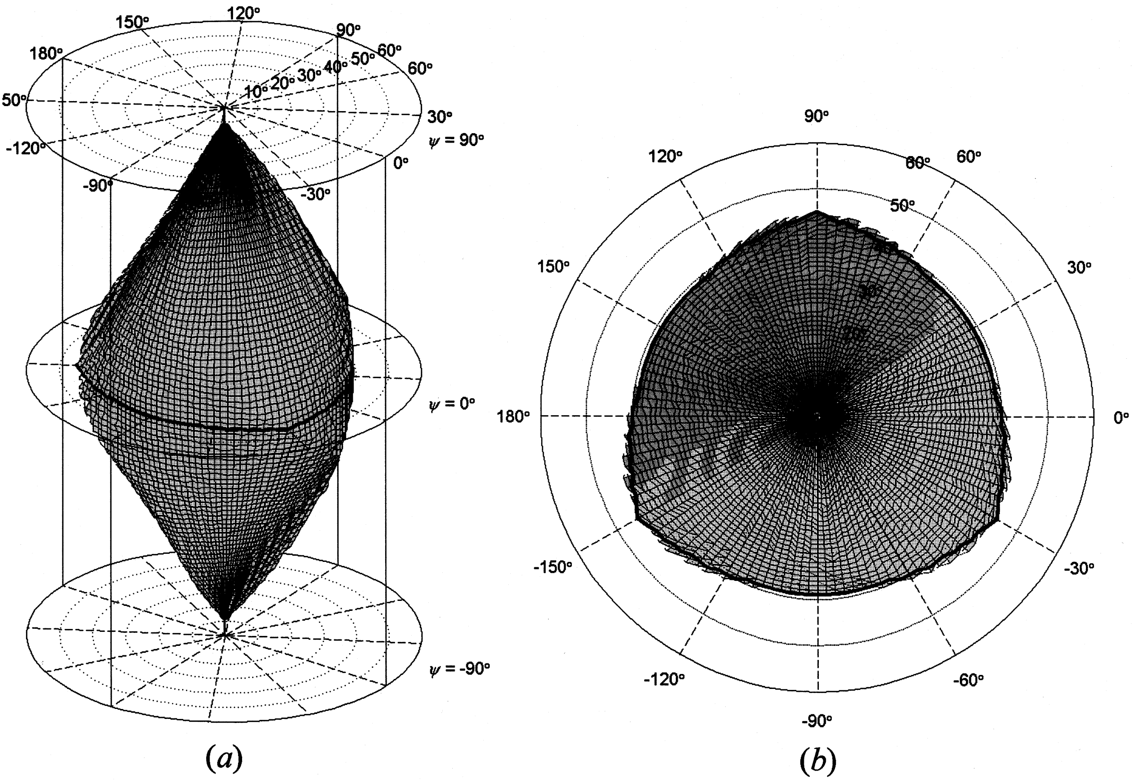


Fig. 5. (a) Isometric and (b) top views of the GPM's orientation workspace at the position $\mathbf{OC} = [0, 0, -1300]^T$ for which $\psi_{\max} = 84°$ and $\psi_{\min} = -84°$.

of the orientation workspace with respect to the axis $\theta = 0°$. In the second example (Fig. 6), the orientation workspace is computed for a position at which point C is far from the axis of symmetry of the GPM and near the boundary of the constant-orientation workspace for the reference orientation. Note, correspondingly, how the axis of the orientation workspace is shifted away from the axis $\theta = 0°$.

The computation time of the proposed method was established at about 40 min on a 350 MHz Pentium II based PC with 256 Mb RAM. It was observed that more than 70% of the computation time goes for the link interference check. On the other hand, for this GPM as well as for other parallel manipulators [8], it was observed that the main constraint that is violated is the one on the range of the platform joints, i.e., Eq. (4). In fact, link interference was never encountered. Thus, for some parallel manipulators, the link interference check can be disabled, resulting in a great reduction of the computation time.

5. Projected orientation workspace

The projected orientation workspaces for the examples given in Figs. 5 and 6 are, respectively, shown in Figs. 5(b) and 6(b) (the top views of the orientation workspaces). In other words, the

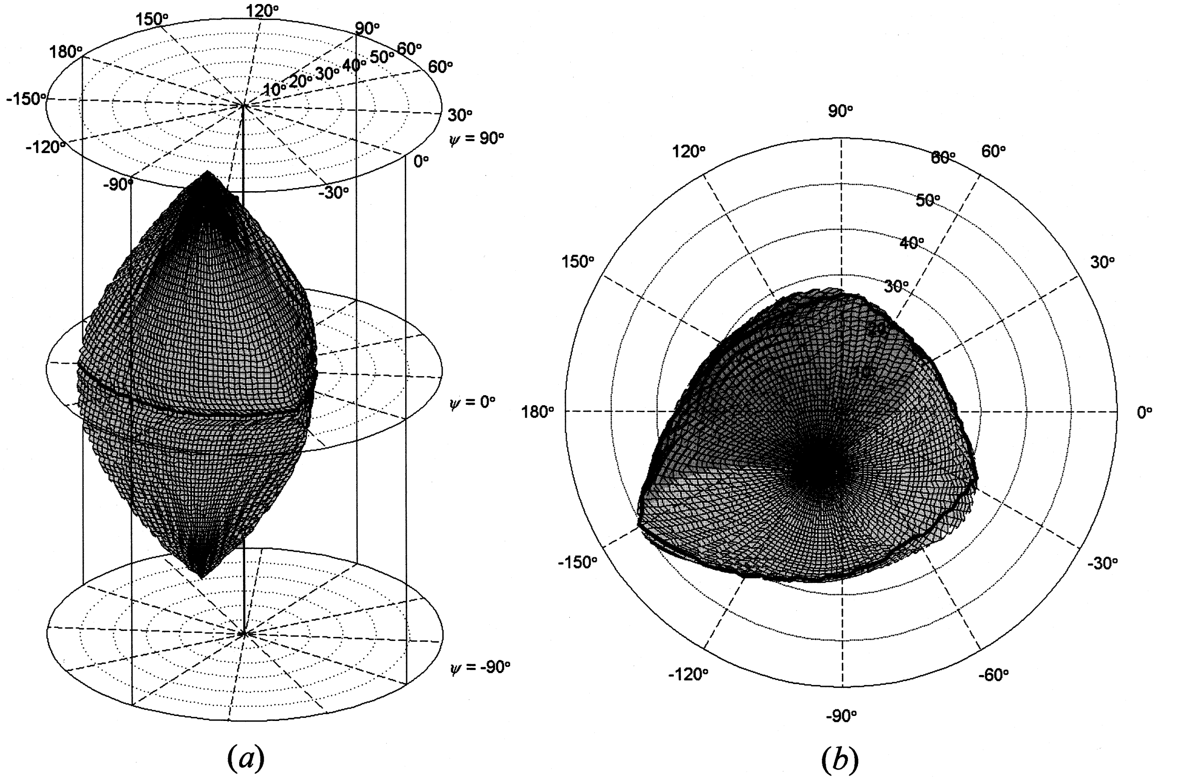


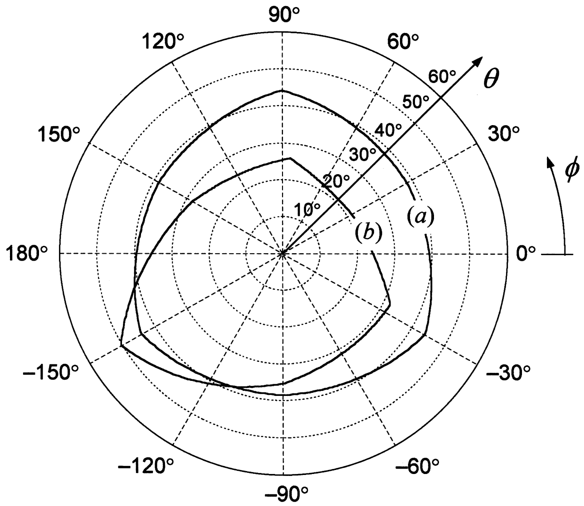
Fig. 6. (a) Isometric and (b) top views of the GPM's orientation workspace at the position $\mathbf{OC} = [200, 250, -950]^{\mathrm{T}}$ for which $\psi_{\max} = 72^\circ$ and $\psi_{\min} = -72^\circ$.

projected orientation workspace can be found by first finding the orientation workspace. As we saw in the previous section, however, the computation of the orientation workspace is a complex and time-consuming task and is often of no direct interest. Thus, for some applications, it would be beneficial to find directly the projected orientation workspace.

Now, observe again Figs. 5(b) and 6(b). The thick curves that may be seen there are the cross-sections of the boundary of the orientation workspace for $\psi = 0^\circ$. It was observed that those curves give a very good approximation to the projected orientation workspace when point C is located near the vertical axis of symmetry of the parallel manipulator, Fig. 5(b), and a fair one when it is far from it, Fig. 6(b). Therefore, with the assumption that the reference orientation is inside the orientation workspace we may propose the following discretization algorithm for the computation of the *approximated* projected orientation workspace.

5.1. Algorithm for the (approximated) projected orientation workspace

S1. Initialize the array $\mathbf{W}_\theta$ with length $(n_\phi + 1)$, where n_ϕ is the number of points to be computed to define the boundary of the projected orientation workspace. This array will store the values of θ for each discrete value of ϕ.



	$\theta_{\min}$	$\theta_{\max}$
(*a*)	38.4°	44.1°
(*b*)	21.0°	50.3°

Fig. 7. Approximation of the projected orientation workspace of the GPM for the positions (a) $\mathbf{OC} = [0, 0, -1300]^T$ and (b) $\mathbf{OC} = [200, 250, -950]^T$.

S2. Set $\phi = 0°, i = 1$, and $\theta = 0°$.

S3. Set $\theta = \theta + \Delta\theta$, where $\Delta\theta$ is the discretization step.

S4. Solve the inverse kinematics problem and check all constraints given by Eqs. (2) to (5).

S5. Repeat steps 3–4, until θ becomes 180° or a constraint is violated.

S6. Set $\mathbf{W}_\theta[i] = \theta$. Set $i = i + 1$ and $\phi = (i - 1)(360°/n_\phi)$.

S7. Set $\theta = \theta - m\Delta\theta$, where m is the number of search steps to go back.

S8. Repeat steps 3–7 until i becomes equal to $n_\phi + 1$.

S9. Set $\mathbf{W}_\theta[n_\phi + 1] = \mathbf{W}_\theta[1]$.

S10. Draw a polar plot with $\mathbf{W}_\theta$ defining the length of the ray at the angles $0°, 360°/n_\phi,\ 2(360°/n_\phi), \ldots, 360°$.

The proposed algorithm was again implemented in MATLAB. Fig. 7 shows the approximated projected workspaces for the same positions as in the examples for the orientation workspace. In this implementation, $n_\phi = 360$, $\Delta\theta = 0.1°$, and $m = 5$. These values guarantee a very smooth curve defining the projected workspace, while still the computation time (including link interference check) is quite small – about 20 s on the same PC. The computation time can be further reduced by implementing a more sophisticated search procedure for determining the first point of the workspace boundary, i.e., at $\phi = 0°$ (e.g. a bisection method).

One point to note is that the proposed algorithm works well only for axisymmetric parallel manipulators. Such parallel manipulators are, for example, most motion simulators, as well as a number of commercial hexapod machines. We also make the assumption that the reference orientation is inside the orientation workspace, or equivalently, that point C is inside the constant-orientation workspace for the reference orientation. Thus, another more general approach should be sought for the parallel manipulators with no axial symmetry.

Let us note that Merlet's algorithm [7] can also be used with the proposed modified set of Euler angles to compute the same approximation of the projected orientation workspace as the one

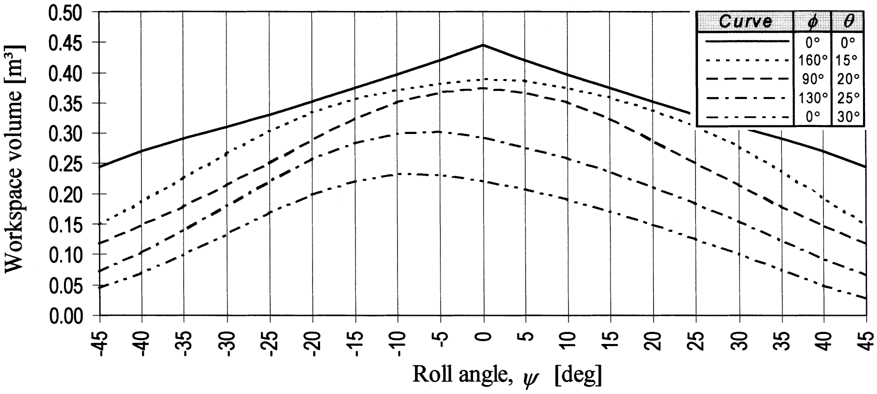


Fig. 8. The volume of the constant-orientation workspace as a function of the roll angle ψ for several directions of the approach vector.

found by our discretization method. For that purpose, the angle ϕ is discretized in the range $[0^\circ, +180^\circ)$, and then for each value, Merlet's algorithm is applied to obtain geometrically the range of the tilt angle θ. In other words, his algorithm is applied to compute the possible rotations of the mobile platform about all horizontal lines passing through point C. Note, however, that his algorithm is much more difficult to implement and we do not expect it to be faster than the proposed simple discretization method.

Finally, let us introduce another interesting property of the modified Euler angles. We implemented the geometrical approach presented in [4] for the computation of the constant-orientation workspace of GPMs (excluding the link interference modeling). The implementation was made in the CAD/CAM system CATIA, similar to the one described in [5]. It was observed that for a fixed direction of the approach vector (ϕ, θ), the volume of the constant-orientation workspace is greatest near $\psi = 0^\circ$ (Fig. 8).

6. Conclusions

A general discretization algorithm for computing the three-dimensional orientation workspace was presented in this paper. The algorithm was based on a set of modified Euler angles and a particular representation of the orientation workspace. The two-dimensional projected orientation workspace was clearly defined and a simple discretization algorithm was introduced for computing an approximation of it in the case of axisymmetric parallel manipulators.

While we introduced a general discussion on the complex issue of orientation workspace of parallel manipulators, our main contribution is for the analysis of those parallel manipulators with an axis of symmetry, used as 5-axis machining centers. The users of such hexapods can take full advantage of the application of the method proposed for the computation of an approximation of the projected orientation workspace. In addition, the proposed modified Euler angles and their property may eliminate the need for complicated trajectory planning algorithms for orienting the mobile platform – just assign values for ϕ and θ, and keep ψ always equal to zero.

Table 1
Data for the GPM (all units are in mm)[a]

i	1	2	3	4	5	6
$\mathbf{OA}_i$	$\begin{bmatrix}-738.035\\-553.122\\0.000\end{bmatrix}$	$\begin{bmatrix}-848.035\\-362.596\\0.000\end{bmatrix}$	$\begin{bmatrix}-110.000\\915.718\\0.000\end{bmatrix}$	$\begin{bmatrix}110.000\\915.718\\0.000\end{bmatrix}$	$\begin{bmatrix}848.035\\-362.596\\0.000\end{bmatrix}$	$\begin{bmatrix}738.035\\-553.122\\0.000\end{bmatrix}$
$\mathbf{CB}'_i$	$\begin{bmatrix}-51.507\\-156.755\\200.000\end{bmatrix}$	$\begin{bmatrix}-161.507\\33.771\\200.000\end{bmatrix}$	$\begin{bmatrix}-110.000\\122.984\\200.000\end{bmatrix}$	$\begin{bmatrix}110.000\\122.984\\200.000\end{bmatrix}$	$\begin{bmatrix}161.507\\33.771\\200.000\end{bmatrix}$	$\begin{bmatrix}51.507\\-156.755\\200.000\end{bmatrix}$
$\mathbf{j}_{A_i} = -\mathbf{j}'_{B_i}$	$\begin{bmatrix}0.433\\0.250\\-0.866\end{bmatrix}$	$\begin{bmatrix}0.433\\0.250\\-0.866\end{bmatrix}$	$\begin{bmatrix}0.000\\-0.500\\-0.866\end{bmatrix}$	$\begin{bmatrix}0.000\\-0.500\\-0.866\end{bmatrix}$	$\begin{bmatrix}-0.433\\0.250\\-0.866\end{bmatrix}$	$\begin{bmatrix}-0.433\\0.250\\-0.866\end{bmatrix}$

[a] In addition: $\ell_{i,\min} = 900$ mm, $\ell_{i,\max} = 1600$ mm, $\alpha_i = \beta_i = 50°$, for $i = 1..6$, and $D = 20$ mm.

The latter guarantees largest constant-orientation workspace and attainment of almost all directions of the approach vector within the projected orientation workspace.

We pointed out that the hybrid algorithm proposed by Merlet [7] cannot be used for finding the *exact* projected orientation workspace. Thus, a *fully* geometrical algorithm has to be devised for computing the exact projection orientation workspace of parallel manipulators.

Appendix A

The following geometrical data describe the GPM used for the examples in this paper (see Table 1).

References

[1] E.F. Fichter, Int. J. Robotics Res. 5 (2) (1986) 157–182.

[2] O. Masory, J. Wang, in: ASME 22nd Biennial Mechanisms Conference, vol. 45, Scottsdale, 1992, pp. 337–346.

[3] J.-P. Merlet, Les Robots Parallèles, Hermès, Paris, 1997.

[4] J.-P. Merlet, Mech. Mach. Theory 29 (8) (1994) 1099–1113.

[5] I.A. Bonev, J. Ryu, A geometrical method for computing the constant-orientation workspace of 6-$\underline{\text{P}}$RRS parallel manipulators, Mech. Mach. Theory 36 (1) (2001) 1–13.

[6] D.-I. Kim, W.-K. Chung, Y. Youm, in: IEEE International Conference on Robotics and Automation, Albuquerque, New Mexico, April 1997, pp. 2986–2991.

[7] J.-P. Merlet, J. Intelligent Robotic Syst. 13 (1995) 143–160.

[8] I.A. Bonev, Analysis and design of 6-DOF 6-PRRS parallel manipulators, Master's Thesis, Kwangju Institute of Science and Technology, Kwangju, December 1998.

[9] F.-C. Yang, E.J. Haug, ASME J. Mech. Design 116 (1994) 111–118.

[10] K.S. Fu, R.C. Gonzalez, C.S.G. Lee, Robotics: Control, Sensing, Vision, and Intelligence, McGraw-Hill, New York, 1987.