Proceedings of the
1999 ASME Design Engineering Technical Conferences
September 12-15, 1999, Las Vegas, Nevada

DETC99/DAC-8646

ORIENTATION WORKSPACE ANALYSIS OF 6-DOF PARALLEL MANIPULATORS

Ilian A. Bonev[*] and Jeha Ryu
Department of Mechatronics
Kwangju Institute of Science and Technology
1 Oryong-dong, Buk-ku, Kwangju 500-712, Korea
Tel: +82-62-970-2389; Email: ryu@kjist.ac.kr

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, a direct method is suggested for the computation of the projected orientation workspace, defined as the set of all possible directions of the approach vector of the mobile platform. Alternative ways of computing these two types of workspaces are also discussed with typical examples.

NOMENCLATURE

A_i: center of base joint i.
B_i: center of platform joint i.
C: center of the mobile reference frame (tool-tip).
$\mathbf{R}$: rotation matrix defined by three Euler angles (ϕ, θ, ψ).
$\mathbf{j}_{A_i}$: unit vector with respect to the base frame along the axis of symmetry of platform joint i.
$\mathbf{j}'_{B_i}$: unit vector with respect to the mobile frame along the axis of symmetry of base joint i.
ℓ_i: length of leg i.
$\mathbf{n}_i$: unit vector along leg i.
α_i: maximum misalignment angle of base joint i.
β_i: maximum misalignment angle of platform joint i.

1 INTRODUCTION

A 6-DOF fully-parallel manipulator, also called a *hexapod*, consists of a *mobile platform* connected by six *legs* to a *base* through respectively spherical and universal joints. Most commonly, the base joints are fixed to the base while the legs are of variable length (e.g. Fichter, 1986; Masory and Wang, 1991). This typical design with six *RRPS* serial kinematic chains, present in most existing hexapods, will be referred to as the *General Parallel Manipulator* (*GPM*). There exist various other architectures of 6-DOF parallel manipulators (see Merlet, 1997).

In evaluating the performance of a 6-DOF parallel manipulator, much concern is given to the workspace factor. As the *complete workspace* is in a six-dimensional (6-D) space for which no graphical representation exists, different subsets of it are usually determined. The most commonly determined subsets are the *constant-orientation workspace* (Bonev and Ryu, 1999; Masory and Wang, 1992; Merlet, 1994), the *reachable workspace*, and the *dextrous workspace* (Kim et al., 1997). 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 workspace to determine and to 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 all 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*.

[*] Currently doctoral student at Département de Génie Mécanique, Université Laval, Québec, Canada.

Very few works exist on the topic of orientation workspace computation. The most detailed work in this area has been presented in (Merlet, 1995), 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 segments on the unit sphere. Thus, the method finds an intersection of the orientation workspace and cannot be used to compute 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 GPM's serial chains.

Another work in this area has been presented in (Romdhane, 1994) where a discretization method is used to compute again a 2-D subset of the orientation workspace of GPMs. In that method, the possible directions of the approach vector are represented as the inside of a general conical surface. That surface is obtained by using a discretization method that investigates the possible rotations about the base *x*-axis, followed by rotations about the base *y*-axis. Thus, the method again finds only an intersection of the orientation workspace and cannot be used to compute the projected orientation workspace.

A more recent work by Benea (1996) deals with the computation of the 3-D orientation workspace of 6-*$\underline{R}$RRS* parallel manipulators using a general discretization method. The author computes the set of standard Euler angles that define all the possible orientations of the mobile platform about a fixed point. Then, he uses several different representations to depict the orientation workspace but in all cases the plots are quite complicated. Despite the fact that the author realizes that the complex nature of the particular orientation workspace representations is due to the coupling between the ϕ and ψ standard Euler angles, he does not come up with another orientation representation.

To avoid the coupling between the standard Euler angles and achieve a more clear representation of the orientation workspace, we propose in this paper a new modified set of Euler angles. With these angles, we represent the orientation workspace in a cylindrical coordinate system, which renders the workspace with a very simple shape. A new discretization algorithm is introduced for this particular choice of modified Euler angles and orientation workspace representation. 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 modified Euler angles and finding an intersection of the orientation workspace. For that purpose, we also introduce a simple two-dimensional discretization method.

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 the 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 the last section 6.

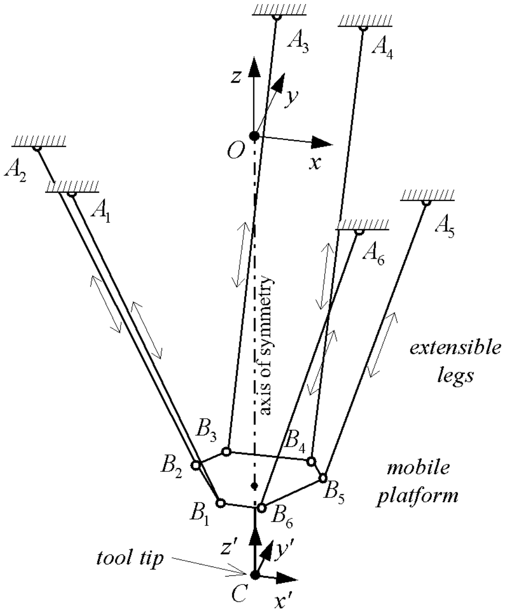

Figure 1. A schematic diagram of an axisymmetric GPM.

2 KINEMATIC CONSTRAINTS

This section summarizes the basic kinematic constraints that limit the orientation workspace. In this paper, we will be concerned only with the GPM, though the same methodology can be applied to other types of parallel manipulators, e.g. with six *$\underline{P}$RRS* kinematic chains (Bonev, 1998). A schematic diagram of a GPM is given in Fig. 1. The centers of the base universal joints are denoted by A_i, and the centers of the mobile platform spherical joints by B_i ($i = 1, \ldots, 6$). A base reference frame is selected fixed to the base at point O, with axes x, y, and 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', and 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 frame 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 leg lengths, denoted by ℓ_i, using the following relation:

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

where vector $\mathbf{CB}'_i$ represents the coordinates of the center of mobile platform joint i with respect to the mobile frame, and vector $\mathbf{OA}_i$ represents the coordinates of the center of base joint i with respect to the base frame. Equation (1) is the solution of the so-called inverse kinematics problem.

There exist 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 leg interference.

2.1 Actuators' Stroke

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

$$\ell_{i,\min} \le \ell_i \le \ell_{i,\max} \text{ for } i = 1,\ldots,6, \tag{2}$$

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

$$\cos^{-1}(\mathbf{j}_{A_i}^T\mathbf{n}_i) \le \alpha_i \text{ for } i = 1,\ldots,6. \tag{3}$$

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

$$\cos^{-1}(\mathbf{j}_{B_i}^T\mathbf{n}_i) \le \beta_i \text{ for } i = 1,\ldots,6. \tag{4}$$

2.3 Leg Interference

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

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

or the minimum distance between every two line segments corresponding to the legs 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 detailed one given in (Masory and Wang, 1992).

It should be pointed out here that, 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 and satisfying all constraints given by Eqs. (2-5). To the best of our knowledge, no direct concern has been given to this compatibility constraint by authors using discretization methods for workspace evaluation (e.g. Fichter, 1986; Masory and Wang, 1992).

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 (Yang and Haug, 1994), direction cosines, etc. While they provide a global parameterization of the orientation, they call for a representation in a 4-D space. To overcome this drawback, three Euler angles may be used to represent the mobile platform orientation. These angles correspond to three or more successive rotations about the base and/or mobile 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 somewhat difficult to interpret in the general case. Since our goal in this paper is to determine not only the orientation workspace but also the projected orientation workspace for 5-axis machining operations it would be advantageous to use the same set of Euler angles in both cases. A clear transition from the orientation workspace to the projected orientation workspace is achieved if the first 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*, corresponds to the last rotation about the mobile z'-axis. Thus, for such a set of Euler angles, the projected orientation workspace will be obtained by projecting the 3-D orientation workspace onto a 2-D space defined by the angles ϕ and θ.

Probably the most intuitive choice of Euler angles ϕ and θ is the one corresponding to the *azimuth* and *zenith* angles that

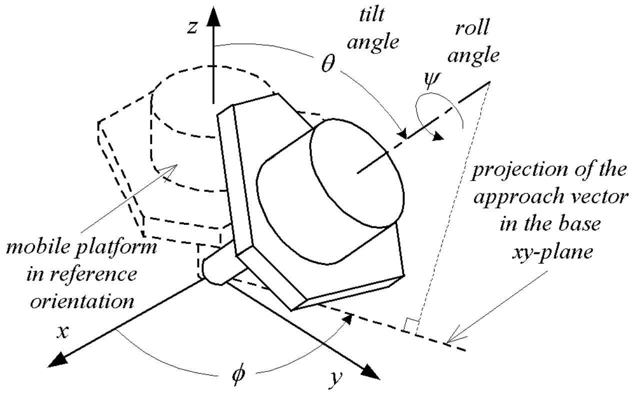

Figure 2. The modified Euler angles defining the platform orientation.

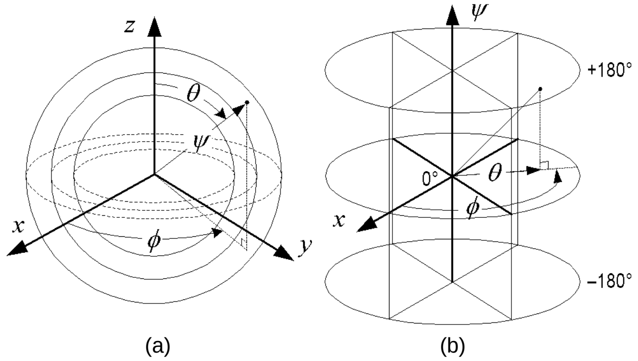

Figure 3. Two of the three possible representations of the orientation workspace.

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.

Up to here, we specified the requirements on the three Euler angles with the desire to achieve a more intuitive representation and a better correlation between the orientation workspace and the projected orientation workspace. An intuitive representation of the mobile platform orientation is essential for easy use by a general machine operator. Simply put, when a triplet of Euler angles is given, the user (operator) must be able to easily associate them with the corresponding orientation of the mobile platform. However, as our main goal is to represent the complete 3-D orientation workspace, we must also impose certain requirements related to the simplicity in the representation of this workspace.

The first requirements set for the Euler angles are met by the *standard Euler angles* that are 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 (Fu et al., 1987). For this choice of Euler angles, the singularity occurs at $\theta = 0°$ and the rotation matrix is defined as

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

where $\mathbf{R}_z(\cdot)$ and $\mathbf{R}_y(\cdot)$ are basic rotation matrices.

The standard Euler angles were used by Benea (1996) to compute and represent the workspace of 6-*<u>R</u>RRS* parallel manipulators. Despite the use of several different representations, the plots of the orientation workspace remained too complicated due to the coupling nature between the Euler angles φ and ψ.

To overcome the problems imposed by the standard Euler angles, we introduce a *modified set of Euler angles*, which to our best 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. Defined in this way, angle ψ is the *roll angle*, angle θ is the *tilt angle*, and angle φ is the angle between the base *x*-axis and the projection of the approach vector onto the base *xy*-plane (Fig. 2). 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*-plane, and making an angle φ with the base *y*-axis. The singularity for this set of Euler angles occurs again at $\theta = 0°$ and the rotation matrix is defined as

$$\mathbf{R} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(-\phi)\mathbf{R}_{z'}(\psi) = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi-\phi). \quad (7)$$

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 single volume having a simple shape.

4 ORIENTATION WORKSPACE

After selecting the set of modified Euler angles for representing the mobile platform orientation, it remains to determine the way to represent the orientation workspace. With the selected set of modified Euler angles, the maximum range of orientations will be $\phi \in [-180°, +180°)$, $\theta \in [-0°, +180°]$, and $\psi \in [-180°, +180°)$, since orientations $(\phi, -\theta, \psi)$ and $(\phi, \theta, \psi \pm 180°)$ are identical. Three alternatives exist for representing the orientation workspace.

The first alternative is to represent the orientation workspace in a Cartesian coordinate system whose axes are the three Euler angles (Benea, 1996). Such a representation is very 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 (Benea, 1996) where ϕ and θ are exactly the azimuth and zenith angles (Fig. 3a). Then, the ray length will correspond to ψ, being $1-\rho$ for $\psi = -180^\circ$, and $1+\rho$ for $\psi = +180^\circ$, where $\rho < 1$. Thus, the orientation workspace will be inside the spherical shell enclosed between the two spheres with radii $1-\rho$ and $1+\rho$, centered at the coordinate system origin. Thus, the projection of the orientation workspace onto the unit sphere will be exactly the projected orientation workspace. The final alternative is to represent the workspace in a cylindrical coordinate system, where ϕ and θ are exactly the polar coordinates and ψ is the z-coordinate (Fig. 3b).

Both the second and the third representations do not pose any problems at the singularity $\theta = 0^\circ$ and are relatively easy to interpret. 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 $+180^\circ$. 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 one double array and the next orientation for which a constraint becomes violated is stored in another double array. The first array will define the inner boundary of the workspace while the second array will define the outer boundary. The problem with this representation is that the inner boundary of the orientation workspace is not visible 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. In fact for the standard Euler angles, Fig. 5.38 of (Benea, 1996) shows that indeed there are several such changes.

To avoid these shortcomings, we choose the third type of representation shown in Fig. 3b. In this representation, the orientation workspace is a single volume with no hidden regions. In addition, the projection of the orientation workspace onto a plane $\psi = const$ 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 new discretization algorithm for determining and representing the orientation workspace:

Algorithm for the Orientation Workspace:

(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 $\psi = const$ between $\psi = -180^\circ$ and $\psi = +180^\circ$ at which the workspace will be computed, and n_ϕ is the number of points to be computed at each plane $\psi = const$. These arrays will store respectively the values of ϕ and θ for the points defining the upper part of the workspace boundary.

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

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 stored 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 $\psi = 0^\circ$, store the geometric center and repeat step 3 only once with the new geometric center and then go to step 5.

S5. Set $\psi = \psi + 360^\circ/n_\psi$.

S6. Repeat steps 3 through 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).

(Phase II) Lower Part of the Orientation Workspace:

S7. Initialize double arrays $\mathbf{W}_{\phi,l}$ and $\mathbf{W}_{\theta,l}$, with dimensions $(n_\psi/2) \times n_\phi$.

S8. Set $\psi = -360^\circ/n_\psi$. Assign to (ϕ_c, θ_c) the values that were stored in step 4 for $\psi = 0^\circ$.

S9. Perform the same as in step 3.

S10. 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.

S11. Set $\psi = \psi - 360^\circ/n_\psi$.

S12. Repeat steps 10 through 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).

(Phase III) Postprocessing and Plotting:

S13. 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 $N_\psi = (\psi_{\max} - \psi_{\min})/(360/n_\psi) + 1$.

S14. Transfer $\mathbf{W}_\phi$ and $\mathbf{W}_\theta$ into $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, so that $\mathbf{X}[i,j] = \mathbf{W}_\theta[i,j]\cos\left(\mathbf{W}_\phi[i,j]\right)$, $\mathbf{Y}[i,j] = \mathbf{W}_\theta[i,j]\sin\left(\mathbf{W}_\phi[i,j]\right)$, and $\mathbf{Z}[i,j] = \psi_{\max} - (i-1)(360/n_\psi)$, where $i = 1,\ldots,N_\psi$ and $j = 1,\ldots,n_\phi$, and $\mathbf{X}[i,j] = \mathbf{X}[i,1]$, $\mathbf{Y}[i,j] = \mathbf{Y}[i,1]$, and $\mathbf{Z}[i,j] = \mathbf{X}[i,1]$, where $i = 1,\ldots,N_\psi$ and $j = n_\phi + 1$.

S15. Plot the closed surface whose nodes are defined in the double arrays $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$.

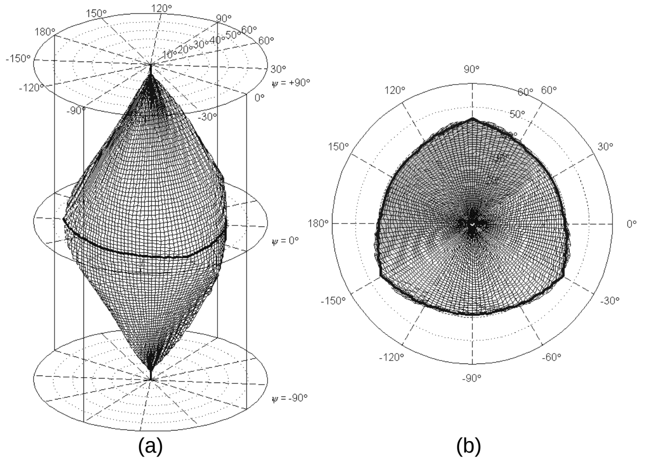

Figure 4. (a) Perspective and (b) top views of the orientation workspace of the GPM for the first position.

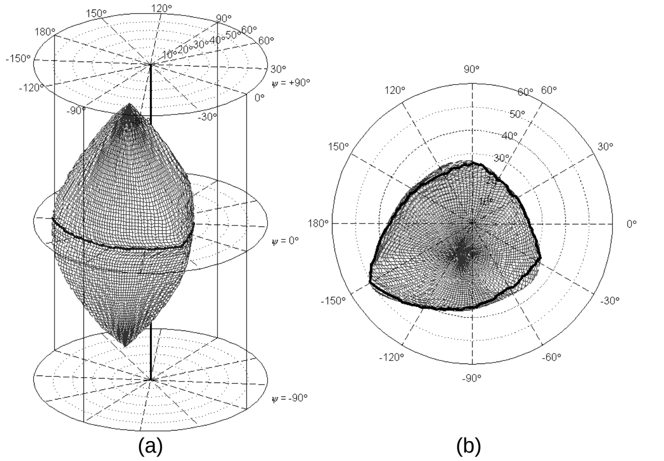

Figure 5. (a) Perspective and (b) top views of the orientation workspace of the GPM for the second position.

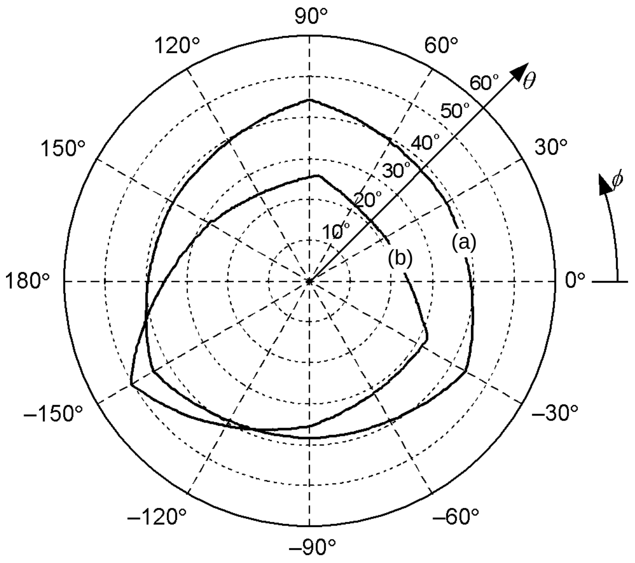

Figure 6. Close approximations of the projected orientation workspace of the GPM for the (a) first and (b) second positions.

The proposed algorithm was implemented in MATLAB for the GPM whose data are given in the Appendix. In our implementation, $n_\psi = 180$ and $n_\phi = 120$. Two examples are presented here. In the first one (Fig. 4), the orientation workspace is computed for a position at which point C lies on the axis of symmetry of the parallel manipulator ($\mathbf{OC} = [0, 0, -1300]^T$). Consequently, we may observe in Fig. 4b the symmetrical shape of the orientation workspace with respect to the axis $\theta = 0°$. For this position the maximum and minimum roll angles are respectively $\psi_{\max} = 84°$ and $\psi_{\min} = -84°$. In the second example (Fig. 5), the orientation workspace is computed for a position at which point C is far from the axis of symmetry of the parallel manipulator and near the boundary of the constant-orientation workspace for the reference orientation ($\mathbf{OC} = [200, 200, -950]^T$). Note, correspondingly, how the axis of the orientation workspace is shifted away from the axis $\theta = 0°$. For this position the maximum and minimum roll angles are respectively $\psi_{\max} = 72°$ and $\psi_{\min} = -72°$.

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 leg interference check. On the other hand, for this GPM as well as for other parallel manipulators (Bonev, 1998), 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, leg interference was never encountered. Thus, for some parallel manipulators, the leg 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 Fig. 4 and Fig. 5 are respectively shown in Fig. 4b and Fig. 5b (the top views of the orientation workspaces). In other words, the projected orientation workspace can be found by first computing 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 (e.g. in 5-axes machining). Thus, for some applications, it would be beneficial to find directly the projected orientation workspace.

Now, observe again Figs. 4b and 5b. The thick curves that may be seen there are the cross-sections of the boundary of the orientation workspace for $\psi = 0°$. 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. 4b) and a fair one when it is far from it (Fig. 5b). Therefore, with the assumption that the reference orientation is inside the orientation workspace we may propose the following two-dimensional discretization algorithm for the computation of the *approximated* projected orientation workspace.

Algorithm for the Projected Orientation Workspace:

S1. Initialize double arrays $\mathbf{W}_\theta$ 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.

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. (2-5).

S5. Repeat steps 3 through 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 through 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 discretization algorithm was again implemented in MATLAB. Figure 6 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 leg interference check) is quite small—about 20 sec 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 parallel manipulators that exhibit a symmetry about the base z-axis. 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.

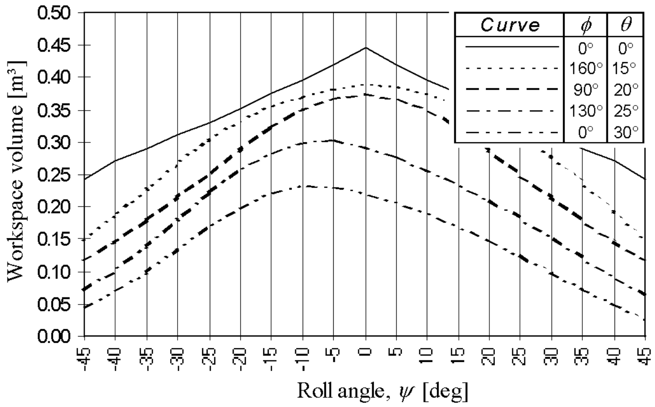

Figure 7. Volume of the constant-orientation workspace as a function of the roll angle ψ for several directions of the approach vector.

Let us note that the algorithm presented in (Merlet, 1995) 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 found by our discretization method. For that purpose, the range of the angle ϕ is discretized in the range $[0°, +180°)$, and then for each value, Merlet's algorithm is applied to obtain geometrically the range of the tilt angle θ. Note, however, that his algorithm is much more difficult to implement and we do not expect that it will 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 (Merlet, 1994) for the computation of the constant-orientation workspace of GPMs (excluding the leg interference modeling). The implementation was made in the CAD/CAM system CATIA, similarly to the one described in (Bonev and Ryu, 1999). It was observed that for a fixed direction of the approach vector (ϕ, θ), the volume of the constant-orientation workspace is largest near $\psi = 0°$ (Fig. 7). In fact, the same was observed by Benea (1996) for the case of 6-$R\underline{R}RS$ parallel manipulators, where he pointed out that the constant-orientation workspace is greatest at $\phi + \psi = 0°$, which in the modified Euler angles is exactly $\psi = 0°$. The same is true also for the 6-$\underline{P}RRS$ parallel manipulators (Bonev, 1998).

6 CONCLUSIONS

A new discretization algorithm for computing the 3-D orientation workspace was presented in this paper. The algorithm is based on a set of modified Euler angles and a particular representation of the orientation workspace. The 2-D 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 believe we introduced a valuable 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, i.e. just assign values to ϕ and θ, and keep ψ always equal to zero. This 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 (1995) cannot be used for finding the exact projected orientation workspace. Thus, our next goal will be to devise a fully-geometric algorithm for computing the exact projection orientation workspace of parallel manipulators based on the vertex space concept discussed in (Bonev and Ryu, 1999).

REFERENCES

Benea, R., 1996, "Contribution à l'étude des robots pleinement parallèles de type 6 R-RR-S," *Ph.D. Thesis*, Université de Savoie, Annecy, France.

Bonev, I. A., 1998, "Analysis and Design of 6-DOF 6-PRRS Parallel Manipulators," *Master's Thesis*, Kwangju Institute of Science and Technology, Kwangju, South Korea.

Bonev, I. A., and Ryu, J., 1999, "Workspace Analysis of 6-PRRS Parallel Manipulators Based on the Vertex Space Concept," *Proceedings of the ASME Design Engineering Technical Conferences*, Las Vegas, Nevada, DETC99/DAC-8647.

Fichter, E. F., 1986, "A Stewart Platform-Based Manipulator: General Theory and Practical Construction," *The International Journal of Robotics Research*, Vol. 5, No. 2, pp. 157-182.

Fu, K. S., Gonzalez, R. C., and Lee, C. S. G., 1987, *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill.

Kim, D-I., Chung, W-K., and Youm, Y., 1997, "Geometrical Approach for the Workspace of 6-DOF Parallel Manipulators," *IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, pp. 2986-2991.

Masory, O., and Wang, J., 1992, "Workspace Evaluation of Stewart Platforms," *Proceedings of the ASME 22nd Biennial Mechanisms Conference*, Scottsdale, Arizona, Vol. 45, pp. 337-346.

Merlet, J.-P., 1994, "Détermination de l'espace de travail d'un robot parallèle pour une orientation constante," *Mechanism and Machine Theory*, Vol. 29, No. 8, pp. 1099-1113.

Merlet, J-P., 1995, "Determination of the Orientation Workspace of Parallel Manipulators," *Journal of Intelligent and Robotic Systems*, Vol. 13, pp. 143-160.

Merlet, J-P., 1997, *Les robots parallèles*, 2nd ed., Hermès, Paris.

Romdhane, L., 1994, "Orientation Workspace of Fully Parallel Mechanisms," *European Journal of Mechanics*, A/Solids, Vol. 13, No. 4, pp. 541-553.

Yang, F-C., and Haug, E. J., 1994, "Numerical Analysis of the Kinematic Working Capability of Mechanisms," *ASME Journal of Mechanical Design*, Vol. 116, pp. 111-118, March.

APPENDIX: DATA FOR THE GPM

Table 1 shows the data for the GPM used in the examples in this paper. In addition, $\ell_{i,\min} = 900$ mm, $\ell_{i,\max} = 1600$ mm, $\alpha_i = \beta_i = 50^\circ$, for $i = 1, \ldots, 6$, and $D = 20$ mm.

Table 1. Geometry of the GPM (all units are in [mm]).

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