6 자유도 병렬로보트의 새로운 오리엔테이션 작업공간 해석

ILIAN A. BONEV, JEHA RYU

광주과학기술원 기전공학과 동력학 및 제어 연구실

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 on 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 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 workspace* (Bonev and Ryu, 1999; Masory and Wang, 1992; Merlet, 1994), the *reachable workspace*, and the *dexterous 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 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 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, that method finds only an intersection of the orientation workspace, and can not be used in the general case to compute the projected orientation workspace. Furthermore, the method can not 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 the last section 6.

2. KINEMATIC CONSTRAINTS

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 *PRRS* kinematic chains (Bonev, 1998). An example 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 ($i = 1 \cdots 6$). 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 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 **R**. For a given position (vector **OC**) and orientation (matrix **R**) of the mobile platform we may compute the necessary leg lengths, denoted by ℓ_i, using the following relation:

$$\ell_i = \| \mathbf{OC} + \mathbf{RCB}'_i - \mathbf{OA}_i \|, \text{ for } i = 1 \cdots 6 \quad (1)$$

where vector $\mathbf{CB}'_i$ represents the coordinates of the center of mobile platform join 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 Actuator's Stroke

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

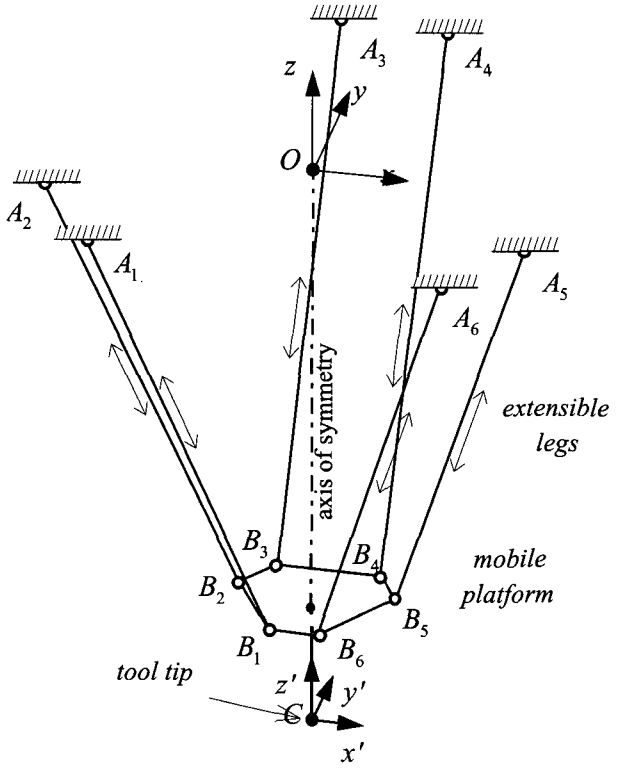


Fig. 1. A schematic diagram of an axisymmetric GPM.

$$\ell_{i,\min} \le \ell_i \le \ell_{i,\max}, \text{ for } I = 1 \cdots 6 \quad (2)$$

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 $\mathbf{j}_A$ be the unit vector with respect to the base frame and 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 $\mathbf{n}_i$. Then, the limits on base joint i impose a constraint, such that

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

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. Let the maximum misalignment angle of that joint be β_i. Then, the limits on mobile platform joint i impose a constraint, such that

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

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

$$\text{distance}(\mathbf{A}_i\mathbf{B}_i,\ \mathbf{A}_j\mathbf{B}_j) \geq D, \text{ for } i = 1\cdots6,\ j = (i+1)..6 \quad (5)$$

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 (Masory and Wang, 1992).

It should be pointed out at this point 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 configuration and 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 (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 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 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 all difficult to interpret in the general case. Since our goal is to determine not only the orientation workspace but also the projected orientation workspace, 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 define the ray direction in a spherical coordinate system. Further, in machining, the angle θ will also correspond to the *tilt angle.*

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. Now, we impose another requirement on the three Euler angles-the orientation workspace should be a single and simply connected set and, when represented in a particular coordinate system, it should have a simple shape.

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

$$\mathbf{R} = \mathbf{R}_z(f)\mathbf{R}_y'(q)\mathbf{R}_z'(y) = \mathbf{R}_z(f)\mathbf{R}_y(q)\mathbf{R}_z(y), \quad (6)$$

where $\mathbf{R}z(\cdot)$ and $\mathbf{R}y(\cdot)$ are the basic rotation matrices.

We tried to determine and represent the orientation workspace of an axisymmetric GPM for this set of Euler angles, but despite the choice of coordinate system, the shape of the workspace seemed to be quite complicated. In an attempt to achieve a simpler shape, 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. 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

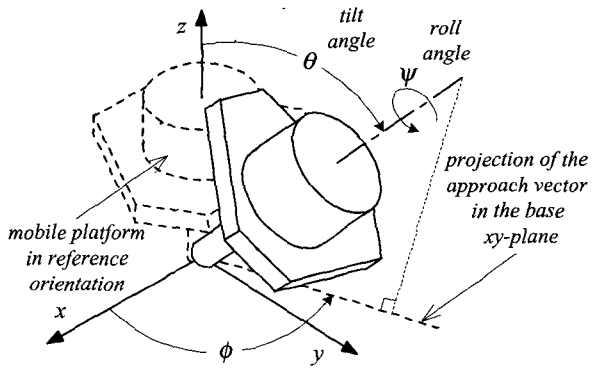


Fig. 2. The modified Euler angles defining the orientation of the mobile platform.

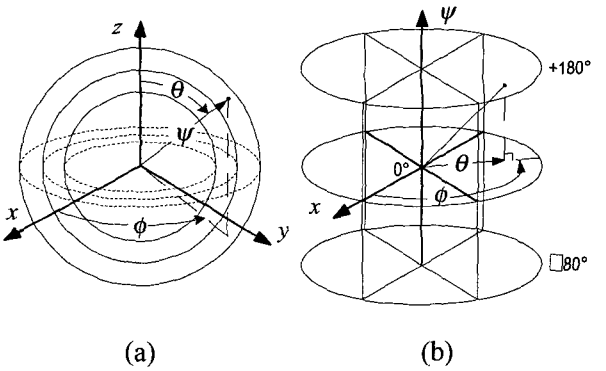


Fig. 3. Two of the three possible representations of the orientation workspace.

approach vector onto the base xy-plane (Fig. 2). Note that for a zero roll angle, the mobile platform is simply titled (rotated) about an axis passing through the mobile frame center, parallel to the base *xy*-frame, and making an angle with the base *y*-axis. The singularity for this set of Euler angles occurs again at orientations for which $\theta = 0°$ and the rotation matrix is defined as

$$\mathbf{R} = \mathbf{R}_z(f)\mathbf{R}_y(q)\mathbf{R}_z(-f)\mathbf{R}_z'(y) = \mathbf{R}_z(f)\mathbf{R}_y(q)\mathbf{R}_z(-f)\mathbf{R}_z(y), \quad (7)$$

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 (ϕ, θ, $\psi - \phi$). As we will see in the next section, the modified Euler angles allows 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 Euler angles for representing the platform orientation, it remains to determine the way to represent the orientation workspace. With the selected set of Euler angles, the maximum range of orientations is $\phi IN[-180°, 180°)$, $\theta \in [0°, 180°]$, and $\psi \in [-180°, 180°)$. Three alternatives exist for representing the orientation workspace.

The first one is to represent it in a Cartesian coordinate system whose axes are the three Euler angles. 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 where ϕ and θ are exactly the azimuth and zenith angles (Fig. 3a). The ray length will correspond to ψ, so that the orientation workspace will be inside a spherical shell, 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. 3b).

Both the second and the third representations do not pose any problems at the singularity $\theta = 0°$ 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°. 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 chose the third type of representation shown in Fig. 3b. 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 $\psi = const$ 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:

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 $\psi=-180°$ and $\psi=180°$ at which the workspace will be computed, and n_ϕ is the number of points to be computed at each plane $\psi=const$. 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 $(\phi_c, \theta_c)=(0°, 0°)$ is the center of the horizontal cross-section of the workspace for $\psi=0°$.

S3. For the current ψ, construct a polar 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 violation are written into the two double arrays.

S5. 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.

S6. Set $\psi=\psi+360°/n_\phi$.

S7. Repeat steps 3 to 5 until ψ becomes greater than 180° or the last horizontal cross-section of the workspace is a single point (i.e. $\psi_{\max}$ is reached).

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

(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 $\psi=-360°/n_\phi$. 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 $\psi=\psi-360°/n_\phi$.

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

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

(Phase III) Postprocessing and Plotting:

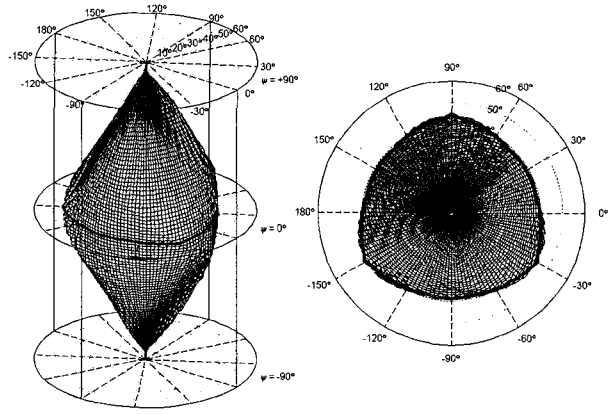
S15. Transfer the values from $\mathbf{W}_{\phi,u}$ and $\mathbf{W}_{\phi,l}$ to $\mathbf{W}_\phi$, and from $\mathbf{W}_{\theta,u}$ and $\mathbf{W}_{\theta,l}$ to $\mathbf{W}_\theta$, which are double arrays of dimension $N_\psi\times n_\phi$, where $N_\psi=(\psi_{\max}-\psi_{\min})/(360/n_\psi)+1$.

S16. Transfer $\mathbf{W}_\phi$ and $\mathbf{W}_\theta$ into **X**, **Y**, and **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 $i=1\cdots N_\psi$ and $j=1\cdots n_\phi$, 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**.

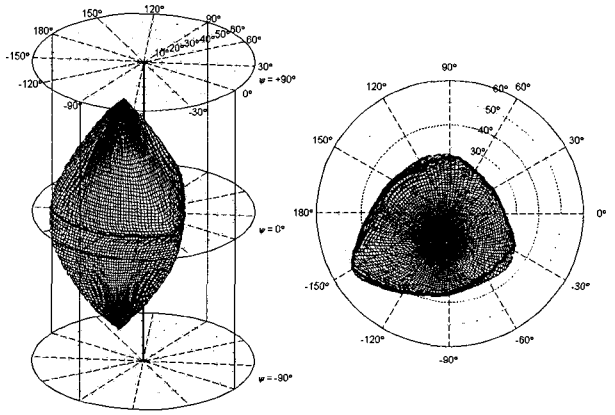
The proposed algorithm was implemented in MATLAB with $n_\psi=180$ and $n_\phi=120$. Two examples are presented here. In the first (Fig. 4), the orientation workspace is computed for a position at which point C lies on the axis of symmetry of the parallel manipulator. Consequently, we may observe in Fig. 4b the symmetrical shape of the orientation workspace with respect to the axis $\theta=0°$. 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. 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 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,



(a) (b)

Fig. 4. (a) Perspective and (b) top views of the orientation workspace of the GPM at the position $\mathbf{OC} = [0, 0, -1300]^T$ for which $\psi_{max} = 84°$ and $\psi_{min} = -84°$.



(a) (b)

Fig. 5. (a) Perspective and (b) top views of the orientation workspace of the GPM at the position $\mathbf{OC} = [200, 200, -950]^T$ for which $\psi_{max} = 72°$ and $\psi_{min} = -72°$.

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 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 (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 $\phi = 0°$. 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 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.

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 3-4, until θ becomes 180° or a constraint is violated.

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

S7. Set $\theta = \theta - m\Delta\theta$, where m is the number of 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 ray length at $0°, 360°/n_\phi, 2(360°/n_\phi), \cdots 360°$.

The proposed algorithm was again implemented in MATLAB. Fig. 6 shows the approximated projected workspaces for the same positions as before. 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

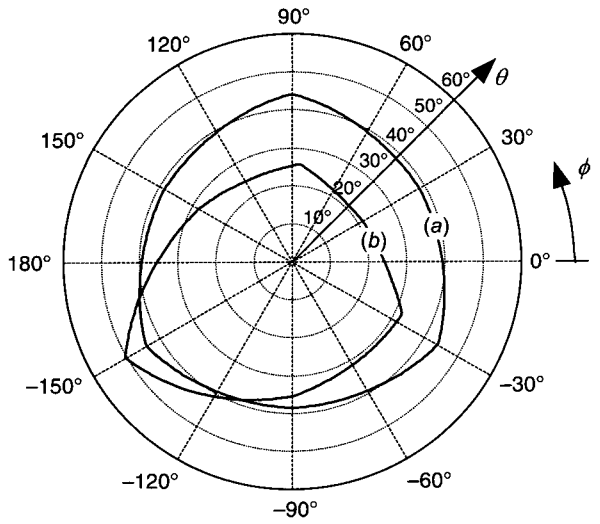


Fig. 6. Close approximation of the projected orientation workspace of the GPM for the positions (a) $\mathbf{OC} = [0, 0, -1300]^T$ ($\theta_{min} = 38.4°$, $\theta_{max} = 44{,}1°$) and (b) $\mathbf{OC} = [200, 200, -950]^T$ ($\theta_{min} = 21.0°$, $\theta_{max} = 50.3°$).

boundary, i.e. at $\phi = 0°$.

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. Thus, another more general approach should be sought for the parallel manipulators with no axial symmetry.

Let us note that Merlet's algorithm 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, Merlets 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.

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 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-just assign values for ϕ and θ, and keep ψ always zero. The latter guarantees 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-geometrical algorithm for computing the exact projection orientation workspace of parallel manipulators based on the vertex space concept discussed in (Bonev and Ryu, 1999).

REFERENCES

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

[2] Bonev, I. A., and Ryu, J., "Workspace Analysis of 6-*PRRS* Parallel Manipulators Based on the Vertex Space Concept," Submitted for review to *ASME Design Engineering Technical Conference*, Las Vegas, Nevada, 1999.

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

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

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

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

[7] Merlet, J-P., 1994, "Determination de l'espace de travail d'un robot parallele pour une orientation constante," *Mechanism and Machine Theory*, vol.

29, no. 8, pp. 1099-1113.
[8] Merlet, J-P., "Determination of the Orientation Workspace of Parallel Manipulators," *Journal of Intelligent and Robotic Systems*, vol. 13, pp. 143-160, 1995.
[9] Merlet, J-P., *Les robots paralleles*, 2nd ed., Hermes, Paris, 1997.
[10] Yang, F-C., and Haug, E. J., "Numerical Analysis of the Kinematic Working Capability of Mechanisms", *ASME Journal of Mechanical Design*, vol. 116, pp. 111-118, 1994.

저자소개

류 제 하 (柳 濟 夏)
1959년 7월 19일생
1982년 서울대학교 기계공학과 (공학사)
1984년 KAIST 기계공학과 (석사)
1991년 The University of Iowa, Mechanical Engineering (공학박사)
1992년 - 1994년 United Defense LP 선임연구원
1995년 - 현재 광주과학기술원 기전공학과 조교수
<관심분야>
- 병렬로보트 기구학/동력학/제어
- Haptic Device for VR Interface
- 차량동력학/HILS/ITS
<연락처>
TEL: 062-970-2389, FAX: 062-970-2384
Email: ryu@kjist.ac.kr

Ilian Al. Bonev
1975년 4월 19일생
1996년 Technical University-Sofia, Bulgaria, Department of Engineering, B.Eng.
1999년 광주과학기술원 기전공학과 (석사)
<관심분야>
- 병렬로보트 기구학/동력학
- 해석/설계/시뮬레이션
<연락처>
TEL: 062-970-2425, FAX: 062-970-2384
Email: ilian@geguri.kjist.ac.kr