Determination of 6D-workspaces of Gough-type parallel manipulator and comparison between different geometries

J-P. Merlet
INRIA Sophia-Antipolis, France

Abstract: We consider in this paper a Gough-type parallel robot whose leg lengths values are constrained to lie within some fixed ranges and for which there may be mechanical limits for the motion of the passive joints.

The purpose of this paper is to present algorithms to determine:

- the *constant orientation workspace*: all the possible locations of the center of the platform which can be reached with a fixed orientation
- the *total orientation workspace*: all the possible locations of the center of the platform which can be reached with any orientation in a set defined by 3 ranges for the orientation angles (the *dextrous workspace* is an example of total orientation workspace case, the three ranges being $[0, 360]$ degree[1])
- the *inclusive orientation workspace*: all the possible locations of the center of the platform which can be reached with at least one orientation among a set defined by 3 ranges for the orientation angles (the *maximal or reachable workspace* is an example of inclusive orientation workspace, the three ranges being $[0, 360]$)

Most of these algorithms are based on a basic method: approximation of the result by a set of 3D or 6D boxes obtained from an initial estimation through a bisection process. The boxes in the result will either fully or partially lie inside the workspace: the bisection stops as soon as all the boxes which does not lie fully inside the workspace have a size that is lower than a fixed threshold.

A companion algorithm enable to verify if any 6D workspace (i.e. a continuous set of poses) lies within the reachable workspace of the robot. The paper includes a comparison between the workspaces volumes of four different robot geometries which shows that for robots of similar dimensions the joints layout has a large influence on the workspace volume.

1 Introduction

1.1 Robot geometry

In this paper we consider a 6 d.o.f. parallel manipulator (figure 1) constituted of a fixed base plate and a mobile plate connected by 6 extensible links. A reference frame (O, x, y, z) is attached to the base and a mobile frame (C, x_r, y_r, z_r) is attached to the moving platform. The leg i is attached to the base with a ball-and-socket joint whose

[1] In all of this paper the angles are expressed in degree

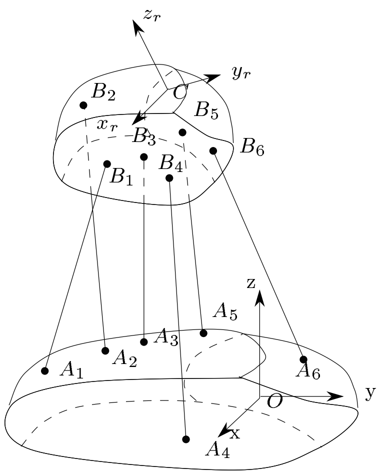

Figure 1: Gough platform

center is A_i while it is attached to the moving platform with an universal joint whose center is B_i. Let ρ_i be the leg lengths (the distance between A_i and B_i), $\boldsymbol{X}$ a 6-dimensional vector defining the pose of the end-effector: the three first components of $\boldsymbol{X}$ are the coordinates of C in the reference frame while the three last components are three parameters describing the orientation of the end-effector. In this paper we will use as orientation parameters the Euler angles ψ, θ, ϕ (the mobile frame is derived from the reference frame by first a rotation around the z axis with an angle ψ, then a rotation around the new x axis with an angle θ and finally a rotation around the new z axis with an angle ϕ). The workspace of this type of robot is restricted as the leg lengths have a minimal and maximal values which will be denoted by $\rho^i_{min}, \rho^i_{max}$ for leg i. A set of leg lengths will be called *valid* if the lengths of all the legs lie within their given limits.

As the leg lengths are functions of both the location and the orientation of the end-effector, computing the workspace of this type of manipulator is a complex task.

1.2 Extended box

We define an *extended box* (or EB for short) as a pair of elements: a cartesian box, which represents the possible locations of the center of the end-effector, and a set of three ranges, one for each of the rotation angles. An EB is therefore composed of a *location part* (the box) and an *orientation part* and defines a 6D workspace for the robot.

1.3 Minimal and maximal leg lengths over an extended box

Let us consider the extremal value of the leg lengths over the set of poses defined by an extended box. The following results hold:

- for an EB whose orientation part is reduced to three fixed values (i.e. the orientation of the platform is fixed) it is possible to compute exactly the extremal values of the leg lengths [Merlet 1997-2]

- for a general EB if both the base and the platform are planar it is possible to compute exactly the extremal values of the leg lengths
- for a general EB and a robot of general geometry it is possible to compute bounds for the extremal values of the leg lengths by using interval analysis [Moore 79]. We may therefore compute the extremum with an arbitrary accuracy.

In the sequel we will denote by $\mathcal{I}_i = [\rho_m^i, \rho_M^i]$ an interval for the length of leg i such that for any pose within an EB the length ρ_i of leg i satisfy $\rho_i \in [\rho_m^i, \rho_M^i]$. The interval $\mathcal{I}_i$ will be denoted the *lengths bound interval*. Note that the bounds ρ_m^i, ρ_M^i may be overestimated (there is no pose within the EB such that the length of leg i is equal to ρ_m^i or ρ_M^i). It must also be mentioned that the method we will use to compute ρ_m^i, ρ_M^i strictly guarantee that for any pose within an EB the leg lengths we will have $\rho_m^i \leq \rho_i \leq \rho_M^i$ even taking into account round-off numerical errors.

2 Constant orientation workspace

The constant orientation workspace (`COW` for short) is defined as the possible locations of C which can be reached with a valid set of leg lengths, the orientation of the platform being fixed (therefore the vectors $\mathbf{B_iC}$ are fixed). Determination of `COW` has been addressed by numerous authors [Jo 89, Masory 1993] and we remind the result only for the completeness of this paper. It can easily be shown that a `COW` is the intersection of 6 volumes, each of them being the difference between a sphere of radius ρ_{max}^i whose center is the point A_i translated by the vector $\mathbf{B_iC}$ and a sphere with same center and radius ρ_{min}^i. The border of the `COW` may therefore be described as a set of regions on various spheres, which can be computed exactly [Gosselin 1992]. Furthermore the intersection of the workspace with a plane is obtained as the intersection of six annular regions [Gosselin 1990]. Therefore the border of such cross-section is constituted of a set of circular arcs which can be computed exactly. Furthermore, this algorithm can be extended to take into account the mechanical limits on the passive joints and the interference between the legs [Merlet 1997] [2].

Figure 2 presents an example of workspace computation as a set of horizontal cross-sections for the four robot geometries defined in section 7.

3 Valid location for the end-effector

We define an orientation space as a set of orientations of the moving platform such that each orientation parameter lies within a fixed range S_ψ, S_θ, S_ϕ.

The problem we want to solve in this section is to determine if for a given location of C there exist at least one orientation within a given orientation space such that the leg

[2] An implementation of this algorithm is available via anonymous ftp at zenon.inria.fr in the directory coprin/Workspace/Gough

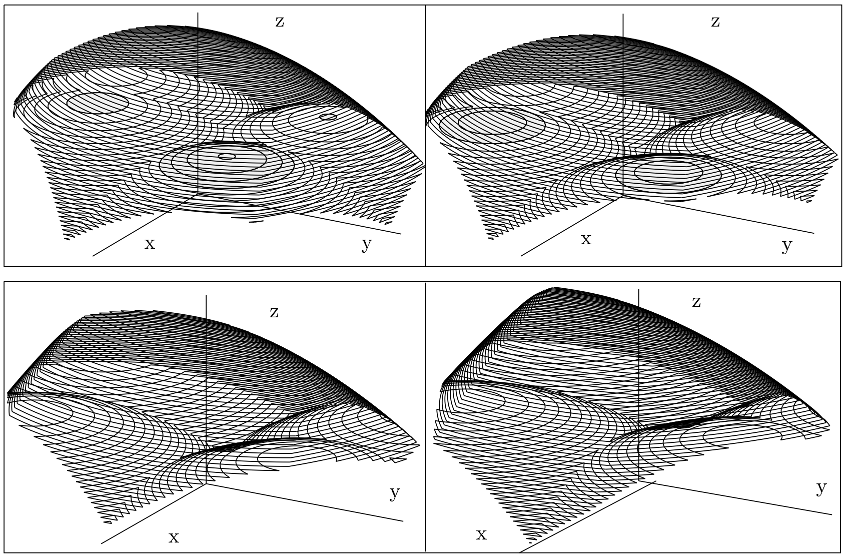

Figure 2: Constant orientation workspace of the four robot geometries defined in section 7. From left to right and top to bottom workspace for the SSM, TSSM, MSSM and MSP. The orientation angles are $\psi = \theta = 10°$, $\phi = 0$. The z axis has a scale factor of 3 in comparison to the x and y axis.

lengths are valid (in which case we will say that the location is *valid*). Note that if the three ranges are defined as $[0, 360]$ then the problem is to determine if C belongs to the reachable workspace of the robot.

3.1 Algorithm principles

We start our algorithm by considering an EB whose location part is reduced to the location of C and whose orientation part is defined as S_ψ, S_θ, S_ϕ. We then compute the lengths bound interval $\mathcal{I}_i = [\rho^i_m, \rho^i_M]$ of each leg for this EB. Three cases may occur:

1. for all legs $[\rho^i_m, \rho^i_M] \subset [\rho^i_{min}, \rho^i_{max}]$: the location is valid
2. for at least one leg we have either $\rho^i_m > \rho^i_{max}$ or $\rho^i_M < \rho^i_{min}$: the location is not valid
3. for at least one leg we have $[\rho^i_{min}, \rho^i_{max}] \subset [\rho^i_m, \rho^i_M]$ while no leg satisfies condition 2.

In the first two cases the algorithm has solved the problem at hand, so we have to deal only with the last case. We split each range of the orientation part of the EB into two ranges (e.g. the range $S_\psi = [\psi_1, \psi_2]$ leads to the ranges $[\psi_1, (\psi_1 + \psi_2)/2], [(\psi_1 + \psi_2)/2, \psi_2])$ and we build a list of EBs which contains the 8 EBs obtained by taking all the possible combinations of the new ranges. The algorithm will consider sequentially each EB of the list and perform the following operations:

1. compute the lengths bound interval of each leg for the current EB
2. if for all legs $[\rho^i_m, \rho^i_M] \subset [\rho^i_{min}, \rho^i_{max}]$ the algorithm exits as the location is valid
3. if for at least one leg we have either $\rho^i_m > \rho^i_{max}$ or $\rho^i_M < \rho^i_{min}$ the algorithm skips to the next EB in the list
4. if for at least one leg we have $[\rho^i_{min}, \rho^i_{max}] \subset [\rho^i_m, \rho^i_M]$ we bisect the current EB, store the 8 new EBs at the end of the list, and we consider the next EB in the list

The algorithm will stop either when the condition of step 2 is fulfilled (the location is then valid) or when we have considered all the EBs in the list, which means that the location is not valid.

On a SUN Ultra 1 workstation the computation time for determining if a point belongs to the maximal workspace (for which $S_\psi, S_\theta, S_\phi = [0, 360]$) ranges from 40ms to 10s (the later case occurs when the possible orientations set for the location is almost reduced to a point in the orientation space).

3.2 Variant algorithms

A simple variant of this algorithm enables us to verify if **all** the orientations within the orientation space can be reached with valid leg lengths. Basically the algorithm is similar to the previous one except that as soon as we find an EB in the list which satisfies the condition of step 3, then the location is not valid while at step 2 we skip to the next box in the list.

Another variant allows computation of all of the possible orientations which can be reached for the given location of C. The algorithm will return these orientations as a set of 3 intervals $\{S_\psi, S_\theta, S_\phi\}$. Step 2 of the previous algorithm will be modified: instead of exiting at this point we store the orientation part of the current EB in the result set and skip to the next EB in the list. Note that it may be necessary to define a minimal width for the orientation ranges to terminate the computation in a reasonable amount of time: the possible orientations will therefore be determined with an accuracy up to this width.

4 Determining a total orientation workspace

A total orientation workspace (`TOW` for short) is defined as the locations of C for which the leg lengths are valid for any orientation in an orientation space defined by three orientation ranges S_ψ, S_θ, S_ϕ. The *dextrous workspace*, which can be computed exactly [Merlet 1997], is a special case of the `TOW` with $S_\psi = S_\theta = S_\phi = [0, 360]$. Our algorithm will compute an approximation of the `TOW` as a set of EBs whose orientation part is S_ψ, S_θ, S_ϕ. Each EB of this set has a status which may be:

- 1: for any pose within the EB the leg lengths are valid

- 2: the center of the location part of the EB belongs to the `TOW` but some point of its location part may not belong to it

- -2: the center of the location part of the EB does not belong to the `TOW` but some point of its location part may belong to it

Therefore, the computed workspace will be an approximation of the `TOW` whose accuracy will depend on the size and number of the EBs with status 2 and -2. The size of an EB will be here defined as the distance between the center of its location part and its vertices. A threshold on the minimal size of the EB will be fixed before starting the computation. The algorithm will also give guaranteed bounds on the workspace volume. The lower bound will be the sum of the volumes of the EBs with status 1 while the upper bound will be the sum of the volumes of all EBs whatever their status is.

4.1 Bounding box of the `TOW`

First note that we may easily find a bounding box of the `TOW`. Indeed we have:

$$\mathbf{OC} = \mathbf{OA_i} + \mathbf{A_iB_i} + \mathbf{B_iC}$$

Therefore, the distance between point C and point A_i cannot exceed $\rho^i_{max} + d_i$ where d_i is the fixed distance between C and B_i. If $\mathcal{B}_i$ denotes a cube centered at A_i and whose edges have length $\rho^i_{max} + d_i$, then any location of C must belong to all of the six $\mathcal{B}_i$. In other words, the intersection of the $\mathcal{B}_i$ define a geometrical object which bound the `TOW` (and any workspace of the robot). Note that this intersection can be computed as a set of cartesian boxes and that we may impose some constraints on this intersection (for example, consider only its components which are over the base).

4.2 Algorithm principle

Our algorithm will start with a list $\mathcal{S}$ of EBs whose location parts enclose the overall workspace of the robot, as described in the previous section. EB_i will denote the i-th EB of the list while n denotes the total number of EBs in $\mathcal{S}$. The computation will be done with an accuracy ϵ, meaning that the EBs with status 2 or -2 will have a size less or equal to ϵ. We start with $i = 0$:

1. if $i > n$ exit, otherwise compute the lengths bound intervals of the 6 legs for EB_i

2. if for all legs we have $\rho^j_m > \rho^j_{min}$ and $\rho^j_M < \rho^j_{max}$ then EB_i has status 1. Store EB_i, increment i and go to step 1

3. if for at least one leg we have $\rho^j_m > \rho^j_{max}$ or $\rho^j_M < \rho^j_{min}$ then EB_i is outside the `TOW`. Increment i and go to step 1

4. if for at least one leg we have $[\rho^i_{min}, \rho^i_{max}] \subset [\rho^i_m, \rho^i_M]$ then:

(a) if the size of EB_i is lower or equal to ϵ test if the center of its location part belongs to the `TOW` using the variant algorithm described in section 3. If yes the EB has status 2 otherwise it has status -2. Increment i and go to step 1

(b) otherwise bisect EB_i, put the 8 new EBs at the end of $\mathcal{S}$. Increment i, $n = n+8$, and go to step 1

This algorithm has been tested on the SSM robot (described in section 7) for determining the `TOW` with $S_\psi = 0$, $S_\theta = [0, 20°]$, $S_\phi = 0$. Table 1 indicates the computation time according to the desired accuracy together with the total volume of the EBs with status 1,2,-2 (denoted EB^1, EB^2, EB^{-2}). Figure 3 presents a cross-section of the result

ϵ	Time	Volume of EB^1	Volume of EB^2	Volume of EB^{-2}
0.74	7mn	226.7	470.8	900.5
0.37	20mn	407.7	283.7	391.7
0.185	1h18mn	537.7	156.6	183.7
0.0925	5h10mn	612.3	81.7	88.9

Table 1: Computation time of a `TOW` according to the desired accuracy ϵ. The total volumes of EBs with status 1, 2, -2 is also indicated

for $z = 56$ and a 3D view of the final result.

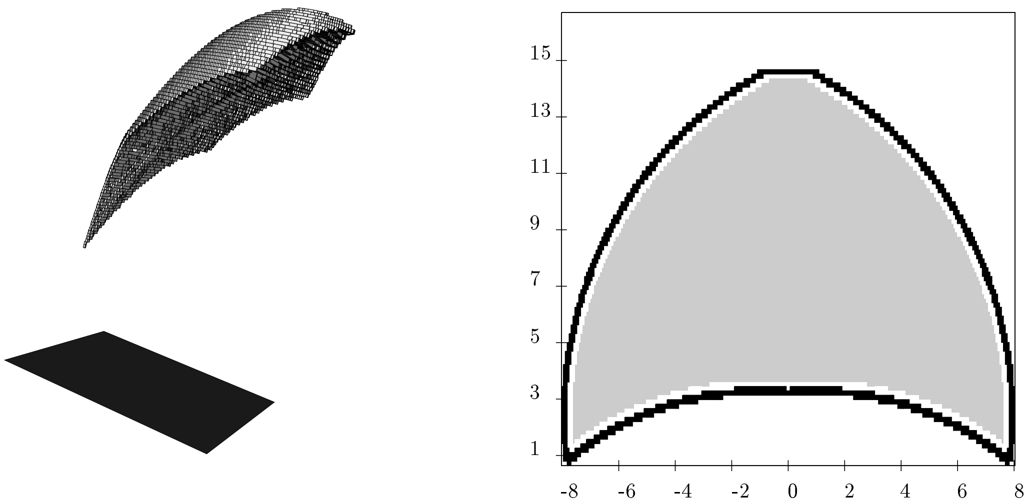

Figure 3: On the left a 3D view of the `TOW` of the SSM for $S_\psi = 0$, $S_\theta = [0, 20°]$, $S_\phi = 0$ (a scale factor of 4 has been applied on the vertical axis). On the right a cross section of this `TOW` for $z = 56$: the color of the EBs is gray, white and black if they have 1, 2, -2.

This algorithm can be adapted to compute a *total translation workspace* defined as all the orientations of the platform that can be reached for any position of C within a given volume. Indeed the new algorithm is obtained by exchanging the role of the location part and orientation part of the EBs. An example of total translation workspace is presented in figure 4. In this example the volume for C is a box.

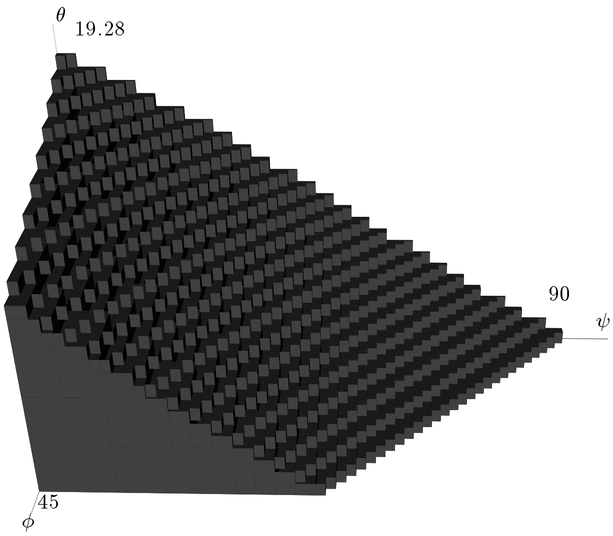

Figure 4: A total translation workspace: here is presented all of the orientations that can be reached by the SSM robot (see section 7) while the center of the platform lies in the box [-0.1,0.1][-0.1,0.1][57,57.1]. The orientation has been restricted to lie within [0,180],[0,90],[0,45] and a scale factor of 3 has been applied on θ

4.3 Workspace verification

The purpose of this section is to describe how it is possible to modify the `TOW` algorithm in order to test if a given 6D workspace is included in the reachable workspace of the robot.

Solving this problem is clearly very useful during the design of a parallel robot. In most applications some "minimal" 6D workspace is specified, while some others requirements (like stiffness, positioning accuracy, minimal platform velocities) are also to be considered. Discarding efficiently the geometries which have not an adequate workspace is therefore of the highest importance to determine the best robot for the task at hand.

4.3.1 Extended box

Suppose that we want to verify that a given EB, EB_0, is included in a `TOW`. This problem can be solved with a few modifications of the `TOW` algorithm:

- first we initialize the list $\mathcal{S}$ with the EB EB_0

- we then proceed with the TOW algorithm with the following modifications:
  - step 3: if an EB in $\mathcal{S}$ is such that for at least one leg we have $\rho_m^j > \rho_{max}^j$ or $\rho_M^j < \rho_{min}^j$, then EB_0 is not included in the TOW as all of the poses of EB_0 are outside the TOW: the algorithm will therefore exit
  - step 4a: the purpose of the algorithm is to give a straight answer on the inclusion of the EB in the TOW: hence the size of the EB is not taken into account and this step is suppressed
  - if there are no more EBs in the list, EB_0 is included in the TOW: indeed EB_0 is the union of all the EBs in the list $\mathcal{S}$ and the algorithm has determined that all these EBs are include in the TOW

4.3.2 General workspace

We may also determine if more complex workspaces are included in a TOW. Let us assume for example that a workspace for C is defined as a 3D reconstruction obtained from a set of polygonal cross-sections. For example, figure 5 shows such a workspace obtained from three horizontal cross-sections. Note that for this example definition of a 3D workspace, it is trivial to determine if the location part of an EB is fully or partly included in the workspace or if it is fully outside the workspace. The TOW algorithm is modified as follow:

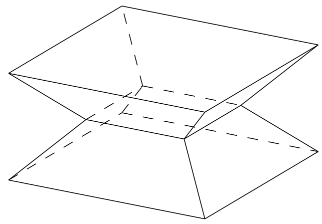

Figure 5: An example of possible workspace for C: it is obtained as a 3D reconstruction from three horizontal polygonal cross-sections

- the list $\mathcal{S}$ is initialized with an EB whose location part is the bounding box of the volume or with a list of EBs such that the union of their location parts enclose the volume
- step 3: the purpose of this step is to determine if the current EB is fully inside, fully outside or partially inside the TOW. As the union of the EBs in the list enclose the workspace we must therefore test if the location part of the current EB is included in the workspace. The current EB may be fully inside, fully outside or partly inside the workspace. So we add to step 3 the following steps:
  - if the location part is fully included in the workspace we continue the algorithm as we have to determine if the current EB is included in the TOW

– if the location part is partly included in the workspace we test if the current EB is included in the TOW. If it is so the part of the EB which is inside the workspace is also in the TOW: we may move to the next EB in the list. If the current EB is not in the TOW its parts which are inside the workspace may still belong to the TOW, so we proceed directly to step 4b and bisects the location part.

– if the location part is outside the workspace we don't have to test if it belongs to the TOW, so we skip the current EB and move to the next EB in the list

- if an EB in $\mathcal{S}$ is inside the workspace (see the above inclusion test) and is such that for at least one leg we have $\rho^j_m > \rho^j_{max}$ or $\rho^j_M < \rho^j_{min}$, the EB does not belong to the TOW and consequently a part of the workspace is not included in the TOW: the algorithm exits

- step 4a: the purpose of the algorithm is to give a straight answer on the inclusion of the workspace in the TOW: hence the size of the EB is not taken into account and this step is suppressed

- if there are no more EBs in the list, the workspace is included in the TOW: indeed the workspace is enclosed in the union of all the EBs in the list $\mathcal{S}$ and the algorithm has determined that all these EBs are include in the TOW

5 Determination of an inclusive orientation workspace

The *inclusive orientation workspace* (IOW for short) is defined as all of the possible locations of C which can be reached with at least one orientation among an orientation space defined by 3 ranges S_ψ, S_θ, S_ϕ. The *maximal or reachable workspace* is a particular case of IOW for which the three ranges are $[0, 360]$ (consequently any orientation is allowed and we seek all the locations that can be reached by C). The problem of determining the reachable workspace for planar robot (a problem which has been solved exactly in [Merlet 1998]) has been addressed by Husty [Husty 1996], Kumar [Kumar 1992] and Luh [Luh 1996]. For 6 d.o.f. Gough-type robots Kim [Kim 1997] computes a rough approximate of the maximal workspace, Haugh [Haugh 1995, Haugh 1998] uses a continuation method to determine approximatively the border of the reachable workspace while Pooran [Pooran 89] proposes a numerical search procedure. To the best of the author knowledge there is no reference whatsoever for the determination of an IOW for which the orientation of the platform is a subset of all the possible orientations.

5.1 Algorithm principle

Our algorithm will compute the IOW as a set of EBs whose orientation part is included in S_ψ, S_θ, S_ϕ. Each of the EBs will have a status as presented for the TOW with an additional status of -1 which indicates that the EB does not belong to the IOW. Note that we will still obtain guaranteed bounds for the workspace volume by considering the volume of the EBs with status 1 (lower bound) or 1, 2, -2 (upper bound).

The algorithm is basically similar to the TOW algorithm except that now we bisect also the orientation part of the EBs. The list $\mathcal{S}$ is initialized with a bounding box of the inclusive orientation workspace as computed in section 4.1. We start with $i = 0$:

1. if $i > n$ then exit
2. if the status of EB_i is equal to -1 increment i and go to step 1
3. if among the set of EBs=$\{EB_0, EB_1, ..EB_{i-1}\}$ we have an EB with status 1 whose location part includes the location part of EB_i, then EB_i is assigned the status -1. Increment i and go to step 1
4. compute the lengths bound interval for EB_i
   (a) if for all the legs we have $\rho^j_m > \rho^j_{min}$ and $\rho^j_M < \rho^j_{max}$ then EB_i is assigned the status 1. If any EB_j in $\mathcal{S}$ has a location part included in the location part of EB_i, then EB_j is assigned the status -1. Increment i and go to step 1
   (b) if for at least one leg we have $\rho^j_m > \rho^j_{max}$ or $\rho^j_M < \rho^j_{min}$, then EB_i is assigned the status -1. Increment i and go to step 1
   (c) if for at least one leg we have $[\rho^i_{min}, \rho^i_{max}] \subset [\rho^i_m, \rho^i_M]$:
      i. if the size of EB_i is lower or equal to ϵ we test if the center of its location part belongs to the IOW:
         A. if the center belongs to the IOW the EB is assigned the status 2. Then we check if any EB_j in$\{EB_0, EB_1, ..EB_{i-1}\}$ has a status -2 and has a location part included in the location part of EB_i, in which case EB_j is assigned the status -1.
         B. if the center does not belong to the IOW EB_i is assigned the status -2
         C. increment i and go to step 1
      ii. otherwise we bisect EB_i and the resulting 64 new EBs are put at the end of $\mathcal{S}$, $n = n + 64$, increment i and go to step 1

This algorithm will stop when the last box of $\mathcal{S}$ has been considered: the IOW will be constituted of all the EBs of $\mathcal{S}$ with status 1, 2 or -2.

The computation time of an IOW will be clearly higher than for a TOW as the bisection process will generate a larger number of EBs (64 instead of 8).

5.2 Managing the bisection process

As the bisection is applied on both the location part and the orientation part of the EB it is necessary to carefully manage the bisection process. For example, assume that the location part of the initial EB is small while its orientation part is large. If we bisect at the same time both parts of the EB, then the size of the location part will quickly reach the desired accuracy while the amplitude of the orientation part will still be large: therefore the algorithm will end up with very few EBs with status 1, most having status 2 or -2.

One possible way to deal with this problem is to bisect the location part of an EB only if the size of the orientation part is lower than a given threshold.

Another possible way to reduce the total number of boxes is not to put the 64 boxes created by the bisection process at the end of the list but to substitute the current box by one of them and put the 63 at the end of the list. In our implementation we have chosen to substitute the current box by the EB among the 64 new EBs whose lengths bound interval is the closest to the allowed interval: clearly this sub-box has the highest probability to be inside the `IOW`. If this is the case the boxes in the list which have the same translation part but a different orientation part will just be discarded.

5.3 Computation time, implementation and examples

5.3.1 Implementation

Our software enable us to generate the `IOW` and `TOW` incrementally. We usually compute first a rough approximation of the workspace using a large value for ϵ with, as starting point, the set of EBs obtained as described in section 4.1. The result of this computation is stored in a file which is used as input for a second step in which we use a lower accuracy (usually half the accuracy of the previous step): the EBs of the input file with status 1 are just stored as such in the result file.

According to the respective size of the location part and the orientation part we may get a large number of EBs with status -2 which in fact do not belong to the `IOW` but have been kept because the size of their orientation part was too large to eliminate them. Indeed over a large orientation workspace each individual leg length may be valid for some orientations but there will be no orientation for which all the 6 leg lengths will be valid at the same time. In order to reduce the number of EBs with status -2 it may be interesting to refine the result. This can be done by considering only the EBs with status -2 in a result file and bisect their orientation part until the amplitude of the ranges is lower than a fixed, small threshold. During this process many EBs will be deleted (as the lengths bound interval will have a lower amplitude) and an EBs will be kept in the result file if and only if it remains one EB with an orientation part having an amplitude lower than the threshold: this is the *refinement* process.

A similar process may be used for the EBs having status 2. Here the purpose is to determine which EBs among them fully lie within the `IOW`. In that case the process will lead to a larger computation time as during the bisection all the EBs with a low amplitude of the orientation part are to be tested until either one of them get status 1 or all the possible EBs have been tested (in the previous case as soon as one of the EB has a small orientation part and still its center does not belong to the `IOW` the refinement is finished).

5.3.2 Computation time

Table 2 indicates the result for the computation of a cross-section of the maximal workspace at $z = 50$ of the SSM (this robot geometry is described in section 7). The indicated computation time for each accuracy (except the first one) is obtained when using as input file the result of the previous step.

ϵ	Time	Area of EB^1	Area of EB^2	Area of EB^{-2}
2.7	2h35mn	1256.07	799.32	1438.8
1.35	4h08mn	1735.66	428.2	999.1
0.338	10h31mn	2059.66	117.75	782.9
0.1693	15h	2102.399396	75.114367	547.2

Table 2: Computation time of a cross-section of the maximal workspace as a function of the accuracy

Figure 6 presents cross-sections of the maximal workspace of the SSM for $z = 50$ with various accuracies.

5.4 Computing a full description of the workspace

A variant of the previous `IOW` algorithm allows theoretically to compute a full description of the workspace of a given robot as a set of EBs which will describe each possible location of C together with all the possible orientation ranges for the location part (hence the result may contain different EBs with the same location part but different orientation part). This variant is obtained by removing all of the statements in the previous algorithm which attribute a status of -1 to an EB. Note that in practice this algorithm can be used only for a robot having a very small workspace as a large number of EBs will be generated.

6 Heuristics for speeding the computation

Clearly the algorithms presented in the previous sections may lead to a large computation time: this is mostly due to the fact that when testing an EB we consider each leg length individually. Indeed it will frequently occur that although the lengths bound interval of each leg may have a non empty intersection with the interval $[\rho_{min}, \rho_{max}]$ there will be no pose within the EB for which the intersection will be not empty *at the same time*. Such an EB will therefore be bisected until a final decision can be made on its status, thereby leading to a large number of EBs.

But we may improve the computation time of our algorithms by using various heuristics.

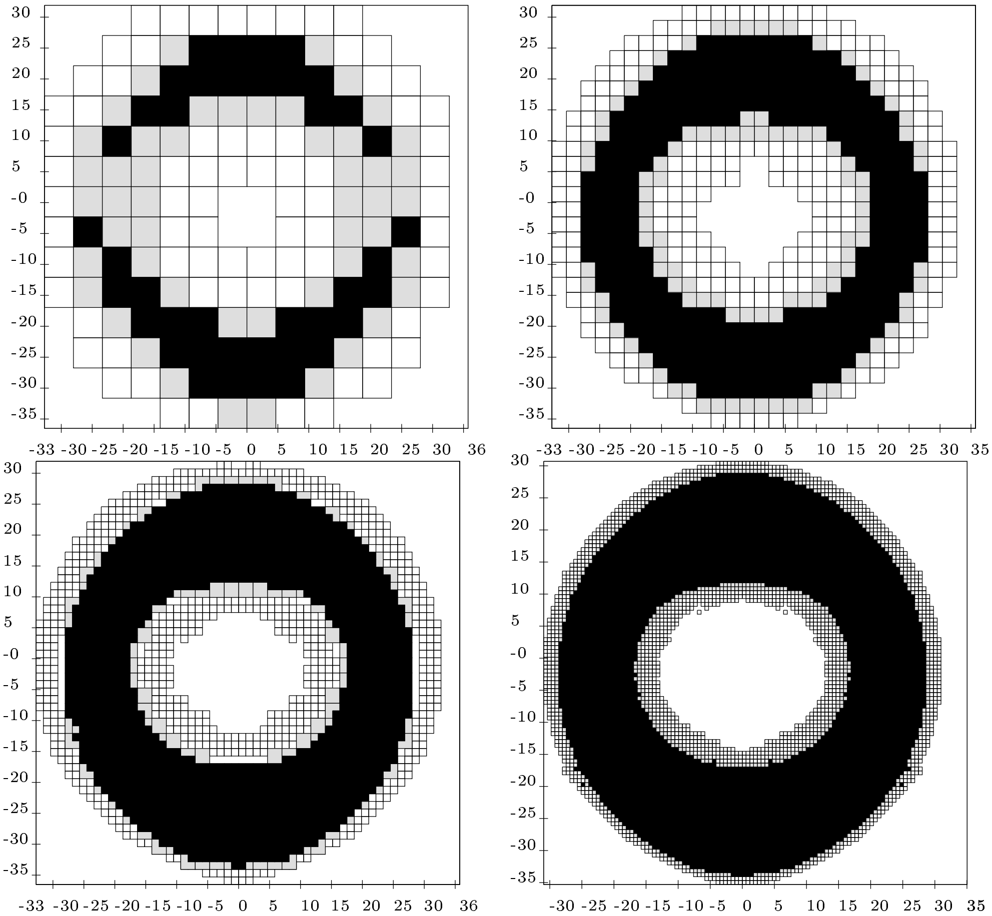

Figure 6: Cross-sections at $z = 50$ of the maximal workspace of the SSM with accuracy 3.36, 1.68, 0.84, 0.42. The black area lie fully in the workspace, EBs with status 2 are gray while EBs with status -2 are white.

6.1 Improving the initial guess

The first heuristic will improve the initial estimation of the workspace as described in section 4.1. The leg length ρ is obtained as:

$$\rho^2 = ||\mathbf{AB}||^2 = ||\mathbf{OA} + \mathbf{OC} + \mathbf{CB}||^2 \tag{1}$$

which may be written as:

$$\rho^2 = ||\mathbf{OC}||^2 + U \tag{2}$$

where U is a quantity obtained as a combination of the scalar product of $\mathbf{OA}, \mathbf{CB}, \mathbf{OC}$ and the square of the norm of $\mathbf{OA}, \mathbf{CB}$. Using this equation and the constraint on the leg lengths we deduce:

$$\rho^2_{min} - U \leq ||\mathbf{OC}||^2 \leq \rho^2_{max} + U \tag{3}$$

Using interval analysis it is possible to compute a lower and an upper bound of U and therefore bounds on $||\mathbf{OC}||^2$. For each EB in our initial list we are able to compute the minimum and maximum distance between O and any point in the location part of the EB. Then we reject the EB having a distance which is not compatible with the distance as computed with equation 3.

6.2 Test for fixed orientations

A second possible heuristic makes use of the possibility to compute exactly and efficiently the lengths bound interval for an EB whose orientation part is reduced to a fixed orientation. When considering an EB in the `TOW` and `IOW` algorithms we select a few orientations within the orientation part and compute the lengths bound intervals for these orientations:

- for the `TOW` algorithm if at least one of the lengths bound intervals has no intersection with the limits range, then the EB does not belong to the `TOW`
- for the `IOW` algorithm if one of the lengths bound intervals is valid then the EB belongs to the `IOW`

6.3 The γ test

Let us define γ as the angle between $\mathbf{AC}$ and $\mathbf{CB}$ (figure 7). We have:

$$\cos\gamma = \frac{||\mathbf{AC}||^2 + ||\mathbf{CB}||^2 - \rho^2}{2||\mathbf{AC}||||\mathbf{CB}||} \quad \text{and} \qquad \cos\gamma = \frac{\mathbf{AC}.\mathbf{CB}}{||\mathbf{AC}||||\mathbf{CB}||} \tag{4}$$

In these equations $||\mathbf{CB}||^2$ is a constant. For a given EB it is possible to compute exactly the bounds on $||\mathbf{AC}||$ and using the constraints on the leg lengths and interval analysis the first equation of (4) gives a first set of bounds for $\cos\gamma$, while the second equation enables the establishment of another set of bounds for the same quantity. Then the EBs for which the two sets of bounds are incompatible are rejected.

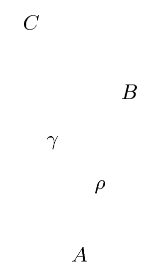

Figure 7: Definition of the γ angle

6.4 Constraints on two legs

Let us consider two legs i, j of the parallel robot. Using equation (2) we get:

$$\rho_i^2 - \rho_j^2 = U_i - U_j \tag{5}$$

Using the constraints on the leg lengths we have:

$$\rho_{min_i}^2 - \rho_{max_j}^2 \leq U_i - U_j \leq \rho_{max_i}^2 - \rho_{min_j}^2 \tag{6}$$

For a given EB we may compute bounds on $U_i - U_j$ using interval analysis and reject the EBs which does not satisfy equation (6). Note that we may further refine this test after having computed the lengths bound interval for this EB using the following equation:

$$\rho_{m_i}^2 - \rho_{M_j}^2 \leq U_i - U_j \leq \rho_{M_i}^2 - \rho_{m_j}^2 \tag{7}$$

This test will be performed for each pair of legs.

7 Comparison of the workspaces of different robot geometries

In this section we consider four different geometries of parallel robot and we intend to compare their respective workspaces. For the three first geometries the attachment points lie on a circle of radius 13 for the base and on a circle of radius 7 for the platform. The attachments points on the base and platform are distributed on their circle according to figure 8. For the SSM (Symmetric Simplified Manipulator) robot we use $\alpha = \beta = 20°$, for the TSSM (Triangular Symmetric Simplified Manipulator) robot we have $\alpha = 20°, \beta = 0$ (the platform is a triangle) while for the MSSM (Minimal Symmetric Simplified Manipulator) robot $\alpha = \beta = 0$ (both the base and platform are triangles). The fourth robot we will study is the MSP robot presented by Stoughton [Stoughton 1993]: the attachment points are successively on two different circles of radii 11, 13 for the platform and 5, 7 for the platform. The MSP geometry has been obtained using an optimization

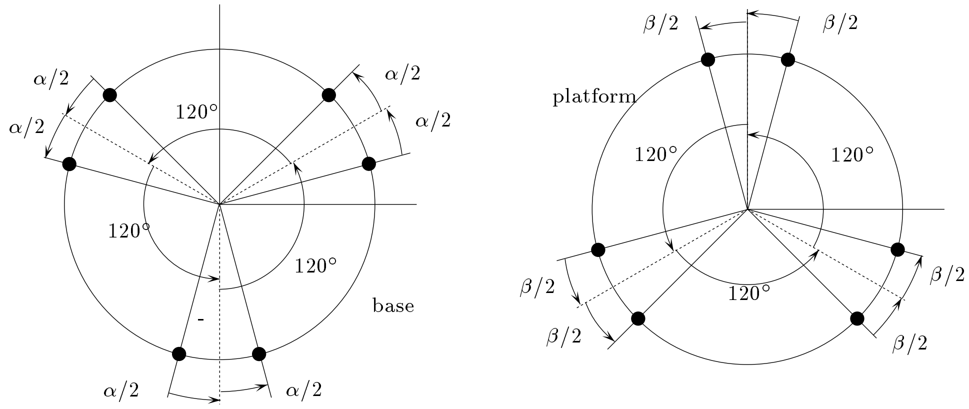

Figure 8: The distribution of the attachment point on the base and platform

process in which the cost function was a weighted sum of the workspace volume and the dexterity. Stoughton has shown that if the weight on the volume was larger than the weight on the dexterity, then the SSM was the optimal design. On the other hand with a larger weight on the dexterity than on the workspace volume the MSP is obtained as the optimal design. The four different robots are presented in figure 9. For all four robots

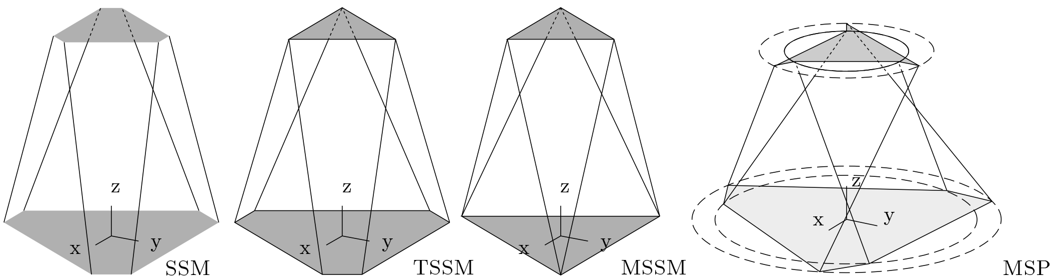

Figure 9: The four different robots whose workspaces will be compared

the minimum leg lengths is 55 and the maximum leg length 60.

7.1 Comparison of the COW workspace

Table 3 indicates the volume of the COW for different orientations of the moving platform, together with the ratio between the volumes of the different robots. The SSM has the larger workspace, followed by the TSSM and then the MSSM. It must be noted that the ratio workspace volume of the SSM divided by the the workspace volume of the TSSM is approximatively constant at 1.3. Similarly this ratio between the SSM and the MSSM is approximatively equal to 1.7 and as a consequence the ratio between the TSSM and MSSM is approximatively equal to 1.25. The MSP has a workspace volume which is in general somewhat smaller than the MSSM and has anyway a largely smaller workspace

Orientation (ψ, θ, ϕ)	MSP	MSSM	TSSM	SSM	$\frac{TSSM}{MSSM}$	$\frac{SSM}{MSSM}$	$\frac{SSM}{TSSM}$
0,0,0	861	950	1214	1576	1.3	1.7	1.3
0,5,0	748	766	968	1271	1.3	1.7	1.3
5,0,0	765	924	1177	1529	1.3	1.7	1.3
5,5,0	685	749	947	1245	1.3	1.7	1.3
5,5,5	613	706	894	1180	1.3	1.7	1.3
0,10,0	502	434	542	745	1.2	1.7	1.4
10,0,0	670	848	1082	1406	1.3	1.7	1.3
10,10,0	405	412	522	720	1.3	1.7	1.4
10,10,10	355	363	468	661	1.3	1.8	1.4

Table 3: Volume of the constant orientation workspace of the different robots for various orientations and their ratio (angles in degrees)

volume than the SSM, as claimed by Stoughton.

7.2 Comparison of the `TOW` workspace

Table 4 indicates the volume of the `TOW` for the orientation ranges $[0, 10°]$, $[0, 10°]$,$[0, 10°]$ which has been computed with an accuracy of 0.067 for the SSM and TSSM, 0.0318 for the MSSM and MSP. The volume of the EBs with status 1,2, -2 is indicated in this table. The real workspace volume will be therefore greater or equal to the volume of the EBs with status 1 while it will be lower or equal to the sum of the volumes of the EBs, whatever their status: hence the the workspace volume V_{SSM} for the SSM will be such that $402.2 \leq V_{SSM} \leq 449$. The accuracies for the computation has been chosen so that there will be not intersection between the different volume ranges: hence any comparison between the different volumes is guaranteed to be exact.

	Volume of EB^1	Volume of EB^2	Volume of EB^{-2}	Total
SSM	402.3	23.1	23.6	449
TSSM	294	17.9	18.5	330.5
MSSM	229.8	7	7.2	244
MSP	233.3	7	7.1	247.4

Table 4: The volumes of the total orientation workspace for the different types of robot. The real volume if guaranteed to be greater or equal to the volume of EB^1 and lower or equal to the Total volume. The accuracy used for this computation was such that there is no intersection between the different volume ranges, which allows for a comparison between the volumes.

We notice that the SSM has still the larger workspace, followed by the TSSM and the MSSM (the MSP has almost the same workspace volume than the MSSM). Table 5 indicates the minimal and maximal ratio between the workspace volume of the different robots (the minimal ratio is obtained by taking the volume of the EBs with status 1 over the total volume while the maximal ratio is the total volume over the volume of the EBs with status 1). Note that these ratio are consistent with the ratio found for the `COW`.

$\frac{TSSM}{MSSM}$	$\frac{TSSM}{MSP}$	$\frac{SSM}{MSSM}$	$\frac{SSM}{TSSM}$	$\frac{SSM}{MSP}$
1.20-1.44	1.19-1.42	1.65-1.95	1.217-1.527	1.625-1.925

Table 5: Minimal and maximal ratio of the `TOW` volume of the different robots

7.3 Comparison of the maximal workspace

Table 6 indicates the volume of the maximal workspace for the four different robots. This computation has been done with an accuracy of 0.12 for the MSP, 0.11 for the MSSM, 0.11 for a first part of the TSSM and 0.43 for a second part and 0.22 for the SSM. Here also the accuracies for the computation has been chosen so that there will be not intersection between the different volume ranges, thus enabling a comparison between the volumes.

The volume of the EBs with status 1,-2, 2 is indicated in this table, with the corresponding number of boxes in parenthesis. Table 7 present the minimal and maximal ratio of the workspace volume. According to these results we may classify the different

	Volume of EB^1	Volume of EB^2	Volume of EB^{-2}	Total
SSM	7848.2 (114328)	900.6 (93284)	1049.4 (109704)	9798.2
TSSM	5182 (298182)	290.8 (202578)	883.7 (104834)	6356.5
MSSM	4144.3 (203342)	331.3 (145278)	419 (279130)	4894.6
MSP	3524.2 (131366)	253.4 (115538)	361.8 (165298)	4139.4

Table 6: The volume of the maximal workspace, divided according to the status of the EB (the numbers in parenthesis indicate the respective number of EBs)

$\frac{TSSM}{MSSM}$	$\frac{SSM}{MSSM}$	$\frac{SSM}{TSSM}$	$\frac{TSSM}{MSP}$	$\frac{SSM}{MSP}$
1.06-1.53	1.6-2.36	1.23-1.9	1.25-1.8	1.9-2.8

Table 7: Minimal and maximal ratio of the maximal workspace volume of the different robots

geometries by increasing workspace volume as MSP, MSSM, TSSM, SSM, the workspace volume of the SSM being approximatively twice the workspace volume of the MSP. Hence a small change in the geometry of the robot may lead to large variation of workspace volume.

Figure 10 presents horizontal cross-sections of the maximal workspace of the SSM, TSSM, MSSM at their nominal altitude (i.e. for the value of z obtained when the leg lengths are all at their middle value).

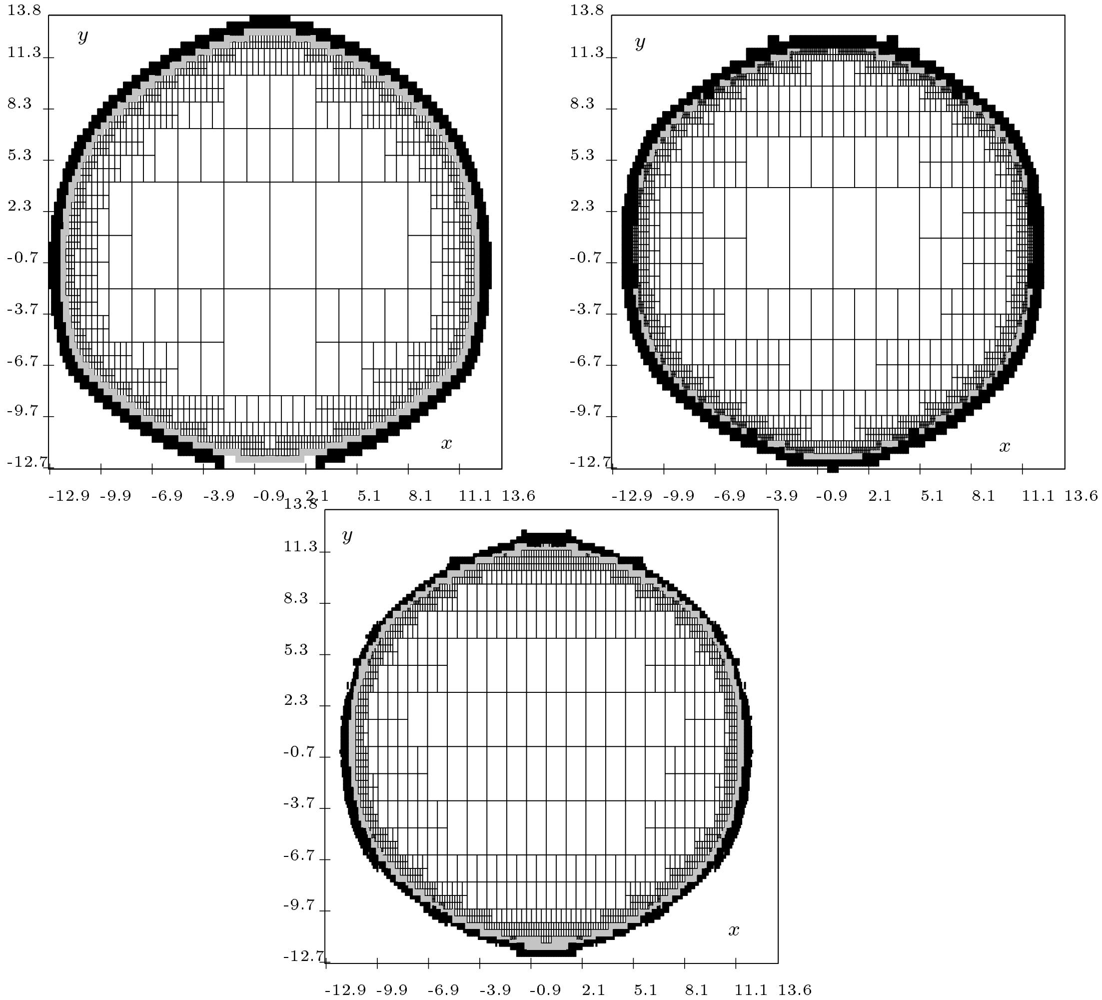

Figure 10: From top to bottom horizontal cross-sections of the maximal workspace of the SSM, TSSM, MSSM at their nominal altitude. The black area represents the EBs with status -2, the gray area the EBs with status 2.

7.4 Comments

Some authors have claimed that a parallel manipulator with common attachment points (like the TSSM and MSSM) have a larger workspace than a more general geometry as the common points leads to less risks of interference between the legs (see for example [Kurtz 1992]). The underlying assumption in such claim is that all the structures have roughly identical workspace volume when considering only the limitation due to the leg lengths. The previous sections show clearly that this is not the case, the SSM having a much more larger intrinsic workspace than the MSSM.

8 Passive joints limits

In the previous sections we have taken into account only the constraints on the leg lengths. In practice however we have to consider that there are also limitations on the motion of the passive joints at A, B. In [Merlet 1997] we have introduced a possible way to describe these limitations. We assume that the designer is able to define a convex pyramid with planar facets whose apex is located at the joint center such that if the joint limits are fulfilled then leg AB will be located inside the pyramid. Figure 11 shows an example of such pyramid. Let $\boldsymbol{n_i}$ be an unit vector perpendicular to the planar facet i. For the joint with center at A the vector $\boldsymbol{n_i}$ is constant while for the joint at B the vector $\boldsymbol{n_i}$ is a constant vector in the mobile frame. If leg AB lie inside the pyramid, then for all facets of the pyramid with apex at A we should have:

$$\boldsymbol{AB}.\boldsymbol{n_i} \leq 0 \tag{8}$$

while for the pyramid with apex at B we should have:

$$\boldsymbol{AB}.R\boldsymbol{n_i} \geq 0 \tag{9}$$

where R is the rotation matrix describing the orientation of the platform.

If we consider now a box workspace interval analysis enable to determine for each facet of the pyramid bounds on the extremum of the quantity appearing at the the left side of equations (8,9). For example for equation 8 we have to evaluate $\mathbf{AB}.\mathbf{n_i}$ which may be written as:

$$\mathbf{AB}.\mathbf{n_i} = \mathbf{AO}.\mathbf{n_i} + \mathbf{OC}.\mathbf{n_i} + \mathbf{CB}.\mathbf{n_i}$$

In this expression $\mathbf{AO}.\mathbf{n_i}$ has a constant value, while $\mathbf{OC}.\mathbf{n_i}$ is only dependent upon the location of C. In a similar way, $\mathbf{CB}.\mathbf{n_i}$ is only dependent upon the orientation of the platform.

Using this possibility we may design an algorithm which verifies if a given box workspace EB_0, part of the reachable workspace due to the leg lengths constraints, belongs to the real reachable workspace of the robot, also taking into account the constraints on the joint limits. We will describe this algorithm for dealing with the constraints on the joints centered at A, dealing with constraints on the joint at B being a simple extension. The algorithm is described by the following sequence:

1. compute the extremum U_i^m, U_i^M of $U_i = \boldsymbol{AB}.\boldsymbol{n_i}$ for each facet of the pyramid
   - if for all i we have $U_i^M < 0$, then EB_0 belongs to the reachable workspace
   - if for at least one i we have $U_i^m > 0$, then EB_0 does not belong to the reachable workspace

2. if we have $U_i^M < 0$ for some joints and $U_i^M > 0, U_i^m < 0$ for some others then
   - bisect both the location part and orientation part of EB_0 and put the resulting boxes in a list
   - compute U_i^m, U_i^M for each box of the list
     - if for one box $U_i^m > 0$, then EB_0 does not belong to the reachable workspace
     - if for all i $U_i^M < 0$ skip the box
     - if $U_i^M > 0, U_i^m < 0$, bisect the box and put the resulting box at the end of the list
3. if we have considered every box of the list, then EB_0 belongs to the reachable workspace

Note that the previous algorithm will also permit checking if a given location of C belongs to a given `TOW`. Furthermore it can be added to the algorithm described in section 4 in order to compute in one step the `TOW` taking into account both the constraints on the leg lengths and on the motion of the passive joints.

8.1 Computing a `TOW`

Now assume that we have computed a `TOW` with a given accuracy ϵ using the algorithm described in section 4. We may want to update this `TOW` by taking into account the joint limits at A. Before describing the procedure we will present a preliminary remark. Let us assume that the orientation part of a box workspace is reduced to one unique orientation. As $\boldsymbol{AB}.\boldsymbol{n_i}$ is linear in term of the coordinates of C, if $U_i^m > 0$ for at least one i, then the box is fully outside of any `TOW` whose orientation ranges includes the orientation of the box.

The procedure for updating the initial `TOW` will consider sequentially each box of a list which is initialized with the boxes of the initial `TOW`.

If the current box has status 1 we will first compute U_i^m, U_i^M for this box. If $U_i^M < 0$ for all i, then the box belongs to the new `TOW`. If $U_i^m > 0$ or $U_i^M > 0, U_i^m < 0$ for at least one i and if the size of the box is greater than ϵ we bisect the location part of the box and put the resulting boxes (which have status 1) at the end of the list. If the size of the box is lower than ϵ we check if the center of the box belongs to the new `TOW` and the box will get status -2 or 2 according to the result.

If the current box has status 2 or -2 (and consequently its size is lower than ϵ) we check if the center of the box belongs to the new `TOW` and the box will get status -2 or 2 according to the result.

At this stage the overall volume of the new `TOW` will be the same as the initial `TOW` as no boxes are deleted in the process: for a given box either its status is modified or it is bisected, the resulting boxes being included in the result. To improve this result we will use the preliminary remark: for each box we will select a few orientations among the

	Volume of EB^1	Volume of EB^2	Volume of EB^{-2}	Total
SSM	192.2	9	9.3	210.5
TSSM	179.1	8.8	9	196.9
MSSM	163.5	8.3	8.6	180.4

Table 8: The volume of the TOW for ψ, θ, ϕ in $[0, 10°]$, taking into account both the constraints on the leg lengths and the possible motion of the passive joints at A.

orientation ranges of the TOW and test if for this orientation of the platform $U_i^m > 0$ for at least one i, in which case the box is deleted.

In order to determine the influence of the joint limits we have applied the previous procedure on the TOW computed in section 7.2. For each robot it is assumed that there are constraints only on the joints at A and that the axis of the corresponding pyramid is the axis of the leg for the pose corresponding to the mid-length of the leg. We use pyramids with 10 facets, the angle between each facet and the pyramid axis being 8 degree (figure 11). The results are presented in table 8. It may be seen that the SSM

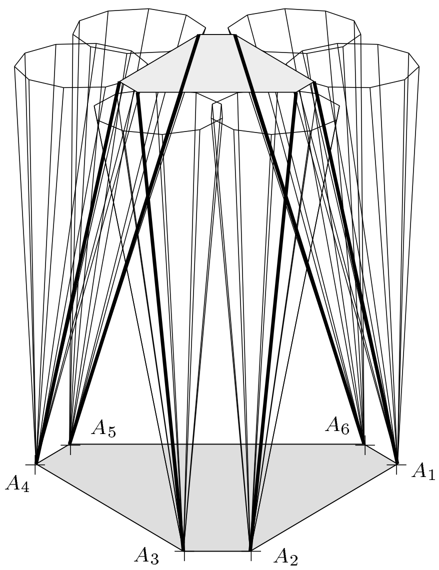

Figure 11: The pyramids with 10 facets describing the possible motion of the passive joints at A of the SSM.

is more sensitive to joint limits than the TSSM and MSSM: the SSM has now only a slightly larger workspace volume than the TSSM volume while the difference between the workspace volume of the SSM and MSSM has been slightly reduced. Figure 12 presents the differences between an horizontal cross-section of the TOW of the SSM if we take into account the joint limits or not.

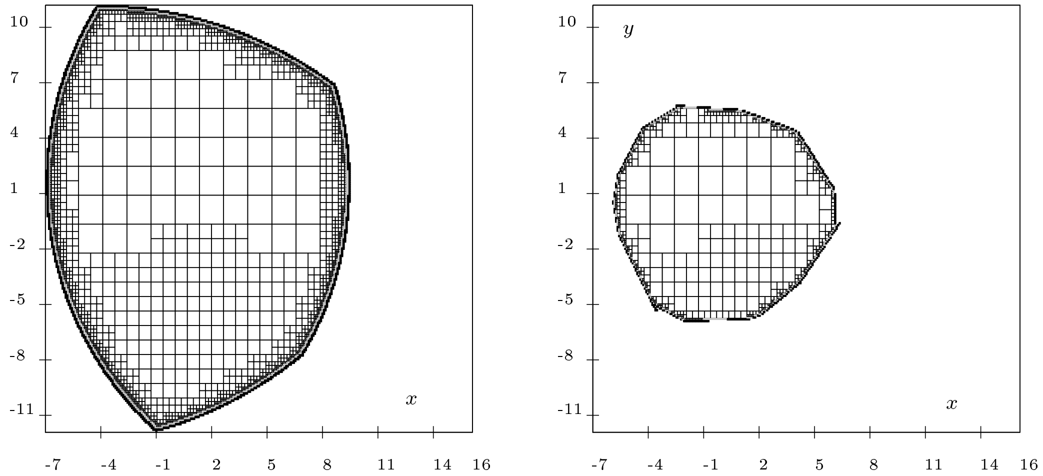

Figure 12: On the left a horizontal cross-section of the TOW of the SSM without taking into account the joint limits. On the right the same cross-section with the effect of the joint limits

8.2 Computing an IOW

Similarly it is possible to update an IOW result file or to compute an IOW which takes into account both the leg length constraints and the passive joints limits.

Boxes with status 1 appear in a result file with two types of orientation parts: an unique orientation which indicates that the box was found inside the IOW for this orientation or a set of orientations. In the first case, we will check if the joint limits are correct for the specific orientation. If this is the case the box belongs to the new IOW, otherwise (or if the orientation part of the box is a set) we will update the orientation part of the box to the full orientation set of the IOW. Then we will start bisecting the orientation part of this box until either a box is found which satisfies the leg length constraints and the joint limits or the amplitude of the orientation part is lower than a given threshold. If no correct box is found and if the size of the box is greater than the accuracy we will bisect the translation part of the box and start again. If the size of the box is lower than the accuracy we will consider a box EB_0^s whose location part is reduced to the center of the initial box and whose orientation part is the full orientation set of the IOW. We then bisect this orientation part until we either find a box belonging to the IOW or we determine that the box center does not belong to the IOW. In the first case the initial box will get status 2 and in the later case it will get status -2.

For the box having initially status 2 we first check if a box having the same location part, but having as the orientation part the full orientation set of the IOW, does not belong to the IOW: we compute the U_i^m, U_i^M for the box and if $U_i^m > 0$ for all i, then the box is outside the IOW. In that case the initial box is discarded. If this check does not eliminate the box we use the same algorithm than described previously except that we start directly at the creation of the box EB_0^s. If during the bisecting process we find out that the box center belongs to the new IOW, the initial box will keep its 2 status otherwise its new status will be -2.

For the box having initially status -2 the algorithm is the same except that we don't have to check the leg lengths.

Note that few boxes with status 2 or -2 will be eliminated in the process. A possible way to decrease the number of boxes will be to consider a set of boxes with the same location part but with different orientation parts, covering the full orientation set of the IOW. Then we check the U_i^m of all these boxes and if for all i and for all boxes the U_i^m is greater than 0, then we discard the box (increasing the number of boxes will reduce the error we make when computing the U_i using interval analysis).

9 Conclusion

The algorithms presented in this paper, mostly based on interval analysis and a bisection process, constitute a first approach to solve the remaining problems regarding the workspace computation of 6 d.o.f parallel robot. The TOW algorithms which enable either the verification of whether a given 6D workspace is included in the reachable workspace or the determination of the locations of C which can be reached with every orientation in a set, are computer efficient and can be used during the design process. On the other hand the IOW algorithm involves a larger computation time but it will not be used in the iterative design process: indeed it will be run only once to get a better characterization of the workspace of a given robot.

As we deal with 6D workspaces we have a representation problem which has been solved by computing projections of the workspace. In this paper, we have favored a representation based on the location of the center of the platform (except for the total translation workspace) but the presented algorithms may be easily easily extended to deal with any 3D projection.

The algorithms have been presented for the Gough-type parallel robot but may be extended to other types of parallel robot. Future work will have as objectives:

- to improve the existing algorithms by considering more than one leg when computing the lengths bound intervals

- to take into account leg interference

10 Acknowledgment

Part of this work was supported by the European ESPRIT project FRISCO (LTR 21.024).

References

[Gosselin 1990] Gosselin C. Determination of the workspace of 6-dof parallel manipulators. *ASME J. of Mechanical Design*, 112(3):331–336, September 1990.

[Gosselin 1992] Gosselin C., Lavoie E., and Toutant P. An efficient algorithm for the graphical representation of the three-dimensional workspace of parallel manipulators. In *22nd Biennial Mechanisms Conf.*, pages 323–328, Scottsdale, September, 13-16, 1992.

[Haugh 1995] Haugh E.J., Adkins F.A., and Luh C.M. Domain of operation and interference for bodies in mechanisms and manipulators. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 193–202. Kluwer, 1995.

[Haugh 1998] Haugh E.J., Adkins F.A., and Luh C.M. Operational envelopes for working bodies of mechanisms and manipulators. *ASME J. of Mechanical Design*, 120(1):84–91, March 1998.

[Husty 1996] Husty M.L. On the workspace of planar three-legged platforms. In *World Automation Congress*, volume 3, pages 339–344, Montpellier, May, 28-30, 1996.

[Jo 89] Jo D.Y. and Haug E.J. Workspace analysis of closed loop mechanisms with unilateral constraints. In *ASME Design Automation Conf.*, pages 53–60, Montréal, September, 17-20, 1989.

[Kim 1997] Kim D.I., Ching W.K., and Youm Y. Geometrical approach for the workspace of 6-dof parallel manipulators. In *IEEE Int. Conf. on Robotics and Automation*, pages 2986–2991, Albuquerque, April, 21-28, 1997.

[Kumar 1992] Kumar V. Characterization of workspaces of parallel manipulators. *ASME J. of Mechanical Design*, 114:368–375, September 1992.

[Kurtz 1992] Kurtz R.L. and Hayward V. Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy. *IEEE Trans. on Robotics and Automation*, 8(5):644–651, October 1992.

[Luh 1996] Luh C.M., Adkins F.A., Haugh E.J., and Qiu C.C. Working capability analysis of Stewart platforms. *J. of Mechanical Design*, 118(2):221–227, June 1996.

[Masory 1993] Masory O., Wang J., and Zhuang H. On the accuracy of a Stewart platform-part II: Kinematic calibration and compensation. In *IEEE Int. Conf. on Robotics and Automation*, pages 725–731, Atlanta, May, 2-6, 1993.

[Merlet 1997] Merlet J-P. *Les Robots parallèles*. Hermès, Paris, 1997.

[Merlet 1997-2] Merlet J-P. Estimation efficace des caractéristiques de robots paralèlles: Extremums des raideurs et des coordonnées, vitesses, forces articulaires et singularités dans un espace de travail en translation. Research Report 3243, INRIA, September 1997.

[Merlet 1998] Merlet J-P., Gosselin C., and Mouly N. Workspaces of planar parallel manipulators. *Mechanism and Machine Theory*, 33(1/2):7–20, January 1998.

[Moore 79] Moore R.E. *Methods and Applications of Interval Analysis.* SIAM Studies in Applied Mathematics, 1979.

[Pooran 89] Pooran F.J. *Dynamics and control of robot manipulators with closed-kinematic chain mechanism.* Ph.D. Thesis, The Catholic University of America, Washington D.C., 1989.

[Stoughton 1993] Stoughton R. and Arai T. A modified Stewart platform manipulator with improved dexterity. *IEEE Trans. on Robotics and Automation*, 9(2):166–173, April 1993.