Determination of the presence of singularities in a workspace volume of a parallel manipulator

J-P. Merlet INRIA Sophia-Antipolis
BP 93
06902 Sophia-Antipolis Cedex, France

Abstract: Determining if singularities exist in the workspace of a parallel manipulator is an important step in the design of a parallel robot. Singularities are obtained when the determinant of the inverse jacobian matrix of the robot is equal to 0. Unfortunately this determinant is complex and its value vary according to the unit chosen for defining the geometry of the robot. Therefore a pure numerical approach is difficult to use. We present here an algorithm enabling to determine the presence of singularities in any type of workspace volume (in which the orientation of the robot is constant), either defined by a geometrical object in the Euclidean space or by a volume in the 6-dimensional articular space. This algorithm is fast and numerically robust. The main component of this algorithm is a program which compute the minimum and maximum values of the determinant for any location of the center of the end-effector in a box: if these values have opposite signs, then at least one singularity exist in the box. More complex workspaces are analyzed using a box decomposition.

Furthermore this algorithm enables to determine the location of the singularities with a guaranteed accuracy by determining the location of a box in which a singularity occurs, the distance between the center of the box and any point in it being lower than the desired accuracy.

1 Introduction

Parallel robot have been extensively studied this recent years and are now starting to appear as commercial product. In this paper we consider a 6 d.o.f. parallel manipulator constituted of a fixed base plate and a mobile plate connected by 6 extensible links. We assume that both the base and mobile plate are planar (figure 1). Designing a parallel robot is a difficult task: the designer has to take into account not only the user's requirements but also hidden features which play an important role in the behavior of the robot. An example of these features is the singularity problem. In some posture of the robot, called *singular configurations*, the robot will gain some degrees of freedom, becoming uncontrollable. Furthermore in these configurations the articular forces may go to infinity, this leading to a breakdown of the robot. Hence, although most of the users will not be aware of this problem, it is essential that the designer verify the presence of singularity in the workspace of the robot.

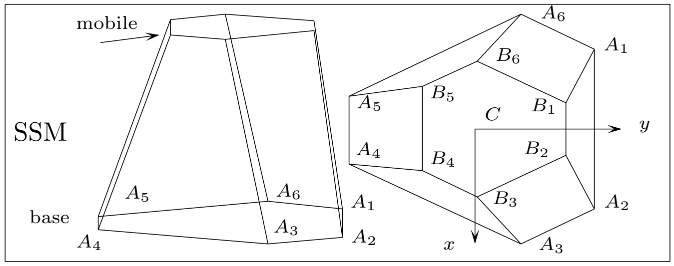

Figure 1: Gough platform with planar base and platform

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. Let A_i, B_i be the attachment points of the legs on the base and mobile platform, ρ_i be the leg lengths. If J^{-1} is the inverse jacobian matrix of the robot the articular velocities vector $\dot{\rho}$ is related to the cartesian/angular velocities vector $\dot{\mathbf{X}}$ of the mobile platform by:

$$\dot{\rho} = J^{-1}\mathbf{X} \tag{1}$$

The inverse jacobian matrix may be written as:

$$J^{-1} = ((\frac{\mathbf{A_iB_i}}{\rho_i} \quad , \quad \mathbf{CB_i} \times \frac{\mathbf{A_iB_i}}{\rho_i})) \tag{2}$$

which is posture dependent. The singular configurations $\mathbf{X_s}$ are defined by:

$$|J^{-1}(\mathbf{X_s})| = 0$$

Finding the set of singular configurations of a given robot is a difficult task: indeed although the inverse jacobian matrix is known the expansion of its determinant leads to an huge expression [3] which is difficult to use. Another approach is based on Grassmann line geometry and has been successful to determine a set of conditions on the posture parameters defining the singularities [4]. Another area of work is the determination of the geometry such that the robot remains always in a singular configuration [5].

But from the design view point we are more interested in the following problem: is there a singularity(s) in a given workspace of a given robot ? To the best of our knowledge this problem has not yet been addressed in the literature. Related works deal with a global conditionning index (like the condition number of the inverse jacobian matrix estimated over the whole workspace [2]). But this index is usually estimated by a discrete method over the whole workspace henceforth may fail to locate singularities.

Before proceeding to the explanation of our algorithm let us make a remark about the inverse jacobian matrix. Let M be the matrix defined by

$$M = ((\mathbf{A_iB_i} \quad , \qquad \mathbf{CB_i} \times \mathbf{A_iB_i})) \tag{3}$$

We have:

$$|J^{-1}| = \frac{|M|}{\prod_{i=1}^{i=6} \rho_i}$$

and consequently the singular configuration are also defined by $|M| = 0$. M will be denoted the *semi-inverse jacobian matrix* of the robot.

2 The algorithm

The purpose of our algorithm is to determine the presence of singularities in a given workspace of the robot in which *the orientation is kept constant*. The basic idea is quite simple: assume that we are able to compute the minimal and maximal value M_m, M_M of $|M(\mathbf{X})|$ for any location of $\mathbf{X}$ in the workspace. This determinant being a continuous function of the coordinates of C singularity(s) will occur in the workspace if and only if the the product $M-m\ M_M$ is equal or lower to 0. We have thus transformed our initial problem into an optimization problem.

This optimization problem will be solved in different steps according to the workspace described by C. First we will solve it when C moves along a segment, then in an horizontal or vertical rectangle and then in a box. In a latter section we will address the optimization problem for more complex workspaces.

2.1 Segment workspace

Let assume that C is moving along a segment defined by its extreme points M_1, M_2. Any position of C along this segment may be defined by:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1M_2} \tag{4}$$

where λ is a scalar in the range [0,1]. Therefore $|M|$ is an algebraic function $P(\lambda)$ in the parameter λ and it is possible to show that $|M|$ is a third order polynomial in this variable. Consequently the extremum of $|M|$ will be obtained by deriving this polynomial and finding the roots of the resulting second order polynomial $P'(\lambda)$. Let $Max(x_0, ..x_n)$ denote the greater element of the set $x_0, ..x_n$ and $Min(x_0, ..., x_n)$ be the lower element of the set. The values of $M-m, M_M$ will be obtained in the following manner:

- if $P'(\lambda)$ has no root in the range [0,1] then $M_m = Min(P(0), P(1))$, $M_M = Max(P(0), P(1))$
- if $P'(\lambda)$ has some root λ_1, λ_2 (2 at most) in the range [0,1], then $M_m = Min(P(0), P(1), P(\lambda_1), P(\lambda_2))$, $M_M = Max(P(0), P(1), P(\lambda_1), P(\lambda_2))$

2.2 Rectangle workspaces

Let x_c, y_c, z_c be the coordinates of C and assume that C is moving inside an horizontal rectangle defined by:

$$x_1 \leq x_c \leq x_2 \qquad y_1 \leq y_c \leq y_2$$

Consequently we introduce two new variables α, β such that the coordinates of C are:

$$x = x_1 + (1 + \sin\alpha)(x_2 - x_1)/2 \qquad y = y_1 + (1 + \sin\beta)(y_2 - y_1)/2$$

$|M|$ could then be computed as function of α, β. The derivatives of $|M|$ with respect to these variables lead to two constraint equations. These equations show that the extremum will be reached either on the border of the rectangle (in which case the result of the previous section can be used) or inside the rectangle. In the latter case the two constraint equations can be combined into a one variable polynomial of order 9 in the unknown $\tan(\beta/2)$. Solving numerically this polynomial leads to the determination of all the pairs (α, β) which may be lead to an extremum of $|M|$. Therefore by computing the extremum on the four edges of the rectangle together with the possible extremum for the interior of the rectangle we will obtain the extremum of $|M|$ for the the whole rectangle.

For vertical rectangles we use the same procedure simply by substituting y_c by z_c. The resolution is then identical.

2.3 Box workspace

Now assume that C is moving in a box defined by:

$$x_1 \leq x_c \leq x_2 \qquad y_1 \leq y_c \leq y_2 \quad z_1 \leq z_c \leq z_2$$

We define three new variables α, β, μ such that:

$$\begin{aligned} x &= x_1 + \frac{(1 + \sin\alpha)(x_2 - x_1)}{2} \\ y &= y_1 + \frac{(1 + \sin\beta)(y_2 - y_1)}{2} \\ z &= z_1 + \frac{(1 + \sin\mu)(z_2 - z_1)}{2} \end{aligned}$$

The determinant of M can be computed with respect to these three variables and its derivatives lead to three constraint equations. These equations show that the extremum will be obtained either for the faces of the box (in which case the result of the previous section can be used) or for a point in the interior of the box. In that latter case the three constraints equations can be combined into a one variable polynomial of order 6 in the unknown $\sin(\mu/2)$. If we solve numerically this polynomial we may determine the values of α, β, μ leading to an extremum of $|M|$ for a point inside the box. Then by taking the extremum for the faces of the box we are able to compute the extremum of $|M|$ for the whole box.

2.4 General workspaces

We will assume here that the workspace is defined by a set of horizontal polygonal cross-sections,the workspace between two sections being the polyhedra obtained by linking the corresponding vertices of the two polygons (figure 2 present an example of such workspace). To determine if there is a singularity within

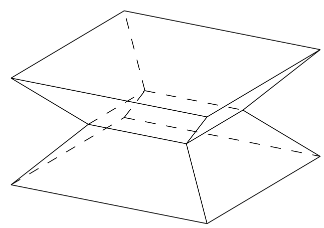

Figure 2: Example of general workspace

this workspace we will split the workspace into as many boxes as necessary and use the determination of the extremum of $|M|$ for each box.

We maintain a list of boxes $B_0, ..$ which is initialized with the bounding box of the full workspace (without lack of generality we will assume here that we have only two cross-sections). At the k step we look at the top box B_k in the list:

1. if B_k is fully outside the workspace we delete it, push the next box of the list on the top and start again
2. if B_k is fully inside the workspace we compute the extremum of $|M|$ for this box and determine if a singularity occur within this box. If yes we have determined that a singularity occur within the workspace and the algorithm stop. If not we push the next box of the list on the top and start again.
3. if B_k is partially inside the workspace we determine if a singularity occur within the box. If there is no singularity within the box we push the next box of the list on the top and start again. If there is a singularity we look at the positions of C for which the extremum of $|M|$ are obtained:

(a) if the positions where the minimum and maximum of $|M|$ are obtained lie within the workspace there is singularity within the workspace and the algorithm stops.

(b) otherwise we split the box B_k into 8 smaller boxes by dividing all its dimension by 2. B_k is deleted from the list , the 8 new boxes are put at the end of the list and we start again

The algorithm will stop either when a singularity is detected or when the list is empty (in which no singularity exists in the workspace).

The computation time is clearly heavily dependent upon the number of cross-sections but a mean computation time for three cross-sections is about 10 s on a SUN SS5 workstation.

Note that this algorithm may be extended to deal with a workspace defined in term of articular coordinates. Indeed being given extremum for the articular coordinates we may deduce a rough box which will include all the the locations of C whose corresponding leg lengths lie within the articular extremum: for example we may choose a box with center at the origin O and whose dimensions will be given by $Min(A_{i_{xyz}} + \rho_i^{max} + ||\mathbf{CB_i}||)$. This box will be the initial box of the list of the algorithm. The only change with the previous algorithm will consist in the method used to determine if a box is inside, outside or partially inside the box defined in term of the articular coordinates. But it is easy to show that for a cartesian box we are able to compute the extremum of the leg lengths for all the points located in the box. Therefore for each new box appearing during the algorithm we will determine the extremum of the leg lengths in order to determine the position of the articular box corresponding to the cartesian box with respect to the articular workspace.

3 Remark on the numerical robustness

First of all let us notice that the matrix M is not invariant with respect to the choice of dimension unit: for a given row the three elements have no unit while the three last one have a length dimension. Consequently we may change at will the value of the determinant just by changing the unit length. A bad choice of unit may result in trouble for the numerical procedure which compute the value of the determinant of M.

Another possible trouble may appear if one of the extremum is close to zero. In that case numerical errors may lead to a change of sign in the value of the determinant, which in turn will lead to an error in the answer of the algorithm (note that the value of M_m, M_M are of no importance, only their signs are essential). An example of such case can be obtained if we consider a vertical box over the base, the mobile platform being parallel to the base. In that case we have:

$$|M| = k\ z_c^3$$

The extremum of $|M|$ will therefore be obtained for the lowest and highest value of z_c and we may choose at will the lowest z_c to get a minimum of $|M|$ as close as 0 as we want.

Finding exactly the sign of determinant is an old problem and this problem has been addressed in many papers. In order to deal with this numerical problem whenever it may occur we have decided to use an exact method as soon as the determinant obtained from the numerical procedure was close to 0. We use the Clarckson method as implemented by H. Bronnïmann. This method enables to compute exactly the sign of a determinant as soon as the elements of the matrix are integers not greater than 2^{53} [1]. As the matrix M is in fact an array of floats we use the largest element s of this matrix as a normalizing factor, each element p being then converted in integer value by dividing the element by s, multiplying the result by 2^{53} and taking the integer p_i closest to the result. Then the sign of the determinant of this new matrix is computed using Clarckson method.

An error may still occur as the new matrix is only an approximation of the matrix M. A possible improvement will be to consider all the integer matrices whose elements are constructed from the elements of M either by taking as p_i the integer part t of $p\ 2^{53}/s$ or $t+1$. All the matrices build from all the possible combinations of p_i will then have their sign of determinant computed and if all the signs are identical we will be completely confident in the result sign.

4 Conclusion

We have presented in this paper a method which enable to determine the presence of singularity in given translation workspace of the robot (either specified in cartesian or articular form). The computation time of this algorithm is quite low. Still we have to deal with the orientation of the mobile platform. This may be done either by using a discretisation in the orientation workspace but we are investigating more efficient approach.

References

[1] Clarkson K. L. Safe and effective determinant evaluation. In *Proc. 33rd Annu. IEEE Sympos. Found. Comput. Sci.*, pages 387–395, 1992.

[2] Gosselin C. and Angeles J. The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator. *J. of Mechanisms, Transmissions and Automation in Design*, 111(2):202–207, 1989.

[3] Mayer St-Onge B. and Gosselin C. Singularity analysis and representation of spatial six-dof parallel manipulators. In J. Lenarčič V. Parenti-Castelli, editor, *Recent Advances in Robot Kinematics*, pages 389–398. Kluwer, 1996.

[4] Merlet J-P. Singular configurations of parallel manipulators and Grassmann geometry. *Int. J. of Robotics Research*, 8(5):45–56, October 1989.

[5] Zsombor-Murray P., Husty M., and Hartmann D. Singular Stewart-Gough platforms with spherocylindrical and spheroconical hip joint trajectories. In *9th World Congress on the Theory of Machines and Mechanisms*, pages 1886–1890, Milan, August 30- September 2, 1995.