Trajectory verification of parallel manipulators in the workspace

Jean-Pierre MERLET
INRIA
Centre de Sophia-Antipolis
BP 93 06902 Sophia-Antipolis, France
E-mail: merlet@cygnusx1.inria.fr

Abstract

We present a fast algorithm for solving the problem of determining if the straight line between two different postures of a parallel manipulator lie fully inside its workspace. This algorithm is based on the analysis of the algebraic inequalities describing the constraints on the workspace (link lengths range, mechanical limits on the joints, interference between the links) and enables to compute which part of the trajectory is outside the workspace. This method is exact if the orientation of the end-effector is kept constant along the trajectory and approximate in the opposite case.

1 Introduction

Let us consider a 6 d.o.f. parallel manipulator as represented on figure 1. It is constituted of a fixed

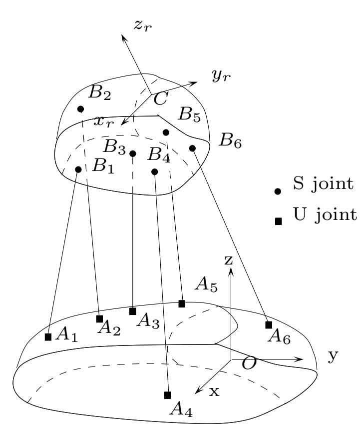

Figure 1: A 6 d.o.f. parallel manipulator.

base plate and a mobile plate connected by 6 variable-length links. One of the extremities of each link is articulated with the base plate through an universal joint and the other extremity is articulated with the mobile plate through a ball-and-socket joint. By changing the 6 link lengths we are able to control the position and orientation of the mobile plate.

Three types of constraint play a role for determining the workspace of parallel manipulators:

- limited range for the link lengths: the linear actuators controlling the links lengths have a limited range. The minimum length of link i will be denoted ρ^i_{min} and the maximum length ρ^i_{max}
- mechanical limits on the passive joints (universal joints and ball and socket joints).
- links interference

The problem of determining the workspace of a parallel manipulator has been addressed by many authors most of them using a discretization method in the parameter's space, some of them assuming that the orientation of the end-effector is kept constant [1], [2],[5], [6] or assuming that a point is fixed for computing the orientation workspace [10],[11],[12],[13].

A completely different approach has been proposed by Gosselin [3] which uses a purely geometrical method for determining the workspace border due to the limited range of the links lengths. This approach has been then first extended to take into account all the constraints limiting the workspace [7] and for computing the orientation workspace when a point of the end effector is fixed [8].

To the best of our knowledge nobody has addressed the problem of verifying a trajectory with respect to the workspace i.e. being given two points in the parameter's space (i.e. two postures for the end effector) is the straight line joining these two points fully inside the workspace of the robot. Clearly this problem is very important for the motion planning of a parallel manipulator.

Let us introduce now some notation. We define two frames, one fixed $(O, \mathbf{x}, \mathbf{y}, \mathbf{z})$ (reference frame) and the other one attached to the end effector $(C, \mathbf{x_r}, \mathbf{y_r}, \mathbf{z_r})$

(relative frame). C will be used to define the position of the end-effector in the reference frame. The following symbols and variables will be used in this paper:

- A_i: center of the passive joint of link i attached to the base of the robot.
- B_i: center of the passive joint of link i attached to the end effector.
- ψ, θ, ϕ: angles defining the orientation of the end effector

2 Trajectory with a fixed orientation

In this section we will assume that the orientation of the end effector is kept constant all along the trajectory. Let us define the start and goal points of the trajectory as M_1, M_2. Consequently any position of the end-effector on the trajectory may be defined as:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda\mathbf{M_1M_2} \quad \text{with} \quad \lambda \in [0,1] \qquad (1)$$

2.1 Limitation on the links lengths

Let us calculate the length ρ of a link for any point on the trajectory between M_1 and M_2. We have:

$$\mathbf{AB} = \mathbf{AM_1} + \mathbf{CB} + \lambda\mathbf{M_1M_2} \qquad (2)$$

which yield to:

$$\rho^2 = \mathbf{AB}.\mathbf{AB}^T = a\lambda^2 + b\lambda + c \qquad (3)$$

Now let us consider the following equation:

$$a\lambda^2 + b\lambda + c - \rho^2_{max} \qquad (4)$$

If this equation has no root and as $a > 0$ then for all λ the equation is positive and the link length is greater than its maximum value on all the trajectory.

Assume now that the equation has two roots x_1, x_2 sorted by higher value. As $a > 0$ the equation will be positive for λ in the interval $]-\infty, x_1[,]x_2, +\infty[$. The intersection of these intervals with the interval $[0,1]$ will define the intervals on λ (i.e. the portion of the trajectory) where the link length will be greater than the maximum link length value.

By using this algorithm for the 6 links and calculating the union I_{max} of all the obtained intervals we will get the portions of the trajectory where at least one link length will be greater than its allowed maximum value.

By changing ρ_{max} by ρ_{min} in equation (4) we then get the union I_{min} of the intervals which define the portions of the trajectory where at least one link length will be lower than its allowed minimum value.

The union of I_{max}, I_{min} will give the portions of the trajectory where at least one link length will be outside its allowed range. If the union is empty the trajectory is fully inside the workspace of the manipulator.

The study of the number of real roots of the previous equations according to their coefficients yield to 6 simplification rules [9]. Due to the lack of space they are not given here.

2.2 Mechanical limits on the passive joints

The mechanical limits on joints like universal joints or ball-and-socket joints can be modelized by a surface which is the border of the allowable zone for the link connected to the joint. Using a similar method as in [7] we assume that this surface can be approximated by a pyramid with planar faces. For the joints attached to the base the center of this pyramid is located at point A (figure 2).

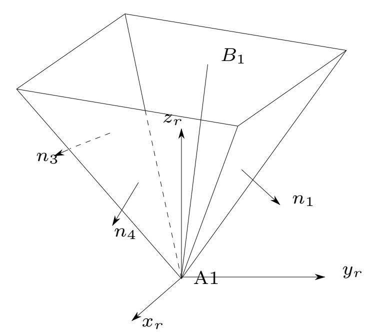

Figure 2: An example of modelization of a constraint on a passive joint located at A_1. If the mechanical limits of the joints are satisfied then the link A_1B_1 is inside the volume delimited by the pyramid (here a pyramid with 4 faces).

As for the constraint on the passive joints attached to the end-effector we may use the same model. We define a pyramid P_i with center B_i such that if the constraint on the joint at B are satisfied then point A_i will lie inside the pyramid from which we deduce another pyramid which will be called *an equivalent pyramid* $P_i^{;}$ to P_i, which center is A_i such that if A_i lie inside P_i then B_i lie inside $P_i^{;}$. Therefore for both kind of joints a similar model can be used.

Let $\mathbf{n_i}$ be the external normal of the ith face of the pyramid associated to the joint attached to the base. If point B lie inside the pyramid we have:

$$\mathbf{AB}.\mathbf{n_i}^T \leq 0 \qquad (5)$$

By using equation (1) we get a linear equation in λ. Let us consider the possible intervals where the inequality is not satisfied. The intersection of these intervals with $[0, 1]$ yield the portion of the trajectory where the constraints on the joint are not satisfied. This algorithm has to be used for all the faces of all the 12 pyramids defining the constraint on the joints. Let I_{pyr_i} be the union of all these intervals for link i.

2.3 Links interference

We define the distance between two links as the minimal distance between any pair of points on the links. It has been shown in [7] that this distance is the minimum of the following distances:
-the distance between the lines associated to the links if their common perpendicular has a point on each link
-the distance between a point B and its projection point on the other link if this point belongs to the link
-the distance between a point A and its projection point on the other link if this point belongs to the link
-the distance between the points of one of the two pairs of points (A_i, B_j)

We assume that link i can be approximated by a cylinder with radius r_i and will say that links i, j interfere if their distance if lower than $d = r_i + r_j$. We will consider now the above cases for links 1 and 2.

2.3.1 Distance between the lines

The distance d_{12} between the lines associated to links 1 and 2 can be written as:

$$d_{12} = \frac{\mathbf{A_1A_2}.(\mathbf{A_1B_1} \times \mathbf{A_2B_2})^T}{||\mathbf{A_1B_1} \times \mathbf{A_2B_2}||} \tag{6}$$

Using equation (1) and writing that the distance between the lines is lower or equal to d yield to a second order inequality:

$$P_1(\lambda) = u_2\lambda^2 + u_1\lambda + u_0 \geq 0 \tag{7}$$

The intervals on λ included in [0,1] such that $P_1(\lambda)$ is positive define the parts of the trajectory where the distance between the lines is lower or equal to d.

Let I_d be the set of all these intervals. Let Q_1, Q_2 be the points on line 1,2 belonging to their common perpendicular. If these points belong to the links for some values of λ in I_d then there is links interference. We define α_1, α_2 such that:

$$\mathbf{A_1Q_1} = \alpha_1\mathbf{A_1B_1} \quad \mathbf{A_2Q_2} = \alpha_2\mathbf{A_2B_2} \tag{8}$$

Point Q_i belongs to link i if α_i is in [0,1]. α_1, α_2 can easily be obtained as:

$$\alpha_1 = \frac{P_{\alpha_1}(\lambda)}{P_{det}} = \frac{s_2\lambda^2 + s_1\lambda + s_0}{t_2\lambda^2 + t_1\lambda + t_0} \tag{9}$$

$$\alpha_2 = \frac{P_{\alpha_2}}{P_{det}} = \frac{r_2\lambda^2 + r_1\lambda + r_0}{t_2\lambda^2 + t_1\lambda + t_0} \tag{10}$$

Let $I_+^{P_i}$ be the intervals included in [0,1] such that P_{α_i} is positive or equal to zero (i.e. $\alpha_i \geq 0$), $I_1^{P_i}$ the intervals in [0,1] where $P_{\alpha_i} - P_{det}(\lambda)$ is negative or equal to zero (i.e. $\alpha_i \leq 1$). The set I_D of intervals of λ in [0,1] where the distance between the links is the distance between the lines and is lower than d is therefore:

$$I_D = I_d \cap (I_+^{P_1} \cap I_1^{P_1}) \cap (I_+^{P_2} \cap I_1^{P_2}) \tag{11}$$

If $I_d = \emptyset$ the distance between the lines (which is a lower bound of the distance between the links) is always greater than d and therefore links interference cannot occur. If $I_d \neq \emptyset$ and $I_D = \emptyset$ the distance between the links will always be different and greater than the distance between the lines.

2.3.2 Distance between the points B_i and their projections

The distance l from point B_1 to line 2 can be written as:

$$l_{B_1^2} = \frac{||\mathbf{B_1B_2} \times \mathbf{A_2B_2}||}{||\mathbf{A_2B_2}||} \tag{12}$$

Using equation (1) and writing that $l_{B_1^2}$ is lower than d yield to:

$$P_1^{B_1^2}(\lambda) = a_2^1\lambda^2 + a_1^1\lambda + a_0^1 \geq 0 \tag{13}$$

and collision will occur if the projection Q_1 of B_1 on line 2 belongs to link 2. Let:

$$\mathbf{A_2Q_1} = \beta_1\mathbf{A_2B_2} \tag{14}$$

the above condition will be fulfilled if β_1 belongs to [0,1]. We have:

$$\beta_1 = \frac{P_2^{B_1^2}(\lambda)}{Q(\lambda)} = \frac{\beta_2^1\lambda^2 + \beta_1^1\lambda + \beta_0^1}{\beta_2^2\lambda^2 + \beta_1^2\lambda + \beta_0^2} \tag{15}$$

Let $I_{B_i^j}$ the intervals included in [0,1] such that $P_1^{B_i^j} \geq 0$ (i.e. $l \leq d$), $P_2^{B_i^j} \geq 0$ (i.e. $\beta_1 \geq 0$), $P_2^{B_i^j} - Q(\lambda) \leq 0$ (i.e. $\beta_1 \leq 1$). The set of intervals $I_{B_i^j},\ i, j \in\ [1, 6], i \neq j$ defines the componants of the trajectory for which interference occurs between links i and j.

2.3.3 Distance between the points A_i and their projections

The distance $l_{A_1^2}$ from point A_1 to line 2 is:

$$l_{A_1^2} = \frac{||\mathbf{A_1B_2} \times \mathbf{A_2B_2}||}{||\mathbf{A_2B_2}||} \tag{16}$$

Using equation (1) and writing that $l_{A_1^2}$ is lower than d yield to:

$$P_1^{A_1^2}(\lambda) = w_2^1\lambda^2 + w_1^1\lambda + w_0^1 \geq 0 \quad (17)$$

under the condition that the projection Q_1 of A_1 on line 2 belongs to link 2. Let:

$$\mathbf{A_2Q_1} = \mu_1\mathbf{A_2B_2} \quad (18)$$

Q_1 belongs to link 2 if μ_1 is in [0,1]. We have:

$$\mu_1 = \frac{P_2^{A_1^2}(\lambda)}{Q(\lambda)} = \frac{\mu_1^1\lambda + \mu_0^1}{\mu_2^2\lambda^2 + \mu_1^2\lambda + \mu_0^2} \quad (19)$$

Let $I_{A_i^j}$ the intervals included in [0,1] such that $P_1^{A_i^j} > 0$ ($l_{A_1^2} \leq d$), $P_2^{A_i^j} \geq 0$ ($\mu_1 \geq 0$), $P_2^{A_i^j} - Q(\lambda) \leq 0$ ($\mu_1 \leq 1$).

The set of intervals $I_{A_i^j}$, $i,j \in [1,6]$, $i \neq j$ defines the componants of the trajectory on which interference between link i and j occurs.

2.3.4 Distance between points A_i and B_j

Using equation (1) the distance between points A_2 and B_1. $||\mathbf{A_2B_1}||^2$ is a second order polynomial in λ $P_{A_2B_1}(\lambda)$.

We denote by $I_{A_iB_j}$ the intervals of λ included in [0,1] such that $P_{A_iB_j}(\lambda) - d^2 \leq 0$. These intervals define the parts of the trajectory the distance from B_j to A_i is lower than d.

The union of the intervals defining forbidden value for λ for each constraint defines the set I_{bad} of intervals forbidden for λ from which we deduce the forbidden parts of the trajectory. We get:

$$I_{bad} = I_{max} \cup I_{min} \cup I_{pyr_i} \cup I_{D_{ij}} \cup I_{B_i^j} \cup I_{A_i^j} \cup I_{A_iB_j} \quad (20)$$

2.4 Computation time

The above algorithms have been implemented in a workspace computation program written in C on a Sun Sparc2 workstation.

The computation time for the verification of a trajectory is approximatively 2ms for the links lengths constraints, 25 ms for checking the interference between each pair of links and 0.3ms for checking a face of a pyramid for the mechanical limits on the joints. The verification of all the constraints the computation time for a trajectory is approximatively 29 ms.

2.5 Examples

We have performed trajectory verification for a prototype of parallel manipulator developed by Arai [1] at the MEL in Tsukuba. Figures 3,4 show trajectories on which forbidden parts are singled out.

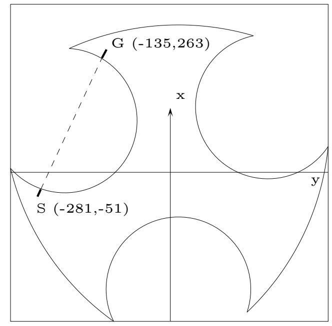

Figure 3: Example of trajectory verification: the forbidden parts of the trajectory are drawn in dashed lines. The constraints are only the links lengths.

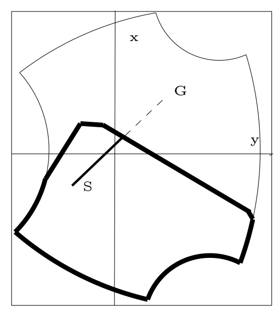

Figure 4: Examples of trajectory verification: the frontier of the workspace where there is no links interference is drawn in thick lines. The forbidden parts of the trajectory are drawn in dashed lines.

3 General Trajectory

In the case of a constant orientation we have seen that the constraints can be expressed under the form of algebraic equations in the variable λ which describe the trajectory. If we introduce now a varying orientation we have now more algebraic constraints as the sines and cosines of the rotation matrix will appear.

In order to get again algebraic constraints equations we will split the trajectory in elementary parts such that the change in the orientation between the extremal points of one elementary part will be small. For each elementary part a linear interpolation for the angles will be used to determine the orientation of the end-effector. As the orientation will affect only the value of the vector **CB** we will use a first or second order approximation for this vector. Let denote M_1, M_2 the extremal points of one elementary part of the trajectory, ψ_1, θ_1, ϕ_1 the Euler's angles describing the orientation of the end-effector at point M_1 and ψ_2, θ_2, ϕ_2 the Euler's angles of the end-effector at point M_2. Between points M_1 and M_2 the position of point C is defined by equation (1) and the rotation angles can be written as:

$$\begin{aligned}\psi &= \psi_1 + \lambda(\psi_2 - \psi_1) \qquad \theta = \theta_1 + \lambda(\theta_2 - \theta_1)\\ \phi &= \phi_1 + \lambda(\phi_2 - \phi_1)\end{aligned}$$

If we use a first order approximation we get:

$$\mathbf{CB}(\psi, \theta, \phi) = \mathbf{CB}(\psi_1, \theta_1, \phi_1) + \lambda \mathbf{U_1} \tag{21}$$

and a second order approximation yield to:

$$\mathbf{CB}(\psi, \theta, \phi) = \mathbf{CB}(\psi_1, \theta_1, \phi_1) + \lambda \mathbf{U_1} + \lambda^2 \mathbf{U_2} \tag{22}$$

where the vectors $\mathbf{U_1}, \mathbf{U_2}$ are only dependent upon the relative position of B and the angles ψ_1,θ_1,ϕ_1, ψ_2,θ_2,ϕ_2.

Under these assumptions we may now analyze the various constraints on an elementary part $\mathcal{T}$ of the trajectory.

3.1 Links lengths constraints

By using equation (1) and a second order approximation (22) we obtain the link length as:

$$P_\rho(\lambda) = a_3\lambda^3 + a_2\lambda^2 + a_1\lambda + a_0 \tag{23}$$

where the a_i are coefficients which are only dependent upon the trajectory and the design of the robot. As in the constant orientation case the analysis of the polynomial $P_\rho(\lambda) - \rho^2_{max}, P_\rho(\lambda) - \rho^2_{min}$ enables to compute the intervals of λ in [0,1] such that the link length is greater than its maximum value or lower than its minimal value.

3.2 Constraints on the passive joints

A second order approximation of **CB** (22) is used together with equation (1) to express the constraint equation (5) which yield to a second order inequality. The analysis of this inequality enables to determine the intervals on λ such that some point of the link lie outside the pyramid. By considering all the set of faces of every pyramid we get which parts of the trajectory does not satisfy the joints constraints. A similar analysis can be done for the passive joints of the mobile plate.

3.3 Links interference

3.3.1 Distance between the lines

By using equation (1) and a first order approximation (21) writing that the distance d_{12} between the lines is lower than d yield to a fourth order inequality in λ, $P(\lambda) \geq 0$.

In order to find at which points of $\mathcal{T}$ the distance between the lines is lower or equal to d we must find the intervals I_d on λ, included in [0,1], where $P(\lambda) \geq 0$.

If the common perpendicular points Q_1, Q_2 of line 1, 2 belong to the links we get a links interference. Let:

$$\mathbf{A_1Q_1} = \alpha_1 \mathbf{A_1B_1} \qquad \mathbf{A_2Q_2} = \alpha_2 \mathbf{A_2B_2} \tag{24}$$

We get:

$$\alpha_1 = \frac{P_{\alpha_1}(\lambda)}{det} = \frac{s_3\lambda^3 + s_2\lambda^2 + s_1\lambda + s_0}{t_4\lambda^4 + t_3\lambda^3 + t_2\lambda^2 + t_1\lambda + t_0} \tag{25}$$

$$\alpha_2 = \frac{P_{\alpha_2}(\lambda)}{det} = \frac{r_3\lambda^3 + r_2\lambda^2 + r_1\lambda + r_0}{t_4\lambda^4 + t_3\lambda^3 + t_2\lambda^2 + t_1\lambda + t_0} \tag{26}$$

Now a procedure similar to the one described in section 2.3.1 can be used to compute the componants of $\mathcal{T}$ where links interference will occur.

3.3.2 Distance between the points B_i and their projections

The distance $l_{B_1^2}$ from point B_1 to line 2 will be lower than d if:

$$P_1^{B_1^2}(\lambda) = g_4^1\lambda^4 + g_3^1\lambda^3 + g_2^1\lambda^2 + g_1^1\lambda + g_0^1 \geq 0 \tag{27}$$

Links interference will occur if the above equation is satisfied and if the projected point Q_1 of B_1 on line 2 belongs to link 2. Let:

$$\mathbf{A_2Q_1} = \beta_1 \mathbf{A_2B_2} = \frac{\beta_2^3\lambda^2 + \beta_1^3\lambda + \beta_0^3}{\beta_2^4\lambda^2 + \beta_1^4\lambda + \beta_0^4} \mathbf{A_2B_2} \tag{28}$$

Q_1 will belong to link 2 if β_1 is in [0,1]. Now a procedure similar to the one described in section 2.3.2 can be used to compute the componants of $\mathcal{T}$ where links interference will occur.

3.3.3 Distance between the points A_i and their projections

Using a first order approximation (21) the distance $l_{A_1^2}$ from point A_1 to line 2 will be lower than d if:

$$P_1^{A_1^2}(\lambda) = h_2^1\lambda^2 + h_1^1\lambda + h_0^1 \geq 0 \quad (29)$$

Collision between links 1 and 2 will occur if the projection point Q_1 of A_1 on line 2 belongs to link 2. Let:

$$\mathbf{A_2Q_1} = \mu_1\mathbf{A_2B_2} = \frac{\mu_1^3\lambda + \mu_0^3}{\mu_2^4\lambda^2 + \mu_1^4\lambda + \mu_0^4}\mathbf{A_2B_2} \quad (30)$$

The above condition will be fulfilled if μ_1 is in [0,1]. Now a procedure similar to the one described in section 2.3.3 can be used to compute the componants of $\mathcal{T}$ where links interference will occur.

3.3.4 Distance between the points A_i and B_j

Using a first order approximation (21) $||\mathbf{A_2B_1}||^2 \leq d$ yield to a second order inequality in λ. This situation may occur only if a condition is satisfied (see [9]).

3.4 Computation time

The computation time for the verification of a trajectory is dependent upon its number of elementary parts. This number is obtained by considering the orientation angle with the greatest variation and by dividing this variation by a constant angle (5° in our implementation).

The mean computation time for the verification of one elementary part is approximatively 16 ms for the links lengths constraints, 430ms for checking the interference between each pair of links and 1ms for checking a face of a pyramid for the mechanical limits on the joint. If we take into account all the constraints for a robot with four-faced pyramids on the base joints we get a total computation time of 450 ms.

3.5 Examples of trajectory verification

We show in figure 5 an example of trajectory verification.

4 Application: motion planning

Let us assume that a given trajectory is outside the workspace of a given robot. We will assume here that the orientation is kept constant all along the trajectory and that the trajectory is in a given horizontal plane. As we know the workspace border we tile it with small square cells (figure 6).

In this tessellation we find the squares which contain the start and goal points. We build then a valued graph whose nodes are the centers of the cells, which are connected by arcs to their neighbour cells. The

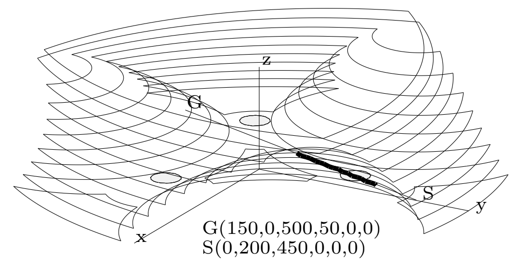

Figure 5: Example of trajectory verification: the forbidden parts of the trajectory are drawn in thick lines. Each of the cross-section of the workspace has been computed for an orientation obtained by a linear interpolation between the orientations at the start and goal positions. The constraints are the links lengths and links interference. The start and goal points are in the workspace but a part of the trajectory is outside the workspace.

value of the arcs is the distance between the nodes if the line joining the nodes lie inside the workspace or an arbitrary large value if the line is outside the workspace (this can be determined by using our verification algorithm). A path between the start and goal points can be found by using a shortest path algorithm in the graph (for example an A^* algorithm [4]) and this path may then be smoothed (figure 6).

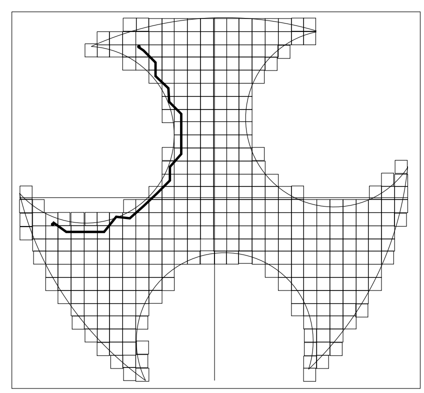

Figure 6: Motion planning: the straight line between the start and goal points is not a valid trajectory. The workspace border is calculated and it is tiled with small square cells. An A^* algorithm enables to find a path between the start and goal position in the workspace. A smoothing algorithm can then be used.

5 Conclusion

We have presented an algorithm enabling to verify if a given trajectory is fully inside the workspace of a parallel manipulator. This workspace is calculated by considering every constraints which can limit the reach of the robot: links lengths range, mechanical limits on the passive joints, links interference. In this algorithm these constraints are expressed as algebraic inequalities which are easily solved. These algebraic inequalities describe exactly the constraints if the orientation of the end-effector is kept constant all along the trajectory and are approximatively exact in the opposite case. By solving these inequalities we can determine if the trajectory is fully inside the workspace or find which part of the trajectory is outside the workspace.

References

[1] Cleary K. and Arai T. A prototype parallel manipulator: kinematics construction, software, workspace results and singularity analysis. In *IEEE Int. Conf. on Robotics and Automation*, pages 566–571, Sacramento, April, 11-14, 1991.

[2] Fichter E.F. A Stewart platform based manipulator: general theory and practical construction. *Int. J. of Robotics Research*, 5(2):157–181, Summer 1986.

[3] Gosselin C. Determination of the workspace of 6-dof parallel manipulators. In *ASME Design Automation Conf.*, pages 321–326, Montréal, September, 17-20, 1989.

[4] Latombe J.C. *Robot Motion planning.* Kluwer Academic Publishers, Boston, , 1991.

[5] Lee K-M and Shah D.K. Kinematic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE J. of Robotics and Automation*, 4(3):354–360, June 1988.

[6] Merlet J-P. Parallel manipulators, Part 1, theory. Research Report 646, INRIA, March 1987.

[7] Merlet J-P. Manipulateurs parallèles, 5eme partie : Détermination de l'espace de travail à orientation constante. Research Report 1645, INRIA, March 1992.

[8] Merlet J-P. Manipulateurs parallèles, 6eme partie : Détermination des espaces de travail en orientation. Research Report 1921, INRIA, May 1993.

[9] Merlet J-P. Manipulateurs parallèles, 7eme partie : Vérification et planification de trajectoire dans l'espace de travail. Research Report 1940, INRIA, June 1993.

[10] Pennock G.R. and Kassner D.J. The workspace of a general geometry planar three degree of freedom platform manipulator. In *ASME Design Automation Conf.*, pages 537–544, Miami, September, 22-25, 1991.

[11] 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.

[12] Weng T-C., Sandor G.N., and Xu Y. On the workspace of closed-loop manipulators with ground mounted rotary-linear actuators and finite size platform. In *ASME Design and Automation Conf.*, pages 55–61, Boston, September, 27-30, 1987.

[13] Williams II R.L. and Reinholtz C.F. Closed-form workspace determination and optimization for parallel robot mechanisms. In *ASME Proc. of the the 20th Biennial Mechanisms Conf.*, pages 341–351, Kissimmee, Orlando, September, 25-27, 1988.