Designing a parallel manipulator for a specific workspace

J-P. Merlet
INRIA Sophia-Antipolis

Abstract: *We present an algorithm to determine all the possible locations of the attachment points of a parallel manipulator which has to reach a desired workspace described by a set of segment with a constant platform orientation. This algorithm takes into account the leg length limits, the mechanical limits on the passive joints and links interference.*

1 Introduction

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. For a parallel manipulator workspace limitations are due to the bounded range of their linear actuators, mechanical limits on their passive joints and links interference. One important step of the design of a parallel manipulator is to define its geometry according to the desired workspace. Various geometrical algorithms for computing the workspace border when the platform's orientation is kept constant, either in 2D or 3D, have been described by Gosselin [3],[5] for the case where only the constraints on the leg lengths limits are considered and in [9] in the case where mechanical limits on the joints and links interference are also taken into account.

The problem which will be addressed in this paper is to find all the possible locations of the passive joints such that the robot may reach a desired workspace. *It must be noted that the proposed algorithms have been implemented in C on a workstation under the X-windows system and every drawings appearing in this paper is a result of this program. Further details can be found in [11]*

This problem has been addressed by very few authors. Claudinon [1] uses a numerical method to find the optimal values of the design parameters which optimize some kinematics and dynamic features of its robot. Stoughton [12] uses a numerical procedure for optimizing the workspace of a specific parallel manipulator whose length limits are known. Liu [6],[7] characterizes some extremal positions as a function of the geometry of the robot and of the extremal link lengths. Gosselin [4] establishes a design rule for spherical 3 d.o.f parallel manipulator which lead to a manipulator with full rotation. The same author has studied the optimization of the workspace of planar three d.o.f parallel manipulator [2].

Let A_i, B_i denote the attachment points of the link on the base and on the platform. For a set of A_i, B_i we attach a reference frame O, x, y, z to the base such that the z coordinate of A_i is equal to 0. In the same manner we attach to the platform a mobile frame C, x_r, y_r, z_r such that the z_r coordinate of B_i is equal to 0. A subscript $_r$ will denote a point or a vector whose coordinates are written in the mobile frame. Let α_i be the angle between the Ox axis and OA_i and β_i be the angle between the Cx_r axis and CB_{i_r} (figure 1).

Under these assumptions we have:

$$\mathbf{A_iO} = \left(\begin{array}{l} -R_1\cos\alpha_i \\ -R_1\sin\alpha_i \\ 0 \end{array}\right) = R_1\mathbf{u_i} \quad \mathbf{CB_{ir}} = \left(\begin{array}{l} r_1\cos\beta_i \\ r_1\sin\beta_i \\ 0 \end{array}\right) = r_1\mathbf{v_i} \tag{1}$$

where $\mathbf{u_i}, \mathbf{v_i}$ are constant unit vectors. The purpose of this paper is to determine the possible values of R_1, r_1 such that the workspace of the corresponding robot include a specific workspace under the following assumptions:

- the specific workspace is defined for a constant orientation of the platform.
- the specific workspace is defined by a set of segments.
- the minimum and maximum value of the leg lengths are known.
- the angles α_i, β_i are known.

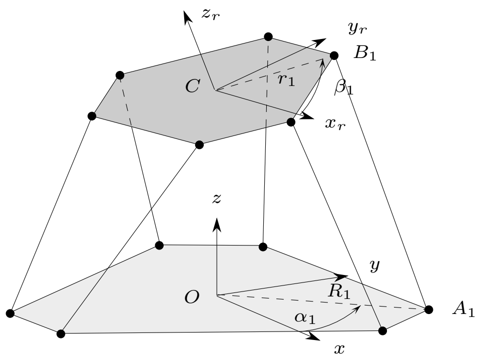

Figure 1: The design parameters for a set of point A_i, Bi are the distances R_1, r_1 between the points and the origins O, C of their frames.

The purpose of the next sections is to determine in the R_1, r_1 plane the border of the region which define the allowable values for R_1, r_1. Such regions will therefore be called the *allowable regions*. We may also remark that the validity of the result can be easily checked by using the trajectory verification algorithm described in [10] which enable to verify if a segment lie completely inside the workspace of a given robot.

2 Allowable regions for the length constraints

2.1 Allowable region for one leg

In this section we will assume that the constraints limiting the workspace are only the leg lengths. The minimum and maximum values of these lengths will be denoted $\rho_{min_i}, \rho_{max_i}$. The position of the robot is defined by the coordinates of C in the fixed frame and the rotation matrix R between the fixed and mobile frame which may be defined by the three Euler's angles ψ, θ, ϕ. A trajectory is defined by two points $M_1(x_1, y_1, z_1)$, $M2(x_2, y_2, z_2)$ and for any C belonging to the trajectory we may write:

$$\mathbf{OC} = \mathbf{OM_1} + \lambda \mathbf{M_1M_2} \tag{2}$$

where λ is a parameter in the range [0,1]. Let us calculate the leg length for a point C on the segment M_1M_2. The leg length ρ is the norm of the vector $\mathbf{AB}$. We have:

$$\rho^2 = ||\mathbf{AB}||^2 = \mathbf{AO}.\mathbf{AO}^T + \mathbf{CB}.\mathbf{CB}^T + 2(\mathbf{AO} + R\mathbf{CB_r}).\mathbf{OC}^T + 2R\mathbf{CB_r}.\mathbf{AO}^T + \mathbf{OC}.\mathbf{OC}^T \tag{3}$$

By using equation (2,1) this equation can be written:

$$\begin{aligned} \rho^2 \;\; = \;\; & R_1^2 + r_1^2 + R_1 r_1 \; R\mathbf{v}.\mathbf{u} + R_1 \mathbf{u}.(\mathbf{OM1} + \lambda \mathbf{M_1M_2})^T + r_1 R\mathbf{v}.(\mathbf{OM_1} + \lambda \mathbf{M_1M_2})^T + \\ & \lambda^2 \mathbf{M_1M_2}.\mathbf{M_1M_2}^T + \mathbf{OM_1}.\mathbf{OM_1}^T + 2\lambda \mathbf{OM_1}.\mathbf{M_1M_2}^T \end{aligned}$$

This equation can be written in various different forms:

$$R_1^2 + r_1^2 + a_0 \; R_1 r_1 + R_1(a_1\lambda + a_2) + r_1(a_3\lambda + a_4) + a_5\lambda^2 + a_6\lambda + a_7 = 0 \tag{4}$$
$$\mathcal{E}(R_1, r_1, \lambda, \rho) = 0 \tag{5}$$
$$F(\lambda) = A_2\lambda^2 + A_1\lambda + A_0 = 0 \tag{6}$$

For equation (4) the a_i coefficients are only dependent on the known design parameters and the coordinates of M_1, M_2. The structure of equation (4) leads to the following theorems:

Theorem 1 *For a given set of ρ, λ the equation defines an ellipsis in the $R_1 - r_1$ plane or has none solution in R_1, r_1.*

Theorem 2 *Let consider a set of R_1, r_1 (which defines a point M in the R_1, r_1 plane). If we have $\mathcal{E} \leq 0$ (i.e. the point M is inside the ellipsis) for a given λ (i.e. for a fixed position of the platform) then the corresponding length of the leg is lower or equal to ρ.*

2.1.1 Computing the allowable region for the maximum length constraint

Consider now the function $\mathcal{E}_{max}(\lambda) = \mathcal{E}(R_1, r_1, \lambda, \rho_{max})$ in the R_1, r_1 plane with λ in the range [0,1]. This function defines a set of ellipsis each of which is called a *maximal ellipsis*. If for any λ in the range [0,1] we have $\mathcal{E}(R_1, r_1, \lambda, \rho_{max}) \leq 0$ then for any position of the platform on the trajectory the leg length is lower of equal to ρ_{max}. Consequently the set of points $M(R_1, r_1)$ such that $\mathcal{E}_{max}(\lambda) \leq 0$ for any λ in [0,1] defines the allowable region for the maximum length constraint. This means that any such M point must be inside all the ellipsis in the set and therefore the allowable region with respect to the constraint $\rho \leq \rho_{max}$ is the intersection $\mathcal{I}$ of all the ellipsis of the set. We denote $\mathcal{E}_{max}(0)$ and $\mathcal{E}_{max}(1)$ the two ellipsis obtained for $\lambda = 0$ and $\lambda = 1$. The following theorems have been proved:

Theorem 3 *As λ vary the center of the corresponding ellipsis lie on a segment which in some cases can be reduced to a point. The angle between the main axis of the ellipsis and the R_1 axis is $\pi/4$.*

Theorem 4 *If a trajectory is such that $\theta = 0$ and $z_1 = z_2$ then the dimensions of every ellipsis in the set $\mathcal{E}(R_1, r_1, \lambda, \rho)$ are constant all along the trajectory. These dimensions will be constant whatever is the trajectory.*

Theorem 5 *If the ellipsis $\mathcal{E}_{max}(0)$, $\mathcal{E}_{max}(1)$ exists then the ellipsis exist for any value of λ in the range [0,1] or the intersection of all the ellipsis of the set is empty.*

Theorem 6 *The intersection $\mathcal{I}$ of all the ellipsis in the set is equal to*

$$\mathcal{E}_{max}(0) \cap \mathcal{E}_{max}(1)$$

Therefore the allowable region is simply the intersection of the ellipsis computed for the extreme points of the trajectory.

2.1.2 Computing the allowable region for the minimum length constraint

Consider now the function $\mathcal{E}_{min}(\lambda) = \mathcal{E}(R_1, r_1, \lambda, \rho_{min})$ in the R_1, r_1 plane. This function defines a set of ellipsis, each of which is called a *minimal ellipsis*. If for a given point M and a given λ we have $\mathcal{E}_{min}(\lambda) > 0$ then the corresponding length leg is greater than ρ_{min}. Therefore for any point belonging to the allowable region this relation as to be verified for all λ in [0,1]. Consequently any point in the allowable region must lie outside the region $\mathcal{U}$ defined by $\mathcal{E}_{min}(\lambda) = 0$. For computing the region $\mathcal{U}$ one possibility is to compute the union of ellipsis for discrete values of λ in the interval $[\lambda_m, \lambda_M]$ (figure 2).

In the case where $\theta = 0$ and $z_1 = z_2$ (all the ellipsis have therefore the same dimensions) an efficient method is to compute the ellipsis for the extreme points of the trajectory, compute the points on the ellipsis where the tangent is equal to the tangent of the line on which lie the center of the ellipsis and to join the points.

2.1.3 Computing the allowable region for all the leg length constraints

The computation of the allowable region for the leg length constraints is done using the following algorithm:

1. compute the maximal ellipsis for the extreme points of the trajectory
2. compute the intersection $\mathcal{I}$ of these two ellipsis. If there is none there is no allowable region.

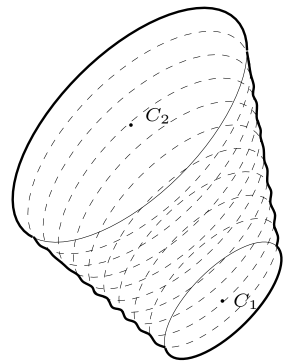

Figure 2: The computation of the union of the forbidden ellipsis. A discrete set of ellipsis are computed an approximation of the union is computed as the union of all these ellipsis (in thick line).

3. compute the union $\mathcal{U}$ of the minimal ellipsis.
4. substract $\mathcal{U}$ to $\mathcal{I}$ to get the allowable region

Figure 3 shows an example of this algorithm.

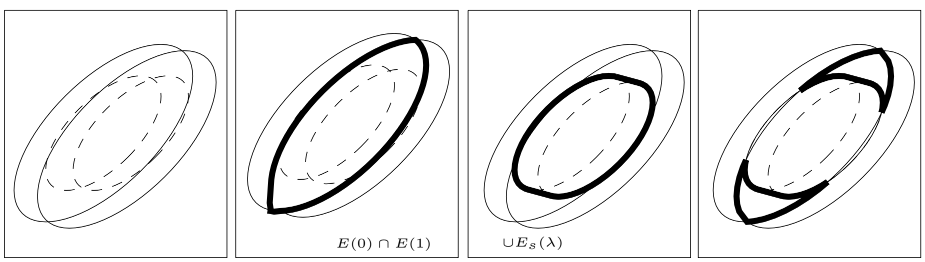

Figure 3: The computation of the allowable region. On the left side are drawn the maximal ellipsis for the extreme point of the trajectory (in thin line) and the minimal ellipsis for the same points (in dashed line). The intersection of the maximal ellipsis is shown on the second drawing. The union of the forbidden ellipsis is shown on the third drawing. The final allowable region is shown in thick line on the last drawing (first segment, trajectory 0, first set of length limits).

2.2 Computing the allowable region for a set of legs

Using the result of the previous sections it is easy to compute the allowable region for the leg lengths constraints for a set of lines. For each leg we express the desired trajectory in the frame specific to the leg and compute its allowable region. Taking a point in each allowable region defines the position of the attachment point and consequently a geometry for the robot.

Consider now the particular case where the attachment points on the base and on the platform lie on two horizontal circles. We compute the allowable region for each leg, the parameters R_1, r_1 being now the radii of the circles. The intersection of the allowable regions for each leg enables to compute the allowable region for the set of legs and therefore the radii of the circles. Any point in the region defines a robot whose workspace will contain the desired trajectories. An example is shown on figure 4: 64 trajectories with identical start point (0,0,50) and end point uniformly distributed on the sphere centered at the start point with radius 5 have been defined.

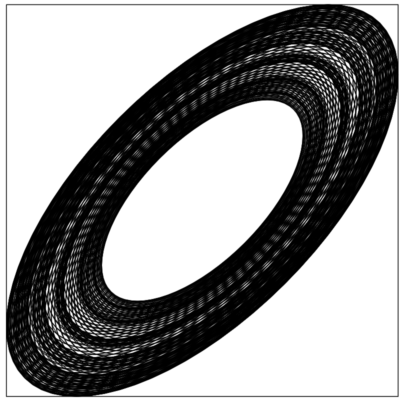

Figure 4: 64 trajectories with identical start point and end point uniformly distributed on a sphere centered at the start point have been defined. The central area defines the robots which include all the 64 trajectories.

3 Allowable region for the mechanical limits on the joints

3.1 Modelization of the mechanical limits

We have described in [9] a method for modeling the mechanical limits on the passive joints. The mechanical limits of a joint can be described through the definition of a pyramid whose apex is the joint center and whose faces are such that if the joint constraints are satisfied then the link will be inside the interior of the pyramid. For the joints attached to the base the center of this pyramid is located at point A (Figure 5).

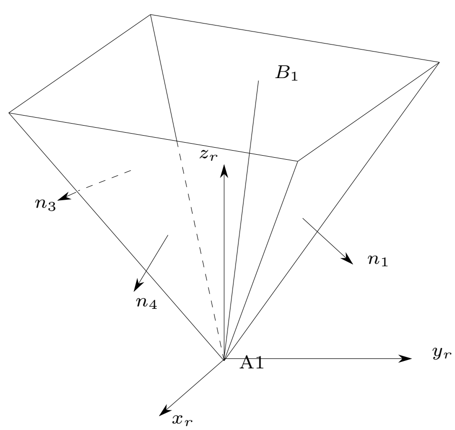

Figure 5: An example of modelization of a constraint on a passive joint located at A_1. If the mechanical limits of the joints are satisfied then link A_1B_1 is inside the volume delimited by the pyramid.

As for the constraints on the passive joints attached to the platform 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_i are satisfied then point A_i lie inside the pyramid. From this pyramid we deduce an *equivalent pyramid* $P_i^{,}$ to P_i, whose center is A_i, such that if A_i lie inside P_i then B_i lie inside $P_i^{,}$. All these pyramids may be defined by the normal to their faces.

3.2 Allowable region for one leg

Let $\mathbf{n^i_j}, j \in [1, k]$ denote the normals to the faces of the pyramid for the joint i. For a given position of the platform the leg A_iB_i will lie inside the pyramid (which means that the position of the leg fulfill the mechanical limits of the joints) if

$$\mathbf{A_iB_i}.\mathbf{n^i_j}^T \leq 0 \quad \forall\, j \in [1, k]$$

We consider a specific leg and a specific face of a pyramid. Using equation (2) the previous inequality can be written as:

$$R_1\mathbf{u}.\mathbf{n}^T + r_1R\mathbf{v}.\mathbf{n}^T + \lambda\mathbf{M_1M_2}.\mathbf{n}^T + \mathbf{OM_1}.\mathbf{n}^T \leq 0 \tag{7}$$

Let us denote $\mathcal{L}(R_1, r_1, \lambda)$ the left side of this inequality. The equation $\mathcal{L}(R_1, r_1, \lambda) = 0$ defines a pencil of lines in the R_1, r_1 plane. All this lines have a constant slope. This pencil of lines defines a region in the plane whose border is constituted of the line $\mathcal{L}(R_1, r_1, 0) = L_0$, $\mathcal{L}(R_1, r_1, 1) = L_1$. One of the lines L_0, L_1 separates the plane in two half-planes which are such that on one side of the line any point $M(R_1, r_1)$ is such that $\mathcal{L}(R_1, r_1, \lambda) \leq 0$ for any λ in [0,1] and on the other side $\mathcal{L}(R_1, r_1, \lambda) > 0$ at least for some values of λ in [0,1]. Therefore this line defines a half-plane which is the allowable region for the joint and for this face of the pyramid.

The process is repeated for each of the face of the pyramid, leading to a set of half-planes. The intersection of these half-planes will be a closed region which define the allowable region with respect to the mechanical limits on the joints. An example of this computation is shown in figure 6.

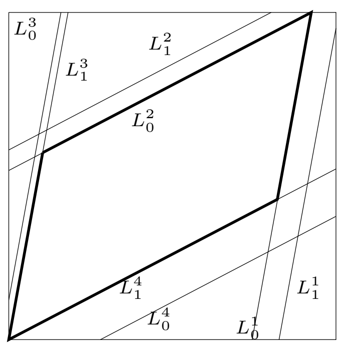

Figure 6: The mechanical limits of this particular joint is described by a four-faced pyramid. We have computed the separating half-plane for each of the faces by computing the L_0, L_1. The intersection of these half-planes define a closed region (in thick line) which is the allowable region for this joint.

3.3 Allowable region for all the legs

As a specific case we may assume that all the joints lie on two horizontal circles and we want to determine the possible radii of these two circles. The process is to compute all the set of allowed half-planes for all the joints and all the trajectories and compute their intersection. Evidently we can combine the results of the two previous sections. We first compute the allowable region for the leg lengths constraints, then the allowable region for the mechanical limits constraints. The intersection of the allowable regions defines the allowable region for all the constraints.

4 Allowable region for links interference

4.1 Principle of the computation of the allowable region

In this section we will assume that the links have no thickness. We want to determine the zone in the R_1, r_1 plane for which there is no interference between any pair of links, i.e. the zone for which the intersection point M between the lines i, j, if any, does not belong to both A_iB_i, A_jB_j. We will consider the case where $i = 1$ and $j = 2$ without lack of generality. If the two lines intersect then:

$$\mathbf{A_1A_2}.(\mathbf{A_1B_1} \times \mathbf{A_2B_2})^T = 0 \tag{8}$$

which can be written as:

$$-R_1r_1(b_1\lambda + b_2R_1 + b_3r_1 + b_4) = 0 \tag{9}$$

where the b_i's are constants given by the geometry and the trajectory of the platform. Various cases can now be considered:

1. $b_1 = b_2 = b_3 = b_4 = 0$: the line intersect whatever is the dimension of the robot for any position on the trajectory of the robot.
2. $b_1 = 0$: the line will intersect if the R_1, r_1 are on a line in the R_1, r_1 plane
3. in the general case the line will intersect if the R_1, r_1 are on a pencil of lines in the R_1, r_1 plane

Each of these cases defines a region $\mathcal{R}$ in the R_1, r_1 plane for which there is intersection of the two lines. In the first case this region is the full plane, in the second case the region is the line $b_2R_1+b_3r_1+b_4 = 0$ and in the last case the region is the zone delimited by the line $b_1+b_2R_1+b_3r_1+b_4 = 0$ and $b_2R_1+b_3r_1+b_4 = 0$ which are the extremal lines of the pencil. Consequently computing this region is easy. But we are interested only in the sub region where the links intersect. To compute this sub region we project the points A_1, A_2, B_1, B_2 onto the plane O, x, y of the reference frame and denote by a superscript ^{P} the projected points. It the segments A_1B_1, A_2B_2 intersect then their projection will intersect too and the intersection point will belong to the link in three cases. The first one is obtained when:

$$(\mathbf{A_1B_1}^P \times \mathbf{B_1B_2}^P)_z > 0 \qquad (\mathbf{B_2A_2}^P \times \mathbf{B_2B_1}^P)_z < 0 \tag{10}$$
$$(\mathbf{A_1A_2}^P \times \mathbf{A_1B_1}^P)_z < 0 \qquad (\mathbf{A_2A_1}^P \times \mathbf{A_2B_2}^P)_z > 0 \tag{11}$$

where the subscript $_z$ denotes the z componant of the vector. The second case is obtained when:

$$(\mathbf{A_1B_1}^P \times \mathbf{B_1B_2}^P)_z < 0 \qquad (\mathbf{B_2A_2}^P \times \mathbf{B_2B_1}^P)_z > 0 \tag{12}$$
$$(\mathbf{A_1A_2}^P \times \mathbf{A_1B_1}^P)_z > 0 \qquad (\mathbf{A_2A_1}^P \times \mathbf{A_2B_2}^P)_z < 0 \tag{13}$$

The last remaining case is obtained when $A_1^P, A_2^P, B_1^P, B_2^P$ are on the same line (in which case the previous inequalities become equalities). This may happen at one point on the trajectory or all along the trajectory in which case the points A_1, A_2, B_1, B_2 lie in the same horizontal plane.

The quantities which appear on the left side of the inequalities can be expressed as function of λ, R_1, r_1. They have all the same generic form:

$$R_1(e_1\lambda + e_2R_1 + e_3r_1 + e_4) \;\text{ or }\; r_1(e_1\lambda + e_2R_1 + e_3r_1 + e_4)$$

which defines a pencil of lines. It must be first noted that the inequalities may define unbounded region. As the other constraints on the workspace lead to bounded region we will consider only a limited portion of the R_1, r_1 plane, for example a square whose dimensions is equal to the maximum dimension of the rectangle which enclosed all the maximal ellipsis. After the computation of $\mathcal{R}$ we consider each set of inequalities. We divide then the square in four regions defined by

$$R_1 \geq 0 \;\; r_1 \geq 0 \qquad\qquad R_1 \geq 0 \;\; r_1 \leq 0$$
$$R_1 \leq 0 \;\; r_1 \geq 0 \qquad\qquad R_1 \leq 0 \;\; r_1 \leq 0$$

In each of this region Q_i the sign of the inequalities is now fully defined by four inequalities:

$$e_1^i \lambda + e_2^i R_1 + e_3^i r_1 + e_4^i \leq \quad \text{or} \geq 0$$

By dealing with this four inequalities we are then able to determine the region of the square for which links interference will occur. By repeating this process for every pairs of links and taking the union of the results we get the region of the R_1, r_1 plane for which at least one pair of links will interfere. Figure 7 shows an example of such computation.

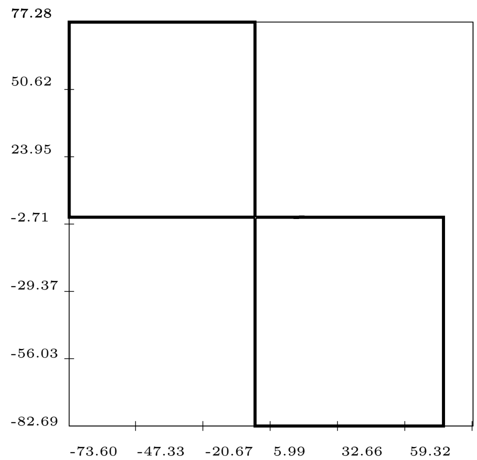

Figure 7: The region in the R_1, r_1 for which link 0 will interfere with some other link (trajectory 2, first set of length limit).

5 Verifying all the constraints

Basically if we want to determine the geometries of the robots such that all the attachement points lie on two circles and which verify all the constraints we compute the allowable regions defined for the leg length constraints and then compute the intersection of the result with the allowable region defined for the mechanical limits on the joint. Then we compute the forbidden region for the link interference and substract it to the previous region. The result defines the location of all the possible attachment points. Figures 8 shows an example of such zone. This figure shows that effectively the two trajectories lie inside the workspace of a robot whose parameters have been taken inside the allowable region.

6 Conclusion

The algorithm presented in this paper enables to compute all the possible location of the attachment points for any robot which can reach a specified workspace. Afterward some other criterion can be used to determine an "optimal" robot using a numerical algorithm with a search domain which is now considerably restricted. The possible criterion might be:

- to minimize the maximum of the articular forces when the robot moves a given load in the specified workspace
- to minimize the maximum of the positionning errors for the platform for a given error of the sensors measuring the leg length

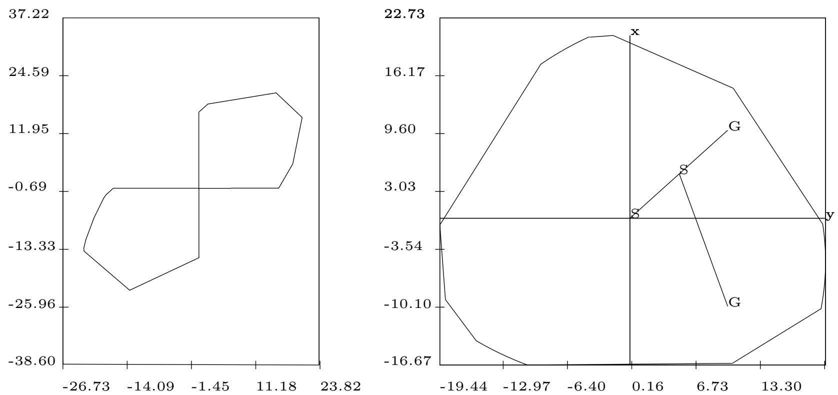

Figure 8: An allowable zone for the whole set of constraints. A robot has been defined by taking the values of R_1, r_1 inside the allowable region. The figure presents the workspace border and the trajectories.

- to maximize the maximal velocity of the platform for given velocities of the actuators
- the robot with no singularities in the workspace

The last point can be solved by computing the singularities surfaces of the robot using the method described in [8] and checking their intersection with the specified workspace. The remaining points are still open problem and will constitute the object of our next research.

7 Appendix: Numerical data of the examples

Angles α_i, β_i in degree and minimum and maximum length of the legs (first set)

link	1	2	3	4	5	6
α_i	35	145	155	265	275	25
β_i	85	95	205	215	325	335
ρ_{min}	30	30	30	30	30	30
ρ_{max}	40	40	40	40	40	40

Trajectory 0						Trajectory 2					
0	0	20	0	0	0	5	0	20	0	20	0
10	10	20	0	0	0	-10	10	20	0	20	0

References

[1] Claudinon B. and Lievre J. Test facility for rendez-vous and docking. In *36th Congress of the IAF*, pages 1–6, Stockholm, October, 7-12, 1985.

[2] Gosselin C. *Kinematic analysis optimization and programming of parallel robotic manipulators.* Ph.D. Thesis, McGill University, Montréal, June, 15, 1988.

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

[4] Gosselin C. and Angeles J. The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator. *J. of Mechanisms, Transmissions and Automation in Design*, 110(1):35–41, March 1988.

[5] 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.

[6] Liu K., Fitzgerald M.K., and Lewis F. Some issues about modeling of the Stewart platform. In *2nd Int. Symp. on Implicit and Robust systems*, Warsaw, 1991.

[7] Liu K., Fitzgerald M.K., and Lewis F. Kinematic analysis of a Stewart platform manipulator. *IEEE Trans. on Industrial Electronics*, 40(2):282–293, April 1993.

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

[9] Merlet J-P. Détermination de l'espace de travail d'un robot parallèle pour une orientation constante. *Mechanism and Machine Theory*, 29(8):1099–1113, November 1994.

[10] Merlet J-P. Trajectory verification in the workspace for parallel manipulators. *Int. J. of Robotics Research*, 13(4):326–333, August 1994.

[11] Merlet J-P. Designing a parallel robot for a specific workspace. Research Report 2527, INRIA, April 1995.

[12] Stoughton R. and Kokkinis T. Some properties of a new kinematic structure for robot manipulators. In *ASME Design Automation Conf.*, pages 73–79, Boston, June, 28, 1987.