Kinematics and synthesis of cams-coupled parallel robots

Jean-Pierre Merlet

Proceeding of CK2005, International Workshop on Computational Kinematics
Cassino, May 4-6, 2005

Paper XX-CK2005

Kinematics and synthesis of cams-coupled parallel robots

J-P. Merlet
INRIA Sophia Antipolis,
BP 93, 06902 Sophia-Antipolis, France
e-mail: Jean-Pierre.Merlet@sophia.inria.fr

ABSTRACT- We consider parallel robots, that will be called *cams-coupled parallel robots*, whose mobile platform has surfaces that are constrained to be in contact at a point with surfaces located on the base. Articulated passive fixed length legs connect also the base to the platform while active legs allow to control the motion of the platform. We investigate the mobility of such robot and shows how the inverse and direct kinematics can be solved for arbitrary contact surfaces. We present then preliminary results regarding the synthesis of such robot.

KEYWORDS: cams, synthesis, trajectory planning, parallel robots

INTRODUCTION

Parallel robots with less than 6 dof have attracted a lot of attention this recent years as they have possibility of application. The design of such robots has followed two possible approaches:

- the mobile platform is constrained by an auxiliary mechanism so that it has only the desired dof
- the mechanical structure itself is designed in such way that it is constrained to have only the desired dof

Although the second approach has led to the development of interesting robot such as the Delta robot [1] or the H4 robot [2] it has a main drawback: the geometrical assumptions that are required for the mobile platform to have less than 6 d.o.f. will never been fully satisfied in practice and the robot will usually exhibit parasitic d.o.f.

In the first approach the auxiliary mechanism is usually a serial mechanism. We investigate here another type of constraint mechanism (figure 1):

- the mobile platform will have m surfaces S_i, each of them being constrained to be in contact with a corresponding surface T_i on the base. It will be assumed that surfaces T_i, S_i are in contact at a point and that some constraint mechanisms impose that the surfaces remain in contact
- the mobile platform and the base will be connected by n legs of fixed length attached to both the base and the platform by ball-and-socket joints

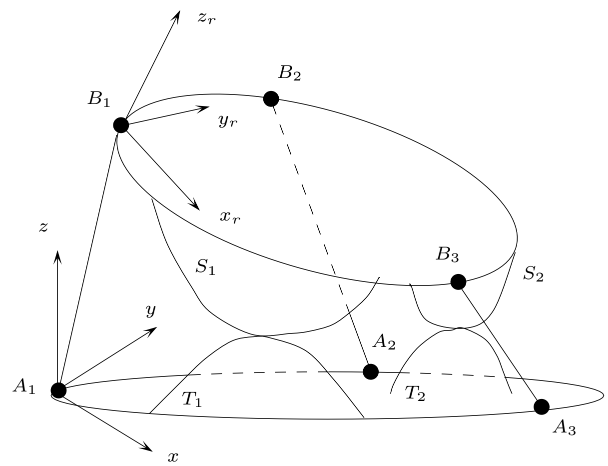

Figure 1: An example of cams-coupled parallel robot

The interest of such mechanism, that will be *called cams-coupled parallel robot*, is that it will allow a broader variety of constrained motion than classical serial mechanisms. Note also a possible biological application as a constraint mechanism of this type has been used by Parenti-Castelli to model the motion of the knee joint [9, 10].

Such constraint mechanism will impose a mobility M to the platform and these DOF may be controlled by adding, for example, M actuated legs.

We define a reference frame $O, \mathbf{x}, \mathbf{y}, \mathbf{z}$ and $A_i = (xa_i, ya_i, za_i)$ will denote the attachment points of the legs of fixed length ρ_i on the base. The attachment points of these legs on the platform will be denoted B_i and we define a mobile frame as $B_1, \mathbf{x_r}, \mathbf{y_r}, \mathbf{z_r}$. A superscript r will be used to denote vectors and coordinates in the mobile frame. In the mobile frame the coordinates of B_i will be denoted xb_i^r, yb_i^r, zb_i^r and without loss of generality we may impose $xb_1^r = yb_1^r = zb_1^r = 0$. In the reference frame the coordinates of B_i will be denoted xb_i, yb_i, zb_i and the pose vector $\mathbf{X}$ of the platform will be defined by the coordinates xb_1, yb_1, zb_1 and the three Euler's angles ψ, θ, ϕ that allow to define the rotation matrix between the mobile and the reference frame.

MOBILITY ANALYSIS

Let S_i be a surface on the mobile platform that is constrained to be in contact with a surface T_i on the base. These surface are defined by the implicit equations

$$F^r_{S_i}(x^r, y^r, z^r) = 0 \qquad F_{T_i}(x, y, z) = 0 \tag{1}$$

Let the vectors $\mathbf{N^r_{S_i}}$ and $\mathbf{N_{T_i}}$ be defined as:

$$\mathbf{N^r_{S_i}} = (\frac{\partial F_{S_i}}{\partial x^r}, \frac{\partial F_{S_i}}{\partial y^r}, \frac{\partial F_{S_i}}{\partial z^r}) \quad \mathbf{N_{T_i}} = (\frac{\partial F_{T_i}}{\partial x}, \frac{\partial F_{T_i}}{\partial y}, \frac{\partial F_{T_i}}{\partial z})$$

These vectors are directed along the normal of S_i, T_i and if the surfaces are in contact at point $M_i = (x_i, y_i, z_i)$ we must have

$$\mathbf{R}\mathbf{N}^{\mathbf{r}}_{\mathbf{S_i}} \times \mathbf{N}_{\mathbf{T_i}} = 0 \tag{2}$$

which induces 2 independent equations. Hence each surface contact led to 4 constraints equations that involve the 3 unknowns x_i, y_i, z_i. If m is the number of imposed surfaces contact we get $4m$ constraint equations for a total of $3m + 6$ unknowns. As n articulated legs impose n constant length constraint equations (assumed to be independent) the mobility M of the robot is obtained as $6 - m - n$. Hence, assuming that m cannot be 0 we get the following possibilities for M: 5 ($m = 1, n = 0$), 4 ($m = 1, n = 1$), ($m = 2, n = 0$), 3 ($m = 1, n = 2$), ($m = 2, n = 1$), ($m = 3, n = 0$), 2 ($m = 1, n = 3$), ($m = 2, n = 2$), ($m = 3, n = 1$), ($m = 4, n = 0$), 1 ($m = 1, n = 4$), ($m = 2, n = 3$), ($m = 3, n = 2$), ($m = 4, n = 1$), ($m = 5, n = 0$).

KINEMATICS

Inverse Kinematics

Consider a cams-coupled parallel robot with mobility M and assume that we fix M parameters among the pose parameters of the platform. This implies that the platform may be only in a finite number of poses (possibly 0 if the robot cannot be assembled) and we want to determine these poses. Let $M_i, i \in [1, m]$ be the m contacts points between the surfaces S_i, T_i. We will denote xm_i, ym_i, zm_i the coordinates of M_i in the reference frame. In the mobile frame the coordinates xm_i^r, ym_i^r, zm_i^r of M_i are the components of the vector $\mathbf{CM_i}^r$. We have

$$\mathbf{B_1 M_i}^r = R^T \mathbf{B_1 M_i}$$

Hence we may express xm_i^r, ym_i^r, zm_i^r as functions of xb_1, yb_1, zb_1 and ψ, θ, ϕ. Hence the unknowns $\mathcal{U}$ of the kinematics problem are the $6 - M$ unknown components of $\mathbf{X}$ together with the $3m$ coordinates xm_i, ym_i, zm_i.

We have $2m$ constraints equations

$$F^r_{S_i}(xm_i^r, ym_i^r, zm_i^r) = 0 \qquad F_{T_i}(xm_i, ym_i, zm_i) = 0 \tag{3}$$

and $2m$ normal equations

$$\mathbf{R}\mathbf{N}^{\mathbf{r}}_{\mathbf{S_i}}(\mathbf{xm_i^r}, \mathbf{ym_i^r}, \mathbf{zm_i^r}) \times \mathbf{N}_{\mathbf{T_i}}(\mathbf{xm_i}, \mathbf{ym_i}, \mathbf{zm_i}) = 0 \tag{4}$$

For the passive legs we have the n constraint equations

$$||\mathbf{A_i B_i}||^2 = ||\mathbf{A_i O} + \mathbf{OB_1} + R\mathbf{CB_1^r}||^2 = \rho_i^2 \tag{5}$$

Solving the $4m + n$ equations system (3,4,5) allows to solve the inverse kinematics problem i.e. finding the 6-M unknown pose parameters of the platform.

Assume for example that we have $m = 2, n = 3$ which implies that the robot has mobility 1. We fix one of the pose parameters and the five remaining one are unknowns. We have also 2 unknown contact points between the surfaces and the total amount of unknowns is 11 while the number of equations is also 11.

Note that the unknowns parameters in $\mathbf{X}$ may easily be bounded. Indeed B_1 belongs to a sphere centered in A_1 with radius ρ_1. Hence we get the following bound:

$$xb_1 \in [xa_1 - \rho_1, xa_1 + \rho_1] \quad yb_1 \in [ya_1 - \rho_1, ya_1 + \rho_1] \quad zb_1 \in [za_1 - \rho_1, za_1 + \rho_1]$$

while the Euler's angles may be restricted to lie in the range $[0, 2\pi]$. As for the coordinates of the M_i we may assume that the platform is a rigid body with finite dimensions that will impose bounds on these coordinates.

In general it will difficult to find a closed-form solution for the system (3,4,5) and we are interested in an approach that will work whatever are the constraint surfaces. If the surfaces have an algebraic definition the system of equations may be solved using homotopy [12], Gröebner basis [11] or elimination theory. But this approach are quite sensitive to the number of equations and to their degree. As the unknowns are bounded a convenient generic tool for solving the equations system is an interval analysis-based solver [4, 6] that will provide all the solutions within the search space. It allows to process any non-linear equations (not being restricted to algebraic equations) and to deal easily with additional mechanical constraints such as limits on the motion of the passive joints. We use our C++ interval analysis library `ALIAS`[1] which is interfaced with Maple. Basically we just define in Maple the cams constraints and the geometry of the base and platform and the equations to solve are derived automatically by Maple that uses for that purpose specific Maple procedures of the `ALIAS` interface to create the necessary C++ code and run it. Note however that interval analysis should not be considered as a "black box" and need some expertise to be used efficiently.

The solving method we use allows one to determine *all* solutions in the search space and these solutions may be computed *exactly* in the sense that we will determine intervals in which there is exactly one solution and that this solution can be computed with a numerical scheme that is guaranteed to converge to the solution (using this scheme within a software allowing multi-precision calculation we are able to compute the solution with an arbitrary number of digits).

The only case in which no unique solution will be determined is when the system has a singular solution. This is also detected by our solving method as will be seen in the example.

Application example

We consider a robot with 2 spherical surface constraints and 3 passive legs. The coordinates of the attachments points of the legs on the base are (in term of any length unit)

$$A_1 = [0, 0, 0] \quad A_2 = [10, 5, 0] \quad A_3 = [-10, 5, 0]$$

and their attachment points on the platform have as coordinates in the mobile frame

$$B_1 = [0, 0, 0] \quad B_2 = [15, 5, 0] \quad B_3 = [15, 5, 0]$$

while their length are $\rho_1 = 12.806, \rho_2 = 15.329, \rho_3 = 15.329$. On the base we have two spherical surfaces T_1, T_2 with radius $r_1 = 5$ and whose centers P_1, P_2 are located at [-7,15,0] and [7,15,0]. The spherical surfaces S_1, S_2 on the platform have a radius r_2 of 5.04580 and their centers C_1, C_2 are located at [-7,7.01168,-4.0657], [7,7.01168,-4.0657]. This mechanism may be seen as a 5-SS mechanism as the 5 points B_1, B_2, B_3, C_1, C_2 are constrained to lie on spheres centered at A_1, A_2, A_3, P_1, P_2 with radii $\rho_1, \rho_2, \rho_3, r_1 + r_2$. A direct consequence is that any point of the mechanism lies on a curve of order 40

[1] www.inria-sop.fr/coprin/logiciel/ALIAS/ALIAS.html

(this is used as one proof that the Gough platform forward kinematics has at most 40 solutions [5, 7]). Consequently if we fix one coordinate of any point of the platform the kinematics will have at most 40 solutions.

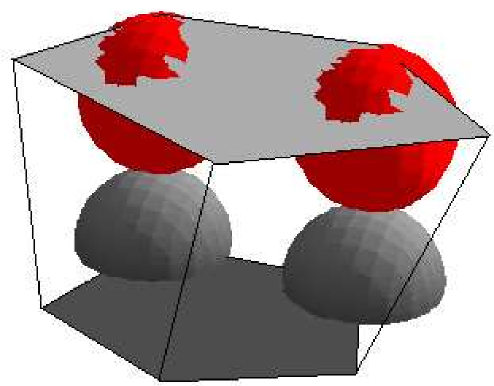

Figure 2: A cams-coupled parallel robot with two set of spherical constraint surfaces

Fixing $\phi = 0$ and restricting the angles ψ, θ to lie in the range $[-\pi/2, \pi/2]$ to avoid leg and platform interference we find two possibles poses for the moving platform:

$$x_c = 0 \;\; y_c = 8.5138 \;\;\; z_c = 9.56635 \;\; \psi = 0 \;\; \theta = 32.57768^\circ$$
$$x_c = 0 \;\; y_c = 8 \;\;\; z_c = 10 \;\; \psi = 0 \;\; \theta = 30^\circ$$

The computation time for solving this example is about 55 minutes on an EVO 410, 1.2 GHz.

In that particular case it must be noted that instead of using the Cartesian coordinates for the 2 surface contact points we may have used spherical coordinates: this will have led to a system of only 9 equations in 9 unknowns that is however solved in a similar computation time.

DIRECT KINEMATICS

As seen previously the mobility of a robot with n passive legs and m cams is $6 - m - n$ and this passive system induces $4m + n$ constraint equations. Adding $6 - m - n$ active links with one d.o.f. will allow to control the robot. These links induce $6 - m - n$ constraint equations and the total amount of constraint equations is $6 + 3m$ for $6 + 3m$ unknowns. Provided that the equations are independent the platform poses belong to a finite set. The direct kinematics is obtained by solving this set of equations.

In the previous example we have added a fourth active leg whose length can be adjusted to control the motion of the robot. The leg is attached to the base at $A_4 = (0, -10, 0)$ and on the platform at $B_4 = (0, -5, 0)$. We assign a value of $\rho_4 = 15.5921$ to the length of this leg. It must be noted that the robot may be seen as a Gough platform for which the direct kinematics may be solved with specific procedures [8]. But for the sake of generality we have used a general solving procedure of `ALIAS`. The search space can easily be determined and we have restricted the z coordinate of B_1 to be positive and the three angles ψ, θ, ϕ to lie in the range $[-\pi/2, \pi/2]$.

Using a cluster of 12 machines (interval analysis is appropriate for distributed computation) we have found 3 solutions to the system of 12 equations in about 5 hours and 50 minutes:

1. $x_c = -4.6237 \quad y_c = 7.560187 \quad z_c = 9.2447 \quad \psi = 0.51046 \quad \theta = 0.2219 \quad \phi = -0.2066$
2. $x_c = 4.6237 \quad y_c = 7.560187 \quad z_c = 9.2447 \quad \psi = -0.51046 \quad \theta = 0.2219 \quad \phi = 0.2066$
3. $x_c = 0 \quad y_c = 8 \quad z_c = 10 \quad \psi = 0 \quad \theta = 0.5236 \quad \phi = 0$

The interval analysis based solver allows to determine that the 2 first solutions are not singular while the third one is singular (and therefore cannot be found using the classical Newton-Raphson iterative scheme).

It can be seen that finding *all* the solutions of the direct kinematics will be usually time consuming, especially as we use a generic scheme that works whatever is the robot structure. But having to determine all solutions (and without any a priori information on the location of the platform) is seldom necessary. Indeed direct kinematics will be usually used in a real-time context but with a priori information on the possible location of the solution. For example we may have determined the pose of the robot at a given sampling time and wish to determine what is the pose of the robot at the next sampling time. Being given the maximal velocity of the robot we may determine a ball within which should lie the solution. Table 1 gives the computation time of the direct kinematics for various ball diameter.

Ball size (unit length,rd)	0.1,0.06	0.06,0.06	0.06,0.006	0.006,0.006
Time (ms)	11630	4920	540	290

Table 1: Computation time of the direct kinematics as a function of the search space diameter

Note that we will determine only the solutions that lie within the search space while the Newton scheme may converge to a solution that is not in the search space. A consequence is that if more than one solution is found it will be better to stop the robot as it is not possible to determine what is its current pose (if the search space is small this also imply that we are close to a singular configuration).

UPPER BOUNDS OF THE NUMBER OF SOLUTIONS FOR THE DIRECT AND INVERSE KINEMATICS

Let assume that the constraint surfaces S_i, T_i are defined by algebraic equations of total degree d_{S_i}, d_{T_i}. The rotation matrix may be defined using an algebraic form with second order coefficients and with a additional second order equation. Hence the system of equations 3 has m equations $F^r_{S_i}$ of degree $2d_{S_i}$ and m equations F_{T_i} of degree d_{T_i}. For the equations 4 the components of the vectors $\mathbf{RN^r_{S_i}}$ will have degree $d_{S_i}+1$ while the vectors $\mathbf{N_{T_i}}$ will have degree $d_{T_i}-1$. Hence the degree of the components of their cross-product will be $d_{S_i}+d_{T_i}$ and the system 4 will have $2m$ equations of such degree. For the passive leg the degree of the equation will be 3. For the inverse kinematics we have to solve $4m+n$ equations system (3,4,5). Using Bezout theorem the upper bound U of the number of solution will be

$$U = 2 \prod_{i=1}^{i=m}(2d_{S_i}) \prod_{i=1}^{i=m}(d_{T_i}) \left(\prod_{i=1}^{i=m}(d_{S_i}+d_{T_i})\right)^2 3^n$$

Assume for example that we have $m = 2, n = 3$ and that the constraint surfaces have degree 2: we get $U = 884736$. Note that this number is clearly an overestimated upper bound: for spherical surfaces we have seen that in general the real upper bound will be 40.

As for the direct kinematics we will add $6-m-n$ legs. Hence the factor 3^n in the previous expression should be substituted by 3^{6-m}. For $m = 2, n = 3$ we will find $U = 2654208$.

PATH ANALYSIS

Consider a robot with mobility 1. We have seen in the above section that after having fixed one of the pose parameters X_1 we are able to determine the possible poses of the platform i.e the pose parameters $X_2, \ldots, X_6$. It may be of interest to draw $X_2, \ldots, X_6$ as function of X_1 when X_1 lies in the range $[X_1^0, X_1^l]$. A first possibility will be to compute these curves by selecting a finite number of values for X_1 and to solve the kinematic problem at each point. Apart from being computer intensive this approach has one drawback: it may be difficult to determine how to connect the solution points obtained for 2 successive values of X_1.

We propose another approach that solve this problem and is based on Kantorovitch theorem [3]. Let a system of n equations in n unknowns:

$$G = \{G_i(x_1, \ldots, x_n) = 0, i \in [1, n]\}$$

each G_i being at least C^2. Let $\mathbf{x_0}$ be a point and $U = \{\mathbf{x}/||\mathbf{x} - \mathbf{x_0}|| \leq 2B_0\}$, the norm being $||A|| = \text{Max}_i \sum_j |a_{ij}|$. Assume that $\mathbf{x_0}$ is such that the Jacobian matrix of the system has an inverse Γ_0 at $\mathbf{x_0}$ and that the following inequalities hold:

$$||\Gamma_0|| \leq A_0 \qquad ||\Gamma_0 G(\mathbf{x_0})|| \leq 2B_0$$
$$\sum_{k=1}^{n} |\frac{\partial^2 G_i(\mathbf{x}}{\partial x_j \partial x_k}| \leq C \ \text{ for } i, j = 1, \ldots, n \text{ and } \mathbf{x} \in U \tag{6}$$

If the constants A_0, B_0, C satisfy $2nA_0B_0C \leq 1$, then there is an unique solution of $G = 0$ in U and the Newton method used with $\mathbf{x_0}$ as estimate of the solution will converge toward this solution. Assume that we have been able to determine the set of r possible platform poses $\{U_1^0, \ldots, U_r^0\}$ at X_1^0 i.e. we have been able to solve the set of equations $G_{X_1^0}$ =(3,4,5) for $X_1 = X_1^0$. We now want to determine the platform poses $\{U_1^1, \ldots, U_r^1\}$ at $X_1 = X_1^2 = X_1^0 + \delta$ in such way that a continuous change from X_1^0 to X_1^1 will transform U_j^0 into U_j^1.

Using interval analysis we may compute for the system $G_{X_1^1}$ obtained for $X_1 = X_1^1$ upper bound for the values of A_0, B_0, C at the pose U_j^0 using the left hand part of the inequalities (6).

If $2nA_0B_0C \leq 1$ the system $G_{X_1^1}$ has a unique solution W in a ball that enclose U_j^0 and hence during a continuous transformation of X_1 from X_1^0 to X_1^1 the solution U_j^0 will be transformed into W: consequently $U_j^1 = U$.

If $2nA_0B_0C > 1$ we define $X_1^1 = X_1^0 + \delta/2^k$ and start again with $k = 1, 2 \ldots$. As B_0 is proportional to $G_{X_1^1}(U_j^0)$ it will decrease and we will be able to find a k such that at X_1^1 we get $2nA_0B_0C \leq 1$.

At this point we have a solution for $X_1 = X_1^1$ and we may repeat the procedure with $X_1 = X_1^2 = X_1^1 + \delta$ until X_1 reaches X_1^l.

This procedure will fail only in two cases:

- due to numerical round-off errors we will never satisfy $2nA_0B_0C \leq 1$: in that case we have to use multi-precision arithmetic to improve the calculation of these constants
- the Jacobian matrix is singular in the vicinity of X_1^j. A local analysis will show if the singularity occurs because 2 branches collapse (and the trajectory is well defined) or because 2 branches cross which means that we will not be able to control the robot around this configuration. As this should be avoided we have hence a nice way to determine the feasibility of the use of the robot

APPLICATION EXAMPLE

We have considered the previous mechanism with spherical surfaces constraints and have calculated the trajectory that will be obtained when the angle ϕ changes from 0 to 0.25 radian, plotting 2500 points on the two branches. The computation time on an EVO 410, 1.2 GHz, is about 5 mn. Figure 3 shows the variation of θ with respect to ϕ. Figure 4 shows the trajectory of point B_1 together with the orientation

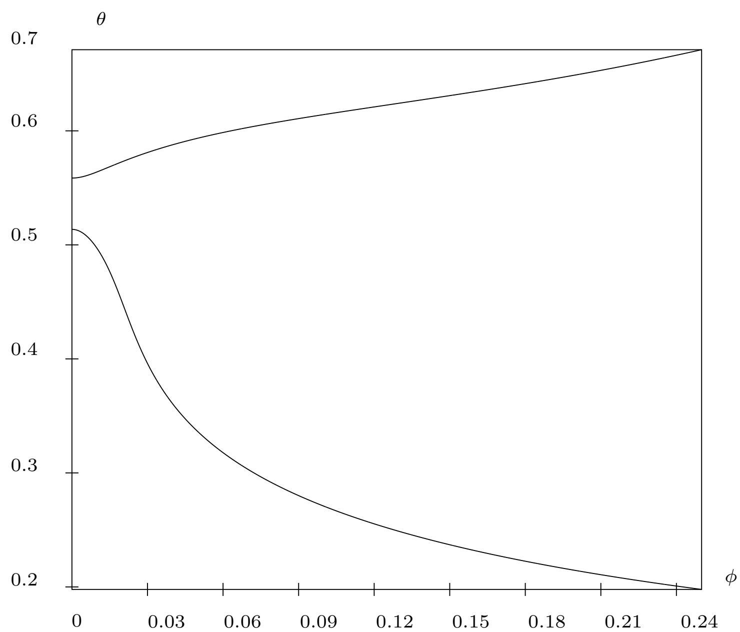

Figure 3: Variation of θ as a function of ϕ

of the mobile frame.

SYNTHESIS OF 1 D.O.F.ROBOT

In this section we will consider a 1 d.o.f. with a passive given constraint mechanism. Parenti-Castelli [9, 10] has considered such mechanism with spherical constraint surfaces to establish a model of a knee joint. Although their model has a trajectory that was closer to experimental data than previously proposed

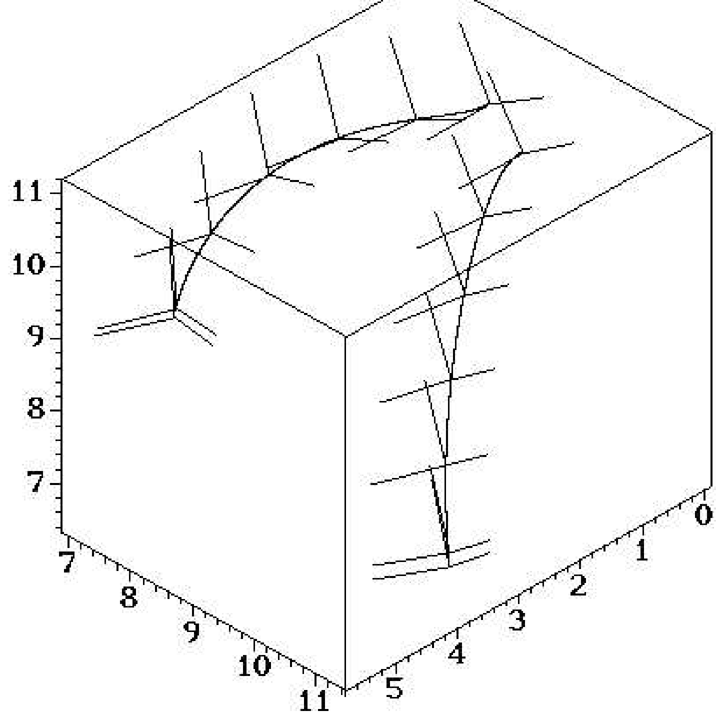

Figure 4: Trajectory of the center of the platform with a representation of the mobile frame

models there was still some large discrepancies. It is indeed quite difficult to *synthesize* such mechanism for following a given trajectory.

SYNTHESIS WITH PRECISION POSES

A first synthesis approach consists in imposing that the mechanism should be able to reach specific poses that are completely or partially defined (for example in the mechanism considered by Parent-Castelli only the orientation is important). As soon as we impose that the trajectory of the robot should go through a precision pose for which we fix the value of a pair of pose parameters the equations (3,4,5) is an over-constrained system of equations with one more equation than unknowns. Consequently we may assume that one of the parameters defining the geometry of the constraint mechanism is an unknown that should be determined so that we end up with a square system of equations.

Hence fixing two coordinates of a precision pose induces a set of $4m + n$ equations with $4m + n$ unknowns, one of which is a geometry parameters. If n coordinates of a precision pose are fixed, then we will have $n - 1$ geometry parameters as unknowns. On the other hand we may choose to use k precision poses with l fixed parameters. We will end up with a square system of $k(4m+n)$ equations with $k(l-1)$ geometry parameters in the unknowns.

As an example we have considered our mechanism with spherical rolling contact, the sphere on the base having a radii of 5, 11.859 while the sphere on the platform have a radii of 4.7, 2.541. The location of the attachment points have the same value than in the previous example but the length ρ_2, ρ_3 are free while the platform should be able to reach the orientation defined by $\psi = \pi/10, \theta = \pi/6, \phi = \pi/10$. We end up with a system of 11 equations in 11 unknowns: the lengths ρ_2, ρ_3, the coordinates of the platform center and the coordinates of the 2 contact points between the spheres. The solutions of this system are again obtained using the `ALIAS` solver.

Constraining the leg lengths to lie in the range [0,40] we find 4 solutions to the system, two of which having the same leg lengths and differing only by the values of the point coordinates. So the design solution are:

$$\rho_2 = 18.4682 \quad \rho_3 = 8.507$$
$$\rho_2 = 19.9878 \quad \rho_3 = 11.27608$$
$$\rho_2 = 20.821 \quad \rho_3 = 10.6685$$

On a Dell D400 laptop the computation time is 49 seconds.

In another example we have fixed $x_c = 2$ and $\psi = \pi/10, \theta = \pi/6$ while the geometrical parameters are still ρ_2, ρ_3. A set of 6 solutions has been found in about 6 minutes with 4 distinct solutions for the ρ_2, ρ_3 unknowns:

$$\rho_2 = 20.821 \quad \rho_3 = 10.668$$
$$\rho_2 = 19.561 \quad \rho_3 = 5.373$$
$$\rho_2 = 14.677 \quad \rho_3 = 15.182$$
$$\rho_2 = 25.4141 \quad \rho_3 = 5.598$$

Using the same precision point we have now fixed $\rho_2 = 20.821$, $\rho_3 = 10.668$ but we have assumed that the constraint surfaces on the base may be written as:

$$(x+7)^2 + (A_1 y - 15)^2 + z^2 = 25$$
$$(x-7)^2 + (A_3 y - 15)^2 + z^2 = 140.6348$$

The geometrical parameters are now the unknowns A_1, A_3 that change the shape of the constraint surfaces. Assuming that these parameters lie in the range [0.1,10] four solutions have been found in about 7 seconds

A_1	1	1.488	1	1.488
A_3	0.6609	0.6609	1	1

If the same geometrical transformation is performed on the constraint surfaces on the platform we will find 2 solutions.

Assume now that we define two precision poses by the values of their orientation angles. We choose:

$$\psi = \pi/10 \quad \theta = \pi/6 \quad \phi = \pi/10$$
$$\psi = \pi/6 \quad \theta = \pi/6 \quad \phi = \pi/6$$

The unknowns are the coordinates of B_1 and the coordinates of the contact points between the spherical surfaces i.e. 18 unknowns for 22 equations: consequently we may choose 4 geometrical parameters for the synthesis. We have chosen the lengths ρ_2, ρ_3 of leg 2 and 3 and the x coordinates of their attachment points on the platform. The leg lengths are restricted to lie in the range [7,23] while the x coordinates of B_2, B_3 lie in the ranges [8,16],[-16,8]. The 3 following solutions have been obtained in about 48mn:

ρ_2^2	ρ_3^2	x_{B_2}	x_{B_3}
290.947	124.1588	10.191	-14.8762
278.97	62.827	8.699	-8.0716
264.315	79.841	8.1947	-6.7929

Similarly for two precision poses we may choose as geometrical parameters 4 unknowns A_1, A_2, A_3, A_4 such that the constraint surfaces on the base may be written as:

$$(A_1x + 7)^2 + (A_2y - 15)^2 + z^2 = 25$$
$$(A_3x - 7)^2 + (A_4y - 15)^2 + z^2 = 140.6348$$

The precision poses are defined by the ψ, θ, ϕ angles with the values $\pi/10, \pi/6, \pi/10$ and $\pi/6, \pi/10, \pi/8$ Two solutions are found in about 49 mn of computation time:

A_1	A_2	A_3	A_4
3.459	3.3018	1.3835	1.4328
3.477	1.0148	1.3835	1.4328

Similarly for three precision poses we may choose as geometrical parameters 4 unknowns A_1, A_2, A_3, A_4 such that the constraint surfaces on the base may be written as:

$$(A_1x + 7)^2 + (A_2y - 15)^2 + z^2 = 25$$
$$(A_3x - 7)^2 + (A_4y - 15)^2 + z^2 = 140.6348$$

and add 2 additional unknowns that will be the lengths of leg 2 and 3. This allows one to end up with a system of 33 equations in 33 unknowns.

The precision poses are defined by the ψ, θ, ϕ angles with the values $\pi/10, \pi/6, \pi/10$ and $\pi/6, \pi/6, \pi/6$, $\pi/11.5, \pi/7.5, \pi/8.2$ Two solutions are found in about 38 hours of computation time on a cluster of 20 machines:

ρ_2^2	ρ_3^2	A_1	A_2	A_3	A_4
454.29	129.184	14.868	4.39259	-9.623	5.665
295.388	86.05	9.46	7.26	-11.87	8.635

As can be seen on these examples synthesis based on precision poses have the drawback that even for a small number of precision poses the synthesis equation are quite difficult to solve. For example for $m = 2, n = 3$ we will get 11 equations for each precisions poses and it seems quite difficult to manage more than 3 precision poses.

Another drawback of this approach is that if more than one precision pose is given we cannot ensure that all the precision poses lie *on the same branch*. This has to be verified afterward using the method described in the section devoted to path reconstruction. We are currently investigating another approach: a reference trajectory is given and we are looking for the geometry parameters that minimize the maximal error between the reference path and the followed one. Interval analysis may still be used in that case with the additional interest that it provides global optimization procedures.

CONCLUSION

Cam-coupled parallel robot are interesting alternate mechanisms to provide complex robot with constrained motion. Interval analysis has be shown to be an adequate tool to solve their inverse and direct kinematics. Although the computation time of the kinematics may be quite large we have seen that it drastically decreases as soon as information on the possible location of the platform is available, as this will be the case in a real-time application. We have also presented a procedure that allows one to compute very efficiently the possible paths of the platform.

We have then investigated a possible synthesis method based on precision poses. It was shown that it was possible to determine various set of the robot's geometrical parameters so that the precision poses will lie on the path of the platform. But the complexity of this approach is very high and it seems quite difficult to manage more than a small number of precision poses. Furthermore it is not possible to verify during the synthesis that the poses will lie on the same robot's path.

References

[1] Clavel R. DELTA, a fast robot with parallel geometry. In *18th Int. Symp. on Industrial Robot*, pages 91–100, Lausanne, April, 26-28, 1988.

[2] Company O., Marquet F., and Pierrot F. A new high speed 4-dof parallel robot. Synthesis and modeling issues. *IEEE Trans. on Robotics and Automation*, 19(3):411–420, June 2003.

[3] Démidovitch B. and Maron I. *Eléments de calcul numérique.* Mir, 1979.

[4] Hansen E. *Global optimization using interval analysis.* Marcel Dekker, 1992.

[5] Husty M.L. An algorithm for solving the direct kinematic of Stewart-Gough-type platforms. *Mechanism and Machine Theory*, 31(4):365–380, 1996.

[6] Jaulin L., Kieffer M., Didrit O., and Walter E. *Applied Interval Analysis.* Springer-Verlag, 2001.

[7] Lazard D. Stewart platform and Gröbner basis. In *ARK*, pages 136–142, Ferrare, September, 7-9, 1992.

[8] Merlet J-P. Solving the forward kinematics of a Gough-type parallel manipulator with interval analysis. *Int. J. of Robotics Research*, 23(3):221–236, 2004.

[9] Parenti-Castelli V. and Di Gregorio R. Parallel mechanisms applied to the human knee passive motion simulation. In *ARK*, pages 333–344, Piran, June, 25-29, 2000.

[10] Parenti-Castelli V. and others . On the modeling of passive motion of the human knee joint by means of equivalent planar and spatial parallel mechanisms. *Autonomous Robots*, 16:219–232, 2004.

[11] Rouillier F. Real roots counting for some robotics problems. In J-P. Merlet B. Ravani, editor, *Computational Kinematics*, pages 73–82. Kluwer, 1995.

[12] Wampler C.W., Morgan A.P., and Sommese A.J. Numerical continuation methods for solving polynomial systems arising in kinematics. *ASME J. of Mechanical Design*, 112:59–68, March, , 1990.