Optimal trajectory planning of a 5-axis machine tool based on a 6-axis parallel manipulator.

Jean-Pierre Merlet, M.-W. Perng, David Daney

Optimal trajectory planning of a 5-axis machine-tool based on a 6-axis parallel manipulator

J-P. Merlet	M-W Perng	D. Daney
INRIA	Tsing Hua University	INRIA
France	Taiwan	France

Abstract: Many new machine-tool are based on a hexapod structure, a parallel robot with 6 DoF, while only 5 of them are needed for the machining operations as the rotation of the end-effector around its normal is not used. We have therefore an extra DoF: we propose an algorithm which is able to determine for almost any trajectory of the machine the values of the extra DoF to ensure that the trajectory lie within the workspace of the robot, is singularity-free and furthermore optimize an arbitrary criterion

1 Introduction

In the last year new machine-tool structures based on parallel mechanical architecture have been proposed. Such structure have interesting features like high stiffness and velocities, accuracy etc.. which are appropriate for machining operations. However although many prototypes have been proposed, most of the designers have encountered difficulties in developing these new machines. They have focused up to now to the design of the basic mechanical components like universal and ball-and-socket joints, linear actuators. But two more aspects have to be considered before obtaining a working machine: determining the optimal geometry of the machine and developing an appropriate control. We address here this last point: a parallel structure has a highly non-linear behavior while classical machine are mostly linear. Hence control algorithms for this type of machine have to be developed. In this regard trajectory planning is an important areas, especially as most of the 5-axis new machine-tools are based on the principle of the Gough platform with extensible legs (figure 1). In this paper we will assume that each leg length should lie in some interval $[\rho_{min}, \rho_{max}]$). We represent a pose of the platform by the coordinates x_C, y_C, z_C of the origin of the mobile frame (supposed to lie on the rotation axis of the spindle) and its orientation by using the classical Euler angles ψ, θ, ϕ. This structure has 6 DoF, but only 5 of them are used in the machining process as the rotation of angle ϕ around the normal of the mobile platform is not used and may be fixed to any arbitrary value. Assume now that a particular time-dependent trajectory has been defined for the parameters $x_C, y_C, z_C, \psi, \theta$. The trajectory planning problem is to determine first what should be the value of the angle ϕ as a function of time, so that the full trajectory lie in the workspace of the robot, i.e. that

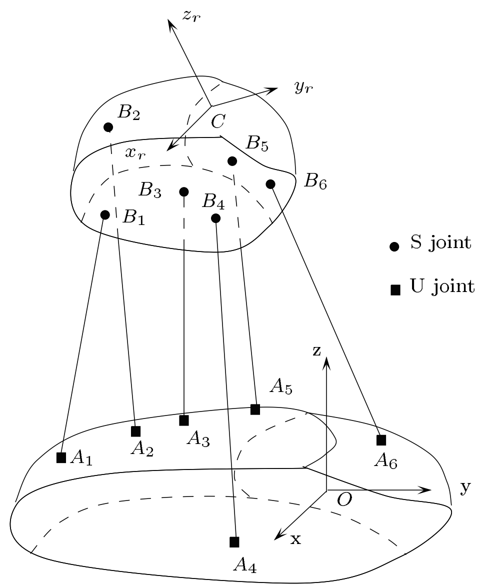

Figure 1: A Gough platform

for any pose on the trajectory the leg lengths lie within their limits. As we will see later on this constraint will not provide a unique solution for ϕ. Thus we will consider another constraint: the trajectory should be singularity-free. Still ϕ may not be uniquely defined by imposing these two constraints and thus we will optimize an additional constraint that we will specify later.

2 Optimizing ϕ at a pose

To illustrate the basic principle of our algorithm we will illustrate it on the simple example of determining the best ϕ for a given pose of the robot. We will call *partial pose* a set of values for the parameters $x_C, y_C, z_C, \psi, \theta$. A pose is therefore defined by a partial pose and a value for the angle ϕ.

2.1 Workspace constraint

For a Gough platform the length of a leg are simply the norm of the vector **AB** which may be written as

$$\mathbf{AB} = \mathbf{AO} + \mathbf{OC} + \mathbf{CB} \tag{1}$$

For a given pose and a given robot the two first elements of the right hand term of this equation are known. The last element, $\mathbf{CB}$ is equal to $R\mathbf{CB_r}$ where R is the rotation matrix, a function of ψ, θ, ϕ, and $\mathbf{CB_r}$ is defined by the coordinate of B in the mobile frame, which are known. Using equation (1) the leg length may be written as follows

$$\rho = ||\mathbf{AB}|| = U \cos\phi + V \sin\phi + W \tag{2}$$

where U, V, W are constants. Consider now the equation

$$\rho^2 - \rho^2_{min} = U \cos\phi + V \sin\phi + W - \rho^2_{min} = 0$$

It is well known that this equation may have up to 2 roots in ϕ. More precisely

- if $V^2 + U^2 - W^2 + 2W\rho^2_{min} - \rho^2_{min} < 0$ this equation has no root. If $Q = W - U - \rho^2_{min}$ is positive, then $\rho > \rho_{min}$ for any ϕ, while if Q is negative we have $\rho < \rho_m in$ for any ϕ
- if $V^2 + U^2 - W^2 + 2W\rho^2_{min} - \rho^2_{min} > 0$ this equation has two roots, $\phi_1, \phi_2 > \phi_1$. If $Q = W - U - \rho^2_{min}$ is positive, then $\rho > \rho_{min}$ for any ϕ in $[\phi_1, \phi_2]$, while if Q is negative we have $\rho > \rho_m in$ for any ϕ in $[0, \phi_1]$ or in $[\phi_2, 2\pi]$.

Hence for a given pose and a given leg we are able to determine first if it exists a ϕ such that the leg length is greater than ρ_{min} and if this is the case one or two ranges I^1_{min}, I^2_{min} for ϕ which define all the possible values of ϕ such that the leg length is greater or equal to ρ_{min}. A similar reasoning may be applied if we substitute ρ_{min} by ρ_{max} and will lead to zero, one or two possible range I^1_{max}, I^2_{max} for ϕ such that $\rho \leq \rho_{max}$ for any ϕ in the range. If the range I^1, I^2 exist for the minimal and maximal leg lengths, then their intersection defines the set S of ranges for ϕ for which $\rho_{min} \leq \rho \leq \rho_{max}$. Up to now we have considered only one leg of the robot. The procedure applied to all legs will lead to 6 sets of ranges $S^1, S^2, \ldots, S^6$ and the intersection $\mathcal{S}$ of these sets defines all the possible values of ϕ such that all the leg lengths lie within their limits. In the following we will assume that $\mathcal{S}$ is not empty.

2.2 Optimizing ϕ

Another constraint that we must consider is that the pose should not be singular. Singularity occurs when the determinant of the inverse jacobian of the robot J^{-1} is zero. The i-th row J^{-1}_i of this matrix is

$$J^{-1}_i = ((\frac{\mathbf{A_iB_i}}{||\mathbf{A_iB_i}||} \quad \mathbf{CB_i} \times \frac{\mathbf{A_iB_i}}{||\mathbf{A_iB_i}||}))$$

We will also use the *semi-inverse* jacobian matrix M defined by its i-th row M_i:

$$m_i = ((\mathbf{A_iB_i} \quad \mathbf{CB_i} \times \mathbf{A_iB_i}))$$

As $||\mathbf{A_i B_i}|| = \rho^i$ we notice that the determinant Δ of J^{-1} is related to the determinant $|M|$ of M by

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

Thus $|M| = 0$ will be equivalent to $\Delta = 0$. For a given partial pose each element of the row of M is a function only of the cosine and sine of angle ϕ. Hence the determinant δ of M will also be only a function of these sine and cosine

$$\delta = \sum r_{jk} \cos^j(\phi) \sin^k(\phi)$$

Two cases may occur for the coefficients r_{jk}. If they are all zero the determinant is always zero; this may occur either for particular geometry of the robot or for specific types of poses (for example if the platform and base are planar and lie in the same plane). In that case we cannot find a solution to the trajectory planning problem. If not all r_{jk} coefficients are zero, then the equation $\delta = 0$ has only a finite number of roots. In that case we may even impose that the dexterity of the manipulator as measures by $|\delta|$ should not be lower than a given threshold δ_{min}. A study similar to the one used for the workspace equations performed on the equations $\delta = \delta_{min}$ and $\delta = -\delta_{min}$ will lead to a set of possible range for the angle ϕ. We will see in a next section how to get these ranges.

2.3 Dealing with additional constraints

As we have seen in the previous section having a non singular pose in the workspace may still lead to multiple possible ranges for the angle ϕ. Thus we may still look for a value of ϕ that optimize another criterion. Although we will see later that our algorithm enable to use a large spectrum of criteria, we will focus on a specific one, which is to minimize the maximal absolute value of the articular forces τ for a given load $\mathcal{F}$ applied on the platform. This is a difficult criteria to deal with as τ is related to $\mathcal{F}$ by

$$\tau = J^T \mathcal{F}$$

and we have seen that the jacobian matrix J has a complex analytical formulation. Trying to determine analytically the exact value of ϕ such that the criteria is minimized is a difficult task. Furthermore if the criteria is changed we will have to perform another theoretical analysis. Therefore we have chosen another approach whose purpose is to be able to deal with almost any criteria that will have to be minimized and to determine a value of ϕ such that the value of the criteria is not greater than its optimal value by a fixed value ϵ, that we will call the *accuracy* of the optimal approach..

2.4 Practical implementation

The first problem we have to deal with is to determine which values of ϕ in the ranges $\mathcal{S}$ led to a non-singular pose. This implies that we have first to compute an analytical form of the coefficients r_{jk} as a function of both the parameters describing the geometry of the robot and the partial pose parameters in order to solve the equation:

$$\delta = \sum r_{jk} \cos^j(\phi) \sin^k(\phi) = 0 \tag{3}$$

Although M may seem to be a quite simple matrix its determinant has a quite large expression which is difficult to obtain, even using symbolic computation software [1]. As a consequence using the analytical form of the coefficient r_{jk} in the implementation and then plugging in the geometry and pose parameters may leads to error in the coefficients, due to rounding errors in the computation. In turn this will lead to bad estimation of the possible ranges for ϕ. We have chosen another approach which is based on a formal-numerical method: the purpose is to use exact computation for the determination of the coefficients and a numerical approach for the determination of the possible ranges for ϕ.

More precisely as soon as the geometry of the robot has been defined, for example by reading the coordinates of the A, B points in a file, the algorithm will run a MAPLE program to determine the value of the coefficients r_{jk}. Using the functionalities of MAPLE we are able to compute exactly this coefficients. Indeed we may use the `convert` instruction of MAPLE to convert exactly each of the coordinates into a rational number. In this way we will get the value of δ as

$$\delta = \sum R_{jk} x^a y^b z^c \cos^d \psi \sin^e \psi \cos^f \theta \sin^g \theta \cos^j(\phi) \sin^k(\phi)/W$$

where $a, b, c, d, e, f, R_{jk}, W$ are integers. The analytical expression of δ will be written by MAPLE in a file. We will see later how this file will be used to determining a singularity-free pose.

Consider now the articular force criteria. We will have to compute the articular forces for a given value of ϕ. If f_i denote the i-th component of the load $\mathcal{F}$ applied on the platform and M^T_{ij} the minor of M^T obtained by removing the i-th row and j-th column we may use the Cramer rule to get

$$\tau_j = \sum_{k=1}^{k=6} \frac{(-1)^{j+1}(-1)^{k+1} \rho_j f_k |M^T_{kj}|}{|M^T|} \tag{4}$$

To get the articular forces we thus need to determine the values of $|M^T_{kj}|$. These terms will be computed by the same MAPLE program that has been used for computing the determinant and their analytical expression will be written in files.

Outside formal computation we will use another mathematical tool which is *interval analysis* [2]. Basically interval arithmetics is similar to real arithmetics except that the numbers we are dealing with are intervals and that basic operators must be re-defined. For example the addition operator "+" on two intervals $X_1 = [\underline{x_1}, \overline{x_1}]$, $X_2 = [\underline{x_2}, \overline{x_2}]$ is defined as the interval

$$X_1 + X_2 = [\underline{x_1} + \underline{x_2}, \overline{x_1} + \overline{x_2}]$$

A nice property of interval arithmetics is that interval operator may be defined for almost any mathematical functions. Furthermore if we apply interval analysis on a function it enable to compute guaranteed lower and upper bounds for the function (that is called an *interval evaluation* of the function), including even rounding errors in the computation. For example imagine that we have to find the bounds of the function in which the number $1/3$ appears. Clearly there is no computer representation of the number $1/3$. In interval analysis this number will be represented on a Sun workstation by the interval:

$$[0.33333333333333332593184650, 0.33333333333333337034076750]$$

Thus the function evaluation will take into account the rounding errors. For the basic interval arithmetics operators we use the package BIAS/Profil and we will use the parser of the `ALIAS` library[1] that take as input a file with an analytical description of a function and the ranges for each variable appearing in the function, and returns the interval evaluation of the function. As an example of the use of the parser consider the analytical description of the determinant δ that has been obtained previously and assume that at some point of the algorithm we have determined that a given range on ϕ should be investigated as including a potential solution to our motion planning problem. We will use the parser to determine an interval evaluation $[\underline{\delta}, \overline{\delta}]$ of δ. If $\underline{\delta} \leq \delta_{min}$ and $\overline{\delta} \geq -\delta_{min}$, then no value of ϕ in the range will satisfy the singularity constraint. On the other hand if $\underline{\delta} \geq \delta_{min}$ or $\overline{\delta} \leq -\delta_{min}$, then any value of ϕ will satisfy the constraint on the workspace and on the singularity and hence we have only to care with the force criteria.

But the parser may also be used to evaluate the force criteria, which enable to describe the criteria in a file (called the *criteria file*) instead of having it rigidly fixed in the program. Indeed the parser may recognize the keywords `tau1`, `tau2`, ... `tau6` and substitute their interval values in the expression:

```
Max(abs(tau1),abs(tau2),abs(tau3),abs(tau4),abs(tau5),abs(tau6))
```

which describe our force criteria. Note that this is a very flexible way to describe an optimization criteria as a change in the criteria will merely amount to a change in the criteria file. For example minimizing the average of the absolute value of the articular forces will be described in the criteria file by:

```
(abs(tau1)+abs(tau2)+abs(tau3)+abs(tau4)+abs(tau5)+abs(tau6))/6
```

We may now describe our basic algorithm which purpose aim at determining a value of ϕ such that the pose is inside the workspace, the absolute value of $|M|$ is not lower than a threshold ϵ_M and the criteria $\mathcal{C}$ at this pose verify $\mathcal{C} > \mathcal{C}_O - \epsilon$ where $\mathcal{C}_O$ is the optimal value of the criteria.

[1] `ALIAS` is an interval-based package developed in the `SAGA` project which enable to analyze and solve system of equations

We will denote by $L_{\hat{\phi}}, U_{\hat{\phi}}$ the lower and upper bounds of the interval evaluation of the criteria obtained for the range $\hat{\phi}$ of the angle ϕ. Similarly $L^M_{\hat{\phi}}, U^M_{\hat{\phi}}$ will represent the lower and upper bound of the interval evaluation of the value of $|M|$. We will also use a *bisection process* for the range $\hat{\phi} = [\phi_1, \phi_2]$: the result of the bisection of this range is the 2 new ranges $[\phi_1, (\phi_1+\phi_2)/2], [(phi_1+\phi 2)/2, \phi_2]$. We will use a list $\mathcal{S}$ of ranges for ϕ with n ranges. This list will be initialized with the n ranges of $\mathcal{S}$ that guaranteed that the pose lie inside the workspace. We will select random value of ϕ in these ranges and compute the force criteria for this values of ϕ in order to initialize the minimal value of the criteria $\mathcal{C}^m$. During the algorithm we will consider the i-th element of the list $\mathcal{S}$ that will be denoted $\mathcal{S}_i$ and we start with $i = 1$. The algorithm proceeds along the following steps:

1. if $i > n$ return $\mathcal{C}^m$
2. compute $L^M_{\mathcal{S}_i}$, $U^M_{\mathcal{S}_i}$
3. if $L^M \leq 0 \leq U^M$ then
   - if $\mathrm{Max}(|L^M|, |U^M|) < \epsilon_M$, then $i = i + 1$ and go to step 1
   - otherwise bisect $\mathcal{S}_i$ and put the two obtained ranges at the end of the list $\mathcal{S}$. Apply $n = n + 2$, $i = i + 1$ and go to step 1
4. otherwise compute $L_{\mathcal{S}_i}$, $U_{\mathcal{S}_i}$
5. if $L_{\mathcal{S}_i} > \mathcal{C}^m - \epsilon$, then $i = i + 1$ and go to step 1
6. select random value for ϕ in $\mathcal{S}_i$, compute the value of the force criteria for these values and update $\mathcal{C}^m$ i.e. if the value of the criteria for one of these values is lower than the current $\mathcal{C}^m$, then this value become the new $\mathcal{C}^m$
7. bisect $\mathcal{S}_i$ and put the two obtained ranges at the end of the list $\mathcal{S}$. Apply $n = n + 2$, $i = i + 1$ and go to step 1

Basically the principle of this algorithm is to consider that the optimal value of ϕ should be in the range that ensure that the corresponding pose is inside the workspace and to bisect recursively these ranges, discarding the ranges for ϕ such that we are sure that the absolute value of $|M|$ is lower than ϵ_M or for which the criteria is certainly not lower than the current optimal value-ϵ. In the bisection process we select random values for ϕ in the ranges that are not discarded, compute the criteria for the corresponding pose and update the optimal value of the criteria if necessary. The bisection process stops when all the ranges in the list have been processed. The algorithm will return a value of ϕ such that the corresponding pose is inside the workspace, is not singular and such that criteria at this point will be not greater than the minimal value by ϵ.

3 Processing a trajectory

Assume now that the parameters of the partial pose are functions of the time T, supposed to lie in the range [0,1], i.e. that we have

$$x = D_x(T) \quad y = D_y(T) \quad z = D_z(T) \quad \psi = D_\psi(T) \quad \theta = D_\theta(T)$$

Similarly we may assume that the forces and torques applied on the end-effector may be defined by some time functions:

$$F_x = S_x(T) \;\; F_y = S_y(T) \;\; F_z = S_z(T) \;\; M_x = R_x(T) \;\; M_y = R_y(T) \;\; M_z = R_z(T)$$

Our aim with respect to the workspace, singularity and optimal criteria constraints remain the same as in the previous section, except that we have to provide some time function for ϕ. Furthermore we want to be able to deal with any type of trajectory, which prohibits the use of exact computation of the optimal values of ϕ which will minimize the criteria. The time function for ϕ will be obtained by computing values for ϕ at a regular time step Δt. Thus if $\Delta t = 0.01$ we will provide at least 101 values for ϕ corresponding to $T = 0, 0.01, 0.02, \ldots 1$. Between two time steps we assume that the value of ϕ used by the control remain the same.

To solve this problem a first approach will be to compute the partial pose at each time step and use the algorithm presented in the previous section to compute the optimal value for ϕ at this pose. But at the partial pose parameters will change between two time steps, this method does not guarantee that the trajectory that will be followed between these time steps will lie inside the workspace and will be singularity-free. Therefore we aim to compute at each time step a range for ϕ that satisfy the following constraints:

- between two successive time steps for any value of ϕ in the range the pose will be in the workspace
- between two successive time steps for any value of ϕ in the range the absolute value of $|M|$ is always greater than ϵ_M.
- for any value of ϕ in the range and for any time between two successive time steps the maximal value of the criteria trajectory should not be greater than the optimal value by ϵ (the optimal value is defined as the lowest possible value of the criteria between the two time steps).

The first step of the algorithm is similar to the one in the previous section: we define the time-varying trajectory of the partial pose parameters in MAPLE format and use MAPLE to compute the analytical expression of $|M^T_{kj}|, |M^T|$, and the external wrench (if it is position or time dependent) and ρ, the leg lengths. All these quantities are written in a file and are now function of ϕ and T. Using the parser we may obtain their interval evaluations for given ranges on T and ϕ.

We will then proceed to the computation of the optimal value of ϕ for the trajectory between two time steps i.e. the value of T is in a range $T_n, T_n+\Delta t$. We will use the same notation than in the previous section and introduce $w(\mathcal{S}_i)$, the width of the range $\mathcal{S}_i$. Consider now the interval evaluation of the determinant of $|M|$: we cannot proceed with the interval evaluation of the articular forces if this interval evaluation includes 0 as this term appears at the denominator. Remember now that this determinant is a function of two unknowns, ϕ and T. It may therefore occur that even for a fixed value of T the interval evaluation of $|M|$ may include 0. This means either that either there is a singularity on the trajectory or that the time step is too large which lead to a large over estimation of $|M|$. To fix this problem we will fix a threshold ϵ_S on $w(\mathcal{S}_i)$: as soon as a range on ϕ has a width lower than ϵ_S, while having still an interval evaluation of $|M|$ including 0 we will divide by two the time step δt and start again the process for the same time step.

There is another occurrence for which it may be necessary to divide the time step δt. Let consider the value of $L_{\mathcal{S}_i}$ and $U_{\mathcal{S}_i}$ and the current value of the criteria $\mathcal{C}^m$. If $L_{\mathcal{S}_i} > \mathcal{C} - \epsilon$ we discard $\mathcal{S}_i$ otherwise if $U_{\mathcal{S}_i} - L_{\mathcal{S}_i} < \epsilon$ we may take random values for ϕ inside $\mathcal{S}_i$ and for T in $[T_n, T_n + \delta t]$, then compute the criteria in this configuration and update the value of $\mathcal{C}^m$ if necessary, being sure that we will get the optimal value of the criteria up to the accuracy ϵ. But it may occur that even for a fixed value of ϕ we have $L_{\mathcal{S}_i} < \mathcal{C} - \epsilon$ and still $U_{\mathcal{S}_i} - L_{\mathcal{S}_i} > \epsilon$. In that case we cannot guarantee the accuracy on the optimal value of the criteria and we will have to divide the time step.

The lower and upper bound of the length of leg j for a given range on T and the range $\mathcal{S}_i$ on ϕ will be denoted $Inf_{\rho_j}(\mathcal{S}_i)$, $Sup_{\rho_j}(\mathcal{S}_i)$.

The algorithm proceeds along the following steps:

1. if $i > n$ return $\mathcal{C}^m$
2. compute $L^M_{\mathcal{S}_i}$, $U^M_{\mathcal{S}_i}$
3. if $L^M \leq 0 \leq U^M$ then
   - if Max$(|L^M|, |U^M|) < \epsilon_M$, then $i = i + 1$ and go to step 1
   - if $w(\mathcal{S}_i) < \epsilon_S$, then $\delta_t = \delta_t/2$ and start again
   - otherwise bisect $\mathcal{S}_i$ and put the two obtained ranges at the end of the list $\mathcal{S}$. Apply $n = n + 2$, $i = i + 1$ and go to step 1
4. otherwise compute $L_{\mathcal{S}_i}$, $U_{\mathcal{S}_i}$
5. if $L_{\mathcal{S}_i} > \mathcal{C} - \epsilon$, then $i = i + 1$ and go to step 1
6. select random values for ϕ in $\mathcal{S}_i$, compute the value of the criteria for these values and update $\mathcal{C}^m$ if necessary
7. if $U_{\mathcal{S}_i} - L_{\mathcal{S}_i} < \epsilon$, $i = i + 1$ and go to step 1

8. if $U_{\mathcal{S}_i} - L_{\mathcal{S}_i} > \epsilon$ and $w(\mathcal{S}_i) < \epsilon_S$, then $\delta_t = \delta_t/2$ and start again
9. bisect $\mathcal{S}_i$ and put the two obtained ranges at the end of the list $\mathcal{S}$. Apply $n = n + 2$, $i = i + 1$ and go to step 1

4 Examples and computation time

In this section we consider the robot described by the following coordinates of the A, B points:

A_1	-9	9	0
A_2	9	9	0
A_3	12	-3	0
A_4	3	-13	0
A_5	-3	-13	0
A_6	-12	-3	0
B_1	-3	7	0
B_2	3	7	0
B_3	7	-1	0
B_4	4	-6	0
B_5	-4	-6	0
B_6	-7	-1	0

4.1 Optimal ϕ at a partial pose

We must distinguish two different types of computation time: the computation time of the MAPLE session and the one for finding the optimal value of ϕ, the MAPLE session being completed. Indeed we may use two different approaches for the calculation: either run a MAPLE session for each partial pose for which we want to compute the optimal ϕ or generates once with MAPLE a generic form of $|M^T|, |M^T_{jk}|$ with a symbolic value for the pose parameters, which enable to determine the optimal ϕ for any particular partial pose by just plugging in the numerical values of the partial pose parameters. If a large number of partial pose have to be tested the later approach is the most convenient.

We have considered the partial pose defined by $x = y = 0$, $z = 56$, $\psi = \theta = 0$ and a load defined by $F_x = 100, F_z = 900$. On a laptop PC the computation time of the MAPLE session for a given partial pose is about 3s and the computation time of the optimal ϕ according to the desired accuracy on the criteria is:

Accuracy	50	10	0.1	0.001
Computation time(ms)	160	170	220	270

for an optimal value of the criteria which is 836.49 for $\phi = 13.22°$. For $\phi = 0$ the criteria is 908 and 1131 for $\phi = 30$ degrees, both poses being in the workspace

ϕ (degree)	$Min(\tau_f - \tau_o)$	$Max(\tau_f - \tau_o)$	A_0	A_f	$G(\%)$
-10	99.6	189.3	949	1078.6	15.95
0	-11.2	82.5	949	960.6	4.43
10	15.9	113	949	983.7	6.13

Table 1: Result of the optimal approach with an accuracy ϵ of 50, with τ_f the criteria for a fixed value of ϕ, τ_o the criteria obtained for the optimal approach, A_o the average value of the criteria for the optimal approach, A_f the average value of the criteria for the fixed value of ϕ and G the average value of $\tau_f - \tau_o$ divided by A_o

of the robot. The optimal ϕ leads therefore to a significant decrease of the maximal force applied on the actuators.

4.2 Optimal ϕ on a trajectory

In this example we want to optimize ϕ for a circular trajectory in the plane $z = 56$, the center of the circle being (0,0) an its radius 3, with $\psi = 0$ and $\theta = 5°$. Using the notation ψ=`p`, θ=`t`, the trajectory file in MAPLE format is:

```
p:=0:
t:=5*Pi/180:
x:=3*sin(2*Pi*T):
y:=3*cos(2*Pi*T):
z:=56:
```

and the forces applied on the platform are supposed to be constant with $F_x = 100, F_z = 900$, the other components being zero.

We have investigated first find the time laws for ϕ using an accuracy ϵ of 50 and have computed the maximum of the absolute value of the articular forces for three fixed values of ϕ: $-10°, 0°, 10°$. In that case the computation time is 1h7mn47s and Table 1 presents the minimal and maximal values of the difference between τ_f, the criteria computed for a fixed value of ϕ and the criteria τ_o, obtained using the optimal approach (hence a negative value of $\tau_f - \tau_o$ indicates that the optimal approach has lead to larger force, which may occur as we compute the optimal value up to the accuracy). The table gives also the average value A_o of the criteria for the optimal approach, the average value A_f for the fixed value of ϕ and the average gain G obtained by using the optimal approach (obtained as the average of $\tau_f - \tau_o$ divided by A_o). Figure 2 presents a plot of the criteria as a function of time for the various possible fixed values of ϕ and for the optimal approach, while figure 4 presents the values for ϕ obtained by the optimal approach as function of time. Table 2 presents the

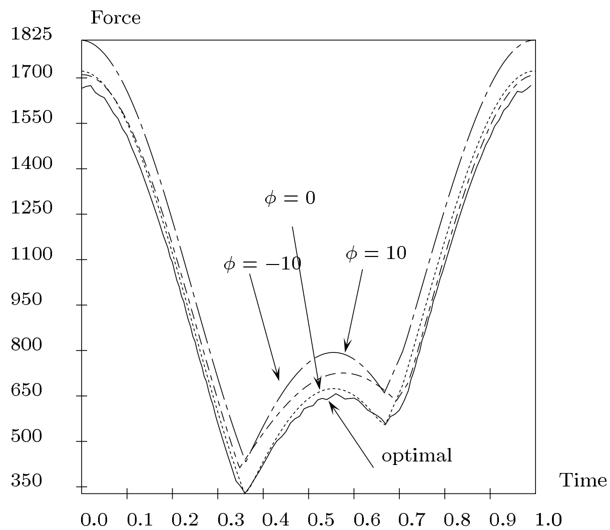

Figure 2: Values of the criteria for fixed values of ϕ and for the optimal approach with an accuracy $\epsilon = 50$.

same result when the accuracy is reduced to 10. In that case the computation time is .5h53mn59s. Figure 3 presents a plot of the criteria as a function of time for various possible fixed values of ϕ and for the optimal approach, while figure 4 presents the values for ϕ obtained by the optimal approach as function of time.

4.3 Optimal ϕ on a trajectory

In this example we consider the manufacturing of a vertical circular cone having an opening angle of 10 degrees. The motion of the platform is hence a circle (here with radius 3) while the orientation of the platform is such that the tool is always perpendicular to the cone. The trajectory of the end-effector is therefore given by:

```
p:=2*Pi*T:
t:=5*Pi/180:
x:=3*sin(2*Pi*T):
y:=-3*cos(2*Pi*T):
z:=56:
```

Note here that for this trajectory we cannot use a fixed value for ϕ as the definition of ψ will lead to a 360 degrees rotation of the platform around the

ϕ (degree)	$Min(\tau_f - \tau_o)$	$Max(\tau_f - \tau_o)$	A_o	A_f	$G(\%)$
-10	107	190.5	940.5	1078.6	16.67
0	-2.5	84	940.5	960.6	5.15
10	23.7	118.5	940.5	983.7	6.42

Table 2: Result of the optimal approach with an accuracy ϵ of 10, with τ_f the criteria for a fixed value of ϕ, τ_o the criteria obtained for the optimal approach, A_o the average value of the criteria for the optimal approach, A_f the average value of the criteria for the fixed value of ϕ and G the average value of $\tau_f - \tau_o$ divided by A_o

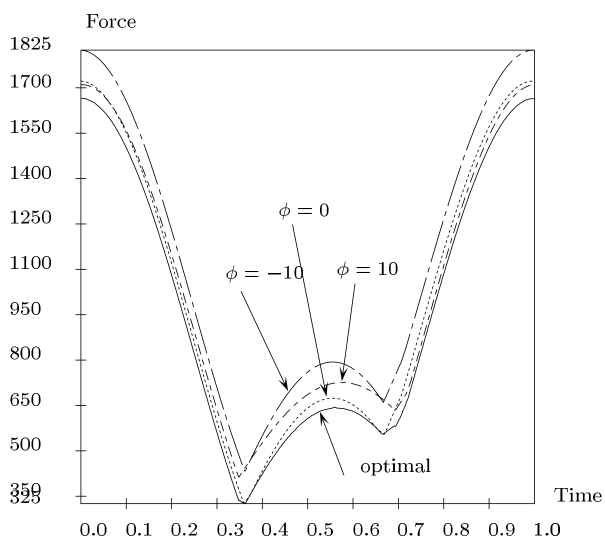

Figure 3: Values of the criteria for fixed values of ϕ and for the optimal approach with an accuracy $\epsilon = 10$.

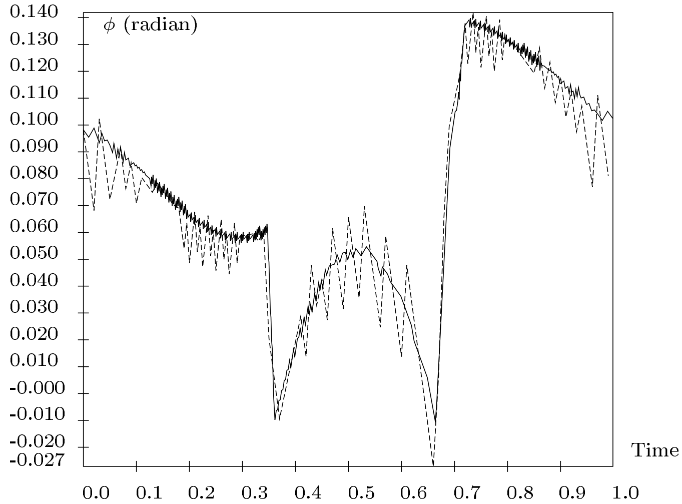

Figure 4: Values of ϕ obtained by the optimal approach for an accuracy $\epsilon = 10, 50$.

vertical axis. To avoid this phenomena a reasonable choice will be to impose $\phi = -\psi$ and we will compare the values of the criteria obtained for this trajectory with the one we will get using our optimization approach.

As for the forces and torques acting on the platform, they are no more constant.If we assume that the force created by weight of the platform and the tool is 900N, while the force exerted by the cone on the tool has an amplitude of 100N and is always perpendicular to the cone (we neglect the tangential force created by the tool), we get the following forces and torques as function of the time T and the angles ψ (denoted `p`) and θ (denoted `t`):

```
Fx:=100*sin(2*Pi*T)*cos(t):
Fy:=-100*cos(2*Pi*T)*cos(t):
Fz:=-900-100*sin(t):
Mx:=-5*cos(p)*sin(t)*(-900-100*sin(t))+500*cos(t)^2*cos(2*Pi*T):
My:=500*cos(t)^2*sin(2*Pi*T)-5*sin(p)*sin(t)*(-900-100*sin(t)):
```

Using these data we have run our optimization algorithm using an accuracy $\epsilon = 50$: figure 5 presents a plot of the criteria as a function of time for $\phi = -\psi$ and for the optimal approach, while figure 6 presents the values for ϕ obtained by the optimal approach as function of time. Table 3 presents the minimal and maximal values of the difference between τ_f, the criteria computed for $\phi = -\psi$ and the criteria τ_o, obtained using the optimal approach (hence a negative value of $\tau_f - \tau_o$ indicates that the optimal approach has lead to larger force, which

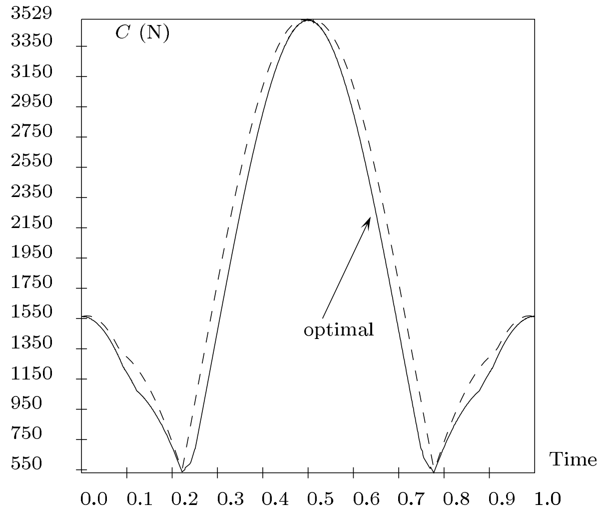

Figure 5: Values of the criteria for $\phi = -\psi$ and for the optimal approach with an accuracy $\epsilon = 50$.

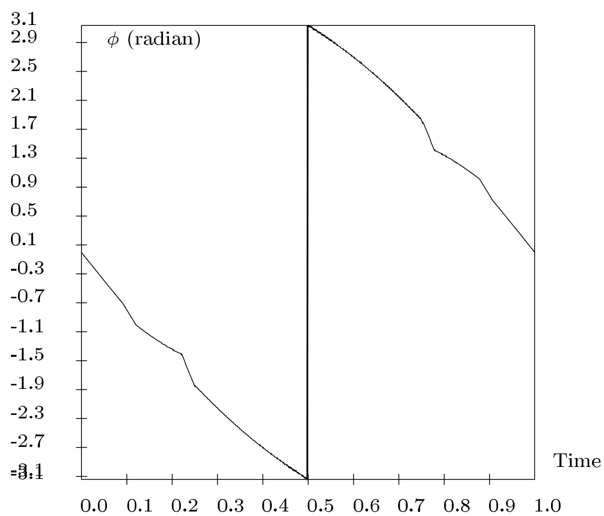

Figure 6: Values of ϕ obtained by the optimal approach for an accuracy $\epsilon = 50$.

$Min(\tau_f - \tau_o)$	$Max(\tau_f - \tau_o)$	A_o	A_f	$G(\%)$
-5.1	324.6	1763.8	1905.5	7.72

Table 3: Result of the optimal approach with an accuracy ϵ of 50, with τ_f the criteria for $\phi = -\psi$, τ_o the criteria obtained for the optimal approach, A_o the average value of the criteria for the optimal approach, A_f the average value of the criteria for the fixed value of ϕ and G the average value of $\tau_f - \tau_o$ divided by A_o

may occur as we compute the optimal value up to the accuracy). The table gives also the average value A_o of the criteria for the optimal approach, the average value A_f for $\phi = -\psi$ and the average gain G obtained by using the optimal approach (obtained as the average of $\tau_f - \tau_o$ divided by A_o). In that case however the computation time is very large (more than 5 days on a SUN Ultra 10 workstation). But this computation time may be largely reduced by using an implementation based on an array of computers as the structure of the algorithm is highly favorable for such an implementation.

5 Conclusion

We have presented an algorithm that is able to deal in a very flexible way with the problem of the trajectory planning of 5-axis machine tool based on a 6-DoF parallel manipulator. For a trajectory we are able to determine the values, if they exist, of the extra degree of freedom which ensure that the trajectory is inside the workspace, is singularity-free and at the same time optimize an additional criteria. Although we are able to compute the optimal values of the extra degree of freedom up to a fixed accuracy we have shown that significant gain may be obtained for the criteria. In some cases however computation time may be quite large but the gain in reduction of the wear on the machine may justify the use of the optimal approach.

References

[1] 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.

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