Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Mechanism and Machine Theory xxx (2004) xxx–xxx
Mechanism and Machine Theory
www.elsevier.com/locate/mechmt

Multi-criteria optimal design of parallel manipulators based on interval analysis

F. Hao, J.-P. Merlet *

INRIA Sophia-Antipolis, BP 93, Sophia-Antipolis Cedex 06902, France

Received 25 July 2003; received in revised form 15 June 2004; accepted 6 July 2004

Abstract

In optimal design problems we have to determine a set of design parameters such that a given mechanism satisfies a list of requirements. In practice however these requirements may be classified as either *compulsory* or *relaxable*. For classical optimal design methodologies, it is very difficult to find the solutions that satisfy compulsory requirements simultaneously and make the best compromise between these two kinds of requirements. So in this paper we propose and illustrate on an example of parallel robots an approach based on interval analysis that allows to determine almost *all* possible mechanism geometries such that all compulsory requirements will be satisfied simultaneously. As using interval analysis all possible solutions will be obtained as a set of regions in the parameter spaces, the best design compromise for the relaxable requirements will be determined by sampling the solution regions.

Keywords: Parallel manipulators; Interval analysis; Optimization

1. Introduction

Compared with serial manipulators, parallel manipulators have many advantages such as higher rigidity, better positioning accuracy, high speed and high load capacity. But one of their draw-

* Corresponding author.
E-mail addresses: fang.hao@sophia.inria.fr (F. Hao), jean-pierre.merlet@sophia.inria.fr (J.-P. Merlet).

0094-114X/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.mechmachtheory.2004.07.002

backs is that their performances depend heavily on their geometry. So optimal design has become a key issue for their development and many researchers have recently paid attention to this problem [1–6].

A classical way to solve optimal design problems is to define a real-valued function C as a weighted sum of requirement indices I_i, which are functions of the design parameter set $\mathscr{P}$.

$$C = \sum_i w_i I_i(\mathscr{P}) \tag{1}$$

where w_i are weights. Numerical procedures are then used to find a set $\mathscr{P}_m$ which minimizes C, usually starting with an initial guess $\mathscr{P}_0 \cdot \mathscr{P}_m$ will be considered as the optimal design solution.

But this method has many drawbacks. First it is assumed that the requirement indices can be defined and that they can be calculated efficiently as the numerical optimization procedure requires a large number of evaluation of these indices. But these assumptions are difficult to satisfy in practice especially for parallel robots: for example what could be the definition of an index indicating that a cube with a given volume must be included in the workspace of the mechanism? As for the performance evaluation it is now well known that *exact* performance evaluation is very difficult to obtain because of complexity reasons or numerical round-off errors in the calculation.

A second problem of cost-function approaches is that the numerical optimization procedure may converge toward a local minimum, which leads to a solution that may be quite far away from the optimal design solution.

In cost-function approaches it is difficult to determine the weights, because the weights not only indicate the priority of the requirements but also tackle with the unit problems of the performance indices. For example for a 3-DOF translational robot if the used performance indices are the workspace volume and positioning accuracy, then for a fair comparison between both criteria a weight with ratio 10^3 must be chosen. Furthermore a small change in the weights may lead to very different optimal design solutions. But until now there are not intuitive rules for determining their values.

The cost-function approaches may also lead to inconclusive results. It was exemplified by Stoughton [7] who was going to design a special kind of Gough platforms with improved dexterity and a reasonable workspace volume. He found that these criteria were varying in opposite directions: the dexterity was decreasing when the workspace volume was increasing. Hence the problem of optimal design has become the problem of determining an acceptable compromise between the two requirements. In most literature authors have been aware of this problem and consider only one performance criterion (see for example [8–18]). But for practical applications it is quite seldom that there is only one criterion to be considered (see for example [19–21]).

Another major drawback of cost-function approaches is that it provides only one optimal design solution. In our opinion an optimal design methodology should provide a set of solutions for the following reasons:

- A designer may not have all the necessary information to make the final design choice (for example he may obtain an arbitrary length for a linear actuator while the end-user will finally decide to use a commercially available actuator with only finite possibilities for the lengths).

- In general there is not a unique solution to a design problem as compromises have to be made. Providing various solutions allows the end user to choose the best design compromise for the problem at hand.

Finally introducing manufacturing errors in the cost-function is difficult. This function may be very sensitive to design errors, so the final instance of the real mechanism has performances that are quite different from the theoretical optimal design.

In this paper we will consider a difficult problem of optimal design of a 6-DOF parallel manipulator with multi-criteria requirements. The contribution of this paper is summarized as follows:

- A new optimal design methodology based on interval analysis is proposed which allows to determine almost all the possible geometries satisfying two compulsory requirements (on the workspace and accuracy).
- Two alternative approaches are introduced to determine the best design compromise for the relaxable requirements.
- A simplified algorithm which is reasonable and acceptable in practice is proposed to speed-up computation.

2. Parallel manipulator kinematics

The mechanical architecture of the considered robot is presented in Fig. 1. It is known as "active wrist" and has been patented by INRIA in 1991 [22]. A mobile platform is connected to six fixed length legs through ball-and-socket joints. An universal joint is located at the other extremity of the leg and the location of the joint center can be changed via the motion of a linear actuator connected to the base (in the considered design the motion axis of the joint is vertical to the base,

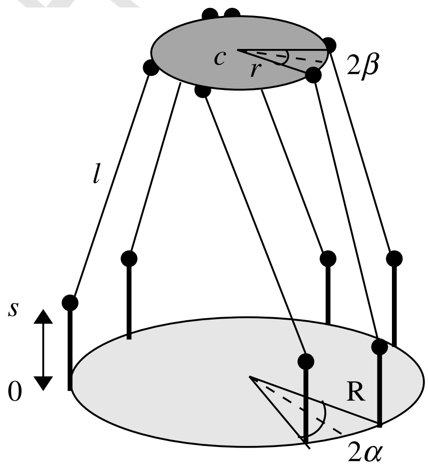

Fig. 1. The parallel manipulator with fixed actuators.

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD = 5.0.1

but this direction may be arbitrary). Controlling the locations of the U-joints allows to control the pose of the mobile platform.

2.1. Design parameters and workspace definition

For simplicity, it is assumed that the attachment points of all the joint centers lie on a circle on the base and the mobile platform. The joint centers are symmetrical with respect to three lines located 120° apart. So the geometry of such parallel manipulators is defined by six design parameters:

- R, r: the radii of the circles on which lie the attachment points of the legs on the base and platform;
- α, β: the half angles between two adjacent attachment points on the base and platform;
- s: the stroke of the actuator;
- l: the length of the leg (supposed to be identical for all legs).

The desired workspace of the robot is denoted as $\mathscr{W}$ and is decomposed into a desired *orientation workspace* and a *translation workspace*. The desired orientation workspace is defined by three ranges of the pitch, roll, yaw angles ϕ_1, ϕ_2, ϕ_3. Similarly the translation workspace is defined by three ranges of the x, y, z coordinates of the reference point C on the platform.

For identical heights of the actuated joints the x, y coordinates will be 0 while the z coordinates will depend on the design parameters. Hence the desired range of the z coordinates is defined as a relative motion z_r around the *nominal height* z_n, which is the z coordinates of the reference point C when all the actuators are at their mid stroke ρ_m. We have:

$$z_n = \rho_m + \sqrt{l^2 - R^2 - r^2 + 2Rr\cos(\gamma)} \quad (2)$$

where

$$\rho_m = \frac{s}{2} \quad (3)$$

$$\gamma = \frac{\pi}{3} - \alpha - \beta \quad (4)$$

so the z coordinates in the reference frame is

$$z = z_n + z_r \quad (5)$$

2.2. Robot kinematics

In this paper it is assumed that the workspace of the parallel manipulator is limited only by the motion ranges of the actuators. Note however that other constraints (such as limited motion of the passive joints) can be considered easily.

From Fig. 2, we have

$$B_i - A_i = [\,d_x \quad d_y \quad d_z - \rho_i\,] \quad (6)$$

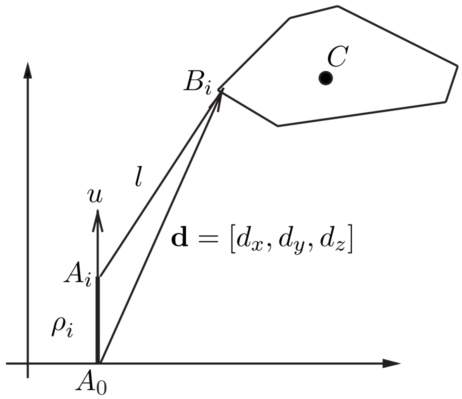

Fig. 2. The leg of the parallel manipulator.

where B_i, A_i are the extremity coordinates of the legs. d_x, d_y, d_z are known for a given pose of the platform and ρ_i is the length of the actuated link. As the norm of $B_i - A_i$ must be equal to l, we get

$$\rho = d_z \pm \sqrt{l^2 - d_x^2 - d_y^2} \quad (i = 1 \ldots 6) \tag{7}$$

Between the two possible solutions we select

$$\rho = d_z - \sqrt{l^2 - d_x^2 - d_y^2} \quad (i = 1 \ldots 6) \tag{8}$$

For a given pose if

$$\rho_{\min} = 0 \leqslant \rho_i \leqslant s = \rho_{\max} \tag{9}$$

is verified, then this pose can be reached by the manipulator.

2.3. Error analysis and singularity

The positioning accuracy of the platform is influenced by a set of parameter errors $\Delta\Theta$ such as measurement errors of the actuated joints, location errors of the attachment points, etc. As these errors are usually small a linear error model is used:

$$\Delta q = J(p, q)\Delta\Theta \tag{10}$$

where q are the poses of the platform and p are the geometry parameters. For a full error model $\Delta\Theta$ will be a large vector, but it has been recognized that the measurement errors of the actuator motions induce the largest errors on the positioning of the platform [23]. Hence the influence of the other parameters is neglected, so

$$\Delta q = J(p, q)\Delta\rho \tag{11}$$

where $J(p, q)$ is 6×6 Jacobian matrix whose inverse J^{-1} is defined by

$$\left[\frac{\mathbf{A_iB_i}}{\mathbf{u} \cdot \mathbf{A_iB_i}} \; \frac{\mathbf{A_iB_i} \times \mathbf{B_iC}}{\mathbf{u} \cdot \mathbf{A_iB_i}} \right] \tag{12}$$

The inverse Jacobian matrix is important for singularity analysis. But in this paper we will not consider the singularity problem because we have already designed an algorithm that allows to check whether there is singularity in a given workspace for a family of robots whose geometry parameters are denned by a set of ranges [24].

It is well known that determining a closed-form of the Jacobian matrix $J(p, q)$ is very difficult. Hence a difficult problem of error analysis is how to express the positioning errors analytically as a function of the sensor errors.

3. Interval analysis

3.1. Interval arithmetics

Interval arithmetics is a simple method that can provide lower and upper bounds for a function with interval unknowns. One of its important advantages is that it allows computer round-off errors to be taken into account. The *interval evaluation* of a function determines an interval that guarantees the inclusion of the exact lower and upper bounds of this function. The simplest interval evaluation method is the *natural evaluation* in which each mathematical operator $\diamondsuit$ of the function is replaced by an interval equivalent $\diamondsuit I$ returning an interval $[\underline{\diamondsuit I}, \overline{\diamondsuit I}]$ such that for all x: in a range $X, \underline{\diamondsuit I(X)} \leqslant \diamondsuit(x) \leqslant \overline{\diamondsuit I(X)}$.

Consider for example the function

$$f(x) = x^2 + x \tag{13}$$

for x in $[-1, 1]$. The interval equivalent of the square function is denned by

$$[a,b]^2 = \begin{cases} [0, \mathrm{Max}(a^2, b^2)] & \text{if } 0 \in [a,b] \\ [\mathrm{Min}(a^2, b^2), \mathrm{Max}(a^2, b^2)] & \text{otherwise} \end{cases} \tag{14}$$

Hence

$$f([-1,1]) = [0,1] + [-1,1] = [-1,2] \tag{15}$$

Note that the interval evaluation of a function depends heavily on its analytical form. For example Eq. (13) is rewritten as

$$f(x) = \left(x + \frac{1}{2}\right)^2 - \frac{1}{4} \tag{16}$$

Using this form, we have

$$f([-1,1]) = \left[-\frac{1}{2}, \frac{3}{2}\right]^2 - \frac{1}{4} = \left[-\frac{1}{4}, 2\right] \tag{17}$$

3.2. Notations for interval analysis

The lower and upper bounds of an interval X will be denoted by $\underline{X}, \overline{X}$ and the width of this interval is $w(X) = \overline{X} - \underline{X}$. The midpoint of an interval X is defined as

$$\text{mid}(X) = \frac{\overline{X} + \underline{X}}{2} \tag{18}$$

An n-dimensional interval set is called a *Box*:

$$X = \{[\underline{X_1}, \overline{X_1}], \ldots, [\underline{X_n}, \overline{X_n}]\} \tag{19}$$

The width w of an n-dimensional interval set X is the maximal width of its interval components.

Bisection is one of the most basic operation of interval analysis. For an n-dimensional interval set X the results of a bisection along the variable x_i are two new interval sets $L(X)$, $R(X)$ defined by

$$L(X) \triangleq \{[\underline{x}_1, \bar{x}_1], \ldots, [\underline{x}_i, (\underline{x}_i + \bar{x}_i)/2], \ldots, [\underline{x}_n, \bar{x}_n]\} \tag{20}$$

$$R(X) \triangleq \{[\underline{x}_1, \bar{x}_1], \ldots, [(\underline{x}_i + \bar{x}_i)/2, \bar{x}_i], \ldots, [\underline{x}_n, \bar{x}_n]\} \tag{21}$$

4. Optimal design for the workspace requirement

In order to use interval analysis we must be able to define an initial range for every design parameter (which is a reasonable assumption in most cases). So the *allowed parameter box* (APB) is defined as an n-dimensional box that contains all the allowable values of the design parameters.

The *feasible parameter boxes* (FPBs) are defined as boxes such that any point belonging to a FPB defines a geometry of the mechanism that satisfies one of the compulsory requirements. In our approach FPBs will be determined by using interval analysis and their union will be an approximation of the region that represents all the mechanisms satisfying one of the compulsory requirements.

The *valid parameter boxes* (VPBs) are the intersections of all FPBs of different compulsory requirements. Points in VPBs define mechanism geometries that satisfy all compulsory requirements simultaneously.

4.1. Determination of the allowed parameter box

In most cases it is possible to obtain initial bounds for the design parameters:

- 0 is an evident lower bound for r, R, but consideration on the interference between the passive joints will lead to a better lower bound. The overall size of the manipulator provides an upper bound. Note that for symmetry reasons an additional constraint is $r \leqslant R$.
- α, β have at least 0 as the lower bound but consideration on the interference between the passive joints will also lead to a better lower bound. The upper bound which is $\alpha, \beta \leqslant \frac{\pi}{3}$ is obtained by considering the symmetry of the attachment point locations.

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD = 5.0.1

- l has a lower bound which is $R - r$, The upper bound is obtained by considering the maximal, size of the robot (note that the accuracy requirement will enable to eliminate too large values of l quickly as the dexterity will be poor in that case).
- s has a lower bound which is the required maximum travel in z direction, the upper bound is determined either by constraints on the overall size or by constraints on commercially available components that will be used in the mechanism.

The desired workspace $\mathscr{W}$ will be provided by the user. Without lack of generality we assume that $\mathscr{W}$ is denned by specified ranges for each pose parameter (but other workspace shape can be used as well).

4.2. Algorithm principle

In order to satisfy the workspace requirement it is necessary to verify that the length of the actuated link ρ_i satisfies

$$\rho_{\min} = 0 \leqslant \rho_i \leqslant \underline{s} = \rho_{\max} \tag{22}$$

for any pose in the workspace. A first algorithm $F_w(P, Q)$ based on interval analysis will take as inputs a design parameter box P and a pose parameter box Q included in the desired workspace $\mathscr{W}$. In $F_w(P, Q)$ we compute the interval evaluation $[\underline{\rho_i}, \overline{\rho_i}]$ for the six actuated links and $F_w(P, Q)$ will return:

- -1 if $\underline{\rho} > \rho_{\max}$ or $\overline{\rho} < \rho_{\min}$ for at least one leg and at least one pose in Q;
- 0 if $\underline{\rho} < \rho_{\min}$ or $\overline{\rho} > \rho_{\max}$ for at least one leg and one pose in Q;
- 1 if $\underline{\rho} \geqslant \rho_{\min}$ and $\overline{\rho} \leqslant \rho_{\max}$ for all legs and all poses in Q.

If $F_w(P, Q)$ returns 1, then any robot whose design parameters lie within P can reach all the poses included in Q. If $F_w(P, Q)$ returns -1, then the design parameters included in P define the parallel manipulators whose geometries do not allow to reach some poses in the desired workspace $\mathscr{W}$. If $F_w(P, Q)$ returns 0 then we cannot decide whether the design parameters in P (or in some parts of P) define the right robot geometries or not, as the overestimation of interval arithmetics may be the reason why $\underline{\rho} < \rho_{\min}$ or $\overline{\rho} > \rho_{\max}$.

Our main algorithm will determine a set of boxes P_i such that $F_w(P_i, Q_j)$ will return 1 for all elements Q_j of a set whose union is the desired workspace $\mathscr{W}$. During the calculation boxes P_i with width lower than a given threshold ϵ will be called *neglected boxes* and will not be considered (although they may be stored in some particular structures): the main motivation for neglecting boxes is to take into account manufacturing errors by choosing ϵ equal to twice the manufacturing errors. Indeed because of manufacturing tolerances, only the FPB whose width is at least twice the manufacturing errors can guarantee that the geometry parameters of the real mechanism lie in the FPB when choosing the center of the FPB as the manufacturing parameters.

The algorithm that determines the FPBs for the workspace requirement is based on $F_w(Q, P)$ and is similar to the algorithm presented in [25]. It uses a list $\mathscr{L} = \{P_i\}$ of n potential FPBs initialized with the APBs, a list $\mathscr{S} = \{Q_j\}$ of m boxes of the pose parameters and thresholds ϵ,$\varpi$of

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD = 5.0.1

the width w of the boxes P_i, Q_j. During each bisection of P_i two elements are added to $\mathscr{L}$ and are placed at the end of the list. Similarly each bisection of Q_j adds two elements to $\mathscr{S}$. The algorithm proceeds as follows:

1. loop 1
   (a) if $i > n$, then EXIT
   (b) if $F_w(P_i, \mathscr{W}) = -1$, then $i = i + 1$, go to 1(a)
   (c) if $F_w(P_i, \mathscr{W}) = 1$, then store P_i as a FPB, $i = i + 1$, go to 1(a)
   (d) if $w(P_i) < \epsilon$, then store P_i as a neglected box, $i = i + 1$, go to 1(a). Otherwise go to loop 2
2. loop 2, with $\mathscr{S} = \{Q_1 = \mathscr{W}\}, j = m = 1$
   (a) if $j > m$, then store P_i as a FPB, $i = i + 1$, go to 1(a)
   (b) if $w(Q_j) < \varpi$, then bisect P_i, $n = n + 2$, $i = i + 1$ and go to 1(a)
   (c) if $F_w(P_i, Q_j) = -1$, then P_i cannot be a FPB, $i = i + 1$, go to 1(a)
   (d) if $F_w(P_i, Q_j) = 1$, then $j = j + 1$, go to 2(a)
   (e) bisect Q_j, $j = j + 1$, $m = m + 2$, go to 2(a)
   (f) end of loop 2
3. end of loop 1

The above algorithm guarantees that all the FPBs of the desired workspace with width larger than ϵ can be determined. Note that this algorithm is an incremental algorithm. Indeed we usually start the calculation with a large value of ϵ and then refine the calculation with a lower value of ϵ only using the boxes that have been neglected during the previous run as the initial data of the list $\mathscr{L}$, thereby reducing a large number of calculation.

In this algorithm other constraints on the workspaces can also be taken into account, for example motion ranges of passive joints. Indeed such constraints can be defined by an inequality constraint $\mathscr{G}(P, Q) \leqslant 0$. In that case $F_w(P, Q)$ will return:

- 1 if $\underline{\rho} \geqslant \rho_{\min}$ *and* $\overline{\rho} \leqslant \rho_{\max}$ *and* $\mathscr{G}(P, Q) \leqslant 0$ for all legs and all poses in Q;
- -1 if $\underline{\rho} > \rho_{\max}$ *or* $\overline{\rho} < \rho_{\min}$ *or* $\mathscr{G}(P, Q) > 0$ for at least one leg and at least one pose in Q;
- 0 otherwise.

Using the same principle $F_w(P, Q)$ can be extended to deal with arbitrary specification of the desired workspace $\mathscr{W}$ as soon as a test $\mathscr{T}(Q)$ has been defined that returns 1 if the set of pose parameters Q belong to $\mathscr{W}$, otherwise it returns 0. In that case $F_w(P, Q)$ will return 1 or -1 only when $\mathscr{T}(Q) = 1$.

5. Optimal design for the accuracy requirement

A classical requirement for accuracy is that the positioning errors of the platform should be less than the fixed threshold ΔX, being given the range $\Delta\rho^m = [\underline{\Delta\rho^m}, \overline{\Delta\rho^m}]$ of the measurement errors on the locations of the actuated joints.

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD = 5.0.1

Eq. (11) could be used but as mentioned before the evaluation of Jacobian matrix $J(x)$ is very difficult especially for 6-DOF parallel robots. On the other hand, the inverse Jacobian matrix $J^{-1}(x)$ can be calculated easily in a closed form. So from Eq. (11), we have

$$\Delta\rho = J^{-1}(P, Q)\Delta Q \tag{23}$$

Assume that P, Q are defined as ranges and T_{ij} is denoted as the absolute value of the interval evaluation of the element of $J^{-1}(P, Q)$ at the ith row and jth column. Then an interval U_i is defined as

$$U_i = \sum_{k=1}^{k=6} T_{ik}\overline{\Delta X_k} \tag{24}$$

Clearly $\overline{U_i}$ is an upper bound of the maximal allowable value for $\Delta\rho_i$ such that the positioning errors of the platform do not exceed ΔX. Similarly $\underline{U_i}$ is a lower bound of this value.

Similar to $F_w(P, Q)$, an algorithm $F_a(P, Q)$ is designed that takes as inputs a design parameter box P and a pose parameter box Q included in the desired workspace $\mathscr{W}$ and will return:

- -1 if there exists i such that $\overline{U_i} < \Delta\rho^m$. In that case for any robot geometry included in P and for any pose within Q, the worst allowable accuracy necessary to obtain the required positioning errors ΔX is lower than $\Delta\rho^m$.
- if $\underline{U_i} \leqslant \overline{\Delta\rho^m}$ for all i. In that case for any parameters included in P and Q the best allowable accuracy necessary to obtain the required positioning errors ΔX is greater than $\Delta\rho^m$. Consequently the positioning errors at any pose in Q induced by sensor errors (bounded by $\Delta\rho^m$) will always be lower than ΔX.
- 0 otherwise.

Using $F_a(P, Q)$ instead of $F_w(P, Q)$ in the algorithm presented in Section 4.2 we obtain an algorithm that allows to determine an approximation of all the robot geometries such that the accuracy requirement will be satisfied over the workspace $\mathscr{W}$.

6. Calculation of the valid parameter boxes

The algorithms allow us to get the FPBs of the workspace and accuracy requirements as a list of design parameter boxes. Hence computing the intersection of the two sets of FPBs is straightforward and so is obtaining the VPBs. Two alternative strategies may be used to speed-up the computation:

1. Use the FPBs of the workspace requirement as the initial list $\mathscr{L}$ to verify the accuracy requirement. In that case the FBPs of the accuracy requirement will be the final VPBs.
2. Modify the algorithm principle to perform $F_w(P, Q)$ and $F_a(P, Q)$ simultaneously so that both the workspace and accuracy requirements will be checked, allowing the direct calculation of the VPBs.

7. Determining optimal parameter values

7.1. Design for relaxable requirements

Using the previous algorithms all the possible design solutions (VPBs) that fulfill all the compulsory requirements are obtained. But each design solution in VPBs may present very different performances apart from the compulsory requirements (Fig. 3), hence relaxable requirements such as inertia, cost, bandwidth, stiffness, etc. may be considered. Two alternative approaches will be used for these types of requirements:

1. *Sampling of the VPBs*. In this approach each box in the VPBs is sampled, providing a list of possible design solutions. The relaxable requirements are evaluated for each design solution, which provides a set of design solutions with various compromises for the relaxable requirements.
2. *Relaxed VPB*. FPBs can be computed also for the relaxed version of the relaxable requirements. Then the intersection of all FPBs will provide all the design solutions that satisfy both the compulsory requirements and the relaxed version of the relaxable requirements.

7.2. Computing the intersection of sorted lists

The motivation in both cases is to provide a list of design parameters presenting different compromises among relaxable requirements. For each relaxable requirement or other additional requirements we sort the design parameters p_i with descending order according to their performance indices and then compute the intersection of the top part of each list (Fig. 4), if the intersec-

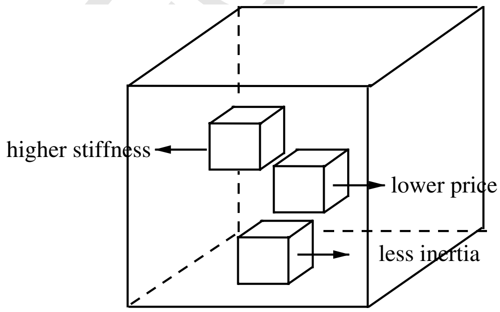

Fig. 3. Performances of each solution in VPBs.

Lower inertia	p_8	p_2	p_3	p_4	p_1	…
Higher stiffness	p_4	p_3	p_6	p_7	p_8	….
Lower price	p_2	p_5	p_8	p_4	p_6	…

Fig. 4. Sorted design parameter lists.

tion is not empty, then it gives the end-user several optimal parameter values (for instance p_4, p_8) with concrete indices to make the final decision, even considering some additional requirements that have not been specified during the design stage. It is clearly an advantage of our approach.

8. Algorithm in practice

The algorithms presented in the previous sections can provide all the design solutions that satisfy all the compulsory requirements simultaneously. But checking all the requirements over the full workspace is very computer intensive as the number of the unknowns (the design parameters and the pose parameters) may be large. Note that the algorithm presented in Section 4.2 also can be used to verify the compulsory requirements for a fixed value of the design parameters, with a greatly reduced computation time. Hence a simplified algorithm is used in practice to speed-up the computation:

1. Compute the VPBs for a relaxed version of the compulsory requirements at some specified poses.
2. Sample the obtained VPBs to get a set of potential design solutions.
3. Use the algorithm to verify each potential design solution over the whole workspace of the robot.

When performing step 3, instead of checking the requirements over the full desired workspace we propose to check them only in a limited set $\mathscr{R}$ of *check segments* connecting specific poses, called *check points*, belonging to $\mathscr{W}$. Hence loop 2 of the algorithm is replaced by a simpler loop performed along the check segments in $\mathscr{R}$. Each pose M on a segment connecting check points M_1, M_2 is described by

$$\mathbf{OM} = \mathbf{OM_1} + \lambda \mathbf{M_1M_2} \tag{25}$$

where λ is an interval parameter in the range [0, 1]. Hence the number of the unknowns in loop 2 is reduced from 6 (the pose parameters) to 1 (λ). The obtained VPBs are called the *relaxed VPBs* and are constituted of:

- the true VPBs;
- boxes defining geometries that satisfy the compulsory requirements at all poses in $\mathscr{R}$ but may not satisfy them over the full $\mathscr{W}$.

9. Application examples

This section presents a numerical example of optimal design based on our approach. It has been implemented using the high level interval analysis package `ALIAS` [1] which relies on the C++ interval arithmetics package BIAS/Profil.

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

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD = 5.0.1

Table 1
Specification for the design parameters

	R (mm)	r (mm)	α (deg)	β (deg)	l (mm)	s (mm)
Lower bound	500	130	10	10	500	1400
Upper bound	550	180	30	30	1200	2200
ϵ	10	10	2.8	2.8	100	100

The ranges and accuracy ϵ of the design parameters are presented in Table 1. The desired workspace is defined as a hyper-cube whose center is the nominal position. The vertices of the desired 6-D workspace are chosen as the check points, whose coordinates are defined by $\{x_i, y_j, z_{rk}, \phi_1, \phi_2, \phi_3\}$ $(i, j, k = 1, 2)$, where $x \in \{-100, 100\}$, $y \in \{-100, 100\}$, $z_r \in \{-500, 500\}$, $\phi_1 = \phi_2 = -\mathrm{p}/9$, $\phi_3 = -\mathrm{p}/6$. The check segments connecting the pair of check points (i, j) where $\{i = 1 \ldots 8, j = i + 1 \ldots 8\}$ are constructed based on equation (25). A very preliminary sequential implementation for the workspace requirement has been tested. The computation time is less than 48 hours on a PC (2.00 GHz) under Linux. All together, 13455 boxes have been tested and 8781 valid parameter boxes have been obtained with a total volume of 1.13958e + 08, the neglected volume consisting of ϵ boxes is 1.40415e + 07. We are quite confident that by using other powerful methods of interval analysis [27,28] the computation time can be reduced drastically.

A preliminary accuracy analysis has been performed by sampling the FPBs of the workspace and computing the worst case of positioning errors at the check points. It is noticed that the positioning accuracy seems to be very sensitive to the design parameters α, β and less sensitive to the other design parameters: the positioning errors increase with the parameters α, β (this observation is coherent with the works presented in [26,7]). As mentioned earlier requirements of workspaces and accuracy seem to be antagonistic. Therefore the result of a cost-function approach satisfying both requirements only reflects the relative weights that are used in the cost-function while our approach allows to obtain the design solutions that satisfy both a minimal workspace and optimal accuracy or the opposite.

10. Conclusion

A new methodology is proposed in this paper for the optimal design of parallel manipulators with multi-criteria requirements. The main differences with other classical approaches are that this methodology allows to obtain *all* the possible design solutions that satisfy a set of compulsory requirements (taking into account manufacturing errors) and make the best compromise for the relaxable requirements.

The prospective works are:

1. Improving the current implementation to reduce the computation time. For example note that verifying the violation or satisfaction of the requirement for a given Q_k is independent from the other Q_l of the list, so a distributed implementation can be used (and is available within `ALIAS`). A distributed implementation may reduce the computation time by a greater factor than the number of slave computers.

2. Automatizing the treatment of the workspace requirement so that other mechanical architectures can be treated as well. Indeed for any mechanical architecture we have a set P of design parameters, a set $\mathscr{S}$ of pose parameters and the robot workspace may be defined by the set of poses Q that satisfy some constraint relations $\mathscr{T}(P, Q) \leqslant 0$. The only differences between two mechanical architectures are the parameters in P, Q and the relations in $\mathscr{T}$. But the principle of our methodology is still valid for any architectures as soon as $P, Q, \mathscr{T}$ have been defined. Hence a symbolic pre-processing may be used to generate automatically this architecture-dependent module whose results will be taken as arguments for an optimal design kernel, thereby allowing to deal with any mechanical architectures with a minimum effort.
3. Extending the compulsory requirements to process other classical performance indices such as stiffness, joint forces/torques, joint velocities, etc.

References

[1] A.G. Erdman, Modern Kinematics, Wiley, New York, 1993.
[2] J. Angeles, The robust design of parallel manipulators, in: 1st Int. Colloquium, Collaborative Research Centre 562, Braunschweig, 29–30 May 2002, pp. 9–30.
[3] V.O. Astanin, V.V. Usov, Multi-objective synthesis of machining center configurations with parallel structure kinematics, in: 2nd Chemnitzer Parallelkinematik Seminar Chemnitz, 12–13 April 2000, pp. 299–312.
[4] K.S. Hong, J.-G. Kim, Manipulability analysis of a parallel machine tool: application to optimal link length design, J. Robotic Syst. 17 (8) (2000) 403–415.
[5] E. Ottaviano, M. Ceccarelli, Optimal design of CAPAMAN (Cassino parallel manipulator) with prescribed workspace, in: F.C. Park, C.C. Iurascu (Eds.), Computational Kinematics, 20–22 May 2001, pp. 35–44.
[6] X.S. Su, A real coded genetic optimal kinematic design of a Stewart fine tuning platform for a large radio telescope, J. Robotic Syst. 18 (9) (2001) 507–516.
[7] R. Stoughton, T. Arai, A modified Stewart platform manipulator with improved dexterity, IEEE Trans. Robotics Automat. 9 (2) (1993) 166–173.
[8] S. Bhattacharya, H. Hatwal, A. Ghosh, On the optimum design of a Stewart platform type parallel manipulators, Robotica 13 (2) (1995) 133–140.
[9] M.D. Bryfogle, C.C. Nguyen, Z.-l. Zhou, S.S. Antrazi, A methodology for geometry design of closed kinematic chain mechanisms, in: IEEE Int. Conf. on Robotics and Automation, Albuquerque, 21–28 April 1997, pp. 2974–2979.
[10] C. Gosselin, Determination of the workspace of 6-dof parallel manipulators, in: ASME Design Automation Conf., Montre'al, 17–20 September 1989, pp. 321–326.
[11] C.M. Gosselin, E. Lavoie, On the kinematic design of spherical three-degree-of-freedom parallel manipulators, Int. J. Robotics Res. 12 (4) (1993) 394–402.
[12] C.-S. Han, D. Tesar, A. Traver, The optimum design of a 6 dof fully parallel micromanipulator for enhanced robot accuracy, in: ASME Design Automation Conf., Montre'al, 17–20 September 1989, pp. 357–363.
[13] Z. Ji, Analysis of design parameters in platform manipulators, ASME J. Mech.-Des. 118 (December) (1996) 526–531.
[14] O. Ma, J. Angeles, Architecture singularities of platform manipulator, in: IEEE Int. Conf. on Robotics and Automation, Sacramento, 11–14 April 1991, pp. 1542–1547.
[15] O. Masory, J. Wang, Workspace evaluation of Stewart platforms, in: 22nd Biennial Mechanisms Conf., Scottsdale, 13–16 September 1992, pp. 337–346.
[16] K.H. Pittens, R.P. Podhorodeski, A family of Stewart platforms with optimal dexterity, J. Robotic Syst. 10 (4) (1993) 463–479.

MAMT 997
16 August 2004 Type
ARTICLE IN PRESS
No. of Pages 15, DTD=5.0.1

[17] R.C. Stamper, C.-W. Tsai, G.C. Walsh, Optimization of a three dof translational platform for well-conditioned workspace, in: IEEE Int. Conf. on Robotics and Automation, Albuquerque, 21–28 April 1997, pp. 3250–3255.
[18] K.E. Zanganeh, J. Angeles, Kinematic isotropy and the optimum design of parallel manipulators, Int. J. Robotics Res. 16 (2) (1997) 185–197.
[19] F. Bernelli-Zazzera, D. Gallieni, Analysis and design of an hexapod mechanism for autonomous payload pointing, in: 46th IAF Congress, Oslo, 2–6 October 1995.
[20] B. Claudinon, J. Lievre, Test facility for rendez-vous and docking, in: 36th Congress of the IAF, Stockholm, 7–12 October 1985, pp. 1–6.
[21] T.R.J. Corrigan, S. Dubowsky, Emulating micro-gravity in laboratory studies of space robotics, in: ASME Design Automation Conf., pp. 109–116, Minneapolis, 11–14 September 1994, pp. 109–116.
[22] J.-P. Merlet, Articulated device, for use in particular in robotics, United States Patent 5,053,687, 1 October 1991.
[23] O. Masory, J. Wang, H. Zhuang, On the accuracy of a Stewart platform-part II: Kinematic calibration and compensation, in: IEEE Int. Conf. on Robotics and Automation, Atlanta, 2–6 May 1993, pp. 725–731.
[24] J.-P. Merlet, D. Daney, A formal-numerical approach to determine the presence of singularity within the workspace of a parallel robot, in: F.C. Park, C.C. Iurascu (Eds.), Computational Kinematics, Seoul, 20–22 May 2001, pp. 167–176.
[25] J.-P. Merlet, An improved design algorithm based on interval analysis for parallel manipulator with specified workspace, in: IEEE Int. Conf. on Robotics and Automation, Seoul, 23–25 May 2001.
[26] J.-P. Merlet, Determination of 6d workspaces of Gough-type parallel manipulator and comparison between different geometries, Int. J. Robotics Res. 18 (9) (1999) 902–916.
[27] M. Kieffer, L. Jaulin, I. Braems, E. Walter, Guaranteed set computation with subpavings, in: W. Kraemer, J.W. Gudenberg (Eds.), Scientific Computing, Validated Numerics, Interval Methods, Kluwer Academic Publishers, Boston/Dordrecht/London, 2001, pp. 167–178.
[28] L. Jaulin, M. Kieffer, O. Didrit, E. Walter, Applied Interval Analysis, Springer-Verlag, London, 2001.