On Robust Mechanical Design of PAR2 Delta-Like Parallel Kinematic Manipulator

Allaoua BRAHMIA
LGMM Laboratory, Faculty of Technology, Université 20 août 1955-Skikda,
PB N° 26 Route Elhadaik, Skikda 21000, Algeria
Email: a.brahmia@univ-skikda.dz

Ridha KELAIAIA *
LGMM Laboratory, Faculty of Technology, Université 20 août 1955-Skikda,
PB N° 26 Route Elhadaik, Skikda 21000, Algeria
Email: r.kelaiaia@univ-skikda.dz

Ahmed CHEMORI
LIRMM, University of Montpellier, CNRS, Montpellier, France
Email: Ahmed.chemori@lirmm.fr

Olivier COMPANY
LIRMM, University of Montpellier, CNRS, Montpellier, France
Email: company@lirmm.fr

Sensitivity analysis of manipulators aims at studying the influence of variations in its own geometric parameters on its performance. This information is useful for evaluating the position error of the end-effector as well as for the synthesis of tolerances. Indeed, the synthesis of tolerances is a very important issue in the design and manufacturing of robot manipulators. In this paper, a sequential procedure for modeling, dimensioning and tolerance synthesis of the parallel kinematic manipulator PAR2 is proposed. For optimal dimensional design, an approach based on the optimization of the workspace is proposed, taking into account several constraints; followed by a numerical matrix analysis based deterministic method for sensitivity analysis whose performance is studied in terms of accuracy. To calculate the optimal dimensional tolerances, a new tolerance synthesis method is used. The effect of geometric tolerance on accuracy is analyzed.

1 Introduction

Robust design of a mechanism aims at making its performance optimal and insensitive to variations. When variations are ignored, unrobust designs may result. However, the sensitivity analysis of manipulators consists in studying the influence of variations in geometric parameters on its performance. This information is inherently useful for the evaluation of the position error of the end-effector as well as for tolerance synthesis. The calculation of part tolerance in manufacturing and assembly phases of robots is crucial in the design of the robots, since it directly affects the product quality and the manufacturing cost.

Parametric variations are unavoidable due to the machining and assembling of the mechanism. They are called parameter uncertainty in the design phase when the actual values are unknown. Such parameter uncertainty might result in the dramatic changes of the performances if the performances are sensitive to the variations. Robust design would prevent this catastrophic design result and sensitivity analysis is the priority. The concept of robustness was first introduced by Taguchi [1] in 1978. He presented a planned experimental method for selecting the values of design variables so that the effects of uncontrollable parameters on the system performance is minimized. However, robust design consists in minimizing the sensitivity of performance against variations without controlling the causes of these last ones [2]. In the literature, several authors have contributed to the formulation of robust design problems. In [3], the parametric sensitivity of a 5-DoF parallel manipulator was studied with respect to the mass and stiffness performance on the basis of response
surface method and performance reliability. Essential parameters were then selected to be the design variables of the robust design concerning the stiffness and mass [4], and elastodynamic performance [5] of the parallel manipulator. The parametric uncertainty were measured by a stochastic model and the optimal performances were reliable [6]. The implemented robust design is crucial since it affects directly the mechanism quality and manufacturing cost. It is also useful for the evaluation of the position error as well as for tolerance synthesis. Deterministic methods based on matrix numerical analysis are used for tolerance calculation and sensitivity analysis of mechanisms, where the robustness problem is commonly known as “Conditioning” [7–9]. Caro [10] has developed an effective method for tolerance synthesis, based on a robust design approach, this method is divided into two steps: the first step deals with calculating robust dimensions using an appropriate robustness index; as for the second step, a tolerance synthesis method is developed for calculating the optimal tolerance box. This method was successfully applied to evaluate the sensitivity of the end-effector of a three-axis Orthoglide (3 translation degrees of freedom (dof)). Caro et al. [11] studied the influence of dimensional and angular variations on the position of the end-effector of an orthoglide at three dof of translation. Indeed, two sensitivity indices were used, one for the study of position sensitivity and the other one for the study of orientation sensitivity. Rout et al. [12, 13] proposed to use a probabilistic approach to model the effects of noise factors on a 2RR plane manipulator, and adopted an experimental design technique to select optimal tolerances of kinematic and dynamic parameters for minimum performance variation. Rout et al. [14] proposed an evolutionary technique for selecting the optimal tolerance parameters of a 2RR planar manipulator at 2 dof. In [15] Kim et al. have developed an effective method to evaluate the reliability of dimensional tolerances and joint clearances. The kinematic reliability of the positioning and orientation repeatability of an RRR manipulator are evaluated analytically using the AFOSM (First Order Second Moment) method. A stochastic method is proposed for the modelling of the mechanism. This method may help designers to choose dimensional tolerances and joint clearances to obtain an optimal performance of robotic manipulators. Yang et al. [16] proposed an approach based on multi-objective optimization of parallel tracking mechanism by taking into account simultaneously several performance criteria such as workspace, kinematic, stiffness, and dynamic performances and considering parameter uncertainty. Xianzhen et al. in [17] developed a method for robust design of tolerances dedicated to function generation mechanisms. They have improved and applied the Taguchi method to determine the optimal tolerances of components to minimize the total assembly cost, while satisfying the precision requirement of the mechanism. The effectiveness of the proposed method is illustrated through an example of four-bar function generation. In their work, Goldsztjein et al. [18, 19] proposed a method for tolerance synthesis of parallel robots. The local uniqueness hypothesis is used to calculate the upper limit of the position error. This technique uses the Kantorovich theorem for numerical calculation. Another technique based on the optimization of the Generaled Reduced Gradient was proposed by Trang et al. [20] to calculate the tolerances of robot parts. This algorithm is used to solve a multi-variable nonlinear optimization problem. By definition, the difference between the upper and lower limits of the nominal value of a design variable is called “tolerance” [21]. Tolerance is a very important concept in the design and manufacturing phase. Several studies in the literature were focused on the relationship between dimensional tolerances and manufacturing cost. The manufacturing cost of a mechanism increases when their dimensional tolerances are tight. Besides the effect of tolerance on the robot performance is crucial, since this last one is very sensitive to variations in the dimensions of the robot components.

In this paper, we propose the calculation of the dimensional tolerances of PAR2 parallel manipulator [22–24] through the minimization of the position error of the end-effector of the robot. To calculate the optimal dimensional tolerances, a new tolerance synthesis method is used. This method is known as optimal tolerance box method (Brahmia-TB). The rest of the paper is organized as follows: In section 2, the problem of robust design in the purely deterministic domain is presented. Then, the optimal dimensioning of PAR2 parallel manipulator is presented in section 3. The tolerance synthesis of the PAR2 parallel manipulator is illustrated in section 4.

2 Modeling of the planar PAR2 parallel manipulator

2.1 Geometric description

It’s a parallel mechanical architecture with two DOFs, composed of two motorized (active) kinematic chains and two constraint passive chains built in the transverse plane to increase the stiffness of the robot in that plane (Fig. 1 and Fig. 2). This design allows two translations in the vertical plane, while guaranteeing a good stiffness in the transverse plane [22–26]. Indeed, the passive arms are used to prevent, as far as possible, perpendicular movements (out of the plane xoz, as illustrated in Fig. 2). It is worth to note that for the controlled directions, the influence of the passive arms is done in second order on the xz position. In this case this influence can be neglected. It is worth to note that for simplification purpose of realisation of the architecture, it
was considered that the two branches $P_1A_1B_1$ and $P_2A_2B_2$ are identical. Therefore, the geometric parameters of this manipulator (Fig. 3) are as follows:

$L_a$: Arm length $P_iA_i$, ($i \in \{1, 2\}$

$L_b$: Arm length $A_iB_i$

$d$: Distance between points $B_1$ and $B_2$ on the mobile platform

$D$: Distance between the axes of motorized rotary links.

$(O-x,y)$: The reference frame related to the base, such as $"O"$ is the center of $P_1P_2$

$C(x,z)$: is the center of $B_1B_2$

### 2.2 Forward Kinematic Model (FKM)

The FKM of the PAR2 parallel manipulator expresses the operational coordinates $[x \ z]^T$ (position of the end-effector) as a function of the joint coordinates $[q_1 \ q_2]^T$. It can be obtained from the following kinematic constraint:

$$L_b = ||B_iA_i||$$ (1)

Leading to a system of two equations with two unknowns:

$$\begin{align*}
(x_{B_1} - x_{A_1})^2 + (z_{B_1} - z_{A_1})^2 &= L_b^2 \\
(x_{B_2} - x_{A_2})^2 + (z_{B_2} - z_{A_2})^2 &= L_b^2
\end{align*}$$ (2)

The resolution of this system gives the following [24]:

$$z = -\frac{\beta \pm \sqrt{\beta^2 - 4\alpha \gamma}}{2\alpha}$$ (4)

$$x = az + b$$ (5)

In equation (4), we take the sign $(−)$ because the kinematic chains of the robot evolve in the negative half-plane ($z < 0$).

With:

$$\begin{align*}
x_{A_1} &= x_{P_1} + L_a \cos(q_1) = \frac{D}{2} + L_a \cos(q_1) \\
x_{A_2} &= x_{P_2} - L_a \cos(q_2) = -\frac{D}{2} - L_a \cos(q_1) \\
x_{B_1} &= x + \frac{d}{2} \\
x_{B_2} &= x - \frac{d}{2} \\
z_{A_1} &= -L_a \sin(q_1) \\
z_{A_2} &= -L_a \sin(q_2) \\
z_{B_1} &= z_{B_2} = z
\end{align*}$$

$$\begin{align*}
a &= \frac{L_a(\sin(q_1) - \sin(q_2))}{D - d + L_a(\cos(q_1) + \cos(q_2))} \\
b &= \frac{L_a(\cos(q_1) - \cos(q_2))(D - d)}{D - d + L_a(\cos(q_1) + \cos(q_2))} \\
c &= b - L_a \cos(q_1) - \left(\frac{D - d}{2}\right) \\
\alpha &= a + 1 \\
\beta &= 2ac + 2L_a \cos(q_1) \\
\gamma &= c^2 + (L_a \cos(q_1))^2
\end{align*}$$

2.3 Determination of the workspace of the PAR2 parallel manipulator

One of the most important performance criteria to consider when designing a parallel kinematic manipulator is the workspace. The approach of design of PAR2 parallel kinematic manipulator was done in two stages: the first one was to size the active part; then in the second stage, the passive part was designed in a way that it admits all possible positions of the mobile platform. To determine the workspace of PAR2 parallel manipulator, only the active chains are considered; consequently, the second stage is not represented in this study. In this case, the workspace is delimited by the angles $\theta, q_1, q_2, \psi_1$ and $\psi_2$ (cf. Fig.4). It is represented by the surface described by the following equation [24]:

$$S_W = \int_W dW = \frac{1}{2} \int_{\theta_{\text{min}}}^{\theta_{\text{max}}} (L_w^{\text{max}} - L_w^{\text{min}}) \, d\theta$$ (6)

With:

$$\begin{align*}
\frac{4\pi}{3} &\leq \theta \leq \frac{5\pi}{3} \\
\frac{\pi}{6} &\leq (q_1 + \psi_1) \leq \frac{11\pi}{12}
\end{align*}$$ (7)

$\theta$: must be chosen such that singular configurations are avoided. The determination of $L_w^{\text{min}}$ and $L_w^{\text{max}}$ is performed in the following way: $OC = OP + P_1B_1 + P_2C$

The distance $L_w$ between the origin of the fixed base (point $O$) and the centre of the mobile platform (point $C$) is obtained by the projection of the equation (8) on the $(xoz)$
The condition number of the matrix $J$ plane:

Fig. 4. Illustration of the angles delimiting the robot’s workspace.

$$L = \sqrt{x_c^2 + z_c^2},$$

$$\|\overrightarrow{OC}\| = \frac{D}{2},$$

$$\|\overrightarrow{OP}\| = \sqrt{L_o^2 + L_b^2 - 2 L_o L_b \cos(q_1 + \psi_1)},$$

$$\|\overrightarrow{P_1 B_1}\| = \sqrt{L_o^2 + L_b^2 - 2 L_o L_b \cos(q_1 + \psi_1)},$$

$$\|\overrightarrow{B_1 \dot{C}}\| = \frac{d}{2}.$$

$L_W = \|\overrightarrow{OC}\| = \sqrt{(\frac{D-d}{2})^2 + L_o^2 + L_b^2 - 2 L_o L_b \cos(q_1 + \psi_1) + (D-d)(L_o \cos(q_1) - L_o \cos(\psi_1))}$

Where:

$L_W \rightarrow L_{W_{\text{min}}}$ if: $(q_1 + \psi_1) \rightarrow \frac{\pi}{6}$

$L_W \rightarrow L_{W_{\text{max}}}$ if: $(q_1 + \psi_1) \rightarrow \frac{11\pi}{12}$

2.4 Kinematic performance

The kinematic performance of PAR2 parallel manipulator can be defined as the ability of its end-effector to:

1. Accurately and easily perform small arbitrary movements around a point in the workspace [24];
2. Apply in all directions of the workspace, forces and moments [27].

The condition number of the matrix $J^{-1}$ noted as $\kappa_J$ is used to measure the performance of the robot because the Jacobian matrix is homogenous [28]. In case, where the Jacobian matrix isn’t homogenous (contains different units) the condition number has no clear physical meaning, as the rotations are transformed arbitrarily into “equivalent” translations [29, 30]. It is defined by the ratio between its largest and smallest singular values, which are respectively denoted by $\tau_{\text{max}}(J^{-1})$ and $\tau_{\text{min}}(J^{-1})$:

$$\kappa_J = \frac{\tau_{\text{max}}(J^{-1})}{\tau_{\text{min}}(J^{-1})}$$

The Jacobian matrix is obtained through the time derivative of equations (2) and (3) leading to:

$$J_x \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} + J_q \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} = 0$$

Hence:

$$J^{-1} = -J_q^{-1}J_x$$

With:

$$J_x = \begin{bmatrix} (x - \frac{D-d}{2}) - L_o \cos(q_1) + L_a \sin(q_1) \\ (x + \frac{D-d}{2}) + L_o \cos(q_2) + L_a \sin(q_2) \end{bmatrix}$$

$$J_q = \begin{bmatrix} J_{q_{11}} & 0 \\ 0 & J_{q_{22}} \end{bmatrix}$$

$$J_{q_{11}} = \left[ (x - \frac{D-d}{2}) - L_o \cos(q_1) \right] L_a \sin(q_1) + \left[ z + L_a \cos(q_1) \right] L_o \cos(q_1)$$

$$J_{q_{22}} = -\left[ (x + \frac{D-d}{2}) + L_o \cos(q_2) \right] L_a \sin(q_2) + \left[ z + L_a \sin(q_2) \right] L_o \cos(q_2)$$

2.5 Analysis of the singularities

There are several methods (geometrical, algebraic, etc.) are used to determine the singular configurations [30, 31]. In this work, the method based on the determination of the roots of the determinant of the PAR2 parallel manipulator inverse Jacobian matrix is employed to analyse and find these configurations. The matrices $J_x$ and $J_q$ can be determined by using the equiprojectivity propriety of the velocities in the forearm [25]:

$$V_{A_1, A_2 B_i} = V_{B_i, A_i B_i}$$

Where $V_{A_1}$ and $V_{A_2}$ denote the vectors giving respectively the velocities of points $A_1$ and $B_i$. It leads to the following result:

$$J_x = \begin{bmatrix} A_1 B_1 . e_x & A_1 B_1 . e_z \\ A_2 B_2 . e_x & A_2 B_2 . e_z \end{bmatrix}$$

And

$$J_q = \begin{bmatrix} (A_1 B_1 \times P_i A_1) . e_y \\ 0 \\ (A_2 B_2 \times P_i A_2) . (-e_y) \end{bmatrix}$$

Where: $e_x = [1 \ 0 \ 0]^T$, $e_y = [0 \ 1 \ 0]^T$ and $e_z = [0 \ 0 \ 1]^T$.

The employed method consists in the analysis of the two Jacobian matrices (i.e. $J_x$ and $J_q$). Accordingly, we distinguish the following cases [30]:

1. Type 1 (Serial singularity: $J_q$ is singular): $|J_q| = 0 \Rightarrow (A_1 B_1 \times P_i A_1)(A_2 B_2 \times P_i A_2) = 0 \Rightarrow A_1 B_1 \times P_i A_1 = 0$ or $A_2 B_2 \times P_i A_2 = 0$.

This singularity appears when one of the arms $A_i B_i$ becomes parallel to the forearms $P A_i$ in the same kinematic chain. In this case, it is not possible for the manipulator to generate velocities of the end-effector in some directions. These singularities represent the limits of the reachable workspace. In these configurations, the robot loses one or more degree (s) of freedom.

2. Type 2 (Parallel singularity: $J_x$ is singular): $|J_x| = 0$, means that $A_1 B_1$ and $A_2 B_2$ are coplanar, which corresponds to the appearance of uncontrollable mobilities of the end-effector, because it is possible to move it while the motorized joints are blocked. These singularities can exist inside the reachable workspace of the robot, which may introduce additional difficulties for the
trajectory planning. In these configurations, the robot gains one or more degree(s) of freedom and the stiffness of the robot is locally lost.

3. Type 3 (both $J_{e}$ and $J_{i}$ are singular):
   In this case, the end-effector can move while the actuators are blocked and vice versa. Fore more details about the selected dimensions, the reader can refer to Table 1, this case cannot be reached.

3 Optimal dimensioning of PAR2 parallel manipulator

The problem of optimal dimensioning of PAR2 parallel manipulator can be formulated as follows:

**Find an optimal vector $P^*_1 = [(D-d)^*, L^*_a, L^*_b]^T$ that:**

$$
\min F(P) = \min \left[ f_1 = -S_w \right] \quad f_2 = \kappa_l
$$

Subject to:

$$
\begin{align*}
\frac{19\pi}{12} &\le q_1 \le 2\pi \\
\pi &\le q_2 \le \frac{17\pi}{12} \\
0.07 m &\le D - d \le 0.18 m \\
0.20 m &\le L_a \le 0.6 m \\
0.20 m &\le L_b \le 1.0 m
\end{align*}
$$

3.1 Resolution approach

There are different ways to resolve this problem: The most usual one consists of treating successively the objectives (the result may give advantage to extreme solutions). Other techniques consist in transforming the multi-objective optimization problem into a single-objective optimization problem, for which there exist various methods (goal attainment, method of compromise, etc.). In our case, the above optimization problem can be resolved by transforming it into a single-objective optimization problem using the own inequality constraint method [27], keeping only one objective function and transforming the others in the form of inequalities.

**Presentation of the method:**

Let us consider the following problem of multi-objective optimization:

**Find a vector $P^* = [p_1^*, p_2^*, \ldots, p_n^*]^T$ that:**

**Minimizes** $F(P) = [f_1(P), f_2(P), \ldots, f_k(P)]^T$

with:

$$
g_m(P) \le 0 \text{ (m inequality constraints)}$$

and

$$
h_l(P) = 0 \text{ (l equality constraints)}$$

$P^* \in \mathbb{R}^n$: Vector of the decision variables

$F(P^*) \in \mathbb{R}^k$: Vector of the objectives function

$g(P^*) \in \mathbb{R}^m$: Vector of inequality constraints

$h(P^*) \in \mathbb{R}^l$: Vector of equality constraints

To transform this Multi-optimization problem into a Single-optimization problem, we propose to proceed as follows:

- We choose an objective to optimize as a priority;
- We choose an initial constraints vector;
- We transform the problem by keeping the priority objective and by transforming the other objectives into constraints of inequality. Consequently, we get:

Find a vector $P^* = [p_1^*, p_2^*, \ldots, p_n^*]^T$ that:

**Minimizes** $f_k(P)$

With:

$$
f_1(P) \le \varepsilon_1$$

$$f_{k-1}(P) \le \varepsilon_{k-1}$$

$$g_m(P) \le 0 \text{ (m inequality constraints)}$$

and

$$h_l(P) = 0 \text{ (l equality constraints)}$$

$P^* \in \mathbb{R}^n$: Vector of the decision variables

$F(P^*) \in \mathbb{R}^k$: Vector of the objectives function

$g(P^*) \in \mathbb{R}^m$: Vector of inequality constraints

$h(P^*) \in \mathbb{R}^l$: Vector of equality constraints

This set of constraints delimits the search space for searching the optimal solution. By applying the proper Inequality Constraints method on the problem defined in section 3, we obtain:

- The workspace ($f_1 = S_w$) is chosen as a priority objective to optimize,
- The Kinematic performance ($f_2 = \kappa_l$) objective function is transformed into constraint ($\kappa_l \le 5$),
- $\varepsilon=[5 2\pi -19\pi \pi -0.07 0.18 -0.20 0.45 -0.20 0.45]^T$ is chosen as an initial constraints vector.
- The obtained single-objective optimization problem is then given by:

Find a vector $P^* = [(D-d)^*, L^*_a, L^*_b]^T$ that:

**Minimizes** $S_w$

Subject to:

$$
\kappa_l \le 5$$

$$\frac{19\pi}{12} \le q_1 \le 2\pi$$

$$\pi \le q_2 \le \frac{17\pi}{12}$$

$$0.20 m \le D - d \le 0.70 m$$

$$0.20 m \le L_a \le 0.60 m$$

$$0.20 m \le L_b \le 1.00 m$$

The choice of $\kappa_l \le 5$ is justified by: When $\kappa_l \to \infty$ (see equation (9)), $J^{-1}$ becomes a singular matrix. Physically, this means that the PAR2 parallel manipulator is in a singular configuration, and when $\kappa_l \rightarrow 1$, in this case the configuration is called isotropic and the robot end-effector has the same facility to move in all the directions, which is highly desirable. On the other hand, the PAR2 parallel manipulator (which is dedicated to high-speed applications for a wide range of assembly, picking, material handling, packaging, quick transfer in micro-engineering, and pharmaceutical industries, etc.) loses his capacity of velocity. For this reason, we have chosen $\kappa_l \le 5$ to keep this capacity.

The results of workspace optimization are summarized in Table 1.
Table 1. Summary of the results of workspace optimization.

<table>
<thead>
<tr>
<th>Vector ( P_i^o ) of the obtained optimal geometric parameters</th>
<th>Optimal objective function</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D - d )</td>
<td>( L_a )</td>
</tr>
<tr>
<td>0.440 m</td>
<td>0.400 m</td>
</tr>
</tbody>
</table>

4 Tolerance synthesis of the PAR2 parallel manipulator

For the tolerance analysis, we use a deterministic method based on numerical matrix analysis, this method requires a linear relationship between the performance of the system to be designed and the design variables. However, the use of the dynamic model, which is often a non-linear relationship (introduction of the couples required to actuate the two joints), does not allow to give results.

In this paper, we do not pretend to obtain the tolerances of all the parts (i.e. the design drawing which specify all of their dimensions and tolerances). At our designer level, we give only indications about the principal dimensions. In order making the tolerance analysis robust, we need to minimize the sensitivity to dimensional variations, i.e. the dimensional tolerances must be small values, which makes the manufacturing cost high. So, the problem is to minimize the manufacturing cost while always respecting the constraint on the position error (always less than 10 µm). As part of our technique for dimensional tolerances calculation technique, a two-steps sequential approach is proposed. In the first step (first optimization), The Tolerance Box method is used for the synthesis of tolerances. It considers that a variation in the design variables, located on the hyper-ellipsoid boundary \( \xi(\|\delta f\|_{\max}) \), will generate a variation in performance of normal equal to \( \|\delta f\|_{\max} \). The optimal dimensional tolerances \( \Delta y_{opt} \) are calculated by resolving the following optimization problem:

\[
\begin{align*}
\max_u \prod_{i=1}^{k} |u_i| \\
\text{such that } U(u_1, u_2, \ldots, u_k) \in \xi(\|\delta f\|_{\max}) \\
|u_i| \geq \Delta y_{imin}, i = 1, \ldots, k
\end{align*}
\]

\( \Delta y_{imin} \) is the minimum allowed dimensional tolerance for the variable \( \Delta y_{opt} \). The optimal tolerances \( \Delta y_{opt} \) (Fig. 5) of the design variables \( y_i \) are straightforwardly deduced from the vector \( U \), solution of the previous optimization problem:

\[
\Delta y_{opt} = |u_i|, i = 1, \ldots, n
\]

Point \( U \) must belong to \( \xi(\|\delta f\|_{\max}) \) if and only if \( U^T J_u y J_u = \|\delta f_{\max}\|_2^2 \). \( V \) is the eigenvector associated with the maximum singular value of the Jacobian matrix of sensitivity \( J_u \). Knowing that the vector \( U(u_1, u_2, \ldots, u_k) \) is none other than the vector \( \Delta y_{opt} (\Delta y_{1opt}, \Delta y_{2opt}, \ldots, \Delta y_{kopt}) \), then, the second step (second optimization) consists in optimizing the vector \( U(u_1, u_2, \ldots, u_k) \) towards a more robust solution \((U^*(u_{1opt}, u_{2opt}, \ldots, u_{kopt})\)). This optimization requires the introduction of the design parameter sensitivity criterion. However, we must choose the parameter that has the most influence on the robot sensitivity (i.e. \( \Delta y_{jopt} \)), then we multiply this parameter by a reduction coefficient \( K \) which will be determined later. Indeed, the choice of the design variable for which the robot is more sensitive is made in two ways:

- If a sensitivity study is performed, our design variable is selected directly
- In case there is no sensitivity study, we perform tests with our algorithm until the desired tolerances are obtained.

After the determination of the tolerance \( \Delta y_{jopt} \), the optimization calculation is repeated with the following new constraint:

\[
u_{jopt}^* = K_1 u_{jopt}, j = 1, \ldots, m
\]

Such as : \( m \): the number of main tolerances (errors).

In order not to have too tight tolerances of \( \Delta u_{jopt}^* \), it is necessary to choose a value of the coefficient \( K_1 \) such as \( K_1 u_{jopt} \leq \Delta u_{jopt}^* \), i.e. \( 0.7 \leq K_1 \leq 0.9 \). The choice of the coefficient \( K_1 \), must be done so that the value of the tolerance \( \Delta u_{jopt}^* \) must not be lower than the value of the dimensional tolerance tolerated for the variables \( u_{jopt}^* \). For the calculation of the new dimensional tolerances, the following optimization problem is solved:

\[
\begin{align*}
\max_{\Delta y_{opt}} \prod_{i=1}^{k} |u_{iopt}| \\
\text{such that } U_{opt}^*(u_{1opt}, u_{2opt}, \ldots, u_{kopt}) \in \xi \\
u_{jopt}^* = K_1 u_{jopt}, j = 1, \ldots, m \\
u_{jopt}^* \geq \Delta y_{imin}, i = 1, \ldots, k
\end{align*}
\]

Such as : \( \Delta y_{imin} \) : is the tolerance of the dimension \( y_i \) which has the main error \( \delta y_i \).

\( u_{jopt} \) : is the parameter most influencing the sensitivity of the robot.

Therefore, the new tolerance box, named Brahmia-BT shown in Fig. 6 contains wider tolerances and at the same time guarantees an accuracy that does not exceed 10 µm, and also does not include defective parts (rejects). This technique makes
it possible to decrease the tolerance of a single parameter and increase the tolerances of all other parameters, making the manufacture of the mechanisms less expensive. Fig. 6 illustrates our method of tolerance synthesis, compared to the work done by Jianmin [8] and Caro [10]. We notice that the tolerances box named Jianmin-BT is the largest but includes defective parts (scrap), i.e., mechanisms whose norm of variations in performance is greater than \(\|\delta f\|_{\text{max}}\). While the tolerance box named Caro-BT is the largest one that does not contain defective parts. Our tolerances box named Brahmi-BT, in addition to not containing defective parts, it allows to obtain larger tolerances with a minimum manufacturing cost compared to the Caro-BT tolerances box.

To validate our approach, we applied it for the tolerances synthesis of a parallel architecture manipulator called PAR2. Based on the illustration of Fig. 4, we have:

\[
\overrightarrow{OC} = D_i \overrightarrow{h_i} + L_{a_i} \overrightarrow{u_i} + L_{b_i} \overrightarrow{v_i} + d_i \overrightarrow{g_i} \quad i = 1, \ldots, 2 \tag{15}
\]

With: \(\overrightarrow{h_i}\) is the unit vector \(\frac{\overrightarrow{OP}}{\|\overrightarrow{OP}\|_2}\), \(\overrightarrow{u_i}\) is the unit vector \(\frac{\overrightarrow{PA_i}}{\|\overrightarrow{PA_i}\|_2}\), \(\overrightarrow{v_i}\) is the unit vector \(\frac{\overrightarrow{AB_i}}{\|\overrightarrow{AB_i}\|_2}\), \(\overrightarrow{g_i}\) is the unit vector \(\frac{\overrightarrow{BC}}{\|\overrightarrow{BC}\|_2}\).

\[
\begin{align*}
\overrightarrow{h_1} &= \overrightarrow{x}, \overrightarrow{h_2} = -\overrightarrow{x} \\
\overrightarrow{u_1} &= \cos(q_1) \overrightarrow{x} - \sin(q_1) \overrightarrow{z}, \overrightarrow{u_2} = \cos(q_2) \overrightarrow{x} - \sin(q_2) \overrightarrow{z} \\
\overrightarrow{v_1} &= -\cos(\psi_1) \overrightarrow{x} - \sin(\psi_1) \overrightarrow{z}, \overrightarrow{v_2} = -\cos(\psi_2) \overrightarrow{x} - \sin(\psi_2) \overrightarrow{z} \\
\overrightarrow{g_1} &= -\overrightarrow{x}, \overrightarrow{g_2} = \overrightarrow{x}
\end{align*}
\]

If we consider only the dimensional variations, we obtain after differentiation of equation (15) the following relation:

\[
\delta \overrightarrow{OC} = \delta D_i \overrightarrow{h_i} + \delta L_{a_i} \overrightarrow{u_i} + \delta L_{b_i} \overrightarrow{v_i} + \delta d_i \overrightarrow{g_i} \tag{17}
\]

Where: \(\overrightarrow{OC}\) : Variation of the end-effector position, its components are \([\delta y \ \delta z]^T\)

\(\delta D_i\): Represents the variation of the length \(D_i\) (nominal length \(D_i = 0.275 m\))

\(\delta L_{a_i}\): Represents the variation of the length \(L_{a_i}\) (nominal length \(L_{a_i} = L_a = 0.4 m\))

\(\delta L_{b_i}\): Represents the variation of the length \(L_{b_i}\) (nominal length \(L_{b_i} = L_b = 0.6 m\))

\(\delta d_i\): Represents the variation of the length \(d_i\) (nominal length \(d_i = 0.055 m\))

The relationship between the position error of the end-effector \(C\), \(\delta f_i\) and the dimensional variations \(\delta D_i, \delta L_{a_i}, \delta L_{b_i}, \delta d_i\), can be expressed by the following equation:

\[
\delta f = J \delta y \tag{18}
\]

With:

\[
J = \begin{bmatrix}
1 & -\cos(\psi_1) & -1 & -\cos(q_2) & \cos(\psi_2) & 1 \\
0 & -\sin(\psi_1) & -1 & -\sin(q_2) & -\sin(\psi_2) & 0
\end{bmatrix}
\]

\(\delta y = [\delta D_1 \ \delta L_{a_1} \ \delta L_{b_1} \ \delta d_1 \ \delta d_2 \ \delta d_3 \ \delta d_4 \ \delta d_5 \ \delta d_6]^{T}
\]

The sensitivity matrix \(S\) (see Appendix, equation (22)) is a semi-positive definite matrix, hence \(rank(S) = 2 < k = 8\).

From equation (23), the eigenvalues of the most constraining cylinder are calculated. These eigenvalues are given by: \(\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5 = \lambda_6 = \lambda_7 = 0.5607, \lambda_8 = 7.4393\).

From equation (28), one can deduce:

\[
\frac{\lambda_8}{\lambda_4} = (0.01)^2 = 0.0001(\|\delta f\|_2^2 = (0.01)^2, \text{which is an imposed value}).
\]

Let us now take the modification coefficient \(K = 0.03\) [32, 33].

The eigenvalues of the feasible space can be obtained from equation (26). These eigenvalues are given by

\[
\tilde{\lambda}_1 = \tilde{\lambda}_2 = \tilde{\lambda}_3 = \tilde{\lambda}_4 = \tilde{\lambda}_5 = \tilde{\lambda}_6 = 0.1021 = \tilde{\lambda}_7 = 0.5607, \quad \tilde{\lambda}_8 = 7.4393.
\]

With the eigenvalues and eigenvectors \(P_i\) previously obtained, the characteristic matrix \(\tilde{S} = P D P^T\) (see equation (28)) can then be constructed. The critical ellipse (noted \(\xi_{cri}\)) used in the optimization problem to calculate the optimal tolerances is that corresponding to the angles \(q_i\) and \(\psi_i\). The robot posture equivalent to this position is shown at point \(P1\) in Fig. 7. To calculate the dimensional tolerances \(\Delta D_{opt}, \Delta L_{aopt}, \Delta L_{bopt}, \Delta d_{opt}\), of the lengths \(D_i, L_{a_i}, L_{b_i}, d_i\), respectively, the following optimization problem is proposed:

\[
\begin{cases}
\max_u |u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7\ u_8| \\
\text{such that } U(u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7\ u_8) \in \xi_{cri} \\
u_i \text{Sign}(|V_i|) \geq 0, i = 1, \ldots, 8 \\
|u_i| \geq 0.5 \mu m, i = 1, \ldots, 8
\end{cases}
\]

The constraint \(|u_i| \geq 0.5 \mu m\) is the tolerated dimensional tolerance for the variables \(\delta D_i, \delta L_{a_i}, \delta L_{b_i}\), and \(\delta d_i\). The solution of the optimization problem is calculated using the Matlab function \textit{fmincon}. The "interior-point" algorithm is by default used by the \textit{fmincon} function. The same results are obtained using the algorithms: "SQP" as well as "active-set". Additionally, in order to overcome the problem of local minima, another technique called "GlobalSearch" from Matlab was used, which offers the possibility to find the global
minimum. The obtained results are practically the same as those obtained with the \textit{fmincon} function. We notice that the \textit{fmincon} function (in Matlab environment) implements four different algorithms: interior point, SQP, active set, and trust-region-reflective. The obtained results are summarized in Table 2: $-\Delta l_{\text{opt}} \leq \delta l_{i} \leq \Delta l_{\text{opt}}, -\Delta l_{\text{opt}} \leq \delta l_{j} \leq \Delta l_{\text{opt}}, -\Delta l_{\text{opt}} \leq \delta l_{k} \leq \Delta l_{\text{opt}}, \Delta d_{\text{opt}} \leq \delta d_{l} \leq \Delta d_{\text{opt}}$. For the sake of graphic interpretation, the sensitivity ellipses and the corresponding tolerance box unfortunately cannot be represented graphically.

4.1 Optimization of the $\Delta y_{\text{opt}}$

After several tests carried out on all dimensional tolerances of the robot parameters, we found that the dimensional tolerances $\Delta L_{b1}$ and $\Delta L_{b2}$ are the most influential on the sensitivity of the robot (end-effector accuracy). Then, the dimensional parameter that will be targeted by the reduction are $\Delta L_{b1}$ and $\Delta L_{b2}$. Therefore, according to equation (14), the values of dimensional tolerances $\Delta L_{b1}$ and $\Delta L_{b2}$ are calculated as follows:

$$
\begin{align*}
\Delta L_{b1} &= K1 \Delta L_{b1}\text{opt} \\
\Delta L_{b2} &= K1 \Delta L_{b2}\text{opt}
\end{align*}
$$

Such as $\Delta L_{b1}\text{opt}$ corresponds to $u_{1}\text{opt}$ and $\Delta L_{b2}\text{opt}$ corresponds to $u_{2}\text{opt}$.

If we take $K1 = 0.7$, then the values of $\Delta y_{\text{opt}}$ are therefore:

$\Delta L_{b1}\text{opt} = 1.0228 \text{ } \mu m$.

The calculation of the new optimal robust values of dimensional tolerances $\Delta y_{\text{opt}}$ is performed by the following optimization problem:

$$
\begin{align*}
\max_{\Delta y_{\text{opt}} i=1}^{8} u_{i}\text{opt} \\
\text{such that } u_{i}\text{opt}(u_{1}\text{opt}, \ldots, u_{8}\text{opt}) \in \xi \\
u_{1}\text{opt} = 1.0228 \text{ } \mu m \\
u_{2}\text{opt} = 1.0228 \text{ } \mu m \\
u_{i}\text{opt} \geq 0.5 \text{ } \mu m, i = 1, \ldots, 8
\end{align*}
$$

Where $\Delta y_{\text{opt}}$ is the minimum tolerance of the length $y_{i}$. Assuming that $\Delta y_{\text{opt}} = 0.5 \text{ } \mu m$, The solution of the optimization problem converges on the results mentioned in Table 3. The \textit{fmincon} function of the Matlab is used to solve the optimization problem.

According to Fig.8 [34], when the precision of a machined part is required, its manufacturing cost is increased. In our case, we tried to obtain a compromise, in fact, the tolerances obtained guarantee an end-effector position error lower than $10 \mu m$ whatever the configuration of the end-effector in the manipulator workspace in equation (21) in the appendix:

$$
\| \| \delta f \|_{2}^{2} = \delta y^{T} J_{1}^{T} J_{1} \delta y
$$

It is worth to note that the dimensional tolerances obtained are not too tight, making the manufacturing cost minimal.

4.2 Interpretation of results

To validate our design method, two types of dimensional tolerances were reduced. One concerns the most influential parameters ($\Delta L_{b1}$) on the sensitivity of the robot, where these parameters have been reduced from its initial values by 30%. However, the search for the largest tolerance box that does not include defective parts, led to the increase of all other tolerances (The vector $\Delta y_{\text{opt}}$ after the 2nd optimization with decrease of $\Delta L_{b1}$ and $\Delta L_{b2}$ in Table 3). In this case, the mean value of the position errors for the 13 robot postures ($\Delta_{\text{mean}}$) was decreased from its initial value of 13.93%, which implies an improvement in the robot’s accuracy (Fig.9).

The second reduction concerns the least influential parameter on the sensitivity of the robot, i.e. having a non-main error. This parameter, which is the dimensional tolerance $\Delta D_{\text{opt}}$, has been reduced by 30% from its initial value. The search for the largest tolerance box that does not include defective parts, led to the increase of all other tolerances (The vector $\Delta y_{\text{opt}}$ after the 2nd optimization with decrease of $\Delta L_{b1}$ and $\Delta L_{b2}$ in Table 3). However, the mean value of the position errors was increased from its initial value of 10.07%, which caused the decrease of the accuracy (Fig.10 this can be explained by the contribution of the main parameters $\Delta L_{b}$ with the other parameters to the increase of the mean value
Table 2. Optimized Tolerances (\(\mu m\)).

<table>
<thead>
<tr>
<th>(K1)</th>
<th>(\Delta D)</th>
<th>(\Delta L_a)</th>
<th>(\Delta L_b)</th>
<th>(\Delta d)</th>
<th>(\Delta D)</th>
<th>(\Delta L_a)</th>
<th>(\Delta L_b)</th>
<th>(\Delta d)</th>
<th>Percentage increase in vector tolerances ((%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1.2507</td>
<td>-1.2606</td>
<td>1.4611</td>
<td>1.2535</td>
<td>1.2549</td>
<td>1.2426</td>
<td>-1.4611</td>
<td>-1.2563</td>
<td>9.92</td>
</tr>
<tr>
<td>2</td>
<td>1.2507</td>
<td>1.2606</td>
<td>1.4611</td>
<td>1.2535</td>
<td>1.2549</td>
<td>1.2426</td>
<td>-1.4611</td>
<td>-1.2563</td>
<td>-30</td>
</tr>
</tbody>
</table>

Fig. 9. Variation of the position error. Case : decrease in \(\Delta L_{b, opt}\).

Fig. 10. Variation of the position error. Case : decrease in \(\Delta D_{opt}\).

Fig. 11. Effects of dimensional variations \(\delta D_{opt}^+\), \(\delta L_a^+\), \(\delta L_b^+\), and \(\delta d_{opt}^+\) on robot accuracy (according to variation number).

Percentage increase in vector tolerances \(\Delta y_{opt}\) and \(\Delta d_{opt}\) respectively, for the thirteen robot’s postures (Fig. 7) (the variation number represents the iteration number of the position error calculation loop). It can be observed that the position error is always less than or equal to 10 \(\mu m\). This figure shows that the robot is very sensitive to small dimensional variations. Fig.11 shows the robustness of our design. It has a precision that is less tight compared to the precision of the other ellipses.

When \(\sum_{i=1}^3 (\delta D_i + \delta L_a_i + \delta L_b_i + \delta d_i)\) tends to zero, the accuracy is maximum and it is minimal when \(\sum_{i=1}^3 (\delta D_i + \delta L_a_i + \delta L_b_i + \delta d_i)\) is maximum. If the accuracy is minimal, our design is always robust since the position error is always less than or equal to 10 \(\mu m\).
5 Conclusion

In this work, a sequential procedure for modeling, dimensioning and tolerance synthesis of PAR2 parallel manipulator is presented. A robust deterministic method for the analysis and synthesis of mechanism tolerance is used. After the determination of the most influential parameters on the sensitivity of the robot, they are introduced into the calculation of dimensional tolerances. This procedure allowed us to calculate the optimal tolerance box (Brahmia-TB) of the PAR2 parallel manipulator, so the dimensional tolerances are extracted from this box. To plot the sensitivity ellipses, a theory describing the performance sensitivity distribution has been introduced. To calculate the dimensional tolerances of the robot links lengths, an optimization problem is formulated whose objective function deals with maximizing the space of the tolerance box included in the most constraining sensitivity ellipse. The values of the dimensional tolerances found are robust values. However, even if these values are not too tight, they always keep the accuracy under the boundary of 10 μm. The use of the data from this analysis allowed us to show the influence of dimensional tolerances on the performance of the robot. Accuracy is an illustrated parameter to indicate their sensitivity to small dimensional variations.

This design approach allows to increase the dimensional tolerances of 6 geometric parameters of the PAR2 robot (which present 75% of the total number of parameters) compared to the tolerances calculated by Caro-BT, and despite this, the robot’s accuracy has been improved by 13.93%, this can be explained as follows: the position error is very sensitive to the variation of the influential geometric parameters (ΔLb1 and ΔLb2 in our case). Indeed, our design method makes it possible to produce a robust and optimal mechanism that meets the requirements of the specifications (in terms of precision) with a minimum manufacturing price.

6 Acknowledgment

This research was supported by the Directorate General for Scientific Research and Technological Development (DGRSDT) in Algeria.

References

Appendix

To define a robust design problem, three sets need to be defined, namely [10]:

1. The design variables of a mechanism are generally its dimensions (lengths, orientation, etc.) whose nominal values are controllable. These variables are gathered in the vector \( y = [y_1 \ y_2 \ldots y_n]^T \) of dimension \( n \).
2. The design parameters describing the system environment. They cannot be tuned by the designer. These parameters are gathered in the vector \( h = [h_1 \ h_2 \ldots h_l]^T \) of dimension \( l \).
3. The performance functions are grouped in the vector \( f = [f_1 \ f_2 \ldots f_m]^T \) of dimension \( m \).

If we consider that only the effect of the design variables on both production and design of the system, then the variation in performance caused by the variation in the design variables can be expressed by the following linear expression:

\[
\delta f = J_s \delta y
\]

With: \( J_s \) is the Jacobian sensitivity matrix describing the effect of design variables on the system performance. \( \delta y = [\delta y_1 \ \delta y_2 \ldots \delta y_n]^T \) is the vector of variations of the design variables. It should be noted that the random components of \( \delta y \) are independent and the expanded space by these components of \( \delta y \) is of dimension \( n \). This space is called variation space [8]. For the design of a mechanism to be robust, the sensitivity of its performance to variations in the design variables must be minimal. It is to minimize the variations of \( f_i \), i.e the norm of \( \delta f \). The expression of the square of the Euclidean norm of \( \delta f \) is then defined as follows:

\[
\| \delta f \|^2_T = \delta f^T \delta f = \delta y^T J_s^T J_s \delta y
\]

Putting:

\[
S = J_s^T J_s
\]

\( S \) represents the sensitivity matrix is of dimension \( n \times n \). It has \( n \) eigenvectors and \( n \) eigenvalues, its rank is equal to the number of positive eigenvalues. This matrix \( S \) is diagonalizable and can be rewritten as follows:

\[
S = P \text{diag}(\lambda_i) P^T
\]

With \( P = [p_1, \ldots, p_i, \ldots, p_n], \ i \in 1,\ldots, n \) and \( \lambda_i \) is the \( i \)th eigenvalue and \( p_i \) is the eigenvector associated to the eigenvalue \( \lambda_i \). In the space of variations (of dimension \( n \), the
equation that characterizes geometrically the performance sensitivity distribution is given by [10]:

\[ \| \delta f \|_2 = \sqrt{\lambda_1 r_1^2 + \ldots + \lambda_n r_n^2} \]  

(23)

Where \( r = [r_1, \ldots, r_n]^T \) is the projection of the vector of variations of the design variables in the base formed by the column vectors of \( P \), hence:

\[ \delta y = Pr \]  

(24)

If the matrix \( S \) is positive definite, i.e. \( \text{rank}(S) = n, \lambda_i > 0, i = 1, 2, \ldots, n \). Equation (23) represents a family of hyper-ellipsoids of dimension \( n \) and the parameter \( \| \delta f \|_2 \) is called sensitivity hyper-ellipsoids.

Fig. 12 shows the hyper-ellipsoid of sensitivity of a mechanism with three design variables: \( y_1, y_2 \) and \( y_3 \). The points on the surface of the ellipse have the same norm of performance variation \( \| \delta f \|_2 \). The performance is less sensitive to variations in the direction of \( p_1 \) and more sensitive to variations in the direction of \( p_3 \). If the matrix \( S \) is a semi-positive definite matrix, then \( \text{rank}(S) = r < n, \lambda_1 = \lambda_2 = \ldots = \lambda_{n-r} = 0 \) and \( 0 < \lambda_{n-r+1} \leq \lambda_{n-r+2} \leq \ldots \leq \lambda_n \). Equation (24) describes a family of hyper-cylindroids; each cylindroid has \( (n-r) \) infinite principal axes. In practice, we need to reduce the infinite lengths of the main axes of a cylindroid to some reasonable lengths, since the linear relationship between \( \delta f \) and \( \delta y \) in equation (20) is only valid for \( \| \delta y \|_2 \) relatively small. The method for tuning the lengths of the principal axes is as follows [8]: In a new space called feasible space, such as \( S_f = \delta y, \delta y^T S \delta y \leq Y^2_f, Y^2_f \): is the squared sum of the individual performance of the tolerance, hence:

\[ \| \delta f \|_2 = \sqrt{\sum_{i=1}^m \Delta G_i^*} = Y_f^2 \]  

(25)

Such that, \( G_i^*, i = 1, \ldots, n \) are the performance tolerances.

Jianmin [8], proposed a new expression of eigenvalues \( \lambda_i \) as follows:

\[ \hat{\lambda}_i = \max(\lambda_i, \frac{Y^2_r}{K2\|\hat{y}\|_2^2}), i = 1, \ldots, n \]  

(26)

and the \( n \) principal axis lengths are:

\[ a_i = \frac{Y_r}{\sqrt{\hat{\lambda}_i}} \]  

(27)

Where \( K2 \) is a modification coefficient, that can be chosen between 0.03 and 0.05 according to [32, 33].

\[ \| \hat{y} \|_2 = \sqrt{\hat{y}_1^2 + \hat{y}_2^2 + \ldots + \hat{y}_n^2} \]  

(28)

Where \( \hat{y}_i \) denotes the nominal values of design variables. A matrix \( \hat{S} \) can then be reconstructed as follows:

\[ \hat{S} = \hat{P} \hat{D} \hat{P}^T \]  

(29)

With: \( \hat{D} = \text{diag}(\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_n) \), and \( \hat{S} \) is called the characteristic design matrix corresponding to the feasible space.