Multi-objective Optimization of Parallel Manipulators using a Particle Swarm Algorithm

ANTÓNIO M. LOPES
Instituto de Engenharia Mecânica – Pólo FEUP
Faculdade de Engenharia da Universidade do Porto
Rua Dr. Roberto Frias 4200–465 Porto
PORTUGAL
aml@fe.up.pt

HÉLIO FREIRE1, P. B. DE MOURA OLIVEIRA1, E. J. SOLTEIRO PIRES1,2
1Escola de Ciências e Tecnologia da Universidade de Trás-os-Montes e Alto Douro
Quinta de Prados, 5000–911 Vila Real
PORTUGAL
freireh@gmail.com, oliveira@utad.pt, epires@utad.pt

CECÍLIA REIS
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200-072 Porto
PORTUGAL
cmr@isep.ipp.pt

Abstract: Parallel manipulators have attracted the attention of researchers from different areas such as: high-precision robotics, machine-tools, simulators and haptic devices. The choice of a particular structural configuration and its dimensioning is a central issue to the performance of these manipulators. A solution to the dimensioning problem, normally involves the definition of performance criteria as part of an optimization process. In this paper the kinematic design of a 6-dof parallel robotic manipulator is analyzed. Three performance criteria are formulated and optimal solutions are found through a particle swarm formulation.

Key-Words: Parallel manipulators, robotics, synthesis, optimization, particle swarm optimization

1 Introduction
Parallel manipulators are well known for their high dynamic performances and low positioning errors. In the last few years parallel manipulators have attracted great attention from researchers involved with robot manipulators, robotic end effectors, robotic devices for high-precision tasks, machine-tools, simulators, and haptic devices.

One of the most important factors affecting the performance of a robotic manipulator is its structural configuration. The kinematics, statics, dynamics and stiffness are all dependent upon it. Following the definition of a particular structural configuration, the next step in the manipulator design is its dimensioning. Usually this task involves the choice of a set of parameters that define the mechanical structure of the manipulator. The parameter values should be chosen to optimize some performance criteria, dependent upon the foreseen application [1].

In the last years most of the research in parallel manipulators optimization has been done over several criteria related to the optimization of the manipulator workspace [2]. Other authors choose to optimize the structural stiffness of the manipulator, as this is one of the main advantages of a parallel configuration over a serial one [3]. Finally, some works may be referred where the optimization criteria are related with the manipulability, or dexterity, of the manipulator [4].

Optimization can be a difficult and time-consuming task, because of the great number of optimization parameters and the complexity of the objective functions. However, optimization procedures based on evolutionary approaches have been proved as an effective way out [5].
In this paper the kinematic design of a 6-dof parallel robotic manipulator is analyzed. Three performance criteria are formulated and optimal solutions are found through a particle swarm algorithm formulation.

2 Manipulator structure/kinematics
The mechanical structure of the parallel robot comprises a fixed (base) platform and a moving (payload) platform, linked together by six independent, identical, open kinematic chains (Fig. 1). Each chain comprises two links: the first link (linear actuator) is always normal to the base and has a variable length, \( l_i \), with one of its ends fixed to the base and the other one attached, by a universal joint, to the second link; the second link (arm) has a fixed length, \( L \), and is attached to the payload platform by a spherical joint. Points \( B_i \) and \( P_i \) are the connecting points to the base and payload platforms. They are located at the vertices of two semi-regular hexagons, inscribed in circumferences of radius \( r_B \) and \( r_P \), that are coplanar with the base and payload platforms (Fig. 2).

For kinematic modeling purposes a right-handed reference frame \{B\} is attached to the base. Its origin is located at point \( B \), the centroid of the base. Axis \( x_B \) is normal to the line connecting points \( B_1 \) and \( B_6 \) and axis \( z_B \) is normal to the base, pointing towards the payload platform. The angles between points \( B_1 \) and \( B_3 \) and points \( B_1 \) and \( B_5 \) are set to 120º. The separation angles between points \( B_1 \) and \( B_6 \), \( B_2 \) and \( B_3 \), \( B_4 \) and \( B_5 \), and \( B_3 \) and \( B_4 \) are denoted by \( 2\phi_B \) (Fig. 2).

In a similar way, a right-handed frame \{P\} is assigned to the payload platform. Its origin is located at point \( P \). Axis \( x_P \) is normal to the line connecting points \( P_1 \) and \( P_6 \) and axis \( z_P \) is normal to the payload platform, pointing in a direction opposite to the base. The angles between points \( P_1 \) and \( P_3 \) and points \( P_3 \) and \( P_5 \) are set to 120º. The separation angles between points \( P_1 \) and \( P_2 \), \( P_4 \) and \( P_6 \), and \( P_3 \) and \( P_5 \) are denoted by \( 2\phi_P \) (Fig. 2).

Taking into account the definitions given above, the generalized position of frame \{P\} relative to frame \{B\} may be represented by the vector:

\[
^{b}_p x_{P_{(p)}} = \begin{bmatrix}
x_p \\
y_p \\
z_p \\
\psi_p \\
\theta_p \\
\phi_p 
\end{bmatrix} = \begin{bmatrix}
^{b}_p x_P \\
\phi_B \\
^{b}_p x_{P_{(p)}}
\end{bmatrix}
\]

where \(^b x_{P_{(p)}} \) is the position of the origin of frame \{P\} relative to frame \{B\}, and \(^b x_{P_{(p)}} \) defines an Euler angles system representing the orientation of frame \{P\} relative to \{B\}. Vector \(^p p_i = \begin{bmatrix}
p_{ix} \\
p_{iy} \\
p_{iz} \\
p_{io} \\
p_{ip} \\
p_{iu}
\end{bmatrix} \) represents the position of point \( P_i \) with reference to frame \{P\}, and vector \( b_i = \begin{bmatrix}
b_{ix} \\
b_{iy} \\
b_{iz} 
\end{bmatrix} \) represents the position of point \( B_i \) with reference to frame \{B\}.

2.1 Inverse Position Kinematics
The inverse position kinematic model is used to compute the joints positions for a given manipulator Cartesian position and orientation. The presented model follows the one proposed in [6].

Taking into account a single kinematic chain \( i \), vector \(^p p_i \) may be written in the base frame using the following transformation:

\[
^{b}_p p_i = R_{p_i}^{b} p_i = \begin{bmatrix}
r_{i1}^{p} p_{ix} + r_{i2}^{p} p_{iy} + r_{i3}^{p} p_{iz} \\
r_{i2}^{p} p_{ix} + r_{i3}^{p} p_{iy} + r_{i2}^{p} p_{iz} \\
r_{i3}^{p} p_{ix} + r_{i2}^{p} p_{iy} + r_{i3}^{p} p_{iz}
\end{bmatrix}
\]
where $^a\mathbf{R}_p$ is a matrix representing the orientation of the payload platform frame with reference to the base frame, that may be computed from the Euler angles ($\psi_p$, $\theta_p$, $\phi_p$).

Fig. 3 Schematic representation of a kinematic chain

Subtracting vectors $^a\mathbf{x}_{p,\text{rpo}}$ and $b_i$, then vector $\mathbf{s}_i = [s_x, s_y, s_z]^T$ is obtained. If $s_i$ and $^p\mathbf{p}_{l_b}$ are added, the vector $\mathbf{e}_i = [e_x, e_y, e_z]^T$ is obtained, that is:

$$\mathbf{e}_i = ^a\mathbf{x}_{p,\text{rpo}} - b_i + ^p\mathbf{p}_{l_b} = \mathbf{s}_i + ^p\mathbf{p}_{l_b}$$

$$\begin{bmatrix}
x_p - b_x \\
y_p - b_y \\
z_p - b_z
\end{bmatrix} + \begin{bmatrix}
r_{11}^p & r_{12}^p & r_{13}^p & p_{0x}^p \\
r_{12}^p & r_{22}^p & r_{23}^p & p_{0y}^p \\
r_{13}^p & r_{23}^p & r_{33}^p & p_{0z}^p
\end{bmatrix}
\begin{bmatrix}
x_i \\
y_i \\
z_i
\end{bmatrix} = \begin{bmatrix}
x_{p} - b_x \\
y_{p} - b_y \\
z_{p} - b_z
\end{bmatrix}$$

Vector $\mathbf{a}_i$, aligned with the fixed length arm, is given by $\mathbf{a}_i = \mathbf{e}_i - \mathbf{d}_i + \mathbf{d}_i$, is a vector parallel to $\mathbf{z}_b$, and length $l_i$ (Fig. 3).

Knowing that the 2-norm of $\mathbf{a}_i$ is equal to the arm length, $L$, it follows that:

$$\|\mathbf{a}_i\|_2 = \|\mathbf{e}_i - \mathbf{d}_i\|_2 = L$$

$$\sqrt{e_x^2 + e_y^2 + (e_z - l_i)^2} = L$$

Solving for $l_i$ results in

$$l_i = e_z \pm \sqrt{L^2 - e_x^2 - e_y^2}$$

that is, there are two possible solutions for $l_i$. The solutions corresponding to the manipulator having the universal joints below the payload platform spherical joints are always considered:

$$l_i = e_z + \sqrt{L^2 - e_x^2 - e_y^2}$$

### 2.2 Inverse Velocity Kinematics

Inverse velocity kinematics can be represented by the inverse kinematic jacobian, relating the joints velocities to the manipulator Cartesian-space velocities (linear and angular):

$$\dot{\mathbf{i}} = \mathbf{J}_c \; ^a\mathbf{x}_{p,\text{rpo}}$$

Vector $\dot{\mathbf{i}} = [\dot{i}_1, \dot{i}_2, \ldots, \dot{i}_n]^T$ represents the joints velocities, and vector $^a\mathbf{x}_{p,\text{rpo}} = [^a\mathbf{x}_{p,\text{rpo}}, ^a\mathbf{o}_{p,\text{rpo}}]_b$ represents the Cartesian-space velocities.

The velocity of point $P_i$ is dependent upon the linear and angular velocities of the payload platform. If $^a\mathbf{v}_{p,\text{rpo}}$ denotes that velocity with reference to the base frame (and written in that same frame), then:

$$^a\mathbf{v}_{p,\text{rpo}} = ^a\mathbf{p}_{l_b} + ^a\mathbf{o}_{p,\text{rpo}} \times ^p\mathbf{p}_{l_b}$$

where $^a\mathbf{v}_{p,\text{rpo}}$ and $^a\mathbf{o}_{p,\text{rpo}}$ represent the linear and angular velocities of the payload platform frame with reference to, and written in, the base frame.

Squaring both sides of (4), the following relation is obtained:

$$\mathbf{a}_i^T \mathbf{a}_i = \mathbf{d}_i^T \mathbf{d}_i + \mathbf{e}_i^T \mathbf{e}_i - 2\mathbf{d}_i^T \mathbf{e}_i$$

Differentiating (10), the following expression results:

$$\dot{l}_i + \mathbf{e}_i^T \dot{\mathbf{e}}_i - l_i \mathbf{z}_b^T \mathbf{e}_i - l_i \mathbf{z}_b^T \dot{\mathbf{e}}_i = 0$$

where $\mathbf{z}_b = [0 \ 0 \ 1]^T$ denotes the direction of displacement of the linear actuators.

From (11), and taking into account that $\dot{\mathbf{e}}_i = \dot{\mathbf{p}}_i$, an expression for the linear actuators velocity, $\dot{l}_i$, is obtained:

$$\dot{l}_i = \frac{(\mathbf{e}_i - l_i \mathbf{z}_b)^T}{l_i \mathbf{z}_b^T \mathbf{e}_i - l_i} \times \frac{\left(^a\mathbf{p}_{l_b} + \mathbf{e}_i - l_i \mathbf{z}_b\frac{\mathbf{z}_b^T \mathbf{e}_i - l_i}{l_i \mathbf{z}_b^T \mathbf{e}_i - l_i}\right) \times ^a\mathbf{o}_{p,\text{rpo}}}{L}$$

Following this result, the inverse kinematic jacobian may be written (with reference to the base frame) as:

$$\mathbf{J}_c = \begin{bmatrix}
(\mathbf{e}_i - l_i \mathbf{z}_b)^T \\
(\mathbf{e}_i - l_i \mathbf{z}_b) \\
(\mathbf{e}_i - l_i \mathbf{z}_b)^T \\
\mathbf{z}_b^T \mathbf{e}_i - l_i
\end{bmatrix} \begin{bmatrix}
(\mathbf{p}_{l_b} + \mathbf{e}_i - l_i \mathbf{z}_b)^T \\
(\mathbf{p}_{l_b} + \mathbf{e}_i - l_i \mathbf{z}_b) \\
(\mathbf{p}_{l_b} + \mathbf{e}_i - l_i \mathbf{z}_b)^T \\
\mathbf{z}_b^T \mathbf{e}_i - l_i
\end{bmatrix}$$

### 3 Objective functions

Several performance indexes can be considered for the optimization problem, most of them being based on the manipulator inverse kinematic jacobian [1]. In this work a multi-criteria optimization problem is formulated. Accordingly, the manipulator will be designed for maximum dexterity, maximum stiffness and maximum stiffness isotropy. All the criteria may be mathematically expressed using a function of the kinematic jacobian. In order to obtain a dimensionally homogeneous inverse jacobian matrix, the manipulator payload platform radius, $r_p$, is used.
as a characteristic length. Thus, the same ‘cost’ will be associated to translational and rotational movements. The inverse kinematic jacobian results dependent upon ten variables, four of them are manipulator kinematic parameters: the position and orientation of the payload platform; the base radius \((r_B)\); the separation angles on the payload platform \((\phi_i)\); and the arm length \((L)\). We will consider a particular manipulator pose, corresponding to the centre of the manipulator workspace i.e., \([0\ 0\ 2\ 0\ 0\ 0]^T\) (units in \(r_P\) and degrees, respectively). Thus, for this pose, the inverse jacobian matrix will be a function of the four kinematic parameters.

### 3.1 Dexterity Index

Mathematically, the condition number of the jacobian matrix, \(J_C\), is used a measure of dexterity. Since \(J_C\) is configuration-dependent, the condition number will also be, and may take values between unity (isotropic configuration) and infinity (singular configuration). Isotropic configurations correspond to mechanical structures and poses (positions and orientations) in which the manipulator requires equal joint effort to move or produce forces and/or torques in each direction. Ideally, the manipulator should be isotropic in its whole workspace. The minimization of the condition number leads not only to the maximization of the manipulator dexterity, but also to the minimization of the error propagation due to actuators, feedback transducers and, when \(J_C\) matrix is inverted, numerically induced noise. The condition number is given by

\[
\kappa = \frac{\sigma_{\text{max}}(J_C)}{\sigma_{\text{min}}(J_C)}
\]

(14)

where \(\sigma_{\text{max}}(J_C)\) and \(\sigma_{\text{min}}(J_C)\) represent the maximum and minimum singular values of matrix \(J_C\). This function should be minimized.

### 3.2 Stiffness Index

Stiffness is an important characteristic in many applications, as it affects the precision of the robot. High stiffness typically allows higher operating speeds and lower vibrations of the mechanical structure. Stiffness may be regarded as a measure of the ability of the robot to resist deformation due to the action of external forces. At any point of the robot workspace it can be characterized by the stiffness matrix, which relates Cartesian forces and torques applied on the end-effector to the corresponding linear and angular Cartesian displacements.

Starting with (8) we may write

\[
\Delta l \approx J_C \cdot \Delta x_P \mid_{l_B}
\]

(15)

where \(\Delta l\) represents an infinitesimal joints (actuators) displacement and \(\Delta x_P \mid_{l_B}\) represents the corresponding Cartesian-space infinitesimal displacement. On the other hand, using the duality between differential kinematics and static results in

\[
^p f \mid_{l_B} = J_C^T \cdot \tau
\]

(16)

where \(^p f \mid_{l_B}\) represents the force and torque applied to the payload platform, and \(\tau\) represents the joints forces.

Considering that each manipulator joint has a finite stiffness given by \(k_i\) then the joint force, \(\tau_i\), and displacement, \(\Delta l_i\), are related by \(\tau_i = k_i \cdot \Delta l_i\), according to Hooke’s law. If all joints are identical, we may write:

\[
\tau = K \cdot \Delta l
\]

(17)

with \(K = \text{diag}(k_1, k_2, \ldots, k_n)\) representing the joint-space stiffness matrix.

Using (15) to (17), results in:

\[
^p f \mid_{l_B} = J_C^T \cdot K \cdot J_C \cdot \Delta x_P \mid_{l_B}
\]

(18)

\[K \mid_{l_B} = J_C^T \cdot K \cdot J_C\]

(19)

Moreover, the diagonal elements of the Cartesian-space stiffness matrix are used as the system stiffness value. These elements represent the pure stiffness in each direction. Mathematically, the objective function used to maximize the manipulator stiffness is given by the trace of matrix \(K \mid_{l_B}\). This function should be maximized.

\[
\nu = \text{trace}(k \cdot J_C^T \cdot J_C)
\]

(21)

### 3.3 Stiffness Isotropy Index

The parallel manipulator is considered isotropic with respect to stiffness if it presents the same stiffness in every Cartesian direction. This means the diagonal elements of matrix \(K \mid_{l_B}\) are equal. Mathematically, the objective function used to maximize stiffness
isotropy is given by the standard deviation of the diagonal elements of matrix $\mathbf{K}_b$. The function to be minimized is:

$$
\sigma = \text{std} \left( \mathbf{diag} \left( k \cdot \mathbf{J}_c^T \cdot \mathbf{J}_c \right) \right)
$$

(22)

4 Multi-objective particle swarm optimization

A standard MOPSO follows the algorithm presented in Fig. 4. The swarm, $X(t)$, can be initialized in several ways using a priori knowledge or totally randomly. The MaxiMin MOPSO reported here uses an adaptation of the Non-dominated Sorting GA, (NSGA-II) technique proposed by Deb et al. [7]. This well known technique uses an archive, $A(t)$, in which the best non-dominated swarm solutions are kept in every iteration, according to a sorted ranking scheme. In each iteration, $t$, the swarm $X(t)$, and archive $A(t)$, are merged into a set $R(t)$. The solutions are evaluated in terms of Pareto domination and sorted according to a non-dominated ranking scheme. The new swarm, $X(t+1)$ is obtained selecting solutions from $R(t)$, using both the non-dominated ranking as well as its diversity evaluated using an Euclidean distance metric (more detailed information about NSGA-II can be found in [8]).

The selection of solutions from $R(t)$ as new swarm members is performed using a technique called MaxiMin proposed by Solteiro Pires et al. [9] in the context of a multi-objective genetic algorithm. The overall objective of MaxiMin is to achieve a non-dominated Pareto front as wide as possible and uniformly distributed. More information about MaxMin can be found in [9].

While in the uni-objective PSO, the global best particle of the swarm (or neighborhood) is trivial to find and can be unique for all the swarm, this is not the case in the MOPSO.

**Fig. 4. Standard MOPSO algorithm**

$$
\begin{align*}
    t & = 0 \\
    \text{initialize swarm } X(t) & \\
    \text{evaluate } X(t) & \\
    \text{generate archive } A(t) & \\
    \text{while} & (\text{termination criterion}) \\
    & t = t+1 \\
    & \text{select guiding particles} \\
    & \text{update particles velocity} \\
    & \text{update particles position} \\
    & \text{generate } X(t+1) \\
    & \text{evaluate } X(t+1) \\
    & \text{update } X(t) \\
    & \text{update } A(t)
\end{align*}
$$

In the MaxiMin MOPSO used here [10] each global archive particle can be selected by a particle as its global guide a predefined maximum number of times. For instance if the swarm size is 100 and the archive size is 50 the maximum number of times that each archive member can be selected as guide is defined as 2. This methodology has the advantage to force some particles to select as global guiding particles which are far away than others already selected by neighbors.

**Fig. 5. Selection of the global guide particle**

In the example shown in Fig. 5, applying this technique will force three of the particles to choose the three guides which are further away in the Pareto front. This motivates the exploration of the search space in order to achieve a wide Pareto front and with the MaxiMin algorithm, uniformly spread.

5 Simulation results

This section develops several experiments with three objectives explained $\kappa$, $\upsilon$ and $s$. Initially, the optimization is performed using two objectives at a time ($\kappa \times \upsilon$, $\kappa \times s$ and $\upsilon \times s$). Afterwards, an optimization is executed considering the three objectives simultaneously. The MOPSO particles have four parameters $\{r_B, \phi_P, \phi_B, L\}$. These parameters are allowed to change in the range $r_B \in [1, 2.5]$, $\phi_P \in [0, \pi/9]$, $\phi_B \in [0, \pi /9]$ and $L \in [2, 4.5]$. The MOPSO is executed during 500 iterations.

**Fig. 6. Pareto optimal front for $\kappa \times \upsilon$ optimization**
Fig. 7. Pareto optimal front for $\kappa \times s$ optimization

Fig. 6 to Fig. 9 show the Pareto Fronts obtained for the experiments. Fig. 6 shows the $\kappa \times \upsilon$ optimization front. The front is basically achieved by varying the parameter $r_B$ in the range $[1.771, 2.303]$. The other parameters remain constant: $\phi_P = 0, \phi_B = 0$ and $L = 2.067$. A different result was observed for $\kappa \times s$ optimization, as shown in Fig. 7. Here the parameters vary between $\{1.813, 0, 0, 2.000\}$ and $\{2.500, 0.349, 0.349, 3.230\}$ for $\{r_B, \phi_P, \phi_B, L\}$.

Optimization figures reveal that objectives are quarrelsome and solution diversity along the Pareto front is presented in all experiments.

6 Conclusions

In this paper the kinematic design of a 6-dof parallel robotic manipulator was analyzed. First, a MOPSO was used to solve the optimization problem using up to three objective functions. The objectives considered in the design of parallel robot were the dexterity, stiffness and stiffness isotropy index. The MOPSO converged to a front characterized by multiple sets of optimal alternative solutions. Moreover, the proposed MOPSO provides, in one run, a representative solution set of the Pareto front.

References: