Safa El Hraiech ✉
Youssef Chouaibi
Zouhaier Affi

https://doi.org/10.21278/TOF.503082825
ISSN 1333-1124
eISSN 1849-1391

A DYNAMIC MULTI-OBJECTIVE APPROACH USING INTERVAL ANALYSIS AND GENETIC ALGORITHMS

Summary

This paper considers a multi-objective optimisation problem focusing on translational parallel manipulators. The proposed approach combines interval analysis with genetic algorithms to optimise the dynamic parameters of parallel manipulators. The objectives are to enhance the robot accuracy, while maximising the tolerance intervals of the parameters. This paper's contribution is to introduce an interval method to estimate the error bounds of a dynamic parallel manipulator within the desired workspace. A genetic algorithm is then applied to improve manipulator accuracy and reduce design costs. The resulting Pareto fronts illustrate the trade-off between each pair of objective functions.

Key words: *parallel manipulator; interval analysis; genetic algorithm; optimisation; uncertainty.*

1. Introduction

Optimisation is a core process in engineering and decision making, where the aim is to determine the optimal solution that satisfies a set of constraints and objectives considered simultaneously. This type of problem is called multi-objective optimisation (MOP). This form of optimisation seeks to identify a set of optimal solutions that achieve a balance among multiple conflicting objectives within a given problem. In the context of parallel robots, an optimisation algorithm can be applied to determine the optimal configuration of the robot that satisfies multiple performance criteria such as accuracy [1], speed [2], stability [3], [4] and stiffness [5]. A coupled algorithm refers to an approach where multiple algorithms are integrated and work together to solve the optimisation problem. In the case of parallel robots, the coupled algorithm can involve combining kinematic and dynamic analysis to find the optimal solution of the robot structure [6]. These various performances can be optimised using particle swarm optimisation. Zhao and Zhang applied particle swarm optimisation to analyse the accuracy of a parallel manipulator [7]. Kim and Lee used particle swarm optimisation to propose an algorithm that optimises the parameters of a parallel manipulator's trajectory [8]. This algorithm has proven its effectiveness and convergence in the presence of constraints. Other popular evolutionary algorithms that can be used for multi-objective problems (MOPs) include multi-objective differential evolution (MODE) [9] and multi-objective artificial bee colony (MOABC) [10]. Recent work has proposed frameworks combining CAD/CAE-based kinematic and elasto-dynamic modelling with multi-objective optimisation to enhance both the stiffness and dynamic performance of parallel robots [11].

Stan et al. performed kinematic optimisation using genetic algorithms to maximise the workspace of a 6-degree-of-freedom micro parallel robot [12]. Multi-objective optimisation using genetic algorithms (GA) is a widely used approach for solving complex optimisation problems that concurrently optimise several conflicting objectives. In the context of parallel robots, GA can be used to optimise robot design parameters such as link lengths, joint angles, and other geometric parameters. This optimisation can be conducted for multiple objectives such as workspace, dexterity [13], stiffness [14], and accuracy [15]. Using the derived dynamic model, Wu et al. developed a multi-objective optimisation problem to optimise the geometric and structured parameters of a spherical parallel manipulator [16].

Recently, the dynamic model of a 3-UPU translational parallel robot was derived analytically and the position error was estimated using interval analysis [17]. Later, El Hraiech et al. introduced a GA–Krawczyk method for multi-objective optimisation. In that work, only the robot position error is evaluated. In addition, the study adjusted the parameter tolerance intervals based on a sensitivity analysis, and the resulting Pareto fronts were compared with those obtained using a uniform distribution of parameter uncertainties [18].

The contribution of this paper is the extension of existing interval-based optimisation approaches by evaluating not only the position error, but also the velocity and dynamic errors of a translational parallel manipulator. Optimised and non-optimised mechanisms are compared. A trajectory test is performed, and the resulting improvements in geometric, kinematic, and dynamic accuracy are quantified. The remainder of the paper is organised as follows: In Section 2, the interval analysis method based on the Krawczyk operator is used to estimate manipulator accuracies. In Section 3, the genetic algorithm is combined with the Krawczyk method to determine a set of parameter vectors along with their corresponding tolerance intervals, optimising both the manipulator's accuracy and the design cost. Section 4 presents the simulation results. The Pareto fronts are presented as functions of the various objective functions. Some solutions from these fronts are evaluated and compared to the non-optimised mechanism under an imposed test trajectory to confirm the effectiveness of the proposed approach.

2. Position, velocity, and actuator force errors of TPM using interval analysis

The 3-UPU is taken as an example of translational parallel manipulators (Fig. 1). Under a specific arrangement of the joints' axis, the robot performs exclusively translational movement (geometric conditions). Since the manipulator has a symmetrical structure, the main parameters of the 3-UPU manipulator can be summarised as the base and platform radii r_b, and r_p. For each actuator i, the parameters considered include its angular position α_i, length l_i, linear velocity $\dot{l}_i$, and acceleration $\ddot{l}_i$, as well as the positions of the centres of mass of its lower and upper parts r_{di} and r_{si} and the corresponding masses m_{di} and m_{si}. In addition, the mass of the platform m_p is taken into consideration (where i=1, 2, 3). The kinematic and the dynamic models of the proposed manipulator were developed in the authors' previous work [17].

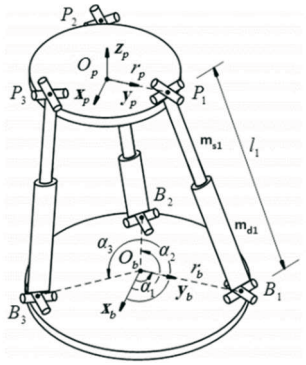

Fig. 1 Structure of the 3-UPU TPM

2.1 The interval method

Interval analysis is used to estimate the manipulator error bounds. The Krawczyk operator interval method is a numerical technique used in interval analysis and computational mathematics. The goal is to find the interval solution of the equation:

$$\mathbf{f}(\mathbf{p},\mathbf{v})=\mathbf{0}. \tag{1}$$

Here, $\mathbf{f}(\mathbf{p},\mathbf{v})$ is a function that relies on the variable vector $\mathbf{v}$ to the parameter vector $\mathbf{p}$. The uncertainties in the parameters are represented by an interval vector $[\mathbf{p}]$. The objective is to determine an interval that encloses all possible variations of the solutions resulting from these parameter uncertainties The proposed algorithm used to predict the error bounds is detailed in Fig. 2:

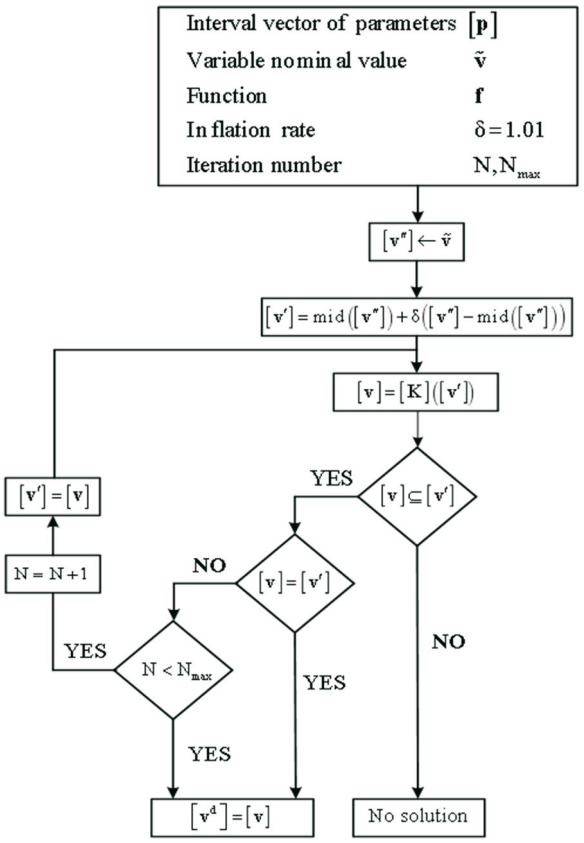

Fig. 2 The proposed Krawczyk algorithm [17]

The Krawczyk method is used to determine an interval box $[\mathbf{v}]$ that contains all possible solutions corresponding to the uncertainties in the design parameters $[\mathbf{p}]$. Initially, the algorithm requires the definition of its inputs, which include the nominal solution and the uncertainties of the design parameters. Using a Newton approach, an initial interval vector $[\mathbf{v}'']$ is constructed. This vector is then expanded through an inflation process to obtain a refined interval vector $[\mathbf{v}']$, which serves as the starting point for the Krawczyk iteration. Next, the Krawczyk operator is applied, using $[\mathbf{v}']$and $[\mathbf{p}]$ as inputs. The operator is evaluated iteratively: at each iteration, the resulting interval $[\mathbf{K}]([\mathbf{v}'])$ is compared with $[\mathbf{v}']$. The iterations continue until $[\mathbf{K}]([\mathbf{v}'])$ is fully contained within $[\mathbf{v}']$, indicating convergence, or until a predefined maximum number of iterations is reached.

2.2 Prediction of the manipulator's bounds of error

To estimate the manipulator's bounds of error, the structural and inertial parameters are considered. The inputs comprise the nominal parameter values along with their associated tolerances, as well as the nominal vectors of position, velocity, and acceleration.

After developing the inverse geometric, kinematic, and dynamic models, the prediction algorithm yields different error bounds for position, kinematic, and actuator force errors as its output.

The details of the prediction algorithm are outlined in [17]. The algorithm was implemented using the INTLAB library in Matlab. The nominal values of the structural and inertial parameters are provided in Table 1. The uncertainty associated with these parameters is set at 0.2% of their nominal values.

Table 1 The nominal values of the structural and inertial parameters

Structural parameters							Inertial parameters		
r_b (m)	r_p (m)	r_s (m)	r_d (m)	α_1 (rad)	α_2 (rad)	α_3 (rad)	M_p (kg)	m_s (kg)	m_d (kg)
0.3	0.05	0.2	0.2	0	$2\pi/3$	$4\pi/3$	3	1	1

The proposed workspace (free of singularities) is defined by:

$$-0.15 \le x\,(\mathrm{m}) \le 0.15, -0.15 \le y\,(\mathrm{m}) \le 0.15, 0.3 \le z\,(\mathrm{m}) \le 0.6 \tag{2}$$

3. Simulation results

In this section, the accuracies of the 3-UPU translational manipulator are assessed. Specifically, the errors in position, velocity and actuator force are analysed across various sections of the workspace, taking into account uncertainties in the structural and inertial parameters. Constant values are assumed for the platform's velocity and acceleration, set to $v = 50$ mm.s^{-1} and $a = 50$ mm.s^{-2}, respectively.
The uncertainties in the actuator length, velocity, and acceleration are assumed to be proportional to their nominal values:

$$\frac{\Delta l_i}{l_i} = 2.10^{-3}, \frac{\Delta \dot{l}_i}{\dot{l}_i} = 2.10^{-3}, \frac{\Delta \ddot{l}_i}{\ddot{l}_i} = 2.10^{-3} \tag{3}$$

where actuator lengths, velocities, and accelerations are computed using the inverse geometric and kinematic models. For a workspace section at z=600 mm, the errors in position, velocity, and actuator force are analysed and their distributions are presented.

3.1 Distribution of the robot's position errors

The position error of the proposed robot is expressed as follows:

$$E_p = \left(\sqrt{E_{px}^2 + E_{py}^2 + E_{pz}^2}\right), \tag{4}$$

where

$$\begin{aligned} E_{px} &= \mathrm{Sup}\left(\left|x_{max} - x\right|, \left|x_{min} - x\right|\right) \\ E_{py} &= \mathrm{Sup}\left(\left|y_{max} - y\right|, \left|y_{min} - y\right|\right) \\ E_{pz} &= \mathrm{Sup}\left(\left|z_{max} - z\right|, \left|z_{min} - z\right|\right), \end{aligned} \tag{5}$$

and where the interval vector $[\mathbf{x}]$ is the result of the Krawczyk algorithm, given as:

$$[\mathbf{x}] = \begin{bmatrix} x_{min} & x_{max} \\ y_{min} & y_{max} \\ z_{min} & z_{max} \end{bmatrix} \tag{6}$$

E_p represents the maximum error value along the **x-**, **y-**, and **z-**axes, where $[x,y,z]$ is the position vector of the robot's platform.

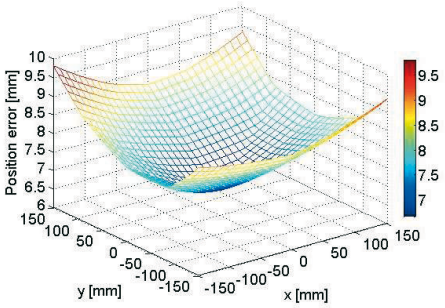

Fig. 3 Geometric error distribution of the robot

A workspace section at z=600 mm is used to illustrate the influence of parameter uncertainties in the geometric error distribution. Fig. 3 shows that the geometric accuracy is minimal at the centre and reaches its maximum value of 9.799 mm in the extreme zones (corresponding to x=150 mm, y=150 mm, z=600 mm).

3.2 Distribution of the robot's velocity errors

Following the same approach, the velocity error is calculated as:

$$\dot{E}_p = \left(\sqrt{\dot{E}_{px}^2 + \dot{E}_{py}^2 + \dot{E}_{pz}^2} \right), \tag{7}$$

where

$$\begin{aligned} \dot{E}_{px} &= \mathrm{Sup}\left(\left| \dot{x}_{max} - \dot{x} \right|, \left| \dot{x}_{min} - \dot{x} \right| \right), \\ \dot{E}_{py} &= \mathrm{Sup}\left(\left| \dot{y}_{max} - \dot{y} \right|, \left| \dot{y}_{min} - \dot{y} \right| \right), \\ \dot{E}_{pz} &= \mathrm{Sup}\left(\left| \dot{z}_{max} - \dot{z} \right|, \left| \dot{z}_{min} - \dot{z} \right| \right), \end{aligned} \tag{8}$$

and where $[\dot{\mathbf{x}}]$ is given as the result of the Krawczyk algorithm, defined as:

$$[\dot{\mathbf{x}}] = \begin{bmatrix} \dot{x}_{min} & \dot{x}_{max} \\ \dot{y}_{min} & \dot{y}_{max} \\ \dot{z}_{min} & \dot{z}_{max} \end{bmatrix}. \tag{9}$$

The nominal velocity vector of the robot's platform is chosen as [0, 0, 0.50 mm.s^{-1}].

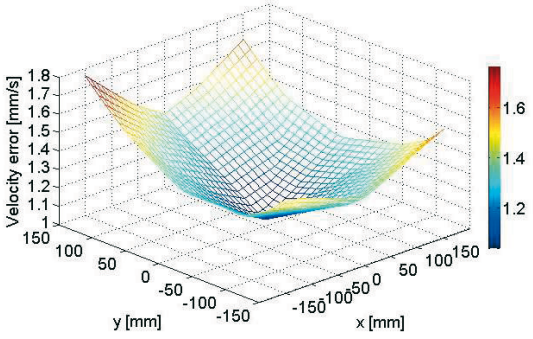

Fig. 4 Velocity error distribution of the robot

Considering the uncertainties in the structural parameters (Δr_b, Δr_p, $\Delta\alpha_i$), as well as in the actuator lengths and velocities, the predicted velocity error is illustrated in Fig. 4. The velocity error reaches its highest value of 1.73 mm.s^{-1} within the extreme zone.

3.3 Distribution of force errors in the robot's actuators

In this section, the dynamic accuracy of the proposed robot is evaluated. The uncertainties in the structural and inertial parameters are taken into account (Δr_b, Δr_p, $\Delta\alpha_i$, Δr_{si}, Δr_{di} ,Δm_{si}, Δm_{di}). The dynamic uncertainties of the actuators are considered $[l_i, \dot{l}_i, \ddot{l}_i]$. The force error of each length can be evaluated separately as follow:

$$\Delta d_i = \mathrm{Sup}\left(\left|d_{i\max} - \tilde{d}_i\right|, \left|d_{i\min} - \tilde{d}_i\right|\right), \tag{10}$$

where the interval vector [**d**] is the result of the dynamic prediction algorithm given as:

$$[\mathbf{d}] = \begin{bmatrix} d_{1\min} & d_{1\max} \\ d_{2\min} & d_{2\max} \\ d_{3\min} & d_{3\max} \end{bmatrix}, \tag{11}$$

where $\tilde{d}_i$ represents the nominal force of the i-th actuator, obtained from the manipulator's inverse dynamic model.

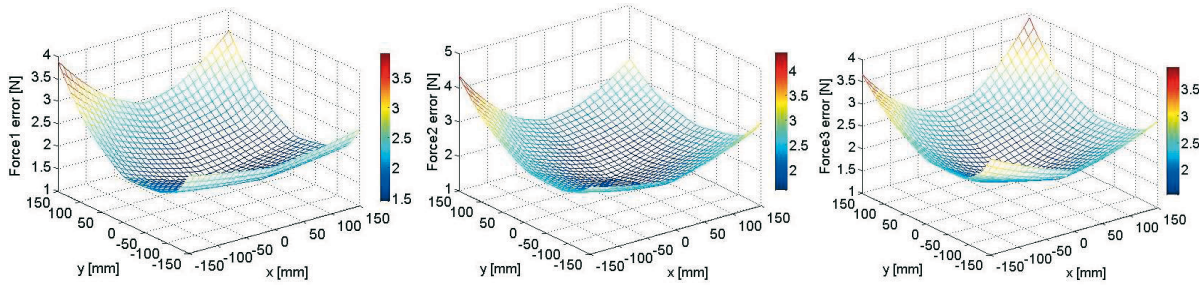

Fig. 5 Distribution of force errors in the actuators

Due to the proportional uncertainties in the parameters (0.2% of the nominal values), the accuracies of the manipulator cannot be tolerated for a parallel manipulator. The position error reaches 9.7 mm (Fig. 3), the velocity error can reach 1.73 mm·s^{-1} in the extreme zones (Fig. 4), and the actuator force errors are seen to reach 3.5 N (Fig. 5).

The robot's precision must be improved. Therefore, alternative parameter uncertainty values need to be considered to guarantee better accuracy and to avoid choosing parameters with very tight intervals (which incurs high costs). An optimisation procedure must be developed while respecting the geometric constraints and the designer's cost limits.

4. Multi-objective optimisation using a genetic algorithm

Faced with a multi-objective optimisation problem, several competing objectives must be simultaneously addressed. These objectives often conflict, so that advancing in one may be to the detriment of another. The proposed method employs MATLAB's Gamultiobj function to compute Pareto-optimal solutions. In this framework, a genetic algorithm (GA) generates an initial population, evaluates fitness, and iteratively applies selection, crossover, and mutation until convergence or a predefined generation limit is reached [18].

The methodology, illustrated in Fig. 6, is implemented using MATLAB software. Based on experimental tuning and literature recommendations, the parameters of the genetic algorithm

are set as follow: the population size and the maximum number of generation are both 100, the crossover probability is 0.8 and the mutation probability is 0.2.

Given the large number of nominal parameters and uncertainties, only the most influential ones are retained. Sensitivity analysis by El Hraiech et al. [17] shows that actuator uncertainties dominate, contributing over 70% to position, kinematic, and force errors.

In the parameter's vector, the actuator lengths are considered as a search interval, while the less effective parameters are kept to their nominal values (actuator velocity and acceleration). Moreover, the angular configuration of the legs is fixed to avoid singularities.

The design parameter vector consists of nominal values and their corresponding tolerance intervals are given by:

$$[\mathbf{p}] = \left[r_{\mathrm{b}}\ {}^{+\Delta r_{\mathrm{b\,max}}}_{-\Delta r_{\mathrm{b\,min}}}, r_{\mathrm{p}}\ {}^{+\Delta r_{\mathrm{p\,max}}}_{-\Delta r_{\mathrm{p\,min}}}, l_1\ {}^{+\Delta l_{1\,\mathrm{max}}}_{-\Delta l_{1\,\mathrm{min}}}, l_2\ {}^{+\Delta l_{2\,\mathrm{max}}}_{-\Delta l_{2\,\mathrm{min}}}, l_3\ {}^{+\Delta l_{3\,\mathrm{max}}}_{-\Delta l_{3\,\mathrm{min}}} \right], \tag{12}$$

where l_i denotes the actuator lengths, with i=1, 2, 3, as given by the inverse geometric model.

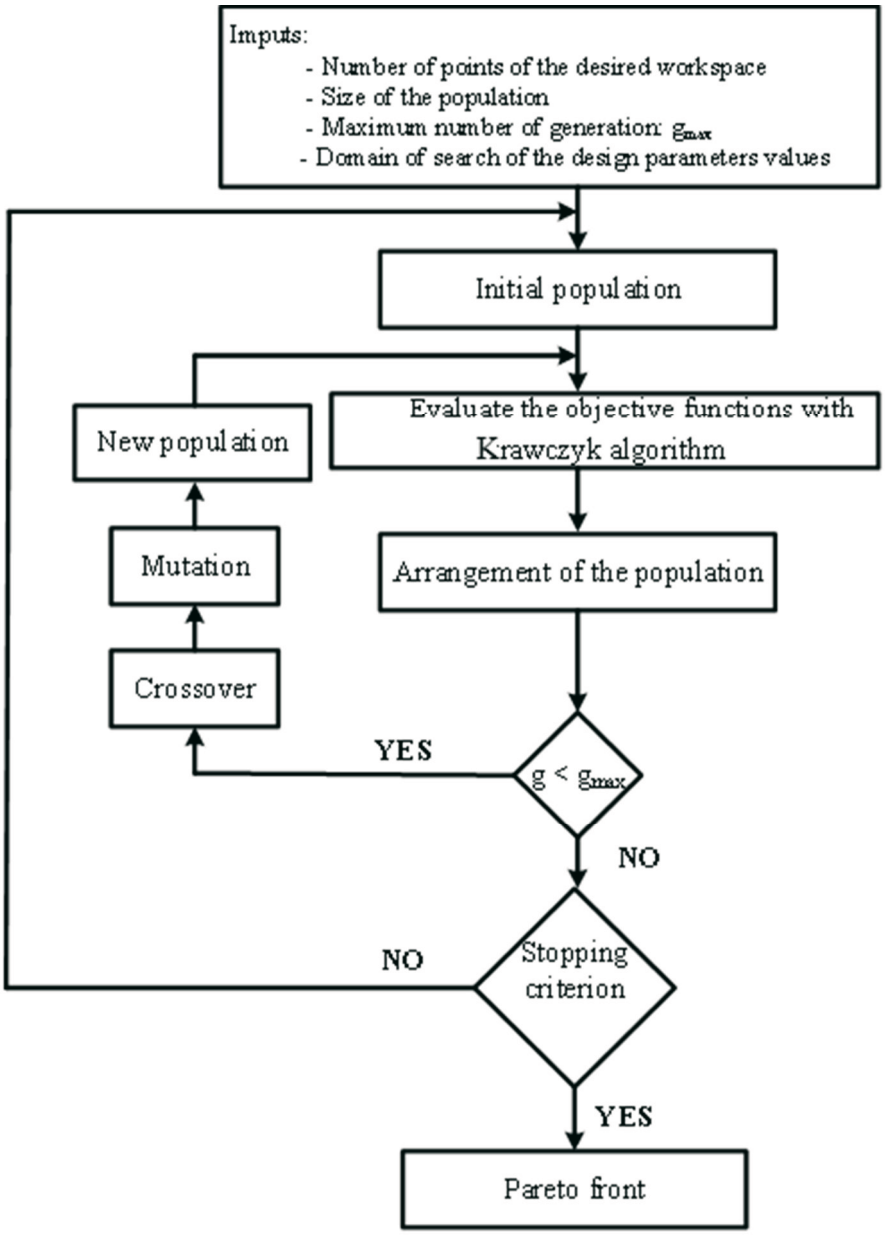

Fig. 6 Optimisation using a genetic algorithm combined with the Krawczyk operator

4.1 Optimisation of the geometric model

The first objective function to be optimised is the robot's position error. This function is formulated in Eq. (13):

$$f_1 = E_{\mathrm{p}} \tag{13}$$

The second objective consists of reducing the robot design cost. This objective is achieved by minimising the inverse of the normalised sum of tolerance intervals for the design parameters, calculated as:

$$f_2 = \frac{1}{\dfrac{\Delta r_{b\min} + \Delta r_{b\max}}{r_b} + \dfrac{\Delta r_{p\min} + \Delta r_{p\max}}{r_p} + \sum_{i=1}^{3} \dfrac{\Delta l_{i\min} + \Delta l_{i\max}}{l_i}}. \tag{14}$$

Each tolerance interval Δp_{i} is normalised by its nominal value p_{i}, ensuring that all parameters contribute proportionally to the tolerance cost.

The following constraint is imposed: the radius of the base is $r_{\mathrm{b}} > r_{\mathrm{p}}$. To avoid singularities, the angular positions are different from each other. The bounds of the design parameters and their tolerance intervals are given in Table 2:

Table 2 Search intervals for the parameters (IGM: inverse geometric model)

Design parameters	Intervals	Uncertainties	Intervals
r_{b} (mm)	[200 800]	Δr_{bmin}, Δr_{bmax} (mm)	[0 1]
r_{p} (mm)	[20 100]	Δr_{pmin}, Δr_{pmax} (mm)	[0 1]
l_1 (mm)	IGM	$\Delta l_{1\mathrm{min}}$, $\Delta l_{1\mathrm{max}}$ (mm)	[0 1]
l_2 (mm)	IGM	$\Delta l_{2\mathrm{min}}$, $\Delta l_{2\mathrm{max}}$ (mm)	[0 1]
l_3 (mm)	IGM	$\Delta l_{3\mathrm{min}}$, $\Delta l_{3\mathrm{max}}$ (mm)	[0 1]

4.2 Optimisation of the kinematic model

In the kinematic model, the first objective consists of minimising the robot's velocity error, while permitting the largest feasible tolerance ranges for the design parameters. The velocity error is assessed throughout the entire workspace, and the maximum deviation along the x, y, and z directions defines the objective function f_3. The second fitness function f_4 aims to minimise the inverse of the sum of parameter errors. The two functions are calculated using the following formulas:

$$\begin{cases} f_3 = \dot{E}_p \\ f_4 = \dfrac{1}{\dfrac{\Delta r_{b\min} + \Delta r_{b\max}}{r_b} + \dfrac{\Delta r_{p\min} + \Delta r_{p\max}}{r_p} + \sum_{i=1}^{3} \dfrac{\Delta l_{i\min} + \Delta l_{i\max}}{l_i}} \end{cases} \tag{15}$$

4.3 Optimisation of the dynamic model

The aim is to reduce the relative actuator's force error in dynamic model optimisation. To represent the dynamic accuracy as a unified function, the relative force error is considered, as defined by:

$$f_5 = \%\Delta \mathrm{d} = \frac{\mathrm{Max}\left(\left|d_{1\max} - \tilde{d}_1\right|, \left|d_{1\min} - \tilde{d}_1\right|\right)}{\left|\tilde{d}_1\right|} + \frac{\mathrm{Max}\left(\left|d_{2\max} - \tilde{d}_2\right|, \left|d_{2\min} - \tilde{d}_2\right|\right)}{\left|\tilde{d}_2\right|} + \frac{\mathrm{Max}\left(\left|d_{3\max} - \tilde{d}_3\right|, \left|d_{3\min} - \tilde{d}_3\right|\right)}{\left|\tilde{d}_3\right|}. \tag{16}$$

The design parameters and the actuators' velocities and accelerations are kept at their nominal values: $r_{\mathrm{d}}, r_{\mathrm{s}}, m_{\mathrm{d}}, m_{\mathrm{s}}, m_{\mathrm{p}}, \alpha_1, \alpha_2, \alpha_3, \dot{l}_1, \dot{l}_2, \dot{l}_3, \ddot{l}_1, \ddot{l}_2, \ddot{l}_3$.

The parameters with search intervals are :

$$r_{\mathrm{b}}, \Delta r_{\mathrm{b\,min}}, \Delta r_{\mathrm{b\,max}}, r_{\mathrm{p}}, \Delta r_{\mathrm{p\,min}}, \Delta r_{\mathrm{p\,max}}, \Delta l_{1\,\mathrm{min}}, \Delta l_{1\,\mathrm{max}}, \Delta l_{2\,\mathrm{min}}, \Delta l_{2\,\mathrm{max}} \Delta l_{3\,\mathrm{min}}, \Delta l_{3\,\mathrm{max}}.$$

The second objective function to be minimised is the inverse of the normalised sum of the parameter intervals, as defined by:

$$f_6 = \frac{1}{\dfrac{\Delta r_{b\,\mathrm{min}} + \Delta r_{b\,\mathrm{max}}}{r_b} + \dfrac{\Delta r_{p\,\mathrm{min}} + \Delta r_{p\,\mathrm{max}}}{r_p} + \displaystyle\sum_{i=1}^{3} \dfrac{\Delta l_{i\,\mathrm{min}} + \Delta l_{i\,\mathrm{max}}}{l_i}}. \tag{17}$$

5. Results and discussion

In this section, the competing objectives consists of selecting the optimal combination of parameters to improve manipulator accuracy and to expand tolerance intervals in order to minimise design costs. The distribution of these competitive functions is represented by the Pareto front.

5.1 The Pareto front of geometric optimisation

In the Pareto front shown in Fig. 7, two solutions, S_1 and S_2, can be distinguished with function f_2 very close. However, the position error is different, ranging from 3.49 to 1.7 mm.

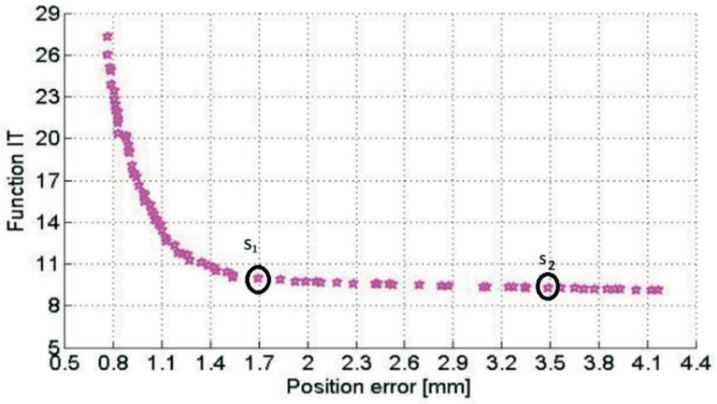

Fig. 7 Pareto front of geometric optimisation

Each solution is represented as a vector:

$$[\mathbf{p}] = \left[r_b \, {}^{+r_b\,\mathrm{max}}_{-r_b\,\mathrm{min}}, r_p \, {}^{+r_p\,\mathrm{max}}_{-r_p\,\mathrm{min}}, l_1 \, {}^{+l_1\,\mathrm{max}}_{-l_1\,\mathrm{min}}, l_2 \, {}^{+l_2\,\mathrm{max}}_{-l_2\,\mathrm{min}}, l_3 \, {}^{+l_3\,\mathrm{max}}_{-l_3\,\mathrm{min}} \right]. \tag{18}$$

The vectors of the parameter nominal values and uncertainties that correspond to S_1 and S_2 are respectively:

$$[\mathbf{p}_1] = \left[455.229 \, {}^{+0.237}_{-0.230}, 20.243 \, {}^{+0.989}_{-0.996}, 644.943 {}^{+0.207}_{-0.138}, 851.298 \, {}^{+0.364}_{-0.369}, 801.235 \, {}^{+0.191}_{-0.203} \right], \tag{19}$$

$$[\mathbf{p}_2] = \left[415.923 \, {}^{+0.740}_{-0.679}, 20.059 \, {}^{+0.998}_{-0.996}, 632.043 {}^{+0.777}_{-0.692}, 824.903 \, {}^{+0.552}_{-0.627}, 777.930 \, {}^{+0.575}_{-0.408} \right] \tag{20}$$

The fitness function values of the selected solutions S_1 and S_2 are given in Table 3:

Table 3 Selected solutions from the Pareto front of geometric optimisation

Solutions	Objective Function f_1 (mm)	Objective function f_2
S_1	1.703	9.900
S_2	3.496	9.274

To ensure a lower positioning error, tighter tolerances can be used for the actuator uncertainties. Conversely, to reduce the robot's design cost, it can be assumed that the radius uncertainties (Δr_b, Δr_p) have minimal influence on the manipulator's position accuracy, allowing these tolerances to be relaxed.

5.2 Pareto front of kinematic optimisation

The Pareto front for the two objective functions, namely velocity error and the function of interval tolerances, is shown in Fig. 8. From this Pareto front, two solution vectors are selected. Table 4 presents the fitness function values while Equations (20) and (21) present their corresponding parameter nominal values and uncertainties.

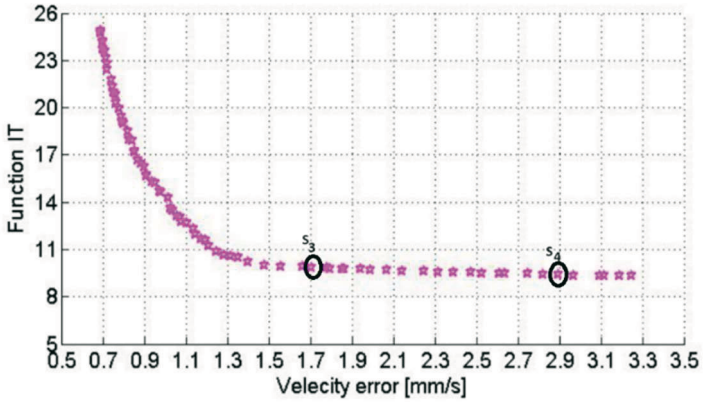

Fig. 8 Pareto front of kinematic optimisation

Table 4 Selected solutions from the Pareto front of kinematic optimisation

Solutions	Objective Function f_2 (mm.s^{-1})	Objective function f_3
S_3	1.703	9.807
S_4	2.969	9.358

The solution vectors $[\mathbf{p}_3]$ and $[\mathbf{p}_4]$ are defined as follows:

$$[\mathbf{p}_3]=\left[574.226\ {}^{+0.135}_{-0.197}, 20.104\ {}^{+0.996}_{-0.994}, 696.396\ {}^{+0.342}_{-0.364}, 937.171\ {}^{+0.269}_{-0.287}, 879.146\ {}^{+0.393}_{-0.317}\right], \quad (21)$$

$$[\mathbf{p}_4]=\left[566.893\ {}^{+0.864}_{-0.908}, 20.044\ {}^{+0.999}_{-0.996}, 692.778\ {}^{+0.747}_{-0.531}, 931.718\ {}^{+0.533}_{-0.496}, 874.124\ {}^{+0.821}_{-0.278}\right]. \quad (22)$$

5.3 Pareto front of dynamic optimisation

The Pareto front of dynamic optimisation is given in Fig. 9.

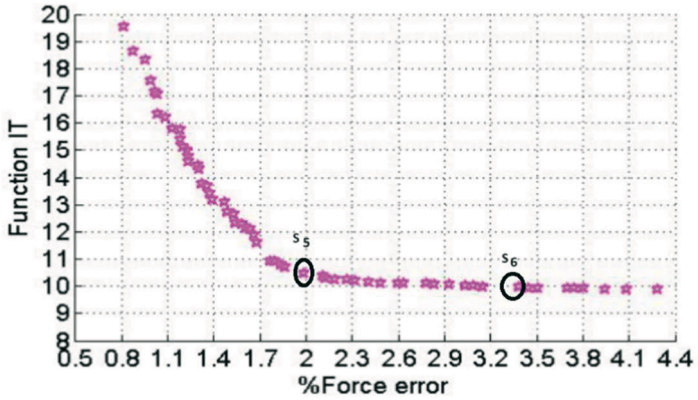

Fig. 9 Pareto front of dynamic optimisation

Table 5 presents the selected solutions of dynamic optimisation S_5 and S_6.

Table 5 Selected solutions from the Pareto front of dynamic optimisation

Solutions	Objective Function f_5 (N)	Objective function f_6
S_5	1.992	10.469
S_6	3.392	9.965

The vectors of the selected solutions are represented by Equations (23) and (24), respectively:

$$[\mathbf{p}_5]=\left[734.283\ {}^{+0.192}_{-0.348},20.122\ {}^{+0.950}_{-0.934},788.896\ {}^{+0.259}_{-0.091},1062.7\ {}^{+0.250}_{-0.113},996.718\ {}^{+0.258}_{-0.123}\right], \quad (23)$$

$$[\mathbf{p}_6]=\left[732.143\ {}^{+0.236}_{-0.647},20.127\ {}^{+0.965}_{-0.971},787.513\ {}^{+0.423}_{-0.656},1060.9\ {}^{+0.482}_{-0.378},995.063\ {}^{+0.235}_{-0.600}\right]. \quad (24)$$

These Pareto fronts provide multiple feasible solutions. Each point represents a solution vector balancing the two objectives. Designers can select a solution according to the specific application, prioritising accuracy (geometric, kinematic, or dynamic), minimising cost, or choosing an intermediate compromise. For kinematics, S_3 offers higher velocity precision, while S_4 favours lower manufacturing cost. For dynamics, S_5 minimises actuator forces, whereas S_6 aims to lower the overall system cost.

From these results, it can be concluded that for the selected optimal solutions S_1, S_2, S_3, S_4, S_5 and S_6 corresponding to the vectors $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$,$\mathbf{p}_4$, $\mathbf{p}_5$ and $\mathbf{p}_6$, the nominal value of the platform's radius is minimal (at the lower bound of the search interval) and the corresponding uncertainties are the largest (at the upper bound of the search interval). On the other hand, to reduce the position error, the uncertainties in actuator lengths are minimised.

With the same vector, it is not possible to optimise all aspects of the robot's accuracy (position, velocity, and actuator forces) simultaneously. For this reason, it is important to obtain each Pareto front separately for different objectives.

6. Accuracy comparison between optimised and non-optimised mechanisms

In this section, an arbitrary trajectory within the robot's workspace is considered in order to evaluate its accuracy in comparison with both the non-optimised mechanism (with parameter uncertainties set at 0.2% of their nominal values) and the optimised mechanism (solutions S_1 and S_5 obtained from the geometric and dynamic optimisations). The details of the proposed trajectory are provided in Table 6.

Table 6 The arbitrary trajectory test

$x(t)$	$y(t)$	$z(t)$
$2t^2+9t$-150	$150\cos(\pi t)$	$30t$+300

The improvement in accuracy along this trajectory, denoted by I, is quantified for the different cases as follows:

$$I = \sum_{i=1}^{N} \frac{E_{(\text{non-optimised})} - E_{(\text{optimised})}}{E_{(\text{non-optimised})}}, \tag{25}$$

where $E_{(\text{non-optimised})}$ is the error of the non-optimised error (uncertainties 0.2% of the nominal values): position, velocity and normalised force error. $E_{(\text{optimised})}$ is the error of the optimised mechanism (solution S_1 and solution S_5). The simulation results are given in Fig. 10.

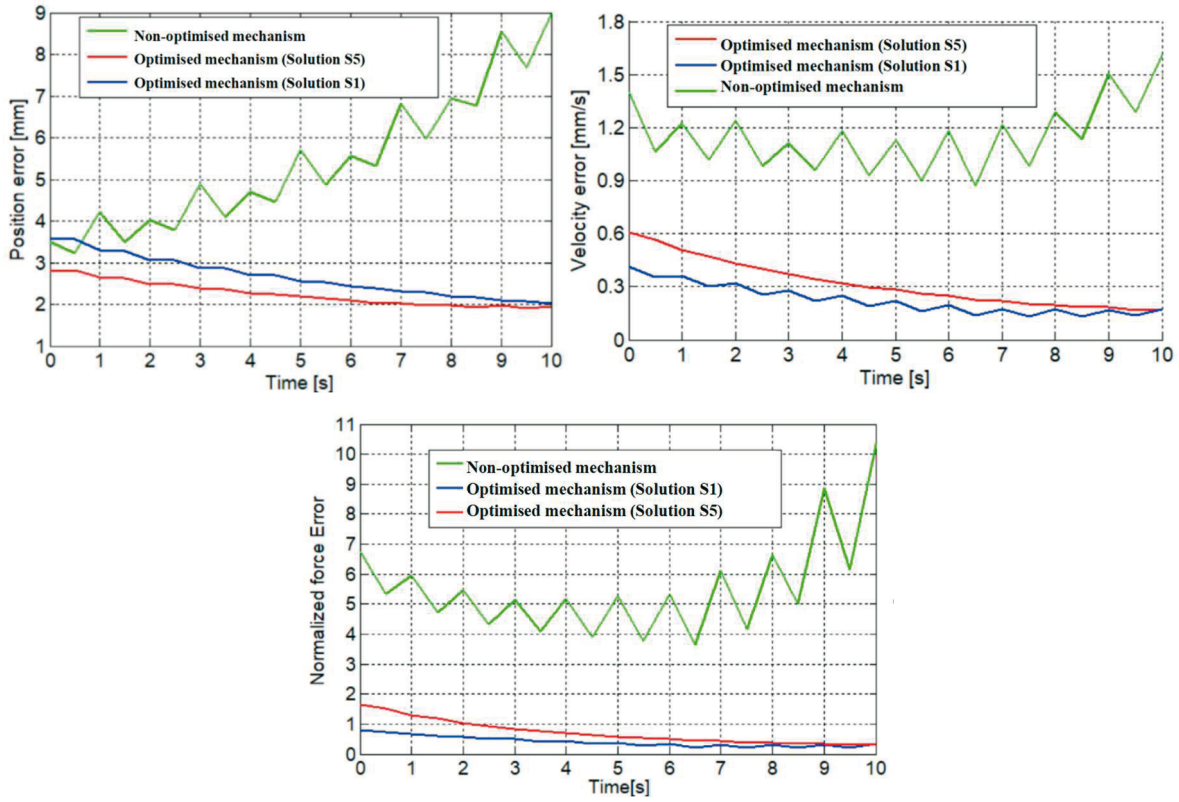

Fig. 10 Manipulator accuracy for the optimised and non-optimised mechanisms

The corresponding improvements in position, velocity, and normalised force along the proposed trajectory are summarised in Table 7.

Table 7 Accuracy improvement (%) for the proposed test trajectory

	%Position_ improvement	%Velocity_ improvement	%Normalised force _ improvement
Solution S_1	60.18	59.76	93.07
Solution S_5	40.32	42.41	88.30

The results indicate a minimum improvement of 40% in geometric accuracy, at least 42% in kinematic accuracy, and more than 88% in dynamic accuracy.

7. Conclusion

In this paper, the multi-objective optimisation of a 3-UPU robot, as an example of a translational parallel manipulator, is addressed, taking into account uncertainties in its design and dynamic parameters. The proposed algorithm integrates interval analysis, based on the Krawczyk operator, with genetic algorithms to enhance the manipulator accuracy and to expand the tolerance intervals, thereby reducing design costs. The results present various Pareto fronts that depict different optimal solutions corresponding to the competitive objective functions. A trajectory test confirmed the improvements in accuracy for the evaluated parameter vectors, validating the effectiveness of the proposed method.

The proposed algorithm can be extended to optimise other types of robots.

It should be acknowledged that the validation conducted in this study is limited to numerical simulations, as no experimental prototype was available. This represents a limitation of the present work. Future research will therefore focus on performing physical experiments to confirm the effectiveness of the optimised design under real operating conditions.

REFERENCES

[1] B. Liu, F. Zhang, and X. Qu, "A method for improving the pose accuracy of a robot manipulator based on multi-sensor combined measurement and data fusion", *Sensors (Switzerland)*, vol. 15, no. 4, pp. 7933–7952, 2015. https://doi.org/10.3390/s150407933

[2] Y. Liu, J. Zhao, Y. Yao, Q. Cao, and J. Cui, "High-speed parallel robot dynamic modelling based on PLC", *J. Supercomput.*, vol. 76, no. 5, pp. 3158–3172, 2020. https://doi.org/10.1007/s11227-018-2530-3

[3] L. Kong, H. Wang, P. Zhang, Y. Zhao, G. Chen, and L. Zhao, "Stability analysis for a planar parallel manipulator with the capability of self-coordinating the load distribution", *Chinese J. Mech. Eng. (English Ed.*, vol. 28, no. 4, pp. 821–829, 2015. https://doi.org/10.3901/CJME.2015.0310.058

[4] Y. Yang, L. Tang, H. Zheng, Y. Zhou, Y. Peng, and S. Lyu, "Kinematic stability of a 2-DOF deployable translational parallel manipulator", *Mech. Mach. Theory*, vol. 160, p. 104261, 2021. https://doi.org/10.1016/j.mechmachtheory.2021.104261

[5] J. Aginaga, I. Zabalza, O. Altuzarra, and J. Nájera, "Improving static stiffness of the 6 - R US parallel manipulator using inverse singularities", *Robot. Comput. Integr. Manuf.*, vol. 28, no. 4, pp. 458–471, 2012. https://doi.org/10.1016/j.rcim.2012.02.003

[6] J. Wang and C. M. Gosselin, "A new approach for the dynamic analysis of parallel manipulators", *Multibody Syst. Dyn.*, vol. 2, no. 3, pp. 317–334, 1998. https://doi.org/10.1023/A:1009740326195

[7] H. Zhao and C. F. Zhang, "Accuracy analysis of a parallel robot with particle swarm optimization", *CIS 2009 - 2009 Int. Conf. Comput. Intell. Secur.*, vol. 2, pp. 142–145, 2009. https://doi.org/10.1109/CIS.2009.45

[8] J. J. Kim and J. J. Lee, "Trajectory optimization with particle swarm optimization for manipulator motion planning", *IEEE Trans. Ind. Informatics*, vol. 11, no. 3, pp. 620–631, 2015. https://doi.org/10.1109/TII.2015.2416435

[9] D. Liang, Y. Mao, Y. Song, and T. Sun, "Kinematics, dynamics and multi-objective optimization based on singularity-free task workspace for a novel SCARA parallel manipulator", *J Mech Sci Technol*, vol. 38, pp. 423–438, 2024. https://doi.org/10.1007/s12206-023-1235-6

[10] Q. Cui, P. Liu, H. Du, H. Wang, and X. Ma, "Improved multi-objective artificial bee colony algorithm-based path planning for mobile robots", *Front. Neurorobot.*, vol. 17, 2023. https://doi.org/10.3389/fnbot.2023.1196683

[11] Y. Ma, W. Sun, H. Wu, B. Li, Qi. Liu, S. Liu, C. Dong and D. Peng, "A digital design framework for the dimensional optimization of parallel robots based on kinematic and elasto-dynamic performance", *Sci. Rep.*, vol. 14, no. 1, pp. 1–21, 2024. https://doi.org/10.1038/s41598-024-80413-2

[12] S. D. Stan, V. Maties, and R. Balan, "Stochastic optimization method for optimized workspace of a six degree of freedom micro parallel robot", *2008 2nd IEEE Int. Conf. Digit. Ecosyst. Technol. IEEE-DEST 2008*, pp. 502–507, 2008. https://doi.org/10.1109/DEST.2008.4635170

[13] C. Lahdiri, H. Saafi, M. Abdelfattah, and M. A. Laribi, "Optimal design of a new redundant spherical parallel manipulator with an unlimited self-rotation capability", *Robotica*, vol. 43, no. (1), pp. 300–315, 2025. https://doi.org/10.1017/S0263574724001851

[14] A. Antonov, “Design Optimization of a Parallel–Serial Manipulator Considering Stiffness Criteria”, *Robotics*, vol. 13, no. https://doi.org/10.3390/ robotics13120176, pp. 1–23, 2024. https://doi.org/10.3390/robotics13120176

[15] A. Kaveh, K. Laknejadi, and B. Alinejad, “Performance-based multi-objective optimization of large steel structures”, *Acta Mech.*, vol. 223, no. 2, pp. 355–369, 2012. https://doi.org/10.1007/s00707-011-0564-1

[16] G. Wu, S. Caro, S. Bai, and J. Kepler, “Dynamic modeling and design optimization of a 3-DOF spherical parallel manipulator”, *Rob. Auton. Syst.*, vol. 62, no. 10, pp. 1377–1386, 2014. https://doi.org/10.1016/j.robot.2014.06.006

[17] S. El Hraiech, A. Chebbi, Z. Affi, and L. Romdhane, “Error estimation and sensitivity analysis of the 3-UPU translational parallel robot due to design parameter uncertainties”, *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.*, vol. 0, no. 0, p. 095440621879367, 2018. https://doi.org/10.1177/0954406218793673

[18] S. EL Hraiech, A. H. Chebbi, Z. Affi, and L. Romdhane, “Genetic algorithm coupled with a Krawczyk method for multi- objective design parameters optimization of the 3-UPU manipulator”, *Robotica*, vol. 38, pp. 1–17, 2020. https://doi.org/10.1017/S0263574719001292

Submitted: 10.9.2025

Accepted: 06.02.2026

Safa El Hraiech*
Mechanical Engineering Laboratory (LGM), National Engineering School of Monastir, University of Monastir, Monastir, Tunisia

Youssef Chouaibi
Mechanical Engineering Laboratory (LGM), National Engineering School of Monastir, University of Monastir, Monastir, Tunisia
Higher Institute for Technological Studies of Sidi Bouzid, Sidi Bouzid, Tunisia

Zouhaier Affi
Mechanical Engineering Laboratory (LGM), National Engineering School of Monastir, University of Monastir, Monastir, Tunisia

*Corresponding author:
safa_el_hraiech@hotmail.fr