Open Access

Multi-Objective Design of Parallel Manipulator Using Global Indices

F. A. Lara-Molina*[,1], J. M. Rosário[1] and D. Dumur[2]

[1]*Department of Mechanical Design, Mechanical Engineering School, State University of Campinas, Campinas SP, Brazil*

[2]*Automatic Control Department, SUPELEC, Plateau de Moulon, F 91 192 Gif sur Yvette cedex, France*

Abstract: The paper addresses the optimal design of parallel manipulators based on multi-objective optimization. The objective functions used are: Global Conditioning Index (GCI), Global Payload Index (GPI), and Global Gradient Index (GGI). These indices are evaluated over a required workspace which is contained in the complete workspace of the parallel manipulator. The objective functions are optimized simultaneously to improve dexterity over a required workspace, since single optimization of an objective function may not ensure an acceptable design. A Multi-Objective Evolution Algorithm (MOEA) based on the Control Elitist Non-dominated Sorting Genetic Algorithm (CENSGA) is used to find the Pareto front.

Keywords: Parallel manipulator, optimal design, multi-objective genetic algorithm, stewart-gough platform.

1. INTRODUCTION

Parallel manipulators have closed kinematic chains; due to this feature, they have some advantages over serial manipulators. Among these advantages are: low inertia, high stiffness, high speed, and better position accuracy. However, parallel manipulators have some disadvantages: a reduced workspace and kinematic singularities inside the complete workspace. Parallel manipulators have been widely used in high performance positioning systems, such as flight simulators [1], machine-tools [2], biped locomotion systems [3] and surgical robots [4].

The optimal design problem of parallel manipulators consists in determining a set of design parameters of the parallel manipulator to guarantee an optimal criterion. Many works have addressed the optimal design of parallel manipulators with different approaches depending on the characteristics to be optimized. In those works, the main approach has been to solve the optimal design problem using diverse methods, in which the optimum criteria are: the maximum dexterity [5,6], the maximum stiffness [7], the minimum position error in the movable platform [8], the maximum velocities in the platform [9] and the required maximum workspace [10-13].

Proposing new synthesis methods for the optimal design of parallel manipulators ensures high kinematic performance for the growing number of new applications. Optimal design often depends on some required characteristics, such as required workspace or limits on the dimensions of the manipulator. For this purpose, it is interesting to evaluate simultaneously dexterity measures to improve kinematic performance. Aiming to achieve good dexterity, high payload capability, and uniformity of the dexterity over a required workspace, an optimal design method is proposed to optimize the geometric architecture of a parallel manipulator.

Some works have been addressed following this direction. Ceccarelli and Lanni [14] proposed a design formulation for the workspace restricted to a serial manipulator with three revolute joints; this was formulated as a multi-objective optimization problem by using the workspace. Hao and Merlet [15] proposed an optimal design methodology based on interval analysis that allows one to determine all possible geometries satisfying two simultaneous and compulsory requirements, accuracy and workspace, through an algorithm with acceptable computational performance. Lara *et al.* [16] developed a virtual environment simulation of kinematics and dynamics of Stewart-Gough platforms to test different control strategies and thus establish the relation between kinematic and dynamic performance of parallel manipulators. They remarked the necessity of optimal kinematic design. Lanni and Ceccarelli [17] have also used a multi-objective optimum algorithm for finger grippers with respect to an imposed workspace area which considers four different objective functions. Gao *et al.* [18] optimized the system stiffness and dexterity of a parallel manipulator using optimization methods based on artificial intelligence techniques: genetic algorithms and artificial neural networks. They demonstrate the validity of artificial intelligence for optimal design of parallel manipulators.

In the works mentioned above, the requirements of dexterity are generally evaluated at a local kinematic configuration, consequently, for a constant kinematic configuration and not for a required workspace. Thus, global measures have not always been used in optimal design, with the exception of: [7,9,13,19].

In the same way, single-objective optimization (SOO) is not enough to ensure an optimal design of parallel manipulators. For this reason, some authors have developed new performance indices that combine different kinematic mea-

*Address correspondence to this author at the Departamento de Projeto Mecânico - Faculdade de Engenharia Mecânica - UNICAMP. Rua Mendeleiev, s/n - Cidade Universitária "Zeferino Vaz" - Barão Geraldo - Caixa Postal 6122 - CEP: 13.083-970 - Campinas - SP, Brazil; Tel: +551935213166; Fax: +551932893722; E-mail: lara@fem.unicamp.br sures, for example: Miller [13] used an index that combines dexterity and workspace, Zhang [9] proposed a global index to measure the joint velocity in an imposed workspace.

In this work, we propose a multi-objective design of kinematic parallel manipulators based on Multi-Objective Evolutionary Algorithm (MOEA). For this purpose, three kinematic performance indices are the objective functions for optimization, these indices are: Global Conditioning Index (GCI), Global Payload Index (GPI), and Global Gradient Index (GGI). The multi-objective optimization (MOO) is performed using CENSGA to search for the Pareto front which is a set of optimal design parameters. Thus, two simulations are performed to demonstrate the design procedure. In the first simulation, a GAs is used to maximize the GCI and examine the behavior of the other indices, GGI, and GPI. The second simulation, CESGA is used to reach the optimal design parameters.

This paper is organized in several sections. Section 2 presents the kinematic and geometric modeling, to introduce the design parameters. In section 3, the three global performance indices are presented: GCI, GPI, and GGI. In section 4, the optimization problem of parallel manipulator is formulated. Section 5 presents the simulations and results. Finally Section 6 covers the conclusion.

2. KINEMATIC AND GEOMETRIC MODELING

Fig. (**1**) shows the Stewart-Gough platform. Six identical legs connect the movable platform to the fixed base by universal joints denoted by *U* at points B_i and spherical joints denoted by *S* at points A_i, $i = 1, 2, ..., 6$, respectively. Both the universal and the spherical joints are passive. Each leg has an upper and a lower member connected by a pris-matic joint denoted by *P*. The prismatic joint is activated to extend/ retract the leg. The universal joint can be substituted by a spherical joint without changing the motion of the movable platform.

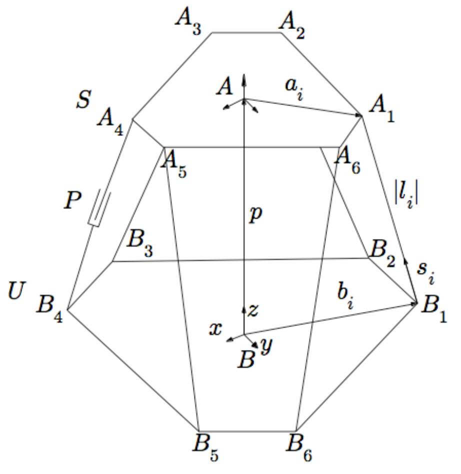


Fig. (1). Spatial Stewart-Gough platform.

The six degrees of freedom (DOF) in the movable platform are linear and angular motions. Linear motions consists of longitudinal motions in *x*-*y*-*z*-axes. Angular motions are expressed as Euler angles with respect to *x*-*y*-*z*-axes.

Two coordinate frames $\{A\}$ and $\{B\}$ are attached to the movable and fixed based platforms respectively. The vector $\mathbf{b}_i = [b_{ix}\ b_{iy}\ b_{iz}]^T$ describes the position of the reference point B_i with respect to the frame $\{B\}$; thus $\mathbf{b}_i$ is expressed as:

$$\mathbf{b}_i = \begin{bmatrix} r_b \cos(\psi_i) \\ r_b \sin(\psi_i) \\ 0 \end{bmatrix} = \begin{bmatrix} b_{ix} \\ b_{iy} \\ b_{iz} \end{bmatrix} \qquad (1)$$

$$\psi_i = \frac{i\pi}{3} - \frac{\varphi_b}{2} \qquad i = 1, 3, 5$$

$$\psi_i = \psi_{i-1} + \varphi_b \qquad i = 2, 4, 6$$

In the same way, the vector $\mathbf{a}_i = [a_{ix}\ a_{iy}\ a_{iz}]^T$ describes the position of the reference point A_i with respect to the reference frame $\{A\}$; then $\mathbf{a}_i$ is expressed as:

$$\mathbf{a}_i = \begin{bmatrix} r_a \cos(\Psi_i) \\ r_a \sin(\Psi_i) \\ 0 \end{bmatrix} = \begin{bmatrix} a_{ix} \\ a_{iy} \\ a_{iz} \end{bmatrix} \qquad (2)$$

$$\Psi_i = \frac{i\pi}{3} - \frac{\varphi_a}{2} \qquad i = 1, 3, 5$$

$$\Psi_i = \Psi_{i-1} + \varphi_a \qquad i = 2, 4, 6$$

The Stewart-Gough platform geometry was defined with two coplanar semiregular hexagons; the first corresponds to the base hexagon and the second to the movable platform. According to equations (1) and (2), the Stewart-Gough platform can be defined with five kinematic design parameters: r_b is the radius of the fixed base, r_a is the radius of the movable platform, φ_b is the spacing angle of the vectors $\mathbf{b}_i$, φ_a is the spacing angle of the vectors $\mathbf{a}_i$ and finally, z_0 which is the center *z*-axes coordinate of the constant workspace. Consequently, a vector ***n*** of design parameters is defined.

$$\lambda = \begin{bmatrix} r_a & r_b & \varphi_a & \varphi_b & z_0 \end{bmatrix}^T \qquad (3)$$

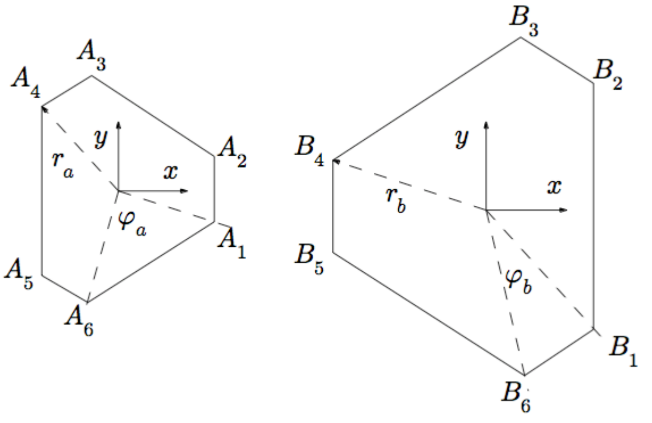


Fig. (2). Stewart-Gough platform geometry: joints of movable and fixed base platform.

The pose of the centroid $\{A\}$ of the movable platform respect to frame $\{B\}$ is described by six generalized coordinates

$$\mathbf{x} = \begin{bmatrix} x & y & z & \alpha & \beta & \gamma \end{bmatrix}^T = \begin{bmatrix} \mathbf{p} \\ \theta \end{bmatrix} \tag{4}$$

According to [20], the differential kinematics can be written as,

$$\dot{\mathbf{l}} = \mathbf{J}(\mathbf{x},\lambda)\dot{\mathbf{x}} \tag{5}$$

in which $\mathbf{J}(\mathbf{x},\lambda)$ is the 6 by 6 Jacobian matrix of Stewart-Gough platform, $\dot{\mathbf{l}} = \begin{bmatrix} \dot{l}_1 & \cdots & \dot{l}_6 \end{bmatrix}^T$ are the articular coordinates velocities and $\dot{\mathbf{x}} = \begin{bmatrix} \dot{\mathbf{p}} & \omega \end{bmatrix}^T$ are the generalized coordinates velocities.

$$\mathbf{J}(\mathbf{x},\lambda) = \begin{bmatrix} \mathbf{s}_1^T & (R\mathbf{a}_1 \times \mathbf{s}_1)^T \\ \mathbf{s}_2^T & (R\mathbf{a}_2 \times \mathbf{s}_2)^T \\ \vdots & \vdots \\ \mathbf{s}_6^T & (R\mathbf{a}_6 \times \mathbf{s}_6)^T \end{bmatrix} \tag{6}$$

Remarking that the Jacobian matrix depends on the design parameters ***n*** and the pose **x**. $\mathbf{s}_i$, such as it is presented in Fig. (**1**). In the same way, the static model can be whiten as,

$$\tau = \mathbf{H}(\mathbf{x},\lambda)\mathbf{g} \tag{7}$$

in which $\mathbf{H}(\mathbf{x},\lambda) = J^T(\mathbf{x},\lambda)$, $\tau = [\mathbf{f}\ \mathbf{m}]^T$ is the total force and moment in the centroid $\{A\}$ of the movable platform, and $\mathbf{g} = \begin{bmatrix} g_1 & \cdots & g_6 \end{bmatrix}^T$ the forces along the legs.

Since the Jacobian matrix establishes a kinematic relation between articular and generalized coordinates, the Jacobian matrix is useful in evaluating the performance for different design parameters and poses. However, when $|\mathbf{J}(\mathbf{x},\lambda)|$ is close to zero, there are problems in the kinematic performance, because the matrix is near a singular point and there may be numerical difficulties in calculating the inverse matrix; thus, the position accuracy is decreased, small errors in actuators cause a large error in the movable platform. Analogously, some loads in the movable platform cannot be controlled by the forces in the legs due to $|\mathbf{H}(\mathbf{x},\lambda)|$ being close to zero. This phenomenon is related to singularities within the complete workspace [21].

Hence, performance indices, to measure the kinematic condition over the workspace are essential to prevent the problems from carry to singularities. Accordingly, performance indices are used in optimal design to determine the optimal kinematic design parameters of parallel manipulators.

3. KINEMATIC PERFORMANCE INDEX

Dexterity is an important factor to be considered in optimal design of parallel manipulators. Intuitively, dexterity is the skill to handle an object with accuracy. The dexterity refers to a kinematic characteristic of the manipulator that is measured in terms of the Jacobian matrix due to the physical meaning.

Any m x x matrix **J** can be factored in the following form:

$$\mathbf{J}_{m\times n} = [\mathbf{U}^T]_{m\times m}[\Sigma]_{m\times n}[\mathbf{V}]_{n\times n} \tag{8}$$

in which Σ is a diagonal composed of the singular values. For $m = 6$ and $n = 6$, we have six singular values $\sigma_1 \geq \sigma_2 \geq$ K $\sigma_5 \geq \sigma_6 \geq 0$. The condition number is $k(\mathbf{J}) = \sigma_1 / \sigma_6$.

The condition number of Jacobian matrix $k(\mathbf{J})$ has been used to measure the dexterity in a kinematic pose of a parallel manipulator. As a measure of dexterity it varies from 1 (isotropic condition) to infinite (singular condition). The condition number is related to the accuracy of the parallel manipulator in a specific configuration.

The condition number expresses the dexterity at an specific pose. The global kinematic indices use the condition number to express a dexterity measure of a workspace.

3.1. Global Conditioning Index

A commonly used criterion to evaluate the dexterity is the global conditioning index (GCI) [22]. The reciprocal of the condition number is used as the measure of local dexterity, thus $1/k(\mathbf{J})$ varies from 0 (singular condition) to 1 (isotropic condition). For a defined workspace w, the GCI is defined as:

$$GCI = \frac{\int_w 1/k(\mathbf{J})dw}{\int_w dw} \tag{9}$$

In practice, the GCI is approximated by a discrete sum. Thereby, the workspace w is discretized in N_w points.

$$GCI \simeq \frac{1}{N_w}\sum_{i=1}^{Nw} 1/k_i(\mathbf{J}) \tag{10}$$

in which, i is one of the discretizing points in the workspace.

The greater the global conditioning index, the more dexterity over the workspace, therefore the global conditioning index should be maximized.

3.2. Global Gradient Index

This index was first used by Kurtz and Hayward [23] and has also been used by Lee *et al.* [24]. This index is used to measure the uniformity of kinematic dexterity, the relative variation of this measure should be as small as possible to ensure an uniform dexterity over the workspace. Any small displacement of the movable platform would increase the singular condition, thus increasing the position error.

In order to verify uniformity of any function, the magnitude of its gradient is evaluated. Thus, a local measure of flatness of dexterity is given by $\|\nabla 1/k(\mathbf{J})\|$. Integrating this to obtain a global measure would not be a good idea. Hence, the global gradient index is defined as follows:

$$\nabla GCI = \max_w \|\nabla 1/k(\mathbf{J})\| \tag{11}$$

GGI is approximated by calculating the maximum of local flatness of dexterity over the workspace. If GCI is zero, then the dexterity is uniform throughout the workspace. The local gradient is calculated numerically, because there is no analytic expression for this purpose. The gradient will be approximated by first-order difference equations. Thus, the gradient for a six DOF manipulator can be approximated as:

$$\nabla 1/k(\mathbf{J}) = \left[\frac{\partial 1/k(\mathbf{J})}{\partial x}, \frac{\partial 1/k(\mathbf{J})}{\partial y}, \frac{\partial 1/k(\mathbf{J})}{\partial z}, \right.$$
$$\left. \frac{\partial 1/k(\mathbf{J})}{\partial \alpha}, \frac{\partial 1/k(\mathbf{J})}{\partial \beta}, \frac{\partial 1/k(\mathbf{J})}{\partial \gamma}\right] \qquad (12)$$

in which

$$\frac{\partial 1/k(\mathbf{J})}{\partial x} \simeq \frac{1/k(\mathbf{J}(x+\Delta x, y, z, \alpha, \beta, \gamma))}{\Delta x}$$
$$- \frac{1/k(\mathbf{J}(x, y, z, \alpha, \beta, \gamma))}{\Delta x} \qquad (13)$$

The smaller the GGI, the less variation of dexterity over the workspace; therefore the global gradient index should be minimized.

3.3. Global Payload Index

Payload capability measures have been widely used in parallel manipulator design [25-27]. This index represents force transmission capability, thereby determining if the manipulator can support loads on the movable platform. In the equation (7), the relationship between the forces in legs and the force and total moment at the centroid of the movable platform was presented. Thus, $\tau = \mathbf{H}(\mathbf{x}, \lambda)\mathbf{g}$. Using the left pseudoinverse, the minimum norm solution to this equation is,

$$\mathbf{g} = \mathbf{J}(\mathbf{x}, \lambda)(\mathbf{J}(\mathbf{x}, \lambda)^T \mathbf{J}(\mathbf{x}, \lambda))^{-1} \tau \qquad (14)$$

From the singular value decomposition theorem, the bounds on ||**g**|| can be established as,

$$\| \tau \| / \sigma_{max} \leq \| \mathbf{g} \| \leq \| \tau \| / \sigma_{min} \qquad (15)$$

Since the minimum force bound depends on the minimum singular value σ_{min}, this measure is used to calculate the payload capability; thus, $1/\sigma_{min}$ is the local value of payload capability, whose minimum value is 0 in the singular condition; and it does not have a maximum limit value bound. Maximizing this value would reduce the force along the legs. The global payload index (GPI) is defined as.

$$GPI = \frac{\int_w \sigma_{min} dw}{\int_w dw} \qquad (16)$$

Therefore, the larger the GPI, the bigger the force transmission in the movable platform.

Finally, according to the three global performance indices presented, the optimal design corresponds to selecting the design parameters simultaneously and over a required workspace, so that they:

- maximize dexterity, *i.e.*, maximize GCI.
- maximize payload transmission capability, *i.e.*, maximize GPI.
- maximize dexterity uniformity over the required workspace, *i.e.*, minimize GGI.

The conventional Jacobian matrix expresses a mixed relation of both translational and rotational degrees of freedom. The elements of the conventional Jacobian matrix have nonhomogenous physical units. Therefore, the use of performance indices such as the condition number of the Jacobian matrix may lack in physical meaning. To avoid the unit inconsistency problem, Gosselim [28] proposed a formulation of a dimensionally homogeneous Jacobian matrix of planar and spatial manipulator, the new condition number is invariant under of scaling of manipulator.

We we treat separately orientation and position dexterity [5,9]. Let us rewrite the Jacobian matrix

$$\mathbf{J}(\mathbf{x}, \lambda) = \begin{bmatrix} \mathbf{J_p} & \mathbf{J_R} \end{bmatrix} \qquad (17)$$

Thus, $k(\mathbf{J_p})$ and $k(\mathbf{J_R})$ respectively give measures for position and orientation dexterity.

To guarantee position and orientation dexterity, they are applied in design by computing separately the global kinematic indices.

4. METHODOLOGY

Optimal design aims to reach the optimal geometric configuration according to objective functions and geometric constraints. For parallel manipulators, optimization is addresses the maximization of kinematic dexterity and payload capability and the minimization of variation of kinematic dexterity over a required workspace. Since these three aspects are considered simultaneously, then a multi-objective optimization is considered.

4.1. Formulation of Parallel Manipulator Optimization

Accordingly, the optimal design problem became in a multi-objective optimization problem, in which two or more conflicting objective functions are optimized subject to certain constrains. Based on kinematic performance indices, the optimal kinematic performance is achieved when dexterity and force transmission capability are maximized and variation of dexterity over a required workspace is minimized. The required workspace and geometric constraints are important issues in optimization.

For practical purposes, the required workspace w is hyper-parallepiped with constant orientation; the workspace is shown in Fig. (**3**), in which $x_w = y_w = 2z_w$.

Thus, $$w = [-x_w/2, x_w/2] \times [-y_w/2, y_w/2] \times$$
$$[-z_w/2, z_w/2] \times [\varnothing] \times [\varnothing] \times [\varnothing]$$

The maximum and minimum length limit of the legs, for the required workspace, is given by

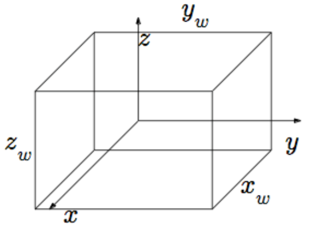


Fig. (3). Hyper-parallepiped required workspace.

$$v_i^{min}(\mathbf{x},\lambda) \le l_i \le v_i^{max}(\mathbf{x},\lambda) \quad (18)$$

In which, v_i is the inverse kinematic. Moreover, restrictions on the design parameters are considered as functions of the dimensions of the required workspace. The geometric restriction is given by the following equality.

$$\sum_{i=2}^{m} \lambda_i = x_w \quad (19)$$

Thus, the multi-objective optimization to reach the optimal design parameters of the parallel manipulator is given by

$$\max_{\lambda}\{GCI(\lambda), GPI(\lambda)\} \qquad \min_{\lambda}\{GGI(\lambda)\}$$

subject to

$$\sum_{i=2}^{m} \lambda_i = x_m$$

$$r_a, r_b \in [r_{min}, r_{max}], \varphi_a, \varphi_b \in [\varphi_{min}, \varphi_{max}]$$

$$z_0 \in [z_{min}, z_{max}]$$

$$\forall \mathbf{x} \in w$$

$$w = [-x_w/2, x_w/2] \times [-y_w/2, y_w/2] \times$$

$$[-z_w/2, z_w/2] \times [\varnothing] \times [\varnothing] \times [\varnothing] \quad (20)$$

To solve this optimization, genetic algorithms are reviewed in the next section.

4.2. Genetic Algorithm

Traditional optimization methods use local search by sequential procedures, such as gradient and Hessian, which compare the values of the subsequent points and then examine the relative optimal points; hence, local optima can be taken for global optima. Conventional methods can fail, since they converge to a local optima.

Genetic Algorithms (GAs) are heuristic search algorithms based on the mechanism of natural selection and natural genetics initially proposed by Holland [29]. GAs are high performance and robust optimization methods to solve engineering problems. They have been used in a variety of engineering fields such as in machine design [30].

In general, a genetic algorithm has five basic components:

1. A genetic representation of solutions to the problem [31], the diagram of this procedure is presented in Fig. (**4**)
2. A way to create an initial population of solutions
3. Selection of the population for next generation 4. An evaluation function rating solutions in terms of their fitness
4. Genetic operators that alter the genetic ascendants during reproduction

GAs start with an initial set of random solutions, this set of solutions are called the population. Each individual of the population, which is a chromosome, represents a potential solution to the problem. The encoding is a genetic representation of the chromosome. In the evaluation, a measure of fitness is assigned to each individual. Individuals called parents are selected, the parents contribute to the population at the next generation. Some individuals of the population suffer genetic operations to create new individuals through stochastic transformations. There are two types of genetic operations: crossover and mutation. Crossover creates new individuals by combination of the parts of two parents; and mutation creates new individuals by randomly altering chromosome characteristics to guarantee genetic diversity in the population. New individuals of the population are called offspring. A new population is formed by selecting the more fit individuals from the present population and the offspring population. After successive iterations called generations, the algorithm converges to the best individual, which hopefully represents an optimal solution to the problem [31].

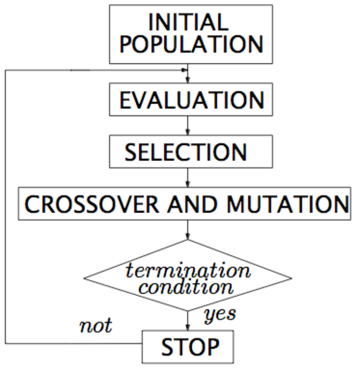


Fig. (4). Genetic algorithm.

4.3. Pareto Front

The principle of multi-objective optimization is different from single-objective optimization. In the single-objective optimization case, the solution of the optimization aims to obtain the best solution over all other alternatives. In the case of multi-objective optimization, there is not necessary a best solution due to conflicts among all objective functions. Therefore, the result of the multi-objective optimization is a set of solutions for multiple objectives. The solutions are called non-determinant or Pareto-optimal solutions when an improvement in one objective requires a degradation of another. The multi-objective optimization can be formally represented as follows:

$$\min\{z_1 = f_1(x), z_2 = f_2(x), \ldots, z_q = f_q(x)\}$$

subjectto

$$g_j(x) \le 0 \quad j = 1, 2, \ldots, m \quad (21)$$

S is used to denote the feasible region in the decision space and Z is used to denote the feasible region in the criterion space.

$$S = \{x \in R^n \mid g_j \le 0, j = 1, 2, \ldots, m, x \ge 0\} \quad (22)$$

in which, $z \in R^n$ is a vector of n decision variables, $f(x)$ is the objective function and g_i are m inequalitiy constraint functions.

$$Z = \{z \in R^q \mid z_1 = f_1(x), z_2 = f_2(x), \ldots,$$

$$z_q = f_q(x), x \in S\} \tag{23}$$

A given $z^0 \in Z$, it is a Pareto solution if and only if there does not exist another point $z \in Z$, such that

$$z_k z_k^0 \quad \text{forsome} \quad k \in \{1, 2\ldots, q\}$$

$$z_l \geq z_l^0 \quad \text{forall} \quad l \neq k \tag{24}$$

4.4. Controlled Elitist Non-dominated Sorting Genetic Algorithm

Multi-Objective Evolutionary Algorithm (MOEA) are used to find the Pareto front. Aiming to find the real Pareto front, an evolutionary algorithm can be used to find multiple Pareto font solutions in a single simulation run.

The MOEA used to solve this optimization is the Controlled Elitist Non-dominated Sorting Genetic Algorithm (CENSGA), a variant of NSGA-II [32].

CENSGA [33] offers a quantitative control over the algorithm selection. Since, it is important to maintain the diversity of the population to converge to Pareto front, the activity of CE focused in two factors: smooth the elitism of the NSGA II, and limits the number of individuals on each Pareto front. Elitism in the NSGA-II is in two phases. Elitism in the CE has two steps. First, next generation population is selected from the best individuals using the Pareto criterium from the parent and offspring of present population. Then, the selection is performed with a random procedure. Furthermore, the number of individuals on each Pareto front is limited by a geometric decreasing function, in which the reduction rate is *r*.

5. SIMULATION AND DISCUSSION

Three simulations are performed using matlab optimization tool [34] and simulink. The first simulation is single objective optimization to maximize the global conditioning index and examine the behavior of the other indices, GGI and GPI. The second simulation is multi-objective optimization to reach the optimal parameters, optimizing all performance indices simultaneously. In both simulations, only $\boldsymbol{J_p}(\mathbf{x},\lambda)$ is considered to compute the global performance indices, since the orientation over the hyper-parallepiped workspace is constant. Finally, throughout the dynamical simulation of dynamics and control, position error analysis is performed to validate multi-objective optimization results.

5.1. Optimization of Global Conditioning Index

In the first example, the five design parameters of vector λ are optimized for obtaining the maximum GCI and therefore maximum dexterity on the Stewart-Gough platform showed in Figs. (**1** and **2**). For this optimization, the behavior of global gradient GGI and global payload GPI are evaluated to determine how these indices vary when GCI is maximized in the hyper-parallepiped workspace with constant orientation.

In this case, the optimization problem is formulated as follows.

$$\max_{\lambda}\{GCI(\lambda)\}$$

subject to

$$-r_a + r_b = 0.1\text{m}$$

$$r_a, r_b \in [0.05\text{m}, 0.2\text{m}], \varphi_a, \varphi_b \in [3, 117]$$

$$z_0 \in [0.05\text{m}, 0.25\text{m}]$$

$$\forall \quad \mathbf{x} \in w$$

$$w = [-0.05\text{m}, 0.05\text{m}] \times [-0.05\text{m}, 0.05\text{m}] \times [-0.025\text{m}, 0.025\text{m}] \times [0] \times [0] \times [0]$$

After some preliminary simulations, we use the parameters in Table **1** to run GAs.

Table 1. Parameters Used for Runing GA

Parameter	Setting
Population size	50
Maximum of generations	52
Encoding type	Real
Selection strategy	Stochastic sampling
Crossover type	Scattered
Mutation type	Adaptive

The evolution of the design parameters to the best individual is shown in Fig. (**5**). Through a simultaneous adjustment of five parameters the design parameters converge to $\lambda_b = \begin{bmatrix} 0.096\text{m} & 0.197\text{m} & 9.5 & 116.8 & 0.108\text{m} \end{bmatrix}^T$ and after 40 generations.

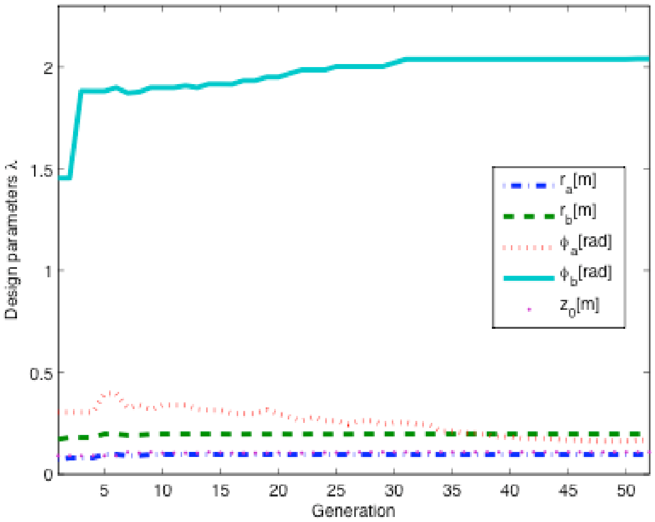


Fig. (5). Evolution of design parameters for GCI optimization with Gas.

The evolution of the GCI as a function of the generations is presented in Fig. (**5**), in which, the best GPI converged in 40 generations and its value is GCI=0.814. In Fig. (**6**), we can appreciate how the GGI varies when GCI is maximized, remembering that the condition number is non-dimensional; GGI suffers a small decrease along the evolution of the optimization; the total decrease is 3 units per meter, from 17 to 14 units per meter. When GCI is maximized, GGI holds constant. Thus, with increasing dexterity, the uniformity of dexterity inside the hyper-parallepiped workspace holds constant.

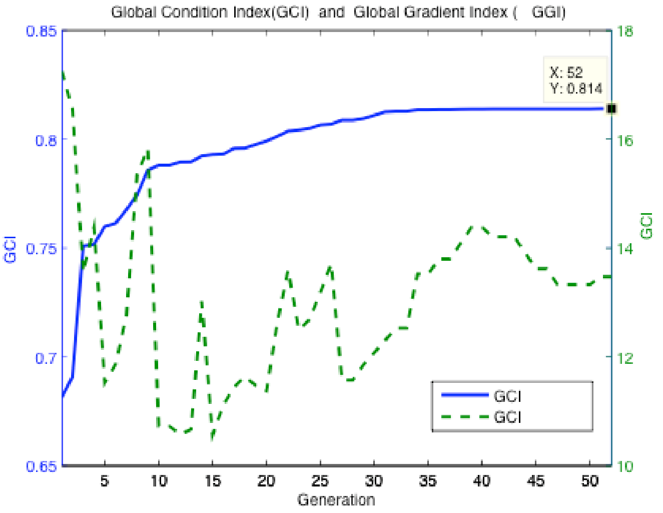


Fig. (6). GCI optimization with GAs and GGI behavior.

In the same way, for this optimization: the maximization of GCI, the evolution of the GCI, and the behavior of GPI is present in Fig. (**7**), when GCI is maximized, GPI decreases; therefore, when dexterity is maximized, the force transmission capability decreases.

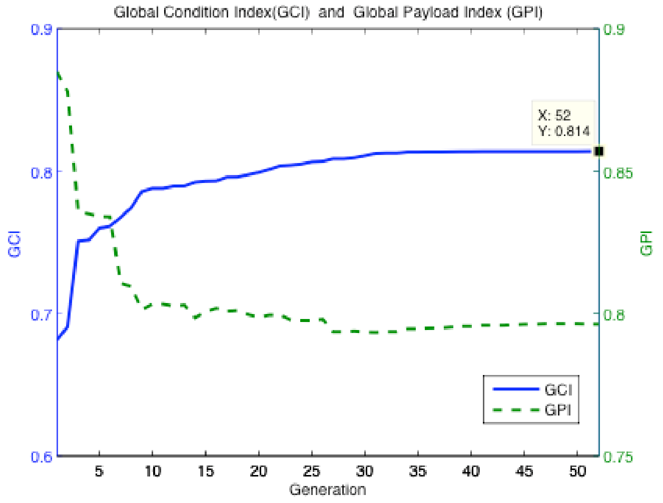


Fig. (7). GCI optimization with GAs and GPI behavior.

In Fig. (**8**) the hyper-parallepiped workspace w is inside of constant-orientation workspace of Stewart-Gough platform, the constant-orientation workspace is calculated with the optimal design parameters λ_b. Both volumes have constant orientation $\theta = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$.

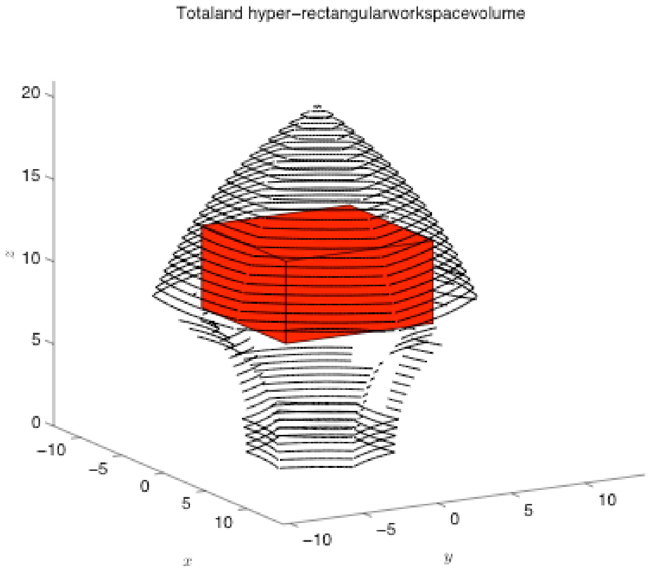
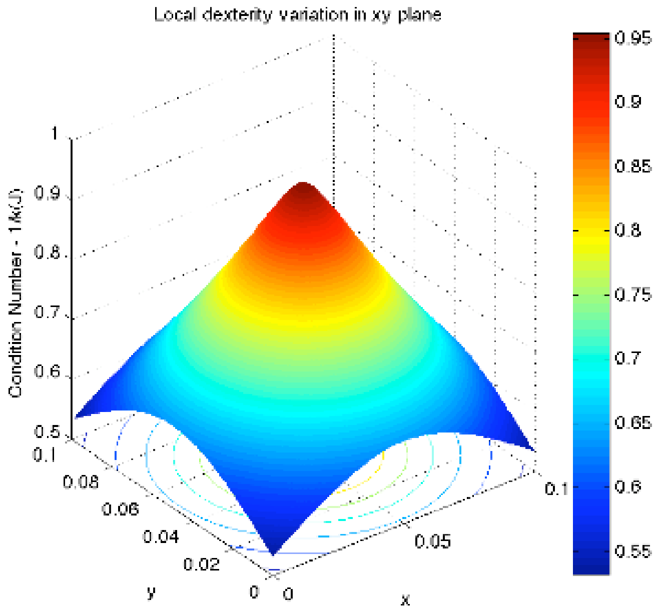


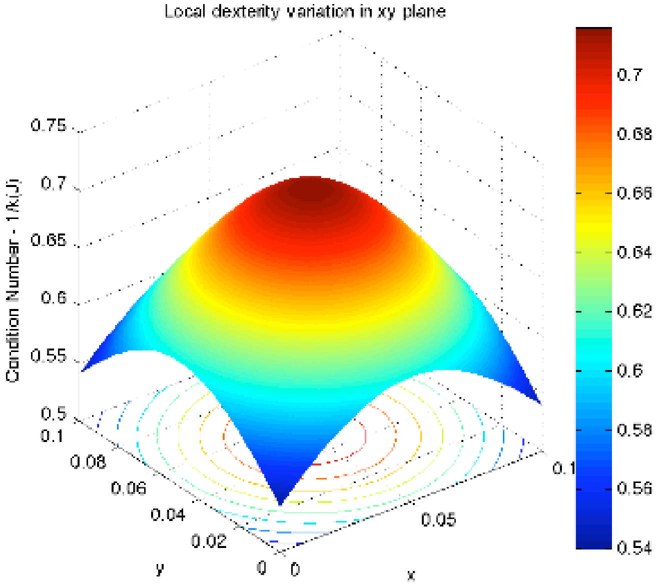
Fig. (8). Constant-orientation and hyper-parallepiped workspace.

The inverse condition number $1/k(\mathbf{J}_p)$ is evaluated to determine how the local dexterity varies inside hyper-parallepiped workspace. For this purpose, the local dexterity is calculated for the xy plane and the followings z coordinates: $z = z_0$ in Fig. (**9a**), and $z = z_0 + z_m$ in Fig. (**9b**). Althought the dexterity was maximized, the uniformity of the dexterity is very low the over the hyper-parallepiped workspace, this can be appreciated in Figs. (**9a** and **9b**). Hence, it is not sufficient maximize the GCI, because any small movement would reduce dexterity, causing configurations with high sensitive to error. This problem would be solved with dexterity maximization and dexterity variation minimization.



(**a**) xy plane for $z = z_0$

(Fig. 9) Contd…..



(**b**) *xy* plane for $z = z_0 + z_m$

Fig. (9). Local dexterity $1/k(\mathbf{J}(\mathbf{x},n_p))$ in *xy* plane for $z = z_0$

We can conclude that GCI, GGI, and CPI have conflicts between them. In the following section, the performance indices: GCI, GGI, and GPI will be optimized simultaneously to find the optimal design parameters such that satisfy optimal the condition.

5.2. Multi-Objective Optimization

Multi-objective optimization delivers a set of optimal solutions that corresponds to the Pareto front solution. This set of solutions is used in the design process to select one optimal design parameter.

$$\max_{\lambda}\{GCI(\lambda), GPI(\lambda)\} \quad \min_{\lambda}\{GGI(\lambda)\}$$

subject to

$$-r_a + r_b = 0.1\text{m}$$

$$r_a, r_b \in [0.05\text{m}, 0.2\text{m}] \qquad \varphi_a, \varphi_b \in [3, 117]$$

$$z_0 \in [0.05\text{m}, 0.25\text{m}]$$

$$\forall \quad \mathbf{x} \in w$$

$$w = [-0.05\text{m}, 0.05\text{m}] \times [-0.05\text{m}, 0.05\text{m}] \times$$

$$[-0.025\text{m}, 0.025\text{m}] \times [0] \times [0] \times [0] \qquad (25)$$

After some preliminary simulations, we use the parameters in Table **2** to run CENSGA.

All possible and optimal solutions in decision space are found considering the three different objective functions simultaneously. In Fig. (**10**), the Pareto front is shown in criterion space. Values outside of the Pareto front may be unreachable (unimplementable) or solutions with lower performance (not optimal).

Table 2. Parameters Used for Running CENSGA

Parameter	Setting
Population size	50
Maximum of generations	66
Encoding type	Real
Selection strategy	Tournament
Tournament size	2
Crossover type	Intermediate
Crossover ratio	1
Mutation type	Adaptive
Reduction factor r	0.7

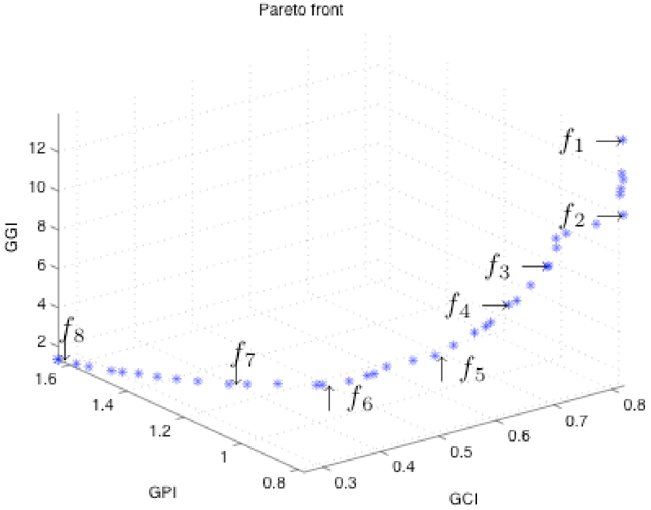


Fig. (10). MOO Pareto front in criterion space.

From the 43 different solutions obtained in MOO, we have selected eight solutions for each of the three objective functions (GCI, GPI, GGI), thus, f_i for $i = 1,\ldots,8$. In Table **3** the Pareto front solutions are presented in the criterion space.

Table 3. Pareto Front, Criterion Space

	GCI	GPI	GGI
f_1	0.8199	0.7808	13.9766
f_2	0.8223	0.7831	10.0887
f_3	0.7113	0.8207	8.1121
f_4	0.6594	0.8555	6.2877
f_5	0.5671	0.9278	3.9260
f_6	0.4487	1.0822	2.3368
f_7	0.3708	1.2519	1.8549
f_8	0.2670	1.6408	1.3313

In the same way, in Table **4**, the Pareto front f_i is given in decision space. It corresponds directly to the set of optimal design parameters for the Stewart-Gough platform.

Table 4. Pareto Front, Decision Space

	r_a[m]	r_b[m]	φ_a[]	φ_b[]	z_0[m]
f_1	0.0945	0.1941	3.09	116.79	0.120
f_2	0.0945	0.1939	3.07	116.97	0.117
f_3	0.0949	0.1944	3.19	116.79	0.153
f_4	0.0945	0.1951	4.85	112.03	0.160
f_5	0.0948	0.1949	4.58	115.08	0.193
f_6	0.0953	0.1949	6.30	110.77	0.237
f_7	0.0974	0.1976	10.56	89.68	0.249
f_8	0.0990	0.1981	12.11	12.08	0.249

We can observe how the kinematic proprieties vary along the Pareto front. This observation is useful for selecting design parameters from the optimal solution set.

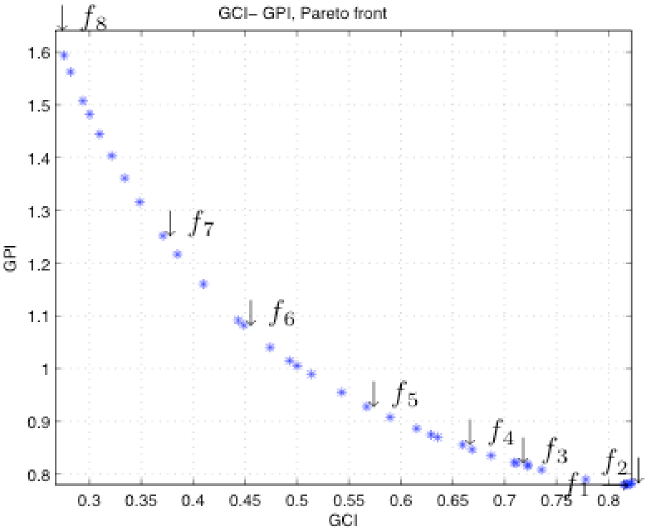


Fig. (11). MOO Pareto front, GCI and GPI.

In Fig. (**11**), when kinematic dexterity increases, the payload capability decreases. The ideal solution would be to maximize both proprieties, but for practical purposes, an intermediate solution must be chosen depending on the required working conditions. For example, if a high load must be carried by the manipulator and position accuracy is not preponderant, GPI must have maximum value, thus f_8 would be selected.

Another important consideration is the uniformity of kinematic dexterity to ensure a high positioning accuracy over the required workspace. In Fig. (**12**) the Pareto front is showed. When the kinematic dexterity increases, the uniformity of kinematic dexterity decreases due to the increment of GGI. Again, we have two conflicting requirements, hence, to select the design parameters from the optimal solution set, the use of an intermediate solution is recommended. Thus, it is clear that the selection of the solution with higher GCI would not be useful because the kinematic dexterity would decrease considerably in the hyper-parallepiped workspace limit as presented in Figs. (**9a** and **9b**) of section 5.1.

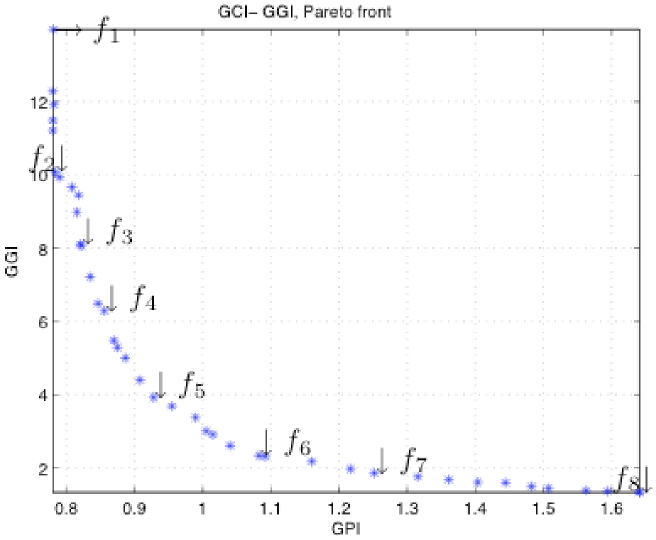


Fig. (12). MOO Pareto-optimal front, GCI and GGI.

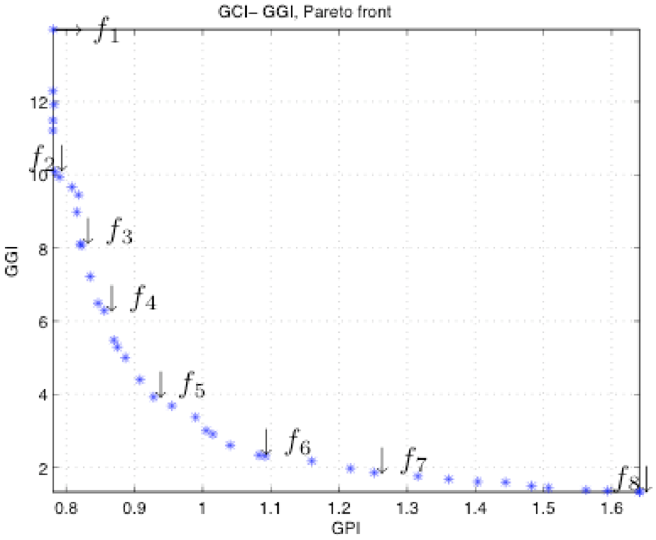


Fig. (13). MOO Pareto front, GPI and GGI.

In Fig. (**13**), we can see that increasing of payload capability also increases and the uniformity of kinematic dexterity over the required workspace. The optimal solution from Pareto front is f_8, considering only two kinematic proprieties: payload capability and uniformity of dexterity. However, for selecting the optimal design parameters kinematic dexterity also has to be considered.

6. CONCLUSION

This work presented a design procedure to reach kinematic parameters of a parallel manipulator, optimizing three global kinematic performance indices. Optimization design is a complex procedure because compulsory and, at the same time, contradictory objective functions have to be satisfied. Optimal criteria were established for the kinematic proprieties: payload capability, uniformity of the dexterity, and kinematic dexterity.

Consequently, the CENSGA was shown to be a robust optimization tool to reach optimal parameters for geometric configuration of parallel manipulators. The CENSGA provides the Pareto front, a set of optimal parameters to select the geometric configuration of the parallel manipulator in the design procedure.

Future work will relate to considering kinematic constraints of parallel manipulators in the model and optimization process. Among these kinematic constraints we can find: actuator limits, passive joint limits, and leg collision. In the same way, we will include additional geometric parameters to optimize different geometric architectures and not only standar form of the Stewart-Gough platform.

7. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of Fundação de Amparo à Pesquisa do Estado de São Paulo - FAPESP.

REFERENCES

[1] S. Advani, D. Giovannetti, and M. Blum, “Design of a hexapod motion cueing system for the nasa ames vertical motion simulator,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2002.

[2] Y. Ting, Y. Chen, and H. Jar, “Modeling and control for a Gough-Stewart platform CNC machine,” *Journal of Robotic Systems*, vol. 21, no. 11, pp. 609-623, 2004.

[3] Y. C. Sugahara, G. Hashimoto, K. Ceccarelli, M. Lim, and A. H.-O. Takanishi, “Experimental stiffness measurement of WL-16RII biped walking vehicle during walking operation,” *Journal of Robotic Systems and Mechatronics*, vol. 19, no. 3, pp. 272-280, 2007.

[4] M.Wapler, V. Urban, T.Weisener, J. Stallkamp, M. Durr, and A. Hiller, “A Stewart platform for precision surgery,” *Transactions of the Institute of Measurement and Control*, vol. 25, no. 4, pp. 329-334, 2003.

[5] Y. Lou, G. Liu, and Z. Li, “Randomized optimal design of parallel manipulators,” *Automation Science and Engineering, IEEE Transactions on*, vol. 5, no. 2, pp. 223-233, 2008.

[6] S. Bandyopadhyay, and A. Ghosal, “An algebraic formulation of kinematic isotropy and design of isotropic 6-6 Stewart platform manipulators,” *Mechanism and Machine Theory*, vol. 43, no. 5, pp. 591-616, 2008.

[7] X.-J. Liu, Z.-L. Jin, and F. Gao, “Optimum design of 3-DOF spherical parallel manipulators with respect to the conditioning and stiffness indices,” *Mechanism and Machine Theory*, vol. 35, no. 9, pp. 1257-1267, 2000.

[8] Y. X. Su, B. Duan, and C. H. Zhenh, “Genetic design of kinematically optimal fine tuning Stewart platform for large spherical radio telescope,” *Mehatronics*, vol. 11, pp. 821-835, 2001.

[9] Y. Zhang, and Y. Yao, “Kinematic Optimal Design of 6-UPS Parallel Manipulator,” in *Proceedings of the 2006 IEEE International Conference on Mechatronics and Automation*, 2006, pp. 2341-2345.

[10] E. Ottaviano, and M. Ceccarelli, “Optimal design of CAPAMAN (cassino parallel manipulator) with prescribed workspace,” *Robotica*, vol. 20, no. 2, pp. 159-166, 2002.

[11] J. P. Merlet, “Designing a parallel manipulator for a specific workspace,” *The International Journal of Robotics Research*, vol. 16, no. 4, pp. 545-556, 1997.

[12] R. Boudreau, and C. M. Gosselin, “La synthese d’une plate-forme de Gough-Stewart pour un espace atteignable prescrit,” *Mechanism and Machine Theory*, vol. 36, pp. 327-342, 2001.

[13] K. Miller, “Optimal design and modeling of spatial parallel manipulators,” *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 127-140, 2004.

[14] M. Ceccarelli, and C. Lanni, “A multi-objective optimum design of general 3R manipulators for prescribed workspace limits,” *Mechanism and Machine Theory*, vol. 39, no. 2, pp. 119-132, 2004.

[15] F. Hao, and J.-P. Merlet, “Multi-criteria optimal design of parallel manipulators based on interval analysis,” *Mechanism and Machine Theory*, vol. 40, no. 2, pp. 157-171, 2005.

[16] F. Lara, J. Rosario, and D. Dumur, “Experimental environment for modelling, analysis and simulation of Stewart platform manipulator,” *International Journal of Factory Automation, Robotics and Soft Computing*, vol. 1, pp. 90-95, 2008.

[17] C. Lanni, and M. Ceccarelli, “An optimzation problem algorithm for kinematic design of mechanisms for two-finger grippers,” *The Open Mechanical Engineering Journal*, vol. 4, no. 14, pp. 49-62, 2009.

[18] Z. Gao, D. Zhang, and Y. Ge, “Design optimization of a spatial six degree-of-freedom parallel manipulator based on artificial intelligence approaches,” *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 180-189, 2010.

[19] S. Bai, “Optimum design of spherical parallel manipulators for a prescribed workspace,” *Mechanism and Machine Theory*, vol. 40, no. 2, pp. 200-211, 2010.

[20] L.-W. Tsai, *Robot Analysis The Mechanics of Serial and Parallel Manipulators*. A Wiley-Interscince Publication: New York, 1999.

[21] B. Dasgupta, and T. S. Mruthyunjaya, “The Stewart platform manipulator: a review,” *Mechanism and Machine Theory*, vol. 35, no. 1, pp. 15-40, 2000.

[22] J. Angeles, and C. Gosselin, “A global performance index for the kinematic optimization of robotic manipulators,” *Journal of Mechanical Design*, vol. 113, no. 3, pp. 220-226, 1991.

[23] R. Kurtz, and V. Hayward, “Multiple-goal kinematic optimization of a parallel sphericalmechanism with actuator redundancy,” *IEEE Transactions on Robotics and Automation*, vol. 8, no. 15, pp. 644-651, 1992.

[24] S. H. Lee, J. H. Lee, B.-J. Yi, S. H. Kim, and Y. K. Kwak, “Optimization and experimental verification for the antagonistic stiffness in redundantly actuated mechanisms: a five-bar example,” *Mechatronics*, vol. 15, no. 2, pp. 213-238, 2005.

[25] J. Yoon, and J. Ryu, “Design, fabrication, and evaluation of a new haptic device using a parallel mechanism,” *IEEE/ASME Transactions on Mechatronics*, vol. 6, no. 3, pp. 221-233, 2001.

[26] X.-J. Liu, J.Wang, and G. Pritschow, “On the optimal kinematic design of the PRRRP 2-DOF parallel mechanism,” *Mechanism and Machine Theory*, vol. 41, no. 9, pp. 1111-1130, 2006.

[27] X.-J. Liu, J. Wang, and G. Pritschow, “Performance atlases and optimum design of planar 5R symmetrical parallel mechanisms,” *Mechanism and Machine Theory*, vol. 41, no. 2, pp. 119-144, 2006.

[28] C. Gosselin, “Dexterity indices for planar and spatial robotic manipulators”. In: *Proceedings. 1990 IEEE International Conference on*, in *Robotics and Automation, 1990.* 1990, vol. 1, 13-18, pp. 650-655.

[29] J. H. Holland, *Adaptation in Natural and Artificial Systems*. The MIT press Cambridge, MA, USA, 1992.

[30] Z. Yang, X. Gu, X. Liang, and L. Ling, “Genetic algorithm-least squares support vector regression based predicting and optimizing model on carbon fiber composite integrated conductivity,” *Materials & Design*, vol. 31, no. 3, pp. 1042-1049, 2010.

[31] M. Gen, and R. Cheng, *Genetic Algorithms & Engineering Optimization*. A Wiley-Interscience Publication: New York, 2000.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitista multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, pp. 182-197, 2002.

[33] K. Deb and T. Goel, *Lecture Notes in Computer Science*, vol. 1993/2001 of *Lecture Notes in Computer Science*, ch. Controlled

Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. Springer Berlin / Heidelberg, 2001, pp. 67-81.

[34] MathWorks, *Genetic Algorithm and Direct Search Toolbox TM, UsersGuide*. Matlab, 2009.

Received: March 09, 2010 Revised: June 11, 2010 Accepted: June 19, 2010