BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 53, No. 1, 2005

Multi criteria optimum design of manipulators

M. CECCARELLI*, G. CARBONE and E. OTTAVIANO

LARM: Laboratory of Robotics and Mechatronics DiMSAT – University of Cassino
Via G. Di Biasio 43, 03043 Cassino (Fr), Italy

Abstract. A suitable use of software packages for optimization problems can give the possibility to formulate design problems of robotic mechanical systems by taking into account the several aspects and behaviours for optimum solutions both in design and operation. However, an important issue that can be even critical to obtain practical solutions can be recognized in a proper identification and formulation of criteria for optimability purposes and numerical convergence feasibility.

In this paper, we have reported experiences that have been developed at LARM in Cassino by referring to the above-mentioned issues of determining a design procedure for manipulators both of serial and parallel architectures. The optimality criteria are focused on the well-recognized main aspects of workspace, singularity, and stiffness. Computational aspects are discussed to ensure numerical convergence to solutions that can be also of practical applications. In particular, optimality criteria and computational aspects have been elaborated by taking into account the peculiarity and constraint of each other. The general concepts and formulations are illustrated by referring to specific numerical examples with satisfactory results.

Key words: robotics, manipulators, performance criteria, optimum design.

1. Introduction

Robotized manipulation is more and more used in industrial applications and even in non-industrial environments manipulators are needed more and more to help human beings and/or execute manipulative tasks. The duality between serial and parallel manipulators is not anymore understood as a competition between the two kinematic architectures. The intrinsic characteristics of each architecture make each architecture as directed to some manipulative tasks more than an alternative to the other. The complementarities of operation performance of serial and parallel manipulators make them as a complete solution for manipulative operations. The differences but complementarities in their performance have given the possibility in the past to treat them separately, mainly for design purposes. Several analysis results and design procedures have been proposed in a very rich literature in the last two decades. Significant works on the topics can be considered the pioneer papers [1–8] and more recently the papers [9–11], just to cite few references in a very rich literature. Only recently, it has been possible to consider simultaneously several design aspects in design procedures for manipulators. Modern design procedures make use more and more of the formulation of optimization problems that can be solved by using well-established mathematical techniques in commercial software packages.

At LARM in Cassino, since the beginning of 90's a research line has been dedicated to the development of analysis formulation of manipulator performances, [12–17], that could be used in proper optimization problems by taking advantage of the peculiarity of the solving techniques in commercial softwares, [18–23]. Recent results are reported in [24] as regarding serial manipulators, and in [25] referring to parallel manipulators, just to cite illustrative experiences. However, since the manipulative characteristics are fundamental for the operation and design of both manipulator architectures, the authors have attempted analysis procedures that could be used with few adjustments and no great computational efforts in the performance analysis of both serial and parallel architectures. The several experiences have been summarized in the recent textbook [26] in which the analysis of workspace, singularity, and stiffness has been attached in a unified approach. Following this idea, in this paper we have attempted to treat the design problem of serial and parallel manipulators in a unified formulation, since the optimality design criteria can be differentiated only in characterization of the results and not so much in the numerical algorithms. Therefore, in this paper basic characteristics for manipulation purposes, such as workspace, singularity, and stiffness, are overviewed with numerical evaluation procedures that are useful in a design optimization problem for both serial and parallel manipulators. The feasibility of the approach and proposed formulation has been tested and illustrative examples are reported, also with the aim to clarify the computational efforts.

1.1. The problem and its formulation. Manipulators are said useful to substitute/help human beings in manipulative operations and therefore their basic characteristics are usually referred to human manipulation performance aspects.

*e-mail: ceccarelli@unicas.it

A well-trained person is usually characterized for manipulation purpose mainly in terms of positioning skill, arm mobility, arm power, movement velocity, and fatigue limits. Similarly, robotic manipulators are designed and selected for manipulative tasks by looking mainly to workspace volume, payload capacity, velocity performances, and stiffness. Therefore, it is quite reasonable to consider those aspects as fundamental criteria for manipulator design. But generally since they can give contradictory results in design algorithms, a formulation as multi-objective optimization problem can be convenient in order to consider them simultaneously. Thus, the optimum design of manipulators can be formulated as

$$\min \mathbf{F}(\mathbf{X}) = \min\left[f_1(\mathbf{X}), f_2(\mathbf{X}), \dots f_N(\mathbf{X})\right]^T \tag{1}$$

subjected to

$$\mathbf{G}(\mathbf{X}) < 0; \tag{2}$$

$$\mathbf{H}(\mathbf{X}) = 0, \tag{3}$$

where $^{\mathrm{T}}$ is the transpose operator; $\mathbf{X}$ is the vector of design variables; $\mathbf{F}(\mathbf{X})$ is the vector of objective functions that express the optimality criteria, $\mathbf{G}(\mathbf{X})$ is the vector of constraint functions that describes limiting conditions, and $\mathbf{H}(\mathbf{X})$ is the vector of constraint functions that describes the design prescriptions. In particular, optimality criteria for manipulator design can be identified in performance evaluations regarding with positioning and orientating capability, velocity response, and static behaviour. Positioning and orientation capability can be evaluated by computing position and orientation workspaces that give the reachable regions by the manipulator extremity as function of the mobility range of the manipulator joints. Position workspace refers to reachable points by a reference point on manipulator extremity, and orientation workspace describes the angles that can be swept by reference axes on manipulator extremity.

Thus, an objective function f_1 can be formulated as regarding with a numerical evaluation f_{PW} of the position workspace as

$$f_1 = f_{PW}. \tag{4}$$

Similarly, a numerical evaluation f_{OW} of the orientation workspace can be used as objective function as

$$f_2 = f_{PW}. \tag{5}$$

Velocity response can be evaluated by looking at the velocity mapping that can be described by the Jacobian of the manipulator. The Jacobian is also useful to identify singular configurations (singularities) of a manipulator at which degrees of freedom are lost or gained producing undesirable motion uncertainties or self-motions that should be avoided in a controlled movement. Thus, Jacobian evaluation f_J can be used as objective function as

$$f_3 = f_J. \tag{6}$$

Static behaviour can be evaluated by computing the stiffness characteristics that are responsible also for the accuracy of the manipulative operation. Therefore, compliant response can be conveniently used as optimality design criterion, when similarly to the workspace capability the compliance of a manipulator is evaluated through the position and orientation counterparts. Thus, an evaluation f_{ST} of linear compliant displacements can be used as objective function as

$$f_4 = f_{ST}, \tag{7}$$

as well as an evaluation f_{SO} of the angular compliant displacements as

$$f_5 = f_{SO}. \tag{8}$$

Thus, the multi-objective function $\mathbf{F}$ is formulated with computer-oriented algorithms when its components f_i ($i = 1, \dots, 5$) are computed numerically through suitable analysis procedures. It is worth to note that the above-mentioned split into translational and orientational aspects can be adopted also for Jacobian analysis and however it has been thought mainly for simplifying the computational efforts and for interpretation aims, without neglecting the coupling effects due to spatial design and operation of manipulators.

Similarly, the constraint functions $\mathbf{G}$ and $\mathbf{H}$ can be formulated by using suitable evaluation of design and operation constraints as well as those additional constraints that are needed for computational issues. Thus, the problem of achieving optimal results from the formulated multi-objective optimization problem consists mainly in two aspects, namely to choose a proper numerical solving technique and to formulate the optimality criteria with computational efficiency. Indeed, the solving technique can be selected among the many available, even in commercial software packages, by looking at a proper fit and/or possible adjustments to the formulated problem in terms of number of unknowns, non-linearity type, and involved computations for the optimality criteria and constraints. On the other hand, the formulation and computations for the optimality criteria and design constraints can be conceived and performed by looking also at the peculiarity of the numerical solving technique.

Those two aspects can be very helpful in achieving an optimal design procedure that can give solutions with no great computational efforts and with possibility of engineering interpretation and guide. Since the formulated design problem is intrinsically high no-linear with objective functions that can be competitive the solution will be obtained when the numerical evaluation of the tentative solutions due to the iterative process converges to a solution that can be considered optimal within the explored range Thus, one cannot ensure the character of global optimum of the solution that has been obtained with a numerical search. A solution can be always considered as a local optimum, and by performing several optimization calculations as function of initial guesses it is only possible to characterize this local optimum within the inspected range of the design parameters. This last remark makes clear once more the influence of suitable formulation with computational efficiency for the involved criteria and constraints in order to have a design procedure, which is significant from engineering viewpoint and numerically efficient.

2. Performance evaluation for optimality criteria

Once the numerical technique is chosen or is advised for solving the proposed multi-objective optimization problem, the main efforts can be addressed to the formulation of common algorithms for numerical evaluation of optimality criteria and design procedure constraints. In the following, main aspects are overviewed by emphasizing the common numerical evaluations both for serial and parallel manipulators in terms of workspace, singularity, and stiffness.

2.1. Workspace evaluation. The workspace is one of the most important Kinematic properties of manipulators, even by practical viewpoint because of its impact on manipulator design and location in a workcell. A general numerical evaluation of the workspace can be deduced by formulating a suitable binary representation of a cross-section. A cross-section can be obtained with a suitable scan of the computed reachable positions and orientations, once the direct kinematic problem has been solved to give the positions and orientations as functions of the kinematic input joint variables. A binary matrix P_{ij} can be defined in the cross-section plane for a cross-section of the workspace as follows: if the (i, j) grid pixel includes a reachable point, then $\mathrm{P}_{ij} = 1$; otherwise $\mathrm{P}_{ij} = 0$, as shown in Fig. 1.

For example, one can consider a cross-section at a given value of Z-Coordinate, then a point in the grid is indicated as P_{ij}, with i along X-axis and j along the Y-axis, namely,

$$i = \left[\frac{x + \Delta x}{x}\right], \qquad j = \left[\frac{y + \Delta y}{y}\right] \tag{9}$$

where i and j are computed as integer numbers. Therefore,the binary mapping for a workspace cross-section can be given as

$$\mathrm{P}_{ij} = \begin{cases} 0 \text{ if } \mathrm{P}_{ij} \notin W(H) \\ 1 \text{ if } \mathrm{P}_{ij} \in W(H) \end{cases} \tag{10}$$

where $W(H)$ indicates workspace region; $\in$ stands for 'belonging to' and $\notin$ is for 'not belonging to'. In addition, the proposed binary representation is useful for a numerical evaluation of the position workspace by computing the cross-sections areas A_z as

$$\mathrm{A}_z = \sum_{i=1}^{i_{\max}} \sum_{j=1}^{j_{\max}} (\mathrm{P}_{ij}\ \Delta x\ \Delta y) \tag{11}$$

and finally, the workspace volume V can be computed with respect to the number of slices n_z along Z-axis, Fig. 1, as

$$\mathrm{V} = \sum_{n_z} \mathrm{A}_z\ \Delta z \tag{12}$$

Similarly, a numerical evaluation of orientation workspace can be carried out by using the formulation of Eqs (9) to (12) in order to compute the corresponding orientation performance measures cross-sections areas A_φ, and orientation workspace volume V_φ, when a 3D representation of the orientation capability is obtained by using three angular coordinates as Cartesian coordinates. One can use Eqs (9) to (12) in order to eval uate any cross-section by properly adapting the formulation to the scanning cross-section plane and intervals. Therefore, the optimum design problem with objective functions $f_1(\mathrm{X})$ and $f_2(\mathrm{X})$ can be formulated as finding the optimal design parameters values to obtain the position and orientation

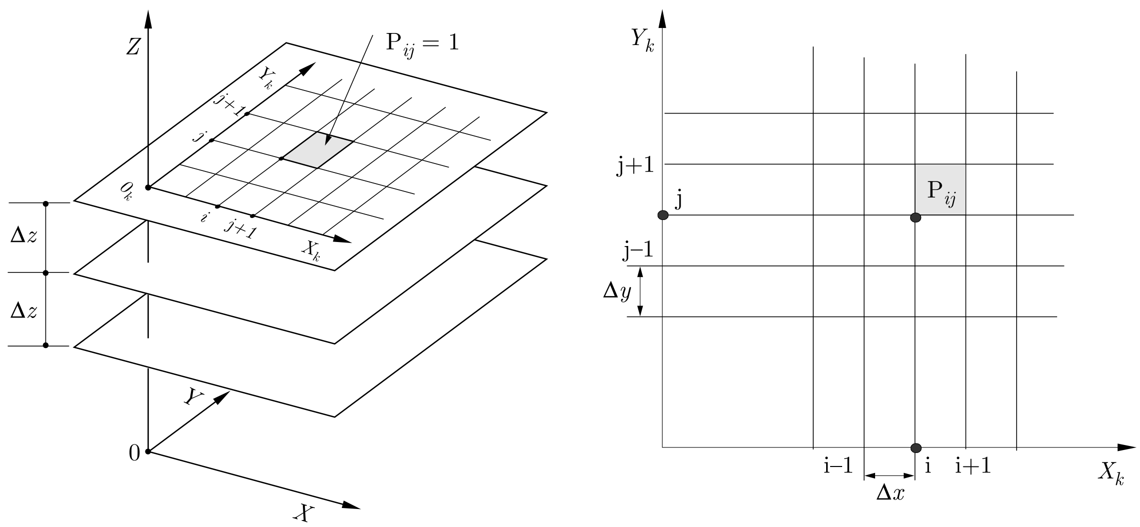


Fig. 1. A general scheme for binary representation and evaluation of manipulator workspace

workspace volumes that are as close as possible to prescribed ones. This can be formulated in suitable form as

$$f_1(X) = \left|1 - \frac{V_{pos}}{V'_{pos}}\right| \tag{13}$$

$$f_2(X) = \left|1 - \frac{V_{or}}{V'_{or}}\right| \tag{14}$$

where | . | is the absolute value. The workspace optimization problem can be also subject to design constraints such as

$$|x_{\max} - x_{\max}| \leq 0;\ |y_{\max} - y_{\max}| \leq 0;\ |z_{\max} - z_{\max}| \leq 0, \tag{15}$$

$$|\varphi_{\max} - \varphi_{\max}| \leq 0;\ |\Psi_{\max} - \Psi_{\max}| \leq 0;\ |\theta_{\max} - \theta_{\max}| \leq 0, \tag{16}$$

where the left-hand values correspond to the computed volume V and prime values describe the prescribed parallelepiped volume V' by using the extreme reaches.

3. Singularity analysis

Design requirements and operation feasibility can also be focused conveniently on a free singularity condition. In fact, it is desirable to ensure a given workspace volume within which the manipulator extremity can be movable, controllable, and far enough from singularities.

The instantaneous relationship between the velocity in the Cartesian Space and active joint velocity can be expressed as

$$\mathrm{A}\theta = \mathrm{B}t \tag{17}$$

where A and B are two Jacobian matrices of a manipulator; $\dot{\theta}$ is the vector of joint rates, and t is the twist array containing the linear velocity vector $\boldsymbol{\nu}$ and the angular velocity vector $\boldsymbol{\omega}$).

Usually condition of singular configurations can be represented by surfaces in the Configuration Space and they can be obtained by vanishing the determinant of the two Jacobian matrices A and B. In particular, matrix A gives the inverse kinematics singularities; and B gives the direct kinematics singularities. Direct kinematics singularities are inside the workspace and in such configurations a manipulator loses its rigidity, becoming locally movable, even if the actuated joints are locked.

The concept of singularity has been extensively studied and several classification methods have been defined. Manipulator singularities can be classified into three main groups. The first type of singularity occurs when a manipulator reaches internal or external boundaries of its workspace and the output link loses one or more d.o.f.s. The second type of singularity is related to those configurations in which the output link is locally movable even if all the actuated joints are locked. This is called 'self-motion'. The third type is related to linkage parameters and occurs when both the first and second types of singularities are involved. Singularities can also be differentiated as configuration singularities, architecture singularities, and formulation singularities. The first type of singularity is related to particular configurations of the manipulator. Architecture singularities are caused by certain architectures; they do not depend on the specific configuration of the manipulator, and they are inherent to the kinematic design of a manipulator. Formulation singularities are due to the adopted model and formulation for numerical analysis and they can be avoided simply by changing formulation method.

In summary, manipulator singularities arise whenever A, B, or both, become singular. Thus, a distinction can be made among three types of singularities, by considering Eq. (17), namely:

– the first type of singularity occurs when A becomes singular but B is invertible, being

$$\det \mathrm{A} = 0 \quad \text{and} \quad \det \mathrm{B} = 0 \tag{18}$$

– the second type of singularity occurs only in closed kinematic chains and arises when B becomes singular but A is invertible, i.e.

$$\det \mathrm{A} \neq 0 \quad \text{and} \quad \det \mathrm{B} \neq 0 \tag{19}$$

– the third type of singularity occurs when A and B are simultaneously singular, while none of the rows of B vanish. Under a singularity of this type, configurations arise for which the movable plate can undergo finite motions even if the actuators are locked or, equivalently, it cannot resist forces or moments into one or more directions over a finite portion of the workspace, even if all actuators are locked. A finite motion can be very small but even very large to be considered sometimes as an extra d.o.f. for specific manipulator configurations.

In general, in parallel manipulators each leg is connected to the moving platform through articulation points B_i by means of spherical joints. The determinants of A and B are a function of the shape and size of the moving platform, magnitude, and direction of the vectors $\boldsymbol{d}_i$ of each leg, and unit vectors $\boldsymbol{e}_i$ of vectors $\boldsymbol{b}_i$ from the center point of the moving platform to the connecting joints. This can be formulated as

$$\mathbf{B} = \begin{bmatrix} \mathbf{d}_1 & (\mathrm{R}\mathbf{b}_1) \times \mathbf{d}_1 \\ \cdots & \cdots \\ \mathbf{d}_6 & (\mathrm{R}\mathbf{b}_6) \times \mathbf{d}_6 \end{bmatrix} \tag{20}$$

$$\mathbf{A} = \begin{bmatrix} \mathbf{d}_1 \times \mathrm{R}\mathrm{r}_\mathrm{p}\mathbf{e}_1 & 0 \\ \cdots & \cdots \\ 0 & \mathbf{d}_6 \times \mathrm{R}\mathrm{r}_\mathrm{p}\mathbf{e}_6 \end{bmatrix} \tag{21}$$

in which R is the rotation matrix of the moving platform with respect to the base frame. The above-mentioned expressions can be representative also of the Jacobian of serial manipulators when the vector d_i is considered to be the i-th link vector and the vector $\mathrm{R}\mathrm{r}_\mathrm{p}\mathrm{e}_i = \mathrm{R}b_i$ is the $(1+1)$-th link vector.

Because of the above-mentioned expressions the Jacobian matrix is pose dependent and non-isotropic. Consequently, performances such as rigidity, velocities, and forces, which can be expressed as functions of the Jacobian are pose-dependent and therefore it is important to consider the Jacobian in a rational design procedure, also because of those influences. Indeed, one can propose an objective function f_3 that can be deduced by analyzing the analytical expression of the determinants of matrices A and B in the form

$$f_3(\mathrm{X}) = -\frac{\min(\det \mathrm{A}) \min(\det \mathrm{B})}{|\det \mathrm{A}_0|\ |\det \mathrm{B}_0|} \tag{22}$$

that will take into account all the situations in a singularity analysis, when the initial guess values A_0 and B_0 are considered. It is worth noting that the determinants of A_0 and B_0 must be chosen not equal to zero.

3.1. Stiffness evaluation. Stiffness and accuracy of a robotic architecture are strongly related to each other since positioning and orientating errors are due to compliant displacements and construction and assembling errors. The last errors can be evaluated by a kinematic analysis (calibration) by considering uncertainties in the kinematic parameters due to tolerances of construction and assembling of the robotic manipulator chain.

The stiffness properties of a manipulator can be defined through a 6×6 matrix that is called 4 'Cartesian stiffness matrix K'. This matrix gives the relation between the vector of the compliant displacements $\Delta\mathbf{S} = (\mathrm{S}x, \mathrm{S}y, \mathrm{S}z, \mathrm{S}\varphi, \mathrm{S}\psi, \mathrm{S}\theta)$ occurring at the movable plate when a static wrench $\mathbf{W} = (\mathrm{F}_x, \mathrm{F}_y, \mathrm{F}_z, \mathrm{T}_x, \mathrm{T}_y, \mathrm{T}_z)$ acts upon it, and $\mathbf{W}$ itself in the form

$$\mathbf{W} = \mathrm{K}\ \Delta\mathbf{S} \tag{23}$$

The stiffness matrix can be numerically computed by defining a suitable model of the manipulator, which takes into account the lumped stiffness parameters of links and motors. Each spring coefficient k_i refers to the sum of the lumped stiffness parameters of the motor and leg structure along the axial direction of the i-leg link. The coefficient K_{Ti} is the torsion stiffness parameter of the joint for each link taking into account of the angular compliance of joints and actuators. It is well known that the stiffness matrix is configuration dependent. Therefore, it must be computed as a function of input parameters, which are the strokes of linear actuators or angles of revolute actuators.

A 6×6 stiffness matrix K can be derived through the composition of suitable matrices. The first matrix $\mathrm{C_F}$ gives all the wrenches $\mathbf{W_L}$, acting on manipulator links when a wrench $\mathbf{W}$ acts on the manipulator extremity according to the expression

$$\mathbf{W} = \mathrm{C_F}\ \mathbf{W}_\mathrm{L} \tag{24}$$

with the matrix $\mathrm{C_F}$ representing the force transmission capability of the manipulator mechanism.

The second matrix $\mathrm{K_p}$ gives the possibility to compute the vector $\Delta\boldsymbol{v}$ of all the deformations of the links when each wrench $\mathbf{W_{Li}}$ on a i-th link given by $\mathbf{W}$, acts on the legs according to

$$\mathbf{W}_\mathrm{L} = \mathrm{K_p}\ \Delta\boldsymbol{v} \tag{25}$$

with the matrix Kp grouping the spring coefficients of the deformable components of a manipulator structure.

The third matrix $\mathrm{C_K}$ gives the vector $\Delta\boldsymbol{S}$ of compliant displacements of the manipulator extremity due to the displacements of the manipulator links, as expressed in the form

$$\Delta\boldsymbol{v} = \mathrm{C_K}\ \Delta\boldsymbol{S} \tag{26}$$

Therefore, the stiffness matrix K can be computed as

$$\mathrm{K} = \mathrm{C_F K_p C_K} \tag{27}$$

The stiffness matrix K can also be used to compute accuracy performances. In fact, the vector of compliance displacements $\Delta\boldsymbol{S}$ can be computed by using Eq. (23) once the matrix K is determined when a static wrench acting on the movable plate is given.

From the above-mentioned considerations two objective functions that take into account stiffness performances can be defined as

$$f_4(\mathbf{X}) = \left|1 - \frac{\Delta\boldsymbol{S}_\mathrm{d}}{\Delta\boldsymbol{S}_\mathrm{g}}\right| \tag{28}$$

$$f_5(\mathbf{X}) = \left|1 - \frac{\Delta\boldsymbol{S}_\mathrm{d}}{\Delta\boldsymbol{S}_\mathrm{d}}\right| \tag{29}$$

where $\Delta\boldsymbol{S}_d$ and $\Delta\boldsymbol{S}_g$ are compliant displacements along X, Y, and Z-axes, $\Delta\boldsymbol{R}_d$ and $\Delta\boldsymbol{R}_g$ are compliant rotations about φ, θ and ψ, d and g stand for design and given values, respectively. Criterion f_4 and f_5 of Eqs. (28) and (29) can be considered separately or in a single objective function component, according to the specific requirements. But this formulation needs

$$\det K \neq 0 \tag{30}$$

that can be used as additional constraint.

4. Numerical examples

The practical feasibility of the proposed optimum design procedure has been experienced at LARM with several design tests that refer also to prototypes of manipulator architectures for robotic systems [27]. In the following, two specific examples are discussed by illustrating numerical results that clarify the practical feasibility of the design procedure as applied to the cases of a 6R PUMA-like robot and a CaPaMan (Cassino Parallel Manipulator) Kinematic design, when numerical analyses are differentiated but the numerical evaluations are computed with same formulations.

There is a number of alternative methods to solve numerically a multi-objective optimization problem as reported in [28]. In particular, in the following examples the proposed multi-objective optimization design problem has been solved by considering the min-max technique of the Matlab Optimization Toolbox [29] that makes use of a scalar function of the vector function $\mathbf{F}(\mathbf{X})$ to minimize the worst case values among the objective function components. Using Sequential Quadratic Programming that is successfully used for solving optimization problems with non-linear objective functions and constraints and several variables can search the minimum value. This numerical procedure works in such a way that at each step k a solution is found along a search direction δ with variable update φ. The iteration continues until the vector of variables converges. The numerical procedure has been developed so that the formulation for the manipulator design has been easily included within the solving procedure for the optimization problem by using the facilities of the Optimization Toolbox of Matlab [29], which permits an easy arrangement for an optimum design with analytical expressions. The herein used numerical procedure for the solution of a multi-objective optimization design problem is outlined in the flowchart of Fig. 2.

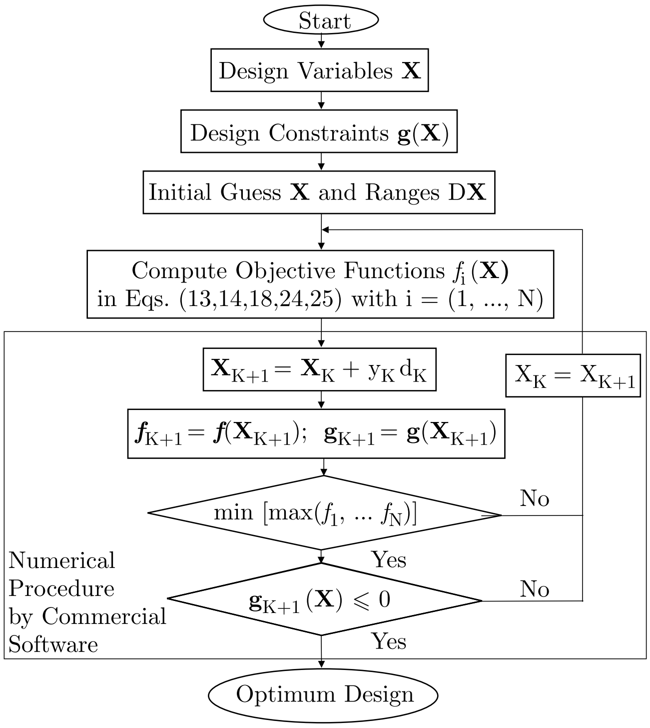


Fig. 2. A flowchart for optimum design procedure both for serial and parallel manipulators

4.1. A 6R serial manipulator A 6 dofs PUMA-like manipulator has been considered to test the engineering feasibility of the above-mentioned formulation for optimum design of manipulators as specifically applied to serial architectures. Simplified stiffness models for a PUMA-like manipulator are shown in Fig. 3. In the model of Fig. 3a, the stiffness properties of links and motors have been considered as lumped parameters and modeled as linear springs and torsion springs respectively. A further simplification can be obtained by considering the links as rigid. In this case, the stiffness model consists of the six lumped stiffness parameters of the motors K_{T1} to K_{T6} as shown in Fig. 3b. Moreover, if the only contributions to the overall compliance are given by the motor compliances, the Cartesian stiffness matrix K can be computed through Eq. (23) with

$$\mathrm{C_F} = \mathrm{J}^{-t};\ \mathrm{C_K} = \mathrm{J}^{-1} \tag{31}$$

where J is the Jacobian matrix of the PUMA-like robot; the matrix $\mathrm{K_P}$ in Eq. (23) can be computed as a diagonal matrix of the lumped stiffness parameters of the motors of PUMA-like robot. The stiffness matrix of the PUMA-like robot that can be computed through Eqs. (27) and (31) as function of the input angles α_i (with $i = 1, \ldots, 6$). The expressions of the input angles α_i can be computed as function the position and orientation of the end-effector (x, y, z, ϕ, ψ, θ) from the well-known inverse Kinematics of PUMA-like robot.

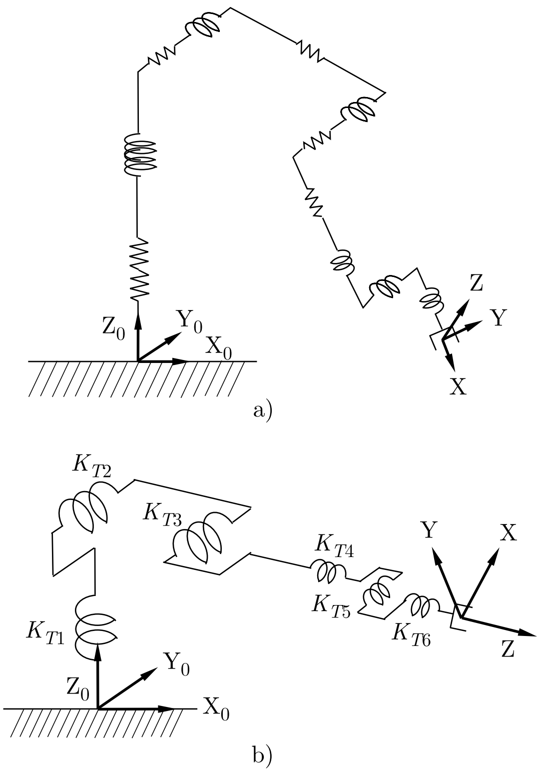


Fig. 3. Stiffness models for the PUMA-like robots: a) a model with linear and torsion springs; b) a model with only torsion springs

A singularity condition can be stated as in Eq. (22) where A and B can be obtained from the Jacobian matrix J. Position and orientation workspace volumes can be conveniently evaluated by using the well-known closed-form Kinematics of the Puma manipulator.

Results of the proposed design procedure as applied to the PUMA-like architecture are reported in Figs. 4–6 and Tables 1 and 2. In particular, Fig. 4 shows the evolution of the objective functions versus number of iterations. The sum $\boldsymbol{F}$ of all the objective functions converges to a value that is about 10% of the initial value after 100 iterations approximately. Figure 5 shows the evolution of the design parameters versus the number of iterations. Figure 6 shows the evolution of the design constraint versus the

Table 1
Design parameters of the optimum designed PUMA-like manipulator

	a_2 (mm)	a_3 (mm)	d_3 (mm)	d_4 (mm)	Δa_1 (deg)	Δa_2 (deg)	Δa_3 (deg)	Δa_4 (deg)	Δa_5 (deg)	Δa_6 (deg)
Initial guess	431.8	20.3	125.4	431.8	180	180	180	90	90	90
Optimal value	1000.0	100.0	269.6	1000.0	180	180	180	90	90	90

Table 2
Design characteristics of optimum solution for PUMA-like manipulator of Table 1

	Δx (mm)	Δy (mm)	Δz (mm)	$\Delta\phi$ (mm)	$\Delta\psi$ (mm)	$\Delta\theta$ (mm)	Sx (mm)	Sy (mm)	Sz (mm)	$S\phi$ (mm)	$S\psi$ (mm)	$S\theta$ (mm)
Initial guess	529.2	472.4	625.0	180	180	180	1.70	1.30	1.40	0.01	0.08	0.01
Optimal value	1305	1200	1485	180	180	180	1.50	1.90	1.60	0.01	0.11	0.01

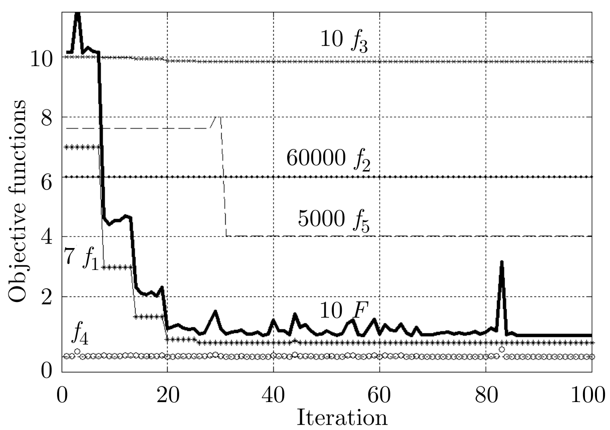


Fig. 4. Evolution of the function $\boldsymbol{F}$ and its components versus number of iterations for the example of PUMA-like robot optimal design: position workspace volume as f_1; orientation workspace volume as f_2; singularity condition as f_3; compliant displacements and rotations as f_4 and f_5

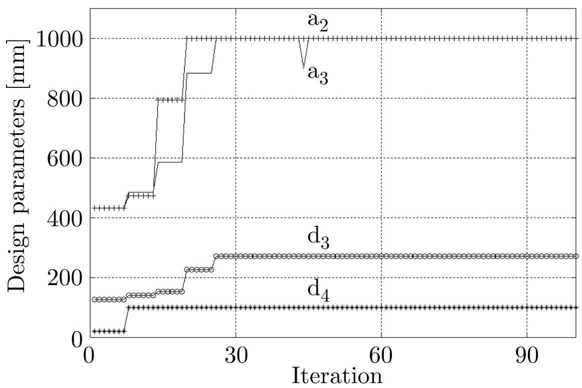


Fig. 5. Evolution of design constraints versus number of iterations for the example of PUMA-like robot optimal design

number of iterations. Table 1 shows the initial guess and the optimal values of design parameters for the PUMA-like manipulator. Table 2 shows the main characteristics of the PUMA-like manipulator for the initial guess and the optimal values of design parameters that are reported

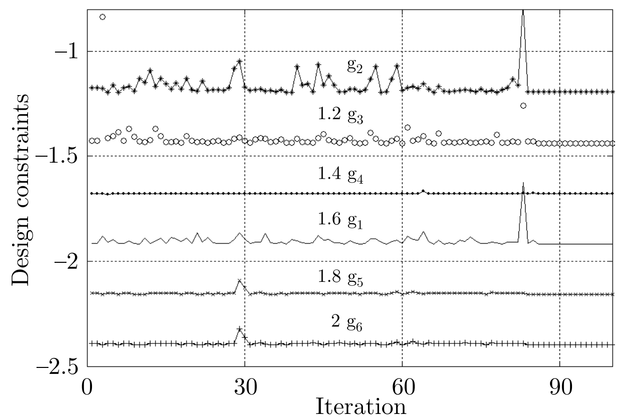
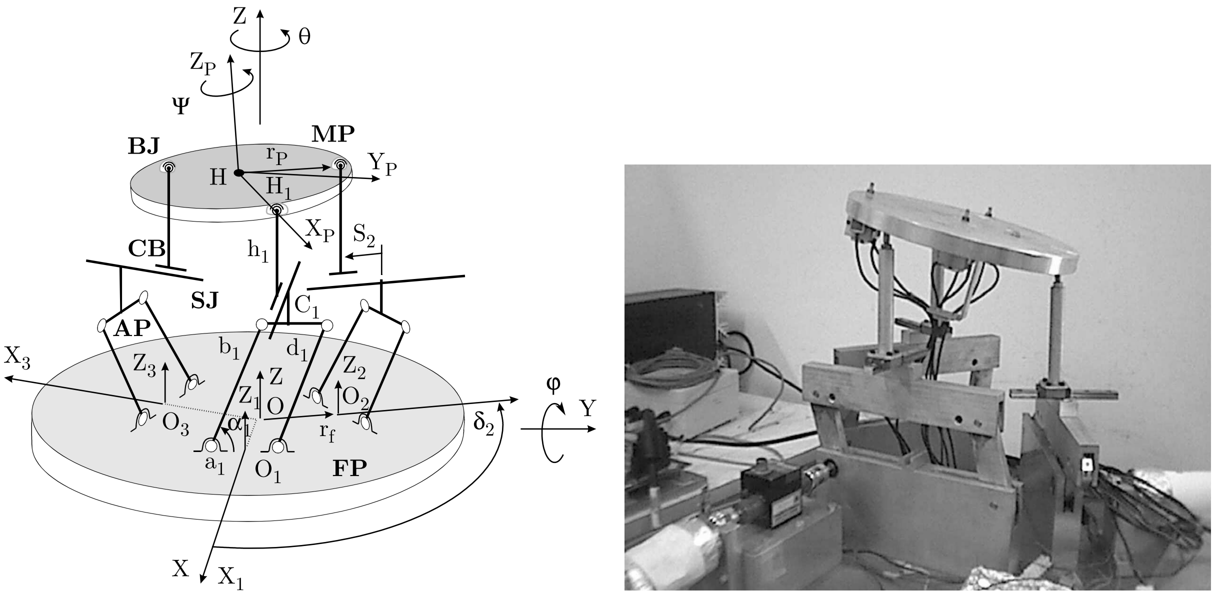


Fig. 6. Evolution of design parameters versus number of iterations for the example of PUMA-like robot optimal design

in Table 1. In Table 1 the initial guess values have been chosen equal to a real robot PUMA 562. The optimal values for the link lengths a_2, a_3, d_3, d_4 are considerably different from the initial guess. In particular, they are bigger since their bigger sizes increase the workspace. The ranges of the input angles $\Delta\alpha_1$ to $\Delta\alpha_6$ have remained unchanged because their initial values are close to the optimal ones. Comparing the initial and optimal values in Table 2 one can note that the position workspace is at least doubled while the orientation workspace has remained unchanged. The stiffness performance is still within the desired value even if the link lengths are more than two times bigger.

4.2. A 3 d.o.f. parallel manipulator. The CaPaMan manipulator has been considered to test the engineering feasibility of the above-mentioned formulation for optimum design of manipulators as specifically applied to parallel architectures. CaPaMan architecture has been conceived at LARM in Cassino, where a prototype has been built for experimental activity. Indeed, by using the existent prototype, simulations have been carried out also to validate the proposed optimum design by considering several guess solutions and imposing workspace and stiffness characteristics of the built prototype.



a) b)

Fig. 7. CaPaMan (Cassino Parallel Manipulator) design: a) Kinematic diagram; b) a built prototype at LARM

A schematic representation of the CaPaMan manipulator is shown in Fig. 7a), and the prototype is shown in Fig. 7b).

Position and orientation workspace volumes can be conveniently evaluated by using the closed-form Kinematics of the CaPaMan manipulator.Singularity analysis for CaPaMan manipulator has been reported in [30] and matrices A and B have been formulated in a form that is useful also for the proposed optimality criterion.

Stiffness analysis of CaPaMan has been reported in [31]. By modeling each leg of CaPaMan as in Fig. 8, the stiffness matrix of CaPaMan can be derived as

$$\mathrm{K_{CaPaMan}} = \mathrm{M_{FN}}\ \mathrm{K_P}\ \mathrm{C_P^{-1}}\ \mathrm{A_d^{-1}} \tag{32}$$

where $\mathrm{M_{FN}}$ is a 6×6 transmission matrix for the static wrench applied on H to points $H_1 H_2$ and H_3 of each leg; $\mathrm{K_p}$ is a 6×6 matrix with the lumped stiffness parameters of the 3 legs; $\mathrm{A_d}$ is a 6×6 matrix obtained by using the Direct Kinematics of the CaPaMan; $\mathrm{C_p}$ is a 6×6 matrix giving the displacements of the links of each leg as a function of the displacements of points H_1, H_2 and H_3. The lumped stiffness parameters has been assumed as $k_{bk} = k_{dk} = 2.625 \times 10^6$ N/m and $\mathrm{k_{Tk}} = 58.4 \times 10^3$ Nm/rad, [31].

In the numerical example, for evaluation and design purposes we have assumed $r_p = r_f$, $\mathrm{a_k} = c_k$, $b_k = d_k$. Results of the proposed design procedure as applied to the CAPAMAN architecture are reported in Figs. 9–11 and Table 3 and 4. In Table 3 the optimal values are different from the initial ones except than s_k and c_k, which have remained unchanged since their initial values are close to the optimal ones. Comparing the initial and optimal values in Table 4 one can note that the position and orientation workspaces increased and are close to the prescribed ones. The stiffness performance is still within the desired value.

Table 3
Design parameters for optimal CaPaMan design of Figs. 7 to 11

	a_k (mm)	b_k (mm)	h_k (mm)	r_p (mm)	α_k (deg)	S_k (mm)
Initial guess	200.0	140.0	130	109.5	60;130	50.0
Optimal value	200.0	161.0	109.3	98.6	35;145	50.0

Table 4
Design characteristics of optimum solution for optimal CaPaMan design of Figs. 7 to 11 and Table 3

	V_{PW} (mm^3)	V_{OW} (deg^3)	Sx (mm)	Sy (mm)	Sz (mm)	$S\phi$ (deg)	$S\psi$ (deg)	$S\theta$ (deg)
Initial guess	3.8×10^6	7.2×10^6	0.0044	0.37	0.0030	0.2×10^{-3}	0.2×10^{-3}	0.0039
Optimal value	3.9×10^6	7.9×10^6	0.0075	0.37	0.0033	0.3×10^{-3}	0.3×10^{-3}	0.0098

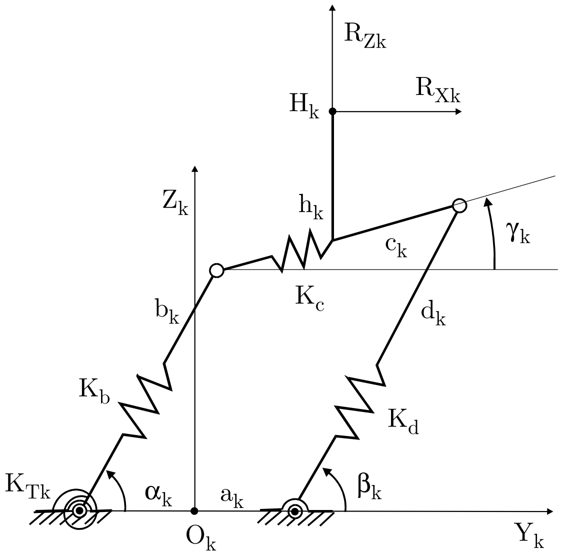


Fig. 8. A scheme for stiffness evaluation of a CaPaMan leg

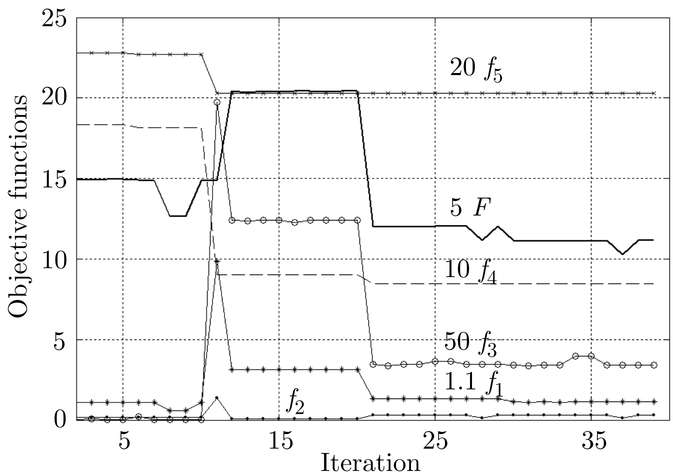


Fig. 9. Evolution of the function $\boldsymbol{F}$ and its components versus number of iterations for the example of CaPaMan optimal design

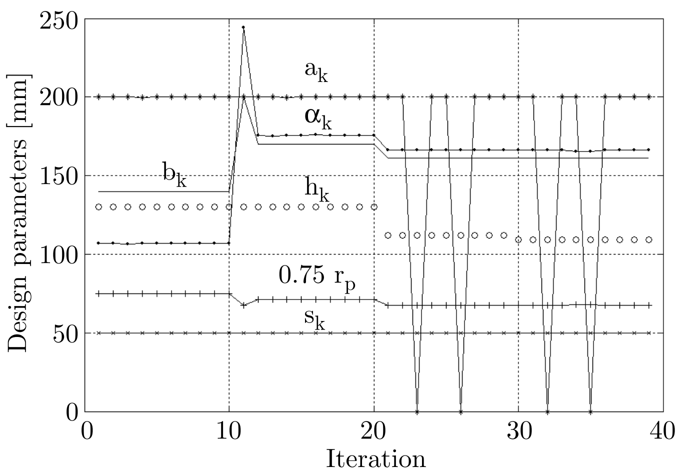


Fig. 10. Evolution of design constraints versus number of iterations for the example of CaPaMan optimal design

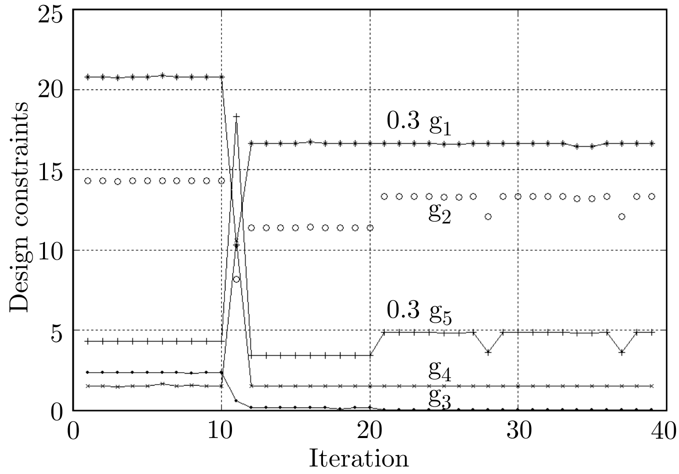


Fig. 11. Evolution of design parameters versus number of iterations for the example of CaPaMan optimal design

5. Conclusion

In this paper a multi-objective optimum design procedure for manipulators is outlined by using optimality criteria and numerical aspects. A multi-objective optimization problem is formulated by referring to basic performance of both parallel and serial manipulators. Additional objective functions can be used to extend the proposed design procedure to more general but specific design problems. The feasibility of such a complex design formulation for robotic manipulators has been illustrated by referring to experiences developed at LARM in Cassino, when optimality criteria and numerical aspects are formulated by taking into account the peculiarity and constraints of both serial and parallel manipulator architectures.

References

[1] B.E. Shimano and B. Roth, "Dimensional synthesis of manipulators", *3rd RoManSy*, pp. 166–187 (1980).

[2] B. Roth, "Analytical design of open chains", *3rd International Symposium on Robotics Research, MIT Press*, pp. 281–288 (1986).

[3] R. Vijaykumar, K.J. Waldron and M.J. Tsai, "Geometric optimization of serial chain manipulator structures for working volume and dexterity", *The International Journal of Robotics Research* 5 (2), 91–103 (1986).

[4] C. Gosselin and J. Angeles, "The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator", *ASME Journal of Mechanisms, Transmissions, and Automation in Design* 110, 35–41 (1988).

[5] B. Paden and S. Sastry, "Optimal kinematic design of 6R manipulators", *The International Journal of Robotics Research* 7 (2), 43–61 (1988).

[6] S. Manoochehri and A.A. Seireg, "A computer-based methodology for the form synthesis and optimal design of robot manipulators", *Journal of Mechanical Design* 112, 501–508 (1990).

[7] C. M. Gosselin, "The optimum design of robotic manipulators using dexterity indices", *Robotics and Autonomous Systems* 9, 213–226 (1992).

[8] F.C. Park, "Optimal robot design and differential geometry", *Transaction ASME, Special 50th Anniversary Design Issue* 117, 87–92 (1995).

[9] J.A. Carretero, R.P. Podhorodeski, M.A. Nahon and C.M. Gosselin, "Kinematic analysis and optimization of a new three degree-of freedom spatial parallel manipulator", *ASME Journal of Mechanical Design* 122 (1), 17–24 (2000).

[10] 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* 35 (9), 1257–1267 (2000).

[11] J. Angeles, "The robust design of parallel manipulators", *1st International Colloquium Collaborative Research Center* 562, Braunschweig, pp. 9–30 (2002).

[12] M. Ceccarelli, "Optimal design and location of manipulators", in: *Computational Dynamics in Multibody Systems*, M.F.O.S. Pereira and J.A.C. Ambrosio (eds), Kluwer, Dordrecht, pp. 131–146 (1995).

[13] M. Ceccarelli, "A formulation for the workspace boundary of general n-revolute manipulators", *IFToMM Journal Mechanism and Machine Theory* 31, 637–646 (1996).

[14] M. Ceccarelli, "An analytical design of telescopic manipulator arms for prescribed workspace", *Advances in Robot Kinematics: Analysis and Control*, Kluwer, Dordrecht, pp. 247–254 (1998).

[15] M. Ceccarelli, "Designing two-revolute manipulators for prescribed feasible workspace regions", *ASME Journal of Mechanical Design* 124, 427–434 (2002).

[16] M. Ceccarelli and G. Scaramuzza, "Analytical constraints for a workspace design of 2R manipulators", *Computational Kinematics '95*, J.P. Merlet and B. Ravani (eds), Kluwer, Dordrecht, pp. 173–182 (1995).

[17] M. Ceccarelli and G. Carbone, "A stiffness analysis for CaPaMan (Cassino Parallel Manipulator)", *Mechanism and Machine Theory* 37 (5), 427-439 (2002).

[18] A.V. Mata and M. Ceccarelli, "Funciones objectivo para la optimizaciòn de la cadena cinemàtica de robots", *1° Congreso Iberoamericano de Engeneria Mecanica,* Madrid 3, 47–54 (1993), (in Spanish).

[19] M. Ceccarelli, "A synthesis algorithm for three-revolute manipulators by using an algebraic formulation of workspace boundary", *ASME Journal of Mechanical Design* 117, 298–302 (1995).

[20] M. Ceccarelli, "Workspace analysis and design of open chain manipulators", *Computing Anticipatory Systems*, Dubois D.M. (ed.), American Institute of Phyiscs, New York, pp. 338–355 (1998).

[21] E. Ottaviano and M. Ceccarelli, "Optimal design of CaPaMan (Cassino Parallel Manipulator) with prescribed position and orientation workspace", *9th International Conference on Control and Automation*, Dubrovnik, Paper No. Med01-009 (2001).

[22] E. Ottaviano and M. Ceccarelli, "Optimum design of parallel manipulators for workspace and singularity performances", *Proceedings of the Workshop on Fundamental Issues and Future Research Directions for Parallel Mechanisms and Manipulators*, Quebec City, pp. 98–105. (2002)

[23] C. Lanni, S.F.P. Saramago and M. Ceccarelli, "Optimal design of 3R manipulators by using classical techniques and simulated annealing", *Journal of the Brazilian Society for Mechanical Sciences* XXIV, pp. 294–302 (2002).

[24] M. Ceccarelli and C. Lanni, "A multi-objective optimum design of general 3R manipulators for prescribed workspace limits", *Mechanism and Machine Theory* 39 (2), 119–132 (2004).

[25] E. Ottaviano and G. Carbone "A procedure for the multiobjective design of parallel manipulators", *12th International Workshop on Robotics in Alpe-Adria-Danube Region RAAD 2003*, Cassino, CD Proceedings, Paper 061RAAD03 (2003).

[26] M. Ceccarelli, *Fundamentals of Mechanics of Robotic Manipulation*, Kluwer, Dordrecht 2004.

[27] LARM webpage, http://webuser.unicas.it/weblarm/larmindex.htm, (2004).

[28] G. Vanderplaats, *Numerical Optimization Techniques for Engineers Design*, McGraw-Hill, New York, 1984.

[29] A. Grace, *Optimization Toolbox User's Guide*, The Matlab Works Inc. 1992

[30] E. Ottaviano and M. Ceccarelli, "Optimal design of CaPaMan (Cassino Parallel Manipulator) with prescribed workspace", *IFToMM Electronic Journal Computational Kinematics EJCK* 1 (1), Comp. Kin. CK2001, Seoul, 35–44 (2002)

[31] M. Ceccarelli "A stiffness analysis for CaPaMan (Cassino Parallel Manipulator)", *Proc. of Conference on New Machine Concepts for Handling and Manufacturing Devices on the Basis of Parallel Structures*, VDI 1427, Braunschweig, pp. 67–80 (1998).