Robotica (2008) volume 26, pp. 229–239. © 2007 Cambridge University Press
doi:10.1017/S0263574707003839 Printed in the United Kingdom

# An optimum path planning for Cassino Parallel Manipulator by using inverse dynamics

G. Carbone†, M. Ceccarelli†, P. J. Oliveira‡, S. F. P. Saramago‡ and J. C. M. Carvalho‡

*†Laboratory of Robotics and Mechatronics, DiMSAT, University of Cassino, Via Di Biasio 43 - 03043 Cassino, (Fr), Italy.*
*E-mails: carbone@unicas.it, ceccarelli@unicas.it*
*‡Federal University of Uberllândia Av. Joao Naves de Avila, 2160-38408100 Uberlandia (MG), Brazil.*
*E-mails: saramago@ufu.br, jcmendes@mecanica.ufu.br*

(Received in Final Form: February 14, 2007. First published online: September 10, 2007)

**SUMMARY**

In this paper, a novel algorithm is formulated and implemented for optimum path planning of parallel manipulators. A multi-objective optimisation problem has been formulated for an efficient numerical solution procedure through kinematic and dynamic features of manipulator operation. Computational economy has been obtained by properly using a genetic algorithm to search an optimal solution for path spline-functions. Numerical characteristics of the numerical solving procedure have been outlined through a numerical example applied to Cassino Parallel Manipulator (CaPaMan) both for path planning and design purposes.

KEYWORDS: Robotics; Path Planning; Parallel Manipulators; Simulation.

## 1. Introduction

In the last two decades, parallel architectures have been extensively studied because they show advantages such as higher stiffness and accuracy positioning with respect to serial architectures, and they can operate at high velocities and accelerations as mentioned for example in [1]. These characteristics suggest their use in many applications, for example in assembly and disassembly processes, packing, tool and object handling, milling machines and motion simulation. Thus, several new parallel mechanisms have been conceived, designed and built together with a development of theoretical and practical investigations like those that are presented in refs. [1–13].

Cassino Parallel Manipulator (CaPaMan) is a parallel manipulator, having three degrees of freedom (dof), which has been conceived at the Laboratory of Robotics and Mechatronics (LARM) at Cassino, Italy.[13] Performance and suitable formulation for kinematics, statics and dynamics have been investigated and results are reported in refs. [14–22]. A prototype has been built, and a successful application of CaPaMan as an earthquake simulator, which can reproduce a really happened earthquake is reported in refs. [23–25].

In many industrial applications, the movements of robot manipulators are planned manually in an *ad hoc* manner so that they do not usually perform tasks with optimal paths. Generally, this method for path planning cannot give the best paths or/and the maximum permissible speeds at all points along programmed paths. Nevertheless, when repetitive processes are prescribed, it is possible to develop a methodology to move a robot manipulator along a specified optimum path. This can be achieved through a formulation of a suitable optimisation problem. Considerable research activity has been carried out in order to obtain optimal paths with serial robots, and the corresponding literature is very rich. For example, in ref. [26], Lin *et al.* have proposed a procedure to determine a cubic polynomial joint trajectory through an algorithm for minimizing the traveling time subject to physical constraints on joint velocities, accelerations and jerks. In ref. [27], Shin and Mckayhave presented a solution to the problem of minimizing the power consumption of moving a serial robotic manipulator along a specified end-effector path subject to input torque/force constraints, by taking into account the dynamics of the manipulator. Other optimum path planning methods can be found in refs. [28–36].

A clear tendency exists in looking at models in nature to represent processes that can be called "intelligent." There is strong evidence that natural processes related to human beings are very well developed, and they can be adapted to the engineering world by bringing in surprising results in a lot of applications. Among the computational paradigms that have been derived by looking at models in nature very interesting are simulated annealing, artificial neural networks and evolutionary computation. In the family of the so-called evolutionary computation methods, that include a growing number of paradigms and methods, genetic algorithms are considered to be the most promising. Genetic algorithms (G.A.) are computational search methods that are based on the mechanisms of natural evolution and genetics. In G.A., a population of possible solutions for a problem evolves in agreement with probabilistic operators that can be formulated as biological metaphors, so that, in general, individuals represent better solutions as the evolutionary process continues.[37]

In this paper,[1] a general formulation has been proposed for optimum path planning for parallel manipulators by

[1] A preliminary version of this paper has been presented as paper no. 28 at MUSME 2005, IFToMM-FeIbIM International Symposium on Multibody Systems and Mechatronics held on 6–9 March 2005 in Uberlandia, Brazil.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

taking into account mechanical energy of the actuators and total travelling time for a formulation of a multi-objective function. Feasible trajectories have been defined by using spline functions that can be obtained through off-line computation. The nature of the proposed optimisation problem is such that in general, several local minima exist and can be observed throughout a numerical procedure. A G.A. is a probabilistic search algorithm capable of finding the global minimum amongst many local minima, particularly in the case where traditional techniques fail. For this reason, a G.A. has been used for solving the numerical optimisation problem for the optimal path of parallel manipulators containing contradictory optimality criteria. The proposed procedure has been applied to CaPaMan as a practical example, also in order to further improve its dynamic performance by reducing its power consumption through a minimisation of the needed actuator energy and total travelling time in trajectory motion. A numerical example of the optimum path planning for CaPaMan is reported in order to show the numerical efficiency and feasibility of the proposed procedure.

## 2. CaPaMan Prototype

CaPaMan architecture has been conceived at LARM Laboratory of Robotics and Mechatronics in Cassino, where a prototype has been built for experimental activity since 1996.[13]

A schematic representation of the CaPaMan manipulator is shown in Fig. 1, where the fixed platform is FP and the moving platform is MP. MP is connected to FP through three identical leg mechanisms and is driven by the corresponding articulation points $H_1$, $H_2$ and $H_3$. An articulated parallelogram AP, a prismatic joint SJ and a connecting bar CB compose each leg mechanism. AP's coupler carries the SJ, and CB transmits the motion from AP to MP through SJ; CB is connected to the MP by a spherical joint BJ, which is installed on MP. CB may translate along the prismatic guide of SJ keeping its vertical posture, and BJ allows MP to rotate in the space. Each plane, which contains AP, is rotated of $\pi/3$ with respect to the neighbor one. Figure 2 shows a built prototype whose main design parameters are listed in Table I.

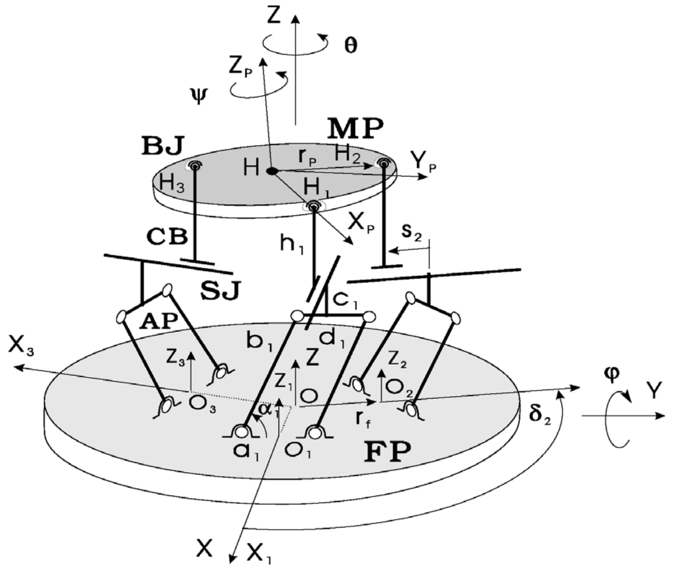


Fig. 1. Kinematic chain and design parameters of CaPaMan.

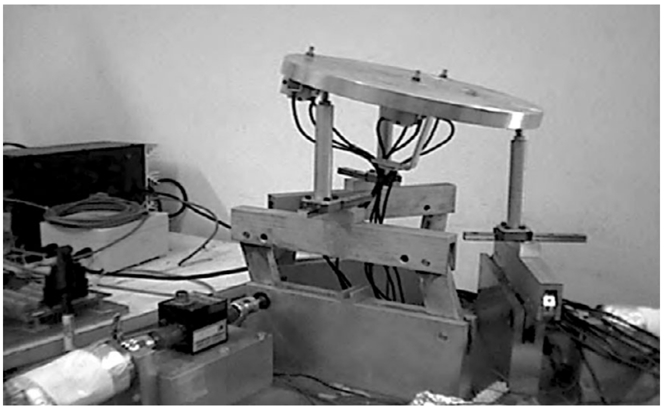


Fig. 2. Prototype of CaPaMan with accelerometers and a dynamic torsionmeter at LARM in Cassino.

Table I. Sizes and motion parameters of the built prototype of CaPaMan, Figs.1 and 2.

| $a_k = c_k$ | $b_k = d_k$ | $H_k$ | $r_P = r_f$ | $\alpha_k$ | $s_k$ |
|---|---|---|---|---|---|
| [mm] | [mm] | [mm] | [mm] | [deg] | [mm] |
| 200 | 80 | 100 | 109.5 | 45; 135 | −50; 50 |

Particularly, design parameters of the $k$th leg are identified through: $a_k$, which is the length of the frame link; $b_k$, which is the length of the input crank; $c_k$, which is the length of the coupler link; $d_k$, which is the length of the follower crank; and $h_k$, which is the length of the connecting bar. The kinematic variables are: $\alpha_k$, which is the input crank angle; and $s_k$, which is the stroke of the prismatic joint. Finally, the size of MP and FP are given by $r_p$ and $r_f$, respectively, $H$ is the center point of MP, $O$ is the center point of FP, $H_k$ is the center point of the $k$th BJ, and $O_k$ is the middle point of the frame link $a_k$ (Fig. 1). The motion of MP with respect to FP can be described by considering a world frame O-XYZ, which is fixed to FP, and a moving frame H-$X_pY_pZ_p$, which is fixed to MP.

The symmetry characteristics of CaPaMan architecture have been useful to formulate analytical dynamic equations to compute the input torques, which are necessary for a given motion trajectory of the movable platform as reported in refs. [16–18]. Assumptions have been made in order to simplify the equations such as the effects of link elasticity, and viscous damping of the joints have been neglected; links are assumed to be rigid bodies, and the joints are frictionless and have no clearance. In addition, only the inertial effects of the movable platform has been considered since the legs of parallel architectures are lighter than the movable plate. Successively, the inertial effects of the legs and prismatic joint have been superposed.

By neglecting the friction on prismatic and spherical joints, the only forces that are applied to the rods CB by the mobile platform are those, which are contained in the plane of the articulated parallelogram, i.e. $F_{ky}$ and $F_{kz}$ as shown in

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

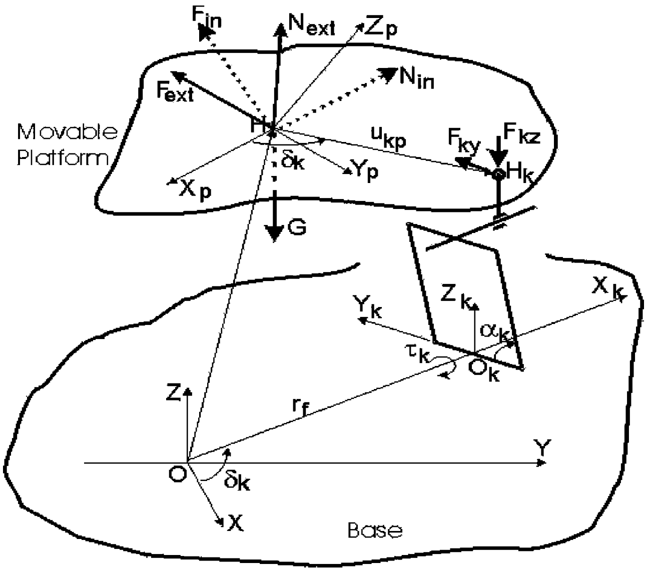


Fig. 3. Forces acting on the mobile platform of CaPaMan.

Fig. 3. The $F_{kx}$ component determines the sliding of the prismatic joint along $s_k$ coordinate. Thus, the components of the resultant force **F** and torque **N** can be computed as

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \\ N_x \\ N_y \\ N_z \end{Bmatrix} = \begin{Bmatrix} -\frac{\sqrt{3}}{2}F_{2y} + \frac{\sqrt{3}}{2}F_{3y} \\ F_{1y} - \frac{1}{2}F_{2y} - \frac{1}{2}F_{3y} \\ F_{1z} + F_{2z} + F_{3z} \\ -u_{1z}F_{1y} + \frac{1}{2}u_{2z}F_{2y} + \frac{1}{2}u_{3z}F_{3y} + u_{1y}F_{1z} \\ \quad + u_{2y}F_{2z} + u_{3y}F_{3z} \\ -\frac{\sqrt{3}}{2}u_{2z}F_{2y} + \frac{\sqrt{3}}{2}u_{3z}F_{3y} - u_{1x}F_{1z} \\ \quad - u_{2x}F_{2z} - u_{3x}F_{3z} \\ \frac{1}{2}(\sqrt{3}u_{2y} - u_{2x})F_{2y} - \frac{1}{2}(\sqrt{3}u_{3y} + u_{3x})F_{3y} \\ \quad + u_{1x}F_{1y} \end{Bmatrix} \tag{1}$$

with

$$\begin{Bmatrix} u_{kx} \\ u_{ky} \\ u_{kz} \end{Bmatrix} = r_p R \begin{Bmatrix} \cos\delta_k \\ \sin\delta_k \\ 0 \end{Bmatrix}, \quad (k = 1, 2, 3). \tag{2}$$

when the values $\delta_1 = 0$; $\delta_2 = 2\pi/3$; $\delta_3 = 4\pi/3$ are considered.

Equations (1) and (2) can be solved in a closed form to obtain an explicit expression for forces $F_{ky}$ and $F_{kz}$. Referring to Fig. 2 and once the reaction forces in the spherical joints $H_k$ are computed, the torque $\boldsymbol{\tau}_{Pk}(k = 1, 2, 3)$ on the input crank shaft of each articulated parallelogram can be obtained by considering only the inertial effects of the movable platform in the form

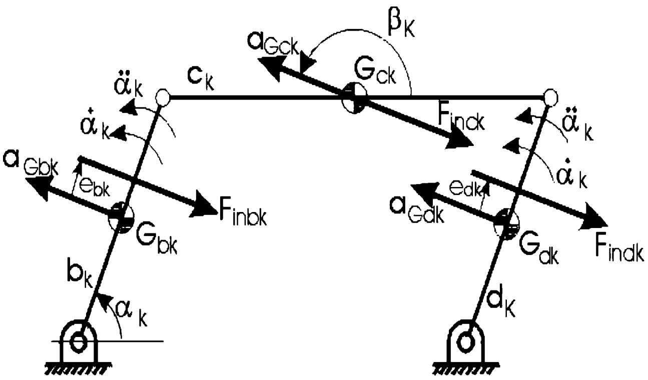


Fig. 4. Inertia forces arising on each articulated parallelogram of CaPaMan legs.

$$\tau_{Pk} = \frac{F_{kz}\,b_k}{2}\frac{\sin 2\alpha_k}{\sin\alpha_k} - F_{ky}b_k\left(\frac{h_k}{c_k\tan\alpha_k} + 1\right) \times \left(1 - \frac{h_k}{h_k\cos\,\alpha_k + c_k\sin\,\alpha_k}\right)\sin\alpha_k \tag{3}$$

where $b_k$, $c_k$ and $h_k$ are the geometrical dimensions shown in Fig. 1; $F_{ky}$ and $F_{kz}$ are the reaction forces in the spherical joints $H_k$.

The contribution of the legs to the inverse dynamics of the CaPaMan can be determined by a kinematic analysis of the articulated parallelograms. The centres of mass of the links can be identified as shown in Fig. 4.

A kinetostatic analysis and superposition principle can be used for computing separately the inertial and gravitational effects of the links $b_k$, $c_k$ and $d_k$, and then they can be combined to give the input torque $\boldsymbol{\tau}_{\boldsymbol{Mk}}$ as

$$\boldsymbol{\tau}_{\boldsymbol{Mk}} = 2l_{bk}F_{inbk}\sin(\alpha_k - \beta_k + \pi) + F_{23k}b\,\sin(\alpha_k + \pi - \gamma_k) + b\left[m_{bk}\cos\alpha_k + \frac{m_{ck}}{2}\frac{\sin 2\alpha_k}{\sin\alpha_k}\right]g \tag{4}$$

The angle $\beta_k$ in Eq. (4) defines the direction of the acceleration of the mass centre of the $k$th link with respect to the horizontal axis as shown in Fig. 4. It is assumed to be positive counterclockwise. Similarly, the angle $\gamma_k$ in Eq. (4) defines the direction of the reaction force vector acting on the ground pivot of link $d_k$, and it is also assumed to be positive counterclockwise. The terms $l_{bk}$, $F_{23k}$ and $\gamma_k$ in Eq. (4) can be written as

$$l_{bk} = \frac{\mathrm{b}}{2} + \frac{I_{Gbk}\ddot{\alpha}_k}{F_{inbk}}\frac{1}{\sin(\alpha_k - \beta_k + \pi)}$$

$$F_{23k} = \sqrt{\left\{F_{inck}\left[\cos(\beta_k + \pi) + \frac{\sin(\pi - \beta_k)}{2\tan\alpha_k}\right]\right\}^2 + \left[\frac{F_{inck}\sin(\beta_k + \pi)}{2}\right]^2} \tag{5}$$

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

$$\gamma_k = \text{tg}^{-1}\left\{\left[F_{inck}\left(\cos(\beta_k+\pi)+\frac{\sin(\pi-\beta_k)}{2\tan\alpha_k}\right)\right]\Big/\left[\frac{F_{inck}\sin(\beta_k+\pi)}{2}\right]\right\}$$

where $I_{Gbk}$ is the inertia matrix with respect to link center of mass $G_{bk}$; $F_{inbk}$, $F_{inck}$ and $F_{indk}$ are the inertia forces of each link of the articulated parallelograms. Details of derivations of the terms in Eq. (5) are reported in refs. [16, 17].

The superposition principle can be used again in order to obtain the total torque $\tau_k$ on the input shaft of the articulated parallelogram as the sum of the torques computed in Eqs. (3) and (4) in the form

$$\tau_k = \tau_{Pk} + \tau_{Mk} \tag{6}$$

## 3. Formulation for Optimal Path Planning

The path planning task for a manipulator with $n$ dof can be described using $m$ knots in the trajectory of each $k$th joint of a manipulator. The prescribed task can be given by the initial and final points $P_0$ and $P_m$ of the trajectory. The movement of the manipulator can be obtained by the simultaneous motion of the $n$ joints in order to perform the prescribed task.

Among the many available criteria, one can assume the energy aspect as one of the most significant performance in order to optimize the manipulator operation, since the energy formulation can consider simultaneously dynamic and kinematic characteristics of the performing motion. It should also be considered that a maximisation of the operation speed of a manipulator corresponds to a minimisation of the total travelling time. Thus, minimum mechanical energy of actuators and minimum total travelling time can be considered together in a multi-objective function.

In a multi-criteria optimisation, one deals with a design variable vector $\mathbf{x}$, which satisfies all the constraints and makes the scalar performance index as small as possible. This index is calculated by taking into account each component of an objective function vector $\mathbf{f}(\mathbf{x})$. An important feature of such multi-criteria optimisation problem is that the optimizer has to deal with conflicting objectives. A possible approach to this problem is the so-called compromise programming. This approach does not provide unique solution to the problem but a set of solutions named as Pareto-optima set.[38] Weighting objectives is one of the most usual and simple alternative approaches for multi-objective optimisation problems. In this case, the objective function $f$ can be determined by the linear combination of the $r$ criteria $f_1, \ldots, f_r$, together with the corresponding weighting factors $w_1, \ldots, w_r$ in the form

$$f(\mathbf{x}) = \sum_{i=1}^{r} w_i f_i(\mathbf{x}) \tag{7}$$

Usually, weighting factors are assumed with the conditions $0 \leq w_i \leq 1$ and $\sum_{i=1}^{r} w_i = 1$. Then, it is possible to generate the Pareto-optima set for the original problem by varying the weights $w_i$ in the objective function.

An optimality criterion concerning the energy aspects of the path motion can be conveniently expressed in terms of the work that is needed by the actuators. In particular, the work by the actuators is needed for increasing the kinetic energy of the system in a first phase from a rest condition to actuators states at which each actuator is running at maximum velocity. In a second phase bringing the system back to a rest condition, the kinetic energy will be decreased to zero through the actions of actuators and brakes. The potential energy of the system will contribute to size the necessary work by the actuators, and friction effects in the joints can be assumed as negligible as compared to the actions of actuators and brakes. In addition, it is to note that because of motion capability of CaPaMan, its centre of mass will move with a short path with less then 30 mm in vertical direction. Therefore, the potential energy change can be considered negligible if it is compared with the kinetic energy that is due to a desired fast CaPaMan motion.

Thus, we have considered convenient to use the work $W_{\text{act}}$ done by the actuators in the first phase of the path motion as an optimality criterion for optimal path generation as given by the expression

$$W_{\text{act}} = \sum_{k=1}^{3}\left[\int_0^{t_k} \tau_k \dot{\alpha}_k \, dt\right] \tag{8}$$

in which $\tau_k$ is the $k$th actuator torque; $\alpha_k$ dot is the $k$th shaft angular velocity of the actuator; and $t_k$ is the time coordinate value delimiting the first phase of path motion with increasing speed of the $k$th actuator.

Therefore, trying to minimize the ratio $W_{\text{act}}/W_{\text{act0}}$ with $W_{\text{act0}}$ as a prescribed value, has the aim to size at the minimum level the design dimensions and operation actions of the actuators in generating a path between two given extreme positions. The prescribed value $W_{\text{act0}}$ has been chosen as referring to the power of a commercial actuator that has been considered suitable for CaPaMan operation.

Indeed, in general, once the actuator work is minimized, the energy aspects of the system operation will be optimized consequently.

A scalar objective function can be proposed in order to consider minimum actuator work $W_{\text{act}}$ and minimum travelling time $Tt$ simultaneously in the form

$$\min f = w_1 \frac{W_{\text{act}}}{W_{\text{act0}}} + w_2 \frac{Tt}{Tt_0} \tag{9}$$

subject to

$$\alpha_k^b \leq \alpha_k(t) \leq \alpha_k^u, \quad (k = 1, 2, 3) \tag{10}$$

$$Tt^b \leq Tt \leq Tt^u \tag{11}$$

where $b$ and $u$ stand for lower and upper values of the variables within their feasible ranges, respectively; $\alpha_k$ is the $k$th joint variable; $t$ is the time variable in the interval $[0, Tt]$ for the path between $P_o$ and $P_m$. $Tt$ is the total travelling time at the end point $P_m$ when $t = 0$ is assumed at the initial point $P_0$; $Tt_0$ is total travelling time for the initial guess solution.

It is worth noting that the proposed multi-objective function in Eq. (9) is composed of two competitive terms.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

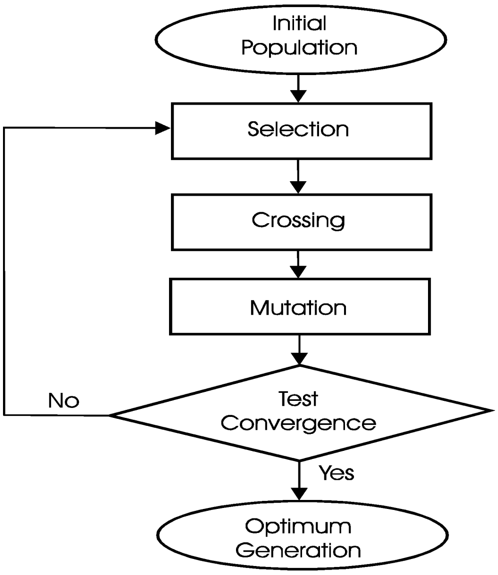


Fig. 5. Flowchart for procedures in genetic algorithms.

In general, the first term $w_1(W_{\text{act}}/W_{\text{act0}})$ decreases when the total travelling time $Tt$ increases while the second term $w_2(Tt/Tt_0)$ does the opposite. Therefore, the optimisation process does not converge to the minimum permitted motion time $Tt^b$. The proposed formulation in Eqs. (8–11) requires the computation and consideration of the manipulator kinematics and dynamics. Torque computation has been formulated in Section II, and motion kinematics will be addressed in Section IV by using suitable algorithms for path planning.

Genetic algorithms (G.A.) were introduced by Holland in ref. [39], and they can be understood as being the processes for directed random search according to the abovementioned definition. The main characteristics of the G.A. technique are presented in the flowchart of Fig. 5. G.A. operate on a population of points not concentrating all the search effort on only one point as conventional methods do. Moreover, G.A. operate in a space of coded solutions, and not in the search space directly. They do not require derivation or any other function knowledge in order to be operated. They need the value of the objective function for each individual of the population only. They use probabilistic transitions and not deterministic ones. Thus, an optimisation procedure improves solutions until an optimal one is found similarly to the evolution process in a search along successive generations.

## 4. Formulation for Trajectory Modelling

In order to determine joint trajectories, one can use the given initial and final points $P_0$ and $P_m$ in the Cartesian coordinates. These given points can be transformed into the joint coordinates by solving the inverse kinematics. Then, cubic polynomial can be chosen to describe the joint trajectories since Proportional Derivative control law is generally used for manipulator actuators in industrial robots. B-splines are often used as interpolating functions to represent a trajectory of mechanical systems. An important characteristic is that they allow to control the degree of continuity between two adjacent segments. This fact is important because smooth transition is required for path planning in many applications such as in robotics. Another important characteristic of the cubic B-splines is that they satisfy the convex hull property, which allows the refinement of a trajectory. Thus, referring to the three legs of CaPaMan, each trajectory function $\alpha_k(t)$ can be modelled by a uniform cubic B-Spline in the form

$$\alpha_k(t) = \sum_{i=0}^{m} p_i^k B_{i,d}^k(t), \qquad (m \geq 3, k = 1, 2, 3) \tag{12}$$

where $p_i^k$ (with $i = 0, \ldots, m$) are the $m+1$ control parameters of the knot points corresponding to the trajectory function $\alpha_k(t)$; $B_{i,d}$ are the functions that can be defined by using the Cox de Boor recurrence formulas,[40] with $d = 3$ for cubic spline, in the form

$$\begin{aligned} B_{i,1}(t) &= \begin{cases} 1, & \text{if } t_i \leq t \leq t_{i+1} \\ 0, & \text{if } t_i > t \text{ or } t > t_{i+1} \end{cases} \\ B_{i,d}(t) &= \frac{t - t_i}{t_{i+d-1} - t_i} B_{i,d-1}(t) \\ &\quad + \frac{t_{i+d} - t}{t_{i+d} - t_{i+1}} B_{i+1,d-1}(t), \qquad \forall t. \end{aligned} \tag{13}$$

Since $\alpha_k(t)$ is a cubic, its $j$th derivatives with respect to $t$ can be straightforward computed as

$$\frac{d^j \alpha_k(t)}{dt^j} = \sum_{i=0}^{m} p_i^k \frac{d^j B_{i,d}^k(t)}{dt^j}. \tag{14}$$

Thus, the optimisation design variables for path planning are the control parameters $p_k^i$ of each joint trajectory together with the total time $Tt$ of path travelling. The initial cubic B-spline control points can be obtained by a guess fitting a cubic polynomial between the initial and final points of each trajectory. Thus, for the CaPaMan case with $n = 3$ input actuators, the total number of design variables for path planning is $(n \times m) + 1$.

## 5. Numerical Example of Optimal Path Planning

The Matlab code Genetic Algorithms Optimisation Toolbox (GAOT)[41] has been used to find an optimum path through G.A. A general analysis code has been developed by involving a dynamical model and a trajectory planning of a robot. Then, it has been implemented into the proposed solving procedure according to the flowchart in the Fig. 6.

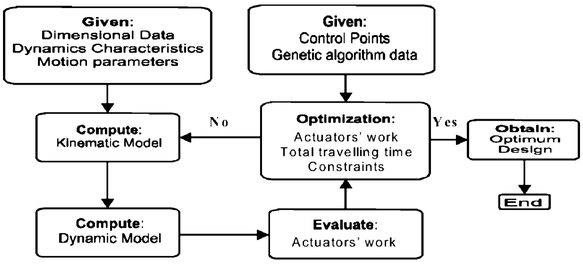

Given: Dimensional Data Dynamics Characteristics Motion parameters
Given: Control Points Genetic algorithm data
Compute: Kinematic Model
No
Optimization: Actuators' work Total travelling time Constraints
Yes
Obtain: Optimum Design
End
Compute: Dynamic Model
Evaluate: Actuators' work

Fig. 6. Flowchart of the proposed numerical procedure for optimum path planning of CaPaMan parallel manipulator.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

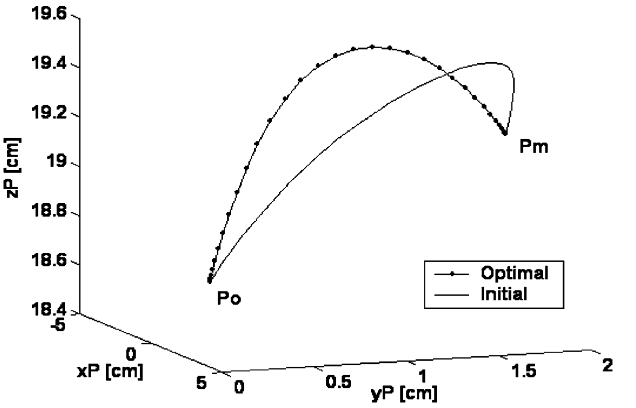
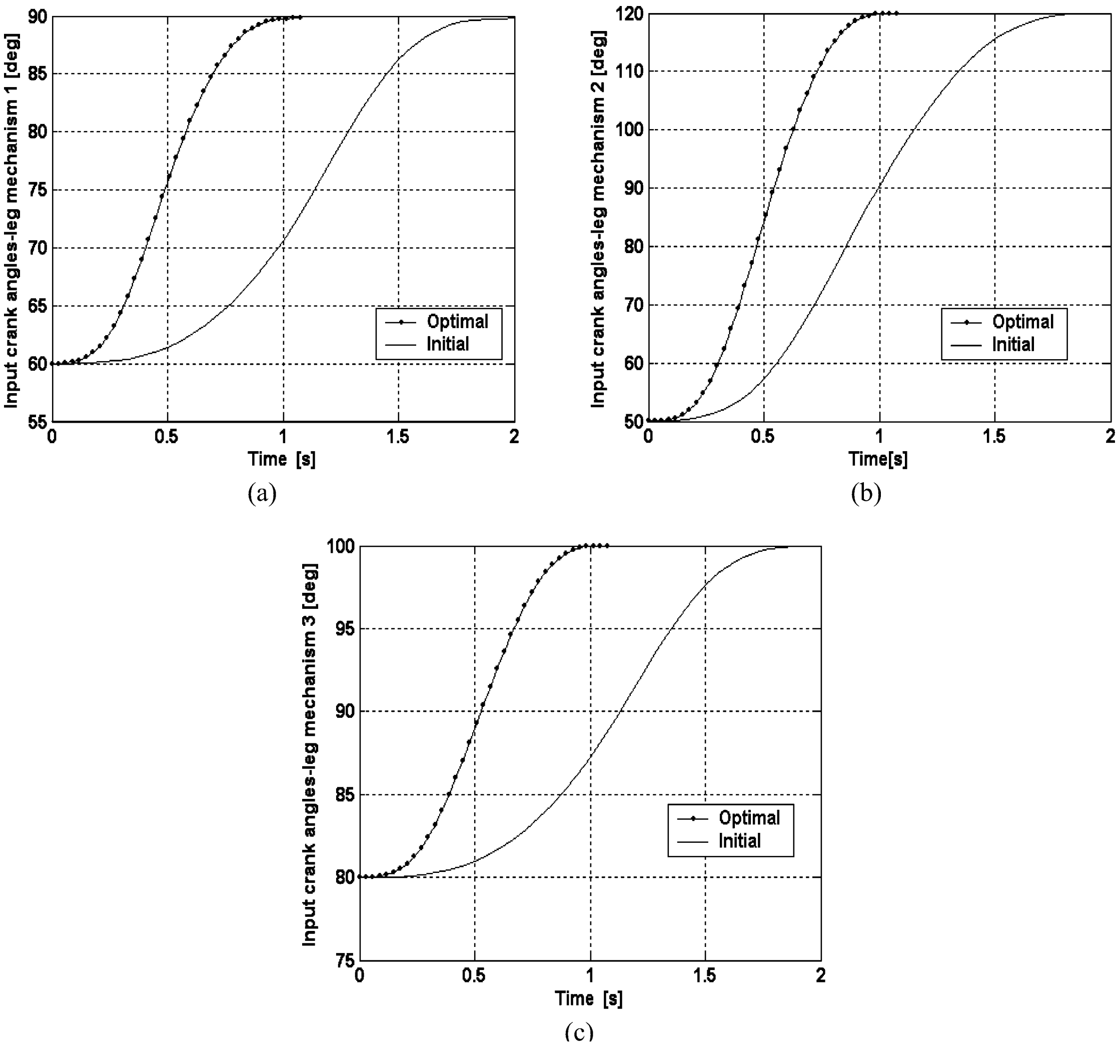


Fig. 7. Numerical results for the case of Table II for $w_1 = 0.2$ and $w_2 = 0.8$ in the form of a 3D plot of the position of the centre of the movable plate of CaPaMan as function of time.

In particular, a dimensional data, dynamic characteristics and motion parameters have been used for computing the kinematics and dynamics of the system. The dynamic model has been used for computing the mechanical energy that is needed for the motion. The proposed optimisation procedure requires inputs in the form of control points, genetic algorithm data and computed mechanical energy. If the total energy and the travelling time are not optimal under given constraints, the procedure searches for a different path. If the total energy and travelling time are optimal, then the procedure stops and gives as output the identified optimal path.

A numerical example referring to CaPaMan is reported by assuming a starting position given by $\alpha_1 = 60°$, $\alpha_2 = 50°$, $\alpha_3 = 80°$ and a final position given by $\alpha_1 = 90°$, $\alpha_2 = 120°$, $\alpha_3 = 100°$. Dimensional data of CaPaMan are reported in Table I. The robot is considered as starting at rest and coming to a full stop at the end of the trajectory. Thus, $\dot{\alpha}_k(0) = \dot{\alpha}_k(Tt) = 0$ for all the joint trajectories. Singularities of CaPaMan have been investigated in ref. [19]. In particular, CaPaMan has only one singular configuration within its working range, and it is given by the three input shafts at 90° position (vertical configuration). This configuration has been avoided in the path planning by adding a proper inequality constraint. Moreover, additional inequality constraints have been implemented for limiting the motion of each input crank of CaPaMan to its feasible working range that is from 45° to 135° (Table II).



(a)

(b)

(c)

Fig. 8. Time evolution of the input crank angles for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 7. (a) Leg mechanism 1. (b) Leg mechanism 2. (c) Leg mechanism 3.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

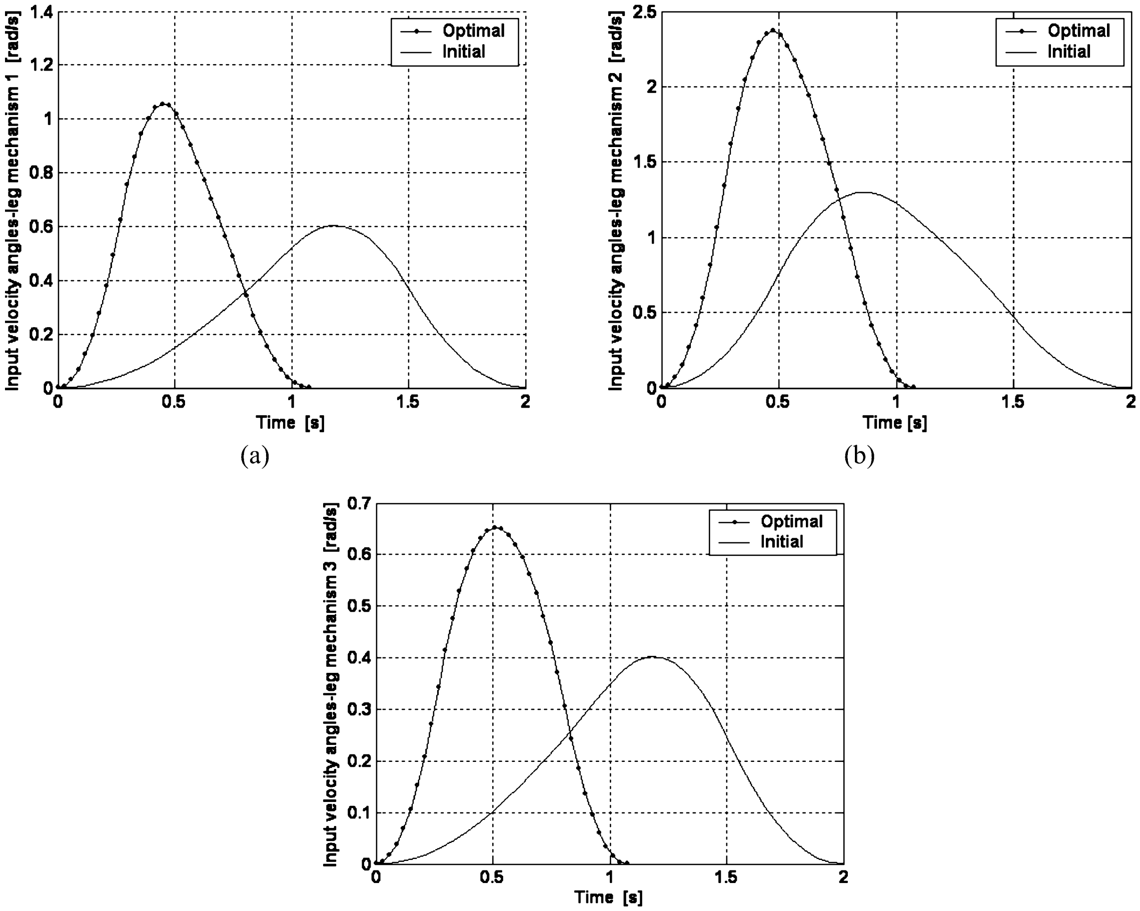
Table II. Optimum values as obtained from the optimisation process with $w_1 = 0.2$ and $w_2 = 0.8$.

| | Objective function $f$ | Work by actuators $W_{\text{act}}$ [Joule] | Travelling time $Tt$ [second] |
|---|---|---|---|
| Initial guess value | 1.00 | 4.11 | 2.0 |
| Optimum value | 0.63 | 3.90 | 1.1 |
| Performance improvement | 37% | 5.1% | 45% |

A suitable number of control points is needed to obtain a proper B-spline. The higher is the number of control points, the more complex is the profile that can be obtained. But, any increase in the number of control points produces also an increase in the computational costs. The number of control points has been chosen as equal to 8 for the reported case of study while the time interval has been divided into $N = 32$ steps. It is worth noting that the number of control points and steps within the time interval have been chosen after simulation trials as a good compromise between accuracy and computational efficiency. Thus, the number of the design variable is $(n \times m) + 1 = 25$ where $n = 3$ is the number of actuators for CaPaMan and $m = 8$ is the chosen number of control points. In addition, the following parameters have been used for the G.A. as suggested by authors' experience: 200 individuals in the population; 300 generations; binary strings for representation of individuals; roulette wheel selection with elitism. Crossover and mutation probabilities have been chosen as equal to 0.60 and 0.02, respectively. These values have been chosen in order to achieve a good compromise between accuracy of results and computational costs.

A numerical case of study is reported as referred to CaPaMan, Fig. 1 and Table I, with the afore mentioned path data. The initial guess and optimal values are compared in Table II that shows a significant improvement of the performance index for the optimal trajectory in Fig. 7. This result is obtained using GAOT code in a Pentium 4 Computer with a computational time of less than 10 min. Moreover, for this application, the weighting coefficients have been assumed as $w_1 = 0.2$ and $w_2 = 0.8$ in order to give more emphasis to the minimisation of the total travelling time as required in several practical applications.

Figure 7 shows a 3D plot of the initial guess and optimal path of the centre of the movable plate of CaPaMan. In particular, the guess path between the given points $P_o$ and $P_m$ has been chosen as a cubic spline with three intermediate knots that are arranged in such a way so as to set the initial and final accelerations to zero. It can be observed that the computed optimal trajectory is smooth and fulfills the given constraints for the initial and final points of the path.



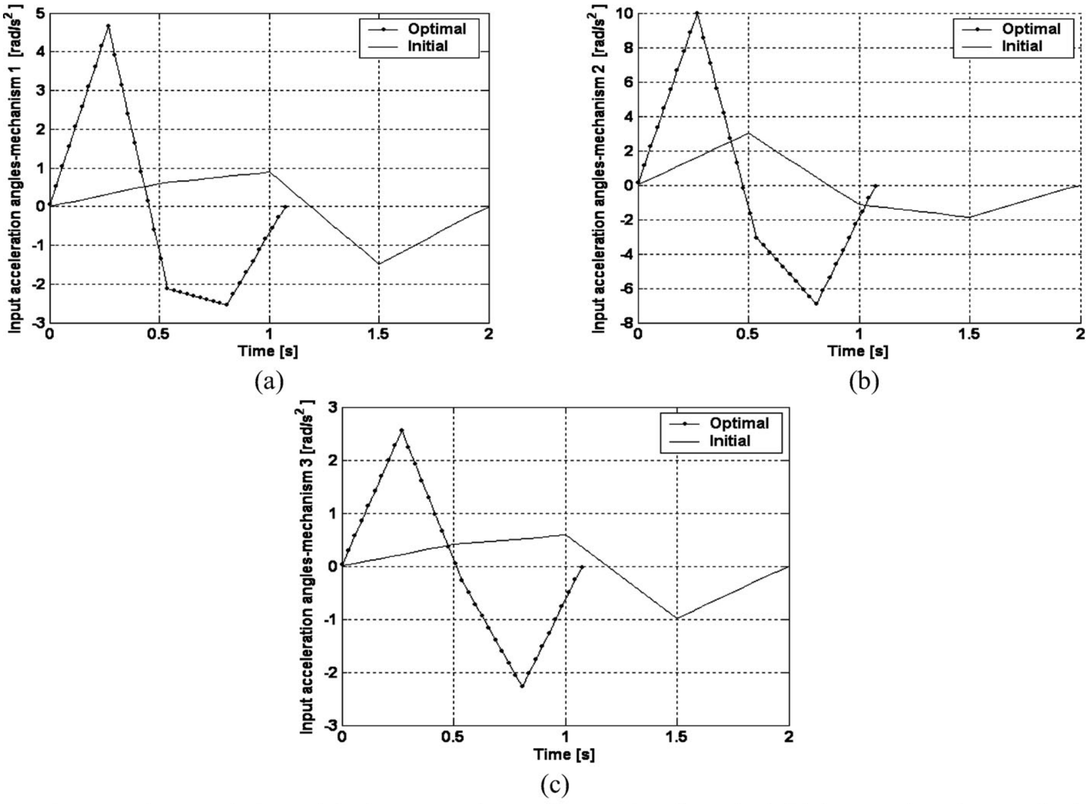
(a)

(b)

(c)

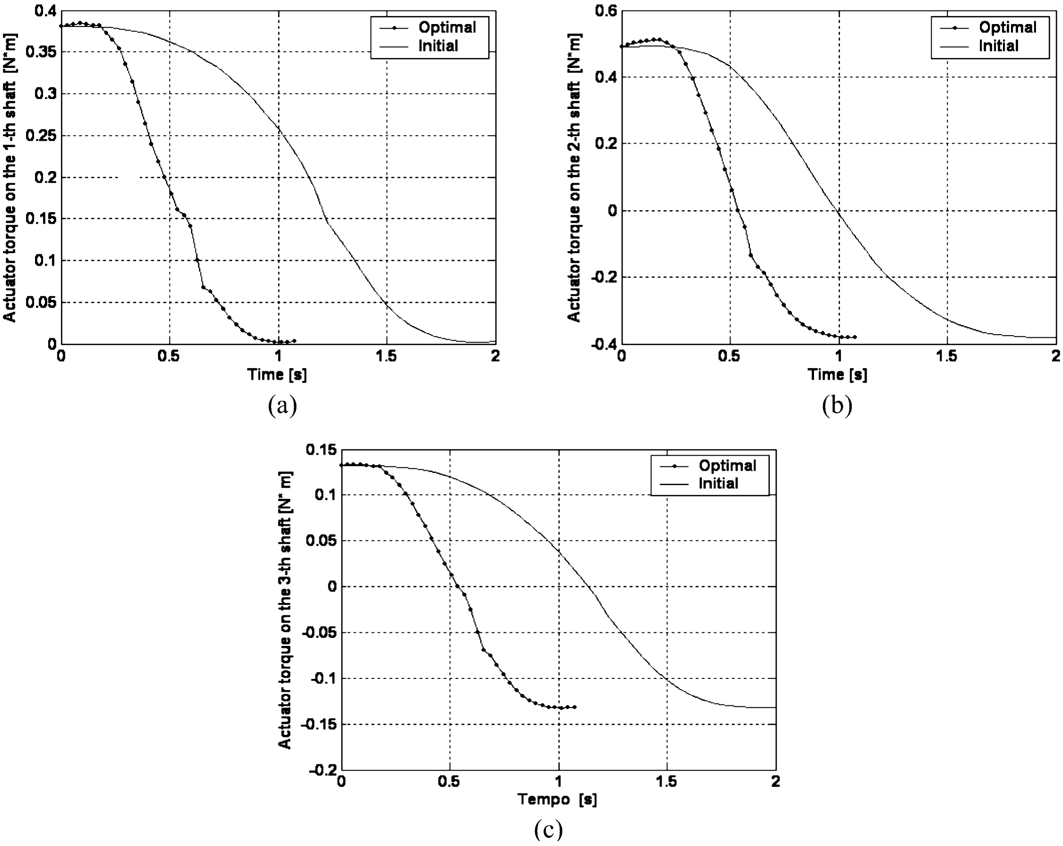
Fig. 9. Time evolution of the input crank angular velocities for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 7. (a) Leg mechanism 1. (b) Leg mechanism 2. (c) Leg mechanism 3.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press



(a) (b) (c)

Fig. 10. Time evolution of the input crank angular accelerations for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 7. (a) Leg mechanism 1. (b) Leg mechanism 2. (c) Leg mechanism 3.



(a) (b) (c)

Fig. 11. Time evolution of the actuator torques on input cranks for initial guess (continuous line) and optimal solution (dotted line) for the numerical example with CaPaMan in Fig. 7. (a) Leg mechanism 1. (b) Leg mechanism 2. (c) Leg mechanism 3.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

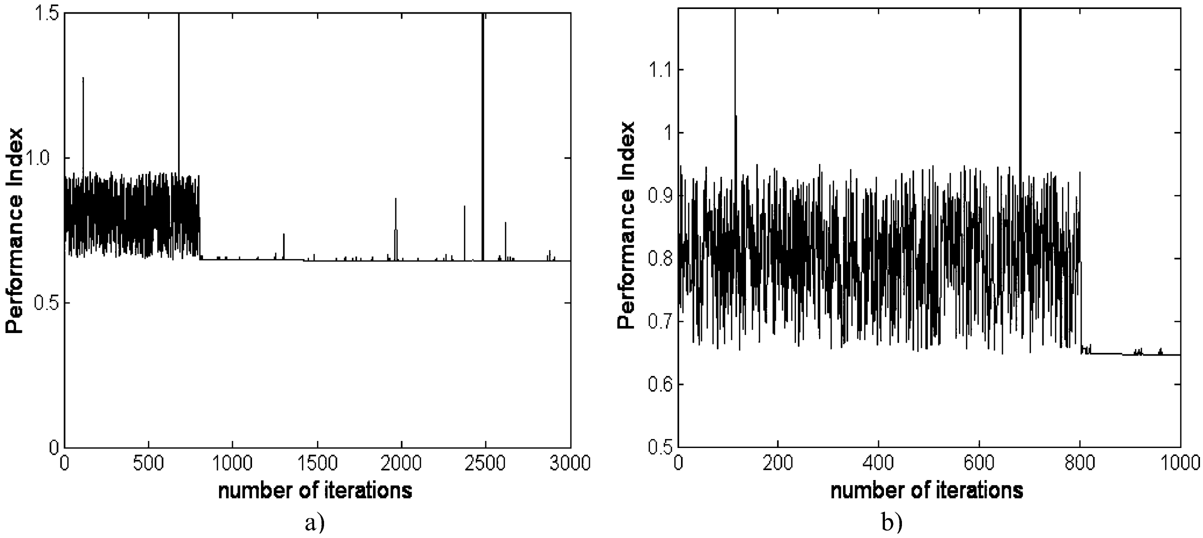


Fig. 12. Evolution of the objective function for the case of Fig. 7 as function of the number of iterations. (a) Full plot. (b) Zoomed view from 0 to 1000 iterations.

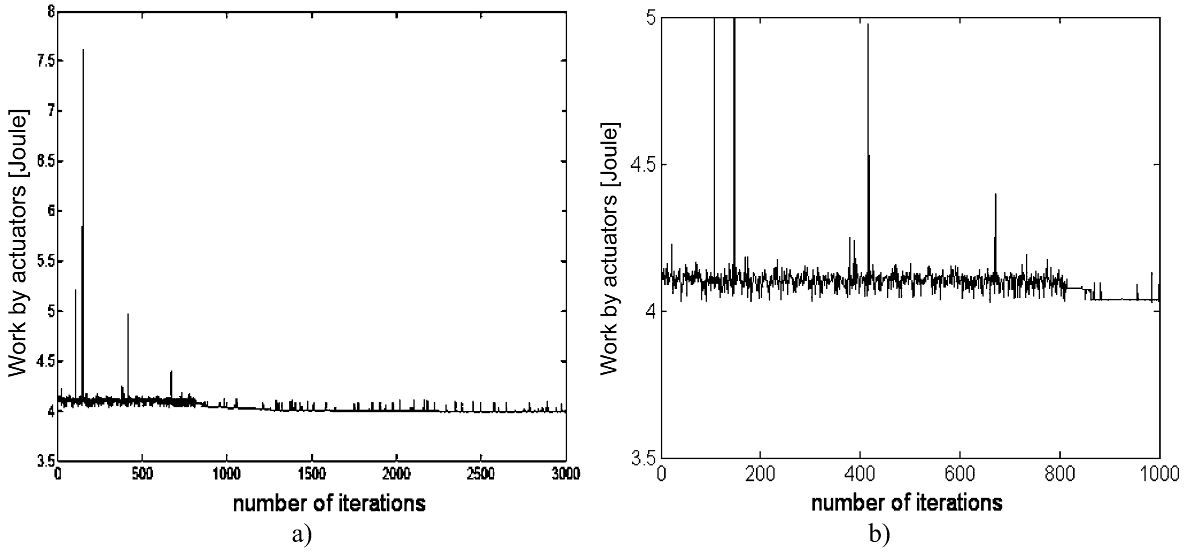


Fig. 13. Evolution of the total manipulation energy for the case of Fig. 7 as function of the number of iterations. (a) Full plot. (b) Zoomed view from 0 to 1000 iterations.

Figure 8 shows the initial guess and optimal trajectories of the input crank angles. It can be observed that although the total travelling time has been strongly reduced, the optimal curves are smooth.

Figure 9 shows the initial guess and optimum value of angular velocity for the input cranks. It is worth noting that the maximum values of angles' velocity in the optimum case are almost twice the maximum values velocity in the initial guess. Thus, using the optimal solution, desired motion of CaPaMan could be achieved almost in half time with respect to the initial solution. Namely, the final position is reached in 1.1 s in the optimum solution and in 2 s in the initial guess case, as also reported in Table II. Moreover, the maximum values of angular velocity for the input cranks in the optimum solution are 2.3 rad/s at the most [see Fig. 9(b)], and this value is still within the feasible working ranges for the actuators of CaPaMan. Similar considerations can be deduced for acceleration plots in Fig. 10. In fact, the maximum value of acceleration for the optimum solution is limited to 10 rad/s$^2$ as shown in Fig. 10(b). Moreover, the crank angle accelerations for the optimum solution in Fig. 10 have a feasible smooth profile too. In addition, both velocities and accelerations do not show high frequency oscillations that can be dangerous for durability of actuator operation.

Figure 11 shows a comparison of initial guess and optimal actuator torque on the input shafts. It can be observed in Fig. 11 that the optimal torque has a smooth shape, oscillations of torque are limited, and a feasible maximum value of the torque is obtained as equal to 0.5 Nm [see Fig. 11(b)]. Moreover, the maximum values of the torque that are required for the optimum solution do not increase significantly with respect to initial guess.

Figure 12 shows the evolution of the objective function $f$ as function of the number of iterations. In particular, the objective function $f$ converges to a value of 0.63 from an initial value of 1.00 after less than 800 iterations with a performance improvement of 37% as reported in Table II.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

Moreover, Fig. 13 shows the evolution of the energy as function of the number of iterations. In particular, the needed energy $E$ decreases from 4.11 to 3.90 Nm/s$^2$ with performance improvement of 5.1% as also reported in Table II. Thus, an improvement of all the components of the objective function $f$ has been achieved as summarized in Table II.

It is important to remind that a random optimisation has been used for the proposed case of study. The initial population has been considerably evolved, and after several generations, the population has converged to an optimal solution. However, in the case of conflicting objectives, usually, the set of optimal solutions contains more than one solution. Thus, the optimum result represents a compromise solution among the functions that contribute to the multi-objective function given by Eq. (9). Of course, in the case the user changes the weighting coefficients, different results will be obtained.

## 6. Conclusions

A general optimum path planning procedure has been proposed as applied specifically to parallel manipulators by using suitable formulation for kinematics and dynamics of path motion. In particular, optimality criteria have been identified in work by actuators and travelling time of the manipulator trajectory. A numerical case of study is illustrated by referring to the parallel manipulator named as CaPaMan. The results of the optimum procedure show the engineering feasibility of the proposed formulation in order to improve the dynamics performance of a parallel manipulator and to reduce power consumption in a path planning application.

## Acknowledgments

The first author thanks the University of Cassino for a grant in the frame of the international exchange program, which has permitted him to spend a period of study in the year 2003 at the Federal University of Uberlandia, Uberlandia, Brazil. The third author is thankful to FAPEMIG for a partial financial support. The forth and fifth authors are thankful to the Brazilian Ministry of Education (Capes) for a grant, which has permitted them to spend a sabbatical year at the Laboratory of Robotics and Mechatronics in Cassino, Italy.

## References

1. J.-P. Merlet, *Parallel Robots* (Springer Verlag, Dordrecht, Germany, 2005).
2. D. Stewart, "A Platform with Six Degrees of Freedom," *Proceeding of the Institute of Mechanical Engineers*, London, U.K. **180** (1965) pp. 371–386.
3. R. Clavel, "DELTA: A Fast Robot with Parallel Geometry," *Proceedings of the 18th International Symposium on Industrial Robots*, Lausanne, Switzerland (1988) pp. 91–100.
4. C. Gosselin and J. Angeles, "The optimum kinematic design of a planar three-degree-of-freedom parallel manipulator," *ASME J. Mech., Transm., Autom. Design* **110**, 35–41 (1988).
5. J. P. Merlet and C. Gosselin, "Nouvelle architecture pour manipulateur parallèle a six degrées de liberté," *Mech. Mach. Theory* **26**(1), 77–90 (1991).
6. K. Miller, "Experimental Verification of Modelling of Delta Robot Dynamics by Direct Application of Hamilton's Principle," *Proceedings of the IEEE International Conference on Robotics and Automation*, Nagoya, Japan (1995) pp. 532–537.
7. L. W. Tsai and R. Stamper, "A Parallel Manipulator with Only Translational Degrees of Freedom," *Proceedings of the ASME DETC*, Irvine, California (1996), CD Proceedings, Paper DETC1996/MECH-1152.
8. J. Wang and C. M. Gosselin, "Kinematic analysis and singularity representation of spatial five-degree-of-freedom parallel mechanisms," *J. Robot. Syst.* **14**(2), 851–869 (1997).
9. V. Parenti-Castelli, R. Di Gregorio and F. Bubani, "Workspace and Optimal Design of a Pure Translation Parallel Manipulator," *Proceedings of the XIV National Congress on Applied Mechanics*, Como, Italy (1999) Paper 17.
10. R. Di Gregorio, "A new family of spherical parallel manipulators," *Int. J. Robot.* **20**(4), 353–358 (2002).
11. A. Kosinska, M. Galicki and K. Kedzior, "Design of parameters of parallel manipulators for a specified workspace," *Int. J. Robot.* **21**(5), 575–579 (2003).
12. M. Karouia and J. M. Hervé, "Non-overconstrained 3-dof spherical parallel manipulators of type: 3-RCC, 3-CCR, 3-CRC," *Int. J. Robot.* **24**(1), 85–94 (2006).
13. M. Ceccarelli, "A new 3 dof spatial parallel mechanism," *Mech. Mach. Theory* **32**(8), 895–902 (1997).
14. M. Ceccarelli and G. Figliolini, "Mechanical Characteristics of CaPaMan (Cassino Parallel Manipulator)," *Proceedings of the 3rd Asian Conference on Robotics and its Application*, Tokyo, Japan (1997) pp. 301–308.
15. M. Ceccarelli and G. Carbone, "A stiffness analysis for CaPaMan (Cassino Parallel Manipulator)," *Mech. Mach. Theory* **37**(5), 427–439 (2002).
16. J. C. M. Carvalho and M. Ceccarelli, "A Dynamic Analysis for Cassino Parallel Manipulator," *Proceedings of the 10th IFToMM World Congress*, Oulu, Finland **3** (1999) pp. 1202–1207.
17. J. C. M. Carvalho and M. Ceccarelli, "The Inverse Dynamics of Cassino Parallel Manipulator," *Proceedings of the 2nd Workshop on Computational Kinematics*, Seoul, Korea (2001) pp. 301–308.
18. J. C. M. Carvalho and M. Ceccarelli, "A closed form formulation for the inverse dynamics of Cassino Parallel Manipulator," *J. Multibody Syst. Dynam.* **5**(2), 185–210 (2001).
19. E. Ottaviano, C. M. Gosselin and M. Ceccarelli, "Singularity Analysis of CaPaMan: A Three-Degree of Freedom Spatial Parallel Manipulator," *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea (2001) pp. 1295–1300.
20. F. Pugliese, Experimental Validation of CaPaMan (Cassino Parallel Manipulator) *Master Thesis* (Cassino, Italy: LARM, University of Cassino, 1998).
21. P. D. Fino, Dynamic Simulation of CaPaMan (Cassino Parallel Manipulator) *Master Thesis* (Cassino, Italy: LARM, University of Cassino, 1999).
22. M. Ceccarelli, F. Pugliese and J. C. M. Carvalho, "An Experimental System for Measuring CaPaMan Characteristics," *Proceedings of the 8th International Workshop on Robotics in Alpe-Adria-Danube Region*, Munich, Germany (1999) pp. 31–36.
23. J. C. M. Carvalho and M. Ceccarelli, "Seismic Motion Simulation Based on Cassino Parallel Manipulator," *Proceedings of the XVth Brazilian Congress on Mechanical Engineering*, Campinas, Brazil (1999), CD Proceedings.
24. M. Galvagno, Application and Experimental Validation of CaPaMan (Cassino Parallel Manipulator) as Earthquake Simulator *Master Thesis* (Cassino, Italy: LARM, University of Cassino, 2001).
25. M. Ceccarelli, E. Ottaviano and M. Galvagno, "A 3-DOF Parallel Manipulator as Earthquake Motion Simulator," *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision*, Singapore (2002) Paper P1073.

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press

26. C. S. Lin, P. R. Chang and J. Y. S. Luh, "Formulation and optimisation of cubic polynomial joint trajectories for industrial robots," *IEEE Trans. Autom. Control* **28**, 1066–1073, (1983).
27. K. G. Shin and N. D. Mckay, "A dynamic programming approach to trajectory planning of robotic manipulators," *IEEE Trans. Autom. Control* **AC-31**(6), 491–500 (1986).
28. Y. C. Chen, "Solving robot trajectory planning problems with uniform cubic B-Splines," *Optimal Control Appl. Methods* **12**, 247–262 (1991).
29. S. Zeghloud, B. Blanchard and J. A. Pamanes, "Optimisation of kinematics performances of manipulators under specified task conditions," *Proceedings of the 10th CISM-IFToMM Symposium Romansy 10*, Gdansk, Poland (1994) pp. 247–252.
30. J. Zhao and S. X. Bai, "Load distribution and joint trajectory planning of coordinated manipulation for two redundant robots," *Mech. Mach. Theory* **34**, 1155–1170 (1999).
31. C. G. Johnson and D. Marsh, "Modelling robot manipulators with multivariate B-Splines," *Int. J. Robot.* **17**(3), 239–247 (1999).
32. F. C. Park, J. Kim and J. E. Bobrow, "Algorithms for Dynamics–Based Robot Motion Optimisation," *Proceedings of the 10th World Congress on the Theory of Machines and Mechanics*, Oulu, Finland (1999) pp. 1216–1221.
33. Y. K. Choi, J. H. Park, H. S. Kim and J. H. Kim, "Optimal trajectory planning and sliding mode control for robots using evolution strategy," *Int. J. Robot.* **18**(4), 423–428 (2000).
34. S. F. P. Saramago and M. Ceccarelli, "An optimum robot path planning with payload constraints," *Int. J. Robot.* **20**, 395–404 (2002).
35. S. F. P. Saramago and M. Ceccarelli, "Effect of some numerical parameters on a path planning of robots taking into account actuating energy," *Mech. Mach. Theory* **39**(3), 247–270 (2004).
36. S. F. P. Saramago and S. J. Valder, "Trajectory modeling of robots manipulators in the presence of obstacles," *J. Optim. Theory Appl.* **110**(1), 17–34 (2001).
37. D. E. Goldberg, Genetic Algorithms in Search Optimisation, and Machine Learning (Addison-Wesley, Reading, MA, 1989).
38. H. Eschenauer, J. Koski and A. Osyczka, *Multicriteria Design Optimisation*, (Springer-Verlag, Berlin, Germany, 1990).
39. Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs* (Springer-Verlag, Berlin, Germany, 1995).
40. J. D. Foley, A. Van. Dam, S. K. Feiner and J. F. Hughes, *Computer Graphics: Principles and Practice*, 2nd ed. (Addison-Wesley, Reading, MA, 1990).
41. C. R. Houck, J. A. Joinez and M. G. Kay, A Genetic Algorithm for Function Optimisation: A Matlab Implementation, NCSU-IE Technical Report (1995).

https://doi.org/10.1017/S0263574707003839 Published online by Cambridge University Press