Robotica (2011) volume 29, pp. 683–690. © Cambridge University Press 2010
doi:10.1017/S0263574710000500

Architecture optimization of 4PUS+1PS parallel manipulator

G. Abbasnejad, H. M. Daniali* and A. Fathi

Department of Mechanical Engineering, Babol University of Technology, P.O. Box 484, Babol, Iran

(Received in Final Form: August 5, 2010. First published online: September 10, 2010)

SUMMARY

The main goal of this paper is the design of 4PUS+1PS parallel manipulator, using an optimization problem that takes into accounts the characteristics of the workspace and dexterity. The optimization problem is formulated considering constraints on actuated and passive joint limits. A comparison between quantum particle swarm Optimization (QPSO) and PSO is developed. Two numerical examples are presented, which reveal the advantages of QPSO to PSO. Moreover, it is shown that by introducing the dexterity index as a quality measure throughout the workspace, the parallel manipulator is improved at the cost of a minor reduction in its workspace.

KEYWORDS: Parallel manipulator; Workspace; Dexterity index; Optimization; PSO; QPSO.

1. Introduction

Parallel manipulators have received more and more attention over the last two decades. This popularity is due to the fact that they possess some specific advantages over their serial counterparts; such as better dynamic performance, higher stiffness, and load carrying capacity. The most studied type of parallel manipulator is without doubt the so-called general Gough–Stewart platform, a fully parallel manipulator introduced by Gough as a universal tire-testing machine in the early 1950s[1] and proposed as a flight simulator by Stewart in 1965.[2] However, as 6-degree-of-freedom (DOF) is not always required for many applications. In the recent years, many parallel manipulators with fewer than 6-DOF have been introduced, such as the famous DELTA robot with three translational DOF,[3] 3-universal-prismatic-universal (UPU) and 3-prismatic-revolute-cylindrical (PRC) parallel robots with pure translational motions,[4,5] 4-DOF parallel manipulators with parallel active limbs,[6,7] fully isotropic 3T1R, 4-DOF parallel mechanism and 2T2R, 4-DOF hybrid parallel manipulator.[8]

One major area of application of parallel manipulators is flight and motion simulations. For this type of application, rotational freedoms play a major role, while translations are of lesser importance.[9] However, one translational freedom, the heave, is of great significance in flight simulation.[9] Unfortunately, Gough–Stewart platforms most commonly used as flight simulators cannot provide good heave simulation. Hence, if one were to choose a subset of the platform freedoms for the purposes of flight simulation, a natural choice would be to keep the three rotational freedoms and only one translation.[9–11] Therefore, 4PUS+PS parallel manipulator, as depicted in Fig. 1, can be used in the design of flight and motion simulators. The manipulator is composed of a moving platform attached to the fixed base through four active and one passive legs. The aim of using the passive leg is to limit the DOF to the desired ones.

Kinetostatic performance measures how well the system behaves with regard to force and motion transmission. One of the most common performance indices found in the literature is dexterity.[12–17] Several dexterity measures, such as service angle, manipulability, condition number[13,18] have been extensively developed and applied to both serial and parallel mechanisms. On the other hand, compared to their serial counterparts, parallel manipulators suffer a main disadvantage of relatively small workspace.[19–21] Thus, it is natural to optimize the parallel manipulators to achieve maximum total workspace volume. However, it is possible that a parallel manipulator optimized for total workspace may result in a manipulator with undesirable kinematic characteristics such as poor dexterity. Therefore, some others optimized parallel manipulators based on a mixed performance that is a weighted sum of dexterity index and the workspace.[22–24]

In the recent years, a number of evolutionary algorithms have been developed to solve multiobjective optimization problems. Genetic algorithm (GA)[25–27] and simulated annealing[28,29] have attracted some attentions. They do not require conventional mathematical assumptions and thus possess better global search abilities than the conventional optimization algorithms. Particle swarm optimization (PSO) was introduced by Kennedy and Eberhart as an alternative to GA.[30–32] PSO is a population-based optimization tool, which could be easily applied to solve various function optimization problems, as well as the problems that can be transformed to functional optimization problems. Qingsong used PSO to optimal design of a novel micromanipulator.[23] He showed that PSO shares the same abilities of GA to handle the optimization problem, but with some advantages such as faster convergence rate and leading to the same fitness value by different initial starting values. In refs. [33, 34], Sun *et al.* introduced quantum theory into PSO and proposed a quantum-behaved PSO (QPSO) algorithm, which guarantees finding optimal solutions in the search space. Moreover, experiment results on some widely used benchmark functions show that the QPSO works better than the standard PSO and should be a promising algorithm.[33–35] In this work, PSO

* Corresponding author. E-mail: mohammadi@nit.ac.ir

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

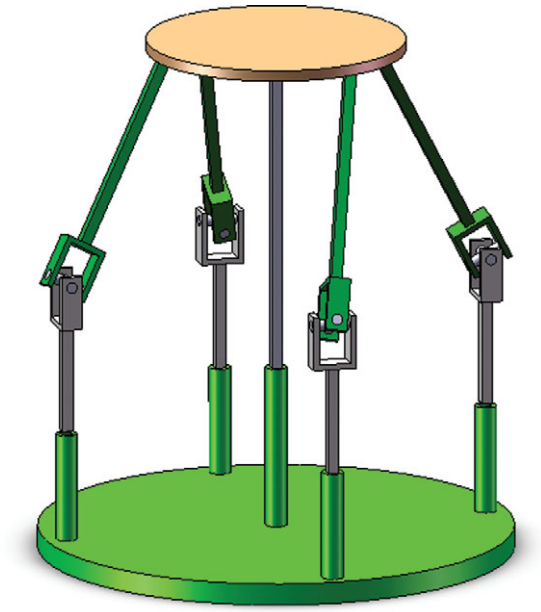


Fig. 1. 4PUS+1PS parallel manipulator.

and QPSO will be employed and compared for the optimal designs of 4-PUS+1PS parallel manipulator. Moreover, we optimize the manipulator based on a mixed performance index, namely, a weighted sum of dexterity index and the volume of the workspace.

2. Description and Mobility Analysis of the Manipulator

The architecture of the manipulator is composed of a moving platform, a fixed base, four PUS-type active legs with the linear actuators, and one PS-type passive leg to limit the translational DOF of the moving to the z-axis. The PUS-type active legs connect the moving platform to the base through an active prismatic actuator P at B_i, a universal joint U at C_i, and a spherical joint S at A_i. Moreover, the passive PS leg connects the moving platform to the base with prismatic and spherical joints, as depicted in Fig. 2. For the sake of analysis, a fixed Cartesian reference coordinate frame Ox,y,z is fixed to the base at point O, and a moving coordinate frame Pu,v,w is fixed to the moving platform at point P. According to this particular design, when four actuated joint variables, r_1, r_2, r_3, and r_4 have equal magnitudes, the P-frame is parallel to O-frame. Angle α is defined between vectors $\overrightarrow{OA}_1$ and $\overrightarrow{OA}_2$, which are symmetric with the vectors $\overrightarrow{OA}_3$ and $\overrightarrow{OA}_4$, respectively. The parameters, φ, ψ, and θ are, respectively, defined as the three Euler angles rotating about the z-, x-, and y-axes of the fixed reference frame in sequence. The transformation from the moving frame to the fixed one can be described by a position vector $\mathbf{r}_0$ along z-axis, and a 3×3 rotation matrix R_O^P, which can be expressed as the following:

$$\begin{aligned}\mathbf{R}_O^P &= \mathbf{R}_y(\theta)\mathbf{R}_x(\psi)\mathbf{R}_z(\phi)\\ &= \begin{bmatrix} c\theta c\phi + s\psi s\theta s\phi & -c\theta s\phi + s\psi s\theta c\phi & c\psi s\theta \\ c\psi s\theta & c\psi c\theta & -s\psi \\ -s\theta c\phi + s\psi c\theta s\phi & s\theta s\phi + s\psi c\theta c\phi & c\psi c\theta \end{bmatrix},\end{aligned} \tag{1}$$

where c and s stand for cos and sin, respectively.

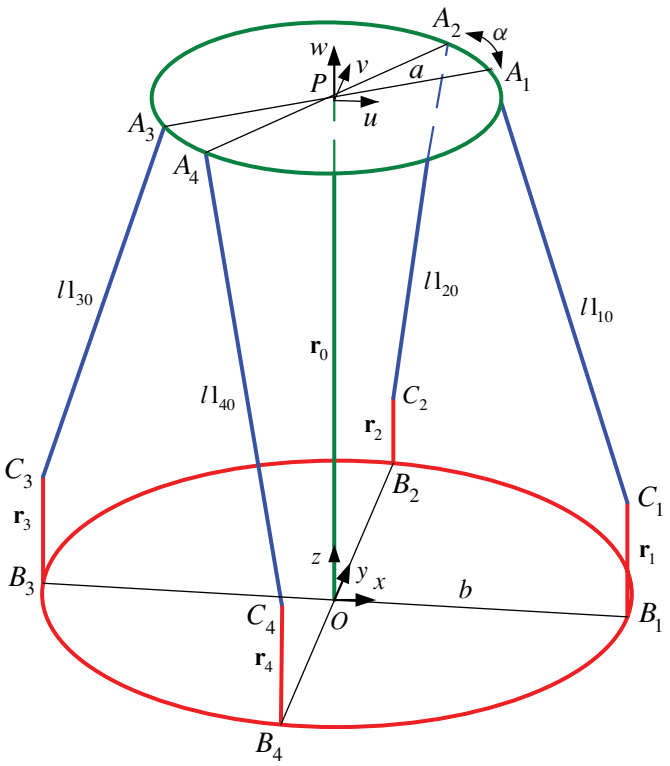


Fig. 2. The geometric model of 4PUS+1PS parallel manipulator.

The DOF of the manipulator is calculated by the general **G**rübler–**K**utzbach formula as follows:

$$F = \lambda(n - g - 1) + \sum_{i=1}^{g} f_i = 6 \times (11 - 14 - 1) + 28 = 4, \tag{2}$$

where λ represents the order of the task space, n is the number of links, g is the number of joints, and f_i denotes the DOF of joint i. Moreover, the DOF of the manipulator are translation along z-axis and the three rotations.

3. Jaconian Matrix Generation

Let $\mathbf{d} = [r_1\ r_2\ r_3\ r_4]^T$ be a four-dimensional array of the joint variables and $\mathbf{x} = [r_0\ \psi\ \theta\ \varphi]^T$ be a four-dimensional array of variables, which describe the elevation and the orientation of the moving platform in the Cartesian space. The position vectors $\mathbf{b}_i$ of points $B_i (i = 1, 2, 3, 4)$ in the reference frame O and the position vectors a_i^p and $\boldsymbol{a}_i$ of points A_i in the frames P and O, respectively, can be expressed as follows:

$$\mathbf{b}_1 = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}, \quad \mathbf{b}_3 = \begin{bmatrix} -b \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{b}_4 = \begin{bmatrix} 0 \\ -b \\ 0 \end{bmatrix},$$

$$\mathbf{a}_1^p = \begin{bmatrix} a\cos\left(\dfrac{\pi - 2\alpha}{4}\right) \\ a\sin\left(\dfrac{\pi - 2\alpha}{4}\right) \\ 0 \end{bmatrix}, \quad \mathbf{a}_2^p = \begin{bmatrix} a\cos\left(\dfrac{\pi + 2\alpha}{4}\right) \\ a\sin\left(\dfrac{\pi + 2\alpha}{4}\right) \\ 0 \end{bmatrix},$$

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

$$\mathbf{a}_3^p = \begin{bmatrix} -a\cos\left(\frac{\pi+2\alpha}{4}\right) \\ -a\sin\left(\frac{\pi+2\alpha}{4}\right) \\ 0 \end{bmatrix}, \quad \mathbf{a}_4^p = \begin{bmatrix} -a\cos\left(\frac{\pi+2\alpha}{4}\right) \\ -a\sin\left(\frac{\pi+2\alpha}{4}\right) \\ 0 \end{bmatrix}$$

$$\mathbf{a}_i = \mathbf{R}_O^P \mathbf{a}_i^p \tag{3}$$

From Fig. 2, the vector closure equation for the ith PUS leg can be written as follows:

$$\mathbf{b}_i + r_i\hat{\mathbf{k}} + l\mathbf{l}_{i0} = r_0\hat{\mathbf{k}} + a_i, \quad i = 1\ldots4, \tag{4}$$

where $\mathbf{l}_{i0}$ is the unit vector representing the direction of vector $\overrightarrow{B_iC_i}$, while r_i represents its magnitude; $\hat{\mathbf{k}}$ is the unit vector along z-axis. Differentiating both sides of Eq. (4), with respect to time, yields

$$v_0\hat{\mathbf{k}} + \boldsymbol{\omega}_p \times \mathbf{a}_i = v_i\hat{\mathbf{k}} + l\boldsymbol{\omega}_i \times \mathbf{l}_{i0}, \quad i = 1\ldots4, \tag{5}$$

where $\times$ denotes the cross product operator between vectors; v_0 is the linear velocity of the moving platform along z-axis; $\boldsymbol{\omega}_p$ represents the angular velocity vector of the moving platform; v_i is the velocity of the ith linear actuator and $\boldsymbol{\omega}_i$ represents the angular velocity vector of link C_iA_i.

The passive variables $\boldsymbol{\omega}_i$ can be eliminated by left multiplying both sides of Eq. (5) by $\mathbf{l}_{i0}$, namely,

$$\mathbf{l}_{i0}^T v_0\hat{\mathbf{k}} + \mathbf{l}_{i0}^T(\boldsymbol{\omega}_p \times \mathbf{a}_i) = v_i\mathbf{l}_{i0}^T\hat{\mathbf{k}}, \quad i = 1\ldots4. \tag{6}$$

In the light of the even permutation rules for triple product, Eq. (6) can be written as follows:

$$\mathbf{l}_{i0}^T v_0\hat{\mathbf{k}} + (\mathbf{a}_i \times \mathbf{l}_{i0})^T\boldsymbol{\omega}_p = v_i\mathbf{l}_{i0}^T\hat{\mathbf{k}}, \quad i = 1\ldots4. \tag{7}$$

One can write Eq. (7) in matrix form as follows:

$$\mathbf{J}_x\dot{\mathbf{x}} = \mathbf{J}_q\dot{\mathbf{d}}, \tag{8}$$

where

$$\mathbf{J}_x = \begin{bmatrix} l_{10z} & (\mathbf{a}_1 \times \mathbf{l}_{10})^T \\ l_{20z} & (\mathbf{a}_2 \times \mathbf{l}_{20})^T \\ l_{30z} & (\mathbf{a}_3 \times \mathbf{l}_{30})^T \\ l_{40z} & (\mathbf{a}_4 \times \mathbf{l}_{40})^T \end{bmatrix}, \quad \mathbf{J}_q = \begin{bmatrix} \mathbf{l}_{10}^T\hat{\mathbf{k}} & 0 & 0 & 0 \\ 0 & \mathbf{l}_{20}^T\hat{\mathbf{k}} & 0 & 0 \\ 0 & 0 & \mathbf{l}_{30}^T\hat{\mathbf{k}} & 0 \\ 0 & 0 & 0 & \mathbf{l}_{40}^T\hat{\mathbf{k}} \end{bmatrix},$$

$$\dot{\mathbf{x}} = \begin{bmatrix} \boldsymbol{v}_0 \\ \boldsymbol{\omega}_p \end{bmatrix}, \quad \dot{\mathbf{d}} = [\dot{r}_1 \quad \dot{r}_2 \quad \dot{r}_3 \quad \dot{r}_4]^T.$$

Moreover, Eq. (8) can be written as follows:

$$\dot{\mathbf{d}} = \mathbf{J}\dot{\mathbf{x}}, \tag{9}$$

where

$$\mathbf{J} = \mathbf{J}_q^{-1}\mathbf{J}_x. \tag{10}$$

The *condition number* of a given matrix is well known to provide a measure of invertibility of the matrix.[36] It is thus natural that this concept found its way in this context. Indeed, the condition number of the Jacobian matrix was proposed as a figure of merit to minimize when designing manipulators for maximum accuracy.[37] In order to determine the condition number of the Jacobian matrices, we must order their singular values from largest to smallest. However, in this case, the singular value associated with positioning, is dimensionless, while those associated with orientation have units of length, thereby making impossible such an ordering. This dimensional in-homogeneity can be resolved by introducing a normalizing *characteristic length*.[38] Upon dividing the three *orientation columns*, i.e., the last three columns, of the Jacobian by radius of the base platform, a nondimensional Jacobian is obtained whose singular values are nondimensional as well.[39]

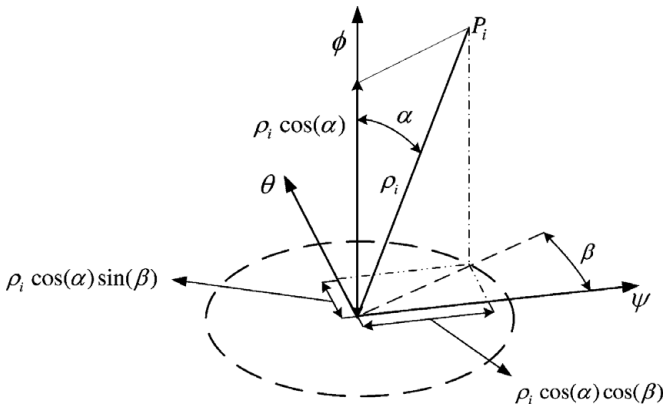


Fig. 3. Determination of boundary points.

4. Volume of Workspace

Before a workspace can be realized, the physical constraints such as the limits of spherical joint cone angle and the actuators motion ranges must be considered. We mainly consider the effects imposed by spherical joints since they are not easy to manufacture. Defining a cone angle, the manipulator is assumed in its home position, where $r_i = 0$ for $i = 1\ldots4$. In this pose, the angle between any of links C_iA_i and normal vector of the moving platform is χ_0. But changing the orientation of the moving platform yields a new angle between links C_iA_i and the normal vector of the moving platform which is termed χ_i for $i = 1\ldots4$. Then, the difference between χ_i and χ_0 is defined as the cone angle for the associated spherical joint, namely,

$$\xi_i = |\chi_i - \chi_0|. \tag{11}$$

We assume this angle as the cone angle in the following discussion. Workspace is defined as all poses that can be reached by the moving platform. As it is discussed in Section 3, the manipulator at hand has 4 DOF, namely, the elevation of the moving platform and its orientation, which can be defined by the three Euler angles. Therefore, the volume of its four-dimensional workspace is given as follows:

$$V = \int_{h_{0\min}}^{h_{0\max}} \int_{\varphi_{\min}}^{\varphi_{\max}} \int_{\theta_{\min}}^{\theta_{\max}} \int_{\psi_{\min}}^{\psi_{\max}} d\psi d\theta d\phi dz \tag{12}$$

One can define a spherical coordinate as is depicted in Fig. 3 for the three rotational degrees of freedom. Therefore,

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

Eq. (12) can be written as:

$$V = \int_{h_{0\min}}^{h_{0\max}} \int_0^{\pi} \int_0^{2\pi} 1/3\rho^3 \sin(\alpha) d\beta d\alpha dz \tag{13}$$

where ρ, α, and β are the spherical coordinates of this three-dimensional space, as depicted in Fig. 3.

5. Global Dexterity Index

Several performance criteria could be taken into account, as for example service angle, manipulability, and condition number. But a global dexterity index (GDI) based on the condition number is defined in [13] to characterize the kinematic accuracy of a manipulator over its whole workspace as follows:

$$\text{GDI} = \frac{\int_V (1/k)\, dV}{V}, \tag{14}$$

where V is the total workspace volume and k denotes the condition number of the homogeneous Jacobian. Substituting the values of dV from Eq. (13) into Eq. (14) yields.

$$\text{GDI} = \frac{1}{V} \int_{h_{0\min}}^{h_{0\max}} \int_0^{\pi} \int_0^{2\pi} \int_0^{\rho(\alpha,\beta,z)} \frac{1}{k} r^2 \sin(\alpha)\, dr\, d\beta\, d\alpha\, dz, \tag{15}$$

where r is understood in the cylindrical coordinates.

Finally, Eq. (15) is approximated numerically as follows:

$$\text{GDI} = \frac{1}{V} \sum_{z_i=z_{\min}}^{z_{\max}} \sum_{\alpha_i=0}^{\pi} \sum_{\beta_i=0}^{2\pi} \sum_{r_j=0}^{\rho_i} \frac{1}{k_j} r_j^2 \sin(\alpha_i)\, \Delta r\, \Delta\beta\, \Delta\alpha\, \Delta z. \tag{16}$$

6. Optimization Techniques

6.1. Particle swarm optimization

The PSO algorithm belongs to the category of swarm intelligence techniques. The swarm intelligence concepts are inspired by the social behavior of flocking animals such as swarms of birds, ants, and fish school. PSO was first developed and introduced as a stochastic optimization algorithm by Kennedy and Eberhart.[30] The approach can be viewed as a distributed behavioral algorithm that performs a multidimensional search. PSO has been found to be useful in a wide variety of optimization tasks. It is a population-based algorithm that exploits a population of individuals to probe promising regions of the search space. The individual behavior is affected either by the best-local or the best-global individual. The performance of each individual is measured using a fitness function. The population is referred as a swarm and individuals are called particles. The particles move in a multidimensional search space with adaptable velocity. In PSO, the particles keep a record of the global best position ever attained by them. This property helps the particles to search the multidimensional space faster. Let us consider an optimization problem with n-dimensional design space. Assume that there are m particles in a swarm, and ith particle in a swarm is represented as a vector.

$$\begin{aligned} &\mathbf{x}_i \in R^n, \\ &\mathbf{x}_i = [x_{i1}, x_{i2}, x_{ij} \ldots, x_{in}]^T, \quad i = 1, 2, \ldots, m. \end{aligned} \tag{17}$$

The velocity of the particle moving in the n-dimensional search space is

$$\mathbf{v}_i = [v_{i1}, v_{i2}, \ldots, v_{in}]^T, \quad i = 1, 2, \ldots, m. \tag{18}$$

In every search iteration, each particle is updated by the two "best" values, namely, pbest and gbest. The former is the best solution (fitness) it has achieved so far. The fitness value is also stored. The best position (the position giving the best fitness value) of the ith particle is recorded and represented as follows:

$$\mathbf{b}_i = [b_{i1}, b_{i2}, \ldots, b_{in}]^T, \quad i = 1, 2, \ldots m. \tag{19}$$

The latter value, which is the global best, is the best value obtained so far by any particle in the population. The index of the position vector of gbest is represented by the symbol g. Then velocity of the particles is adapted using the following equations:

$$\begin{aligned} \mathbf{v}_i(l+1) = {} & w\mathbf{v}_i(l) + c_1 r_1(\mathbf{b}_i(l) - x_i(l)) \\ & + c_2 r_2(\mathbf{b}_g(l) - \mathbf{x}_i(l)), \end{aligned} \tag{20}$$

$$\mathbf{x}_i(l+1) = \mathbf{x}_i(l) + \mathbf{v}_i(l+1), \tag{21}$$

where w is the inertia weight, c_1, c_2 represent positive acceleration constants, and r_1, r_2 are uniformly distributed random numbers $r_1, r_2 \in [0, 1]$. The first term in Eq. (20) relates to the current velocity of the swarm, the second term represents the local search, while the third one represents the global search pointing toward the optimal solution. The inertia weight w is employed to control the impact of the previous history of velocities on the current velocity of each particle. Thus, the parameter w regulates the tradeoff between the global and local exploration abilities of the swarm. The basic steps in the PSO algorithm can be summarized as follows:

(1) The swarm with random position values and random initial velocities is initialized.
(2) The velocity vector for each particle in the swarm is determined using the knowledge of the best position ever attained by each particle and the swarm as a whole and also the previous position of each particle in the swarm.
(3) The current position of each particle is modified using the velocity vector and the previous position of each particle.
(4) Steps 2 and 3 are repeated until a termination criterion is achieved.

6.2. Quantum particle swarm optimization

Sun *et al.* introduced a new optimization technique, namely, QPSO to deal with the global convergence, which cannot be guaranteed by the classical PSO.[33–34] In this algorithm,

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

according to uncertainty principle, the vectors x_i and $\mathbf{v}_i$ cannot be determined and consequently the term trajectory is meaningless. In this model, the state of a particle is depicted by wave function $\psi_{(x,t)}$, instead of position and velocity. According to this algorithm, the position of the particles are calculated as follows:

$$\begin{aligned} \mathbf{x}_{(t+1)} &= \mathbf{p}_i - \beta \times (m\mathrm{Best} - \mathbf{x}_t) \times \mathbf{ln}(1/u) \quad \text{if } k \geq .5, \\ \mathbf{x}_{(t+1)} &= \mathbf{p}_i + \beta \times (m\mathrm{Best} - \mathbf{x}_t) \times \mathbf{ln}(1/u) \quad \text{if } k < .5, \end{aligned} \tag{22}$$

where

$$\mathbf{p}_i = \varphi \times \mathbf{b}_i + (1 - \varphi) \times \mathbf{b}_g$$

$$m\mathbf{Best} = \frac{1}{m} \sum_{i=1}^{m} \mathbf{b}_i .$$

In which $m\mathbf{Best}$ is the mean of all the best positions of the population, k, u, and φ are random number distributed uniformly on [0, 1], respectively, and β is a parameter, which can be tuned to control the convergence speed of the algorithm.

7. Optimization

Design variables: As it is shown in Fig. 2, b radius of base platform, l length of links $C_i A_i$, a radius of moving platform, and the angle of γ are the parameters of the manipulator. Optimization of the manipulator involves searching for these parameters as design variables that yield to a design with desirable characteristics of robot. But to perform an optimization independent of design dimension, the first three variables are divided by Δd, the range of motion of prismatic joints. Therefore, we will work with the followings homogeneous parameters, namely, $\frac{a}{\Delta d}, \frac{b}{\Delta d}, \frac{l}{\Delta d}, \alpha$.

The Objective function: Here, we optimize the manipulator based on a mixed performance that is a weighted sum of dexterity index and the volume of the workspace. Then, the objective function is defined as[22]

$$\eta = w_d \frac{V}{V^*} + (1 - w_d)(\mathrm{GDI}), \tag{23}$$

where V is the workspace volume, weight parameter $w_d (w_d \in [0, 1])$ describes the weight of workspace volume in the mixed index, GDI represents average of the inverse of condition number in the workspace, and V^* denotes the maximum value of V to scale workspace index into the range of [0,1]. The optimization is performed in two steps; first the workspace volume of the manipulator is optimized, then the optimization is performed by considering the new index as defined in Eq. (23).

7.1. Workspace optimization

As explained earlier, to maximize the workspace of the manipulator, the objective function is defined as follows:

Max V

Subject to:

(1) $0.5 \leq \frac{a}{\Delta d} \leq 5, 0.5 \leq \frac{b}{\Delta d} \leq 5, 0.5 \leq \frac{l}{\Delta d} \leq 5,$
and $0^\circ \leq \alpha \leq 90^\circ$

(2) $0 \leq \xi_i \leq 30^\circ, \quad i = 1 \ldots 4$

(3) $0.05 \leq \frac{1}{k_i}.$

The last constraint is to avoid the design of the manipulator going near singular configurations.

The effectiveness of the QPSO and PSO algorithms is shown here through their application in the optimization problem. Concerning PSO, the results are all based on using an inertia weight, w of 0.4, and the value of acceleration constants C_1 and C_2 are set to be the same and equal to 2. Regarding to QPSO, the coefficient β decreases from 1.0 to 0.5 linearly when the algorithm is running, also the iteration number and population size are set to 120 and 30, respectively. The best result of 10 independent run is elaborated in Table I, which shows that the both algorithms converge to the same results. Comparison of the convergence

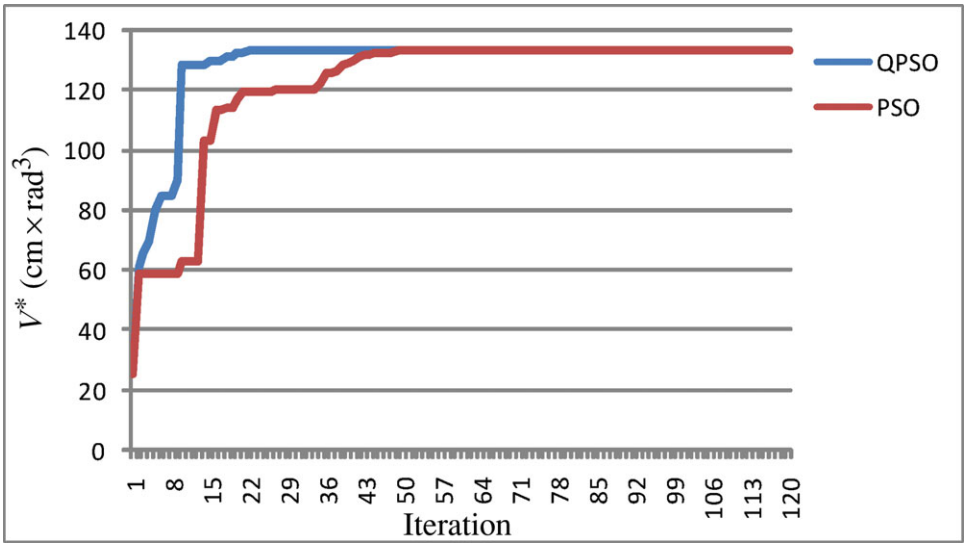


Fig. 4. Convergence of QPSO and PSO to optimize workspace.

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

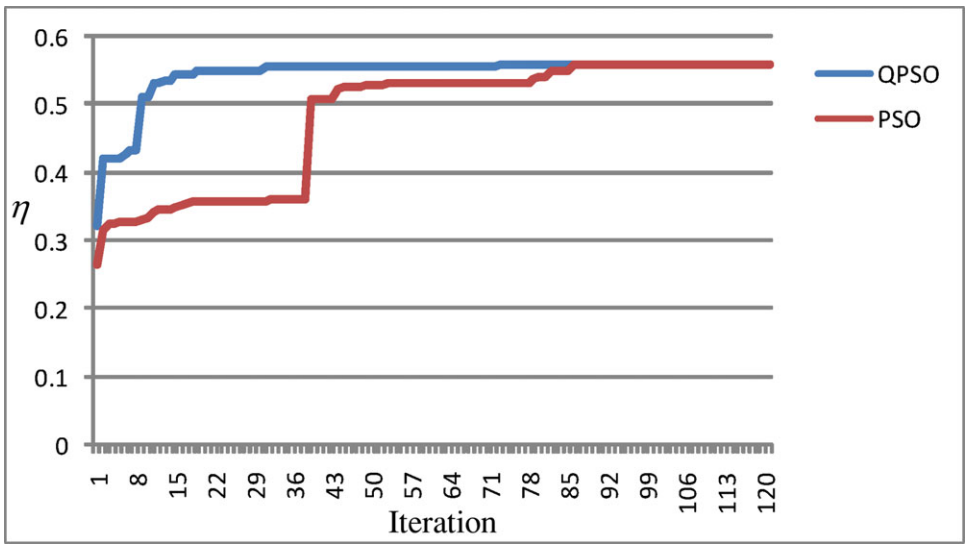


Fig. 5. Convergence of QPSO and PSO to optimize η.

process of the algorithms is presented in Fig. 4, which shows that QPSO has faster convergence speed.

7.2. Weighted optimization

In this part, to improve the manipulator performance all over its workspace, η is used as the objective function

Table I. Optimization results of workspace by QPSO and PSO.

Workspace volume (cm)(rad^3)	GDI	a/Δd	b/Δd	l/Δd	α°
133.2083	0.0836	0.500	1.5614	2.0391	54.4042

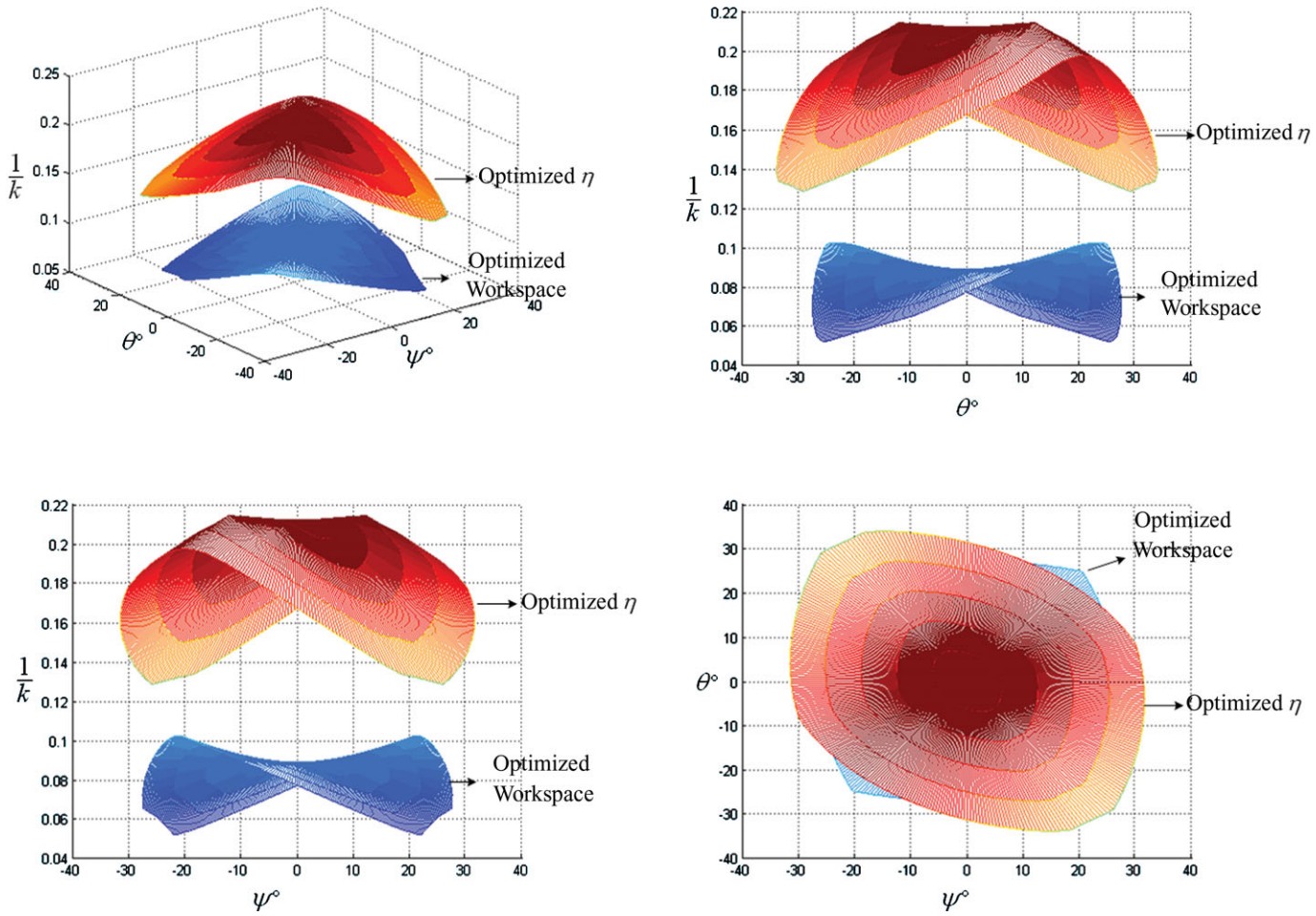


Fig. 6. The inverse of condition numbers of the designs at $h_0 = 390\,\text{cm}$, $\varphi = 0^\circ$.

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

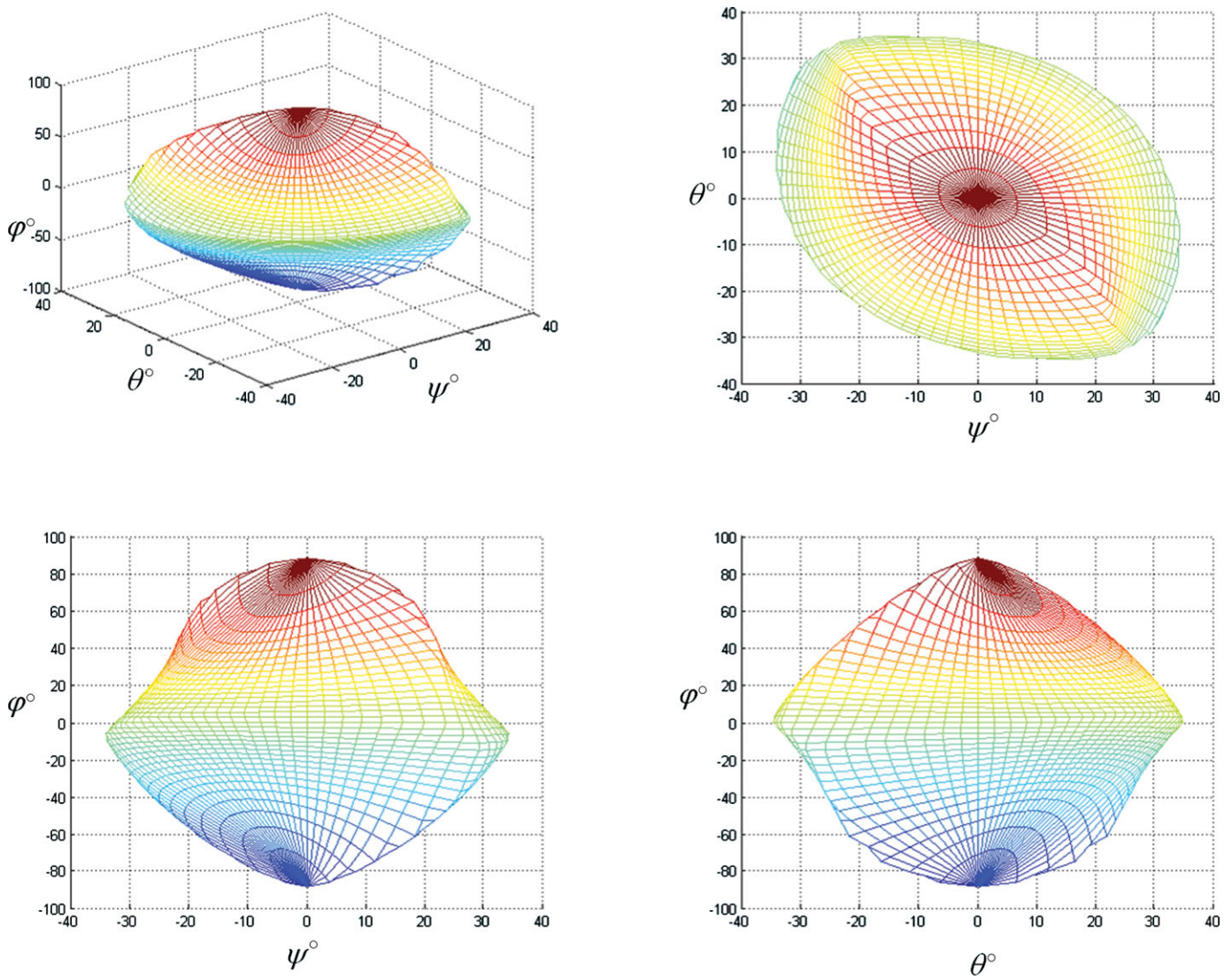


Fig. 7. The rotational workspace of the optimal manipulator for $h_0 = 370\,\text{cm}$.

with $w_d = 0.5$. Therefore, the objective function is defined as follows:

$$\text{Max}\ \eta = w_d \frac{V}{V^*} + (1 - w_d)(\text{GDI})$$

Subject to:

(1) $0.5 \le \dfrac{a}{\Delta d} \le 5, 0.5 \le \dfrac{b}{\Delta d} \le 5, 0.5 \le \dfrac{l}{\Delta d} \le 5,$
and $0^\circ \le \alpha \le 90^\circ$
(2) $0 \le \xi_i \le 30^\circ, \quad i = 1 \ldots 4$
(3) $0.05 \le \dfrac{1}{k_i}.$

All the parameters in QPSO and PSO algorithms are set as the same as in the previous section. The best optimization results of 10 independent runs are presented in Table II, and the convergence of the algorithms are shown in Fig. 5.

Result of Table II shows that GDI increased by 123.8%, while the workspace volume is reduced by only 6%. Moreover, the inverse of condition numbers of the designs, at $h_0 = 390\,\text{cm}$, $\varphi = 0^\circ$, are depicted in Fig. 6.

Table II. Optimization results of η by QPSO and PSO.

η	Volume(cm)(rad^3)	GDI	a/Δd	b/Δd	l/Δd	α°
0.5585	123.6894	0.1871	0.500	0.500	1.2835	25.87

As it is depicted in Fig. 6, using the weighted objective function leads to a manipulator with better dexterity. Moreover, the rotational workspace of the optimal manipulator for $h_0 = 390\,\text{cm}$ is shown in Fig. 7.

8. Conclusions

In this paper, the optimal design of 4PUS+1PS parallel manipulator has been presented. First, the kinematic optimization has been carried out with the goal of reaching maximal workspace volume. Then, the objective function, incorporating two performance indices, has been formulated in order to determine those architectures, which yield an optimum compromise between dexterity and workspace volume. It has been shown that the dexterity of the manipulator can be improved considerably in the cost of

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press

minor reduction in its workspace volume. PSO and QPSO evolutionary algorithms have been used to reliably find all prospective design candidates in a four-dimensional parameter space. Both PSO and QPSO lead to the same results. However, QPSO did show faster convergence rate to the optimal solutions than did the PSO.

References

1. V. E. Gough and S. G. Whitehall, "Universal Tyre Test Machine," *Ninth International Technical Congress*, FISITA (May, 1962) pp. 117–137.
2. D. Stewart, "A platform with six degrees of freedom," *Proc. Inst. Mech. Eng.* **180**(5), 371–386 (1965).
3. R. Clavel, "DELTA, a Fast Robot with Parallel Geometry," *Proceedings of 18th international symposium on industrial robots*, Lausanne (1988) pp. 91–100.
4. RDi. Gregorio and V. Parenti-Castelli, "A Translational 3-DOF Parallel Manipulator," **In:** *Advances in Robot Kinematics: Analysis and Control* (J. Lenarcic and M. L. Husty, eds.) (Kluwer Academic Publishers, Dordrecht, 1998), pp. 49–58.
5. Y. Li and Q. Xu, "Kinematic analysis and design of a new 3-DOF translational parallel manipulator," *ASME J. Mech. Des.* **128**, 729–737 (2006).
6. R. I. Alizade and C. Bayram, "Structural synthesis of parallel manipulators," *Mech. Mach. Theory* **39** (8), 857–870 (2004).
7. F. Gao, W. Li and X. Zhao, "New kinematic structures for 2-, 3-, 4-, and 5-DOF parallel manipulator designs," *Mech. Mach. Theory* **37**, 1395–1411 (2002).
8. W.-J. Chen, "A Novel 4-dof Parallel Manipulator and its Kinematic Modeling," *IEEE International Conference on Robotics and Automation*, Seoul (23–25 May, 2001) pp. 3350–3355.
9. D. Zlatanov and C. M. Gosselin, "A family of new parallel architectures with four degrees of freedom," *EJCK* **1**(1, paper 6) (2002).
10. C. Reboulet, Rapport d'avancement projet VAP, thème 7, phase 3. Rapport de Recherche 7743 (1991) CNES/DERA, Janvier.
11. J. Gallardo-Alvarado, J. Mari'a Rico-Marti'nez, and G. Alici, "Kinematics and singularity analyses of a 4-dof parallel manipulator using screw theory," *Mech. Machine Theory* **41**, 1048–1061 (2006).
12. C. Gosselin and J. Angeles, "The optimum kinematic design of a spherical three-degree-of-freedom parallel manipulator," *ASME Trans. J. Mech. Trans. Autom. Des.* **110** (2), 202–207 (1989).
13. C. Gosselin and J. Angeles, "A global performance index for the kinematic optimization of robotic manipulators," *ASME Trans. J. Mech. Des.* **113** (3), 220–226 (1991).
14. R. Kurtz and V. Hayward, "Multiple-goal kinematic optimization of a parallel spherical mechanism with actuator redundancy," *IEEE Trans. Robot. Autom.* **8**(5), 644–651 (1992).
15. O. Ma and J. Angeles, "Optimum Architecture Design of Platform Manipulator," *Proceedings of the IEEE International Conference on Robotics Automation* (June, 1991) pp. 1131–1135.
16. K. H. Pittens and R. P. Podhorodeski, "A family of Stewart platforms with optimal dexterity," *J. Rob. Syst.* **10**(4), 463–479 (1993).
17. H. R. Mohammadi Daniali, P. J. Zsombor-Murray and J. Angeles, "The isotropic designs of two general classes of planar parallel manipulators," *J. Robot. Syst.* **12**(12), 795–805 (1995).
18. J. Angeles, *Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms*, 2nd ed. (Springer-Verlag, New York, 2003).
19. J.-P. Merlet, "Designing a parallel manipulator for a specific workspace," *Int. J. Robot Res.* **16**(4), 545–556 (1997).
20. R. Boudreau and C. M. Gosselin, "The synthesis of planar parallel manipulators with a genetic algorithm," *J. Mech. Des.* **121**(4), 533–537 (1999).
21. M. A. Laribi, L. Romdhane and S. Zeghloul, "Analysis and dimensional synthesis of the DELTA robot for a prescribed workspace," *Mech. Mach. Theory* **42**, 859–870 (2007).
22. Y. LI and Q. XU, "Optimal Kinematic Design for a General 3-PRS Spatial Parallel Manipulator Based on Dexterity and Workspace," *The Eleventh International Conference on Machine Design and Production*, Antalya, Turkey (13–15 Oct. 2004).
23. Q. Xu and Y. Li, "Kinematic analysis and optimization of a new compliant parallel micromanipulator," *Int. J. Adv. Robot. Syst.* **3**(4), 351–358 (2006).
24. G. Pond and J. A. Carretero. "Architecture optimisation of three 3-PRS variants for parallel kinematic machining," *Robot. Comput.-Integr. Manuf.* **25**(1), 64–72 (2009).
25. P. Chedmail and E. Ramstein, "Robot mechanism synthesis and genetic algorithms," *Proc. IEEE Int. Conf. Robot. Autom.* **4**, 3466–3471 (1996).
26. T. P. Bagchi, "Multiobjective robust design by genetic algorithms," *Mater. Manuf. Process.* **18**(3), 341–354 (2003).
27. D. Zhang, Z. Xu, C. M. Mechefske and F. Xi, "Optimum design of parallel kinematic tool-heads with genetic algorithms," *Robotica* **22**, 77–84 (2004).
28. H. Zhuang, K. Wang and Z. S. Roth, "Optimal Selection of Measurement Configurations for Robot Calibration Using Simulated Annealing," *Proceedings of the IEEE International Conference on Robotics and Automation*, San Diego (1994) pp. 393–398.
29. J. J. Paredis and P. K. Khosla, "Kinematic design of serial link manipulators from task specifications," *Int. J. Robot. Res.* **12**(3), 274–287 (1993).
30. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, vol. IV, Piscataway, NJ (1995) pp. 1942–1948.
31. M. Clerc and J. Kennedy "The particle swarm-explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evolutionary Comput.* **6**(1), (2002) 58–73.
32. K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Trans. Evolutionary Comput.* **8**(3), (2004) 211–224.
33. J. Sun, B. Feng and W. B. Xu, "Particle Swarm Optimization with Particles Having Quantum Behavior," *IEEE Proceedings of Congress on Evolutionary Computation* (2004) pp. 325–331.
34. J. Sun, W. B. Xu and B. Feng, "A Global Search Strategy of Quantum-Behaved Particle Swarm Optimization," *Cybernetics and Intelligent Systems Proceedings of the 2004 IEEE Conference* (2004) pp. 111–116.
35. M. Xi, J. Sun and W. Xu, "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position," *Appl. Math. Comput.* **205**, 751–759 (2008).
36. G. H. Golub and C. F. Van Loan, *Matrix Computations* (The John Hopkins University Press, Baltimore, 1989).
37. J. K. Salisbury and J. J. Craig, "Articulated hands: force control and kinematic issues," *Int. J. Robot. Res.* **1**(1), 4–17 (1982).
38. F. Ranjbaran, J. Angeles, M. A. Gonzalez-Palacios and R. V. Patel, "The mechanical design of a seven-axes manipulator with kinematic isotropy," *J. Intell. Robot. Syst.* **14**, 21–41 (1995).
39. R. S. Stoughton and T. Arai, "A modified Stewart platform manipulator with improved dexterity," *IEEE Trans. Robot. Autom.* **9**(2), 166–173 (1993).

https://doi.org/10.1017/S0263574710000500 Published online by Cambridge University Press