Robotica (2010) volume 28, pp. 381–395. © Cambridge University Press 2009
doi:10.1017/S0263574709005700

Optimal design of manipulator parameter using evolutionary optimization techniques

B. K. Rout* and R. K. Mittal

Mechanical Engineering Group, Birla Institute of Technology and Science, Pilani – 333031, Rajasthan, India

(Received in Final Form: April 9, 2009. First published online: July 7, 2009)

SUMMARY
A robot must have high positioning accuracy and repeatability for precise applications. However, variations in performance are observed due to the effect of uncertainty in design and process parameters. So far, there has been no attempt to optimize the design parameters of manipulator by which performance variations will be minimum. A modification in differential evolution optimization technique is proposed to incorporate the effect of noises in the optimization process and obtain the optimal design of manipulator, which is insensitive to noises. This approach has been illustrated by selecting optimal parameter of 2-DOF RR planar manipulator and 4-DOF SCARA manipulator. The performance of proposed approach has been compared with genetic algorithm with similar modifications. It is observed that the optimal results are obtained with lesser computations in case of differential evolution technique. This approach is a viable alternative for costly prototype testing, where only kinematic and dynamic models of manipulator are dealt with.

KEYWORDS: Manipulator performance variations; Noise factors; Positional error; Orientation error; Mean positional error: Mean orientation error; Orthogonal array; Evolutionary optimization technique.

1. Introduction

There are many industrial applications where manipulator is required to carry out precise task with high accuracy and repeatability. However, industrial manipulators fail to deliver the desired performance because the accuracy and repeatability of the manipulator is affected by computational error, machining tolerance, joint clearances and misalignment, flexibility effect of links, gear backlash, and host of other static and dynamic effects. These effects are called as noise factors, which are difficult to model and costly to control. Hence, design of industrial manipulator to satisfy desired performance requirement is a complex task. In last decade and a half, the philosophy of developing stable products and processes that exhibit minimum sensitivity to uncontrollable noises has been prevalent in research community. However, robot manufacturer adopt traditional methods of experimenting with prototypes that are time consuming and expensive to manufacture. In order to assuage these difficulties kinematic and dynamic models of the manipulator and an evolutionary optimization approach are used to obtain the optimal parameters, which are insensitive to noises. The kinematic and dynamic models of manipulators are nonlinear and coupled. Thus, explicit modeling of noises will make dynamic model complex. A modification in evolutionary technique has been proposed. This modification is a worst-case approach to incorporate the effect of noises and simulate the performance. From this technique optimal kinematic and dynamic parameters of manipulator, i.e. link lengths and link mass, are obtained. These parameters are expected to be insensitive to noises and deliver minimum performance variations.

1.1. Background of research

Lot of research has been conducted to address kinematic error analysis of robot end-effectors, calibration of robot manipulators, and parameter optimization of robots for various performance criteria. Khatib and Burdick[1] investigated the dynamic characteristics of manipulators and developed a method for the dynamic optimization. The dynamic optimization aimed at providing the largest isotropic and uniform bounds on the magnitude of end-effector acceleration at both low and high velocities. Manoochehri and Seireg[2] developed computer programme for form synthesis and optimal design of robot manipulator using dynamic programming approach while Shiller and Sundar[3] addressed designing of multi-degree of freedom (DOF) systems for optimal dynamic performance based on the acceleration lines. Khatib and Bowling[4] investigated the problem of manipulator design for increased dynamic performance and used optimization techniques to determine the design parameters, which improve manipulator performance. Stocco *et al.*[6] proposed a new global isotropy index (GII) to quantify the configuration-independent isotropy of a robot's Jacobian or mass matrix and presented a new discrete global optimization algorithm to optimize either the GII or some local measure without placing any condition on the objective function. Carretero *et al.*[7] undertook architecture optimization of a 3-DOF parallel mechanism and demonstrated that specific values of design variables allow minimization of parasitic motion, i.e. motions in the three unspecified motion coordinates while Rao and Bhatti[8] proposed a probabilistic approach to simulate manipulator kinematic and dynamic performance in terms of reliabilities.

Genetic algorithms (GAs) and its variants have been extensively used in many fields such as controls, parameter

* Corresponding author. E-mail: bijoykumarraut@yahoo.com

identifications, modular robot design, planning, scheduling, and image processing. However, application of evolutionary techniques to determine optimal parameters of manipulator for minimum performance variations is rare. Chocron and Bidaud[5] proposed a method for task-based design of modular robotic systems using GA and introduced a 3D kinematic description for modular serial manipulators and a two-level GA to optimize their topology from task specifications. Shiakolas *et al.*[9] applied three evolutionary techniques to optimize the required torque of robots for a defined motion subject to different constraints. Tian and Collins[10] studied the base placement problem for a 2-DOF robot and described the feasible area for the robot base, and formulated manipulability measure. Kim[11] presented the method to determine the kinematic parameters of 2-DOF manipulator with a parallelogram five-bar link mechanism from a given task, i.e. how to map a given task into the kinematic parameters. Tabandeh *et al.*[12] presented a GA for solving the inverse kinematics of a serial robotic manipulator. The algorithm was capable of finding multiple solutions of the inverse kinematics through niching methods. Huang *et al.*[13] presented a minimum-torque path planning scheme for space manipulator and used GA to minimize the objective function. He *et al.*[14] presented mathematical methods for the placement of serial robot manipulators with respect to pre-defined target points in arc welding applications where robot is mounted on a crossbeam and proposed adaptive genetic algorithm to dynamically modify the parameters of GA in terms of simulated annealing mechanism. Dolinsky *et al.*[15] introduced a new inverse static kinematic calibration technique based on genetic programming, which is used to establish and identify model structure and parameters. The technique identified the true calibration model avoiding the problems of conventional methods. Coello Coello *et al.*[16] developed a technique that combines GA and the weighted min–max multi-objective optimization method for robot design. Han *et al.*[17] developed a modular manipulator and the method of task-based design. They proposed two-step design algorithm which determines robot configuration using kinematic relations and determines link lengths using the proposed efficient GA. Later, Rout *et al.*[18] optimized the kinematic and dynamic parameters of a manipulator using evolutionary optimization techniques for optimal energy usage and subject to different physical constraints. Rout and Mittal[19] discussed a combined array design of experiment approach to screen the statistically significant parameters of manipulator.

In the direction of modular manipulator design, Khosla and his coworkers at Carngie Mellon University and Chen and his coworkers at Nanyang Technical University have made significant progress. Paredis and Khosla opined that the serial manipulators are not general-purpose manipulators. They addressed the problem of mapping kinematic task specifications into a kinematic manipulator configuration. In this regard, an analytical solution is proposed which determines the Denavit–Hartenberg (DH) parameters of a nonredundant manipulator with joint limits that can reach a set of specified positions/orientations in an environment that may include parallelepiped-shaped obstacles. Paredis and Khosla[20] dealt with two important issues in relation to modular reconfigurable manipulators, namely, the determination of the modular assembly configuration optimally suited to perform a specific task and the synthesis of fault tolerant systems. Paredis *et al.* and Khosla[21] developed a Reconfigurable Modular Manipulator System (RMMS). These modules are assembled in a large number of different configurations to tailor the kinematic and dynamic properties of the manipulator to the task at hand. The control software for the RMMS automatically adapts to the assembly configuration by building kinematic and dynamic models of the manipulator, which is very transparent to the user. Paredis[22,23] developed an agent-based implementation that runs on a distributed network of workstations where its aim was to speed up the search by increasing the computational resources. They implemented a modified GA to run in parallel according to a master slave message-passing model. The most important contribution was the development of the agent-based design framework for Task-Based Design (TBD). In TBD all the components of a rapidly deployable fault tolerant manipulator system are tied together. At later stage, Yang and Chen[24] introduced the optimization of modular reconfigurable robot configurations for specific task requirements. The Minimized Degree-of-Freedom problem is formulated as a design optimization problem. Several task-related kinematic performance measures are considered as the design constraints. Based on the problem-specific coding schemes, the evolutionary algorithm (EA) approach is employed to search the optimal solutions. Chen[25] suggested that the design of a task-oriented robot configuration becomes a discrete design optimization problem. A task-performance-related objective function is formulated and employed software agent concept to GA to determine the optimal configuration. Later Chen[26] described the development of a component-based technology robot workcell that can be rapidly configured to perform a specific manufacturing task.

Differential evolution (DE) Algorithm is a new evolutionary approach, proposed by Storn[27,28] to minimize nonlinear and nondifferentiable continuous space functions. Price and Storn[29,30] presented a generalized algorithm on DE to optimize variety of problems. Similar to GA, it has been applied to various fields successfully. Recently Hacker and Lewis[31] discussed evolutionary-based techniques for parameter design optimization. They optimized the systems with multiple local optima and one or more uncertain design parameters and presented an approach that utilizes both local and global optimization algorithms to find good design points more efficiently than either could alone. Bagchi[32] proposed multi-objective GA to uncover the dependency among the key design factors and obtain robust performance.

From above discussion, it is evident that the techniques to highlight designers' and manufactures' perspective in manipulator design optimization process are rare. Therefore, an evolutionary optimization approach is proposed. The aim of this approach is to optimize the parameters in such a way that it would be insensitive to the effects of noises. To implement above philosophy, a modification in existing evolutionary optimization approach is proposed. In this approach, the evolutionary optimization technique is combined with orthogonal array (OA) of the Taguchi method to optimize the design parameters of manipulator. The OA is a fractional factorial design used in design of experiments technique and assures a balanced comparison of levels of any factor or interaction of factors.[34] It uses all the genetic operators of DE with modification in 'cost function' evaluation process. The cost function of each population member is evaluated after incorporating the effect of noises. To incorporate the effect of noises in the population member, designated OA is embedded in evolutionary technique. The designs created by OA experiments are used to simulate the performances. The simulated performances of designs are consolidated to a "performance measure." This "performance measure" becomes the "cost function" of the population member. Likewise, cost functions for other population members are obtained. Subsequently, the population members are allowed to participate in the evolution process. In this process, improved designs are achieved in successive generations, from a pool of candidate designs. This approach has been illustrated by optimizing the parameters of 2-DOF RR planar manipulator and 4-DOF SCARA manipulator while carrying out a specified task, for optimum performance.

The remainder of this paper is organized as follows. In Section 2, evolutionary optimization approach and proposed modification is discussed. The worst-case approach to simulate the performance of manipulator is discussed in Section 3. Formulation for optimal design parameter of manipulator is discussed in Section 4. Proposed approach is illustrated in Section 5. Finally, the results are presented and compared in Section 6.

2. Optimization Approaches

Many traditional search and optimization techniques are available but these cannot be used to optimize the parameters of manipulator for optimal performance. The major disadvantages of conventional optimization techniques are that they work using local information to decide which point to explore next, ready-made objective function, and good initial guess. This leads to the solution being trapped at local optimum, which depends on the degree of nonlinearity. However, population-based algorithms are found to have a better global perspective than the conventional methods.[40] Recently, Onwubolu and Babu[40] compiled new techniques and their applications to various disciplines of engineering and management.

The evolutionary techniques available do not have features to handle the effect of uncertainty on decision variables, which cause variations in objective function. Hence, a simple modification in existing evolutionary technique is proposed. This modification has been implemented in DE evolutionary optimization approach. To simplify the explanation, the working principle of DE technique is discussed first in Section 2.1 and proposed modification is discussed in Section 2.2. To present the computational superiority of the proposed approach, its performance is compared with GA. DE and GA are population based search and optimization methods and hence the comparison between them is justified. By this approach the optimal parameters of a product or process can be obtained, which is insensitive to noises and deliver the optimal performance.

2.1. *Differential evolution technique*

DE is increasingly applied to various search and optimization problems in the recent past. The advantage of DE is its simple structure, ease of use, speed, and robustness.[17,18] DE is a parallel search method that operates on D-dimensional parameter vector. The number of vectors is equal to user-defined population size. The initial population vectors are chosen randomly. A cost function C is used to rate the individual vector according to their capability to minimize the objective function. The evolution process starts with selection of a target vector. Then, it randomly selects two other vectors and generates a difference vector, which is multiplied with a user-defined weighting factor F to obtain "weighted difference vector." The weighted difference vector and randomly chosen mutation vector are added to create a noisy vector, which is subjected to crossover process with the target vector to generate the trial vector. The cost function of trial vector is then compared with the cost function of original target vector. The vector having low cost function is allowed into the new population. Same procedure is adopted for the entire population member to get a next generation. This process is continued until a termination criterion is satisfied, i.e. desired number of generations/constraints. The detailed algorithm for DE technique is discussed as follows:

2.1.1. Algorithm for differential evolution optimization technique. A cost function C is used to rate the individual vector according to their capability to minimize/maximize the objective function f. The key parameters of control are NP: the population size, CR: the crossover probability, F: the weight factor (scaling factor), and D: the number of design variables.

Step 1. Initialize the value of D, NP, CR, F, and number of generation.
Step 2. Initialize all the vector population randomly for given upper and lower limit.
Step 3. Evaluate the cost C of each vector.
Step 4. Perform mutation, crossover, selection, and evaluation of the objective function for a specified number of generations.

(a) For each vector X_t (target vector), select three distinct vectors X_a, X_b, and X_c randomly from a current population other than vector X_t.
(b) Generate difference vector $X_d = (X_a - X_b)$.
(c) Multiply weighted factor F to difference vector to obtain "weighted difference vector," i.e. $F \times X_d$.
(d) Perform mutation by adding weighted difference vectors to the third vector X_c to get noisy vector $X_n = F \times (X_a - X_b) + X_c$.
(e) Perform crossover with probability CR for each target vector X_t with noisy vector X_n to create a trial vector.
(f) Evaluate the cost C, if the trial vector is X_n else use saved value of X_t.
(g) Perform selection for each target vector X_t by comparing its cost function with that of the trial vector. For minimization/maximization problem, vector with lower/higher cost function is selected for next generation.

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

(h) Select the next target vector of the population; repeat the steps 4(a)–4(g) NP times to obtain next generation.

Step 5. Check for the termination criterion, stop, if it is satisfied else, go to step 4, i.e. repeat the same procedure for next generation.

To evaluate the computational effort spent in reaching the optimal solution, a parameter called "number of function evaluation" (NOF) is considered for a specified number of generations. The NOF is determined by subjecting the trial vector to the bound and constraint check. If the vector succeeds, then it is sent for cost function evaluation and NOF is incremented. The vector is rejected, if it fails.

2.2. *Proposed differential evolution approach*

From above discussion, it is quite clear that the population member is evaluated once for cost function and suitable for deterministic optimization problems. This paper focuses on reduction of performance variations of a product caused by uncertainty in design and process parameters and selection of optimal design parameters. A modification in existing approach is proposed and its implementation strategy is discussed later.

To incorporate the effect of noises in performance, probabilistic approach could have been used where the effects of noise factors are assumed to follow particular probability density functions. However, this approach considers the effect of noise factors as uncertainties ranges ($\pm T_x/2$) in place of probability distributions, where T_x is the tolerance of design/process factor x. This deviation captures the worst-case scenario around the nominal dimension of each factor. Usually the tolerance of a factor is specified with the nominal dimension. To incorporate the effect of noises, i.e. tolerances, in design and process factors systematically OAs proposed by Taguchi[34] are selected and used. Selection of a particular type of OA is dependent on the number of design and process parameters for which the effect of noises is required to be incorporated. Thus, any set of deviations of design and process factors representing tolerances would result in number of designs. The number of designs is equal to the number of points present in an OA (hyper rectangular design space), in place of single design. The major advantage in use of OA is that it helps the evolutionary technique to consider a fraction of the space around the population member. This method is efficient as compared to other stochastic optimization methods, which requires large number of sample point to ensure high accuracy.

Orthogonal arrays are a special set of Latin squares, constructed by Taguchi to lay out the product design experiments. It is essentially a matrix of numbers (1s and 2s) arranged in rows and columns, where each row represents the level of the factors in each run, and each column represents a specific parameter that can be changed in each run.[23] The numbers (1s/2s) in the row indicate the parameter level. The columns of OA are orthogonal because the columns for the independent parameters are orthogonal to one another and parameter level occurs same number of times.

To explain the proposed approach, an arbitrary L_N (2^c) OA is selected and shown in Table I, where, N, 2, and c represent the number of experiments, the number of levels of each factor, and the number of columns in the array, respectively. The size of selected OA depends on the number of levels and factors for which the effect of noise needs to be incorporated. If a population member (vector) X with n independent design and process factors is represented by $X = [x_1, x_2, \ldots, x_n]$ and the noise for these factors are expressed with the help of tolerances, $\pm T_{x_1}/2$, $\pm T_{x_2}/2, \ldots,$ $\pm T_{x_n}/2$, then the design and process factors with the effect of noise are represented by several candidate designs. This set of N candidate designs is shown in Table II. The 1s or 2s of a row in OA provide the directions of deviation about the nominal dimension of design and process parameters. Therefore, the corner points of OA are at a distance of $+T_x/2$ or $-T_x/2$ from the nominal dimension.

Table I. A $L_N(2^c)$ orthogonal array.

Expt. No.	Factor x_1	Factor x_2	...	Factor x_n
1	1	1	.	1
2	2	1	.	2
.	.	.	.	.
.	.	.	.	.
N	1	2	.	2

Table II. Candidate design and performance after incorporating the effect of noise.

Expt. No.	Factor x_1	Factor x_2	...	Factor x_n	Performance (y_i)
1	$x_1 - T_{x_1}/2$	$x_2 - T_{x_2}/2$	.	$x_n + T_{x_n}/2$	y_1
2	$x_1 + T_{x_1}/2$	$x_2 - T_{x_2}/2$	.	$x_n + T_{x_n}/2$	y_2
.	.	.	.	.	.
.	.	.	.	.	.
N	$x_1 + T_{x_1}/2$	$x_2 + T_{x_2}/2$	.	$x_n + T_{x_n}/2$	y_N

By this strategy, each candidate design produces a series of performances, as opposed to single point design where every population member delivers a particular performance. Thus, each design has finite set of performance values. This evaluation procedure provides worst-case scenario for the determination of performance variations. The corner points of the OA are evaluated for the performance (y_i) and consolidated to a performance measure using the set of y_i s. The performance measure becomes the cost function of the population member and used in evolution process. Except above procedure, all other steps of DE algorithm remain same to obtain the next generation. In this way, the algorithm directs the solutions to the optimal region. Flow chart for the proposed modification is shown in Fig. 1.

3. Approach to Simulate the Performance of Manipulator

In manipulator's performance analysis, unknown variations in material distribution, uncertainty in manufacturing and assembly, friction, joint clearances, fluctuations in joint torques, and uncertainty in boundary condition cause variations in performance. As it is known that, the dynamic model is highly coupled and nonlinear, simulation of performance of a manipulator incorporating the effect of

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

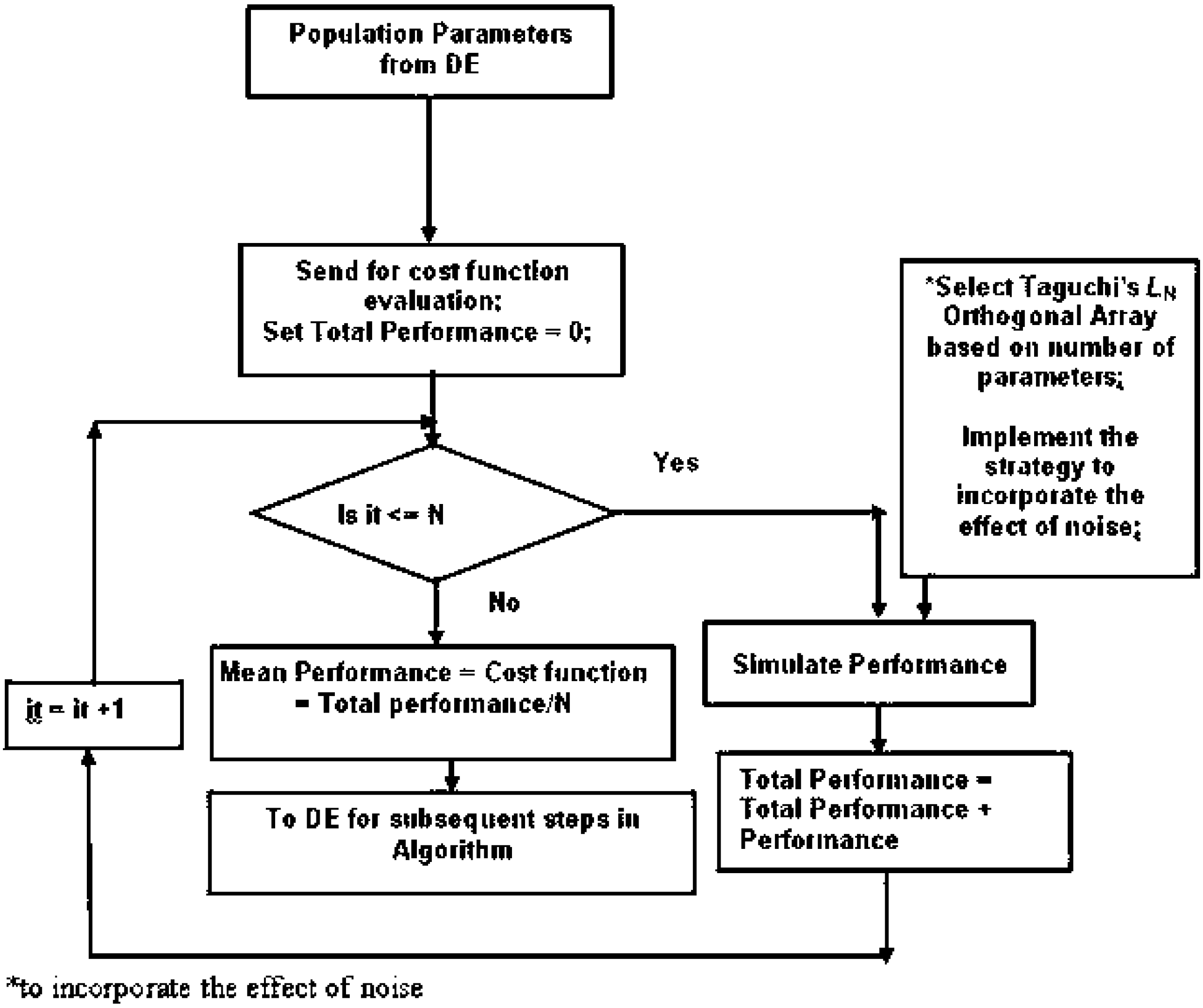

Fig. 1. Flow chart for the proposed evolutionary approach.

noise becomes difficult. Given the randomness of robot parameters, models for their probability distributions are needed. For the sake of conciseness, the variations of the parameter are assumed to obey Gaussian distributions with nonzero mean and nonidentical standard deviations. Therefore, the variations in length and mass of the links are assumed to follow the Gaussian distribution and standard deviations are dependent on the specified tolerance for link lengths and link masses.

Robot performing commanded task can be modeled as a stochastic process which is driven by actuators, and when the time interval between sample values of supplied torque is small, the process becomes highly correlated over time. As torque being supplied to manipulator joint is in time order and highly correlated, the supplied torques by the actuators are treated as random vectors. These vectors are assumed to follow Gaussian stochastic process with Markov properties,[7] in which the future values of a Markov process depend only on immediate past or present but not all past events. The strategy adopted to simulate the joint torque and its description is available in ref. [19]. To determine the torque required at the joints of the manipulator following approach is used.

3.1. Simulation of joint torque for a trajectory

The torque required at joint of manipulator is determined based on the type of trajectory chosen to perform the task. For the manipulator a cubic trajectory is considered to carry out the task. A smooth motion between initial and target points, the functions $q(t)$ and $\dot{q}(t)$ have to be smooth, where q_i, q_f are joint coordinates and $\dot{q}_i = 0$, $\dot{q}_f = 0$ are joint velocities, respectively, and time to reach the destination is T_g. The initial (i) and target (f) boundary conditions are

$$q(0) = q_i, \quad q(T_g) = q_f, \quad \dot{q}(0) = 0, \quad \dot{q}(T_g) = 0. \quad (1)$$

Let q_i and q_f be the initial and target point values, respectively. The time law of cubic trajectory is given[38] by

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3, \quad (2)$$

where, a_0, a_1, a_2, a_3 are constants. The constants for each of n joint variables are determined by applying boundary conditions to (2), the constants are obtained by

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & T_g & T_g^2 & T_g^3 \\ 0 & 1 & 2T_g & 3T_g^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} q_i \\ \dot{q}_i \\ q_f \\ \dot{q}_f \end{bmatrix}. \tag{3}$$

Using above information, joint torque for a manipulator is simulated. The fluctuation in supplied joint torque vector $\tau(t)$ is assumed to follow Gaussian stochastic process with Markov properties and simulated as a time series. The time series is a sequence of observation taken sequentially over time. These observations in a time series are regarded as a sample realization from an infinite population of such time series that could have been generated by the stochastic process. The stochastic model, which can be used for simulation of supplied joint torque vector, is the auto-regressive process model. In this model, the current values of the process are expressed as a finite, linear aggregate of previous values of the process and a shock a_t.

3.2. Computation of manipulator performance

To simulate the performance of a manipulator inverse dynamics and forward dynamics principles are used. Inverse dynamics approach is used to determine the torque vector $\tau(t)$ required at the manipulator joints. For the computation of the torque vector, numerical values of manipulator design parameters, Cartesian coordinates of start and destination point, time to reach, and type of trajectory to reach the destination are used. Then the effect of noise is incorporated in design parameters and torque vector using method discussed in Section 2.2. Using forward dynamics approach, the dynamic model is integrated numerically. From this numerical integration, the joint accelerations, velocities, and positions, i.e. $q(t), \dot{q}(t), \ddot{q}(t)$ are computed. Finally, joint coordinates $q(t)$ are transformed, using the kinematic model, to obtain the Cartesian coordinates and the orientation. The Cartesian coordinates and orientations are used for the computation of positional error (ε_i) and orientation error (ν_i) at the destination while moving along a trajectory. These results become the performance of manipulator with the effect of noises.

3.3. Manipulator performance

For the optimization of design parameters of manipulator, mean positional error ($\bar{\varepsilon}$) and mean orientation error ($\bar{\nu}$) are used as the performance measure. These measures are computed using ε_i and ν_i, where ε_i is the vector distance between the actual point (x_{ia}, y_{ia}, z_{ia}) reached by the end-effector and desired point (x_d, y_d, z_d) in ith experiment.[19] Similarly, ν_i is the orientation error between the actual orientation $(\phi_{ia}, \varphi_{ia}, \psi_{ia})$ reached by the end-effector and desired orientation $(\phi_d, \varphi_d, \psi_d)$ of the ith experiment and expressed by

$$\nu_i = \sqrt{(\phi_d - \phi_{ia})^2 + (\varphi_d - \varphi_{ia})^2 + (\psi_d - \psi_{ia})^2}. \tag{4}$$

Mean positional and orientation error: Mean positional error ($\bar{\varepsilon}$) is already defined in ref. [19] and mean orientation error ($\bar{\nu}$) is defined as

$$\bar{\nu} = \frac{1}{N}\sum_{i=1}^{N} \nu_i, \tag{5}$$

where N is the number of experiments conducted.

4. Formulation of Optimization Problem for Parameters of Manipulator

One of the major challenges faced by the robot designers is to select an optimal parameter, which delivers the desired performance in presence of various uncertainties. Present work discusses an approach to select optimal kinematic and dynamic parameters of the manipulator with effect of noises. The aim of the optimization problem is to find the design parameters of manipulator that minimize the mean positional error and mean orientation error at the destination while moving along a particular trajectory. The performance measures used for investigation are already discussed and design parameter of manipulator for optimization process is given by $X = [x_1, x_2, \ldots, x_i, \ldots, x_n]^T$. The mathematical form of the optimization problem is stated as

$$\text{Find } X = [x_1, x_2, \ldots, x_i, \ldots, x_n]^T$$

$$\text{to Minimize } f(x) = \bar{\varepsilon} + \bar{\nu} \tag{6}$$

$$\text{subject to } x_{il} \le x_i \le x_{iu}, \tag{7}$$

where x_{il} and x_{iu} are the lower and the upper bounds of the design parameters i. The value of i change from 1 to n. The constraints on the joint limits or range of motion of the manipulator are imposed and defined as

$$\theta_{j,\min} \le \theta_j \le \theta_{j,\max}, \tag{8}$$

where θ_j is the joint variable for joint j. It is assumed that the required joint torque to perform the specified task is available without any restriction.

4.1. Procedure for the optimization of design parameters of manipulator

The objective function of problem, design parameters, and constraints are identified. For the optimization, desired start and destination points in Cartesian space and desired orientation, time for the motion, and type of trajectory are assumed. The effect of noises for the design and process parameters are expressed in terms of tolerances, and control parameter bounds are chosen.

The design parameters are randomly generated in DE routine and values are checked for any constraint violation. These parameters are sent for cost function evaluation, if they do not violate the bounds. The cost function is obtained using strategy discussed in Section 2.2. One function evaluation is completed when one set of design variables is analyzed.

The OA suitable for the parameter optimization problem is selected. It helps in incorporating the effect of noises in design and process parameters to generate several design

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

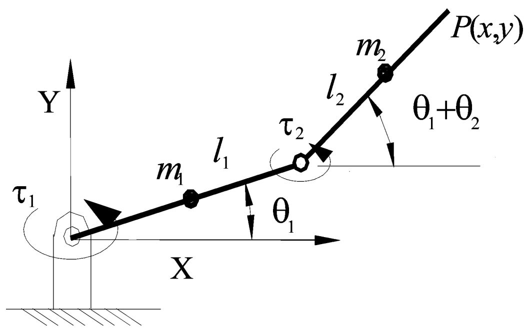

Fig. 2. A 2-DOF RR planar manipulator and its parameters.

combinations. These design combinations are utilized to simulate the performance, i.e. positional error and orientation error, at the destination point. To simulate the performance method discussed in Section 3 is used. In this way performance of one design combination, i.e. one corner point of OA, is evaluated. Likewise, other corner points of OA are evaluated for performances. These performances are consolidated to the defined performance measure, i.e. mean positional error and mean orientation error. These measures become the cost function for one population member. Subsequently, all population members are evaluated for cost function. Then the entire population members participate in evolution process to reach the optimum value. This process is repeated until the termination criterion is satisfied.

5. Optimization of Design Parameters of 2-DOF RR Planar Manipulator and 4-DOF SCARA Manipulator

To illustrate the methodology, 2-DOF RR planar manipulator and 4-DOF SCARA manipulator (shown in Figs. 2 and 3) are considered. The purpose behind taking second example is to show the robustness of the proposed method. As their kinematic and dynamic models are used in optimization process, brief descriptions of models are presented in different sections. The manipulator models are based on the DH parameters and the parameters are expressed in the SI units and the angles are in degrees.

5.1. Kinematic and dynamic models of 2-DOF RR planar manipulator

The kinematic model in terms of homogenous transformation matrix ${}^0T_2$ is available in ref. [19]. Based on Lagrange–Euler formulation the dynamic behavior of joint i of the manipulator with contributions of viscous friction is given by[26]

$$\tau_i = \sum_j M_{ij}\ddot{q}_j + \sum_j \sum_k h_{ijk}\dot{q}_j\dot{q}_k + B_i\dot{q} + G_i, \quad (9)$$

where M_{ij} is the symmetric inertia matrix, h_{ijk} are the centrifugal and Coriollis force coefficients, G_i is the gravity force vector, τ_i is the joint torque vector, B_i is the viscous coefficient of friction. Therefore joint torques $\tau_1(t)$ and $\tau_2(t)$ of manipulator in generalized form are

$$\tau_1(t) = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + H_1 + B_1\dot{\theta}_1 + G_1 \quad (10)$$

$$\tau_2(t) = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + H_2 + B_2\dot{\theta}_2 + G_2 \quad (11)$$

where $M_{11} = \left(\frac{1}{3}m_1 l_1^2 + m_2 l_1^2 + \frac{1}{3}m_2 l_2^2 + m_2 l_1 l_2 C_2\right)$

$$M_{12} = \left(\frac{1}{3}m_2 l_2^2 + \frac{1}{2}m_2 l_1 l_2 C_2\right) = M_{21}, \quad M_{22} = \left(\frac{m_2}{3}l_2^2\right)$$

$$H_1 = -m_2 l_1 l_2 S_2 \dot{\theta}_1\dot{\theta}_2 - \frac{m_2}{2} l_1 l_2 S_2 \dot{\theta}_2^2, \quad H_2 = \frac{m_2}{2} l_1 l_2 S_2 \dot{\theta}_1^2$$

$$G_1 = \left(\frac{m_1}{2} + m_2\right) g l_1 C_1 + \frac{m_2}{2} g l_2 C_{12}, \quad G_2 = \frac{m_2}{2} g l_2 C_{12}$$

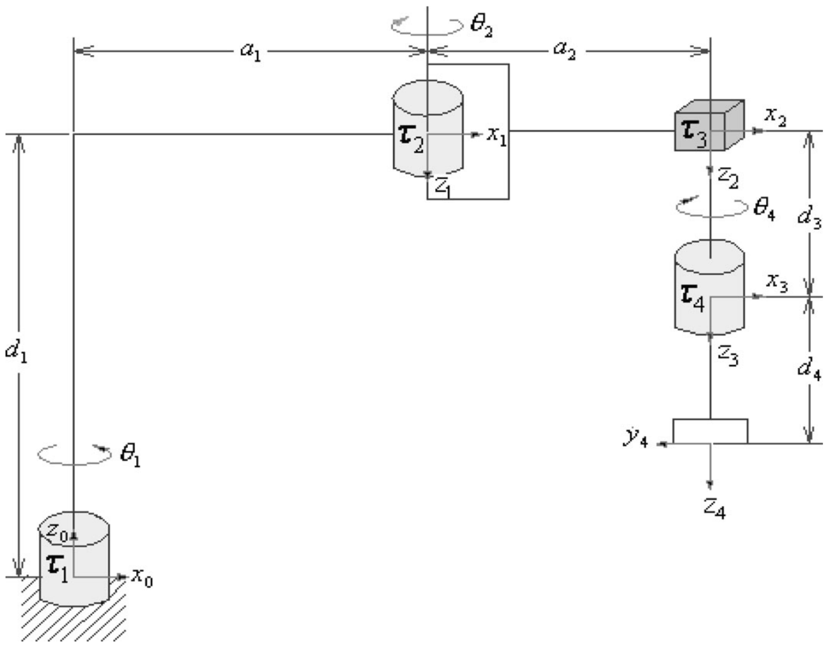

Fig. 3. Link and joint coordinates of SCARA manipulator.

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

and $\dot{\theta}_1, \dot{\theta}_2, \ddot{\theta}_1, \ddot{\theta}_2$ are velocities, accelerations of joints 1 and 2, respectively. In addition, g is the acceleration due to gravity in negative y-axis direction. It can be observed that there are four parameters, i.e. l_1 and l_2, m_1 and m_2 for which optimal parameters are desired. The joint coordinate is computed in terms of link parameters and violations are checked. Using inverse kinematics joint variables θ_1 and θ_2 are given by

$$\theta_1 = \tan^{-1}\left[\frac{y(l_1 + l_2C_2) - xl_2S_2}{x(l_1 + l_2C_2) + yl_2S_2}\right]. \tag{12}$$

Assuming, $D = (x^2 + y^2 - l_1^2 - l_2^2)/2l_1l_2$, joint variable θ_2 is obtained as

$$\theta_2 = \tan^{-1}\left[\frac{\sqrt{1 - D^2}}{D}\right]. \tag{13}$$

The objective function of the problem is specified in (6). The constraints of the optimization problem remain same as specified in (7) and (8) and no constraints are posed on the torque required at the joints to perform the task. This manipulator has limited orientation capabilities, hence orientation error is not considered. Then, the objective is to minimize the end-effector's mean positional error ($\bar{\varepsilon}$). To compute $\bar{\varepsilon}$, positional error (ε_i) is defined as

$$\varepsilon_i = \sqrt{(x_\mathrm{d} - x_{i\mathrm{a}})^2 + (y_\mathrm{d} - y_{i\mathrm{a}})^2}, \tag{14}$$

where $(x_\mathrm{d}, y_\mathrm{d})$ are the coordinates of the desired point and $(x_{i\mathrm{a}}, y_{i\mathrm{a}})$ the coordinates of actual point.

5.2. *Kinematic and dynamic models of 4-DOF SCARA manipulator*

To obtain optimal design parameters of a 4-DOF SCARA manipulator kinematic and dynamic models are derived. The following relationships are formulated for the SCARA to specify the corresponding cost function. Referring to Fig. 3, the joint variables q_i are constrained to lie in the joint space work envelope; where $\beta \geq 0$ is the physical constraint imposed on joint 2, and h and H are the joint constraints on the prismatic joint 3.[36]

$$\begin{bmatrix} -\pi \\ -\pi + \beta \\ h \\ -\pi \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} \leq \begin{bmatrix} \pi \\ -\pi - \beta \\ H \\ \pi \end{bmatrix}. \tag{15}$$

Therefore, the locus of points p reachable by the tool tip satisfies the following inequalities:

$$a_1^2 + a_2^2 - 2a_1a_2\cos\beta \leq p_1^2 + p_2^2 \leq (a_1 + a_2)^2 \tag{16}$$

$$d_1 - d_4 - H \leq p_3 \leq d_1 - d_4 - h \tag{17}$$

where p_1, p_2, p_3 are the x, y, and z co-ordinates of the tool tip and a_1, a_2, d_1, and d_4 are the link kinematic parameters.

$$a_{i\mathrm{l}} \leq a_i \leq a_{i\mathrm{u}} \tag{18}$$

$$d_{i\mathrm{l}} \leq d_i \leq d_{i\mathrm{u}} \tag{19}$$

Table III. Kinematic parameters of 4-DOF SCARA manipulator.

Axis	θ	d	a	α
1	q_1	d_1	a_1	π
2	q_2	0	a_2	0
3	0	q_3	0	0
4	q_4	d_4	0	0

$$m_{i\mathrm{l}} \leq m_i \leq m_{i\mathrm{u}} \tag{20}$$

where $a_{i\mathrm{l}}, d_{i\mathrm{l}}\ m_{i\mathrm{l}}$ and $a_{i\mathrm{u}}, d_{i\mathrm{u}}\ m_{i\mathrm{u}}$ are the lower and the upper bounds of the length and mass of link i, respectively. The link and joint parameters according to DH algorithm are listed in the Table III.

The tool position with respect to base is given by

$${}^0T_4 = \begin{bmatrix} C_{1-2-4} & S_{1-2-4} & 0 & a_1C_1 + a_2C_{1-2} \\ S_{1-2-4} & -C_{1-2-4} & 0 & a_1S_1 + a_2S_{1-2} \\ 0 & 0 & -1 & d_1 - q_3 - d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

where the notation C_{1-2-4} denotes $Cos\,(q_1 - q_2 - q_4)$ and similarly, S_{1-2-4} denotes Sin $(q_1 - q_2 - q_4)$. The vector representation of dynamic model of SCARA manipulator is given by

$$\tau_1 = M_{11}\ddot{q}_1 + M_{12}\ddot{q}_2 + M_{14}\ddot{q}_4 + H_1 + B_1\dot{q}_1 \tag{22}$$

$$\tau_2 = M_{21}\ddot{q}_1 + M_{22}\ddot{q}_2 + M_{24}\ddot{q}_4 + H_2 + B_2\dot{q}_2 \tag{23}$$

$$\tau_3 = M_{33}\ddot{q}_3 + B_3\dot{q}_3 + G_3 \tag{24}$$

$$\tau_4 = M_{41}\ddot{q}_1 + M_{42}\ddot{q}_2 + M_{44}\ddot{q}_4 + B_4\dot{q}_4 \tag{25}$$

where M_{ij} are the elements of 4×4 inertia matrix, H_i are the Coriollis and centrifugal components, G_i are the elements of gravity matrix, and B_i are the viscous friction coefficient at the individual joint. Complete derivation of the dynamic model is avoided to maintain brevity.

5.3. *Forward dynamics and numerical integration*

The closed-form solutions for above models are difficult to obtain, the torque obtained from inverse dynamics is integrated to compute the joint coordinates, velocities, and accelerations with the effect of noises. In this paper, Euler numerical integration method[36] has been used, to obtain joint angular accelerations, velocities, and positions. The torque equation for the manipulator at the start point $t = 0$ is given by

$$\tau_0 = M(q)\ddot{q}_0 + h(q_0, \dot{q}_0) + G(q_0) + B(\dot{q}_0). \tag{26}$$

The joint acceleration can be computed as

$$\ddot{q}_0 = M^{-1}(q_0)\,[\tau_0 - h(q_0, \dot{q}_0) - G(q_0) - B(\dot{q}_0)] \tag{27}$$

and future positions and velocities are obtained by integrating Eq. (27) forward in time steps of size Δt. Iteratively the angular velocities and positions at the $(i + 1)$ th instance are

Table IV. Manipulator task specifications.

Case	Coordinates of start point (x_i mm, y_i mm)	Coordinates of destination point (x_f mm, y_f mm)	Time to travel (s)
(i)	(650, 0)	(400, 300)	2
(ii)	(650, 50)	(400, 300)	2
(iii)	(650, 100)	(400, 300)	2
(iv)	(650, 50)	(−400, 300)	2
(v)	(650, 100)	(−400, 300)	2
(vi)	(400, 300)	(650, 0)	2

obtained numerically using

$$\dot{q}_{i+1} = \dot{q}_i + \ddot{q}_i \Delta t \tag{28}$$

$$q_{i+1} = q_i + \dot{q}_i \Delta t + \frac{1}{2}\ddot{q}_{i+1}(\Delta t)^2 \tag{29}$$

For each iteration, Eqs. (27)–(29) are used to compute the angular acceleration, velocity, and position, respectively.

6. Simulation and Discussion

The MATLAB software is used as the programming tool and codes are written to implement the proposed evolutionary optimization approach.

6.1. Selection of optimal parameters of 2-DOF RR planar manipulator

To simulate the performance and optimize the parameters of 2-DOF RR planar manipulator following numerical values are assumed. The effect of task on optimal parameters is investigated using several cases and specifications are provided in Table IV.

6.1.1. Assumed process parameters. Following process parameters are assumed for the optimization:

(I) Weight parameter for first-order autoregressive process $(\phi_1) = 0.8$
(II) Time step for numerical integration $(\Delta t) = 0.001$s
(III) Values of viscous friction at joints, B_1 and B_2 are 3.5 and 2 Ns/m, respectively
(IV) Trajectories to perform the tasks are cubic time law
(V) Joint constraints are $0 \le \theta_1 \le \pi$ and $0 \le \theta_2 \le \pi$ radian

The parameter bounds and tolerance ranges for design and process parameters of manipulator are specified in Table V. These values are kept same for all the tasks.

The control parameters of the proposed DE technique have been selected by trial and error method. The parameters for which optimal results are obtained with less NOF are selected for the investigations. The values of parameters are NP = 30, CR = 0.5, and $F = 0.8$. It is decided to run the DE algorithm for 100 generations, by which stochastic nature of performance will continually improve. The manipulator has four parameters (lengths of the two links and its corresponding masses). Correspondingly, for these parameters there will be four noise factors. In addition

Table V. Upper and lower limits and tolerance of design and process parameters.

Parameters	Lower limit	Upper limit	Tolerance
Lengths of link1 l_1 (mm)	450	550	±0.3
Lengths of link2 l_2 (mm)	350	450	±0.3
Mass of link1 m_1 (kg)	6	10	±0.015
Mass of link2 m_2 (kg)	4	8	±0.015
Torque at joint 1 $\tau_1(t)$ (Nmm)	Variable	Variable	±50
Torque at joint 2 $\tau_2(t)$ (Nmm)	Variable	Variable	±50

Table VI. L_8 orthogonal array.

Experiment No.	Column number 1	2	3	4	5	6	7
1	1	1	1	1	1	1	1
2	1	1	1	2	2	2	2
3	1	2	2	1	1	2	2
4	1	2	2	2	2	1	1
5	2	1	2	1	2	1	2
6	2	1	2	2	1	2	1
7	2	2	1	1	2	2	1
8	2	2	1	2	1	1	2

Table VII. Assumed population member values sent by DE.

l_1 (mm)	l_2 (mm)	m_1 (kg)	m_2 (kg)
425	375	6	4

to above noise factors, fluctuation in joint torque is also considered as a noise factor. Hence, there are six noise parameters and two levels for each noise parameter, i.e. $(-T_x/2$ and $+T_x/2)$, will have 2^6 combinations. Simulation of performance for 32 combinations is impractical and computation expensive. Thus, the L_8 orthogonal array proposed by Taguchi[34] has been selected, to reduce the number of experiments and amount of computation. The rational behind this selection is the number of parameters for which the effect of noises needs to be incorporated to simulate the performance. The L_8 OA shown in Table VI is a 2^{7-3} fraction factorial design to study maximum of seven parameters. This OA has been useful in studying the effect of noises on performance due to six parameters. From seven columns of OA, first six columns are allotted to six design and process parameters, and last column is left unassigned. The noises are deviations of the tolerance value from nominal parameter value. These represent worst-case tolerance deviations and satisfy the 3-sigma limits of normal Monte Carlo variability. Each row of OA is a noise combination that is treated as repetitive data for each member of population.

The assumed noises of six design and process parameters are specified in Table IV. To explain the use of OA in DE approach, an example has been considered. The detailed use of OA in DE has been provided in Tables VII and VIII, respectively. A sample design parameter (population

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

Table VIII. Combinations created from the population member values using OA.

Expt. No.	Column number parameter					
	1 l_1 (mm)	2 l_2 (mm)	3 m_1 (kg)	4 m_2 (kg)	5 $\tau_1(t)$ (Nmm)	6 $\tau_2(t)$ (Nmm)
1	424.7	374.7	5.985	3.985	−50	−50
2	424.7	374.7	5.985	4.015	+50	+50
3	424.7	375.3	6.015	3.985	−50	+50
4	424.7	375.3	6.015	4.015	+50	−50
5	4253	374.7	6.015	4.985	+50	−50
6	425.3	374.7	6.015	4.015	−50	+50
7	4253	375.3	5.985	3.985	+50	+50
8	425.3	375.3	5.985	4.015	−50	−50

Table IX. Optimal parameters from modified DE technique.

Case	Link 1 length l_1 (mm)	Link 2 length l_2 (mm)	Link 1 mass m_1 (kg)	Link 2 mass m_2 (kg)	Mean positional error $\bar{\varepsilon}$ (mm)	Number of function evaluations
(i)	500.6007	449.9971	9.977832	7.999904	2.85089	1790
(ii)	519.4249	449.98374	9.986784	7.999962	2.84476	1994
(iii)	547.97659	449.99262	9.997117	7.999872	2.83117	2011
(iv)	450.00103	350.06106	6.007054	7.99826	2.35964	1975
(v)	450.00299	350.09026	6.003299	7.994213	2.41807	2193
(vi)	450.2831	449.97079	9.929354	7.99912	2.62124	1959

member) value used for optimization is specified in Table VII. In Table VIII, all the eight possible combinations are generated for the four parameters l_1, l_2, m_1, and m_2. The performance is simulated for each combination and mean positional error is computed. Computed mean positional error becomes the cost function of the population member and participates in evolution process. The optimal results obtained from this approach have been presented in Table IX. In these tables, the optimal values of the design parameters, the cost function value, and NOFs are presented. It is observed that the approach consistently obtains smaller function value with less function evaluations. The NOFs is an indication of the computation effort spent in reaching the optimum function value for the same number of generations. In case of DE after the trail vector is computed each member of the trail vector is subjected to bound check and only those members who pass the bound check will have cost function evaluation. Thus, the function evaluations of the members, which fail the bound check, have been avoided and this results in reduction of the number of function evaluations. The objective function history for tasks following cubic trajectory are shown in Fig. 4.

Simulated performance measure of the manipulator is the indicator of worst-case performance. Hence, the result after 100 generations point to a region where the manipulator will not deliver performance worse than what has been achieved. It is observed that the worst-case performance is in a range of 2.85–2.35 mm for the tasks (i)–(vi).

6.1.2. Optimal parameters of manipulator using modified genetic algorithm. To compare the performance of above approach same modification is implemented in GA. In this case, evaluation of fitness function is carried out in discussed manner. Rest of the steps remain same as discussed in Deb.[39] The control parameters are selected based on trial and error method for which it provides optimal solution with less computation and time. The value of control parameters are chosen as: NP = 30, Cross over probability = 0.8 and mutation probability = 0.175. Using same parameter bounds, noises, tasks, and numerical values, another computer program is developed and simulations are run. To compare the performance of GA, simulation is run for 100 generations. The results of this optimization process are presented in Table X and function history of GA is presented in Fig. 5 for manipulator performing different tasks.

In Fig. 5, the objective function value decreases with slight fluctuation at the start and the end. In all these cases, objective function value reduces monotonically. After 100 generations, the mean positional error value ranges from 3.236 to 2.525mm for the tasks (i)–(vi). Finally, from Fig. 4, monotonic decrease in objective function value is observed but this is not the case for GA. It is observed that the objective function value of DE converged by 80th generations whereas it converged slowly for GA. In case of DE, NOFs required for convergence are less whereas GA required fixed 3030 NOFs to obtain the optimal solution. It lead to reduction in computation time. Computational time saving is one of the important outcomes of the above investigation. This result became evident when optimal parameters of 4-DOF SCARA manipulator are found out.

6.2. Selection of optimal parameters of 4-DOF SCARA manipulator

To investigate the effectiveness of the proposed method, desired modifications are made in the computer program to select the optimal parameters of SCARA manipulator. Performance of the proposed DE method is compared

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

Table X. Optimal parameters from modified GA.

Case	Link 1 length l_1 (mm)	Link 2 length l_2 (mm)	Link 1 mass m_1 (kg)	Link 2 mass m_2 (kg)	Mean positional error $\bar{\varepsilon}$ (mm)	Number of function evaluations
(i)	500.90856	448.94164	7.706868	7.627576	2.98985	3030
(ii)	474.13563	447.41994	9.953008	7.720614	2.99325	3030
(iii)	543.46479	431.60096	6.322149	7.602735	3.09823	3030
(iv)	452.34867	360.78641	7.03435	7.697754	2.49708	3030
(v)	531.86493	430.93319	9.415876	7.70108	3.03738	3030
(vi)	529.85308	429.91547	7.672331	4.650942	2.73784	3030

with the performance of GA. The effects of task on optimal parameters are investigated using several cases and its specifications are provided in Table XI. In this table W_0, W_1 are tool configuration at the start and destination, respectively. To simulate the performance, following numerical values are assumed and provided in Table XII. The constraint values for the design variables, the link lengths, and mass parameters are defined to be the same for both the optimization approaches. The tolerance range for design and process parameters are assumed and

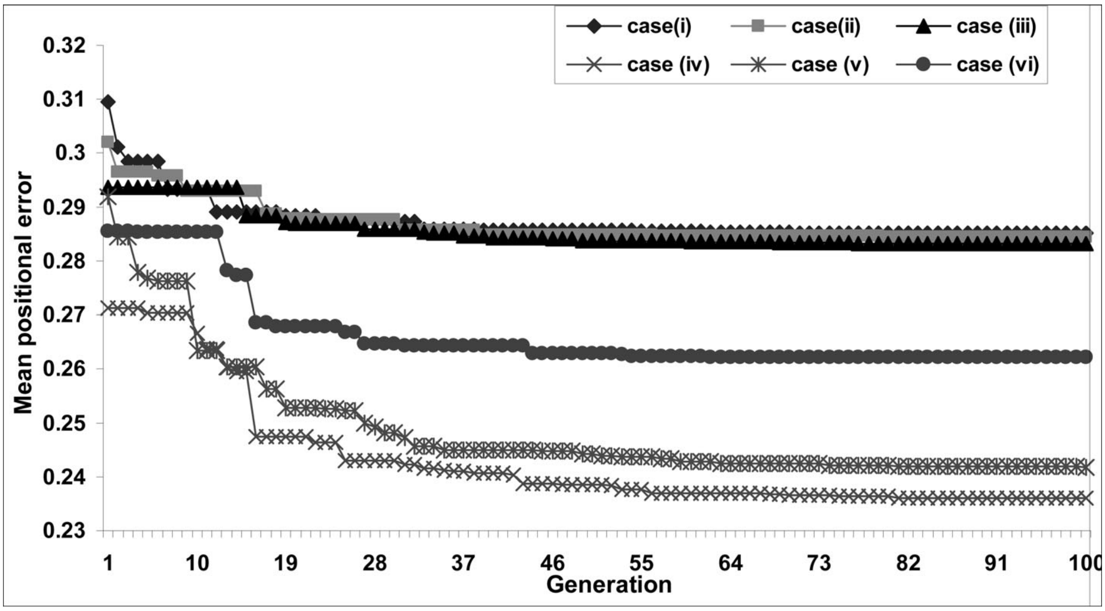

Fig. 4. Function history for modified DE technique.

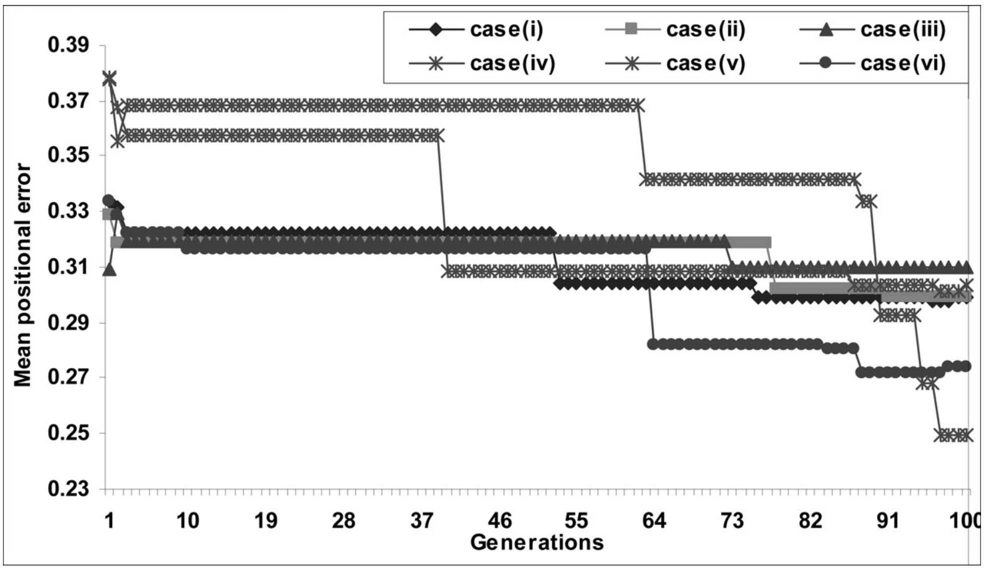

Fig. 5. Function history with modified GA.

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

Table XI. Task specifications for 4-DOF SCARA manipulator.

Task	Tool configuration at the start	Tool configuration at the destination	Motion time (s)
Task 1	$W_0 = \begin{bmatrix} 600 \\ 0 \\ 580 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$W_1 = \begin{bmatrix} 550 \\ 325 \\ 550 \\ 0 \\ 0 \\ 1.3243 \end{bmatrix}$	2
Task 2	$W_0 = \begin{bmatrix} 700 \\ 0 \\ 600 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$W_1 = \begin{bmatrix} 550 \\ 325 \\ 560 \\ 0 \\ 0 \\ 1.3243 \end{bmatrix}$	2
Task 3	$W_0 = \begin{bmatrix} 560 \\ 555 \\ 600 \\ 0 \\ 0 \\ 1 \end{bmatrix}$	$W_1 = \begin{bmatrix} 520 \\ 325 \\ 590 \\ 0 \\ 0 \\ 1.3243 \end{bmatrix}$	2

Table XII. Constraints for optimization in DE and GA.

Link length parameter (mm)	Link mass parameter (kg)	Joint ranges
$420 \le a_1 \le 430$	$8 \le m_1 \le 22$	$-\pi + 15 \le q_2 \le \pi - 15$
$370 \le a_2 \le 380$	$13 \le m_2 \le 17$	$15 \le d_3 \le 180$
$872 \le d_1 \le 882$	$8 \le m_3 \le 12$	
$195 \le d_4 \le 205$	$3 \le m_4 \le 7$	

Table XIII. Tolerance for design and process parameters.

Tolerance of link masses m_1, m_2, m_3 and m_4	Tolerance of link lengths and joint offsets a_1, a_2, d_1 and d_4	Tolerance of supplied torque $\tau_1(t)$, $\tau_2(t)$, $\tau_3(t)$ and $\tau_4(t)$
± 0.015 kg	± 0.3 mm	± 50 Nmm

given in Table XIII. Control parameters for the optimization process are selected by trial and error method for which NOF is lowest. For DE algorithm the control parameters are Number of generations $= 100$, NP $= 30$, CR $= 0.5$, and $F = 0.8$. The control parameters for GA are: Population size $= 40$, Number of generations $= 100$, Crossover probability $= 0.8$, and Mutation probability $= 0.175$. These values are used in the respective optimization processes and simulations are run.

There are 12 design and process parameters for SCARA manipulator and two levels for each noise parameter will result in 2^{12} combinations. To reduce computation and incorporate the effect of noises in all these parameters, suitable OA has been selected. It is observed that the L_{16} OA proposed by Taguchi fulfills the requirement.[34] This OA has 15 columns and 16 rows and presented in Table XIV. First 12 columns of this OA are assigned to design and process parameters and remaining columns are left unassigned. The procedures adopted to incorporate the effect of noises are already discussed in Section 2.2 and a representative example is also presented for 2-DOF RR manipulator. The results of the optimization process utilizing the two evolutionary techniques are presented in Tables XV and XVI. In these tables, the values of the design variables, the objective function value, and number of function evaluations for each technique are presented. It is observed that the DE algorithm converges to optimal value by 85 generations, whereas GA converged very slowly in two cases. DE approach consistently obtains smaller function values with smaller number of function evaluations and generations compared to GA. The objective function history for each case is shown in Figs. 6 and 7.

The number of function evaluations is an indication of the computational effort to reach the optimum value for the same number of generations. It is observed that the cost function for DE reduces in a monotonic fashion. Possible reasons for the computational saving can be attributed to the

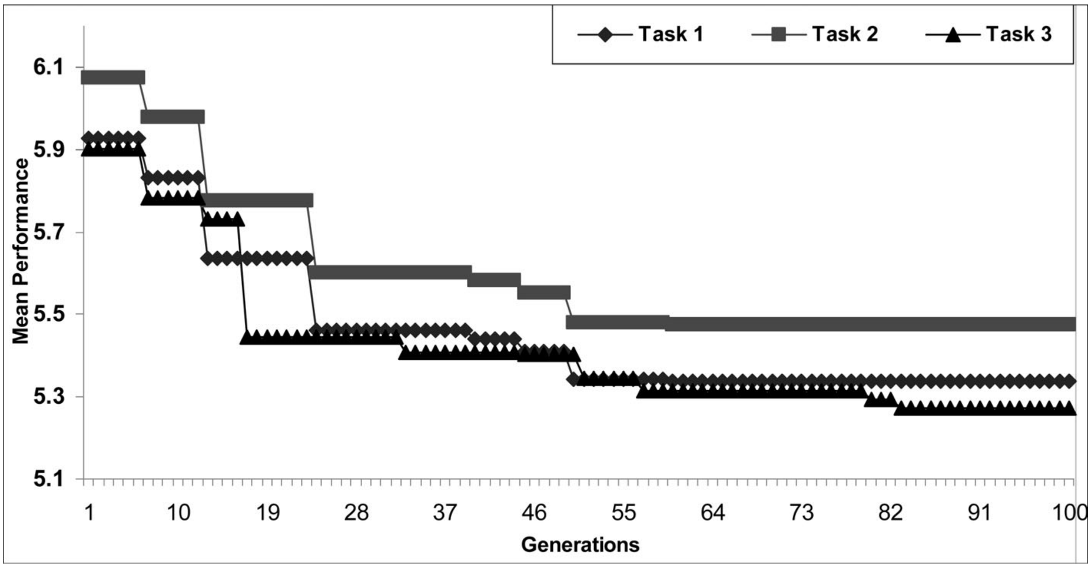

Fig. 6. Function history with modified DE.

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

Table XIV. L_{16} orthogonal array.

	Column no.														
Experiment no.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
3	1	1	1	2	2	2	2	1	1	1	1	2	2	2	2
4	1	1	1	2	2	2	2	2	2	2	2	1	1	1	1
5	1	2	2	1	1	2	2	1	1	2	2	1	1	2	2
6	1	2	2	1	1	2	2	2	2	1	1	2	2	1	1
7	1	2	2	2	2	1	1	1	1	2	2	2	2	1	1
8	1	2	2	2	2	1	1	2	2	1	1	1	1	2	2
9	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
10	2	1	2	1	2	1	2	2	1	2	1	2	1	2	1
11	2	1	2	2	1	2	1	1	2	1	2	2	1	2	1
12	2	1	2	2	1	2	1	2	1	2	1	1	2	1	2
13	2	2	1	1	2	2	1	1	2	2	1	1	2	2	1
14	2	2	1	1	2	2	1	2	1	1	2	2	1	1	2
15	2	2	1	2	1	1	2	1	2	2	1	2	1	1	2
16	2	2	1	2	1	1	2	2	1	1	2	1	2	2	1

Table XV. Optimal parameters of 4-DOF SCARA manipulator using modified DE.

Tasks	a_1 (mm)	a_2 (mm)	d_1 (mm)	d_4 (mm)	m_1 (kg)	m_2 (kg)	m_3 (kg)	m_4 (kg)	Mean performance	NOF
Task 1	425.237251	378.517724	881.049511	200.3349	18.85535	14.29462	11.99494	6.867591	5.337672	1845
Task 2	425.2373	378.5177	881.0495	200.3349	18.85535	14.29462	11.99494	6.867591	5.474955	1617
Task 3	426.6888	371.2306	873.3842	195.3365	20.95148	16.88338	11.96033	6.917716	5.271751	1678

Table XVI. Optimal parameters of SCARA using modified GA.

Tasks	a_1 (mm)	a_2 (mm)	d_1 (mm)	d_4 (mm)	m_1 (kg)	m_2 (kg)	m_3 (kg)	m_4 (kg)	Mean performance	NOF
Task 1	426.0661	372.2595	875.2899	198.0456	18.25867	16.4518	11.82267	6.845226	5.410652	4040
Task 2	426.252	377.4075	872.0849	195.0988	19.37171	13.21991	11.17549	6.679829	5.97753	4040
Task 3	423.0654	379.8969	876.1713	195.4028	18.89455	14.45755	11.93294	6.883373	5.344304	4040

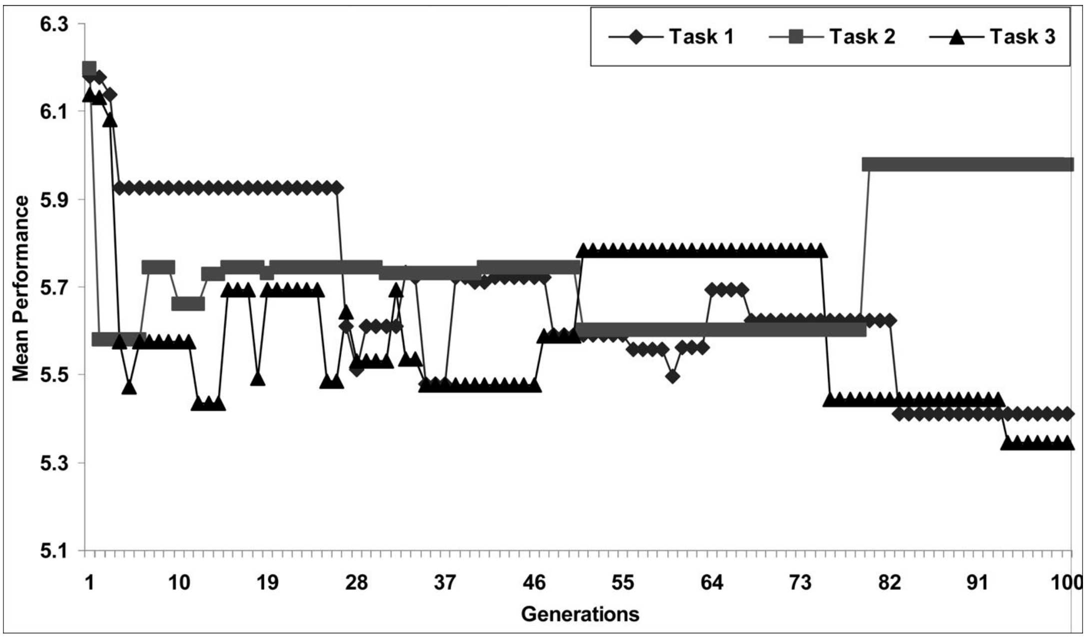

Fig. 7. Function history with modified GA.

https://doi.org/10.1017/S0263574709005700 Published online by Cambridge University Press

strategy of evolution in DE. Focus of the present investigation is to reduce the performance variations by proper selection of design parameters. It is observed that the optimal design parameters are different for different tasks. As an industrial manipulator is desired to perform several tasks, therefore, design of a manipulator to deliver optimal performance in all types of task will be difficult. One of the possibilities will be to design the manipulator for worst-case situation.

7. Conclusion

This paper discusses a systematic methodology for selection of kinematic and dynamic parameters, i.e. link lengths and masses, of manipulator, while performing a task. An evolutionary optimization approach has been proposed which couples DE optimization techniques with OA of the Taguchi method to incorporate the effect of noises in optimization process. The performance of proposed approach is compared with the performance of GA with similar modifications. To illustrate the method, parameters of a 2-DOF RR planar manipulator and a 4-DOF SCARA manipulator are considered. The objective is to minimize mean positional and orientation error of manipulator subject to different constraints. In this optimization process, the kinematic and dynamic models of the manipulators are used. The results indicate that the DE converges quickly with less number of generations and function evaluations. Hence, fast performance of DE indicates that this approach can be a viable optimization technique. Proposed approach is suitable for optimal product parameter design with effect of design and process parameter noises.

References

1. O. Khatib and J. Burdick, “Optimization of dynamics in manipulator design: The operational space formulation,” *Int. J. Rob. Automat.* **2**, 90–98 (1987).
2. S. Manoochehri and A. A. Seireg, “A computer based methodology for the form synthesis and optimal design of robot manipulators,” *ASME J. Mech. Des.* **112**, 501–507 (1990).
3. Z. Shiller and S. Sundar, “Design of multi-degree-of-freedom mechanisms for optimal dynamic performance,” *ASME J. Mech. Des.* **115**, 199–206 (1993).
4. O. Khatib and A. Bowling, “Optimization of the Inertial and Acceleration Characteristics of Manipulators,” *Proceedings of IEEE International Conference on Robotics and Automation*, MN, Vol. 4 (1996) pp. 2883–2889.
5. O. Chocron and P. Bidaud, “Genetic Design of 3D Modular Manipulators,” *Proceedings of IEEE International Conference on Robotics and Automation*, Albuquerque, NM, Vol. 1 (1997) pp. 223–228.
6. L. Stocco, S. E. Salcudean and F. Sassani, “Fast constrained global minimax optimization of robot parameters,” *Robotica* **16**, 595–605 (1998).
7. 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,” *J. Mech. Des.* **122**, 17–24 (1987).
8. S. S. Rao and P. K. Bhatti, “Probabilistic approach to manipulator kinematics and dynamics,” *Relb. Eng. Syst. Safety* **72**, 47–58 (2001).
9. P. S. Shiakolas, D. Koladiya and J. Kebrle, “Optimum robot design based on task specifications using evolutionary techniques kinematic, dynamic, and structural constraints,” *Inverse Prob. Eng.* **10**, 359–375 (2002).
10. L. Tian and C. Collins, “Optimal placement of a two-link planar manipulator using a genetic algorithm,” *Robotica* **23**, 169–176 (2005).
11. J. Y. Kim, “Task based kinematic design of a two DOF manipulator with a parallelogram five-bar link mechanism,” *Mechatronics* **16**, 323–329 (2006).
12. S. Tabandeh, C. Clark and W. Melek, “A Genetic Algorithm Approach to Solve for Multiple Solutions of Inverse Kinematics Using Adaptive Niching and Clustering,” *Proceedings of IEEE Congress on Evolutionary Computation*, Vancouver, British Columbia, Canada (2006) pp. 1815–1822.
13. P. Huang, J. Yan, Y. Xu, W. Xu and B. Liang, “Genetic Algorithms-Based Minimum Torque Path Planning for Space Manipulator,” *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 1 (2006) pp. 3575–3579.
14. G. He, H. Gao, G. Zhang and L. Wu, “Using Adaptive Genetic Algorithm to the Placement of Serial Robot Manipulator,” *Proceedings of IEEE International Conference on Engineering of Intelligent Systems* (2006) pp. 1–6. doi:10.1109/ICEIS.2006.1703222.
15. J. U. Dolinsky, I. D. Jenkinson and G. J. Colquhoun, “Application of genetic programming to the calibration of industrial robots,” *Comp. Indus.* **58**, 255–264 (2007).
16. C. A. Coello Coello, A. D. Christiansen and A. H. Aguirre, “Using a new GA-based multi-objective optimization technique for the design of robot arms,” *Robotica* **16**, 401–414 (1998).
17. J. Han, W. E. Chung, Y. Youm and S. H. Kim, “Task Based Design of Modular Robot Manipulator using Efficient Genetic Algorithm,” *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM (April 1997).
18. B. K. Rout, N. N. Sharma and R. K. Mittal, “Optimal Design of Manipulator Parameters Using Evolutionary Optimization Techniques,” *Proceedings of 3rd International Conference on Autonomous Robots and Agents*, Palmerston North, Massey University New Zealand (Dec 12–14, 2006) pp. 659–664.
19. B. K. Rout and R. K. Mittal, “Screening of factors influencing the performance of manipulator using combined array design of experiment approach,” *Rob. Comp. Integr. Manuf.* **25**(3), 651–666 (2008). doi:10.1016/j.rcim.2008.05.004.
20. C. J. J. Paredis and P. K. Khosla, “Synthesis Methodology for Task Based Reconfiguration of Modular Manipulator Systems,” *Proceedings of the 6th International Symposium on Robotics Research*, Hidden Valley, PA (Oct. 2–5, 1993).
21. C. J. J. Paredis and P. K. Khosla, “Kinematic design of serial link manipulators from task specifications,” *Int. J. Rob. Res.* **12**(3), 274–87 (1993).
22. C. J. J. Paredis, H. B. Brown and P. K. Khosla,“A Rapidly Deployable Manipulator System,” *Proceedings of the IEEE International Conference on Robotics and Automation*, Minneapolis, MN (April 1996) pp. 1434–1439.
23. C. J. J. Paredis, An Agent-Based Approach to the Design of Rapidly Deployable Fault Tolerant Manipulators, *Ph.D. Thesis* (Department of Electrical and Computer Engineering, Carnegie Mellon University, August 1996).
24. G. Yang and I.-M. Chen, “Task-based optimization of modular robot configurations-minimized degree-of-freedom approach,” *Mech. Mach. Theory* **35**(4), 517–540 (2000).
25. I.-M. Chen, “Rapid response manufacturing through a rapidly reconfigurable robotic workcell,” *Rob. Comp. Integr. Manuf.* **17**, 199–213 (2001).
26. I.-M. Chen, “Realization of a rapidly reconfigurable robotic workcell,” *J. Jpn. Soc. Precision Eng.* **66**(7), 1024–1030 (2000).
27. R. Storn, “Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim.* **11**, 341–359 (1995).
28. R. Storn, *Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay* (International Computer Science Institute, Berkeley, California, 1995).

29. K. Price and R. Storn, "A simple evolution strategy for fast optimization," *Dr. Dobb's J.* **264**, 18–24 (1997).
30. K. Price and R. Storn, "Home page of differential evolution," URL: http://www.ICSI.Berkeley.edu/_storn/code.html (2006).
31. Kurt Hacker and Kemper Lewis, "Robust Design Through the use of a Hybrid Genetic Algorithm," *Proceedings of DETC'02, ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Montreal, Canada (Sep. 29–Oct. 2, 2002).
32. T. P. Bagchi, "Multi-objective robust design by genetic algorithms," *Mater. Manuf. Process.* **18**, 341–354 (2003).
33. P. G. Hoel, S. C. Port and C. J. Stone, *Introduction to Stochastic Processes* (Houghton Mifflin Company, Boston, USA, 1986).
34. S. H. Park, *Robust Design and Analysis for Quality Engineering* (Chapman and Hall, London, 1998).
35. G. E. P. Box, G. M. Jenkins and C. R. Gregory, *Time series Analysis: Forecasting and Control*, 3rd ed. (Pearson Education, Singapore, 1994).
36. J. J. Craig, *Introduction to Robotics Mechanics and Control*, 2nd ed. (Pearson Education, Singapore, 1989).
37. R. K. Mittal, I. J. Nagrath, *Robotics and Control* (Tata McGraw-Hill Publishing Company Limited, New Delhi, 2003).
38. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control* (John Wiley & Sons, Singapore, 1989).
39. K. Deb, *Optimization for Engineering Design: Algorithms and Examples* (Prentice Hall of India Private Limited, New Delhi, 2003).
40. G. C. Onwubolu and B. V. Babu, *New Optimization Techniques in Engineering* (Springer, Heidelberg, Germany, 2004).