Available online at www.sciencedirect.com
SCIENCE DIRECT®
PERGAMON
Engineering Applications of Artificial Intelligence 15 (2002) 241–252
Engineering Applications of
ARTIFICIAL INTELLIGENCE
www.elsevier.com/locate/engappai

Optimization techniques applied to multiple manipulators for path planning and torque minimization

Devendra P. Garg[a,*], Manish Kumar[b]

[a] Department of Mechanical Engineering, Duke University, Box 90300, Durham, NC 27708-0300, USA
[b] Department of Mechanical Engineering and Matl's Science, Duke University, Box 90300, Durham, NC 27708-0300, USA

Accepted 21 September 2002

Abstract

This paper presents the formulation and application of a strategy for the determination of an optimal trajectory for a multiple robotic configuration. Genetic Algorithm (GA) and Simulated Annealing (SA) have been used as the optimization techniques and results obtained from them compared. First, the motivation for multiple robot control and the current state-of-art in the field of cooperating robots are briefly given. This is followed by a discussion of energy minimization techniques in the context of robotics, and finally, the principles of using genetic algorithms and simulated annealing as an optimization tool are included. The initial and final positions of the end effector are specified. Two cases, one of a single manipulator, and the other of two cooperating manipulators carrying a common payload illustrate the proposed approach. The GA and SA techniques identify the optimal trajectory based on minimum joint torque requirements. The simulations performed for both the cases show that although both the methods converge to the global minimum, the SA converges to solution faster than the GA.

Keywords: Genetic algorithm; Simulated annealing; Cooperating robots; Path planning; Torque minimization; Performance index

1. Introduction

Multiple cooperating robots can be used to facilitate various operations undertaken in manufacturing industry. For example, tasks such as assembly of components and flexible material handling operations can be more efficiently carried out using two manipulators instead of one. Moreover, multiple robots make manufacturing systems more flexible and these systems become capable of handling more complex operations. However, most of these operations are repetitive in nature and do not require much decision making. All of these considerations inspire research in real time control of multiple cooperating robotic systems. A lot of research has been carried out in the robotics and manufacturing automation (RAMA) Laboratory at Duke University for coordinating position, motion, and grasping force for multi-manipulator systems acting in synchronization (Garg, 2000; Garg, 1992; Nagchaudhuri and Garg, 1992). A number of control strategies, such as fuzzy logic (Prabhu and Garg, 1998; Woodard and Garg, 1999; Prabhu and Garg, 1995) neural networks (Garg et al., 1999; Prabhu and Garg, 1996; Ananthraman and Garg, 1993a, b), general redundancy optimization method (Zhang and Ferch, 1998), and two level hierarchical fuzzy logic for hyperredundant cooperating robots (Ivanescu and Bizdoaca, 2000) have been published in technical literature.

Specifically, Zhang and Ferch (1998) have applied an approach to online automatic learning of a B-spline fuzzy controller. The objective was to enable a two-arm manipulator system to perform complex cooperating tasks such as jointly carrying a rigid object. This controller model directly connects the sensor inputs to the compensation motion. By using the adaptation of control actions in all possible situations through practicing of the robots in the real environment, uncertainties of the robot-object model can be taken into account. Due to on-line learning approach, the

*Corresponding author.
E-mail addresses: dpgarg@duke.edu (D.P. Garg), manish@duke.edu (M. Kumar).

0952-1976/02/$ - see front matter
PII: S0952-1976(02)00067-2

compliant motion of the robot system can be adapted to new situations in a relatively short time. Similarly, Ivanescu and Bizdoaca (2000) have proposed a two-level hierarchical fuzzy controller to solve the control problem for a multi-chain robotic system formed by tentacle manipulators grasping a common object with hard point contacts. The control system consists of two parts: the first component is a conventional controller which implements a control strategy based on Lyapunov stability and the second one is an adaptive fuzzy controller which adjusts the control parameters by the output of the first level controller.

In a recent paper, Kwon and Lee (1998) have proposed a new force distribution scheme for multiple cooperating robots in which duality theory of non-linear programming (NLP) is combined with the quadratic programming (QP) approach. The optimal force distribution of the problem is formulated as a QP problem with both linear and quadratic constraints, and an efficient algorithm obtains its solution. The use of quadratic constraints considerably reduces the number of constraints, thus enabling the dual method of NLP to be used in the solution algorithm. Moreover, it can treat norm constraints without approximation, such as bound of the norm of the force exerted by each robot. Murphy et al. (1991) have formulated a problem of a system of two mobile cooperating robots that form a closed kinematic chain. The formulation includes the full dynamic interactions from arms to platforms and from arm tip to arm tip, and the possible translation and rotation of the platform. The equations of motion are shown to be identical in structure to the fixed platform cooperative manipulator dynamics. The solution to cooperative motion is able to incorporate any form of solution to the forward dynamics of a topological tree of manipulators and a platform.

In the area of path planning, Han et al. (1997) have presented an intelligent navigation architecture for micro-robots playing soccer games. In their proposed navigation system, the central path planner uses a genetic searching algorithm to generate and modify consecutive via-points that micro-robot soccer players must follow to avoid moving obstacles and reach the goal position. The low-level on-line navigation algorithm is also available for each micro-robot, which accomplishes dynamic local path planning and tracking of each path between via-points generated from the central path planner. To facilitate the path planning procedure, the position and orientation of each mobile robot as well as the ball and goal post are detected using a color vision system in which the dotted scanline method is developed and applied to the rows and columns of digitized image plane. It is evident from a review of current literature that a whole lot of research effort is underway that deals with the problem of controlling cooperative robots.

Energy requirement has been an important aspects of a physical system and its minimization is generally desirable. A lot of research has been carried out to search for the trajectory generation strategies based on the concept of energy minimization. For example, Hirakawa and Kawamura (1996) have proposed a method to solve the trajectory generation problem in redundant degree of freedom manipulators. They have used a variational approach and the B-Spline curve is introduced for minimization of the consumed electrical energy of a robot manipulator system. The application of this method is oriented to repeated jobs realized by industrial robot manipulators. Similarly, Delingette et al. (1992) have presented a method to generate curvature constrained trajectories for which the curvature profile is a polynomial function of arc length. An algorithm based on the deformation of a curve by energy minimization allows solving general geometric constraints, which had not been possible by previous methods. Furthermore, they were able to take into account the limitation of radius of curvature of the robot by controlling the extrema of curvature along the path.

The subject of energy minimization continues to be of interest in the robotics and automated manufacturing context. For example, in a related research effort, Garg and Ruengcharungpong (1992) have proposed a strategy for force balance and energy optimization for cooperating manipulators. For simulation, two SCARA robots forming a closed kinematic chain were controlled using their individual controllers. A position control strategy was used for each robot, and the corresponding end effector forces were calculated. These forces were equalized and corresponding power used was computed. They employed linear programming technique to calculate external forces such that the power used in the direction of motion was minimized.

In path planning problems, the number of feasible paths between the initial position and final position of a robot is often very large, and the goal is not necessarily to determine the best solution, but to obtain an acceptable one according to certain requirements and certain constraints. Various search methods have been developed (e.g., calculus-based methods, enumerative schemes, random search algorithms, etc.) for the robot path-planning problem. Enumerative schemes are not effective when the search space is too large to explore all possible paths. Random search algorithms are probabilistically complete, but may take a long time to find a solution. In their study, Pin and Culioli (1992) have applied a projected sub gradient algorithm to solve the minimax problem for joint torque distribution optimization, but the run time was long and the result obtained was a local minimum. Chen and Zalzala (1997) have applied Genetic Algorithmic approach to multi-criteria motion planning of mobile manipulator systems. Minimum distance of travel and path safety were considered as the two criteria for the mobile robot path planning. The emphasis of the study was placed on using genetic algorithms to search for global optimal solutions and solve the minimax problem for manipulator torque distribution. Various simulation results from two examples show that the proposed genetic algorithm approach performs better than the conventional search methods. Similarly, Sexton and Gupta (2000) have carried out a comparative evaluation of genetic algorithm and back-propagation for training neural networks (NNs) for five chaotic time series. Their results show that the Genetic Algorithms are superior to Back-Propagation in effectiveness, ease-of-use and efficiency for training NNs. For every problem considered, the Genetic Algorithm approach was found to provide statistically superior solutions in less CPU time. Painton and Campbell (1995) have used Genetic Algorithmic optimization techniques to design an optimization model that identifies the types of component improvements and the level of effort spent on those improvements to maximize one or more performance measures (e.g., system reliability or availability) subject to constraints (e.g., cost) in the presence of uncertainty about component failure rates. Results and comparison with enumeration of the configuration space show that genetic algorithms perform very favorably in the face of noise in the output and they are able to find the optimum over complicated, high dimensional, nonlinear space in a tiny fraction of the time required for enumeration.

Genetic Algorithm (GA) based search and optimization techniques have recently found increasing use in machine learning, robot motion planning, scheduling, pattern recognition, image sensing and many other engineering applications. In principle, GAs are search algorithms based on mechanics of natural selection and natural genetics. They combine survival of the fittest among the string structures with randomized yet structured information exchange to form a search algorithm with innovative flair of natural evolution.

A GA starts with a random creation of a population of strings and thereafter generates successive populations of strings that improve over time (Goldberg, 1989). The processes involved in the generation of new populations mainly consist of the following operations that are illustrated in Fig. 1.

1. *Reproduction*: Reproduction is a process in which individual strings are copied according to their objective function values, 'f' (also called fitness function), which measures profit, utility or goodness that needs to be maximized. Strings with a higher fitness value have a probability of contributing one or more offspring in the next generation. The reproduction operator may be implemented in an algorithmic form in a number of ways such as Roulette wheel selection, rank selection, or steady state selection. Once a string has been selected for reproduction, an exact replica of the string is made. This string is then entered into the mating pool, a tentative new population for further genetic operator action.
2. *Crossover*: After reproduction, simple crossover may proceed in two steps. First, members of newly reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows: an integer position 'k' along the string is selected uniformly at random between 1 and string length l minus one i.e., $(1, l-1)$. Two new strings are created by swapping all the characters between positions $(k+1)$ and l inclusively.
3. *Mutation*: Mutation is a random alteration of the value of a string position. In binary coding, this

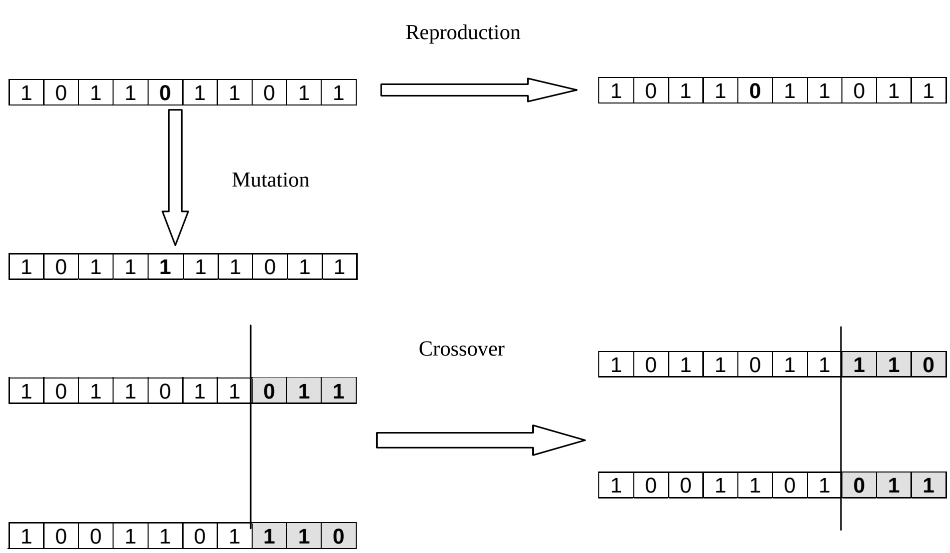

Fig. 1. Schematic representation of basic genetic algorithm operations.

means changing a 1 to 0 and vice versa. In GA, its probability of occurrence is generally kept small, as a higher occurrence rate would lead to a loss of important data. GA, with 100 % mutation rate becomes random search in the solution space.

GAs have proven their robustness and usefulness over other search techniques because of their unique procedures that differ from other normal search and optimization techniques in four distinct ways:

1. GAs work with coding of a parameter set, not the parameters themselves.
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivative or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic rules.

In the context of determining a suitable manipulator trajectory, a number of methods have been proposed and researched in recent years. For example, Monteiro and Madrid (1999) have used GA to plan the stages of the trajectory of a robot arm called Jeca III. They have proposed the use of GA to plan a trajectory with obstacle avoidance and implement joint space using classical GA. This is achieved in two stages: initial positioning, which locates the end effector of robot arm in first point of trajectory, and incremental positioning which moves the end effector to the next point of trajectory. Pires and Machado (1999) have used GA to generate collision free trajectories for robotic manipulators with the objective to minimize the path length and ripple in time evolution of robot positions and velocities. They have used direct kinematics for this purpose and have presented results for several redundant and non-redundant robot manipulators.

Similarly, Watanabe et al. (1999) have described a method for the path planning of an omni-directional mobile manipulator by applying an evolutionary strategy. Initial and final orientations and arrival time are specified in advance. The approach automatically selects points in a wide range of data points, minimizing or maximizing the total cost function, which consists of several sub-cost functions such as motion smoothness, movable range of joint, singular orientation etc. The points are then used to form a trajectory by fitting in a B-spline curve. In their recent paper, Choi et al. (1999) have used GA for trajectory optimization and applied it to biped robots. They have proposed a method to find optimal via-points using GA that minimizes the sum of deviation of velocities and acceleration as well as jerk to obtain continuity on the entire trajectory interval and energy distribution. The continuous velocity and acceleration at the via points ensure a smooth biped walking.

Makino et al. (1999) have proposed the development of a motion planning system which yields an optimal work pass for an autonomous agricultural vehicle in a farm land. Their system consists of two parts: global path planning and local motion planning. The global path planning component works to acquire an optimal work path for the whole field. In this case, the optimal work path is the lowest traveling cost from a start point to a goal point. The local motion planning component acquires the optimal path and plans an optimal control policy in a headland. These components are implemented with simulated annealing, TABU search, GA and reinforcement learning algorithm.

This paper investigates the use of simulated annealing (SA) also applied for path planning problem. Simulated annealing works on a principle analogous to the formation of crystals in cooling solids. A solid material heated beyond certain temperature will become liquid and when cooled back slowly, it will form crystals in a minimal energy state. Similar to GA, the algorithm is based on formulation of a fitness or cost function that represents the relative merit of a point in search space. The search algorithm operates as follows. The algorithm starts with a point in solution space. From the algorithm's current position, a neighboring point is chosen at random. The difference in the fitness function value between the new point and the current point, ΔC, is calculated. The difference is used together with the current system temperature, t, to calculate the probability of the new position being accepted. The probability is given by the distribution $e^{-\Delta C/t}$. The process continues with the same temperature t either for given number of iterations or until a given number of positions have been occupied, at which time the value of t is decreased. The temperature decreases until no transitions are possible, so the system remains frozen in one position. This freezing occurs only when ΔC is positive for all neighboring points, which means that the position must be a local minimum or may be a global minimum. The advantages of simulated annealing include:

(a) SA can process cost functions possessing quite arbitrary degrees of nonlinearities, discontinuities, and stochasticity.
(b) SA can process quite arbitrary boundary conditions and constraints imposed on these cost functions.
(c) SA be implemented quite easily with the degree of coding quite minimal relative to other nonlinear optimization algorithms
(d) SA can statistically guarantee finding an optimal solution.

However, SA has some drawbacks too. For example:

(a) SA can be quite time-consuming to find an optimal fit.

(b) SA can be difficult to fine tune to specific problems, relative to some other fitting techniques.
(c) SA can suffer from "over-hype" and faddish misuse, leading to misinterpretation of results.

The research results presented in this paper make use of adaptive simulated annealing (ASA), an algorithm presented in Ingber (1996), which overcomes many of the above shortcomings. ASA is a global optimization algorithm that relies on randomly importance-sampling the parameter space, i.e., in contrast to utilizing deterministic approaches. This algorithm fits empirical data to a theoretical cost function over a *D*-dimensional parameter space, adapting for varying sensitivities of parameters during the fit.

Simulated Annealing has been used extensively in non-linear optimization problems. For example Bowman and Ingber (1997) have presented a set of tools that can be applied as such supplemental indicators, based on stochastic nonlinear multivariate modeling used to benchmark Janus simulation to exercise data from the U.S. Army National Training Center (NTC). As a prototype study, they used ASA to explicitly fit Janus data, deriving coefficients of relative measures of effectiveness, and developing a sound intuitive graphical decision aid, canonical momentum indicators (CMI), faithful to the sophisticated algebraic model.

Moreover, SA has been used in field of finance, neuroscience, robotics and lots of other fields. Ingber and Mondescu (2001), in their paper, have used ASA code for designing optimum trading model. They have described an end-to-end real-time S&P futures trading system. They developed inner-shell stochastic nonlinear dynamic models, and derived CMI from a fitted Lagrangian used by outer-shell trading models dependent on these indicators. Recursive and adaptive optimization using ASA has been used for fitting parameters shared across these shells of dynamic and trading models.

In the field of robotics as well, simulated annealing has been extensively used for optimization problems. In motion planning of robots, determining minimum distance between two bodies is one of the most frequently encountered problem. Some earlier approaches propose to partition the concave object into convex sub objects and then solve the convex problem between all possible sub-objects combination. Carretero et al. (2001) in their paper, have proposed an optimization approach without portioning the concave object into several convex sub-objects. They have used simulated annealing to solve the concave problem. They have proposed to replace the objects' geometry by a set of points on the surface of the body. This reduces the problem to finding the combination of two points for which the distance will be minimum.

For motion planning and obstacle avoidance for mobile robots, artificial potential field methods provide simple and effective motion planners for practical purpose. However, formation of local minima, which can trap a robot before reaching its goal, is a major problem. Park et al. (2001) have presented and applied the mobile robot path planning technique which integrate the artificial potential field approach with simulated annealing.

This research paper proposes a path planning method that searches the manipulator configuration for an optimum path based on performance index value defined. An optimum path is the one that requires the minimum amount of torques and hence leads to low energy consumption. The research makes use of both GA and SA algorithms and makes a comparison of performances of these two algorithms for this optimization problem.

2. Problem formulation

Given the initial and final positions of a robot end effector, the problem of finding an optimal path to be followed is considered in this research paper. The problem consists of locating a specific path that requires the least amount of torque amongst several possible paths. It is evident that the end effector, in moving between any two specified end points, can follow a variety of paths. All such paths require different amounts of torque depending upon the distance covered, the velocity and the acceleration achieved, and the payload carried.

3. Approach

Each joint trajectory is assumed to be a polynomial of 4th degree in time, t. If $\theta(t)$ is the angle between a link and the x-axis or made with the other link, a fourth-order polynomial can be expressed in the form:

$$\theta(t) = at^4 + bt^3 + ct^2 + dt + e, \tag{1}$$

where $a,\ b,\ c,\ d$ and e are parameters to be determined based on end conditions (Yoshikawa, 1990). Angle 'θ' is considered to be positive in counterclockwise direction.

The problem of a two-link manipulator system has been chosen for this study because of its simplicity and for the ease of demonstrating the validity of the approach proposed. However, the methodology is equally applicable to more complex manipulators having a larger number of joints. The initial and final boundary conditions of angular displacement and angular velocity are applied to two equations for each link. Applying these boundary conditions gives relationship between the four coefficients of the polynomial in terms of the fifth. Optimal value of this fifth coefficient ($a1$ and $a2$ for link 1 and link 2) is determined.

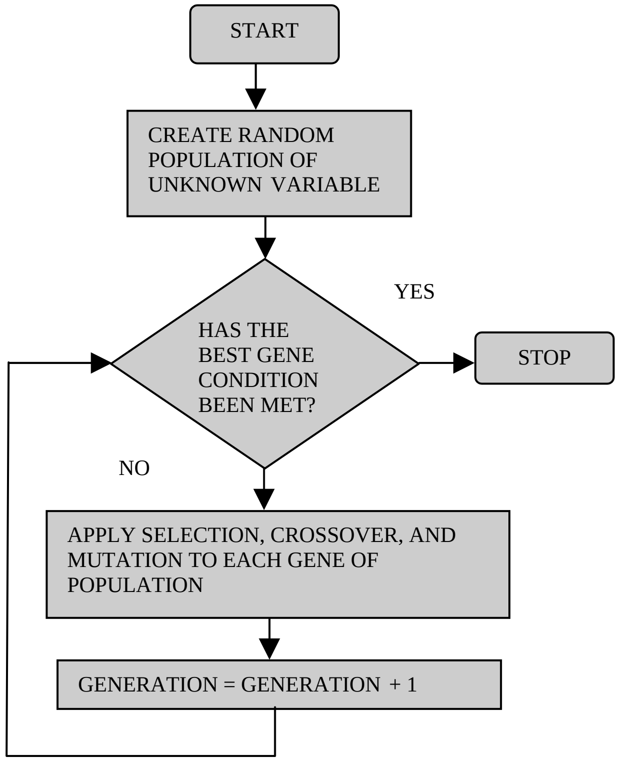

Fig. 2. Flow chart representing the optimization process using the genetic algorithm strategy.

In this paper, the GA and ASA technique have been used to solve the above problem of finding the two variables left to be determined. Flowchart in Fig. 2 shows the process that a GA takes to find an optimum solution.

4. Examples

Two different yet related situations have been considered below.

Case I: A simple two-link revolute joint manipulator robot (see Fig. 3).

The following relationship has been formulated to specify the corresponding fitness function:

$$\text{Performance Index (PI)} = \sum \left| \sqrt{\tau_1^2 + \tau_2^2} \right|, \tag{2}$$

where τ_1 and τ_2 are the actuator torques applied at joints 1 and 2, respectively.

Case II: Two cooperating robots carrying a payload of mass 'M' (see Fig. 4).

The fitness function in this case is specified by

$$\text{Performance Index (PI)} = \sum \left| \sqrt{\{\tau_{11}^2 + \tau_{12}^2 + \tau_{21}^2 + \tau_{22}^2\}} \right|, \tag{3}$$

where τ_{11} and τ_{21} are actuator torques applied at joints 1 and 2 of first robot, respectively; and τ_{12} and τ_{22} are actuator torques applied at joints 1 and 2 of second robot, respectively.

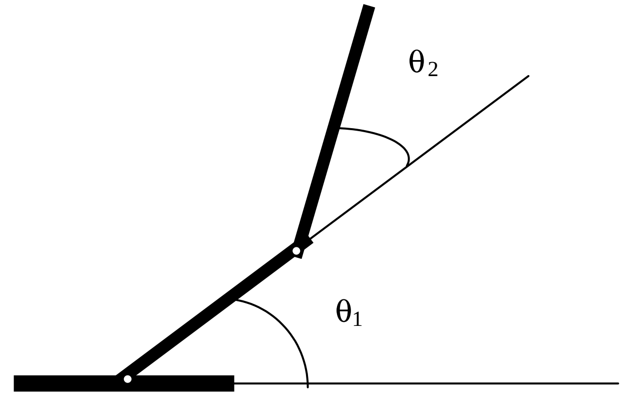

Fig. 3. Schematic diagram of a two link manipulator.

The summation sign in Eqs. (2) and (3) denotes that the quantities are being summed at particular interval of time for the total duration of travel.

Case II has an added complexity of the estimation of internal forces that come into play because of the maneuvering of a common object by two robots. This problem of estimating internal forces for optimum torque requirement was also solved using GA and ASA. The internal forces that come into play are planar forces. That means there are two components of forces acting on each robot. Thus, four equations are needed to calculate four components of the forces acting on both the robots. Motion of object in X and Y direction gives two equations. The following two relationships are assumed, with parameters n_1 and n_2 to be optimized by GA and ASA.

$$Fx_1 = n_1 Fx_2, \tag{4}$$

$$Fy_1 = n_2 Fy_2, \tag{5}$$

where Fx_1 and Fx_2 are forces acting in the x-direction on robot 1 and 2, respectively, and Fy_1, Fy_2 are forces acting in the y-direction on robot 1 and 2, respectively.

The end effector of the second robot simply follows the path described by the end effector of the first robot. The path of the first robot is determined by generating the parameters using GA and ASA. That leaves the joint angles of the second robot to be determined in the same coordinate frame. This problem was solved by applying the Newton–Raphson method. The joint angles for the second robot were determined using the position of end effector known from the first robot.

The computed torque τ applied to the joints (Craig, 1989) is given by the following equation:

$$\tau = M(\Theta)\ddot{\Theta} + V(\Theta, \dot{\Theta}) + G(\Theta), \tag{6}$$

where n is the number of joints, $M(\Theta)$ is $(n \times n)$ mass matrix, $V(\Theta, \dot{\Theta})$ is the $(n \times 1)$ vector of centrifugal and

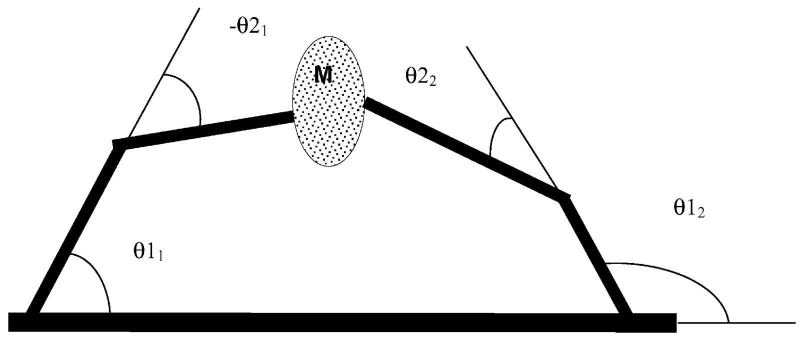

Fig. 4. Schematic diagram of two link cooperating robots carrying a payload of mass 'M'.

Coriolis terms, and $G(\Theta)$ is the $(n \times 1)$ vector of gravity terms.

5. Simulation

The simulator consists of:

1. The GA code (1998) (which initializes population of two parameters in the first case and three parameters in the second case) uses its operators to find optimum value of these coefficients based on the expression for performance index.
2. An ASA code which finds the optimum parameter value with respect to a PI based on ASA Code (Ingber, 1996).
3. Kinematics and dynamics solver which determines angular displacement, angular velocity, angular acceleration and torque.
4. Plotter which plots graphs illustrating various properties.

In both of the cases considered in this paper, a population of 100 elements (main population), randomly created, is used. Each of these elements represents a trajectory.

6. Simulation results

The GA and ASA techniques use their operators and functions to find the value of $a1$ and $a2$ for which the performance index is minimum. These values of $a1$ and $a2$ determine the trajectory followed by the end effector. The trajectory thus obtained has minimum torque requirement value and thus low energy requirement. For simulation, following parameters have been used:

Length of Link1 (Robot1): 0.2 m
Length of Link2 (Robot1): 0.4 m
Length of Link1 (Robot2): 0.8 m
Length of Link2 (Robot2): 0.9 m
Mass of Link1 (Robot1): 1 kg
Mass of Link2 (Robot1): 2 kg

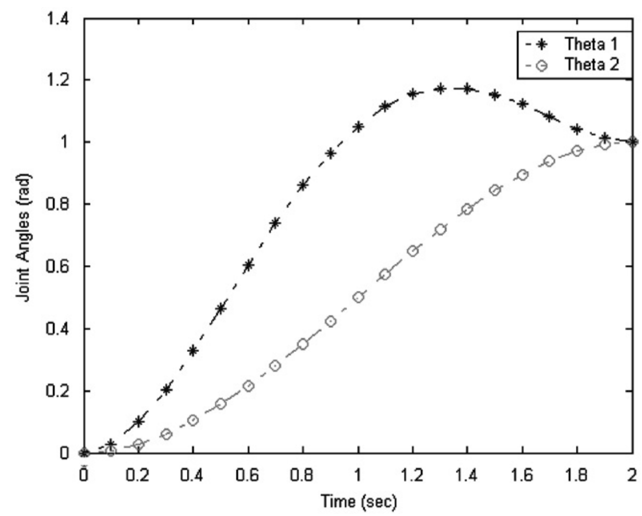

Fig. 5. Joint angles vs. time for Case I.

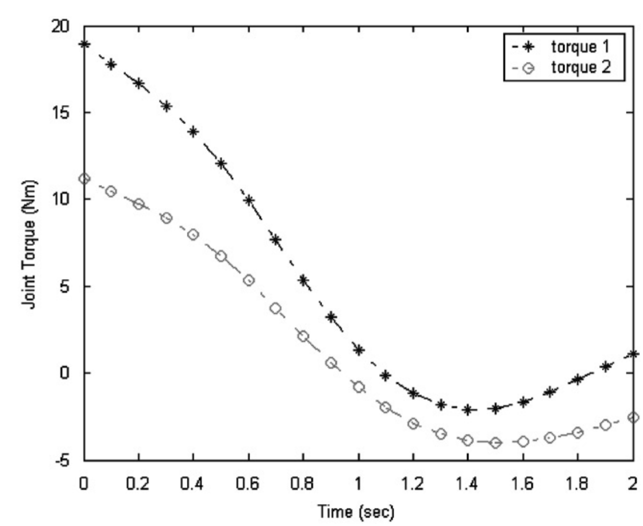

Fig. 6. Joint torques for both joints vs. time for Case I.

Mass of Link1 (Robot2): 4 kg
Mass of Link2 (Robot2): 4.5 kg
Mass of payload: 2 kg
Distance between the base points of two robots:1m

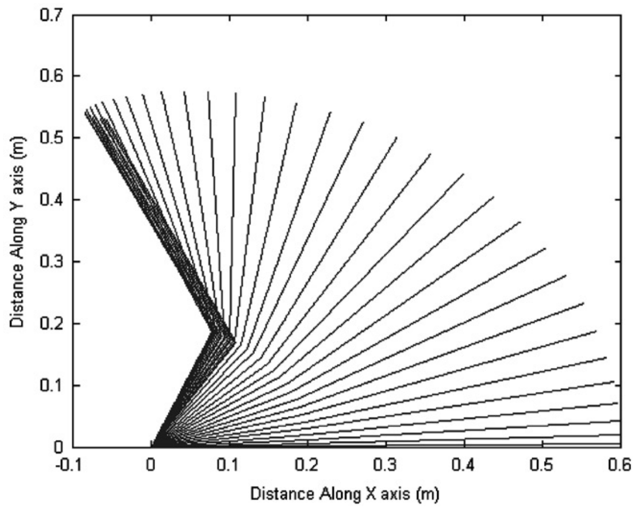

Fig. 7. Successive temporal positions of the manipulator links at 0.05 s interval for Case I.

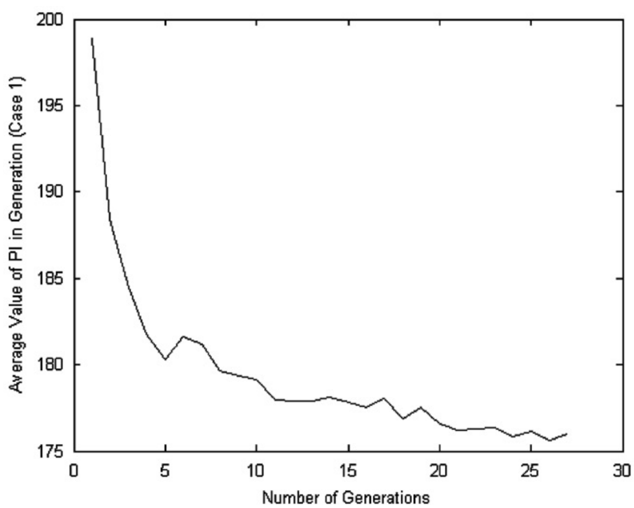

Fig. 8. Average value of performance index (PI), in a generation vs. number of generations for Case I obtained using GA.

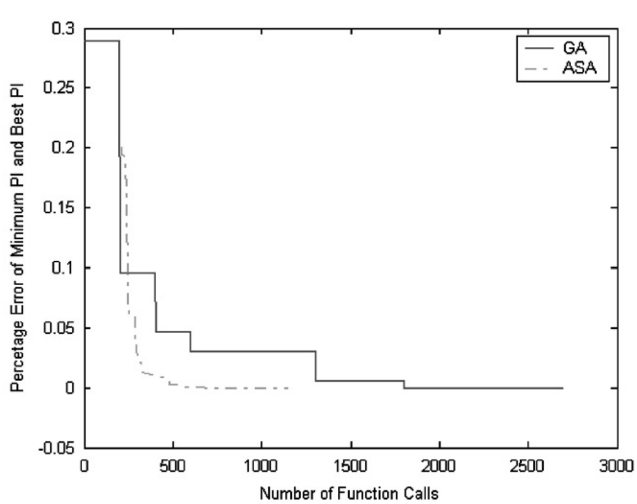

Fig. 9. Percentage error ((Minimum PI – Best) $\times$ 100/Best PI) vs. number of function calls for Case I.

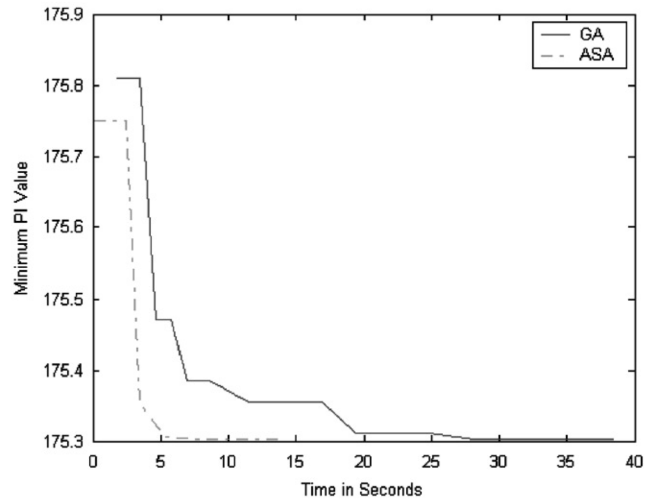

Fig. 10. Minimum PI value reached vs. time required to reach that point for Case I.

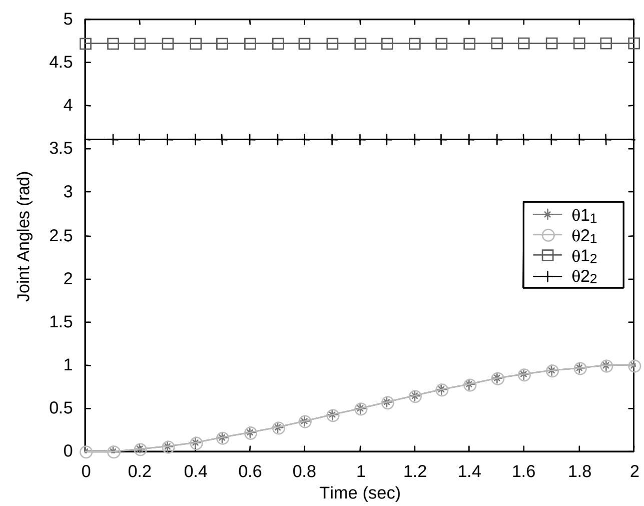

Fig. 11. Joint angles vs. time for Case II.

The results obtained from the simulation for both cases considered are shown in Figs. 5–11 and are discussed below:

Case I: Initial and final joint angles of the robotic manipulator have been specified as

$$\theta_1(\text{initial}) = 0 \text{ rad}, \quad \theta_1(\text{final}) = 1 \text{ rad},$$
$$\theta_2(\text{initial}) = 0 \text{ rad}, \quad \theta_2(\text{final}) = 1 \text{ rad}.$$

The arrival time has been specified to be equal to 2 s.

Fig. 5 shows the variation of θ_1 (Theta 1) and θ_2 (Theta 2) (angles as shown in Fig. 3) with time. The plot shows that both θ_1 and θ_2 start from initial position at $t = 0$ s, and reach the final position at $t = 2$ s. Fig. 6 shows the variation of torques applied to first (torque 1) and second joint (torque 2) vs. time. It can be seen from the plot of torques that the initial torque requirement is high and it decreases with time. It becomes negative when the joint actuator attempts to rotate the first arm in the opposite direction.

Fig. 7 shows the temporal position of the manipulator links at a time interval of 0.05 s. Fig. 8 shows the average value of performance index (GA optimization) in a particular generation versus number of generation. A continuous decrease in average value of performance index is also indicative of a smooth convergence to a solution. Fig. 9 shows the percentage error, which is the error of minimum PI value at certain iteration and the optimum or Best PI value, plotted against the number of function calls for both of the optimization techniques. It can be seen that the rate of convergence of ASA is very fast as compared GA. ASA takes approximately 300 function calls to reach appreciably close to the optimum solution, while GA takes approximately 1300 function calls to reach that value. The optimum value of PI or the Best PI is 175.302. Fig. 10 is a similar plot in which the minimum PI value is plotted against time taken to reach that point in search space by respective algorithms. ASA takes approximately 7 s to converge to the optimum solution while GA takes approximately 19 s to converge.

Following are the values of parameter that have been obtained:

$$a1 = 0.5512; b1 = -2.4548; c1 = 2.9548, d1 = 0, e1 = 0;$$
$$a2 = 0; b2 = -0.2500; c2 = 0.7500, d2 = 0, e2 = 0;$$

Case II: This case has the same initial conditions as the first case. The end effector of the second robot simply follows the path taken by end effector of first robot in a master-slave mode.

Fig. 11 shows variation of $\theta 1_1$, $\theta 2_1$, $\theta 1_2$ and $\theta 2_2$ (angles as shown in Fig. 4). The simulation yields a zero value for both $a1$ and $a2$. The curves for $\theta 1_1$ and $\theta 2_1$ versus time are analytical functions given by Eq. (1), and are coincident. Fig. 12 shows the variation of torque requirement with time. Again, it can be seen that the torque requirement is initially high and it becomes negative in order to slow down the motion of links.

Similar to Case I, the average value of performance index (GA optimization) versus number of generations lapsed also decreased and stabilized showing convergence of algorithm to a solution, which is evident in Fig. 13. Fig. 14 shows percentage error obtained by both GA and SA algorithms for Case II plotted against the number of function calls. In this case, the minimum value of PI is 812.7050. This plot also shows that ASA converges to optimum solution much faster than a GA. Fig. 15 shows minimum PI value plotted against time taken to reach that value for both algorithms. Finally,

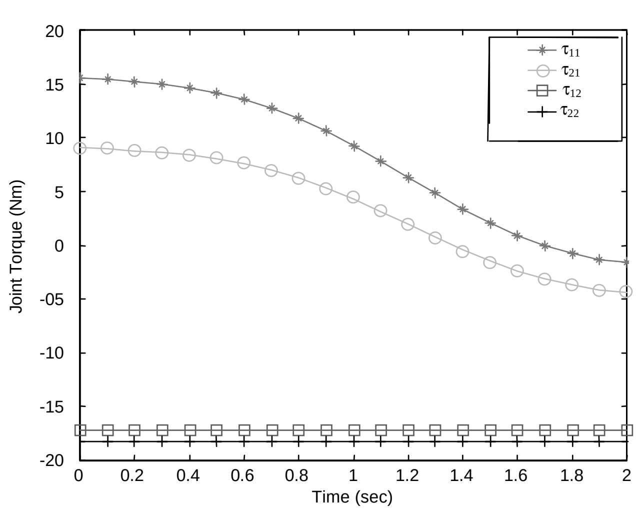

Fig. 12. Joint torques vs. time for Case II.

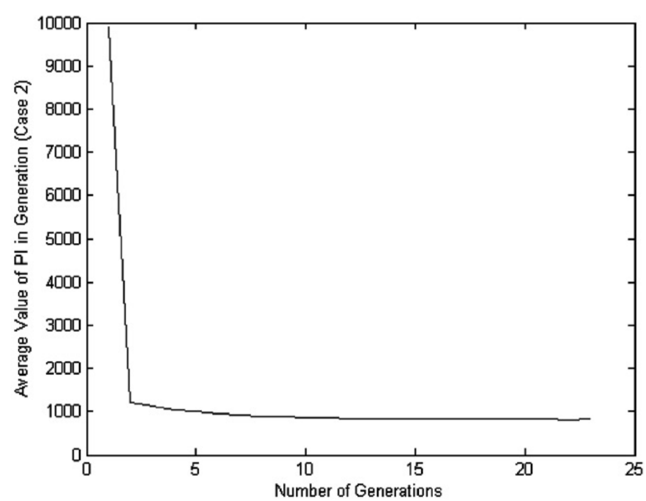

Fig. 13. Average value of PI, Best PI, in a Generation vs. number of generations for Case II.

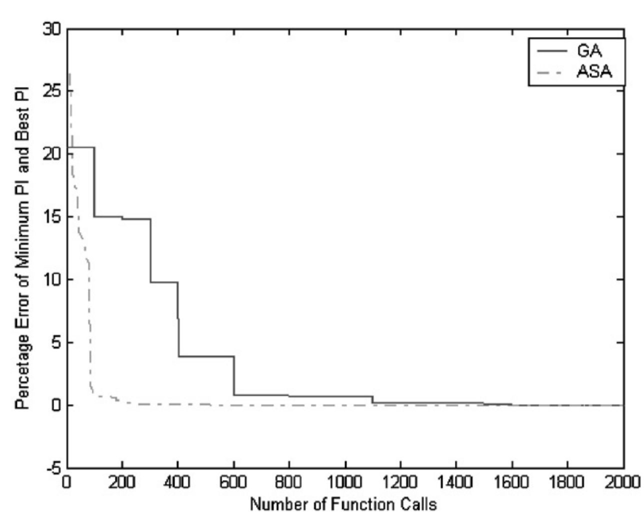

Fig. 14. Percentage error ((Minimum PI – Best) × 100/Best PI) vs. number of function calls for Case II.

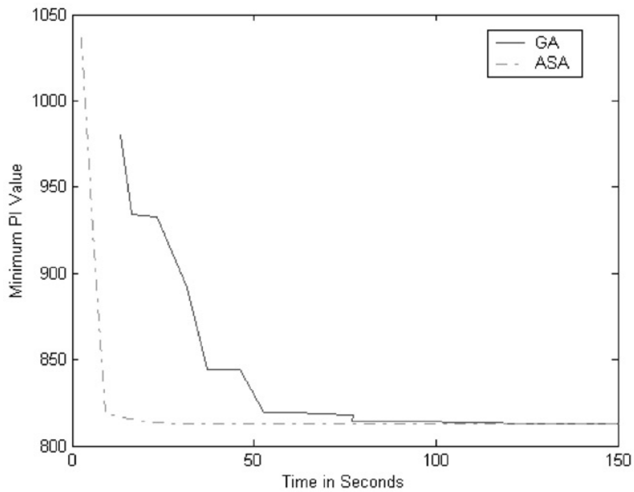

Fig. 15. Minimum PI value reached vs. time required to reach that point for Case II.

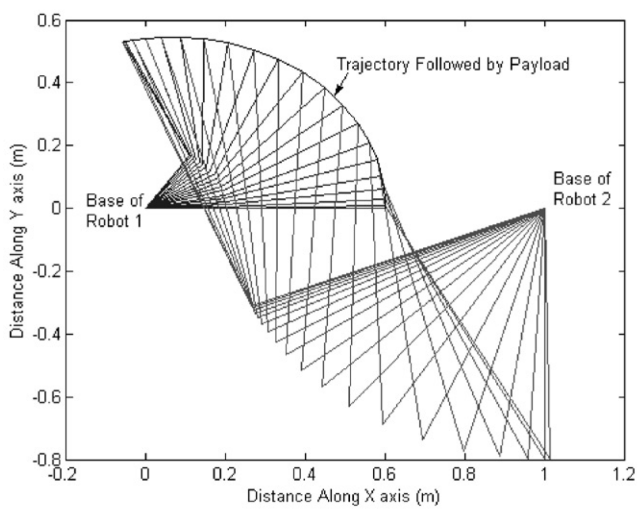

Fig. 16. Successive temporal positions of the manipulator links at 0.1 s interval for Case II.

Fig. 16 shows the temporal position of the manipulator links at a time interval of 0.1 s.

Following are the values of parameter that have been obtained for Case II:

$$a1 = 0;\ b1 = -0.2500;\ c1 = 0.7500,\ d1 = 0, e1 = 0;$$
$$a2 = 0;\ b2 = -0.2500;\ c2 = 0.7500,\ d2 = 0, e2 = 0$$

and

$$n_1 = 0.500,\ n_2 = 0.500.$$

7. Conclusions

In this paper, two optimization schemes, ASA and GA were used to find an optimal trajectory for robotic applications. The optimum trajectory thus obtained had minimum torque requirements. The case of a two-link robot and two cooperating robots were considered. Initial and final position and arrival time were specified. The simulation results reported in the paper demonstrate the superior nature of ASA as compared to GA for the problem considered. Both the algorithms reached to same solution which confirms that the solution found was a global minimum.

Both ASA and GA are powerful search and optimization technique. Genetic Algorithms are mathematically less complex, and relatively simple and easy to code. The above application, however shows faster convergence rate of ASA as compared to GA.

Acknowledgements

The financial award provided by the National Science Foundation to support this research under grant number CMS 99-08177 is gratefully acknowledged.

References

Ananthraman, S., Garg, D., 1993a. Training backpropagation and CMAC neural networks for control of a SCARA robot. International Journal on Engineering Applications of Artificial Intelligence 6 (2), 105–115.

Ananthraman, S., Garg, D., 1993b. Neurocontrol of cooperative dual robot manipulators. Intelligent Control Systems, ASME Special Publication Number DSC 48, 57–65.

Bowman, M., Ingber, L., 1997. Canonical momenta of nonlinear combat. Proceedings of the 1997 Simulation Multi-Conference, San Diego, CA, USA.

Carretero, J.A., Nahon, M.A., Ma, O., 2001. Solving distance problems with concave bodies using simulated annealing. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Hawai, USA, pp. 1507–1512.

Chen, M., Zalzala, A., 1997. A genetic approach to motion planning of redundant mobile manipulator systems considering safety and configuration. Journal of Robotic Systems 14, 529–544.

Choi, S., Choi, Y., Kim, J., 1999. Optimal walking trajectory generation for a biped robot using genetic algorithm. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems 3, Kyongju, South Korea, 1456–1461.

Craig, J., 1989. Introduction to Robotics: Mechanics and Control, 2nd Edition. Addison-Wesley Publishing Company, Inc., Reading, MA.

Delingette, H., Herbert, M., Ikeuchi, K., 1992. Trajectory generation with curvature constraint based on energy minimization. Proceedings of the IEEE/RSJ International Workshop on Intelligent Robots and Systems, Osaka, Japan, pp. 206–211.

FlexGA^{TM}, 1998. User's Guide, FlexGA^{TM}, Version 1.0, Flexible Intelligence Group, LLC.

Garg, D., 1992. Supervisory control for coordination of multiple robots. The International Journal of Systems Science 23 (10), 1703–1716.

Garg, D., 2000. Coordinated control of multiple robots. Proceedings of the Japan–USA Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering, Ho Chi Minh City, Vietnam, pp. 69–84.

Garg, D., Anantharaman, S., Prabhu, S., 1999. Neural network applications. In: John G.W. (Ed.), Wiley Encyclopedia of Electrical and Electronics Engineering, Vol. 14. John Wiley, New York, NY, pp. 255–265.

Garg, D., Ruengcharungpong, C., 1992. Force balance and energy optimization in cooperating manipulation. Proceedings of the 23rd Annual Pittsburgh Modeling and Simulation Conference, Pittsburgh, PA, USA, Vol. 23, Pt. 4, pp. 2017–2024.

Goldberg, D., 1989. Genetic Algorithms in Search, Optimization & Machine Learning. Addison-Wesley Longman, Inc., Reading, MA, New York.

Han, W., Baek, S., Kuc, T., 1997. GA based on-line path planning of mobile robots playing soccer games. Proceedings of the 40th Midwest Symposium on Circuits and Systems, Sacramento, CA, USA, Vol. 1, pp. 522–525

Hirakawa, A., Kawamura, A., 1996. Proposal of trajectory generation for redundant manipulators using variational approach applied to minimization of consumed electrical energy. Proceedings of the Fourth International Workshop on Advanced Motion Control, Tsukuba, Japan, Part 2, pp. 687–692.

Ingber, L., 1996. Adaptive simulated annealing (ASA): lessons learned. Controls and Cybernetics 25, 33–54.

Ingber, L., Mondescu, R.P., 2001. Optimization of trading physics models of markets. http://www.ingber.com/markets01_optim_trading.pdf.

Ivanescu, M., Bizdoaca, N., 2000. A two level hierarchical fuzzy controller for hyperredundant cooperative robots. Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, CA, USA, pp. 3170–3175.

Kwon, W., Lee, B., 1998. A new optimal force distribution scheme of multiple cooperating robots using dual method. Journal of Intelligent and Robotic Systems 21, 301–326.

Makino, T., Yokoi, H., Kakazu, Y., 1999. Development of a motion planning system for an agricultural mobile robot. Proceedings of the 38th Annual Conference of the SICE, Morioka, Japan, 1999, pp. 959–962.

Monteiro, D., Madrid, M., 1999. Planning of robot trajectories with genetic algorithms. Proceedings of the First Workshop on Robot Motion and Control, Kiekrz, Poland, pp. 223–228.

Murphy, S., Wen, J., Saridis, G., 1991. Simulation of cooperating robot manipulators on a mobile platform. IEEE Transactions on Robotics and Automation 7 (4), 468–478.

Nagchaudhuri, A., Garg, D., 1992. Load sharing and internal forces in multiple cooperating manipulators: a new perspective. Proceedings of the 23rd Annual Pittsburgh Modeling and Simulation Conference, Pittsburgh, PA, USA, Vol. 23, Pt. 4, pp. 2025–2032.

Painton, L., Campbell, J., 1995. Genetic algorithms in optimization of system reliability. IEEE Transactions on Reliability 44 (2), 172–178.

Park, M.G., Jeon, J.H., Lee, M.C., 2001. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. Proceedings of IEEE International Symposium on Industrial Electronics 3, Pusan, Korea, 1530–1535.

Pin, F., Culioli, J., 1992. Optimal positioning of combined mobile platform-manipulator systems for material handling tasks. Journal of Intelligent Robotic System, Theory and Application 6, 165–182.

Pires, E., Machado, J., 1999. A trajectory planner for manipulators using genetic algorithms. Proceedings of the IEEE International Symposium on Assembly and Task Planning, Porto, Portugal, pp. 163–168.

Prabhu, S., Garg, D., 1995. Fuzzy reinforcement compliance control for robotic assembly. Proceedings of the 10th International Symposium on Intelligent Control, Monterey, CA, USA, pp. 623–628.

Prabhu, S., Garg, D., 1996. Artificial neural network based robot control: an overview. Journal of Intelligent and Robotic Systems 15 (4), 333–365.

Prabhu, S., Garg, D., 1998. Fuzzy logic based reinforcement learning of admittance control for automated robotic manufacturing. International Journal on Engineering Applications of Artificial Intelligence 11, 7–23.

Sexton, R., Gupta, J., 2000. Comparative evaluation of genetic algorithm and backpropagation for training neural networks. Information Sciences 129, 45–59.

Watanabe, K., Kiguchi, K., Izumi, K., Kunitake, Y., 1999. Path planning for an omnidirectional manipulator by evolutionary computation. Proceedings of the Third International Conference on Knowledge Based Intelligent Information Systems, Adelaide, Australia, pp. 135–140.

Woodard, S., Garg, D., 1999. A numerical optimization approach for tuning fuzzy logic controllers. IEEE Transactions on Systems, Man and Cybernetics, Part B 29 (4), 565–569.

Yoshikawa, T., 1990. Foundations of Robotics: Analysis and Control. MIT Press, Cambridge, MA.

Zhang, J., Ferch, M., 1998. Rapid on line learning of compliant motion for two-arm coordination. Proceedings of the IEEE International Conference on Robotics & Automation, Leuven, Belgium, pp. 1853–1858.

Devendra P. Garg received his Bachelor of Science degree from Agra University, Bachelor of Engineering degree from University of Roorkee, M.S. in Mechanical Engineering degree from the University of Wisconsin, and a Ph.D. degree from New York University. Dr. Garg taught at the University of Roorkee as a Lecturer and Reader in Mechanical Engineering; as an Instructor at New York University; and as an Assistant Professor, and Associate Professor of Mechanical Engineering at the Massachusetts Institute of Technology. He was also the Chairman of Engineering Projects Laboratory, while at MIT. Currently, he is a Professor of Mechanical Engineering at Duke University, and Director of the Robotics and Manufacturing Automation Laboratory in the Department. While on leave from Duke University since July 1, 1992, Dr. Garg served until August 31, 1998, as Director of the Dynamic Systems and Control Program in the Division of Civil and Mechanical Systems at the National Science Foundation, Arlington, VA. Professor Garg is a recipient of the TCM award and the New York University's Founder's day award for outstanding scholastic achievement. He is an active member of the American Society of Mechanical Engineers (ASME), and Past Chairman of the ASME's Dynamic Systems and Control Division (DSCD). He has guest-editored two Special Issues of ASME Transactions. Dr. Garg has served as Chairman of the Advisory Panel, and Chairman of the Honors Committee of the DSCD, ASME. In addition, he has served as a voting member of the ASME National Nominating Committee and Chairman of the Nominating Committee of the ASME Systems and Design Group (SDG), for which he has also served as Vice Chairman. While serving at NSF, Dr. Garg was appointed as the first Chairman of the Strategic Planning and Evaluation Committee (SPEC) of the Directorate for Engineering. Dr. Garg has received the Dedicated Service Award, and the DSCD Leadership Award from the American Society of Mechanical Engineers, and the Outstanding Work Performance Award, and the Cooperative Team Effort Award from the National Science Foundation. His research interests include control system synthesis, vehicle dynamics, robotics and automated manufacturing, and application of control theory to socio-economic systems. He is an author of two books and numerous research publications in technical journals in USA and abroad.

Manish Kumar graduated from the Indian Institute of Technology, Kharagpur, India in 1998. There he earned a Bachelors of Technology degree with Honors in Mechanical Engineering. Following his under-

graduate studies, Manish worked as a design engineer for two years for the Hindustan Petroleum Corporation Ltd. In 2000, he joined Mechanical Engineering and Materials Science Department of Duke University as a graduate student where he earned his Masters of Science degree in May 2002. He is currently working towards his Ph.D. degree in Mechanical Engineering. Manish's research interests lie mainly in robotics, controls and dynamic systems. His current research effort includes sensor fusion where sensory information from various sensing devices, such as the force-torque and vision sensors etc are used to efficiently control and achieve cooperative manipulation of multiple robots.