2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009; Coimbatore; 9 - 11 December 2009;
Category number CFP0995H; Code 79534; Article number 5393780, Pages 1156-1161
DOI- http://dx.doi.org/10.1109/NABIC.2009.5393780

An Appropriate Tool for Optimizing the Workspace of 3R Robot Manipulator

S.Panda
Manufacturing Science & Engineering
VSS University of Technology
Burla,India
geetikasalwan@gmail.com

D.Mishra
Manufacturing Science & Engineering
VSS University of Technology
Burla,India
dmishrauce@gmail.com

B.B.Biswal
Training & Placement
National Institute of Technology
Rourkela,India
bibhuti.biswal@gmail.com

***Abstract*—Robotic manipulators with three-revolute (3R) positional configurations are very common in the industrial robots (IRs). The capability of a robot largely depends on the workspace (WS) of the manipulator apart from other parameters. With the constraints in mind the optimization of the workspace is of prime importance in designing the manipulator. The present work aims at obtaining an optimal design of manipulators with three-revolute joints. The optimization problem is formulated considering the workspace volume as the objective function, while constraints are imposed to control the total area. Subsequently the problem is solved using Sequential quadratic programming (SQP, fminmax, goal attainment, constrained non linear minimization) and genetic algorithms (GAs) and a comparision is made. The four different optimization techniques were used to solve numerical example imposing same condition to demonstrate the efficiency of the optimization processes. Numerical example is presented to validate the proposed methodology.**

***Keywords*-robot workspace; genetic algorithm; sequential quadrataic programming, goal attainment; constrained non linear minimization**

I. INTRODUCTION

A robot manipulator structure can be subdivided into a regional structure and orientation structure. The regional structure consists of the arm, which moves the end-effectors to a desired position in the workspace of the robot manipulator. The orientation structure comprised of the links that, rotates the end effectors to the desired orientation in the workspace. In this study, the regional structure of the robot manipulators is examined rather then the orientation structure. The workspace, also called work volume or work envelope of a manipulator, is the volume of space, which the end effectors of the manipulator can reach. The size and shape of the workspace depends on the coordinate geometry of the robot arm, and also on the number of degrees of freedom. Some workspaces are quite flat, confined almost entirely to one horizontal plane. Others are cylindrical; still others are spherical. Some workspaces have very complicated shapes. When choosing a robot arm for a certain industrial purpose, it is important that the workspace is large enough to encompass all the points that the robot arm will need to reach, but it is wasteful to use a robot arm with a workspace much larger than necessary. The workspace of a robot is an important criterion in comparing manipulator geometries. Optimization of workspace plays a very important role in ranking of workability and flexibility. There is a close relationship between the kinematics performance and design of robot manipulators. Because of this, several kinematics-related criteria have been suggested for designing a well-conditioned robot manipulator that has optimum workspace. A fundamental characteristic that must be taken into account in the dimensional design of robotic manipulators is the volume of their workspace. The workspace volume should be computed with high accuracy as it influences the manipulator's dimensional design, its positioning in the work environment, and its dexterity to execute tasks. In this paper the design of manipulators with three-revolute joints (3R) is reformulated as an optimization problem that takes into account the characteristics of the workspace. The computation of workspace volume by the approach proposed here requires the knowledge of boundary condition. To calculate the area, one must know the boundary of the cross-section, which can be known from the joint parameter. This research proposes an algebraic formulation to estimate the cross section area, and workspace volume. This paper also discusses the results obtained by using four different numerical techniques: sequential quadratic programming (SQP) methods (minmax optimization, goal attainment, constrained non linear minimization) and genetic algorithms (GAs). The presence of voids and singularities and the discontinuous generation of the envelope greatly increase the complexity of the calculation of the algebraic formulation for a correct mathematical model of the workspace volume. Moreover, the objective function presents several local maxima and is extremely nonlinear. These factors greatly increase the difficulties involved in the optimization process, justifying the use of different optimization techniques to validate the results. Several investigations have focused on the properties of the workspace of open chain robotics with the purpose of emphasizing its geometric and kinematic characteristics, and to devise analytical algorithms and procedures for its design. The manipulator workspace is defined as the region of reachable points by a reference point on the extremity of a manipulator chain, according to Gupta and Roth [1]. Ceccarelli [2] presented an algebraic formulation to determine the workspace of revolute manipulators. In that paper the workspace boundary is obtained from the envelope of a torus family which is traced by the parallel circles cut in the boundary of a revolving hyper-ring. The formulation is a function of the dimensional parameters in the manipulator chain and specifically of the last revolute joint angle, only. The workspace mathematical developed by Ceccarelli is of crucial importance, however the manipulators optimal design is not considered. The formulation developed by Tsai and Soni [3] is used in this work to obtain the equation of cross sectional area and volume. Lanni et al. [4] investigated and solved the design of manipulators modelled as an optimization problem that takes into account the characteristics of the workspace. They applied two different numerical techniques; the first using sequential quadratic programming (SQP) and the second involving a random search technique (simulated annealing). The above-mentioned methodology cannot be applied to calculate the workspace volume in case there is a ring void; however it was an important initial study. Abdel-Malek et al. [5] proposed a generic formulation to determine voids in the workspace of serial manipulators. Wenger [6] demonstrated that it is possible to consider a manipulator's execution of non-singular changing posture motions in the design stage. Some researches have focused on determining the workspace boundary and on detecting the presence of voids and singularities in the workspace. Saramago et al. [7] proposed a form of characterizing the workspace boundary, formulating a general analytical condition to deduce the existence of cusp points at the internal and external boundaries of the workspace. Ceccarelli and Lanni [8] presented a suitable formulation for the workspace that can be used in the design of manipulators, which was formulated as a multi-objective optimization problem using the workspace volume and robot dimensions as objective functions. Bergamaschi et al.[9] discussed an optimum design of

In the present paper, the optimization problem is defined in such a way that the objective function is the workspace volume. Constraints have been included to restrict manipulator sizes to practical values. The methodology is tested by the comparison of the results obtained with four different optimization techniques confirming the validity of using the formulation for the workspace.

II. FORMULATION FOR OPTIMAL DESIGN

A general open chain 3R manipulator with three revolute joint is sketched in Fig.1, in which the design parameters are represented using the Denavit and Hartenberg (D-H) notation [10] for the link size as $a_1, a_2, a_3, s_2, s_3, \alpha_1, \alpha 2$, ($s_1$ is not meaningful since it shifts the workspace up and down). The 3R robot, as shown in Fig. 1 with its kinematic parameters is found to be capable of having a maximum working space as discussed in [11].

In the workspace synthesis problem, where it is required to enclose a prescribed accessible region or volume to arrive

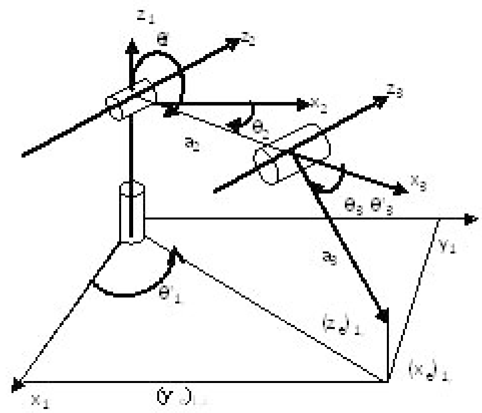

Figure 1. The workspace parameter of 3R manipulator

at the specific base location, link-length values and the rotational displacements at each of the revolute joints of the 3R robot are required. Let $(x_e, y_e, z_e)_{j,i}$ represent the coordinates of the given i^{th} position of the end point of the robot arm in reference frame $\{X_i, Y_j, Z_j\}$ which is attached to joint j. In Fig. 1 for a given location of robot end $(x_e, y_e, z_e)_{i,j}$, the joint displacements $(\theta_1')_i, (\theta_2')_i$, and $(\theta_3')_i$ can be calculated from

$$(\theta_1^{'})_i = \tan^{-1}(y_e/x_e)_{1,i} \tag{1}$$

$$(\theta_2^{'})_i = \cos^{-1}\frac{(z_e)_{1,i} - s_1}{l_i} - \cos^{-1}\frac{l_1^2 + (a_2^2 - a_3^2)}{2a_2 l_i} \tag{2}$$

$$(\theta_3^{'}) = \cos^{-1}\frac{l_i^2 - (a_2^2 + a_3^2)}{2a_2 a_3} \tag{3}$$

where, $l_i = \{(x_e)_{1,i}^2 + (y_e)_{1,i}^2 + ((z_e)_{1,i} - s_1)^2\}^{1/2}$

The cross-section area A and the solid of revolution volume V (generated by A) of the workspace as derived by Tsai and Soni [10] are given below.

$$A = [\cos(\theta_3^{'})_{\min} - \cos(\theta_3^{'})_{\max}](\Delta\theta_2^{'})(a_2)^2 \tag{4}$$

$$\begin{aligned} X_{cg} = {} & \Delta\theta_2^{'}(a_2)^3[\sin\theta_2^*(\cos(\theta_3^{'})_{\min} - \cos(\theta_3^{'})_{\max}) \\ & - \sin\theta_2^*[\sin^2(\theta_3^{'})_{\min} - \sin^2(\theta_3^{'})_{\min}]/2 \\ & - \cos\theta_2^*[\sin 2(\theta_3^{'})_{\min} - \sin 2(\theta_3^{'})_{\max} \\ & -[\sin 2(\theta_3^{'})_{\min} - \sin 2(\theta_3^{'})_{\max}]/A \end{aligned} \tag{5}$$

$$V = (\Delta\theta_1^{'})X_{cg}A \tag{6}$$

where,

$$\Delta\theta_1^{'} = (\theta_1^{'})_{\max} - (\theta_1^{'})_{\min}(rad),$$

$$\Delta\theta_2^{'} = (\theta_2^{'})_{\max} - (\theta_2^{'})_{\min}(rad), and$$

$$\theta_2^* = [(\theta_2^{'})_{\max} + (\theta_2^{'})_{\min}]/2(rad).$$

X_{cg} is the centroid.

$$0.01 \leq a_i \leq a_i^u \quad (7)$$

Where, the maximum value of is assumed during the optimization process. Suitable optimization techniques (SQP-minmax, goal attainment, constrained nonlinear minimization, GA etc.) can be used to find out the optimum value of volume and the corresponding values of $(\theta_1^{'})_{min}$,$(\theta_1^{'})_{max}$ $(\theta_2^{'})_{min}$,$(\theta_2^{'})_{max}$,$(\theta_3^{'})_{min}$,$(\theta_3^{'})_{max}$. This provides the necessary design data for synthesized 3R robot. These algorithms minimize the worst-case value of a set of multivariable functions, starting at an initial estimate, for numerical optimization.

III. REVISION OF OPTIMIZATION METHODS

A brief review of the optimization methods used is presented below.

A. Sequential Quadratic Programming

The basic concept of SQP is to model a nonlinear programming problem by using an iterative algorithm in which, at a current iterate x_k, the step to the next iterate is obtained through information generated solving a quadratic sub problem. This sub-problem is assumed to reflect, in some way, the local properties of the original problem. The main idea is the formulation of a sub-problem based on a quadratic approximation of the Lagrangian function [12]. The nonlinear programming problem to be solved is

$$\min_x F_c : R^n \rightarrow R \quad (8)$$

subject to

$$h_i(x) = 0, i = 1, 2, 3, ..., m, g_i(x) = 0, i = 1, 2, 3..., p \quad (9)$$

in which $F_c(x)$ is the objective function; m is the number of the equality constraints h(x);p is the number of inequality constraints g(x) and x is the vector containing the design parameters a_2, a_3, x_e, y_e, z_e (dimensional parameters of the robot manipulator and end point). To take nonlinearities in the constraints into account while maintaining the linearity of the constraints in the sub problem, the SQP method uses a quadratic model of Lagrangian function Λ as the objective. Let the Lagrangian function given by

$$\Lambda(\mathrm{x}_k, u_i, \mathrm{v}_i) = F_c(\mathrm{x}_k) + \sum_{i=1}^{m} u_i h_i\ (\mathrm{x}_k) + \sum_{i=1}^{p} \mathrm{v}_i g_i(\mathrm{x}_k) \quad (10)$$

$$\nabla_x \Lambda(x_k, u_i, v_i) = \nabla F_c(x_k) + \sum_{i=1}^{m} u_i \nabla h_i(x_k) + \sum_{i=1}^{p} v_i \nabla g_i\,(x_k) \quad (11)$$

In above equations, u_i and v_i represent the Lagrangian multipliers, ∇F_c is the gradient of objective function F_c at x_k; ∇h_i and ∇g_i are the Jacobian matrices of the constraints. According to Nocedal and Wright [12], the SQP framework can be extended easily to the nonlinear problem (8) and (9), in this case, to model the problem it is necessary linearize both the inequality and equality constraints to obtain

$$\min_{s_x} \left\{ F_c(x_k) + \nabla F_c(x_k)^T\ s_x + \frac{1}{2} s_x^T H_{ek} s_x \right\} \quad (12)$$

$$\Delta h_i(x_k)^T s_x + h_i(x_k) = 0, i = 1, 2, ., m,$$

$$\Delta g_i(x_k)^T s_x + h_i(x_k) = 0, i = 1, 2, ., m, \quad (13)$$

where x_k is the vector containing the design parameters at iteration k, T denotes the transpose operation; $s_x = x - x_k$ is the search direction; is taken as the Hessian matrix of the Lagrangian function at x_k . The numerical procedure starts with an initial guess of the manipulator chain solution and, during each iteration k, the quadratic programming problem is solved to provide a search direction s_x. The Solution s_x can be used to generate a new iterate x_{k+1}, for some selection of the step-length parameter ψ_k, as

$$x_{k+1} = x_k + \psi_k s_x \quad (14)$$

To continue to the next iteration, a new estimate for the Lagrangian multipliers is necessary a usual approach is to use optimal multipliers of the quadratic sub-problem. Let us denote by u_{qp} and v_{qp} such multipliers. Thus, the updated multipliers u_{k+1} and v_{k+1} are obtained as follows

$$u_{k+1} = u_k + \psi_k s_u; s_u = u_{qp} - u_k$$

$$v_{k+1} = v_k + \psi_k s_v; s_v = v_{qp} - v_k \quad (15)$$

Summarizing, a SQP technique solves the optimization problem defined by Eqs. (8) and (9) by computing the search direction by means of Eqs. (10)-(15).

1) Minmax Optimization: minimax algorithm involves minimizing the worst-case value of a set of multivariate functions, possibly subject to linear and non linear constraints. fminimax internally reformulates the minimax problem into an equivalent nonlinear linear programming problem by appending additional (reformulation) constraints to the existing constraints and then optimize it using SQP method. The steps to synthesize a desired 3R robot using fminmax algorithm is presented below.

Step 1 (i) Guess the initial value of x_i. x_i= [a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. (ii) Find($\theta_1^{'}$) using equation 1. let $f(x_i)$ =($\theta_1^{'}$), x_{i+1} = x_i+ ψ_i ,find $f(x_{i+1})$, if $f(x_{i+1}) > f(x_i)$ then store $f(x_{i+1})$ i.e. maximum for ($\theta_1^{'}$) ,$(\theta_1^{'})_{max}$, else x_i = x_{i+1} repeat step (ii), Repeat the same procedure for ($\theta_2^{'}$) & ($\theta_3^{'}$) till $(\theta_2^{'})_{max}$ & $(\theta_3^{'})_{max}$ are determined.

Step 2 (i) Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. (ii)

Find $(\theta_1^{'})$ using equation 1. let $f(x_i)$ = $(\theta_1^{'})$, x_{i+1} = $x_i + \psi_i\delta_i$,find $f(x_{i+1})$ if $f(x_{i+1}) < f(x_i)$ then store $f(x_{i+1})$ i.e. minimum for $(\theta_1^{'})$, $(\theta_1^{'})_{min}$ else x_i = x_{i+1} repeat step (ii) Repeat the same procedure for $(\theta_2^{'})$ & $(\theta_3^{'})$ till $(\theta_2^{'})_{min}$ & $(\theta_3^{'})_{min}$ are determined.

Step 3 Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location.

Step 4 Determine the cross section area A using eq. (4) Determine the X_{cg} using eq. (5)

Step 5 Determine volume $V(x_i)$ using eq. (6) and compute constraint using eq. (7).

Step 6 x_{i+1} = x_i+$\psi_i\delta_i$ find $V(x_{i+1})$

Step 7 if $V(x_{i+1})$-$V(x_i)$< e (tolerance) then find optimum parameter for robot manipulator. else x_i = x_{i+1} repeat from step 4 till convergence.

2) Goal Attainment: The goal attainment method involves expressing a set of design goals, $F^* = \{F_1^*, F_2^*, F_3^* \ldots\}$ which is associated with a set of objectives, $F(x) = \{F_1(x), F_2(x), F_3(x) \ldots, F_m(x)\}$ the problem formulation allows the objectives to be under- or over-achieved, enabling the designer to be relatively imprecise about initial design goals. The relative degree of under- or over- achievement of the goals is controlled by a vector of weighting coefficients, $w = \{w_1, w_2, w_3 \ldots, w_m\}$ and is expressed as a standard optimization problem using the following formulation.

$$\underset{\gamma \in R, x \in \Omega}{minimize}\ \gamma \tag{16}$$

Such that $F_i(x) - \omega_i \leq F_i^*$, i = 1 2 3 4$\ldots m$

The term introduces an element of slackness into the problem, which otherwise imposes that the goals be rigidly met. The weighting vector, w, enables the designer to express a measure of the relative tradeoffs between the objectives. For instance, setting the weighting vector w equal to the initial goals indicates that the same percentage under or over attainment of the goals, F^* is achieved. Hard constraints can be incorporated into the design by setting a particular weighting factor to zero (i.e. $w_i = 0$). The goal attainment method provides a convenient intuitive interpretation of the design problem, which is a solvable using standard optimization procedure.

The goal attainment problem involves reducing the value of a linear or nonlinear vector function to attain the goal values given in a goal vector. The relative importance of the goals is indicated using a weight vector. The goal attainment problem may also be subject to linear and nonlinear constraints. The followings are the steps to synthesize a desired 3R robot using goal attainment algorithm.

Step 1 (i) Guess the initial value of x_i. x_i= [a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. Set the goal value =F_1^* for $(\theta_1^{'})$ assign weight w_i (ii) Find$(\theta_1^{'})$ using equation 1. let $f(x_i)$ =$(\theta_1^{'})$, x_{i+1} = x_i+ $\psi_i\delta_i$, find $f(x_{i+1})$, if $f(x_{i+1})$-$w_1^* f(x_i)$ > F_i^* then store $f(x_{i+1})$ i.e. maximum for $(\theta_1^{'})$,$(\theta_1^{'})_{max}$, else x_i = x_{i+1} repeat step (ii), Repeat the same procedure for $(\theta_2^{'})$ & $(\theta_3^{'})$ till $(\theta_2^{'})_{max}$ & $(\theta_3^{'})_{max}$ are determined.

Step 2 (i) Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. Set the goal value =F_1^* for $(\theta_1^{'})$ assign weight w_i (ii) Find $(\theta_1^{'})$ using equation 1. let $f(x_i)$ = $(\theta_1^{'})$, x_{i+1} = $x_i + \psi_i\delta_i$, find $f(x_{i+1})$
if $f(x_{i+1})$- $w_1^* f(x_i)$ < F_i^* then store $f(x_{i+1})$ i.e. minimum for $(\theta_1^{'})$,$(\theta_1^{'})_{min}$, else x_i = x_{i+1} repeat step (ii), Repeat the same procedure for $(\theta_2^{'})$ & $(\theta_3^{'})$ till $(\theta_2^{'})_{max}$ & $(\theta_3^{'})_{max}$ are determined.

Step 3 Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location.Set the goal value F_1^* for $(\theta_1^{'})$, assign weight w_i.

Step 4 Determine the cross section area A using eq. (4) Determine the X_{cg} using eq. (5)

Step 5 Determine volume $V(x_i)$ using eq. (6) and compute constraint using eq. (7).

Step 6 x_{i+1} = x_i+$\psi_i\delta_i$ find $V(x_{i+1})$

Step 7 if $V(x_{i+1})$-$w(x_i^*)V(x_i)$< F_1^* then find optimum parameter for robot manipulator. else x_i = x_{i+1} repeat from step 4 till convergence.

3) Constrained Nonlinear Minimization: SQP doesn't distinguish between linear and nonlinear constraints but constrained non linear minimization (fmincon) algorithm does the same. The constraints are modelled by functions that return a vector. Each entry in that vector stands for a single constraint. fmincon only handles real variables. fmincon is a gradient-based method that is designed to work on problems where the objective and constraint functions are both continuous and have continuous first derivatives fmincon might only give local solutions. When the problem is infeasible, fmincon attempts to minimize the maximum constraint value. The objective function and constraint function must be real-valued i.e. they cannot return complex values. The followings are the steps to synthesize a desired 3R robot using goal attainment algorithm.

Step 1 (i) Guess the initial value of x_i. x_i= [a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. (ii) Find$(\theta_1^{'})$ using equation 1. let $f(x_i)$ =$(\theta_1^{'})$, x_{i+1} = x_i+ s_i, where s_i is the search direction. s_i= $x - x_i$, find $f(x_{i+1})$, if $f(x_{i+1}) > f(x_i)$ then store $f(x_{i+1})$ i.e. maximum for $(\theta_1^{'})$,$(\theta_1^{'})_{max}$, else x_i = x_{i+1} repeat step (ii), Repeat the same procedure for $(\theta_2^{'})$ & $(\theta_3^{'})$ till $(\theta_2^{'})_{max}$ & $(\theta_3^{'})_{max}$ are determined.

Step 2 (i) Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location. (ii)

Find $(\theta_1^{'})$ using equation 1. let $f(x_i)$ = $(\theta_1^{'})$, x_{i+1} = x_i + $\psi_i s_i$, find $f(x_{i+1})$, if $f(x_{i+1})$ < $f(x_i)$ then store $f(x_{i+1})$ i.e. minimum for $(\theta_1^{'})$, $(\theta_1^{'})_{min}$ else x_i = x_{i+1} repeat step (ii) Repeat the same procedure for $(\theta_2^{'})$ & $(\theta_3^{'})$ till $(\theta_2^{'})_{max}$ & $(\theta_3^{'})_{max}$ are determined.

Step 3 Guess the initial value of x_i. x_i=[a_2 a_3 x_e y_e z_e] manipulator parameter and end point location.

Step 4 Determine the cross section area A using eq. (4) Determine the X_{cg} using eq. (5)

Step 5 Determine volume $V(x_i)$ using eq. (6) and compute constraint using eq. (7).

Step 6 x_{i+1} = x_i+$\psi_i s_i$ find $V(x_{i+1})$

Step 7 if $V(x_{i+1})$-$V(x_i)$< e (tolerance) then find optimum parameter for robot manipulator. else x_i = x_{i+1} repeat from step 4 till convergence.

B. Genetic Algorithm

Genetic algorithm (GA) is an iterative technique based on random search with adaptations of the search minimum in coordinate directions. It can be useful to overpass small variations in the objective function by using a probabilistic criterion in order to avoid stopping the procedure at a local minimum. This algorithm allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness", i.e., that minimizes the cost function. A simple genetic algorithm performs basically three operations: selection, crossing and mutation. As in natural genetics, evolution goes hand-in-hand with variety; hence, it is important for individuals to display different degrees of adaptation to their environment. This means that the initial population covers the search space in the best possible way. The initial population formed of N individuals is usually generated in an aleatory manner or through some heuristic process, for example, can be written as:

$$x_i = x_i^l \frac{x_i^u - x_i^l}{2^m - 1} \sum_{j=0}^{m-1} b_{ij} 2^j \qquad (17)$$

where b_{ij} represents the bit string segment of length m for encoding the i^{th} element of the objective variable vector x; x^l and x^u are the side constraints. In the Selection operation, a temporary population of N individuals is generated considering the proportional probability of each individual with respect to its relative adaptability in the population. The individuals presenting low adaptability will be more prone to disappear. There exist a great number of selection operators. In the roulette wheel selection, using the fitness value F_{ci} of all strings, the probability of selecting the i^{th} string is

$$p_i = F_{ci} / \sum_{j=1}^{N} F_{cj} \qquad (18)$$

The Crossover operator works by selecting two individuals that will exchange genetic material. It is also an aleatory process which occurs with a probability established by the user. As in nature, Mutation is a rarely occurring event whose purpose is to ensure that important genetic material is not irrevocably lost. Thus, similarly to the evolution process in the search for the most adapted solution along successive generations, the optimization procedure improves these solutions until the optimal one is identified. The process ends when the maximum number of generations is reached, or, according to the stagnation concept, when an improvement in the population is observed after several serial iterations.

Reproduction, crossover, and mutation are three operatorsin GA, which are simple and straightforward. Reproduction operator selects good strings and crossover operator recombines good substrings from two good strings together to form a better substring. Mutation operator alters a string locally to create a better string. A termination criterion is then checked. If the termination criterion is not satisfied, the population of solutions is modified by three main operators and a new population is created. The generation counter is incremented to indicate that one generation of GA is completed.

IV. RESULTS AND DISCUSSION

One of the most complicated problems in manipulator kinematics is to find the optimal geometry. Mathematical equations that describe the behaviour of robot kinematics are nonlinear; also have plenty of terms in general. The complexity of the optimal design problem urges to develop fast prototyping, which allows robot designers to expose structural defects of mechanism by studying the behaviour of their prototypes instead of analysing troublesome mathematical models. Modern mathematics does not posses generic techniques for having closed-form solutions to nonlinear equations. Hence iterative methods are still used for solving complicated system. In this work, a minimax algorithm, goal attainment algorithm, constrained non linear minimization algorithm is used. These algorithms minimize the worst-case value of a set of multivariable functions, starting at an initial estimate, for numerical optimization. This optimization is generally referred to as the minimax problem. It uses a sequential quadratic programming (SQP) method. The genetic algorithm is also used to optimize the workspace volume.

It is important to mention here that a manipulator is always designed for all kind of initial conditions. Therefore the optimization of joint parameters for the manipulator has been obtained by use of SQP algorithms and GA. The control parameters and the initial guess values are selected on trial and error method for which these algorithms provide optimal solution with lesser numbers of function evaluation. The control parameters used for the analysis of different algorithms are given in Table I. It has been observed from the trial runs of the programme that on increasing the cross over and mutation probabilities the value of objective function

Table I
CONTROL PARAMETERS

fminmax	
Function tolerance	1.00E-06
Constraint tolerance	1.00E-06
Multi-objective goal attainment	
Function tolerance	1.00E-06
Constraint tolerance	1.00E-06
Goal, weight	200 , 0.5
Constrained non linear minimization	
Function tolerance	1.00E-06
Constraint tolerance	1.00E-06
Genetic algorithms	
Population size	100
Number of generations	1000
Cross over probability	0.75
Mutation probability	0.195

Table II
RESULTS OBTAINED THROUGH DIFFERENT ALGORITHMS

Algorithm $\rightarrow$	fminmax	GAt	CNM	GA
$(\theta_1^{'})_{max}$	0.382	0.389	0.374	0.730
$(\theta_1^{'})_{min}$	-0.399	-0.406	-0.392	-0.697
$(\theta_2^{'})_{max}$	0.143	0.158	0.127	-1.688
$(\theta_2^{'})_{min}$	-1.431	-1.426	-1.436	0.0083
$(\theta_3^{'})_{max}$	1.539	1.555	1.523	1.246
$(\theta_3^{'})_{min}$	0.0	0.0	0.0	0.4065
Workspace(u^3)Max	268.3166	205.4048	253.945	265.648
Standard Deviation	11.085	9.869	10.562	0.059
$a_2 = a_3(u)$	1.193	2.728	0.059	0.94

decreases. With cross over and mutation probabilities as 0.75 and 0.175 better results are obtained with lesser numbers of function evaluation. Taking the bounds and initial start values, computer programmes are developed by using optimtool and gatool of Matlab, and the simulations are run to find the optimal solutions.

In order to prove the soundness of the proposed optimization design procedures numerical examples have been reported. The workspace volume optimization is achieved by means of four algorithms.

The initial values of design parameters and end position of manipulator are a_1=0.5u, a_2=0.5u and $(x_e y_e z_e)$ as $(1, 2, 1)$. The results obtained by fminmax, Goal Attainment(GAt), Constrained Nonlinear Minimization(CNM) and genetic Algorithm (GA) are shown in Table II.

V. CONCLUSIONS

The mathematical modelling to determine the workspace volume of 3R manipulators was obtained through an algebraic formulation that characterizes the workspace and cross sectional area of workspace. The efficiencies of the four different algorithms to demonstrate the design process are discussed. It is observed that the optimum configuration of workspace depends mainly on joint displacement angles i.e. θ_1, θ_2, and θ_3. The metaheuristics (i.e. GA) produced better result as compared to the SQP techniques. however, the it requires more number of generations and greater computational time. The three SQP techniques yield close values of objective function and are sensitive to initial value due to the presence of local minima. These facts can be observed through the high values for the standard deviation as shown in Table II. This property of SQP can be used to obtain a final solution when combined with metaheuristics. In future studies hybrid methodologies of SQP and metaheuristics can be developed to optimize the volume of 3R manipulators.

REFERENCES

[1] K.C.Gupta and B.Roth, *Design Considerations for Manipulator Workspace*, ASME Journal of Mechanical Design, vol. 104, p. 704-711, 1982.

[2] M.Ceccarelli, *A Formulation for the Workspace Boundary of General N-revolute Manipulators*, Mechanism and Machine Theory, vol. 31, no. 5, p. 637-646, 1996.

[3] Y.C.Tsai and A.H.Soni, *Workspace Synthesis of 3R, 4R, 5R and 6R Robots* Mechanism and Machine Theory, vol. 20, no. 6, p. 555-563, 1985.

[4] C.Lanni,S.F.P.Saramago,M.Ceccarelli, *Optimal Design of 3R Manipulators using Classical Techniques and Simulated Annealing*, Revista Brasileira de Cincias Mecnicas, vol. 24, no. 4, p. 293-301, 2002.

[5] K.Abdel-Malek,H.-J.Yeh,S.Othman, *Understanding Voids in the Workspace of Serial Robot Manipulators*, Proceedings of Design Engineering Technical Conferences, ASME, Baltimore, MD, p. 1-8, 2000.

[6] P.Wenger, *Some Guidelines for the Kinematic Design of New Manipulators*, Mechanism and Machine Theory, vol. 35, no. 3, p. 437-449, 2000.

[7] S.F.P.Saramago, E.Ottaviano and M.Ceccarelli, *A Characterization of the Workspace Boundary of Three-revolute Manipulators*, Proceedings of Design Engineering Technical Conferences, ASME, Montreal, Canada, vol. 1, p. 34342-34352, 2002.

[8] M.Ceccarelli and C.Lanni, *A Multi-objective Optimum Design of General 3R Manipulators for Prescribed Workspace Limits*, Mechanism and Machine Theory, vol. 39, p. 119-132, 2004.

[9] P.R.Bergamaschi, A.C.Nogueira and S.F.P.Saramago, *An Optimal Design of 3R Manipulators taking into account Regular Workspace Boundary*, Proceedings of 18th International Congress of Mechanical Engineering, ABCM, So Paulo, Brasil, 2003.

[10] J.Denavit and R.S.Hartenberg, *A kinematics Notation for Lower Pair Mechanism Based on Matrices*, Journal of Applied Mechanics, ASME, p. 215-221, 1955.

[11] Y.C.Tsai and A.H.Soni, *The Effect of Link-Parameters on the Working Space of General 3R Robot Arms*, Mechanism and Machine Theory, vol. 19, p. 9-16, 1984.

[12] J.Nocedal and S.J.Wright, *Numerical Optimization*, Springer Series in Operations Research, Springer, New York, 1999.