

2nd International Conference on Innovations in Automation and Mechatronics Engineering,  
ICIAME 2014

## Design Optimization of Serial Link Redundant Manipulator: an approach using global performance metric

Virendra Kumar<sup>a\*</sup>, Soumen Sen<sup>a</sup>, Shibendu S. Roy<sup>b</sup>, Chandan Har<sup>a</sup>, S.N.Shome<sup>a</sup>

<sup>a</sup>CSIR-Central Mechanical Engineering Research Institute, Durgapur-713209, India

<sup>b</sup>National Institute of Technology, Durgapur-713209, India

---

### Abstract

In this paper, an approach for design optimization of serial-link redundant manipulator is presented with exemplar optimizations on a 3-R planar and a 5-R spatial manipulator. The optimization is based on a global kinematic performance metric, which is chosen to be the Global Conditioning Index (GCI) based on condition number of manipulator Jacobian. The workspaces of 3-R planar and 5-R spatial manipulators are numerically computed using Monte Carlo technique in task space, which is utilized in evaluating the GCI. An algorithm is developed to compute the inverse kinematics of the 3-R planar and the 5-R spatial manipulators in finding the workspace and in turn the optimized solutions. Finally, design optimization problems are formulated for the 3-R and 5-R manipulator cases using the GCI as the objective function and solved (orientation of the end-effector is not included here). Further, genetic algorithm based method is used to solve the problem in validating the optimization results.

© 2014 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of the Organizing Committee of ICIAME 2014.

**Keywords:** Redundant Manipulator, Global Conditioning Index, Manipulator Workspace, Design Optimization, Monte-Carlo technique

---

### 1. Introduction

Redundant manipulators are chosen in many applications in view of obtaining improvements in performance. In redundant manipulator, the extra DOFs can be used to perform a number of secondary tasks in enhancing manipulator's ability [1-2]. Mostly design optimization of a manipulator is based on manipulator Jacobian-based

---

\*Corresponding author. Tel.: +91 3436452048 ; fax: +91 3432546745  
Email address: [vkumar@cmeri.res.in](mailto:vkumar@cmeri.res.in), [virendraug15@gmail.com](mailto:virendraug15@gmail.com)

kinematics performance metrics such as condition number, manipulability, dynamic manipulability etc. The ability to perform a secondary objective/task in addition to the primary task of reachability is sometimes defined as dexterity, characterized by a performance metric. One important problem of manipulator design is deciding the link lengths or their ratios. The design optimization computes the link lengths for an optimal performance in terms of a quantified metric or combination of metrics (in multi-objective formulation). Salisbury and Craig [3] used the condition number of the manipulator Jacobian matrix as an optimization criterion to obtain the dimensions for the fingers of the Stanford/JPL articulated hand. Klein and Blaho [4] used condition number to design a manipulator for isotropy at a working point for a fixed total arm length. Angeles in [5] proposed manipulator *Conditioning Index (CI)*. These metrics are only local measures. Gosselin and Angeles in [6] proposed a “Global Conditioning Index” (GCI) as a ratio of an integral of the reciprocal of condition number of Jacobian over the whole workspace to the volume of the manipulator workspace. This globally defined metric addresses optimization of the manipulator over the entire workspace.

This article proposes design optimization of a serial link manipulator based on the Global Conditioning Index (GCI) as defined in [6]. Computation of GCI requires knowledge of workspace volume of robot manipulator. Many methods exist in literatures which are based on analytical [7-8] or numerical methods [9-10] for determination of manipulator’s workspace. Numerical methods are mostly used for more than three degrees of freedom spatial manipulators to determine approximate boundary surfaces. Among numerical procedures, Monte Carlo method is an interesting choice. Most of the implementations of Monte Carlo method found in literature are in joint space.

The present article proposes a Monte Carlo implementation in the task space for workspace volume evaluations. With the definition of GCI as the objective function, a formulation for link length optimization of the manipulator is presented here. An algorithm is presented for numerical solution of the optimization, with evaluation of GCI in each solution iteration. Exemplar implementations for planar 3-R and spatial 5-R manipulator designs are depicted. To be noted that examples do not deal with end-effector orientation and thereby the unit mismatch in the computation of condition number of Jacobian is avoided.

The article is organized as in the following: Section-2 describes inverse kinematic solutions. Section-3 depicts the procedure for workspace evolution. Then kinematics performance metric is delineated in Section-4. The link length or link ratios optimization procedures for 3-DOF and 5-DOF manipulators are illustrated in Section-5 along with the results, followed by the conclusion and future work in the last section.

## 2. Inverse Kinematics

The forward kinematic equation of an  $n$ -DOF manipulator can be written as

$$x = f(q) \quad (1)$$

Where,  $x \in R^m$  is the end effector pose in  $m$ -dimensional task space,  $q \in R^n$  is the joint space variable vector of dimension  $n$  and  $f$  is a nonlinear vector function obtained from the particular kinematic structure of the manipulator. Inverse kinematics requires to find  $f^{-1}$  in order to find one or more joint angle vectors for a given end-effector position and orientation, such that

$$q = f^{-1}(x) \quad (2)$$

For redundant manipulator, a specific inverse kinematic solution with minimum norm can be obtained by using the Moor-Penrose pseudoinverse of  $J(q)$  [18],

$$\Delta q = J^\#(q)\Delta x \quad (3)$$

where,  $J^\#(q) = J^T(q)(J(q)J^T(q))^{-1}$  (or, a weighted version of the pseudoinverse) which minimizes the quadratic form  $(\Delta q^T \Delta q)$  at the configuration  $q$ . But this does not guarantee avoidance of all occurrences of singularity. A more general solution of inverse kinematic is given by

$$\Delta q = J^\# \Delta x + (I_{n \times n} - J^\# J) \Delta q_0 \quad (4)$$

Eq.(4) is a non-minimum norm solution, where a homogeneous term is added to the minimum norm solution.  $I_{n \times n}$  is an identity matrix of dimension  $n$  and  $\Delta q_0$  is an arbitrary vector, denoting null-space motion. The null-space displacement (the homogeneous term) is added with the minimum-norm-term through a projection operator  $(I_{n \times n} - J^\# J)$ . This non-minimum norm solution allows reaching a secondary objective for dexterity. One way of obtaining  $\Delta q_0$  is through the gradient of a scalar objective function  $\Phi(q)$  (the secondary function to be satisfied for dexterity)

such that  $\Delta q_0 = -k\Delta\Phi(q)$  where  $k$  is a positive constant.

In this paper, instead a numerical algorithmic strategy is taken. The inverse kinematics and redundancy resolution is solved by prioritizing the tasks – the primary task being the reachability to a point in workspace ( $X_g, Y_g, Z_g$ ) and the secondary task being a configuration where the manipulator Jacobian attains a minimum condition number (the most isotropic configuration). For primary task, a geometrical approach is employed to solve inverse kinematics and among all the feasible solutions, the configuration with Jacobian of minimum condition number is selected in an iterative procedure. Two cases are considered here, one planar 3-R structure of Fig.1(a) and one spatial 5-R structure of Fig.1(b).

It is to be noted that to deal with unit mismatch presence in the Jacobian, relating the joint space and task space velocities, the Jacobian is normalized customarily before computing the condition number [5]. However, in the exemplar implementations in this article, the Jacobian is not normalized, since, only the position of the end-effector is considered here (not the orientation).

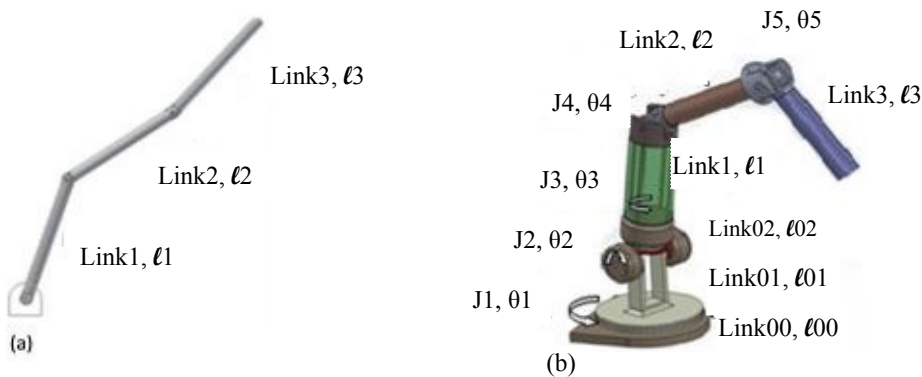


Figure 1. (a) 3R planar robot (b) spatial 5-R manipulator. The nomenclatures of joints and links are shown in the figure. It is assumed that the 3dof shoulder is of intersecting type using a differential shoulder, making  $\ell_{00} = \ell_{01} = \ell_{02} = 0$ .

The inverse kinematics has been divided into two sub-problems for implementation in this article, namely planar and spatial. The algorithm for planar sub-problem is employed for the spatial manipulator, when more than two links lie in a plane. The planar sub-problem involves repetitive use of inverse kinematics solution of 2-link planar arm with respect to of some reference configuration and incrementing the previous joint angle till the maximum value in its range is reached. The condition number of the manipulator Jacobian is computed at each inverse kinematics configuration for one reachable goal point within the workspace. Configuration with minimum condition number may lead to an optimal solution.

First three joints of the 5-R arm (see Fig 1(b)) are of intersecting type using a differential shoulder mechanism with links  $\ell_{00} = \ell_{01} = \ell_{02} = 0$ . Inverse kinematics for the spatial 5-R arm of Fig.1(b) needs specific geometric considerations to be solved. XYZ coordinate frame is attached at the base of the manipulator. The inverse kinematics of 5-R spatial manipulator is evaluated using the following steps:

**Step 1:** From given goal point  $P_g (X_g, Y_g, Z_g)$  draw projection onto X-Y plane and calculate angle  $\theta'_1$  (angle between home reference plane and projection plane on X-Y from goal point) which is evaluated when  $\phi=0$  and  $\theta_3=0$  (swivel arm angle,  $\phi$  is also zero, the swivel arm angle is defined as the angle between arm plane and reference plane, see Fig. 2).

$$\theta'_1 = \tan^{-1}(Y_g / X_g) \quad (5)$$

**Step 2:** Take  $\theta'_2$  corresponding to a angle formed by link1 (when  $\phi=0$  and  $\theta_3=0$ ), such that the distance between elbow1 position  $P'_{e1}$  (see Fig.2) and goal point  $P_g$  is less than or equal to  $(\ell_2 + \ell_3)$ .

**Step3:** Angles  $\theta_4$  and  $\theta_5$  (for elbow1 and elbow2) are independent of swivel arm angle,  $\phi$ . So angle  $\theta_4$  and  $\theta_5$  are calculate as Eq.(6) and Eq.(7)

$$\theta_5 = \pm 2 \tan^{-1} \sqrt{\frac{((\ell_2 + \ell_3)^2 - ((X_g - P'_{elx})^2 + (Y_g - P'_{ely})^2))}{(-(\ell_2 - \ell_3)^2 + (X_g - P'_{elx})^2 + (Y_g - P'_{ely})^2)}} \quad (6)$$

$$\theta_4 = \tan^{-1}((Y_g - P'_{ely}) / (X_g - P'_{elx})) \pm \tan^{-1}((\ell_3 \sin \theta_5) / (\ell_2 + \ell_3 \cos \theta_5)) - \theta_2' \quad (7)$$

$P'_{elx}$ ,  $P'_{ely}$  and  $P'_{elz}$  are the respectively x, y and z coordinate of the elbow1 position  $P'_{e1}$ . (see Fig.2)

**Step4:** Find elbows new positions in swivel arm plane (after rotating the reference plane by swivel arm angle,  $\phi$ ):

Now elbow1 and elbow2 position  $P'_{e1}$  and  $P'_{e2}$  is rotated by an angle  $\phi$  about the  $P_{sg}$  (see Fig.2). After rotation the new elbows positions are  $P_{e1}$  and  $P_{e2}$  respectively. Calculate the new elbows positions using the Rodrigues' rotation formula as given in eq. (8). If elbow1 position vector corresponding to  $P'_{e1}$  is represented by  $r_1$ , then after rotation by angle  $\phi$  about the  $P_{sg}$ , new elbow1 position  $P_{e1}$  is given by  $r_2$  as

$$r_2 = r_1 \cos \phi + (s \times r_1) \sin \phi + s(r_1^T s)(1 - \cos \phi) \quad (8)$$

$$\text{where} \quad s = (P_g - P_s) / |P_g - P_s| \quad (9)$$

$P_s$  is the position of shoulder with respect to base.

**Step 5:** Find first two joint angles ( $\theta_1$  and  $\theta_2$ ) from elbow1 new position  $P_{e1}$ .

$$\theta_1 = \tan^{-1}(P_{ely} / P_{elx}) \quad (10)$$

$$\theta_2 = \tan^{-1}(\sqrt{P_{elx}^2 + P_{ely}^2} / (-P_{elz})) \quad (11)$$

$P_{elx}$ ,  $P_{ely}$  and  $P_{elz}$  are the respectively X, Y and Z coordinate of the elbow1 new position  $P_{e1}$  (see Fig.2).

**Step6:** Roll angle  $\theta_3$  calculation: Evaluate the elbow2 new position  $P_{e2}$  (see Fig.3) using step 4. Elbow2 new position is function of  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  and  $\theta_4$  and from these  $\theta_3$  is calculated as

$$\theta_3 = \cos^{-1}((P_{e2z} + \ell_1 \cos \theta_2 - \ell_2 \cos \theta_2 \sin \theta_4) / (\ell_3 \cos \theta_4 \sin \theta_2)) \quad (12)$$

where  $P_{e2z}$  represent the z- coordinate of elbow2 new position  $P_{e2}$ .

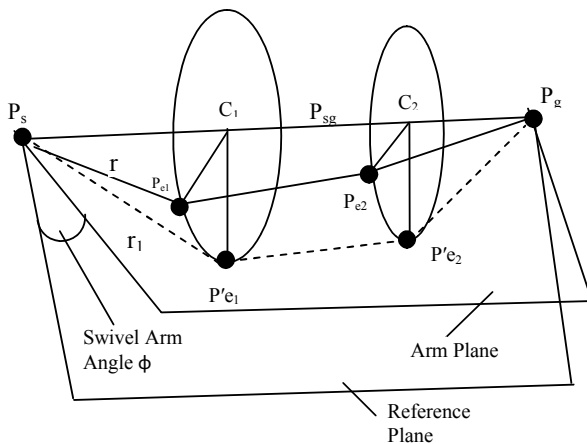


Figure 2. Definition of reference plane ( $\phi=0$ ) and swivel arm angle  $\phi$ , elbow1 and elbow2 rotates about vector  $P_{sg}$  (through shoulder,  $P_s$  to goal,  $P_g$ ) on circles whose centres are  $C_1$  and  $C_2$  respectively. Elbows positions  $P'_{e1}$  and  $P'_{e2}$  on reference plane, elbows positions  $P_{e1}$  and  $P_{e2}$  on arm plane.

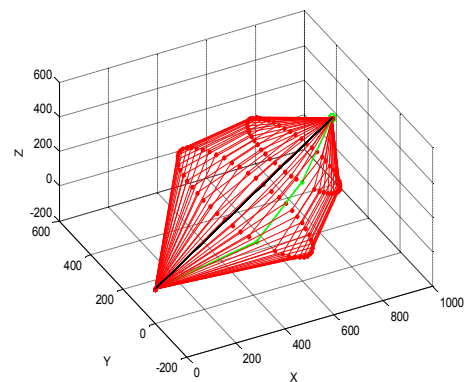


Figure 3. Simulation of feasible Poses for different swivel arm angle,  $\phi$ . The green configuration shows the reference plane and black line shows the rotation axis.

From above six steps one set of 5-R manipulator joints angles  $\theta_1, \theta_2, \theta_3, \theta_4$  and  $\theta_5$  are evaluated. Fig. 3 shows the different configuration of 5-R manipulator for different swivel arm angle,  $\phi$ . In this simulation the shoulder is at position (0,0,0), the goal at position (850,300,200) and link lengths  $\ell_1, \ell_2$  and  $\ell_3$  are 500mm, 300mm and 200mm respectively. In this figure the dotted points indicate the elbows positions. The two circles are formed by the dotted points that indicate the path of the elbow1 and elbow2. Note that from elbow1 position to goal position we have shown only one configuration here (although there are two configurations are possible, elbow up and elbow down configurations). In Fig.3, the green configuration shows the reference plane of the robot arm where swivel arm angle,  $\phi$  is zero.

### 3. Workspace Evaluation

The evaluation of workspace of a manipulator is necessary for the evaluation of Global Conditioning Index. Numerical methods are mostly used for spatial redundant manipulators. Monte Carlo random sampling numerical method was proposed by Rastegar and Perel [9] to generate the workspaces and the boundary surfaces of manipulators using forward kinematics. Rastegar and Perel applied the Monte Carlo method in the joint space of the manipulator to estimate the workspace. Similarly, Alciatore in [10] and Cao [11] also employed the Monte Carlo technique in joint space. This article employs Monte Carlo method for workspace computation in task space in contrary to many implementations in joint space found in literature. Here first a box is defined with sides of length greater than or equal to the double of maximum reach of the manipulator and the uniform random points are generated in the whole volume of the box. For the present case (5-R spatial manipulator) longest side of the box is taken as slightly greater than sum of double the maximum reach lengths ( $2\ell_1 + 2\ell_2 + 2\ell_3$ ). The manipulator base is kept at the origin of the box. An inverse kinematics algorithm is applied for each point in the box and it is checked whether the point is within reachable workspace or not (restricting the computed joint angles within joint motion range). This method does not need to generate the boundary for calculation of the workspace. Let  $N$  be the total number of uniform random points generated within the box and  $N_0$  be the numbers of those points reachable by the manipulator. The workspace volume is then evaluated as: where,  $V$  = volume of the box

$$\text{Workspace volume} = (N_0/N)V \quad (13)$$

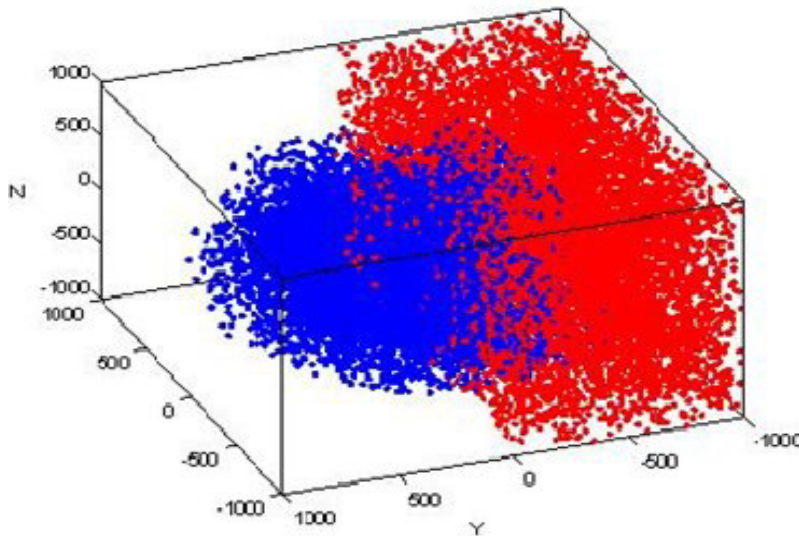


Figure 4. Workspace (dark blue points) of 5-R spatial manipulator. Red points are random points

Fig.4 shows the simulation work space (represented by dark blue points, red points are distributed random points in whole workspace, here shown only half part) of 5-R redundant manipulator (as shown in Fig.1.b.) with joint angle ranges are  $\theta_1 \in [0, \pi]$ ,  $\theta_2 \in [0, \pi]$ ,  $\theta_3 \in [0, \pi]$ ,  $\theta_4 \in [0, \pi]$ ,  $\theta_5 \in [0, \pi]$ , and  $\ell_1, \ell_2$  and  $\ell_3$  are 500mm, 300mm and 200mm respectively.

#### 4. The Global Performance Metric

Global performance metrics have been commonly used for many potential applications in robotics, i.e. optimal design purpose, trajectory planning, redundancy treatment and dexterity analysis. Again dexterity of a robot manipulator is said to be the ability to perform one or more secondary tasks over mere reachability. One of the reasons of having redundancy in a manipulator kinematic structure is to achieve some degree of dexterity. This article chooses the Global Conditioning Index proposed by Gosselin and Angeles [6]. The condition number of the Jacobian,  $J$  can be computed as the ratio of the maximum to minimum singular values of  $J$  i.e,  $k(J) = \sigma_{max}/\sigma_{min}$  where the singular values are obtained through singular value decomposition of  $J$ . Angeles in [5] defined a quantity called as Local Conditioning Index (LCI) equaling reciprocal of the condition number ( $1/k(J)$ ); to be noted that LCI varies between 0 and 1. This index gives a local measure of kinematic performance at a given configuration. In order to get a global performance measure, the local conditioning index is evaluated all over the workspace so that the *Global Conditioning Index (GCI)*,  $\eta$  is given by

$$\eta = A/W ; \text{ where, } A = \int_w 1/k(J)dw \text{ and } W = \int_w dw \quad (14)$$

where,  $W$  being the workspace of the manipulator. Clearly  $0 < \eta < 1$ .

Let  $k_i(J)$  be the condition number of corresponding to a configuration at the  $i^{th}$  reachable point in the workspace. For numerical computation of GCI using discrete points (randomly generated), the above definition reduces to a single average of the reciprocal of condition numbers for all the reachable points. Thus, GCI can be rewritten as

$$\eta = \frac{1}{N_0} \sum_{i=1}^{N_0} 1/k_i(J) \quad (15)$$

where  $i$  represents the  $i^{th}$  individual discrete point in the reachable workspace. To be noted that, on the boundary of workspace  $1/k(J)$  becomes zero. Maximizing the conditioning index will make the manipulator approaching isotropy.

#### 5. Optimization Problem Formulation and Evaluation

This article addresses problem formulation of the optimization, where the performance metric (GCI) chosen is function of link lengths. Instead of absolute link length, length ratios are considered in the formulation, reducing the complexity of the problem. With the definition of GCI as in section 4, the following maximization problem is formulated:

$$\begin{aligned} & \max_{r_i} \{\eta(r_i)\} \\ \text{Subject to} \quad & r_{il} \leq r_i \leq r_{iu} \\ & q_{jl} \leq q_j \leq q_{ju} \\ & L = \sum_{i=1}^n \ell_i \end{aligned}$$

Where,  $j = 1 \text{ to } n$ ;  $i = (n-1)$ ;  $r_i = (\ell_i / \ell_{i+1})$ ;

Subscripts  $u$  and  $l$  denote upper and lower limits respectively,  $r_i$  is the length ratio of link- $i$  to link- $(i+1)$ ,  $q_j$  is the  $j^{th}$  joint angle.

A scheme to solve the optimization problem is shown in Fig. 5. In Fig.1(a), a 3-R manipulator with link lengths  $\ell_1$ ,  $\ell_2$  and  $\ell_3$  shown whose task space is 2DOF. Hence 3-R planar robot becomes a redundant robot. The optimization results of this 3-R planar robot are shown in Fig.6. Here the optimum link ratios are found to be  $\ell_1/\ell_2 = 1.20$  and  $\ell_2/\ell_3 = 1.05$ . Similarly simulation result for 5-R spatial manipulator (see Fig.1(b)) is shown in Fig.7. The optimum link ratios obtained are  $\ell_1/\ell_2 = 1.5$  and  $\ell_2/\ell_3 = 0.9$ .



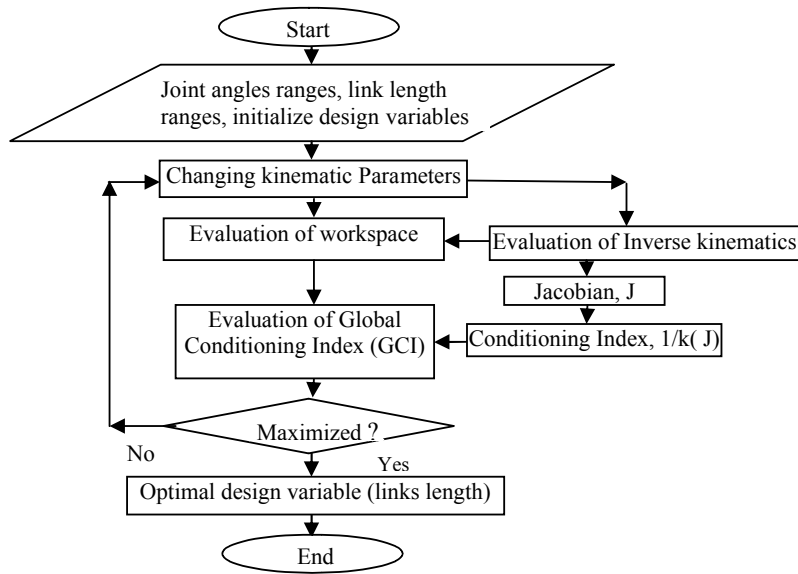


Figure 5. Proposed optimization algorithm

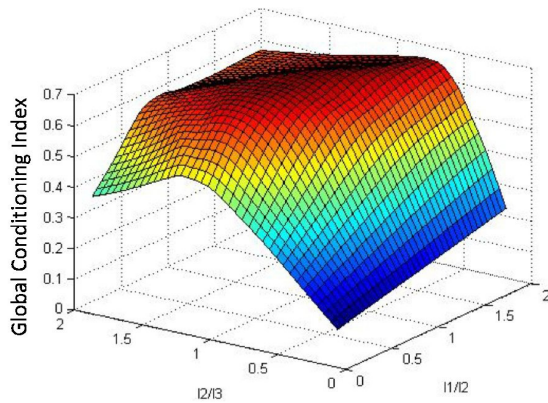


Figure 6. 3D plot of GCI distribution over link ratios of

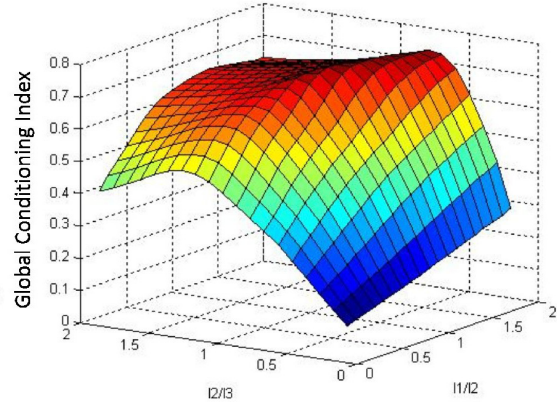


Figure 7. 3D plot of GCI distribution over link ratios of spatial 5-R manipulator.

Optimization problem is also solved by the binary-coded Genetic Algorithm (GA) [12]. Binary-coded GA begins with a population of initial solutions that are selected at random. The fitness is assigned to all the solutions in population. Then population is accomplished by three main operators, namely reproduction, crossover and mutation. In this study, GA-based optimization is implemented by using the Genetic algorithm Matlab toolbox<sup>®</sup>. As the performance of a GA depends on its parameters, a systematic and thorough study is carried out to determine the optimal set of GA-parameters. The following GA-parameters are found to give the best results during the GA-based optimization of 3-DOF manipulator: population size = 20; creation function: uniform; fitness scaling function = rank; selection function = roulette; crossover function = scattered; crossover fraction = 0.8; mutation function = adaptive feasible; maximum number of generations = 50. After 50 generations the optimal link-length ratios are obtained as  $l_1/l_2 = 1.202$  and  $l_2/l_3 = 1.085$ .

Similarly, the following GA-parameters are found to yield the best results during the GA-based optimization of GCI of 5-DOF spatial manipulator arm: population size = 20; creation function: uniform; fitness scaling function =

rank; selection function = roulette; crossover function = scattered; crossover fraction = 0.85; mutation function = adaptive feasible; maximum number of generations = 80. The optimal values of link-length ratios are turned out to be equal to  $\ell_1/\ell_2 = 1.506$  and  $\ell_2/\ell_3 = 0.87$ . It has been observed that the results generated using GA are in excellent agreement with those obtained with numerical method.

## 6. Conclusion and Future Work

This article presents an approach for design optimization of redundant serial link manipulators. Two specific geometries are considered for the implementations, namely a 3-R planar and a 5-R spatial manipulator. It is to be noted that, end-effector orientation is not considered in the exemplar implementations, which eliminates the problem of unit mismatch in using the Jacobian for computing its condition number. However, for general problem with end-effector orientation, the Jacobian must be normalized before computing the condition number. The optimization is based on a performance metric with link lengths as the chosen design variables. The performance metric chosen is a Global Conditioning Index proposed by Gosselin and Angeles, which requires evaluation of the manipulator workspace volume. The workspaces of the manipulators are numerically evaluated using Monte Carlo technique in task space unlike many joint space implementation found in literature. The implementation of Monte Carlo technique in task space is more intuitive; however, Cartesian space implementation requires inverse kinematics to be solved. This article also describes inverse kinematics of the redundant manipulators in consideration here. In doing so, sub-problem is formulated for planar redundancy resolution iteratively. This formulation is further used in doing inverse kinematics for spatial arm. Then the design optimization is formulated using the objective of maximizing the Global Conditioning Index (GCI), where the design variables are link length ratios. The optimization is solved firstly using a grid search method. Then Genetic algorithm technique is utilized to solve the same problem, using genetic algorithm toolbox of Matlab©. The results of the search method are in good agreement with the results obtained using genetic algorithm. This article only addresses the approach to optimize a redundant manipulator based on one performance metric. Complexity analysis of and optimization on the algorithms are part of future work. Future work also will involve multi-objective formulation for optimization of the manipulators, based on various performance measures of redundant robot arms.

## References

- [1] A.A. Maciejewski and C.A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environment", The International Journal of Robotics Research, Vol.4, No.3, pp. 109-116, 1985.
- [2] J. Wang, Y. Li and X. Zhao, "Inverse kinematics and control of a 7-DOF redundant manipulator based on the closed-loop algorithm", International Journal of Advanced Robotics System, Vol.7, No.4, pp. 1-9, 2010.
- [3] J.K. Salisbury and J.J. Craig, "Articulated hands: force control and kinematic issues", The International Journal of Robotics Research, Vol. 1, No. 1, pp. 4-17, 1982.
- [4] C. Klein, B. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators", The International Journal of Robotics Research, Vol. 6, No. 2, pp.72-83, 1987.
- [5] J. Angeles, "The design of isotropic manipulator architectures in the presence of redundancies", The International Journal of Robotics Research, Vol. 11, No. 3, 196-201, 1992.
- [6] C. Gosselin, and J. Angeles, "A global performance index for the kinematic optimization of robotic manipulators," Journal of Mechanical Design, Vol. 113, No. 3, pp. 220-226, 1991.
- [7] D.C.H. Yang and T.W. Lee, "On the workspace of mechanical manipulators", ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, pp 62-69, 1983
- [8] X. Chen, K.C. Gupta, "Geometric modeling and visualization of manipulator workspace", ASME Computers in Engineering, Vol.1, pp.469-474, 1991
- [9] J. Rastegar, D. Perel, "Generation of manipulator workspace boundary geometry using the Monte Carlo method and interactive computer graphics", ASME trends and developments in mechanisms, machines and robotics, Vol.3, pp 299-305, 1988
- [10] D. Alciatore and C. Ng, "Determining manipulator workspace boundaries using the Monte Carlo method and least squares segmentation", Proc. the 23rd ASME Design Technical Conference, pp. 11-14, 1994
- [11] Yi Cao, Haihe Zang, Lan Wu and Tao Lu, "An engineering-oriented method for the three dimensional workspace generation of robot manipulator" Journal of Information & Computational Science, Vol. 8, No 1, pp. 51-61, 2011
- [12] D. Baesley, D. Bull, R. Martin, "An overview of genetic algorithms: Part 1, fundamentals", University computing, Vol. 15, No. 2, pp. 58-69, 1993.