Task Based Optimal Geometric Design and Positioning of Serial Robotic Manipulators

Sasan Barissi and **Hamid D. Taghirad**

Advanced Robotics and Automated Systems (ARAS)
Faculty of Electrical Engineering,
K.N. Toosi University of Technology.

***Abstract*— This paper devises a multi-objective cost function which elaborates different constraints as well as an optimality criterion for design of serial robotic manipulators. In practice, inclusion of different constraints drastically limits the possible range of design parameters. The result of minimizing this multi-objective cost function is compared with another method which locates an optimal solution using a graphical representation. The effectiveness of the proposed cost function is demonstrated by a unified solution for both methods. In addition, possible tolerance of design parameters is compensated by considering a neighborhood around these parameters. Through an illustrative example, it is shown that the inclusiveness and flexibility of the proposed method makes it suitable for geometric design optimization of robotic manipulators.**

I. INTRODUCTION

DESIGN of robotic manipulators has always been a challenging issue. The complexity of structure and multitude of parameters makes the analytical design approach significantly complicated. This complication can be seen in [1], in which the design of specific robotic structures is reported. Moreover, in [2] different criteria has been proposed to simplify the structure and eliminate internal singular points. Exploiting a graphical representation for design of robotic manipulators can be found in [3]. However, in practice considering multi-objective criteria in the design makes these approaches very cumbersome. This may lead to design the dexterous workspace with numerical methods and visualization.

Defining an optimality index is another approach presented in [4,5]. Although these indices work impeccably with one structure, it is useless to compare different robotic structures due to their scale dependence. This problem has been reported in [6] and different optimality measures have been compared in [6,7]. In addition to these scale independent indices, we have considered different types of constraints for a pre-specified task.

In [8,9] the authors are focused in finding global optimization approach to design a well conditioned general purpose robot. However, we are more concerned about adding practical design limitations on a pre-defined trajectory. An analytical approach is also presented in [10], which is difficult or even impossible to solve for the practical robot design with more than 3-DOF. Different approaches are presented in [11-14], in which some kinematic and dynamic limitations are introduced.

In this paper, different kinematic limitations are formulated into a multi-objective cost function. Also, a visual approach is applied for a typical design with few parameters to show the effectiveness of this cost function and as a new approach for design applicable on few parameters. For this purpose, the optimization problem with a general form of kinematic representation is formulated for robotic manipulators. Then, by introducing different constraints, visual optimality patterns are devised; by searching through these patterns an optimal solution is effectively achieved. Next, we have formulated these constraints and an optimality measure within a multi-objective cost function. Finally, an optimization method have been planned to consider a margin for possible tolerance of design parameters as well as finding the minimum value for the cost function. The solution for this unified multi-objective cost function is examined, and analyzed in detail to show the effectiveness of the proposed method in practice.

II. PROBLEM STATEMENT

Denavit-Hartenberg convention [15] is a well accepted way to describe kinematic and geometric structure of a robotic manipulator. Therefore, the general problem of designing a serial manipulator can be recast into finding appropriate Denavit-Hartenberg parameters to optimize some measures. In fact, some of these parameters are geometrical design parameters and others are joint variables by which the posture of robot is changing. Our goal is to find these geometrical design parameters in order to minimize the cost index for different postures in a pre-defined trajectory. Besides, there are some constraints that have to be considered for solving the optimization problem. In general, the optimization problem to be solved can be written as following.

$$\min\{ f(a_i, \alpha_i, d_i, \theta_i, ..., a_{n-1}, \alpha_{n-1}, d_{n-1}, \theta_{n-1})\}$$

$$\text{subject to: } \begin{cases} c_j(x) = 0 \\ c_k(x) \le 0 \end{cases} \qquad (1)$$

in which, $c_j(x)$ and $c_k(x)$ are possible constraints and f is the multi-objective cost function to be minimized. It can be seen that all constraints (linear or nonlinear) can be converted to

the form shown in (1). Analytical solutions for this optimization problem can be found in [6]. But, these solutions are limited to 2R and 3R robots with only one point as a desired task. However, a real task is made up of continuous paths which contain infinite number of task points. Therefore, in this case we have to sample the path to a finite number of task points, and mention the arithmetic mean of cost function for these task points as the cumulative cost. For example, consider the problem depicted in Fig. 1. In this typical design example, 10 task points are assigned on the workspace of the robot which is quarter of a circle with radius of 2 meters. This example is a simplified version of a painting robot to spray color or defrosting liquid over an airplane's fuselage. This typical example is treated in the optimization problem developed in the different sections of this article.

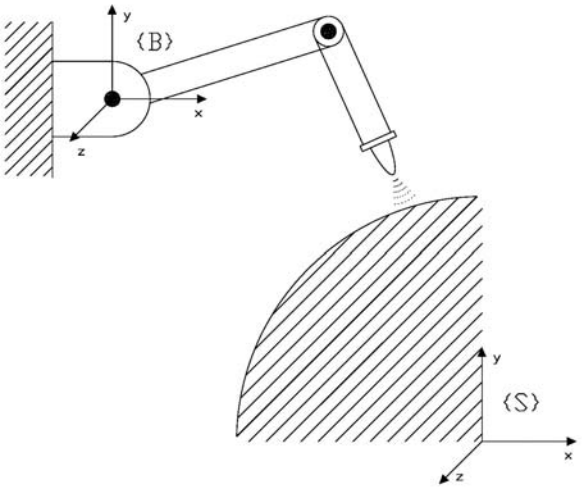

Fig. 1. A typical design example of a painting robot which will be optimized for link lengths and workstation positioning.

The objective is to design the robot link lengths as well as positioning the coordinate system of the workstation in order to satisfy a number of design constraints. Hence, the parameters to be optimized are: link lengths and position of coordinate system {S} relative to {B} which is attached to the airplane's body as depicted in Fig. 1. It should be noted that even in the design procedure of practical robots, many assumptions in the robot's structure have to be made in order to have analytical inverse kinematic solution; such as position and orientation decoupling, computational consideration, or facilitating the specific task for the robot.

III. Optimality Patterns

In accordance to the painting robot example, there are four design parameters to be optimized:

l_1: length of first link from base,

l_2: length of second link from the end of first link,

$[x_c, y_c]$: origin of coordinate system {S} relative to {B}.

We will separate design parameters (l_1 and l_2) from workstation positioning parameters $[x_c, y_c]$ so that we can visualize the effect of variation in l_1 and l_2 on a 2D grayscale map. According to the previous section, we have sampled the continuous path to a finite set of task points. These points can be represented with respect to {B} for a position of workstation $[x_c, y_c]$. Solving the inverse kinematics problem for each of these task points will result sets of solutions for joint variables of the robot [16]. According to our trajectory planning, we will choose a set as a suitable solution. Therefore, in our example we will have 10 sets of joint variables for 10 task points. Each of these sets can uniquely describe the position and posture of the robot at a task point.

Accordingly, we will assign a cost corresponding to the specific position and posture of the robot at each task point. The total cost along the path will be average cost of these task points. Hence, each constant set of $[x_c, y_c]$ and variable sets of $[l_1, l_2]$ can be represented in a map similar to Fig. 2. In this figure, effects of varying link lengths (l_1, l_2) on the cost function can be seen when {S} is positioned in [3m,-2m] relative to {B} (x_c = 3, y_c.= -2). Throughout this paper we call such maps an *Optimality Pattern*. By using this pattern, the cost of each design can be evaluated. In the figure the cost function is calculated solely in accordance with the condition number of Jacobian matrix. We have used the inverse of condition number with a transformation which will be discussed later.

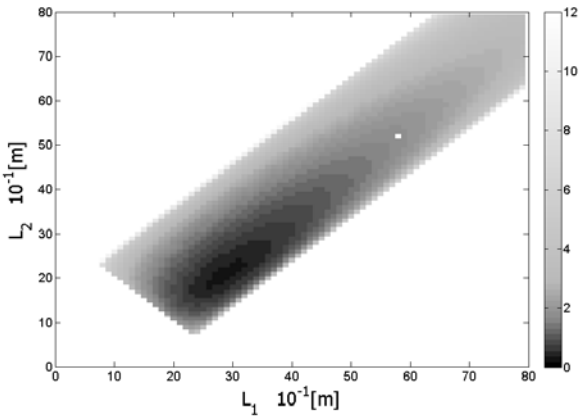

Fig. 2. Optimality pattern using a cost related to the condition number for xc = 3, yc = -2, the best possible answer (minimum cost) can be seen in black color; the white color indicates designs that cannon reach for at least one task point in a trajectory.

In Fig. 2 and other similar grayscale maps in this paper, the x-axis of the graph indicates first link length of the robot (l_1) and the y-axis indicates the second link length (l_2). A unit used for both axes is deci-meter (0.1 meter).The optimal design is where the minimum cost occurs. Thus, the optimal solution is within the dark black region in Optimality Patterns. For example, in Fig. 2 the point [l_1 = 2.9m, l_2 = 2.1m] is the minimum point.

The second step is to include constraints that a designer has to consider for the specific task. In practice, there are many constraints that limit our design options. Here we give the description of some general constraints; precise mathematical formulation of these constraints will be discussed later.

A. Joint Constraint

Practically, joints are bound to articulate along specified limits. This is possibly due to mechanical design limitations or cost reducing issues. As depicted in Fig. 3(a) this constraint has a huge effect on limiting the design workspace. Here the limitations are $|\theta_1| < 45°$ and $|\theta_2| < 45°$ for the first and second joint variables respectively. In the figure the gray colored region (indicated by an arrow) is dissatisfied by the joint constraint; after excluding the non reachable region (white colored), there remains a limited region to choose design parameters.

B. Collision with the Workstation

In some circumstances the structure of robot collides with the body of workstation. In our example, robot's first link might collide with the fuselage. Therefore, the robot must be avoided from getting into the workstation. Limitation due to this constraint is indicated on Fig. 3(b) by an arrow (the gray colored region).

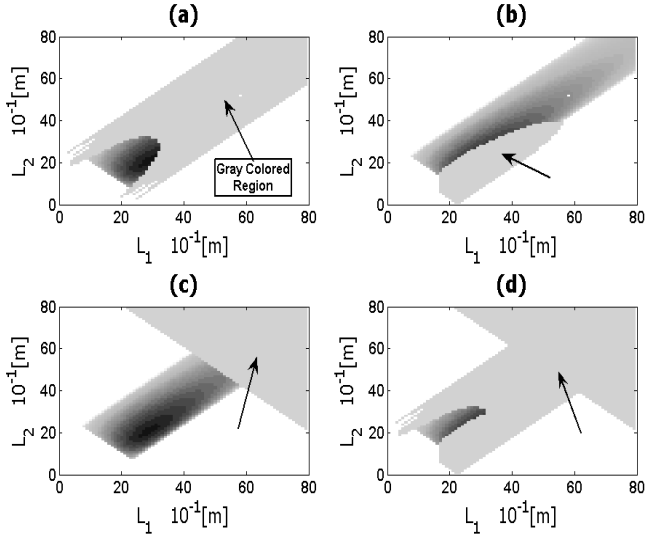

Fig. 3. Optimality patterns with different constraints included (limitations due to constraints are in gray color which is indicated by an arrow in each figure): (a) joint constraint (b) collision with the workstation (c) length constraint (d) all of these three constraints.

C. Length Constraint

In practice, there is limited workspace for a robotic manipulator. Accordingly, we have to limit the maximum length of the robot. Another setback for a long robotic manipulator is its larger range of resonance and flexibility which is unfavorable in this design. The effect of limiting total link length to maximum of 10m for our design example is shown in Fig. 3(c).

Finally, the resulted Optimality Pattern after including all constraints and the optimality measure can be seen in Fig. 3(d). In this figure the possible design region is drastically limited. Thus, we can see that the effects of different constraints is an important factor and a design which only considers an optimality measure will fail to work properly in a task with various limitations.

IV. Search Through Patterns

Now that we have defined the Optimality Pattern and separated design parameters from workstation positioning, we can analyze these patterns and search to find optimal design parameters. First, we will depict the Optimality Pattern for an arbitrary point of [x_c = 3.5, y_c = 0] and its neighborhood. After analyzing these patterns, we will extrapolate our analysis results to find an optimal solution. Optimality Patterns with respect to this point and its neighborhood are shown in Fig. 4. The number indicated on top of each figure shows the minimum cost value. As a reminder, each figure has its own specific set of [x_c, y_c] and the white colored region shows the excluded design region because of different constraints.

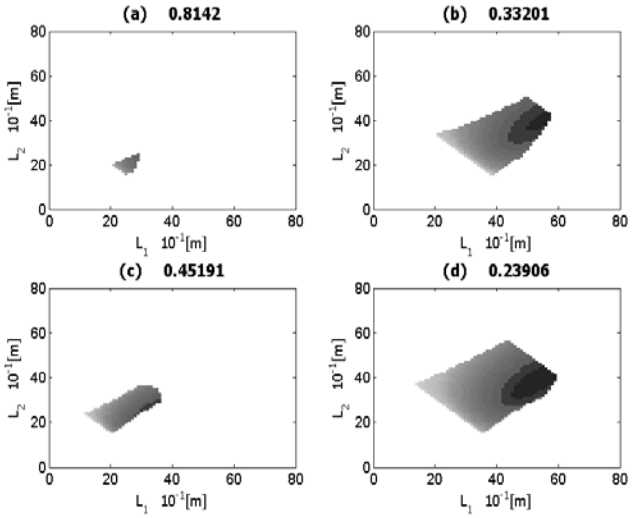

Fig. 4. Optimality patterns for four set of workstation positioning [x_c, y_c]: (a) [3.5,0] (b) [5,0] (c) [3.5,-1.5] (d) [5,-1.5]; optimal region is shown in black color and prohibited region is shown in white color; the minimum cost is indicated on top of each figure.

It can be seen in Fig. 4(a) that if we put the workstation's coordinate system {S} in [3.5,0], we will be restricted to a small region. Moving the origin farther from body to point [5,0] (Fig. 4 (b)) we will gain much freedom to choose design parameters; but even in this point our optimal region (shown in black color in Fig. 4(b)) is in the vicinity of the prohibited region. Moreover, by lowering the workstation in y-axis to [5,-1.5] (Fig. 4(d)), the accessible area for choosing design parameters will expand; also the cost will be lessened. Therefore, decreasing the workstation's origin index in y-axis or increasing in x-axis direction will result better design parameters which will have better cost index and broader margin from forbidden and inaccessible regions.

Thus, we will step forward in x-axis and step down in y-axis to reach a lower cost region. As a result, the point [4.5, -2.5] will be reached to analyze its vicinity. Fig. 5 shows Optimality Patterns for these sets of points. By inspecting these patterns, it can be figured out that Fig. 5 (d) has lost nearly half of its optimal region (shown in black color) and is close to the forbidden region. Also, the optimal region in Fig. 5(a) is about to enter the prohibited region. Notice that

the optimal region will be lost if we move the workstation farther from the robot. As a result, two options will remain: Fig. 5(b) and Fig. 5(c). In this case, Fig. 5(c) is better because it has lower cost and broader margin from inaccessible regions. The optimal link length corresponding to the minimum cost value in Fig. 5(c) is $l_1 = 5.2m$ and $l_2 = 3.7m$ and the optimal position of workstation is $x_c = 4.5m$ and $y_c = -3m$.

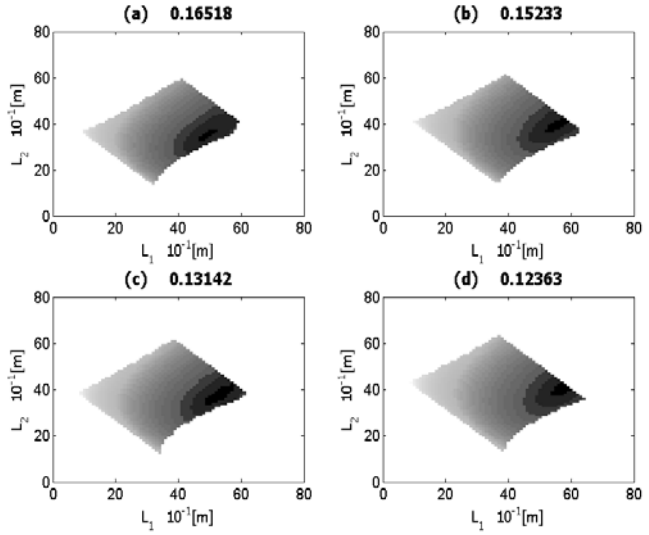

Fig. 5. Optimality patterns for four sets of workstation positioning [x_c, y_c]: (a) [4.5,-2.5] (b) [5,-2.5] (c) [4.5,-3] (d) [5,-3]; optimal region is shown in black color and prohibited region is shown in white color; the minimum cost is indicated on top of each figure.

V. Cost Function

As it can be seen in the trends of figures, in addition to the optimality measures there are various types of constraints that have to be formulated into a cost function. This aim can be accomplished by exploiting the cost function described in (2) and (3) [17,18]. It can be proved that every constraint can be converted into one of these two types:

$$\begin{cases} c_j(x) = 0 ..... j = 1...l \\ c_k(x) \le 0 ..... k = 1...m \end{cases} \quad (2)$$

in which c_j is an equality constraint and l is the total number of such constraints; c_k is inequality constraint with total quantity of m. The cost function (described as *cost(x)*) appropriate for these constraints can be written as:

$$\cos t(x) = \sum_{j=1}^{l} [c_j(x)]^2 + \sum_{k=1}^{m} [\max(0, c_k(x))]^2 \quad (3)$$

An advantage of this cost function is its ability to easily add a new constraint by a simple addition. In what follows a dexterity measure and some typical constraints (which were briefly described in the previous section) will be formulated into a multi-objective cost function.

A. Dexterity Measure

In the robotic community literature many dexterity measures have been introduced which have helped to design robots and interoperate different behaviors. In our design example in order to be able to compare robots with different link lengths, we have to choose a measure which is scale independent. In [6] some well-defined scale independent dexterity measures have been introduced. However, in our paper we prefer to use a simple one like condition number [7]. Since we will not need an analytical solution throughout this paper, its inability to be solved analytically is not a setback [8]. We have, also, changed the range of condition number from [1…∞] to [0…∞] by a transformation as defined by:

$$C_{Dext} = \frac{\sigma_{\max}}{\sigma_{\min}} - 1. \quad (4)$$

in which $\sigma_{\max}$ is the largest and $\sigma_{\min}$ is the smallest singular value of the appropriate Jacobian matrix.

B. Joint Constraints

We can classify joints into two primal forms of revolute and prismatic. In reality both types have some specific mechanical limits. Due to the unified definition of Denavit-Hartenberg parameters for these two joints, without loosing generality we consider the case of revolute joints. Considering this limitation for a revolute joint as [θ_{min} , θ_{max}], formula (5) can added to the multi-objective cost function [18]. By adding this factor to the function, a cost is assigned to a configuration passing these limits.

$$C_{Joint} = \max\{\theta_{\min} - \theta, 0\} + \max\{\theta - \theta_{\max}, 0\} \quad (5)$$

C. Link Length Constraint

Practically, the available space for a robot is bound to some restrictions. Therefore, we have to limit the robot's scale by limiting the length of its links. Besides, the added length will increase the unfavorable construction cost. Another setback for long robots is increased resonance and flexibility. Since our target is to build a robot with maximum stiffness, this flexibility is undesirable in this particular design scheme.

In order to formulate the cost index, it is known that the flexibility of a typical cylindrical link is proportional to the cube of length. Accordingly, the cost index for added flexibility has to be increased proportional to the cube of length. To sum up, formula (6) has been proposed to include a cost index for this constraint.

$$C_{Dim} = \max\{\sum_{i=1}^{n}(l_i) - l_{\max}, 0\} + k_{res}\sum_{i=1}^{n}(l_i^3) \quad (6)$$

In this formula l_{max} is maximum acceptable length of links and k_{res} is constant which depends on elasticity of material and cost of construction material; in our example we have assigned $k_{res} = 10^{-4}$. l_i is the length of i-th link; and n is the total number of links.

D. Reachability Constraint

One approach to formulate this constraint is to measure the distance between the task point and maximum reachable

point of the robot. Another interesting approach proposed in [18] is the use of Generalized Inverse Kinematics (GIK). To briefly explain GIK procedures, the inverse kinematics formulation will be transformed to a polynomial form. Therefore, in case a task point is out of reach, the solution to inverse kinematics will have an imaginary part. Thus, we can formulate this constraint as:

$$C_{\mathrm{Re}ach} = \{imag(u)\}^2 \quad (7)$$

in which u is the solution of transformed inverse kinematics problem for each task point and the *imag* operator derives the imaginary part of the solution.

E. Task Constraints

This group of constraints will avoid the robot to collide with different obstacles. These obstacles might be the workstation or other objects that restrict the maneuvering ability of the robot. To formulate this constraint, it is to find an analytical expression to describe inside of these objects. Afterwards, a cost is assigned which is proportional to the distance between the surface of the object and a point on robot's body. This will act as a spring, repelling the robot from getting into the object. All this process is done for the circle-shaped workstation of our example and proposed in (8) as one way to put this constraint into our cost function.

$$C_{Task} = \sum_{k=1}^{n} \max\{r^2 - [({}^0x_k - x_c)^2 + ({}^0y_k - y_c)^2], 0\} \quad (8)$$

in which r is the radius of a circle-shaped workstation and x_c, y_c are the position of the circle's center with respect to coordinate {0} which is attached to the robot's base. This formulation is for n-linked robot and ${}^0x_k$ is the position of the end of k-th link relative to coordinate {0} which can be calculated with respect to (9).

$${}^0x_k = {}^0T_k \, {}^k x_k \quad (9)$$

in which ${}^0T_k$ is 4-dimentional square transformation matrix which transforms the coordinate system attached to k-th link to the coordinate {0} attached to the base.

VI. Optimization

Up to now, the optimality criterion and different constraints have been separately formulated. To construct the final multi-objective cost function, all these constraints and the optimality measure have to be added with some weights as in (10). In this formula C is the constraint which was described in previous section and w is its respective weight.

$$\cos t(x) = w_{Dext} \times C_{Dext} + w_{Joint} \times C_{Joint} + \cdots \\ w_{Dim} \times C_{Dim} + w_{\mathrm{Re}ach} \times C_{\mathrm{Re}ach} + w_{Task} \times C_{Task} \quad (10)$$

These weights are assigned to rank the importance of constraints and normalize them. The normalization process is conducted in accordance with different units of constraints. In the example, w_{Dim} and w_{Reach} are 10 because of their similar dimensions; these weights are higher compared to w_{Dext} which is 1 due to their importance. Also, w_{Task} =15 and w_{Joint} =100. Having the smallest unit (radian), the joint constraint is assigned the highest weight.

Before solving to find the minimum cost, we have to deal with another issue. In reality, the design parameters like link length or joint twist are capable of having some tolerances which might be due to varying temperature, deformity of structure or imperfect construction. To bear these tolerances the optimal design parameters should have a good margin from the inaccessible region. Therefore, we have to plan an algorithm to search for a low-cost area instead of a low-cost point. This is achieved in (11) by integrating the cost function in a boundary defined by the tolerance of each parameter.

$$F(x) = \iint \cdots \int_{x-b}^{x+b} \cos t(l_1, l_2, \cdots, x_c, y_c, z_c) dl_1 dl_2 \cdots dz_c \quad (11)$$

where b is the maximum tolerance limit for the corresponding parameter; *cost* is the cost function for a single point and *F(x)* is the concluding cost function. Because numerical computation methods are being used in this paper, this integration will be transformed into discrete integration. By widening this tolerance margin (b), we can compensate imperfectness of parameters.

Finally, to solve this optimization problem, we have chosen the Genetic Algorithm (GA) method. On the account of the fact that our multi-objective cost function has multiple parameters with plenty of local optimal points, using gradient-based methods will result a local optima. Also concerning our problem, these methods are highly dependent on initial value of optimization algorithm. Besides, due to high to medium epistasis (independence) of parameters in this optimization problem, the GA method is suitable [19]. In such cases, even the random search methods will work better than Gradient-based methods. The result of optimization based on 10 iterations of GA can be seen in Table I. Results in Table I show the effectiveness of the chosen optimization algorithm.

TABLE I
OPTIMIZATION RESULTS BY GENETIC ALGORITHMS

Itr. No.	L_1	L_2	X_c	Y_c	Cost Value
#1	5.0636	3.7044	4.7417	-2.6017	37.957
#2	4.9952	3.6935	4.6971	-2.5913	37.928
#3	4.9739	3.7179	4.6192	-2.6792	37.725
#4	5.247	3.8606	4.5329	-3.1818	37.225
#5	5.1023	3.7454	4.69	-2.743	37.65
#6	5.1149	3.7681	4.648	-2.8316	37.49
#7	5.0602	3.7542	4.6313	-2.7643	37.555
#8	4.9332	3.6967	4.6095	-2.7342	37.833
#9	4.8429	3.5979	4.7123	-2.5103	38.678
#10	5.3842	3.9809	4.2646	-3.6556	37.382

Even with different initial values for each iteration all the solutions are converging to a unified solution. Besides, the ratio of l_2/l_1 in all of these iterations is 0.74±0.01 which is

close to the ratio of forearm and upper arm of a human body. This fact verifies the results further. The average and Standard Deviation (StdDev) of these 10 iterations can be seen at Table II. Low rates of StdDev particularly when compared to long lengths of the robot, is another factor for validating the results.

TABLE II
OPTIMIZATION RESULTS BY GENETIC ALGORITHMS

.	L_1	L_2	X_c	Y_c	Cost Value
Average	5.07174	3.75196	4.61466	-2.8293	37.7423
StdDev	0.1557	0.1044	0.1369	0.3433	0.4040

Finally, the Optimality Pattern of the final result is illustrated in Fig. 6. As can be seen, a wide range is available to choose appropriate design variables. Moreover, the optimal region (in black color) is wide enough to be away from the forbidden (white colored) region. Besides, the optimum solution resulted in this map is analogous to the one derived in the section IV.

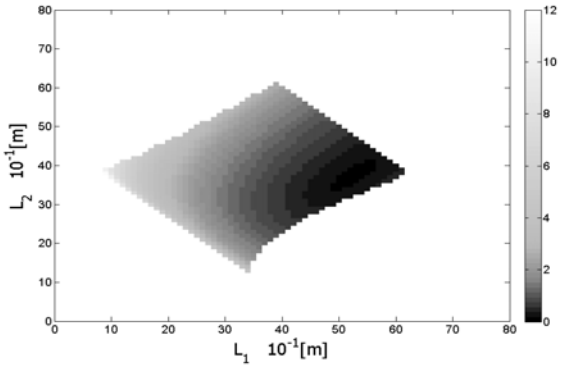

Fig. 6. Optimality pattern for the final optimal design solution: *xc* = 4.6, *yc* = -2.8, the best possible answer can be seen in black color with the minimum value; l_1 = 5.2 , l_2 = 3.7. The minimum cost of the optimal point is 0.1417.

VII. CONCLUSION

This paper has proposed and formulated a multi-objective cost function for the design of serial robotic manipulators. Some important constraints such as link length and task constraints are proposed and integrated into the multi-objective cost function. Besides, a method to consider tolerances of design parameters has been proposed. The method of Genetic Algorithm (GA) is used for optimizing this function which contains many local minimum points. A reasonable matching of solutions for 10 iteration of GA confirms the effectiveness of this algorithm and the proposed cost function. On the other hand, a visual design procedure has been presented. This visual method is particularly useful when analyzing the effect of varying parameters on optimality. Finally, the optimal design solution is verified by a unified solution using both the GA and optimality pattern approaches. The resemblance between ratio of forearm to upper arm in human body and the *l2/l1* ratio of the solution is another reason to validate the solution.

REFERENCES

[1] Y. Tsai, A. Soni, "Workspace synthesis of 3R, 4R, 5R, and 6R robots," *Journal of Mechanism and Machine Theory*, vol. 20, No. 6, pp. 555-563, 1985.

[2] J. M. Hollerbach, "Optimum kinematic design for a seven degree of freedom manipulator," *Robotics Research: The Second International Symposium*, 1985, pp. 349-356.

[3] H. Asada, "Dynamic analysis and design of robot manipulators using inertia ellipsoids," *in Proc. IEEE international Conference on Robotics*, Atlanta, March 1984, pp. 94-102.

[4] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, No. 2, Summer 1985.

[5] J. Angeles, F. Ranjbaran, R. V. Patel, "On the design of the kinematic structure of seven-axes redundant manipulators for maximum conditioning," *in Proc. IEEE International Conference on Intelligent Robotics and Automation*, Nice, France, 1992, pp. 492–499.

[6] J. O. Kim, "Dexterity measures for design and control of manipulators," *IEEE Int. workshop on intelligent robots and systems*, Osaka, Japan, 1991, pp. 758-763.

[7] C. A. Klein, B. E. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *International Journal of Robotics Research*, vol. 6, no. 2, pp. 72–83, Summer 1987.

[8] R. V. Mayorga, J. Carrera, M. M. Oritz, "A kinematics performance index based on the rate of change of a standard isotropy condition for robot design optimization," *Journal of Robotics and Autonomous Systems*, vol. 53, pp. 153–163, Sept. 2005.

[9] L. J. Stocco, S. E. Salcudean, F. Sassani, "On the use of scaling matrices for task-specific robot design," *IEEE Trans. On Robotics and Automation*, vol. 15, no. 5, pp. 958–965, Oct. 1999.

[10] M. Ceccarelli, C. Lanni, "A multi-objective optimum design of general 3R manipulators for prescribed workspace limits," *Journal of Mechanism and Machine Theory*, vol. 39, pp. 119–132, July 2003.

[11] J. Y. Park, P. H. Chang, J. O. Kim, "A global approach for robot kinematics design using the grid method," *International Journal of Control, Automation, and Systems*, vol. 4, no. 5, pp. 575-591, Oct. 2006.

[12] J. Han, W. K. Chung, Y. Youm, S. H. Kim "Task based design of modular robot manipulator using efficient genetic algorithm," *in Proc. IEEE International Conference on Robotics and Automation*, Albuquerque, New Mexico, April 1997, pp. 507–512.

[13] J. S. Rastegar, L. Liu, D. Yin, "Task-specific optimal simultaneous kinematic, dynamic, and control design of high-performance robotic systems," *IEEE/ASME Trans. On Mechatronics*, vol. 4, no. 4, pp. 387–394, Dec. 1999.

[14] J. A. Snyman, D. F. Berner, "The design of a planar robotic manipulator for optimum performance of prescribed tasks," *Journal of Structural Optimization*, Springer-Verlag, vol. 18, pp. 95–106, 1999.

[15] M. W. Spong, S. Hutchinson, M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, John Wiley & Sons, 2006, pp. 76–83.

[16] A. Goldenberg, B. Benhabib, R. Fenton, "A complete generalized solution to the inverse kinematics of robots," *IEEE Journal of Robotics and Automation,* vol. 1, no. 1, pp. 14–20, March. 1985.

[17] J. O. Kim, P. Khosla, "A formulation for task based design of robot manipulators," *in Proc. IEEE International Conf. on Intelligent Robotics and Systems*, Yokohama, Japan, 1993, pp. 2310–2317.

[18] C. Paredis, P. Khosla, "Kinematic design of serial link manipulators from task specifications," *International Journal of Robotics Research*, vol. 12, no. 3, pp. 274–287, June. 1993.

[19] R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*. Hoboken, NJ, John Wiley & Sons, 2004, p. 32.