

## Research Article

# Service Arms with Unconventional Robotic Parameters for Intricate Workstations: Optimal Number and Dimensional Synthesis

Satwinder Singh and Ekta Singla

Mechanical Engineering Department, IIT Ropar, Rupnagar 140001, India

Correspondence should be addressed to Ekta Singla; [ekta@iitrpr.ac.in](mailto:ekta@iitrpr.ac.in)

Received 25 November 2015; Revised 17 February 2016; Accepted 3 March 2016

Academic Editor: Shahram Payandeh

Copyright © 2016 S. Singh and E. Singla. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A task-oriented design strategy is presented in this paper for service manipulators. The tasks are normally defined in the form of working locations where the end-effector can work while avoiding the obstacles. To acquire feasible solutions in cluttered environments, the robotic parameters (D-H parameters) are allowed to take unconventional values. This enhances the solution space and it is observed that, by inducing this flexibility, the required number of degrees of freedom for fulfilling a given task can be reduced. A bilevel optimization problem is formulated with the outer layer utilizing the binary search method for minimizing the number of degrees of freedom. To enlarge the applicability domain of the proposed strategy, the upper limit of the number of joints is kept more than six. These allowable redundant joints would help in providing solution for intricate workcells. For each iteration of the upper level, a constrained nonlinear problem is solved for dimensional synthesis of the manipulator. The methodology is demonstrated through a case study of a realistic environment of a cluttered server room. A 7-link service arm, synthesized using the proposed method, is able to fulfill two different tasks effectively.

## 1. Introduction

The tremendously increasing variety in the robotic services leads to *less repetitive tasks*. To provide solution to the resulting variations, adaptive design techniques and realization strategies have taken the attention of the researchers. Given a cluttered environment with fixed obstacles, as shown in Figure 1, tasks are generally defined as working locations for the robot end-effector. For such intricate workstations with narrow passages, manipulability needs to be induced in the required design to maneuver within the cluttered workcells. Focus of the work is to determine adaptive number of degrees of freedom (dof) for given work scenarios. There is no limitation of keeping degrees of freedom as six or less. To acquire even one connecting path in the given cluttered environments, a design with even larger dof may be used. Novelty of the work lies in enhancing the solution space through unconventional values of the robotic parameters.

In recent works, Yang and Chen [1] presented a study on optimizing the number of degrees of freedom for given tasks. The work is limited to specific conditions, without the description of any obstacles and/or workcells. In another work by Zhang and Wang [2], kinematic redundancy is utilized for avoidance of given obstacles. A fixed degrees-of-freedom manipulator is utilized in the work and no general strategy is provided for any given task and/or workcell. The techniques used to avoid collisions are the utilization of kinematic redundancy and/or having movable platform in most of the works. Modularity in robotic arms is worked upon by few researchers [3, 4] for providing a solution for changes in the environments. Normally, in the modular development strategies, a configuration of modular components is changed and then a systematic method is developed to compute its robotic parameters, thus formulating the kinematic equations. However, to design a service arm for *any given spatial workcell*, a general platform is required, with least input

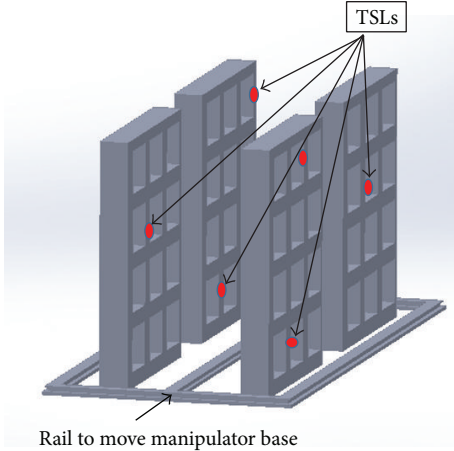


FIGURE 1: A server room environment with task space locations.

required from the user. Importance of the customized design of robotic arms has been discussed in recent works and the algorithms have been presented either with fixed degrees of freedom or with fixed or no environment [5, 6]. The field of task-based design needs to be further explored for algorithms adaptive to given workcells. This work provides one such solution to deal with given cluttered environment, through unconventional robotic parameters and by trading-off the requirement of number of degrees of freedom.

## 2. Problem Formulation

A nested optimization approach is proposed in this work with all the robotic parameters as design variables in the inner loop for dimensional synthesis and a unidirectional problem solving in number of dof in the outer loop. Important aspects related to the problem are the utilization of unconventional robotic parameters and varying number of degrees of freedom:

- (1) Flexibility in the values of robotic parameters (D-H parameters in this work) leads to a larger search domain, which is required for a feasible solution in intricate workstations. For highly cluttered environments, this aspect is expected to play an important role. Related to this work, for a 6-link manipulator, Patel and Sobh [7] utilized a larger range of D-H parameters. The adaptable modules are fabricated to adapt the unconventional values of D-H parameters such as adaptable links [8] with adaptable connectors. The proposed methodology provides a general platform which can provide a robotic arm design corresponding to a given workstation, defined by a workspace model.
- (2) The number of dof is not confined to any predefined value. Kinematic redundancy is kept acceptable in the varying number of dof. This is to gain the inherent advantages of a large number of dof, if required, according to the work scenario.

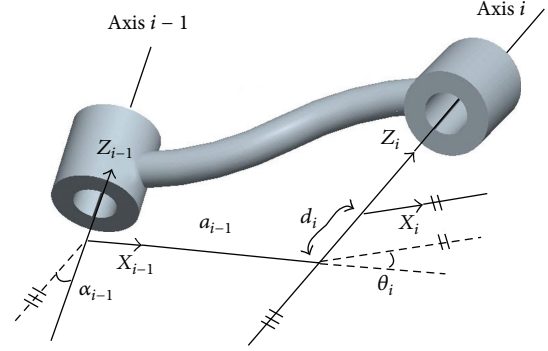


FIGURE 2: D-H representation.

The proposed strategy is illustrated through realistic workcells, with the changes in number of task space locations (TSLs).

*2.1. Varying Number of Degrees of Freedom and Unconventional Robotic Parameters.* For an  $n$ -linked serial manipulator, D-H convention is used to attach the reference frames to the serially joined links (refer to [9]). To define the relation between  $(i-1)$ th and  $i$ th links,  $\forall i \leq n$ , four parameters—twist angle ( $\alpha_{i-1}$ ), link length ( $a_{i-1}$ ), joint offset ( $d_i$ ), and joint angle  $\theta_i$ —are associated.

As shown in Figure 2, the link parameters ( $(\alpha_{i-1})$ ,  $(a_{i-1})$ ) and joint offset ( $d_i$ ) are fixed to provide a manipulator configuration. With  $n$  values of  $\theta_i$ , this configuration will attain one particular posture. By varying the values of each ( $\theta_i$ ), the posture of this configuration can be changed; that is, position of the end-effector can be varied to reach the desired locations (TSLs)—say “ $N$ ” in number.

A transformation matrix, using D-H parameters, defines transformation of frame  $i$  relative to frame  $i-1$ , represented as  ${}^{i-1}T_i$ , and is computed as

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & a_{i-1} \\ S\theta_i C\alpha_{i-1} & C\theta_i C\alpha_{i-1} & -S\alpha_{i-1} & -S\alpha_{i-1}d_i \\ S\theta_i S\alpha_{i-1} & C\theta_i S\alpha_{i-1} & -C\alpha_{i-1} & C\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

For  $n$  dof, comprising  $n+1$  frames,  $n$  number of transformation matrices are computed. The concatenated matrix provides the required transformation from frame- $n$  corresponding to the end-effector to frame-0 attached to the base:

$${}^0T_n = {}^0T_1 {}^1T_2 {}^2T_3 \dots {}^{n-1}T_n. \quad (2)$$

The robotic parameters in manipulator kinematic equations are dependent on number of dof. For a single posture of an  $n$ -link manipulator, total  $4n$  parameters are required to compute kinematic equations. Since the value of “ $n$ ” is not defined *a priori*, the total number of design variables will vary with each change in dof  $n$ . This is handled through a nested optimization problem formulation.

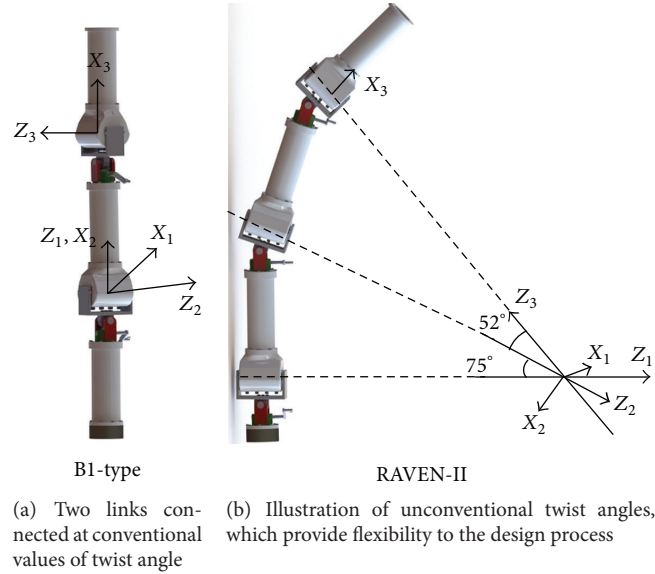


FIGURE 3: Unconventional robotic parameters.

Apart from the possibility of using redundant joints in the design, if and when required, the robotic parameters are also kept flexible in the proposed methodology. Figures 3(a) and 3(b) present the significance of conventional twist angles with normal values as  $0$  or  $\pi/2$  and unconventional twist angles values. The latter provides a larger solution space and thus a possibility of getting solution even in highly constrained workcells.

**2.2. Collision Avoidance.** To work on a general problem with any given cluttered workspace, an obstacle avoidance strategy is required. It is important to check any collision among the robotic links and any of the environmental objects. In this work, emphasis is given on the thorough examination of any configuration in question; that is, the collision is not checked just for end-effector and/or for a few points on the robotic links. For this purpose, the solid model of the workspace is required in Stereolithographic (stl) format which is modelled in Solidworks Premium 2013 version. The stl file provides the connectivity information of modelled environment in triangulated mesh format. The forward kinematic procedure is used to model the robotic arm at every iteration. Each link is assumed to be a rectangular parallelepiped with square cross section of a prescribed width. For collision detection, the proximity query package is utilized. The obstacle avoidance approach computes the minimum distance between the two solid models, represented in their triangulated form. In case there is collision between the two objects, the package furnishes the data about the colliding pairs. The collided pair may belong to the robot links or robot obstacles. The strategy computes a positive minimum distance “ $D$ ” between the models which is utilized in formation of corresponding constraints.

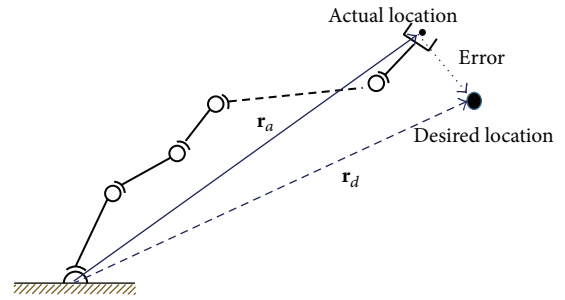


FIGURE 4: Schematic diagram of a serial redundant manipulator.

### 3. Task-Oriented Problem Formulation

A nested bilevel optimization problem is formulated for minimizing the number of dof at upper level (outer loop) and designing a robotic arm for the fixed number of dof in each inner loop.

**3.1. Reachability at Working Locations: Objective Function.** For each current value of “ $n$ ” in the outer loop, robotic parameters are synthesized for *reachability* at the required TSLs. Reachability is measured as a squared error function of *Euclidean distances*, that is, distance between the end-effector position of the manipulator and the specified TSL as

$$P_{\text{error}} = \|\mathbf{r}_d - \mathbf{r}_a\|^2. \quad (3)$$

A schematic diagram of the end-effector position of a redundant manipulator while reaching at the desired task locations is shown in Figure 4. For an  $n$ -link manipulator, working for  $N$  TSLs—with the actual position of the end-effector corresponding to the  $j$ th TSL as vector  $\mathbf{r}_{ja}$  and the

desired TSL position as vector  $\mathbf{r}_{jd}$ —the error square sum for  $N$  number of TSLs is computed as

$$P_{j_{\text{error}}} = \sum_{j=1}^N \left[ \left\| \mathbf{r}_{d_j} - \mathbf{r}_{a_j} \right\|^2 \right]. \quad (4)$$

In Cartesian coordinates, (3) can be expressed as

$$P_{\text{error}} = (x_d - x_a)^2 + (y_d - y_a)^2 + (z_d - z_a)^2. \quad (5)$$

Finally, the cumulative error for all the  $N$  TSLs can be written as

$$P_{\text{error}} = \sum_{j=1}^N \left[ (x_{d_j} - x_{a_j})^2 + (y_{d_j} - y_{a_j})^2 + (z_{d_j} - z_{a_j})^2 \right]. \quad (6)$$

To include the orientation of a frame attached to the end-effector of the manipulator relative to the base frame, Euler angle conventions have been used. With  $\alpha_{td}$ ,  $\beta_{td}$ , and  $\gamma_{td}$  representing the Euler angles corresponding to a particular configuration of the manipulator under consideration, the objective function can be revised as

$$P_{\text{error}} = \sum_{t=1}^N \left[ (x_{td} - x_{ta})^2 + (y_{td} - y_{ta})^2 + (z_{td} - z_{ta})^2 \right] + (\alpha_{td} - \alpha_{ta})^2 + (\beta_{td} - \beta_{ta})^2 + (\gamma_{td} - \gamma_{ta})^2. \quad (7)$$

Corresponding actual orientation parameters ( $\alpha_{ta}$ ,  $\beta_{ta}$ , and  $\gamma_{ta}$ ) can be derived from the concatenative rotation matrix of the end-effector with respect to the base frame.

The actual coordinates of the manipulator end-effector are given by the forward kinematics using D-H parameters ( $a_{i-1}$ ,  $\alpha_{i-1}$ ,  $d_i$ ,  $\theta_i$ ). The transformation matrices are multiplied together to form an arm matrix. The first three elements of the global transformation matrix give the current location of the end-effector in Cartesian form, that is,  $(x_{ia}, y_{ia}, z_{ia})$ . It is worth mentioning here that the base point coordinates can also be considered as design variables, in case the application allows a flexibility in the installation point of the robot:

$$T_{\text{tool}}^{\text{base}} = T_{\text{translation}} T_1^0 T_2^1 T_3^2 \cdots T_n^{n-1}, \quad (8)$$

where  $T_{\text{translation}}$  is the coordinates of the base point.

**3.2. Unconventional Robotic Parameters: Design Variables.** The global transformation matrix is calculated to determine the actual position in Cartesian coordinates. The arm matrix is the function of the number of dof and D-H parameters and all are considered as design variables for the formulated problem. Out of all the D-H parameters, link parameters are fixed for each TSL, while the joint variables are varying to provide  $j$ th configuration. The change in joint variables is required to reach at different TSLs for each particular robotic

posture. With the number of TSLs as “ $N$ ”, total “ $N + n$ ” joint variables are required. The joint variables are expressed as

$$\begin{aligned} & \theta_{11}, \theta_{12}, \dots, \theta_{1n}; \\ & \theta_{21}, \theta_{22}, \dots, \theta_{2n}; \\ & \vdots \\ & \theta_{N1}, \theta_{N2}, \dots, \theta_{Nn}. \end{aligned} \quad (9)$$

The total number of design variables for  $n$ -link manipulator reaching “ $N$ ” TSLs is, therefore,  $(3 + N) * n$ . Since the number of dof “ $n$ ” is also a design variable, it leads to an inherent challenge of handling varying number of design variables in each iteration. Therefore, a bilevel problem is formulated to fix the number of dof at the upper level. The dimensional synthesis variable vector is expressed as  $\mathbf{x}$  and it is further a function of “ $n$ ”.

**3.3. Constraints Handling.** The minimization of the objective function is subject to the constraints due to the limits on the design variables and due to the environmental obstacles.

**3.3.1. Limiting Values of D-H Parameters: Inequality Constraints .** All the D-H parameters and degrees of freedom are design variables in the above formulated problem. The limiting bound/range on all the design variables are imposed as constraints for the optimization method. The limits on D-H parameters for the  $i$ th link can be written as

$$\begin{aligned} a_{il} &\leq a_i \leq a_{iu}, \\ \alpha_{il} &\leq \alpha_i \leq \alpha_{iu}, \\ d_{il} &\leq d_i \leq d_{iu} \quad \text{for all joints as revolute,} \\ \theta_{jil} &\leq \theta_{ji} \leq \theta_{jiu} \quad \forall j, \end{aligned} \quad (10)$$

where  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, N$ , and  $l$  in suffix represents the lower bound, while  $u$  represents the upper. Each inequality in (10) gives rise to a pair of constraints in the form

$$\begin{aligned} g_{i1}(\mathbf{x}) &= a_{il} - a_i \leq 0, \\ g_{i2}(\mathbf{x}) &= a_i - a_{iu} \leq 0. \end{aligned} \quad (11)$$

The bounds on the number of degrees of freedom are also discussed later.

**3.3.2. Cluttered Workstations: Inequality Constraints.** To compute the minimum distance between the robot and the workspace or among the robot links, the workspace and each link of the manipulator are presented in a triangulated mesh format, as discussed in Section 2.2. A function  $D(\mathbf{x})$  is defined as the minimum distance between two nearest triangle pairs. The minimum distance of separation (positive in nature), reported by the package, is returned as  $D$ . The function  $D$  gets a negative value representing the *overall* intersection, when

there is a collision between objects (robot-obstacle or link-link).

The inequality constraint including both cases is represented as

$$g_j(\mathbf{x}) = -D_j \leq 0, \quad j = 1, 2, \dots, N, \quad (12)$$

where  $N$  is the number of TSLs. Thus, the  $j$ th constraint gets active only when there is an intersection of the manipulator with any obstacle and/or among the links, at the posture corresponding to the  $j$ th TSL.

All these constraints, along with the objective function, constitute the problem in  $(\mathbf{x})$ . As discussed earlier, this constrained optimization problem is solved by using augmented Lagrangian method.

#### 4. Binary Search Method: Outer Loop

The binary search method is applied at the outer layer of the formulated bilevel optimization problem. The technique is used to locate the target value in a sorted array [10]. In this work, targeted value is the minimum number of dof at which the manipulator can be designed for reachability at desired TSLs. The range of dof is the array under consideration from which the middle element will be selected and every element of the array is an integer.

It is used to find out the optimal number of degrees of freedom  $n$ . Total number of design variables are dependent on dof as discussed in Section 3.2, that is,  $[(3 + N) * n]$ , where  $N$  is the number of TSLs. “ $n$ ” is an integer value and possesses a finite range of 3–12 in this work.

To initialize the method, the array is split in half and the middle element is selected as an input to the inner loop; that is,

$$n_{\text{mid}}^t = \frac{n_u - n_l}{2}, \quad (13)$$

where  $n_u$  and  $n_l$  are the upper and the lower limit of the number of dof and  $t$  represents the iteration number. In the inner loop, a nonlinear optimization problem has been formulated and augmented Lagrangian method is used to solve the highly constrained problem.  $n_{\text{mid}}^t$  is the value at which design vector  $\mathbf{x}$  is to be determined in inner loop optimization.

In case the solution does not exist at  $n_{\text{mid}}^t$ , it becomes the new  $n_l$  as

$$n_{\text{mid}}^{t+1} = \frac{n_u - n_{\text{mid}}^t}{2}. \quad (14)$$

However, if there exists a design solution at  $n_{\text{mid}}^t$ , then it becomes new  $n_u$  and

$$n_{\text{mid}}^{t+1} = \frac{n_{\text{mid}}^t - n_l}{2}. \quad (15)$$

The iterative process terminates with  $n_u - n_l = 1$ .

#### 5. Augmented Lagrangian Method: Inner Loop

This constrained optimization method uses the combination of duality and penalty aspects. A penalty function is induced in the objective function to check the constraint violation. The Hessian of Lagrangian can be ill-conditioned in some cases due to which it affects the rate of convergence. The dual method can be applied only on convex functions. So, in augmented Lagrangian method, a moderate penalty is applied to augment the objective function into its convex function [11].

Suppose  $f(\mathbf{x})$  is the objective function to be minimized subject to the inequality constraints

$$g_i(\mathbf{x}) \geq 0 \quad i = 1, 2, \dots, I, \quad (16)$$

and equality constraints

$$h_j(\mathbf{x}) = 0 \quad j = 1, 2, \dots, J, \quad (17)$$

where  $I$  and  $J$  are the number of inequality and equality constraints, respectively. The augmented Lagrangian function is then expressed as

$$\begin{aligned} \mathcal{F}(\mathbf{x}) = & f(\mathbf{x}) + \sum_{i=1}^I [\mu_i g_i(\mathbf{x})] \\ & + \frac{1}{2} R \sum_{i=1}^I [\max(0, g_i(\mathbf{x}))]^2 + \sum_{j=1}^J [\lambda_j h_j(\mathbf{x})] \\ & + \frac{1}{2} R \sum_{j=1}^J [\max(0, h_j(\mathbf{x}))]^2, \end{aligned} \quad (18)$$

where  $\mu_i$  and  $\lambda_j$  are the Lagrange multipliers corresponding to  $i$ th inequality and  $j$ th equality constraints, respectively, whereas  $R$  is the penalty parameter.

#### 6. Methodology

For a given workcell (in triangulated mesh format) with the tasks defined as the working locations (in Euclidean space), the problem formulated in the previous sections facilitates the solution of the problem with minimum number of dof. The complete methodology is summarized through the following steps.

##### Upper Level Start

- (1) Define  $n_u$  and  $n_l$  for binary search algorithm.
- (2) Compute the  $n_{\text{mid}}$  value and update it to lower level.

##### Lower Level Start

- (1) Formulate the objective function  $\mathcal{F}(\mathbf{x})$  referring to (18).
- (2) Specify the constraints due to parameter bounds and obstacles as mentioned in Section 3.3.

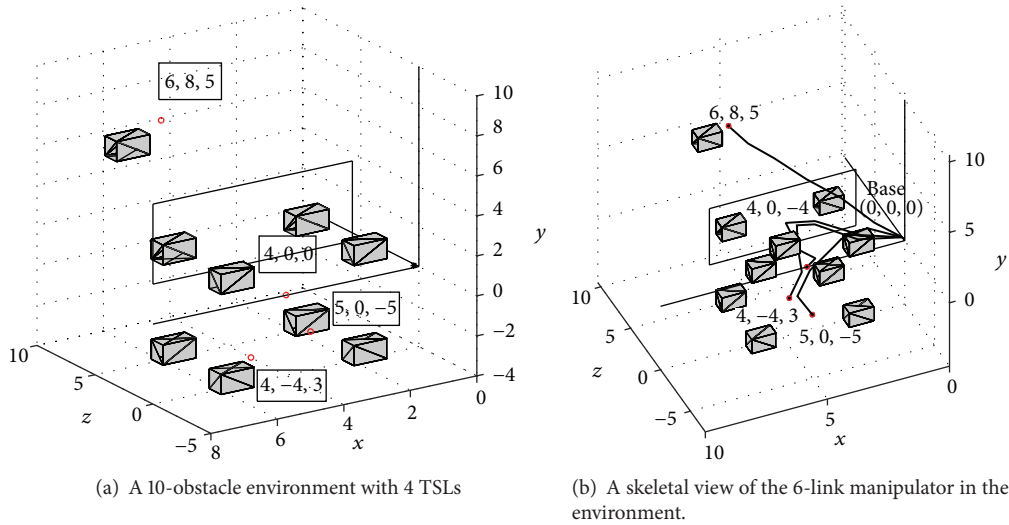


FIGURE 5: An exemplary case study for strategy validation.

- (3) Apply the augmented Lagrangian method to solve the formulated NLP problem. This involves updating the Lagrange multipliers as discussed in previous section.
- (4) Update the solution status to upper level.

#### Lower Level Ends

- (1) Based on the solution status at lower level, update  $n_{\text{mid}}$  according to (14) and (15).
- (2) Check termination criteria. If the condition mentioned after (15) is fulfilled, the corresponding inner loop solution  $\mathbf{x}$  is the required feasible solution with minimum number of dof.

*Upper Level Ends.* It is possible that, on termination, the objective function is not zero, which signifies that no solution is possible within the prescribed range of dof. The range of dof can also be expanded to increase the redundancy. The bounds on the link lengths can be relaxed, if allowed. The range of degrees of freedom is 3 to 12 in this work. Only revolute joint has been taken in the design process which means joint offset will be fixed parameter in all the cases.

## 7. Results and Discussion

The problem, discussed and formulated in the previous sections, is implemented in C++. For the execution of the code, the information of workspace environment in the triangulated mesh format, the task space locations in the environment, base position of the manipulator, and the limits on the D-H parameters (design variables) are required as the input data.

Using the proposed strategy, manipulator design for several environments has been synthesized and some of the results are presented in this section.

TABLE 1: Six-link manipulator design.

S. number	$\alpha_i$	$a_i$	$d_i$	$\theta_{i1}$	$\theta_{i2}$	$\theta_{i3}$	$\theta_{i4}$
1	0	0	0.4	0.74	0.32	0.18	0.55
2	0.97	2.20	-0.4	0.43	0.48	0.69	-0.87
3	0.64	1.99	-0.37	0.03	0.9	0.94	-0.38
4	0.55	1.99	-0.19	-0.14	1.44	0.77	0.54
5	0.77	1.49	-0.24	-0.14	1.26	0.5	0.92
6	0.99	1.49	-0.41	0.07	0.6	0.38	0.70

### 7.1. Example 1: Strategy Validation

*Case 1.* An exemplary workspace is taken into consideration for the validation of the strategy. It has 10 blocks scattered in the space, as shown in Figure 5(a). The required manipulator needs to work at 4 TSLs [(6, 8, 6), (4, 0, 0), (4, -4, 3), and (5, 0, -5)], while avoiding all the blocks. The base of the manipulator is fixed at (0,0,0). The design of an 8-link manipulator has been reported in the work of Singla et al. [5] for the same environment and TSLs. The number of dof was reported fixed in the work.

By using the proposed strategy given in this chapter, the 6-link manipulator can reach all the given TSLs. In the first iteration, the degrees of freedom have been taken as 8 at the outer loop and initialize the augmented Lagrangian method in the inner loop for reachability. The 8-link design outcome is expected because, with same input, it has been reported earlier.

The  $n_{\text{mid}}$ -update is done according to the algorithm discussed in Section 4. The number of dof is updated to 6 at the outer loop and this updated value is checked for the success in the inner loop. Table 1 presents the robotic parameters and Figure 5(b) shows the corresponding skeletal view of the manipulator in the workspace.

Binary search will update the new value of dof that is 5 in this case. At number of dof 5, the method fails to converge.

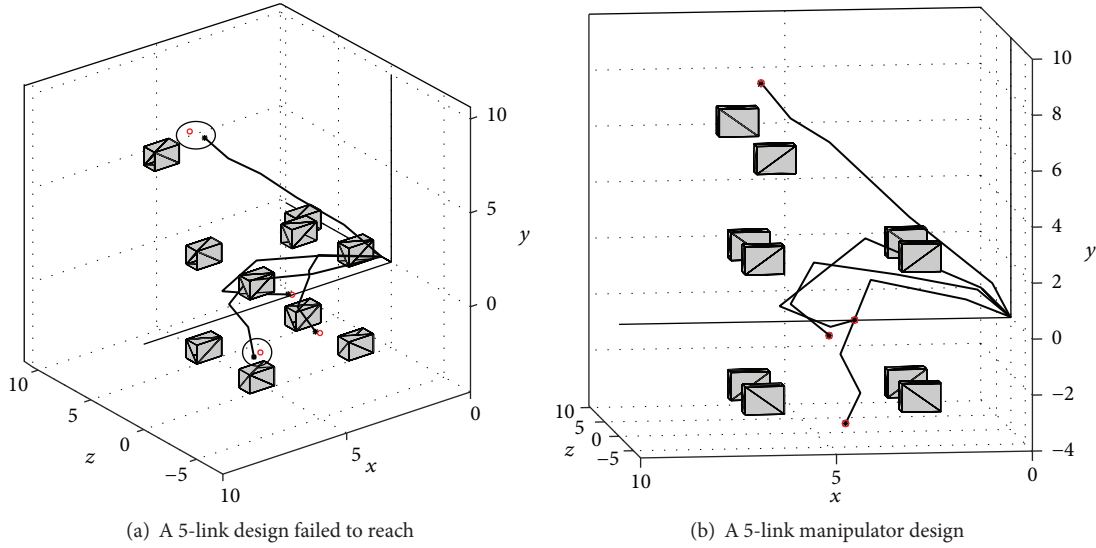


FIGURE 6: A 5-link design by varying the limits of the design variables.

A positive error value that is 1.0139 is left, which signifies that no solution exists at dof 5. The method will terminate here with a 6-link design as an output. The case showcases the importance of varying number of links.

*Case 2.* It is quite possible to achieve the required set of design variables with 5 dof, if variable bounds are relaxed. For 5-dof design, the link length limits in the failed case are  $[3, 3, 2, 1.5, 1.5]$  which are changed to  $[4, 4, 4, 2.5, 2.5]$ . With the changed limits, the reachability is achieved as shown in Figure 6(b). With the initial limits, robot arm was nearly outstretched to its full length and yet not able to reach one of the TSLs, as shown in Figure 6(a). Therefore, the variations in the link lengths are tried and successfully delivered the result.

**7.2. Example 2: Comparison with Conventional Values of D-H Parameters.** This example is included to compare the design results having conventional values of twist angles with the proposed flexibility in twist angles. The environment and the working locations are kept the same as used in the previous case study. In this example, the outer loop is not active and the number of dof is fixed manually at different levels. The problem starts with a 6-link configuration in which the link lengths and the joint offsets can vary and all the twist angles are taken as either perpendicular or parallel. The values of all the twist angles are taken the same as for PUMA configuration. This example is an attempt to compare whether a manipulator with conventional twist angles that is 0 or  $\pm\pi/2$  achieves the objective with same number of dof. If not, then how many dof are required to reach? Hence, the problem started with minimum number of dof which has been calculated in the previous example. It is observed that, with least variation in twist angles, no solution is obtained with number of dof 6, 7, and 9. Twist angle can vary along with other variables from dof 6 onwards because there is no set conventional value of twist angle in redundant manipulators. In this example, with first 6-dof puma configuration,

TABLE 2: Nine-link manipulator design with conventional twist angles.

S. number	$\alpha_i$	$a_i$	$d_i$	$\theta_{i1}$	$\theta_{i2}$	$\theta_{i3}$	$\theta_{i4}$
1	0	0	-0.51	0.41	1.39	3.00	-2.32
2	-1.48	3.06	-0.75	0.77	-2.89	2.01	-1.01
3	-0.01	0.61	-0.23	0.69	2.80	1.74	0.00
4	-1.49	1.99	0.99	0.39	1.34	-0.61	-0.50
5	-1.51	1.79	0.86	0.34	-2.17	-2.66	2.85
6	1.55	1.95	-0.51	0.39	-1.15	-2.99	-1.10
7	-0.01	1.91	-0.28	0.44	1.15	-2.50	0.91
8	0.09	1.81	0.32	0.49	-0.76	-2.98	-0.72
9	0	1.89	0.27	0.50	0.011	-1.13	-0.01

a 9-link manipulator is the design outcome. Corresponding D-H parameters are shown in Table 2 and the configurations are presented in Figure 7. There is an important aspect that with increase in solution space degrees of freedom can be decreased for reaching desired locations.

### 7.3. Example 3: Realistic Environment of a Server Room.

In this case study, a server room environment has been taken into consideration, as shown in Figure 8(a). The manipulator has to reach 3 TSLs,  $(185, 15, 80)$ ,  $(36, 25, 80)$ , and  $(180, 30, 43)$ , as shown in Figure 8(b) and the base point of the manipulator is taken at  $(110, 8, 30)$ . Minimum 6-link manipulator is required to reach all these TSLs. The corresponding design is presented in Table 3 and Figure 8(c) presents the visualization for desired working postures.

In the first case study, it has been shown that, with less number of dof and varying the limits of variable, it is possible to compute the solution set of parameters. Now, in the present scenario, number of TSLs increased from 3 to 5 and placed in such a way that 6-link design failed to reach which means strategy will automatically select the higher number

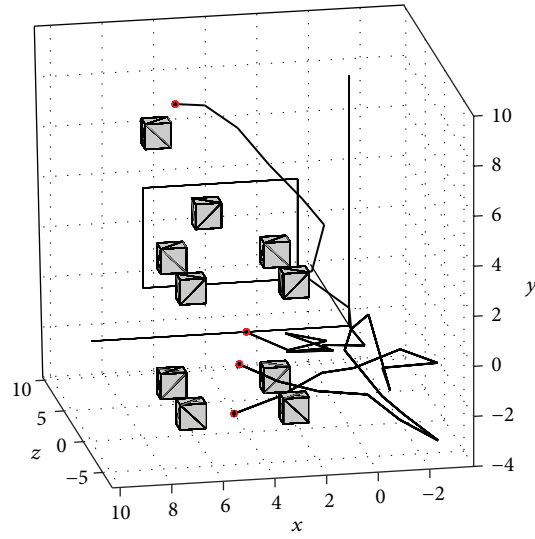
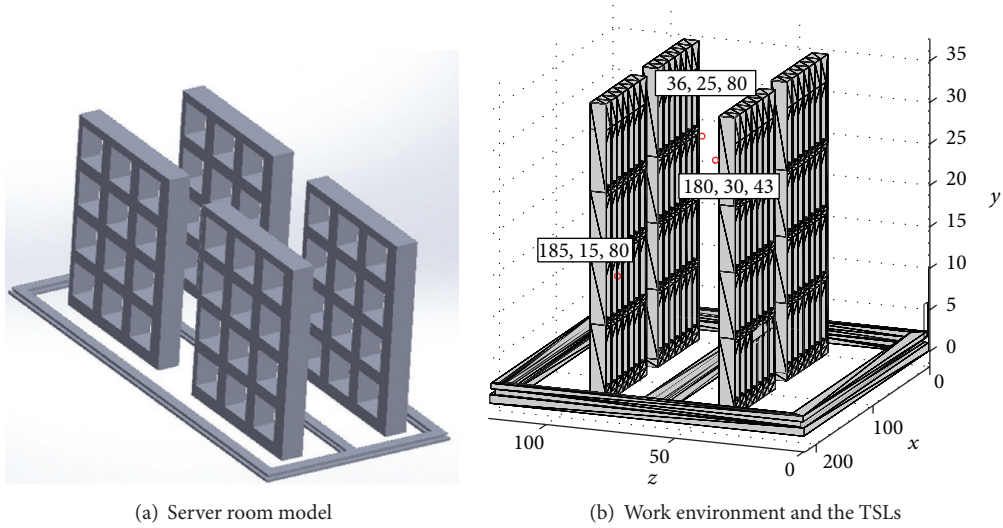
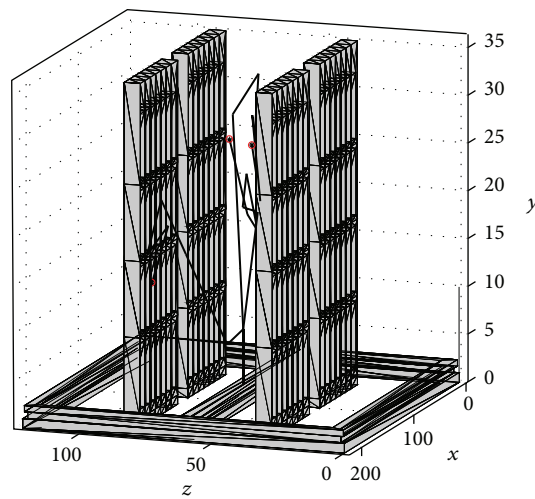


FIGURE 7: A 9-link manipulator design with the first 6 links in PUMA configuration.



(a) Server room model

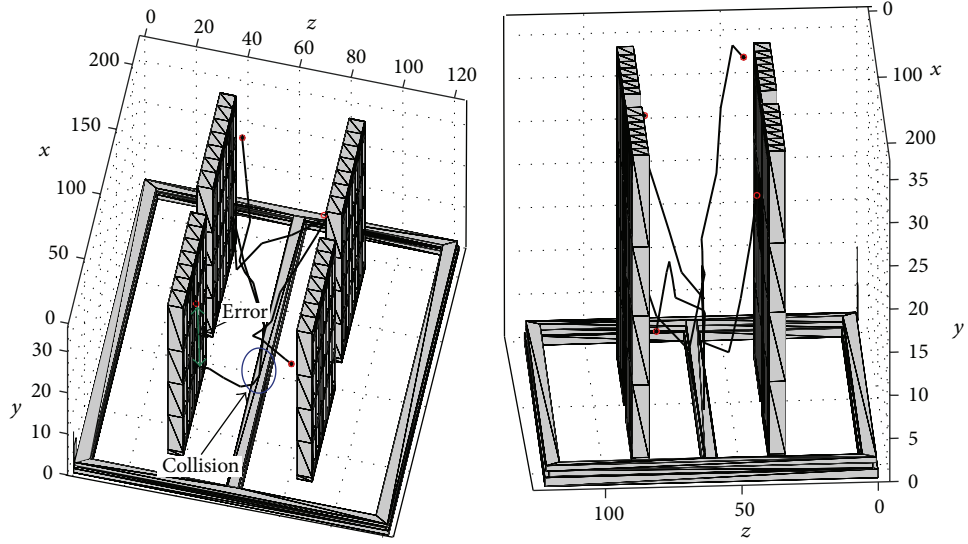
(b) Work environment and the TSLs



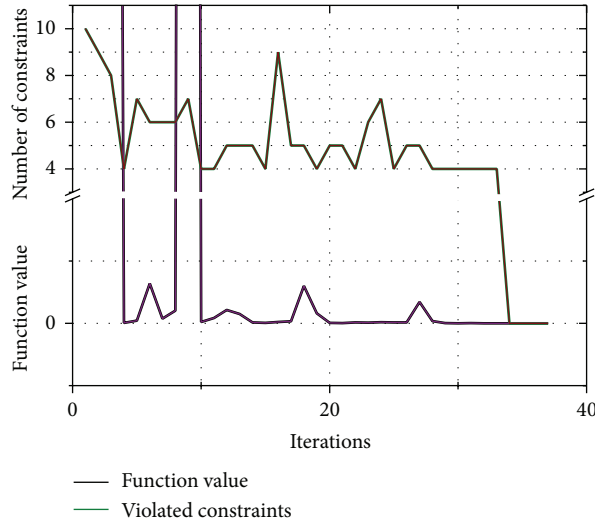
(c) Skeletal view of the 6-link manipulator in the server room

FIGURE 8: A server room case study.





(a) Seven-link manipulator in the server room, intermediate iteration number 5 (b) Seven-link design in the server room: final solution



(c) Number of violated constraints and the objective function evaluation

FIGURE 9: Analysis: the server room case study.

of dof. A 7-link manipulator design is selected to reach 5 TSLs [(185, 15, 80), (36, 25, 80), (180, 30, 43), (36, 35, 43), and (120, 35, 82)] with the same base point as shown in Figure 9(a) showing the violation of constraints in an intermediate iteration. The method will give the solution when there is no collision and all the parameters are converged according to the termination criteria. Figure 9(b) shows the model which is an outcome of a fully converged problem. The corresponding D-H parameters are shown in Table 4.

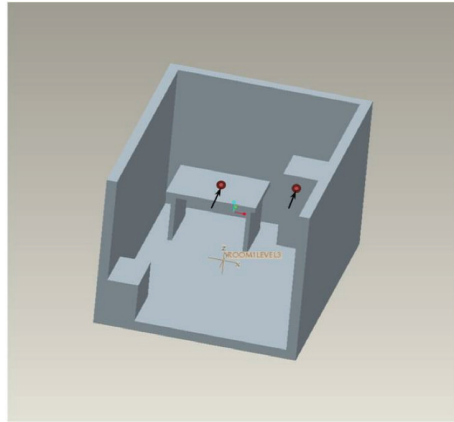
#### 7.4. Example 4: Synthesis for Both Position and Orientation.

This example presents a manipulator synthesis problem for both position and orientation of the end-effector, that is, for the entire arm-wrist combination. The workspace consists of an enclosed environment including a table and a big box,

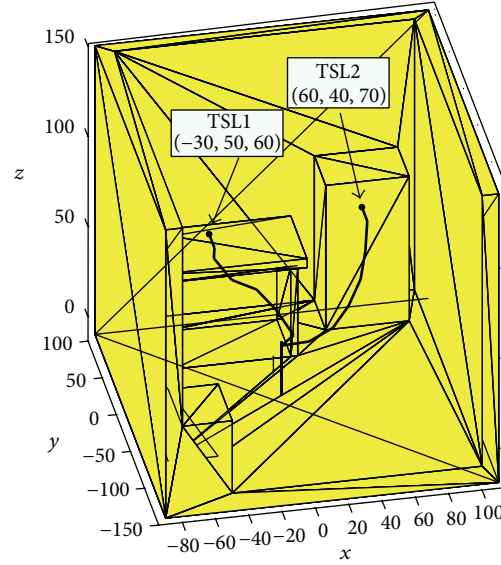
TABLE 3: Six-link manipulator design for server room environment.

S. number	$\alpha_i$	$a_i$	$d_i$	$\theta_{i1}$	$\theta_{i2}$	$\theta_{i3}$
1	0	0	0.09	0.13	1.9	-2.8
2	2.5	20.03	0.09	2.8	1.25	-2.09
3	0.72	20.3	0.08	2.7	0.47	-2.7
4	2.05	20.2	0.06	-0.44	-0.71	0.38
5	2.06	20.3	0.06	-0.75	-0.38	0.032
6	1.75	20.3	0.071	-0.30	0.58	0.08

representing a cupboard, as shown in Figure 10(a). The figure includes two TSLs along with the desired approaching directions of the end-effector. The prescribed positions for these locations are (-30, 50, 60) and (60, 40, 70) with  $(\pi/2, 0, -\pi/2)$



(a) The workspace



(b) The resulting configurations

FIGURE 10: Seven-link manipulator in a room environment with both position and orientation prescribed.

TABLE 4: Seven-link manipulator design for server room environment.

S. number	$\alpha_i$	$a_i$	$\theta_{i1}$	$\theta_{i2}$	$\theta_{i3}$	$\theta_{i4}$	$\theta_{i5}$
1	0	0	-2.72	0.81	-0.26	-0.35	1.89
2	2.49	15.99	0.00	-0.58	0.08	-2.9	-0.89
3	1.05	14.99	0.58	0.50	-0.36	0.53	0.56
4	-0.28	13.99	0.19	-0.07	-0.42	-0.06	2.80
5	1.19	12.97	0.26	-0.82	0.09	0.01	1.79
6	1.33	11.97	-0.04	0.45	0.04	-0.01	-0.43
7	0.73	10.98	0.11	0.06	-0.66	-0.00	-0.25

TABLE 5: Synthesis result: tasks included both specified position and orientation.

S. number	$a_i$	$\alpha_i$	$d_i$	$\theta_{1i}$	$\theta_{2i}$
1	17.4	0.70	-2.1	0.17	0.93
2	17.2	0.11	-1.4	0.31	1.41
3	15.8	1.16	-0.08	0.50	0.53
4	15.8	1.40	1.8	0.10	-0.55
5	16.1	1.01	1.07	-0.43	0.36
6	16.6	0.95	2.6	-0.25	0.90
7	18.7	1.00	3.1	0.76	-0.08

as the three orientation angles, the same for both locations. A 7-link manipulator is synthesized for the situation, with fixed base point locations as  $(0, -30, 30)$ . Table 5 contains the synthesis solution and pictorial view of the corresponding configurations is presented in Figure 10(b).

## 8. Conclusion

In this work, the *variation* in the working environments for a service robotic arm is dealt with through a general

methodology for a constrained workcell. For this purpose, least constraints are applied at the number of dof and over the values of the robotic parameters. A bilevel optimization problem is formulated, involving the minimization of the number of degrees of freedom required to work in the given environment and for the given tasks (positions and orientations). Binary search algorithm is implemented for the outer layer of the optimization problem, which solves the unidirection problem in number of dof. In each of the outer iterations, a nonlinear optimization problem is solved for dimensional synthesis. The proposed methodology is validated through the reduction of the number of dof required for an environment, reported in earlier works. The importance of *flexibility* in robotic parameters is illustrated through two different cases in a realistic workcell of a computer server room where a robotic assistance is required for maintenance.

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

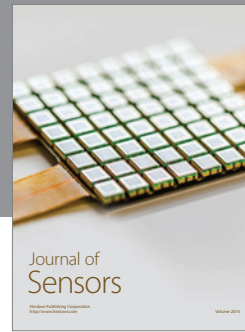
Authors gratefully acknowledge the financial support of Department of Science and Technology (DST), Government of India, for this work.

## References

- [1] G. Yang and I.-M. Chen, "Task-based optimization of modular robot configurations: minimized degree-of-freedom approach," *Mechanism and Machine Theory*, vol. 35, no. 4, pp. 517–540, 2000.
- [2] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network," *IEEE*

*Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 752–759, 2004.

- [3] I.-M. Chen, “Modular robots,” in *Handbook of Manufacturing Engineering and Technology*, pp. 2129–2168, Springer, 2015.
- [4] Z. M. Bi, W. J. Zhang, I.-M. Chen, and S. Y. T. Lang, “Automated generation of the D–H parameters for configuration design of modular manipulators,” *Robotics and Computer-Integrated Manufacturing*, vol. 23, no. 5, pp. 553–562, 2007.
- [5] E. Singla, S. Tripathi, V. Rakesh, and B. Dasgupta, “Dimensional synthesis of kinematically redundant serial manipulators for cluttered environments,” *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 585–595, 2010.
- [6] S. Rubrecht, E. Singla, V. Padois, P. Bidaud, and M. de Broissia, “Evolutionary design of a robotic manipulator for a highly constrained environment,” in *New Horizons in Evolutionary Robotics*, S. Doncieux, N. Bredèche, and J.-B. Mouret, Eds., vol. 341 of *Studies in Computational Intelligence*, pp. 109–121, Springer, 2011.
- [7] S. Patel and T. Sobh, “Task based synthesis of serial manipulators,” *Journal of Advanced Research*, vol. 6, no. 3, pp. 479–492, 2015.
- [8] M. Brandstötter, A. Angerer, and M. Hofbaur, “The curved manipulator (cuma-type arm): realization of a serial manipulator with general structure in modular design,” in *Proceedings of the 14th IFToMM World Congress*, pp. 403–409, October 2015.
- [9] J. Craig, *Introduction to Robotics: Mechanics and Control*, Addison-Wesley Longman Publishing, Boston, Mass, USA, 1989.
- [10] D. E. Knuth, “Optimum binary search trees,” *Acta Informatica*, vol. 1, no. 1, pp. 14–25, 1971.
- [11] K. Deb, *Optimization for Engineering Design: Algorithms and Examples*, PHI Learning, 2012.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

