2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Using Task Descriptions for Designing Optimal Task Specific Manipulators

Sarosh H. Patel and Tarek M. Sobh

Abstract — **Computing the optimal geometric structure of manipulators is one of the most intricate problems in contemporary robot kinematics. Industrial robotic manipulators are designed and built to perform certain predetermined tasks. It is therefore important to incorporate such task requirements during the design and synthesis of the robotic manipulators. Such task requirements and/or performance constraints can be specified in terms of the required end-effector positions, orientations along the task trajectory. In this work, we define, develop and test a methodology that can generate optimal manipulator geometric structures based on the task requirements. Another objective of this work is to guarantee task performance under user defined joint constraints. Using this methodology, task-specific optimal manipulator structures can be generated that guarantee task performance under a set of operating constraints.**

Index Terms— Task Descriptions, Reverse Prototyping, Simulated Annealing, Task Specific Design

I. Introduction

THE rapid growth in manufacturing technologies has increased the need for design and development of optimal machinery. No longer is the emphasis on machinery that works but on machinery that works faster, consumes less power, and is more functional. Designing optimal machinery and processes has become a necessary criterion across all engineering disciplines. The availability of computing power allows us to design and evaluate multiple structures based on user defined criteria and then select the best. In this work we propose a method for designing optimal robotic manipulator structures.

What is the best manipulator configuration for soldering electronic components? What should be the ideal manipulator structure for a painting job? What is optimal manipulator configuration for a material handling job?

S. H. Patel is with the University of Bridgeport, Bridgeport, CT 06604 USA (phone: 203-576-5555; e-mail: saroshp@bridgeport.edu).

T. M. Sobh, is with the University of Bridgeport, Bridgeport, CT 06604 USA (e-mail: sobh@bridgeport.edu).

Robotics researchers over the years have tried to find answers to these questions. But in this case plenty is the problem; there is no unique solution or definite answer to these questions. Instead, in most cases there can be infinite answers to any of the above questions. Equations describing the kinematic behavior of serial manipulators are highly nonlinear with no closed solutions. The difficulty in most cases lies not in finding a solution, but finding the 'best' solution out of the numerous possible solutions, or in other words, an optimal solution.

The field of industrial robotic manipulator design can be broadly classified into general purpose designs and task specific designs. Even though general purpose manipulators are commonplace, they do not guarantee optimal task execution. Since industrial robotic manipulators perform a given set of tasks repeatedly, task-specific or task-optimized manipulator designs are preferred for industrial applications.

The goal of this work is to develop a methodology that can serve as a simple and fast tool for synthesis of robotic manipulators based on task descriptions. The proposed methodology allows a user to enter the task point descriptions and joint constraints, and generates the optimal manipulator geometry (Denavit–Hartenberg (DH) parameters) for the specific task.

II. Existing Approaches & Limitations

The existing methods/techniques for design optimization of serial manipulators can be classified into the following three broad approaches:

A. *Geometric Approach* - Serial robotic manipulators are simple open-loop kinematic chains consisting of interconnected joints and links. The principles of closed loop mechanical chains can be applied to design highly dexterous serial manipulators by assuming the distance between the base of the manipulator and the task point as a fixed and imaginary link in the closed mechanical chain. Link mobility laws can then be used for optimizing the design of manipulators.

Grashof [1] proposed a simple rule to judge the mobility of links in four-link closed kinematic chains. This rule was further extended and developed into Grashof's criterion by Paul [2]. Researchers have applied Grashof's criterion to design manipulators with high dexterity at the given task points. Where dexterity refers to the ability of the

manipulator to attain any orientation about a given point [3]. In [4], [5], authors proposed a method for the optimal design of three-link planar manipulators using Grashof's criterion. In [5] a simple algorithm for the optimal design of three link planar manipulators with full manipulator dexterity at the given task region or trajectory is proposed. Ting introduced the five-link Grashof criterion [6] and extended it to N-link chains [7], [8].

B. *Parametric Optimization Approach* - Parametric optimization techniques are widely adopted for the synthesis and structural optimization of serial manipulators. Parametric optimization is a classical way of solving an optimization problem. One or more criteria that quantify the performance properties of the manipulator, sometimes with associated weighting factors, are maximized or minimized to arrive at a set of optimal design parameters. Condition number was used by Angeles and Rojas to obtain optimal dimensions for a three-DoF manipulator and three-DoF spherical wrist [9]. Craig and Salisbury used the condition number of the Jacobian as design criterion to optimize the dimensions of the fingers of the Stanford articulated hand [10]. In [11], optimal kinematic synthesis of the manipulator structures based on the Yoshikawa manipulability ellipsoid at a given set of task points is presented. Kucuk and Bingul, [12], [13], implement a multi-variable optimization. The manipulator workspace was optimized based on a combination of local and global indices: Structural length index, manipulability measure, condition number, and global conditioning index.

C. *Task-Based Design Approach* - Task-based design of manipulators uses the prior knowledge of application of the manipulator to design the best possible structure that can guarantee task completion. Task specifications can either be kinematic or dynamic. The ultimate goal of task-based design approach is to be able to generate both the manipulator kinematic and dynamic parameters, using the task descriptions and operating constraints [14].

Paredis and Kholsa [15], use the task requirements to find the optimal structure of an all revolute manipulator. Their proposed method involves generating the DH parameters by minimizing an objective function using numerical optimization. However, this method does not verify the structures for non-singular postures at the given task points. In [16], Al-Dios, et al., proposed a method for optimizing the link lengths, masses and trajectory parameters of a serial manipulator with known DH table using direct non-gradient search optimization. Dash, et al. [17], propose a two stage methodology for structure and parameter optimization of reconfigurable parallel manipulator systems. They propose a *'TaskToRobot Map'* database that maps task description to a suitable manipulator configuration depending on the degrees of freedom required for a given task.

D. *Limitations* – Geometric optimization methods are limited to special cases, where certain link mobility laws such as the Grashof's criterion can be applied to design highly dexterous manipulators. This methodology cannot be generalized or extended to design manipulators with prismatic links as an example. Also, this approach does not allow the designer to input multiple task requirements and operating constraints hence task satisfaction cannot be guaranteed using this approach.

The major drawback of the parametric optimization approach lies in the limited scope of the performance parameters themselves. A comprehensive survey of manipulator performance measures and their limitations can be found in [18]. While this approach might be useful to fine tune existing manipulators configurations to improve manipulator's specific parametric performance, for example generating isometric manipulators by adjusting their link lengths, this approach has not evolved into a complete design methodology for manipulators because it only improves upon an existing structure and cannot propose or generate new kinematic structures.

A common limitation of the above two methods is that existing methods do not consider practical design issues such as constrained joint limits. Therefore, they are task independent and hence do not guarantee the non-existence of a better manipulator for a specific task [15]. Also, most methods avoid dealing with prismatic joints, especially the parametric optimization methods.

Since industrial manipulators are expected to do certain tasks repeatedly it is essential that the task requirements are incorporated with in the design process, so that satisfactory task performance is guaranteed. Task based design is a promising avenue for developing a comprehensive manipulator design methodology. Firstly, because it is by definition based on specific task requirements, and secondly because, multiple criteria and constraints can be specified by the designer. The major limitation of existing task-based methodology is that they are limited to the design of manipulators composed of only revolute links, this is in part also due to the inverse kinematic problem.

In this work, we define the necessary and sufficient conditions for a holistic task-based design methodology. Next, we define a function to judge the reachability of a manipulator configuration to the task point(s). This methodology can generate manipulators composed of both revolute as well as prismatic joints. Finally, in the results section we demonstrate the working of this methodology by generating manipulator configurations for real world industrial applications. This methodology can also be used to optimize the structure of an existing manipulator for a specific task.

III. Problem Statement

Though the criteria for optimizing a manipulator can be infinite, in defining a methodology for fast synthesis based on task descriptions, we begin with a set of minimum kinematic performance criteria. The manipulator's ability to easily reach every task point and be able to attain the required orientations at these task points without being in a singular pose is a necessary requirement for any robotic application.

Let's consider a manipulator task that requires the manipulator to reach certain task point with specific orientations. Fig. 1 is an example of such a manipulator task where the end-effector is required to have multiple orientations about a set of task points.

The task descriptions can be specified in terms of the task points *p* that the manipulator is supposed to reach with a specified orientation. Let *P* be the set of *m* task points that define the manipulator's performance requirements. All these points belong to the six-dimensional Task Space (*TS*) that defines both the position and orientation of the manipulator's end-effector. Such that:

$$P = p_i = \{x, y, z, \varphi, \theta, \psi\} \forall \, \mathrm{i} = 1, 2, \ldots, \mathrm{m} \in \mathrm{TS}$$

where *x,y,z* are the real-world coordinates, and φ, θ, ψ are the roll, pitch and yaw angles about the standard Z, Y and X – axis.

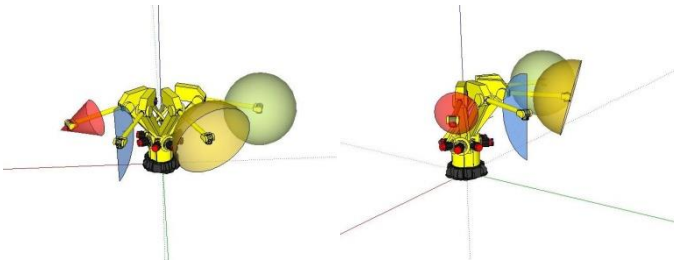

Fig. 1. Example of a manipulator task requirement

Similarly, for an *n* degree of freedom manipulator, the joint vector *q* can be a said to be a point in the *n* ndimensional Joint Space *(Q)*, such that: $q = [q_1, q_2, \ldots, q_n] \in Q$

Each joint vector *q* represents unique manipulator posture and a distinct point in the *n* dimensional Joint Space *(Q)*. The Joint Space assumes there are no joint limitations (fully revolute ideal joints). But in practice the joints are not fully revolute and are bounded by lower and upper bounds. The values of the joint angles are range bound by user defined joint limits (upper and lower bounds). Hence, we define Q_c as the Constrained Joint Space, such that the joint displacements always satisfy the constraints:

$$q_{i,\min} \le q \le q_{i,\max} \; (q_i \in Q_c)$$

When a given manipulator of configuration set *DH,* with joint vector *q* can reach a specific task point *p*, the forward kinematic mapping can be represented as: $f(DH, q) = p$

Hence the problem of finding the right manipulator configuration can be stated as - Find a solution set *DH* in the *3n* dimensional Configuration Space such that there exists at least one *q* in the Constrained Joint Space that can reach the required position and orientation of the end-effector, i.e.,

$$\textit{Find all DH such that } \forall p \in TS; \exists q \in Q_c \mid f(DH, q) = p$$

Even though this might seem to be a necessary and sufficient condition required for designing a manipulator, simulations and experience suggest that this solution set might include some manipulator configurations that are able to reach the one or more of the task points only in singular positions. Such manipulators, if constructed, will not be able to attain good end-effector velocities in one or more directions due to their singular postures at the task point(s) and therefore will have very limited mobility at the required task point(s). Infinite forces have to be applied in order to generate motion along one or more directions at singularities. Therefore such manipulator configurations should be removed from the solution set. We know the necessary and sufficient condition for the existence of a singularity is:

$$\sqrt{\det(J(q).J(q)^T)} = 0$$

For a manipulator with a square Jacobian, as in this case, the equation can be reduced to, $\det(J(q)) = 0$. Therefore the problem can be restated as:

$$\textit{Find all DH such that } \forall \mathrm{p} \in \mathrm{TS}; \exists \mathrm{q} \in \mathrm{Q_c} | \mathrm{f(DH, q)} = \mathrm{p} \text{ and } \det(\mathrm{J(q)}) \neq 0$$

IV. Mathematical Formulation

To determine if the manipulator is able to reach a given task point with required orientation we construct a reachability function. The reachability function determines if the manipulator can reach and orient the end-effector at the task point without violating the joint constraints.

$$reachability(DH) = max\left[min\left(\frac{(q_{i,max} - q_i)(q_i - q_{i,min})}{\left(0.5(q_{i,max} - q_{i,min})\right)^2} \right)_{i=1}^{n} \right]_{j=1}^{g}$$

where *g* is the number of inverse kinematic solutions.

The reachablility function value for different locations of the task point is shown in Table 1. The reachability function will have a maximum value of unity if the manipulator reaches the task point with all joint displacement being mid-range of their joint limits. A reachability value of unity is the ideal case and is only possible with one task point.

We extend the above formulation for reachability to include all *m* points that define the Task Space, as a summation of the reachability function values at each of the individual task points.

$$reachability(DH) = \sum_{\forall p \in TS} \left(max\left[min\left(\frac{(q_{i,max} - q_i)(q_i - q_{i,min})}{\left(0.5(q_{i,max} - q_{i,min})\right)^2} \right)_{i=1}^{n} \right]_{j=1}^{g} \right)$$

To convert the reachablility function into general optimization problem, such that minimizing it will yield optimal solution we add a negative sign. The function then becomes:

$$reachability(DH) = -\sum_{\forall p \in TS} \left(max\left[min\left(\frac{(q_{i,max} - q_i)(q_i - q_{i,min})}{\left(0.5(q_{i,max} - q_{i,min})\right)^2} \right)_{i=1}^{n} \right]_{j=1}^{g} \right)$$

When multiple task points constitute a task goal the reachablility function will have many local minima. This should be kept in mind while selecting a proper optimization algorithm. Using local minimization routines will only yield acceptable solutions but not the optimal solution. Only global minimization routines can deliver an optimal solution to this problem.

V. Solution Methodology

To simplify the problem we make the following assumptions:

1) The robot base is fixed and located at the origin O(0, 0,0).
2) The task points are specified with respect to the manipulator's base frame.
3) If a joint is prismatic, the joint angle (θ) can assume values in the interval [-180, 180].
4) If a joint is revolute, the joint twist angle (α) can assume values [-180, 180].
5) The joint limitations are known to the designer.
6) The last three axes of the six degree of freedom manipulator intersect at a point to form a spherical wrist.
7) To limit the number of inverse kinematic solutions only non-redundant configurations are considered.

TABLE I: Reachability Function Values

Location of the Task Point 'p'	Reachability Function Value
When p is inside the workspace and at least one solution is within joint constraints	[0, 1]
When p is inside the workspace and the best solution has at least one of the joint angles at its extreme position	0
When p is inside the workspace and the best solution is one with all joints displacements mid-range	1

A. Simulated Annealing Algorithm

Many optimization algorithms are available to solve a given global optimization problem. But the choice of the algorithm greatly depends on factors such as the dimensionality of the problem, the nature of the variables (discrete or continuous), availability of a function derivative, etc. A good global optimization method for a given problem can only be found by matching the features of the problem to the algorithm characteristics and its problem handling capabilities.

In this case, the objective or cost function - which is the reachability function - does not have a direct analytical expression, and is computationally expensive to calculate as it in turn depends on calculating the inverse kinematic solutions. It is also important to note here that this problem does not have a formulation for a function derivative nor any function gradient data is available. The objective function will have multiple local and global minima points where the function value attains the desirable value. The search space is also very exhaustive. Keeping in mind the above factors we chose to optimize the problem using Simulated Annealing (SA) algorithm.

Simulated annealing was developed in the 1980s by Scott Kirkpatrick [19] based on a statistical algorithm developed much earlier by Metropolis [20], to improve designs of Integrated Circuit (IC) chips by emulating the actual process of annealing. Simulated Annealing (SA) is a generic probabilistic meta-heuristic algorithm for finding the global minimum of a cost function that has many local minima. The SA algorithm uses randomly generated inputs based on a probabilistic model. Only under certain conditions is a change in the objective function due to a new random input accepted. The acceptance condition for a new input is given as follows:

$$\Delta f_{obj} \leq 0$$

$$\exp\left(-\frac{\Delta f_{obj}}{T}\right) > random[0,1)$$

where Δf_{obj} is the change in the objective function and T is the temperature of the algorithm.

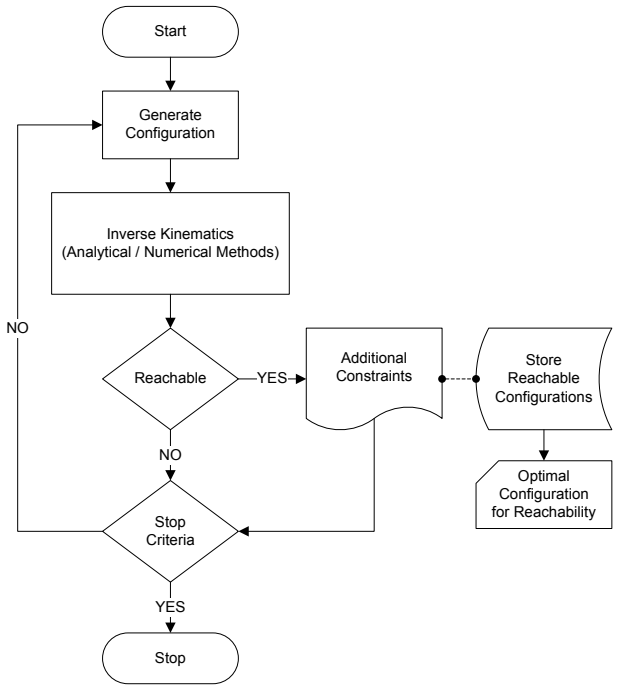

Fig. 2: Proposed design methodology flow chart

Starting with a high temperature, the algorithm, with every iterative step, gradually lowers the temperature simulating the actual annealing process. And, after every fixed number of iterations known as the annealing period, the temperature is raised back again. Higher temperatures mean greater randomization of the input variables. Therefore, a slow annealing method that lowers the temperature gradually will explore the search space to a

greater extent that a fast annealing method that lowers the temperature quickly. At lower temperatures the search space is exploited while at high temperature the algorithm explores the search space.

The algorithm stops when there is no change in the objective function for a certain number of consecutive inputs. SA algorithm remembers the best inputs throughout its run. SA also works well with high dimensionality problems and even when the search space is very extensive.

Fig. 2 shows the flow chart of the proposed methodology. Random configurations are generated by the SA algorithm and tested for the existence of the inverse solutions within the joint limits range. In case a solution exists within the joint constraints, we further test the configurations for singular postures. Every reachable structure found is stored, and the best reachability structure is constantly updated.

B. Inverse Kinematic Module

This methodology works well with both analytical and numerical inverse kinematic modules. If analytical methods such as Generalized Inverse Kinematics (GIK) [21], [22], [23], [24] is used we need to add another term in the reachability function. This is because the GIK method yields complex solutions for points that lie outside the reachable workspace. To eliminate such configurations the reachability function is modified as follows:

$$reachability(DH) = -\sum_{\forall p \in TS}\left(max\left[\sum_{i=1}^{n} -imag\left(q_i\right)^2 + min\left[\left(\frac{\left(real\left(q_i\right)-q_{i,min}\right)\left(q_{i,max}-real\left(q_i\right)\right)}{\left(0.5\left(q_{i,max}-q_{i,min}\right)\right)^2} \right)_{i=1}^{n} \right]_{j=1}^{g} \right)$$

Not all manipulator configurations have closed form solutions hence numerical approaches have to be used in such cases. In this implementation we use a novel numerical approach for calculating the inverse kinematic solutions based on Particle Swarm Optimization (PSO) [25] algorithm. Due to the limited length of the paper this method is not described in this paper.

VI. Experimental Results

In this section we test the proposed methodology to design manipulators based on task point descriptions. For a prismatic link the joint limit is constrained between zero and unity. The joint constraints for the revolute joints are set as follows:

Lower Bound = [-160, -45, -225, -110, -100, -266]

Upper Bound = [160, 225, 45, 170, 100, 266]

Due to the limited length of the paper only structures optimized for best reachability at the task points have been discussed here. However, with a few additional criteria this methodology can also identify optimal structures for best kinematic performance and least power consumption. In the task examples below we merely mention the structure types identified for best kinematic performance and least power consumption.

A. Spherical Task

The spherical goal task is an example of an inspection task where the manipulator needs to achieve different orientations about a given task point in three dimensional space. At the same time it also demonstrates that dexterous manipulators (manipulators that can achieve all possible orientations about a given task point) can be designed using this approach.

In this task the manipulator is required to have the ability to reach a task point from all possible angles. This task involves approaching a point from six different angles separated by 90 degrees, such that they represent the three perpendicular body diagonals of a sphere perpendicular. The task points for a spherical task are given below and the task visualization can be seen in Fig. 3. The red lines within the sphere represent the orientation the end-effector is expected to have while reaching the center of the sphere.

```
Spherical Task = [
        0 0.75 0  0       0       0;
        0 0.75 0 -3.142   0      -3.142;
        0 0.75 0  0       1.565   0;
        0 0.75 0  0      -1.565   0;
        0 0.75 0 -1.372   1.541  -3.142;
        0 0.75 0  1.784  -1.571  -0.213
          ];
```

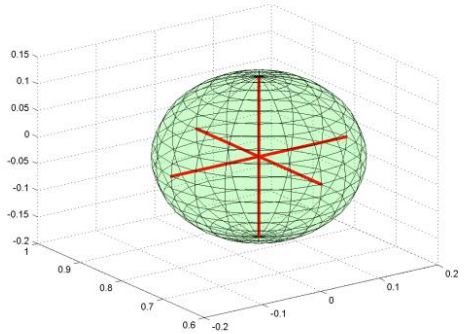

Fig. 3: Task visualization of the spherical task

Upon implementing the proposed methodology for the spherical task, we found that the best configuration for this task with respect to reachability at the task points was a RRR-RRR manipulator. Multiple runs of the algorithm came up with the same RRR-RRR manipulator as the most suited structure for this specific task. While configurations were also able to meet the task requirements with lower reachability function value, no RPP-RRR configuration could complete the task with the given of set constraints. The highest reachability function value of 0.544 was

attained by the manipulator with the DH table shown below –

robot (6 axis, RRRRRR, stdDH)

j	theta	d	a	alpha
1	q1	0.9979	0.8345	-2.375
2	q2	0.7467	0.9979	3.101
3	q3	0.0025	0.9978	2.269
4	q4	0	0	-1.571
5	q5	0	0	1.571
6	q6	0.25	0	0

Superimposed positions of the manipulator executing the task and reaching all the task points with the required orientations, is shown in Fig. 4.

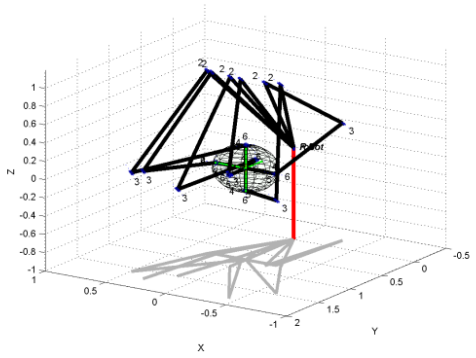

Fig. 4: Manipulator reaching the task points of the spherical task

As expected the best configuration is a RRR-RRR structure, however the joint twist angles are non-intuitive and not conventional. This unique arrangement of the links with the joint twist respect to each other gives the structure the high reachability and kinematic performance. For this task even the best kinematic structure as well as the least power consuming structure was all revolute (RRR-RRR).

B. Horizontal Plane Task

In this task the manipulator is required to have a constant vertically downward orientation about nine points on a horizontal plane. This task mimics a job that industrial manipulators commonly have in the packaging industry, where the manipulator could be either applying labels to the individual products or inspecting individual products in a box for quality control.

This task comprises of nine points that lie in a horizontal plane, the manipulator is supposed to reach all of the task points with the same orientation. The task points for the horizontal plane task are given below and the task visualization can be seen in Fig. 5

```
Horizontal Plane Task = [
                        0.9  -0.5  0 -3.142 0 -3.142;
                        0.9   0    0 -3.142 0 -3.142;
                        0.9   0.5  0 -3.142 0 -3.142;
                        0.7  -0.5  0 -3.142 0 -3.142;
                        0.7   0    0 -3.142 0 -3.142;
                        0.7   0.5  0 -3.142 0 -3.142;
                        0.5  -0.5  0 -3.142 0 -3.142;
                        0.5   0    0 -3.142 0 -3.142;
                        0.5   0.5  0 -3.142 0 -3.142;
                        ];
```

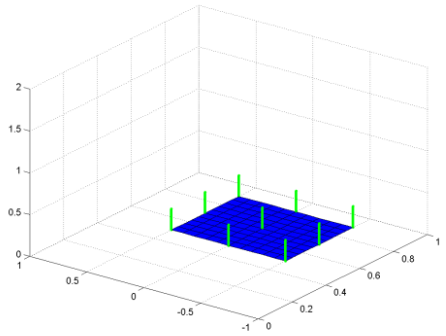

Fig. 5: Task visualization of the horizontal plane task

Just as in the previous task example for this horizontal plane task too, the best reachable configuration for the nine points in this task was a RRR-RRR configuration, with a maximum reachablitly function value of 0.68127. A close second was a RPP-RRR configuration. The DH table for the optimal reachable structure for this task is shown given –

robot (6 axis, RRRRRR, stdDH)

j	theta	d	a	alpha
1	q1	0.2472	0.6404	0.104
2	q2	0.0019	0.6147	1.404
3	q3	0.3707	0.3709	-1.135
4	q4	0	0	-1.571
5	q5	0	0	1.571
6	q6	0.25	0	0

Superimposed positions of the optimal manipulator executing the task and reaching all the task points with the required orientations, is shown in Fig. 6

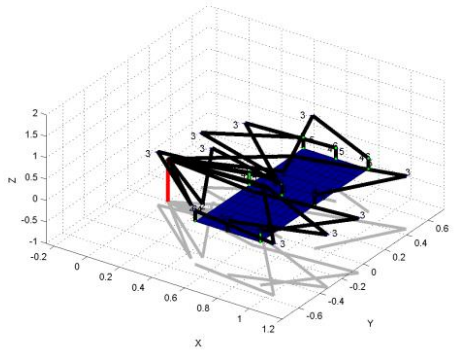

Fig. 6: Manipulator reaching the task points of the horizontal plane task

The optimal structures for both reachability and best kinematic performance were found to be RRR-RRR, but the optimal configuration for least power consumption was identified as an RPP-RRR configuration. This example demonstrates the ability of this methodology to evaluate configurations that include prismatic joints as well. It is important to note here that the in case of a prismatic link the joint limits are set in terms of the allowable linear displacement of the joint. The methodology therefore generates the optimal values for the joint offset, joint twist and link offset.

C. Optimizing Puma560 for a Cone Task

In this example we optimize the Puma560 manipulator's structure using this methodology and compare it with the original manipulator for reachablility. This also demonstrates the ability of the methodology to optimize existing manipulator configurations based on the task descriptions. For this purpose we define a small cone task involving four task points, such that the manipulator orientations at these task points form a conic section. The task points for a cone section goal are as follows:

```
Cone Goal = [
                0 0.7 0 -3.142 1.162 -3.142;
                0 0.7 0  0       1.131  0;
                0 0.7 0 -1.028 0.383 -1.393;
                0 0.7 0 -1.873 0.845 -1.557;
            ];
```

When this original Puma560 was applied to the cone task a reachability function value of -0.248 was achieved. This is clearly a very low value for the reachability indicating that the manipulators joints are close to their limits when reaching the task points.

Next, the proposed methodology is applied to optimize the link lengths of a Puma560 manipulator for better reachability. We only optimize the link lengths and the link offsets and the rest is kept the same as the original Puma560 manipulator. Even the joint constraints too are kept the same as the original manipulator. The DH parameter table of the optimized Puma is shown below.

Puma Mod (6 axis, RRRRRR, stdDH)

j	theta	d	a	alpha
1	q1	0	0	1.571
2	q2	0	0.7299	0
3	q3	0.0207	0.4572	-1.571
4	q4	0.0007	0	1.571
5	q5	0	0	-1.571
6	q6	0.25	0	0

Fig. 7 shows the Puma and optimized Puma manipulator in their home positions. When this manipulator is applied to the task of a cone section presented above we get an improved reachability function value of -0.76187. Fig. 8 shows superimposed positions of the Puma and Optimized Puma performing the cone task. In this case the structural length the optimized Puma is larger than the original Puma manipulator.

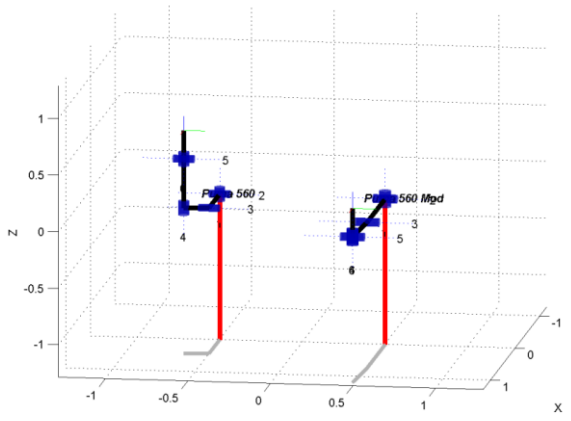

Fig. 7. Puma and Optimized Puma in their home position

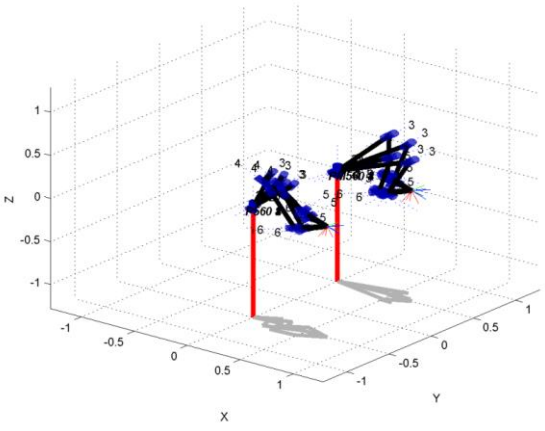

Fig. 8. Superimposed positions of the Puma and Optimized Puma performing the cone task

VII. DISCUSSION

In all the task experiments the initial seed to the algorithm was a set of random values such that the resultant configuration did not constitute any existing structure and did not reach even a single task point. The methodology then iteratively found a set of reachable configurations from which task suitable configurations are selected.

As expected for most of the tasks, the best manipulator structure found happened to be a RRR/RRR manipulator. This supports the fact that most industrial manipulators are RRR robots with spherical wrists as they provide better reachability at the task points and also the ability to orient the end-effector arbitrarily in the workspace.

The manipulator structures that were generated by the methodology for each of the tasks are not ones that would intuitively come to mind for those tasks. Using this task based tool to design manipulators can help the designer in evaluating new and different configurations.

In some cases a few structures failed to reach all the task points with the necessary orientation required for task completion. For example no RPP/RRR configuration could be found that could successfully complete the sphere goal task within the set joint constraints.

VIII. CONCLUSION

In this work we have presented a general methodology for task-specific prototyping of non-redundant serial robotic manipulators. This framework can be used to generate specialized manipulator structures based on the task descriptions and operating constraints. The framework allows for practical joint constraints to be imposed during the design stage of the manipulator. This methodology incorporates the necessary criteria for the design of a manipulator, such as reachability, orientation and non-singularity. However any additional sufficient condition(s) can be specified by the user, for example kinematic performance or power consumption. Using a set of practical task requirements and constraints we have generated the manipulator configurations such that the task performance is guaranteed even under the imposed joint constraints. Also, existing manipulator structures can be tuned for improved task performance using this methodology.

This methodology works well with both analytical and numerical inverse kinematics module. This work can be viewed as part of a broader program to develop a general framework for the reverse prototyping of robotic manipulators based on task descriptions and operating constraints.

REFERENCES

[1] F. Grashof, "Thertische Mshinenlehre," *Leipzig,* pp. 113-183, 1883.

[2] B. Paul, "A reassessment of Grashof's criterion," *Journal of Mechanical Design, Transactions of the ASME,* vol. 101, pp. 515-518, 1979.

[3] R. Vijaykumar, K. J. Waldron, and M. Tsai, "Geometric optimization of serial chain manipulator structures for working volume and dexterity," *The International Journal of Robotics Research,* vol. 5, pp. 91-103, 1986.

[4] R. Li and J. S. Dai, "Orientation angle workspaces of planar serial three-link manipulators," *SCIENCE IN CHINA SERIES E: TECHNOLOGICAL SCIENCES,* vol. 52, pp. 975-985, 2009.

[5] S. Patel and T. Sobh, "Optimal Design of Three-link Planar Manipulators Using Grashof's Criterion," in *Prototyping of Robotic Systems: Applications of Design and Implementation*, T. Sobh and X. Xiong, Eds., illustrated ed USA: IGI Global, 2012, p. 321.

[6] K.-L. Ting, "Five-Bar Grashof Criteria," *Journal of Mechanisms Transmissions and Automation in Design,* vol. 108, pp. 533-537, 1986.

[7] K.-L. Ting, "Mobility Criteria of Single-Loop N-Bar Linkages," *Journal of Mechanisms Transmissions and Automation in Design,* vol. 111, pp. 504-507, 1989.

[8] K.-L. Ting and Y.-W. Liu, "Rotatability Laws for N-Bar Kinematic Chains and Their Proof," *Journal of Mechanical Design,* vol. 113, pp. 32-39, 1991.

[9] J. Angeles and A. Rojas, "Manipulator inverse kinematics via condition-number minimization and continuation," *INT. J. ROBOTICS AUTOM.,* vol. 2, pp. 61-69, 1987.

[10] J. K. Salisbury and J. J. Craig, "Articulated Hands: Force Control and Kinematic Issues," *The International Journal of Robotics Research,* vol. 1, pp. 4-17, March 1, 1982 1982.

[11] T. M. Sobh and D. Y. Toundykov, "Optimizing the tasks at hand [robotic manipulators]," *Robotics & Automation Magazine, IEEE,* vol. 11, pp. 78-85, 2004.

[12] S. Kucuk and Z. Bingul, "Robot Workspace Optimization Based on a Novel Local and Global Performance Indices," in *Industrial Electronics, 2005. ISIE 2005. Proceedings of the IEEE International Symposium on*, 2005, pp. 1593-1598.

[13] S. Kucuk and Z. Bingul, "Comparative study of performance indices for fundamental robot manipulators," *Robotics and Autonomous Systems,* vol. 54, pp. 567-573, 2006.

[14] J. Kim and P. K. Khosla, "A formulation for task based design of robot manipulators," in *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, 1993, pp. 2310-2317 vol.3.

[15] C. J. J. Paredis and P. K. Khosla, "Kinematic Design of Serial Link Manipulators From Task Specifications," *The International Journal of Robotics Research,* vol. 12, pp. 274-287, June 1, 1993 1993.

[16] H. Al-Dois, A. K. Jha, and R. B. Mishra, "Task-based design optimization of serial robot manipulators," *Engineering Optimization,* vol. 45, pp. 1-12, 2012.

[17] A. K. Dash, I. M. Chen, S. H. Yeo, and G. Yang, "Task-oriented configuration design for reconfigurable parallel manipulator systems," *International Journal of Computer Integrated Manufacturing,* vol. 18, pp. 615-634, 2005/10/01 2005.

[18] S. Patel and T. Sobh, "Manipulator Performance Measures - A Comprehensive Literature Survey," *Journal of Intelligent & Robotic Systems,* pp. 1-24, 2014/02/15 2014.

[19] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, pp. 671-80, May 13 1983.

[20] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics,* vol. 21, pp. 1087-1092.

[21] D. E. Pieper and B. Roth, "The Kinematics of Manipulators Under Computer Control," in *2nd Int. Cong. on the Therory of Mach. and Mech.*, Zakopane, Poland, 1969, pp. 159-169.

[22] H. Y. Lee and C. G. Liang, "Displacement analysis of the general spatial 7-link 7R mechanism," *Mechanism and Machine Theory,* vol. 23, pp. 219-226, 1988.

[23] M. Raghaven and B. Roth, "Kinematic analysis of the 6R manipulator of general geometry," presented at the The fifth international symposium on Robotics research, 1990.

[24] L.-W. Tsai and A. P. Morgan, "Solving the Kinematics of the Most General Six- and Five-Degree-of-Freedom Manipulators by Continuation Methods," *ASME J. Mech. Trans. Automat. Des,* vol. 107, p. 12, 1985.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, 1995, pp. 1942-1948 vol.4.