# Goal Directed Design of Serial Robotic Manipulators

Sarosh Patel and Tarek Sobh

*Abstract* — **Computing the optimal geometric structure of manipulators is one of the most intricate problems in contemporary robot kinematics. 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 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-based optimal manipulator structures can be generated that guarantee task performance under set operating constraints.**

***Index Terms— Global Optimization, Reverse Prototyping, Simulated Annealing, Task-Based 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 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? 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 research area of 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. Because 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 for the specific task.

## II. Background Survey

The existing approaches for design and synthesis of serial manipulators can be broadly classified into the following three types:

### A. Geometric Approach

Serial robotic manipulators are 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.

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].

S. Patel is with the Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06605 USA (e-mail: saroshp@bridgeport.edu).

T. Sobh is with the Department of Computer Science and Engineering, University of Bridgeport, Bridgeport, CT 06605 USA (e-mail: sobh@bridgeport.edu).

978-1-4799-5233-5/14/$31.00 ©2014 IEEE

### B. Parametric Optimization Approach

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. Parametric optimization has been one of the widely adopted approaches for the synthesis of serial manipulators. 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.

These parametric optimization methods are task independent and hence do not guarantee the non-existence of a better manipulator for a specific task [14]. Another limitation of this approach is that it has a very limited scope due to the inherent limitations and general shortcomings of the performance metrics. A comprehensive survey of manipulator performance measures and their limitations is found in [15].

### 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 model is to be able to generate both the manipulator kinematic and dynamic parameters, using the task description and operating constraints [16].

Paredis and Kholsa [14], use the task requirements to find the optimal structure of a all revolute manipulator. Their proposed method involves generating the DH parameters by minimizing an objective function using numerical optimization. But, this method does not check for non-singular manipulator postures and the ability of the manipulator to generate effective velocities at the task points. In [17], 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. [18], 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.

## 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 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 $\varphi, \theta, \psi$ are the roll, pitch and yaw angles about the standard Z, Y and X – axis.

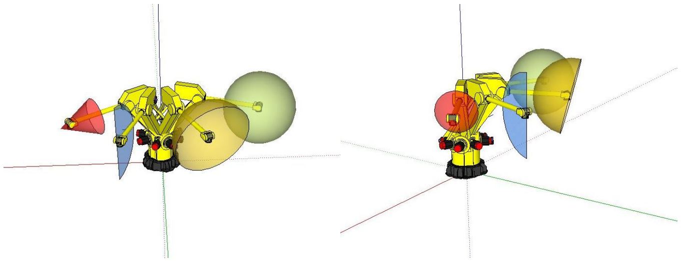


Fig. 1. Example of manipulator task requirement

Similarly, for an *n* degree of freedom manipulator, the joint vecto *q* rcan be a said to be a point in the *n* dimensional 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, $f(DH, q) = p$

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 will suggest that this solution set might include a few manipulators 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). Such manipulators 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. 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, f*ind all DH such that* $\forall p \in TS; \exists q \in Q_c \mid f(DH, q) = p \, and \det(J(q)) = 0$

## 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 within the set joint limitations.

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

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 function values at the individual task points.

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

To convert the function into general optimization problems, such that minimizing them will yield optimal solutions we add a negative sign. The function then becomes:

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

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 to find optimal solutions will yield acceptable solutions but not global solutions. Only global minimization routines will be able to deliver an optimal solution for this problem.

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 |

## 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).
2) The task points are specified with respect to the manipulator's base frame.
3) The joint limitations are known to the designer.
4) If a joint is prismatic, the joint angle (θ) can assume values in the interval [-180, 180].
5) If a joint is revolute, the joint twist angle (α) can assume values [-180, 180].
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.

### A. Simulated Annealing Algorithm

There are many approaches to solve a given global optimization problem. The choice of the algorithms greatly depends on factors such as the dimensionality of the problem, the nature of the variables (discrete or continuous), availability of a function derivative. 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 depends on the inverse kinematic solutions. It is also important to note here that this problem does not have a formulation for a function derivative or any function gradient data. 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 implement the problem using Simulated Annealing (SA) algorithm. The simulated annealing method is a heuristic algorithm.

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. All reachable structures are stored, and the best reachability structure is always updated.

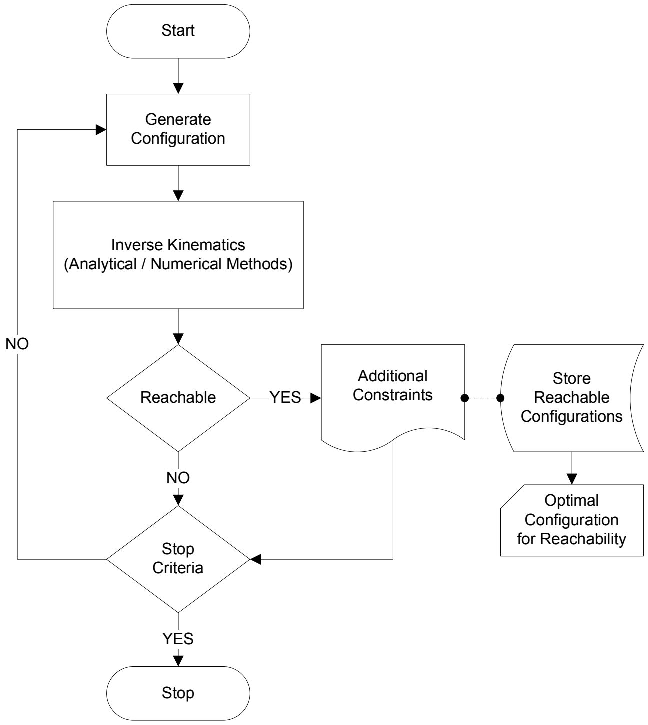

Start
Generate Configuration
Inverse Kinematics (Analytical / Numerical Methods)
NO
Reachable
YES
Additional Constraints
Store Reachable Configurations
NO
Optimal Configuration for Reachability
Stop Criteria
YES
Stop

Fig. 2: Methodology flow chart

Beginning with a high temperature the SA algorithm with every iterative step gradually lowers the temperature simulating the annealing process. And, after every fixed number of iterations, known as the annealing period, the temperature is back raised 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 iterations. SA algorithm remembers the best inputs throughout its run. SA also works well with high dimensionality problems even when the search space is extensive.

### B. *Inverse Kinematic Module*

This methodology works well with both analytical and numerical inverse kinematic modules. In this implementation we use a novel numerical approach for calculating the inverse kinematic solutions for a six degree-of-freedom manipulator. This new inverse kinematic approach uses the Particle Swarm Optimization (PSO) algorithm. Inverse kinematic solutions are found by decoupling the positioning and orienting (wrist) joint angles for the manipulator due to presence of the spherical wrist. This PSO based inverse kinematic module finds solutions that lie within the specified joint constraints, eliminating the need to reject solutions that do not lie within the joint constraints. This PSO inverse kinematic module is not discussed in this paper. Such a two stage optimization methodology is also commonly referred to as the *Greedy Optimization* approach.

## VI. 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 limit 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]

### A. *Spherical Goal*

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 diagonals of a sphere perpendicular to each other. The task points for a sphere goal are given below and can be seen in Fig. 3.

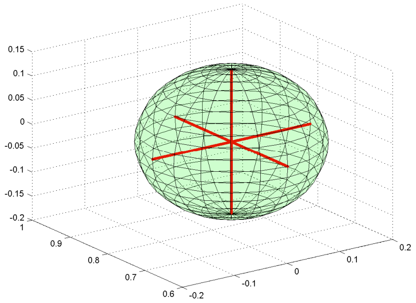
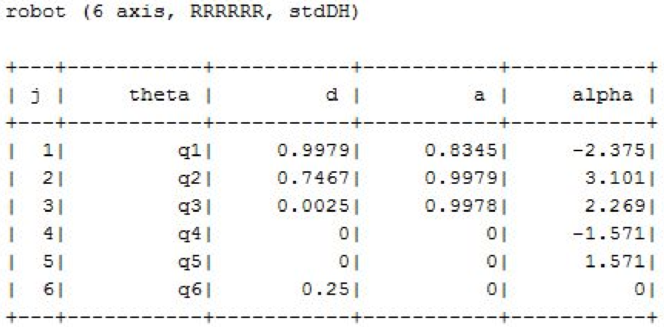


Fig. 3: Task requirements for the Spherical Goal

Based on the evaluations of all possible configurations, the best configuration that has the maximum overall reachability value for this set of points of the sphere is an RRR-RRR manipulator. This configuration has a reachability value of -0.544. The DH parameters of the manipulator are:



```
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.

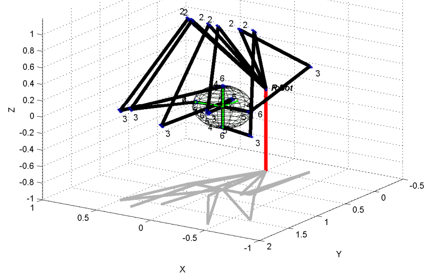


Fig. 4: Manipulator reaching the task points of the Sphere goal

### *B. Circular Ring Goal*

In this task the manipulator is required to reach eight points on the circumference of a circle with the same orientation at all the task points. The task points for the ring goal are given below and can be seen in Fig. 5

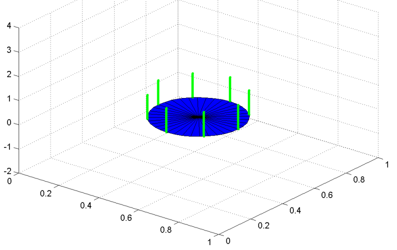
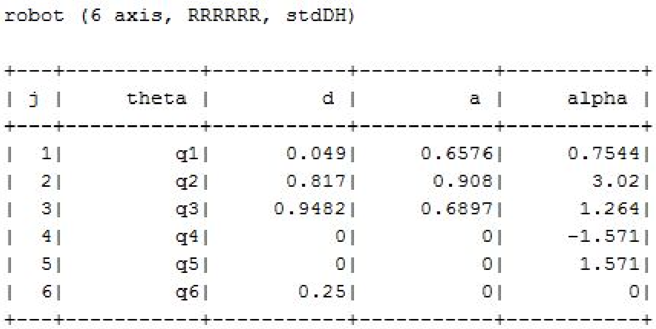


Fig. 5: Task requirements for the Circular Ring Goal

Based on the evaluations of all possible configurations, the best configuration that has the maximum overall reachability value for this set of points of the ring task is an RRR-RRR manipulator. This configuration has a reachability value of -0.833. The DH parameters of the manipulator are:



```
robot (6 axis, RRRRRR, stdDH)

+---+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |
+---+-----------+-----------+-----------+-----------+
|  1|         q1|      0.049|     0.6576|     0.7544|
|  2|         q2|      0.817|      0.908|       3.02|
|  3|         q3|     0.9482|     0.6897|      1.264|
|  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. 6

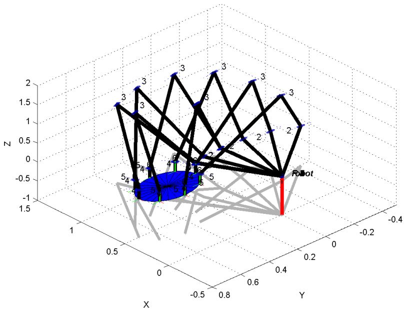


Fig. 6: Manipulator reaching the task points of the Ring Goal

### *C. Horizontal Plane Goal*

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. This task is similar to the task manipulators execute in the packaging / soldering application. The task points for the horizontal plane goal are given below. The task goal can be seen in Fig. 7

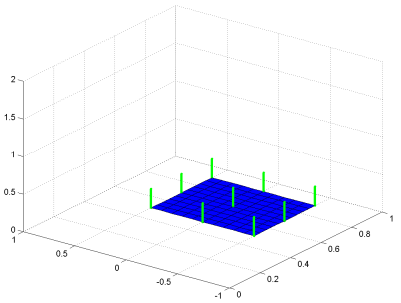
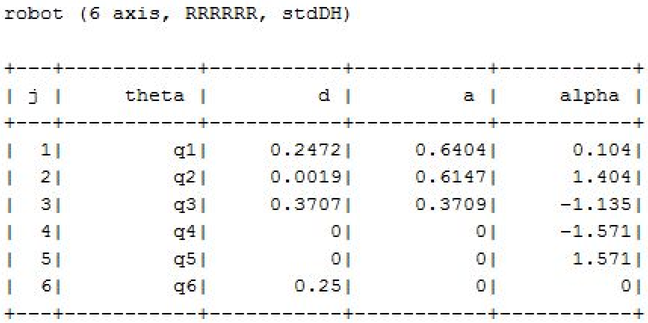


Fig. 7: Task requirements for the Horizontal Plane Goal

After evaluating of all possible configurations, the best configuration that has the maximum overall reachability value for this set of points of the horizontal goal is an RRR-RRR manipulator. This configuration has a reachability value of -0.68127. The DH parameters of the manipulator are:



```
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 manipulator executing the task and reaching all the task points with the required orientations, is shown in Fig. 8

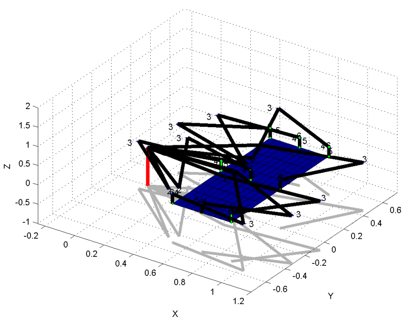


Fig. 8: Manipulator reaching the task points of the Horizontal Plane Goal

## 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 an 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-based prototyping of serial robotic manipulators. This framework can be used to generate specialized manipulator structures based on the task descriptions. The framework allows for practical joint constraints to be imposed during the design stage of the manipulator. This methodology can be used with an inverse kinematics module that can either be analytical, or numerical. 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] 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.

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

[16] 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.

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

[18] 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.