Task Based Design of Serial Robotic Manipulators

Sarosh Patel, Advisor: Prof. Tarek Sobh
Robotics, Intelligent Sensing & Control (RISC) Laboratory,
School of Engineering, University of Bridgeport.

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.

Introduction

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? The rapid growth in manufacturing technologies has increased the need for design and development of optimal machinery.
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. Figures 1-3 shows an example of an task-description.

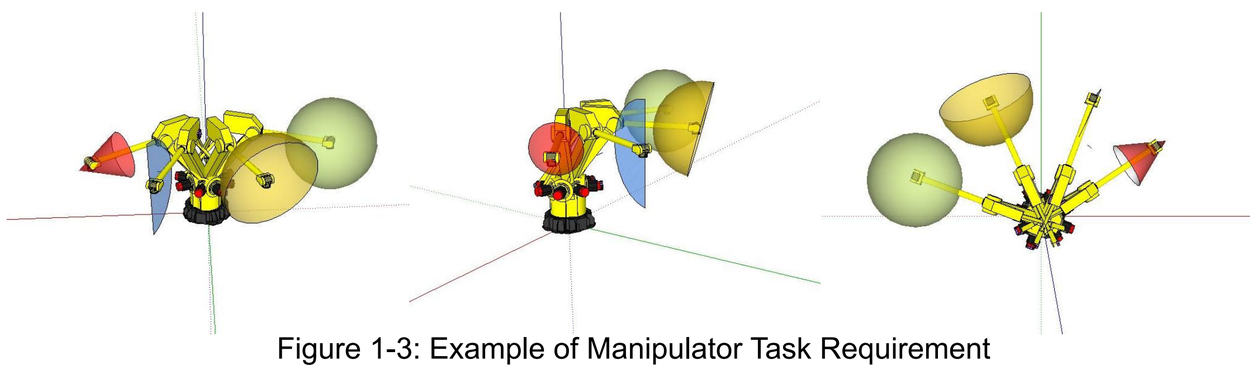

Figure 1-3: Example of Manipulator Task Requirement

Task Based Design

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.
Paredis and Kholsa in their work, 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. 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. 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.

Methodology

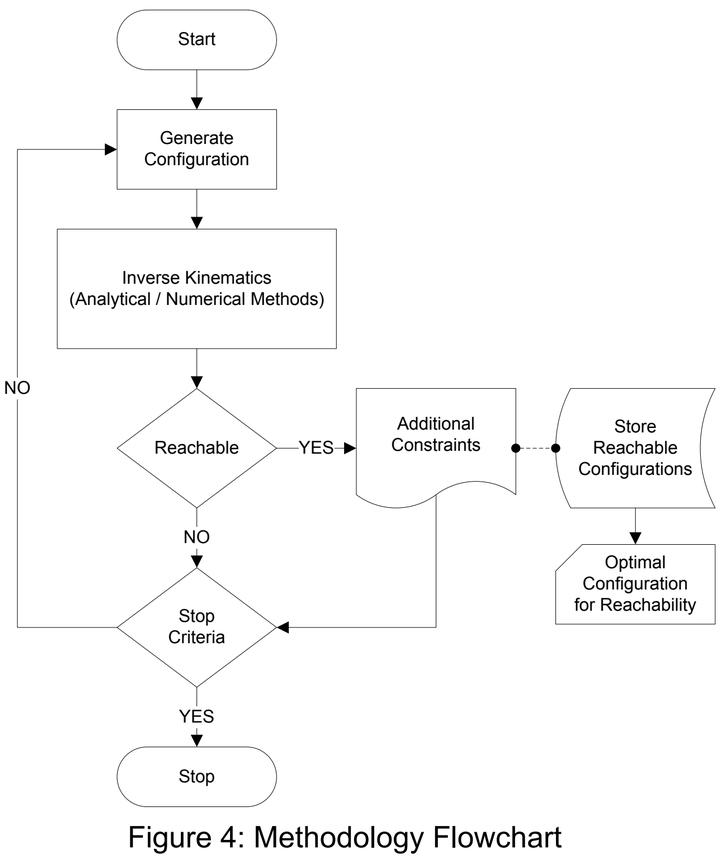

Figure 4: Methodology Flowchart

Figure 2 shows the flow chart of the proposed methodology. Random configurations are generated by the Simulated Annealing 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 simpostures. All reachable structures are stored, and the best reachability structure is always updated. Beginning with a high temperature the SA algorithm with every iterative step gradually lowers the temperature ulating the annealing process. And, after every fixed number of iterations, known as the annealing period, the temperature is back raised again.In this work we assume that the manipulators have a spherical wrist.

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 angles for the manipulator due to presence of the spherical wrist. This PSO based inverse kinematic module find solutions that lie within the 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.

Reachability

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. Where g is the number of inverse kinematic solutions.

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

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. Next, 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. To convert the function into general optimization problems, such that minimizing it will yield optimal solutions we add a negative sign. Therefore, we have:

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

Results & Discussion

We tested the proposed methodology to design manipulators based on task point descriptions the require various positions and orientations. In case of 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]

Spherical Goal

Ring Goal

Horizontal Plane Goal

Vertical Plane Goal

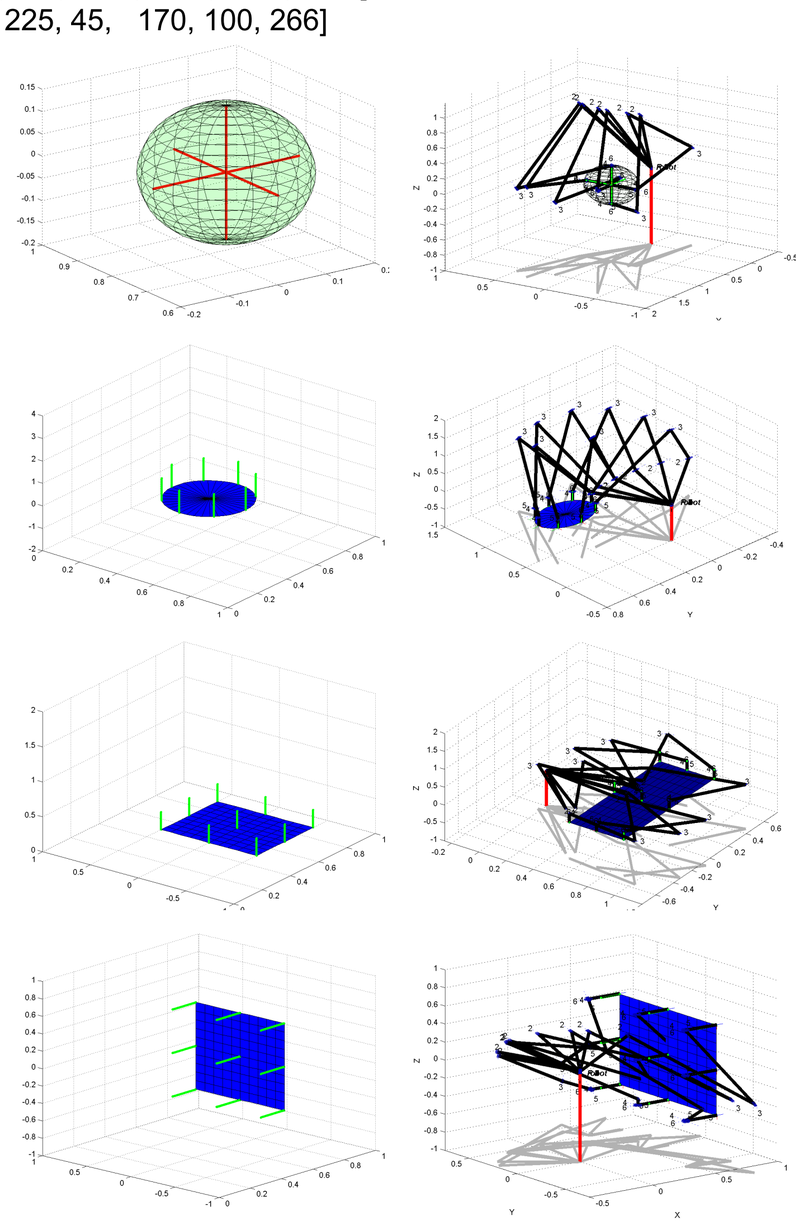

Figure 5-12: Figures to the left show the desired task goal and the figures to the right show the optimal manipulator structure generated by the methodology reaching the task points in the goal with required orientation. The green lines indicate the required task point position and orientation. The blue planes are imaginary , drawn for illustration.

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.

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.