Inverse Kinematics for 8 Degree of Freedom Robotic System

Yuechuan Xue, Xiaoqian Mu

***Abstract*—Robotic kinematic describes the nonlinear relationship between the joint space and task space. Solving the inverse kinematics problem is finding the joints variables that corresponds to the task space positions. Usually, it's hard to solve the inverse kinematics for a multi degree of freedom robot system since it's a nonlinear problem with multiple variables as well as constraints from both the joint limits and task specifications. Considered about the complexity of this problem, the nonlinear optimization algorithm becomes a perfect choice in general cases.**

***Index Terms*—Inverse Kinematics, Robotics, Nonlinear optimization, Sequential Quadratic Programming(SQP).**

I. INTRODUCTION

INVERSE kinematics is one of the fundamental problems in robotics research which objective is to find a robot configuration that leads the robot end effector to desired position and orientation within the limits of the robot manipulator.In other words, inverse kinematics transfers the end effector task space coordinates into joint space coordinates. The position and orientation of the end effector are nonlinearly related to the joint variables which number is the DOF of the robot.

In three-dimensional space, a manipulator with six or more DOF is able to attain any end effector position and orientation. If the manipulator has six DOF, the system with six nonlinear equations can be solved for six joint variables. However, if the robot manipulator has more than six DOF, it is underdetermined since it has more variable than equations. The inverse kinematics problem can be treated as a root finding problem of multi-variables nonlinear polynomial equations. Unfortunately, it is hard to find the analytical solution.

Numerical methods like the Newton-Raphsom(NR) method can be used to solve inverse kinematics problem. The NR method can find solutions if and only if the Jacobian matrix is always reversible and the equations are solvable. The former condition means that the robot should be nonredundant, namely, the number of variables should be the same as the number of equations and constraints, and the robot will not pass through the singular points during moving from the initial configuration to the final configuration. The latter one means the destination position and orientation for the end effector should within the workspace of the robot manipulator.

Usually, the inverse kinematics problem should be considered under the task for robot manipulator. The specification of the robot task will introduce addtional constraints for this problem which make it much more complex. Nonlinear optimization algorithms become a good choice for solving inverse kinematics.

The organization of this report is as follows: The kinematics problem is introduced in Section II. Section III presents the formulation of eight DOF inverse kinematics problem for grasping objects. Section IV shows the principles of SQP algorithm. Simulation result and discussions are in Section V.

II. ROBOT KINEMATICS

Kinematics treats motion of robot manipulator without considering the forces cause it. Within kinematics, the relationship between position, velocity, acceleration and all higher order derivatives of the position variables can be studied. In other words, kinematics of robot manipulators refers to all the geometrical and time-based properties of the motion.

Inverse kinematics is given a desired position in the world frame solving for the joint variables . Since the relationship between joint variables and the position if nonlinear and the number of variables is more then the equations in the relationship, it's impossible to solve the inverse kinematics problem directly.

In kinematics, the transformation matrix $\boldsymbol{T}$ is necessary,

$$\begin{aligned} \boldsymbol{T} &= \begin{bmatrix} \boldsymbol{R} & \boldsymbol{d} \\ \boldsymbol{0} & 1 \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} & d_1 \\ r_{21} & r_{22} & r_{23} & d_2 \\ r_{31} & r_{32} & r_{33} & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{1}$$

in which $\boldsymbol{R}$ is the rotation matrix and $\boldsymbol{d}$ is the transition vector. Elements in $\boldsymbol{R}$ and $\boldsymbol{d}$ are decided by the joint variables of the robot manipulator. Let $\Theta = (\theta_1, \theta_2, ..., \theta_n)$ be the joint variable vector. We can rewrite $\boldsymbol{R}$,$\boldsymbol{d}$ and $\boldsymbol{T}$ as $\boldsymbol{R}(\Theta)$,$\boldsymbol{d}(\Theta)$ and $\boldsymbol{T}(\Theta)$.

Robot manipulator is multiple joints system. To calculate the position of the end effector in the world frame, local frame system for joints should be created and each joint has one aligned local frame. The first frame is the world/base frame

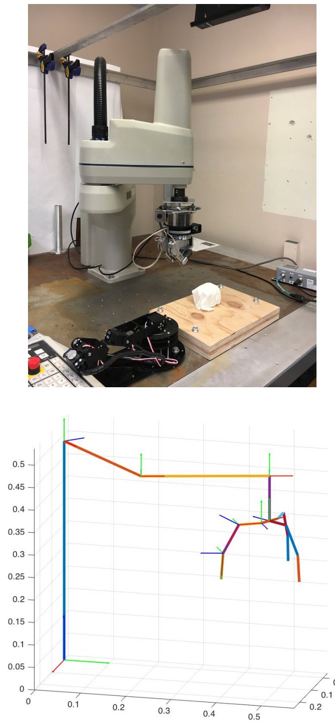

Fig. 1. Frame system for the robot manipulator with red line as x direction, green line as y direction and blue line as z direction.

and $\boldsymbol{T}_i^{i+1}$ is the transformation matrix from the ith frame to $i+1$th frame. Then we have

$$\boldsymbol{p}_w = \boldsymbol{T}_0^1 * \boldsymbol{T}_1^2 * \ldots * \boldsymbol{T}_{n-1}^n * \boldsymbol{p}_e \tag{2}$$

in which $\boldsymbol{p}_w$ is the position of one point in the world frame, $\boldsymbol{p}_e$ is the position in the end effector's local frame and n is the degree of freedom of the robot system.

Our robot system contains one four DOF robot arm and one four DOF robot hand. So the system's DOF is eight. The robot system and it's model with local frames are shown in Fig 1.

III. PROBLEM FORMULATION

Inverse kinematics usually comes with specific robot tasks. So when we try to solve this problem, it is necessary to consider the task constraints. Moreover, robot has it's own workspace, which also can introduce limitations for the inverse kinematics problem.

Consider the grasping task for robot manipulator, which requires the robot's fingers reach desired positions on the surface of the object. This problem can be described as follows:

- All the joint variables should be within the limits of the robot joints
- The target points should be located on the lower links of the robot fingers
- The lower links should be parallel to the tangential plane of the target points

The first characteristic means that the solution should be reasonable. If the joints variables are out of the boundary, the robot can't achieve such configuration and the solution is meaningless. The second property comes form the grasping experiment of human beings since we prefer to use the tip part of our hand rather than the plum to grasp objects. The third one regulate the grasping task by guaranteeing the fingers will not insert into the object.

The nonlinearity together with multiple constraints make the inverse kinematics problem much more complex. We can convert it into a nonlinear optimization problem as follows:

$$\min \sum_{i=0}^{2} \frac{\|(\boldsymbol{p}_i - \boldsymbol{q}_{i1}) \times (\boldsymbol{p}_i - \boldsymbol{q}_{i2})\|}{\|\boldsymbol{q}_i - \boldsymbol{q}_{i2}\|} \tag{3}$$

Subject to

$$\boldsymbol{n}_i \times \boldsymbol{l}_{\perp i} = 0 \tag{4}$$

$$\boldsymbol{n}_i * \boldsymbol{l}_{\perp i} \geq 0 \tag{5}$$

$$t_i * \boldsymbol{q}_{i1} + (1 - t_i) * \boldsymbol{q}_{i2} = \boldsymbol{p}_i \tag{6}$$

$$lb_i \leq \theta_i \leq ub_i \qquad i = 0, 1, \ldots, 7 \tag{7}$$

$$0 \leq t_i \leq 1 \qquad i = 0, 1, 2 \tag{8}$$

In the objective function which minimize the sum of distance from the target points to the lower links, $\boldsymbol{p}_i$ is the ith target point, $\boldsymbol{q}_{i1}$ is the top point of ith lower link and $\boldsymbol{q}_{i2}$ is the tip of ith lower link. $\boldsymbol{n}_i$ is the normal direction of the tangential plane of ith target point and $\boldsymbol{l}_{\perp i}$ is the normal direction of the ith lower link.
Equation (4) guarantees the normal direction of the lower links parallel to the normal directions of the tangential planes which means the lower links parallel to the tangential planes. Inequality (5) shows that the angles between the normal directions of the lower links and normal directions of tangential planes are less than π. (4) and (5) together make sure that the normal vectors are in the same direction.

Equation (6) and (8) mean that $\boldsymbol{p}_i$ is on the segment with two end points $\boldsymbol{q}_{i1}$ and $\boldsymbol{q}_{i2}$. For the inverse kinematics problem, it corresponding to the target points should be on the lower links of the robot hand. (7) gives the mechanical limits of the robot manipulator. In other words, it defines the workspace of the robot.

IV. Sequential Quadratic Programming

In constrained optimization, the general aim is to transform the problem into an easier subproblem that can then be solved and used as the basis of an iterative process. A characteristic of a large class of early methods is the translation of the constrained problem to a basic unconstrained problem by using a penalty function for constraints that are near or beyond the constraint boundary. In this way the constrained problem is solved using a sequence of parameterized unconstrained optimizations, which in the limit (of the sequence) converge to the constrained problem. These methods are now considered relatively inefficient and have been replaced by methods that have focused on the solution of the Karush-Kuhn-Tucker (KKT) equations. The KKT equations are necessary conditions for optimality for a constrained optimization problem. If the problem is a so-called convex programming problem, that is, $f(x)$ and $g_i(x)$, $i = 1, \cdots, m$, are convex functions, then the KKT equations are both necessary and sufficient for a global solution point.

The Kuhn-Tucker equations can be stated as

$$\min f(x) \tag{1}$$

subject to

$$\begin{aligned} g_i(x) &= 0, \quad i = 1, \cdots, m_e & (2) \\ g_i(x) &\leq 0, \quad i = m_e + 1, \cdots, m & (3) \\ \nabla f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \lambda_i \cdot \nabla g_i(\boldsymbol{x}^*) &= 0 & (4) \\ \lambda_i \cdot g_i(\boldsymbol{x}^*) &= 0, \quad i = 1, \cdots, m_e & (5) \\ \lambda_i &\geq 0, \quad i = m_e + 1, \cdots, m & (6) \end{aligned}$$

The equation (4) describes a canceling of the gradients between the objective function and the active constraints at the solution point. For the gradients to be canceled, Lagrange multipliers ($\lambda_i, i = 1, \cdots, m$) are necessary to balance the deviations in magnitude of the objective function and constraint gradients. Because only active constraints are included in this canceling operation, constraints that are not active must not be included in this operation and so are given Lagrange multipliers equal to 0. This is stated implicitly in the last two Kuhn-Tucker equations.

The solution of the KKT equations forms the basis to many nonlinear programming algorithms. These algorithms attempt to compute the Lagrange multipliers directly. Constrained quasi-Newton methods guarantee superlinear convergence by accumulating second-order information regarding the KKT equations using a quasi-Newton updating procedure. These methods are commonly referred to as Sequential Quadratic Programming (SQP) methods, since a QP subproblem is solved at each major iteration (also known as Iterative Quadratic Programming, Recursive Quadratic Programming, and Constrained Variable Metric methods).

A. Active Set Method

Active Set method allows to closely mimic Newton's method for constrained optimization just as is done for unconstrained optimization. At each major iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a QP subproblem whose solution is used to form a search direction for a line search procedure.

B. Quadratic Subproblem

$$\min \frac{1}{2} \boldsymbol{d}^\top H_k \boldsymbol{d} + \nabla f(\boldsymbol{x}_k)^\top \boldsymbol{d} \tag{7}$$

subject to:

$$\begin{aligned} \nabla g_i(\boldsymbol{x}_k)^\top \boldsymbol{d} + g_i(\boldsymbol{x}_k) &= 0, \quad i = 1, \cdots, m_e & (8) \\ \nabla g_i(\boldsymbol{x}_k)^\top \boldsymbol{d} + g_i(\boldsymbol{x}_k) &\leq 0, \quad i = m_e + 1, \cdots, m & (9) \end{aligned}$$

This subproblem can be solved using any QP algorithm. The solution is used to form a new iterate

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k \tag{10}$$

The step length parameter α_k is determined by an appropriate line search procedure so that a sufficient decrease in a merit function is obtained. The matrix H_k is a positive definite approximation of the Hessian matrix of the Lagrangian function. H_k can be updated by any of the quasi-Newton methods, although the BFGS method appears to be the most popular.

A nonlinearly constrained problem can often be solved in fewer iterations than an unconstrained problem using SQP. One of the reasons for this is that, because of limits on the feasible area, the optimizer can make informed decisions regarding directions of search and step length.

C. Updating the Hessian Matrix

At each major iteration a positive definite quasi-Newton approximation of the Hessian of the Lagrangian function, H, is calculated using the BFGS method, where $\lambda_i, i = 1, \cdots, m$, is an estimate of the Lagrange multipliers.

V. Experiment and Discussion

During simulation, desired grasping positions and the normal directions of the tangential planes are calculated from the forward kinematics which can make sure the problem has solution. For such kind of desired position and orientation, SQP method can always find a solution. Compared with Interior Point method, SQP method converges more fast.For the given position and orientation, SQP method runs in about thirty iterations while the Interior Point method runs in about two hundreds iterations.

Fig.2 shows that simulations results by using Interior Point method and SQP method. Fig.3 is the convergence process for the SQP method.

VI. Conclusion

In this project, we solved the inverse kinematics problem for our eight DOF robot system by using nonlinear optimization algorithm.

SQP method can always converge to a reasonable destination. Since it does not rely on the inverse of Jacobian matrix, singular points will not effect the solution of it. We also test other nonlinear optimization methods, some of them not convergent and some are quite slow. SQP is stable and fast for solving the inverse kinematics problem for grasping task.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

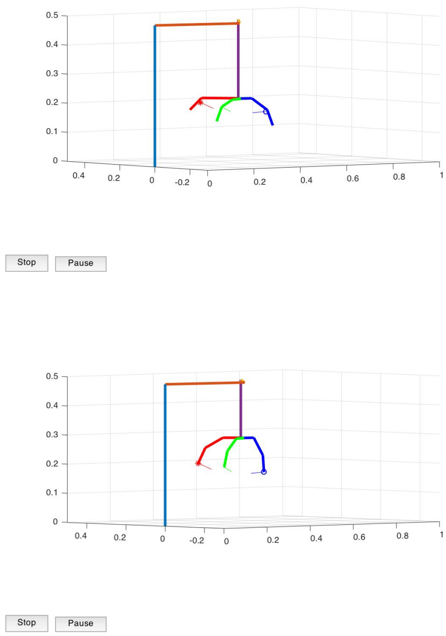

Fig. 2. First one shows the simulation result by using Interior Point method and second one presents the simulation result by suing SQP method.

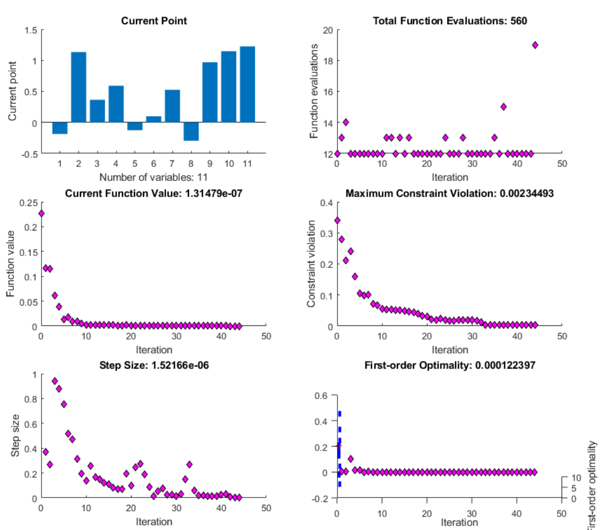

Fig. 3. Convergence process for the SQP method

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

References

[1] Tomomichi Sugihara, *Solvability-Unconcerned Inverse Kinematics by the LevenbergâĂŞMarquardt Method* IEEE TRANSACTIONS ON ROBOTICS, VOL. 27, NO. 5, OCTOBER 2011.