ME/AER 676 Robot Modeling & Control
Spring 2023

Inverse Kinematics

Hasan A. Poonawala

Department of Mechanical Engineering
University of Kentucky

Email: hasan.poonawala@uky.edu
Web: https://www.engr.uky.edu/~hap

Introduction

- The Forward Kinematics problem combines known closed-form expressions for individual homogenous transformations
- No closed-form expression for f in $x = f(q)$ needs to be maintained to obtain x
- Computing the inverse, however, is not as easy
- The inverse kinematics problem is often not even unique, which has algorithmic implications

Inverse Kinematics

Since we know how to build $f(q)$, we arrive at two approaches to inverse kinematics

- Analytic approaches:
  Build the closed-form expression and define a closed-form inverse
- Numerical approaches:
  Numerically search for values of q so that $f(q) = x$, where $f(q)$ is known to us

Analytic Inverse Kinematics

- Complicated to derive, but yields fast computations
- Some robots are designed with geometries that simplify the expressions:
  - The wrist is has three links with intersecting axes of rotation (spherical joint)
  - The end-effector frame coincides with wrist center.

Numerical Inverse Kinematics

- solve optimization:

$$\min_{q} \quad \|x - f(q)\|_2^2$$

- We can add constraints that make the solution unique, or other benefits
- We may also use other measures for the distance between x and $f(q)$

Analytical Inverse Velocity Kinematics

- Instead of $q = f^{-1}(x)$, some tasks require calculating $\dot{q}$ given task space velocity ξ
- If $J(q)$ is square and full-rank, then $\dot{q} = J(q)^{-1}\xi$
- If $J(q) \in \mathbb{R}^{m \times n}$, $m < n$, and $\mathrm{rank}(J(q)) = m$, we may compute

  $$\dot{q} = J^{+}\xi + (I - J^{+}J)b,$$

  where

  $$J^{+} = J^{T}(JJ^{T})^{-1},$$

  and $b \in \mathbb{R}^{n}$ is an arbitrary vector that does not affect ξ.

$$J^{+} = J^{T}(JJ^{T})^{-1}$$

Numerical Inverse Velocity Kinematics

- Instead of 'closed-form' pseudo-inverse, solve optimization:

$$\min_{q} \quad \|\xi - J(q)\dot{q}\|_2^2$$

- Here too, we can add constraints that make the solution unique, or other benefits
- Again, we may also use other measures for the distance between ξ and $\dot{q}$

Differential Inverse Kinematics

- IDEA: To solve $\min_q \|x - f(q)\|_2^2$, use $\dot{q} = J^+\xi$

Differential Inverse Kinematics

- IDEA: To solve $\min_q \|x - f(q)\|_2^2$, use $\dot{q} = J^+\xi$
- If $L(q) = \|x - f(q)\|_2^2$, then

$$\frac{d}{dt}L(q) = (x - f(q))^T (\xi - J(q)\dot{q}) \tag{1}$$

Differential Inverse Kinematics

- IDEA: To solve $\min_q \|x - f(q)\|_2^2$, use $\dot{q} = J^+ \xi$
- If $L(q) = \|x - f(q)\|_2^2$, then

$$\frac{d}{dt} L(q) = (x - f(q))^T (\xi - J(q)\dot{q}) \tag{1}$$

- If we want $L(q) \to 0$, choose

$$\xi - J(q)\dot{q} = -(x - f(q)) \tag{2}$$

$$\implies \dot{q} = J^+ (\xi + (x - f(q))) \tag{3}$$

Differential Inverse Kinematics

- IDEA: To solve $\min_q \|x - f(q)\|_2^2$, use $\dot{q} = J^+ \xi$
- If $L(q) = \|x - f(q)\|_2^2$, then

$$\frac{d}{dt} L(q) = (x - f(q))^T (\xi - J(q)\dot{q}) \tag{1}$$

- If we want $L(q) \to 0$, choose

$$\xi - J(q)\dot{q} = -(x - f(q)) \tag{2}$$
$$\implies \dot{q} = J^+ (\xi + (x - f(q))) \tag{3}$$

- Also works as a task-space position controller, assuming a low-level velocity-tracking loop!

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$
- Plan a trajectory $x(t)$, where $t \in [0, 1]$, such that from $x(0) = x_0$ to $x(1) = x$ (straight line?)

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$
- Plan a trajectory $x(t)$, where $t \in [0, 1]$, such that from $x(0) = x_0$ to $x(1) = x$ (straight line?)
- Convert $x(t)$ to velocities $\xi(t)$

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$
- Plan a trajectory $x(t)$, where $t \in [0, 1]$, such that from $x(0) = x_0$ to $x(1) = x$ (straight line?)
- Convert $x(t)$ to velocities $\xi(t)$
- Integrate $q(1) = \int_0^1 \dot{q}(s)ds = \int_0^1 J(q(s))^+ \xi(s)ds$

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$
- Plan a trajectory $x(t)$, where $t \in [0, 1]$, such that from $x(0) = x_0$ to $x(1) = x$ (straight line?)
- Convert $x(t)$ to velocities $\xi(t)$
- Integrate $q(1) = \int_0^1 \dot{q}(s)ds = \int_0^1 J(q(s))^+ \xi(s)ds$
  - Note that we are building $q(s)$ as we go!

Differential Inverse Kinematics

We may interpret the previous algorithm as trying to solve $x = f(q)$ by the following approach:

- Start with some q_0, calculate $x_0 = f(q_0)$
- Plan a trajectory $x(t)$, where $t \in [0, 1]$, such that from $x(0) = x_0$ to $x(1) = x$ (straight line?)
- Convert $x(t)$ to velocities $\xi(t)$
- Integrate $q(1) = \int_0^1 \dot{q}(s)ds = \int_0^1 J(q(s))^+ \xi(s)ds$
  - Note that we are building $q(s)$ as we go!
- The integration drifts, so we need a correction term

$$\dot{q}(t) = J^+ \xi(t) + \underbrace{J^+ \left(x(t) - f(q(t))\right)}_{\text{error correction}}$$