CS545—Contents IX

- Inverse Kinematics
  - Analytical Methods
  - Iterative (Differential) Methods
    - Geometric and Analytical Jacobian
    - Jacobian Transpose Method
    - Pseudo-Inverse
    - Pseudo-Inverse with Optimization
    - Extended Jacobian Method

- Reading Assignment for Next Class
  - See http://www-clmc.usc.edu/~cs545
The Inverse Kinematics Problem

- Direct Kinematics
  \[ x = f(\theta) \]

- Inverse Kinematics
  \[ \theta = f^{-1}(x) \]

- Possible Problems of Inverse Kinematics
  - Multiple solutions
  - Infinitely many solutions
  - No solutions
  - No closed-form (analytical solution)
Analytical (Algebraic) Solutions

- Analytically invert the direct kinematics equations and enumerate all solution branches
  - Note: this only works if the number of constraints is the same as the number of degrees-of-freedom of the robot
  - What if not?
    - Iterative solutions
    - Invent artificial constraints

- Examples
  - 2DOF arm
  - See S&S textbook 2.11 ff
Analytical Inverse Kinematics of a 2 DOF Arm

Inverse Kinematics:

\[ x = l_1 \cos \theta_1 + l_2 \cos(\theta_1 + \theta_2) \]
\[ y = l_1 \sin \theta_1 + l_2 \sin(\theta_1 + \theta_2) \]

\[ l = \sqrt{x^2 + y^2} \]
\[ l_2^2 = l_1^2 + l^2 - 2l_1l \cos \gamma \]
\[ \Rightarrow \gamma = \arccos \left( \frac{l^2 + l_1^2 - l_2^2}{2l_1l} \right) \]
\[ \frac{y}{x} = \tan \varepsilon \quad \Rightarrow \quad \theta_1 = \arctan \frac{y}{x} - \gamma \]
\[ \theta_2 = \arctan \left( \frac{y - l_1 \sin \theta}{x - l_1 \cos \theta_1} \right) - \theta_1 \]
Iterative Solutions of Inverse Kinematics

- Resolved Motion Rate Control

\[ \dot{x} = J(\theta) \dot{\theta} \quad \Rightarrow \]

\[ \dot{\theta} = J(\theta)^\# \dot{x} \]

- Properties
  - Only holds for high sampling rates or low Cartesian velocities
  - “a local solution” that may be “globally” inappropriate
  - Problems with singular postures
  - Can be used in two ways:
    - As an instantaneous solutions of “which way to take “
    - As an “batch” iteration method to find the correct configuration at a target
Essential in Resolved Motion Rate Methods: The Jacobian

- **Jacobian of direct kinematics:**

\[
x = f(\theta) \quad \Rightarrow \quad \frac{\partial x}{\partial \theta} = \frac{\partial f(\theta)}{\partial \theta} = J(\theta)
\]

- **In general, the Jacobian (for Cartesian positions and orientations) has the following form (geometrical Jacobian):**

\[
J(\theta) = \begin{pmatrix}
  j_{P1} & \cdots & j_{Pm} \\
  j_{O1} & \cdots & j_{Om} \\
\end{pmatrix}
\]

where

\[
\begin{pmatrix}
  j_{P1} \\
  j_{O1}
\end{pmatrix} = \begin{cases}
  \begin{bmatrix}
    z_{i-1} \\
    0 \\
    z_{i-1} \times (p - p_{i-1})
  \end{bmatrix} & \text{for a prismatic joint}
  \\
  z_{i-1} & \text{for a revolute joint}
\end{cases}
\]

\( p_i \) is the vector from the origin of the world coordinate system to the origin of the i-th link coordinate system, \( p \) is the vector from the origin to the endeffector end, and \( z \) is the i-th joint axis (p.72 S&S)
The Jacobian Transpose Method

\[ \Delta \theta = \alpha J^T(\theta) \Delta x \]

- **Operating Principle:**
  - Project difference vector \( \Delta x \) on those dimensions \( q \) which can reduce it the most

- **Advantages:**
  - Simple computation (numerically robust)
  - No matrix inversions

- **Disadvantages:**
  - Needs many iterations until convergence in certain configurations (e.g., Jacobian has very small coefficients)
  - Unpredictable joint configurations
  - Non conservative
Jacobian Transpose Derivation

Minimize cost function

\[
F = \frac{1}{2} \left( \mathbf{x}_{\text{target}} - \mathbf{x} \right)^T \left( \mathbf{x}_{\text{target}} - \mathbf{x} \right)
\]

\[
= \frac{1}{2} \left( \mathbf{x}_{\text{target}} - f(\theta) \right)^T \left( \mathbf{x}_{\text{target}} - f(\theta) \right)
\]

with respect to \( \theta \) by gradient descent:

\[
\Delta \theta = -\alpha \left( \frac{\partial F}{\partial \theta} \right)^T
\]

\[
= \alpha \left( \left( \mathbf{x}_{\text{target}} - \mathbf{x} \right)^T \frac{\partial f(\theta)}{\partial \theta} \right)^T
\]

\[
= \alpha J^T(\theta) \left( \mathbf{x}_{\text{target}} - \mathbf{x} \right)
\]

\[
= \alpha J^T(\theta) \Delta \mathbf{x}
\]
Jacobian Transpose
Geometric Intuition
The Pseudo Inverse Method

\[ \Delta \theta = \alpha J^T(\theta)(J(\theta)J^T(\theta))^{-1} \Delta x = J^\# \Delta x \]

- Operating Principle:
  - Shortest path in \( q \)-space

- Advantages:
  - Computationally fast (second order method)

- Disadvantages:
  - Matrix inversion necessary (numerical problems)
  - Unpredictable joint configurations
  - Non conservative
Pseudo Inverse Method
Derivation

For a small step $\Delta x$, minimize with respect to $\Delta \theta$ the cost function:

$$ F = \frac{1}{2} \Delta \theta^T \Delta \theta + \lambda^T (\Delta x - J(\theta)\Delta \theta) $$

where $\lambda^T$ is a vector of Lagrange multipliers.

Solution:

(1) \[ \frac{\partial F}{\partial \lambda} = 0 \Rightarrow \Delta x = J \Delta \theta \]

(2) \[ \frac{\partial F}{\partial \Delta \theta} = 0 \Rightarrow \Delta \theta = J^T \lambda \Rightarrow J \Delta \theta = J J^T \lambda \]

\[ \Rightarrow \lambda = \left( J J^T \right)^{-1} J \Delta \theta \]

insert (1) into (2):

(3) \[ \lambda = \left( J J^T \right)^{-1} \Delta x \]

insertion of (3) into (2) gives the final result:

$$ \Delta \theta = J^T \lambda = J^T \left( J J^T \right)^{-1} \Delta x $$
Pseudo Inverse
Geometric Intuition

\[ \text{start posture} = \text{desired posture for optimization} \]
Pseudo Inverse with explicit Optimization Criterion

\[ \Delta \theta = \alpha J^\# \Delta x + \left( I - J^\# J \right) (\theta_0 - \theta) \]

- Operating Principle:
  - Optimization in null-space of Jacobian using a kinematic cost function
    \[ F = g(\theta), \quad e.g., F = \sum_{i=1}^{d} (\theta_i - \theta_{i,0})^2 \]

- Advantages:
  - Computationally fast
  - Explicit optimization criterion provides control over arm configurations

- Disadvantages:
  - Numerical problems at singularities
  - Non conservative
Pseudo Inverse Method & Optimization Derivation

For a small step $\Delta x$, minimize with respect to $\Delta \theta$ the cost function:

$$F = \frac{1}{2} \left( (\Delta \theta + \theta - \theta_o) \right)^T (\Delta \theta + \theta - \theta_o) + \lambda^T (\Delta x - J(\theta)\Delta \theta)$$

where $\lambda^T$ is a vector of Lagrange multipliers.

Solution:

1. $\frac{\partial F}{\partial \lambda} = 0 \implies \Delta x = J \Delta \theta$

2. $\frac{\partial F}{\partial \Delta \theta} = 0 \implies \Delta \theta = J^T \lambda - (\theta - \theta_o) \implies J \Delta \theta = J J^T \lambda - J (\theta - \theta_o)$
   
   $\implies \lambda = (J J^T)^{-1} J \Delta \theta + (J J^T)^{-1} J (\theta - \theta_o)$

insert (1) into (2):

3. $\lambda = (J J^T)^{-1} \Delta x + (J J^T)^{-1} J (\theta - \theta_o)$

insertion of (3) into (2) gives the final result:

$$\Delta \theta = J^T \lambda - (\theta - \theta_o) = J^T (J J^T)^{-1} \Delta x + J^T (J J^T)^{-1} J (\theta - \theta_o) - (\theta - \theta_o)$$

$$= J^\# \Delta x + (I - J^\# J)(\theta_o - \theta)$$
The Extended Jacobian Method

\[ \Delta \theta = \alpha \left(J^{\text{ext.}}(\theta)\right)^{-1} \Delta x^{\text{ext.}} \]

- **Operating Principle:**
  - Optimization in null-space of Jacobian using a kinematic cost function
  \[ F = g(\theta), \quad \text{e.g., } F = \sum_{i=1}^{\theta} (\theta_i - \theta_{i,0})^2 \]

- **Advantages:**
  - Computationally fast (second order method)
  - Explicit optimization criterion provides control over arm configurations
  - Numerically robust
  - Conservative

- **Disadvantages:**
  - Computationally expensive matrix inversion necessary (singular value decomposition)
  - Note: new and better ext. Jac. algorithms exist
Extended Jacobian Method

Derivation

The forward kinematics $x = f(\theta)$ is a mapping $\mathbb{R}^n \rightarrow \mathbb{R}^m$, e.g., from a $n$-dimensional joint space to a $m$-dimensional Cartesian space. The singular value decomposition of the Jacobian of this mapping is:

$$J(\theta) = USV^T$$

The rows $[V_i]$ whose corresponding entry in the diagonal matrix $S$ is zero are the vectors which span the Null space of $J(\theta)$. There must be (at least) $n-m$ such vectors ($n \geq m$). Denote these vectors $n_i, i \in [1, n-m]$.

The goal of the extended Jacobian method is to augment the rank deficient Jacobian such that it becomes properly invertible. In order to do this, a cost function $F = g(\theta)$ has to be defined which is to be minimized with respect to $\theta$ in the Null space. Minimization of $F$ must always yield:

$$\frac{\partial F}{\partial \theta} = \frac{\partial g}{\partial \theta} = 0$$

Since we are only interested in zeroing the gradient in Null space, we project this gradient onto the Null space basis vectors:

$$G_i = \frac{\partial g}{\partial \theta} n_i$$

If all $G_i$ equal zero, the cost function $F$ is minimized in Null space.

Thus we obtain the following set of equations which are to be fulfilled by the inverse kinematics solution:

$$\begin{pmatrix} f(\theta) \\ G_i \\ \vdots \\ G_{m-n} \end{pmatrix} = \begin{pmatrix} x \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

For an incremental step $\Delta x$, this system can be linearized:

$$\begin{pmatrix} J(\theta) \\ \frac{\partial G_i}{\partial \theta} \\ \vdots \\ \frac{\partial G_{m-n}}{\partial \theta} \end{pmatrix} \Delta \theta = \begin{pmatrix} \Delta x \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ or } J^{inv} \Delta \theta = \Delta x^{inv}.$$  

The unique solution of these equations is: $\Delta \theta = (J^{inv})^{-1} \Delta x^{inv}$. 
Extended Jacobian
Geometric Intuition