Velocity Kinematics (Pre Lecture)

Dr. Neil T. Dantam

CSCI-534, Colorado School of Mines

Spring 2020

Introduction

Velocity Kinematics

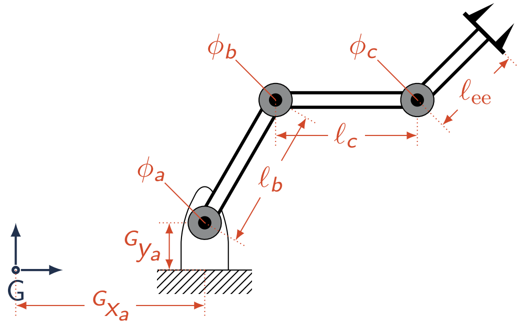
- Say we want to move the end-effector in workspace (Cartesian)
- But, we control the robots joints
- What jointpace velocities produced desired workspace velocity?

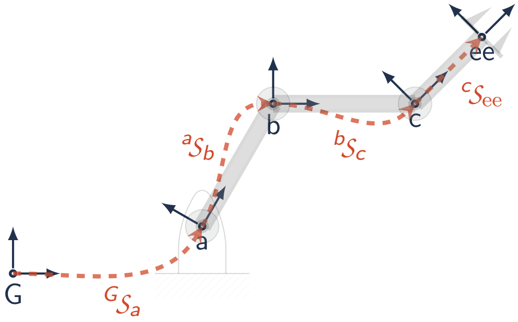
Outcomes

- Know definitions for **linear equations**
- Construct **Manipulator Jacobian**
- Solve **inverse kinematics** equations
- Use Jacobian **nullspace** to incorporate jointspace velocities

Forward Kinematics

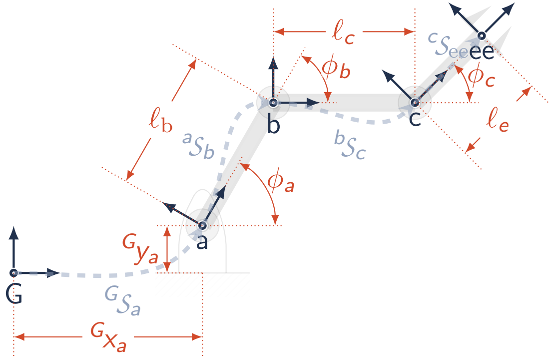
Diagram





Forward Kinematics

Transforms



- **Relative:** $\mathcal{S} = h + \frac{1}{2}\vec{v} \otimes h\varepsilon$
  - ${}^{G}\mathcal{S}_a = \exp\left(\frac{\phi_a}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}{}^{G}v_a \otimes \exp\left(\frac{\phi_a}{2}\hat{\boldsymbol{k}}\right)\varepsilon$
  - ${}^{a}\mathcal{S}_b = \exp\left(\frac{\phi_b}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}\ell_b\hat{\boldsymbol{\imath}} \otimes \exp\left(\frac{\phi_b}{2}\hat{\boldsymbol{k}}\right)\varepsilon$
  - ${}^{b}\mathcal{S}_c = \exp\left(\frac{\phi_c}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}\ell_c\hat{\boldsymbol{\imath}} \otimes \exp\left(\frac{\phi_c}{2}\hat{\boldsymbol{k}}\right)\varepsilon$
  - ${}^{c}\mathcal{S}_{\mathrm{ee}} = 1 + \frac{1}{2}\ell_e\hat{\boldsymbol{\imath}}\varepsilon$
- **Absolute:**

$$ {}^{G}\mathcal{S}_{\mathrm{ee}} = {}^{G}\mathcal{S}_a \otimes {}^{a}\mathcal{S}_b \otimes {}^{b}\mathcal{S}_c \otimes {}^{c}\mathcal{S}_{\mathrm{ee}} $$

How do we relate change: $\frac{d\phi}{dt}$ *vs.* $\frac{d\,{}^{G}\mathcal{S}_{\mathrm{ee}}}{dt}$ *?*

Outline

Systems of Linear Equations

Linear Equation

$$a_1 x_1 + \ldots + a_n x_n = b$$

- Where:
  - x_i is a variable
  - a_i is constant
  - b is constant
- Straight-line in **x**-space

Linear System

$$a_{1,1} x_1 + \ldots + a_{1,n} x_n = b_1$$

$$\vdots$$

$$a_{m,1} x_1 + \ldots + a_{m,n} x_n = b_m$$

$$\Downarrow$$

$$\underbrace{\begin{bmatrix} a_{1,1} & \ldots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \ldots & a_{m,n} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}}_{\mathbf{b}}$$

Matrix Inverse

Definition (Matrix Inverse)

The inverse of square matrix $\mathbf{A}$ is the matrix $\mathbf{A}^{-1}$ such that:

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Fully-determined systems

$$\begin{array}{l|rcl|l}
0. & \mathbf{A}\mathbf{x} & = & \mathbf{b} & \text{Linear system, } \mathbf{A} \in \mathbb{R}^n \times \mathbb{R}^n \\
1. & \mathbf{A}^{-1}\mathbf{A}\mathbf{x} & = & \mathbf{A}^{-1}\mathbf{b} & \text{Multiply by } \mathbf{A}^{-1} \\
2. & \cancelto{\mathbf{I}}{\mathbf{A}^{-1}\mathbf{A}}\mathbf{x} & = & \mathbf{A}^{-1}\mathbf{b} & \text{Cancel} \\
3. & \mathbf{x} & = & \mathbf{A}^{-1}\mathbf{b} & \text{Result}
\end{array}$$

Example: Fully-determined System

Given:
- $x_1 + 2x_2 = 5$
- $3x_1 + 4x_2 = 6$

Find: Values of x_1 and x_2

Solution:
- $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$
- $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 5 \\ 6 \end{bmatrix}$
- $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 1.5 & -.5 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix}$
- $\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -4 \\ 4.5 \end{bmatrix}$

Exercise: Fully-determined System

Given:
- $\phi_1 + \phi_2 + \phi_3 = 1$
- $2\phi_1 + 3\phi_3 = 2$
- $4\phi_2 = 3$

Find: Values of ϕ_1, ϕ_2, ϕ_3

Solution:

Under and Over-determined Systems

Underdetermined

More unknowns than equations.
"Fat" matrix.

$$\begin{aligned} 1x_1 &+ 2x_2 & 3x_3 &= 7 \\ 4x_1 &+ 5x_2 & 6x_3 &= 8 \end{aligned}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Overdetermined

More equations than unknowns.
"Skinny" matrix

$$\begin{aligned} 1x_1 &+ 2x_2 = 7 \\ 3x_1 &+ 4x_2 = 8 \\ 5x_1 &+ 6x_2 = 9 \end{aligned}$$

$$\Downarrow$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

Pseudoinverse

Moore-Penrose Inverse

Definition (Moore-Penrose inverse)

The Moore-Penrose inverse $\mathbf{A}^{+}$ generalizes the matrix inverse to nonsquare matrices:

- $\mathbf{A}\mathbf{A}^{+}\mathbf{A} = \mathbf{A}$
- $\mathbf{A}^{+}\mathbf{A}\mathbf{A}^{+} = \mathbf{A}^{+}$

Computation

- Linearly independent columns:
  - $\mathbf{A}^{+} = \left(\mathbf{A}^{T}\mathbf{A}\right)^{-1}\mathbf{A}^{T}$
  - $\mathbf{A}^{+}\mathbf{A} = \mathbf{I}$
- Linearly independent rows:
  - $\mathbf{A}^{+} = \mathbf{A}^{T}\left(\mathbf{A}\mathbf{A}^{T}\right)^{-1}$
  - $\mathbf{A}\mathbf{A}^{+} = \mathbf{I}$

Least Squares Solutions

Underdetermined

Given: Underdetermined system $\mathbf{Ax} = \mathbf{b}$, i.e., $\mathbf{A}$ has more columns (unknowns) than rows (equations).

Find: Vector $\mathbf{x}$, such that:

$$\begin{aligned} \text{minimize:} \quad & \|\mathbf{x}\| \\ \text{subject to:} \quad & \mathbf{Ax} = \mathbf{b} \end{aligned}$$

Solution: $\mathbf{x} = \mathbf{A}^{+}\mathbf{b}$

Overdetermined

Given: Overdetermined system $\mathbf{Ax} = \mathbf{b}$, , i.e., $\mathbf{A}$ has more rows (equations) than columns (unknowns).

Find: Vector $\mathbf{z}$, such that:

$$\text{minimize:} \quad \|\mathbf{Az} - \mathbf{b}\|$$

Solution: $\mathbf{z} = \mathbf{A}^{+}\mathbf{b}$

Example: Underdetermined System

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Solution

- $\mathbf{x} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^{+} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$
- $\mathbf{x} = \begin{bmatrix} -9.44 & 0.444 \\ -0.111 & 0.111 \\ 0.722 & -0.222 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$
- $\mathbf{x} = \begin{bmatrix} -3.06 \\ 0.111 \\ 3.28 \end{bmatrix}$

Check

- $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} -3.06 \\ 0.111 \\ 3.28 \end{bmatrix} \stackrel{?}{=} \begin{bmatrix} 7 \\ 8 \end{bmatrix}$
- $\begin{bmatrix} 7 \\ 8 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$

Example: Overdetermined System, 0

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

Solution

- $\mathbf{z} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}^{+} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$
- $\mathbf{z} = \begin{bmatrix} -1.33 & -0.333 & 0.667 \\ 1.08 & 0.333 & -0.417 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$
- $\mathbf{z} = \begin{bmatrix} -6 & 6.5 \end{bmatrix}^{T}$

Check

- $\mathbf{e} = \|\mathbf{Az} - \mathbf{b}\|$
- $\mathbf{e} = \left\| \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \begin{bmatrix} -6 \\ 6.5 \end{bmatrix} - \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \right\|$
- $\mathbf{e} = \left\| \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} - \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix} \right\| = 0$

Example: Overdetermined System, 1

$$\begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 7 & 11 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 13 \\ 17 \\ 19 \end{bmatrix}$$

Solution

- $\mathbf{z} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 7 & 11 \end{bmatrix}^+ \begin{bmatrix} 13 \\ 17 \\ 19 \end{bmatrix}$
- $\mathbf{z} = \begin{bmatrix} -2.71 & -1.43 & 1.14 \\ 1.71 & 0.929 & -0.643 \end{bmatrix} \begin{bmatrix} 13 \\ 17 \\ 19 \end{bmatrix}$
- $\mathbf{z} = \begin{bmatrix} -37.9 & 25.9 \end{bmatrix}^T$

Check

- $\mathbf{e} = \left\| \mathbf{A}\mathbf{z} - \mathbf{b} \right\|$
- $\mathbf{e} = \left\| \begin{bmatrix} 1 & 2 \\ 3 & 5 \\ 7 & 11 \end{bmatrix} \begin{bmatrix} -37.9 \\ 25.9 \end{bmatrix} - \begin{bmatrix} 13 \\ 17 \\ 19 \end{bmatrix} \right\|$
- $\mathbf{e} = \left\| \begin{bmatrix} 13.9 \\ 15.7 \\ 19.4 \end{bmatrix} - \begin{bmatrix} 13 \\ 17 \\ 19 \end{bmatrix} \right\| = 1.60$

Outline

Gradient

Definition: Gradient

A vector of partial derivatives of a multi-variable, scalar-valued function.

Given $f : \mathbb{R}^n \mapsto \mathbb{R}$ and $x = f(\phi)$,

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial \phi_1} & \dots & \frac{\partial f}{\partial \phi_n} \end{bmatrix}$$

- Vector field: $\nabla f : \mathbb{R}^n \mapsto \mathbb{R}^n$
- Magnitude and direction (in ϕ) of change of f

Jacobian Matrix

Definition: Jacobian matrix

The matrix of all first-order partial derivatives of a vector-valued function.

Given $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^m$ and $\mathbf{x} = \mathbf{f}(\phi)$,

$$\mathbf{J} = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\phi}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \phi_1} & \dots & \frac{\partial \mathbf{f}}{\partial \phi_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial \phi_1} & \dots & \frac{\partial f_1}{\partial \phi_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial \phi_1} & \dots & \frac{\partial f_m}{\partial \phi_n} \end{bmatrix}$$

- Matrix: $\mathbf{J} \in \mathbb{R}^m \times \mathbb{R}^n$
- Each row i is the gradient of f_i
- Each column j represents the "contribution" of ϕ_j

Linear Equation Solution

i.e., who cares about Jacobians?

Function and Jacobian:

$$\mathbf{x} = \mathbf{f}(\phi) \qquad \text{and} \qquad \mathbf{J} = \frac{\partial \mathbf{f}}{\partial \phi}$$

Chain Rule:

$$\frac{\partial \mathbf{f}}{\partial \phi}\frac{d\phi}{dt} = \frac{d\mathbf{f}}{dt}$$

Change of notation:

$$\mathbf{J}\dot{\phi} = \dot{\mathbf{x}}$$

Matrix Algebra:

$$\dot{\phi} = \mathbf{J}^{+}\dot{\mathbf{x}}$$

Instantaneous $\dot{\mathbf{x}} \leftrightarrow \dot{\phi}$ mapping as a system of linear equations

TF Derivative vs. Velocity

$$\dot{h} = \frac{1}{2}\omega \otimes h \qquad \text{and} \qquad \dot{\mathcal{S}} = \frac{1}{2}\left(\omega + \left(\dot{v} + v \times \omega\right)\varepsilon\right) \otimes \mathcal{S}$$

Derivative

$$\dot{\mathcal{S}}$$

- $\dot{\mathcal{S}} = \frac{\partial \mathcal{S}}{\partial \phi}\dot{\phi}$
- $\frac{\partial \mathcal{S}}{\partial \phi} \in \mathbb{R}^8 \times \mathbb{R}^n$
- Unit constraint / normalization

Velocity

$$\begin{bmatrix} \omega \\ \dot{v} \end{bmatrix}$$

- $\begin{bmatrix} \omega \\ \dot{v} \end{bmatrix} = \mathbf{J}\dot{\phi}$
- $\mathbf{J} \in \mathbb{R}^6 \times \mathbb{R}^n$ (minimal coordinates)
- No normalization
- Geometric interpretation

Manipulator Jacobian Overview

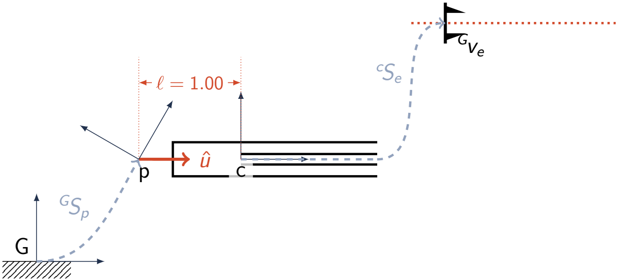
$$\begin{bmatrix} \omega \\ \dot{v} \end{bmatrix} = \mathbf{J}\dot{\phi}$$

$$\begin{bmatrix} \omega \\ \dot{v} \end{bmatrix} = \overbrace{\begin{bmatrix} \begin{pmatrix} \mathbf{j}_{r_1} \\ \mathbf{j}_{p_1} \end{pmatrix} & \begin{matrix} \cdots \\ \cdots \end{matrix} & \begin{pmatrix} \mathbf{j}_{r_n} \\ \mathbf{j}_{p_n} \end{pmatrix} \end{bmatrix}}^{\mathbf{J}} \overbrace{\begin{bmatrix} \dot{\phi}_1 \\ \vdots \\ \dot{\phi}_n \end{bmatrix}}^{\dot{\phi}}$$

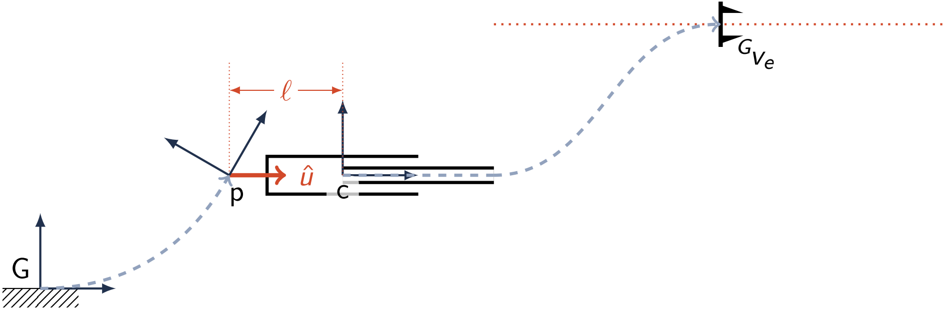
- Velocities:
  - ω: end-effector rotational velocity
  - $\dot{v}$: end-effector translational velocity
  - $\dot{\phi}$: joint-space velocity
- $\begin{pmatrix} \mathbf{j}_{r_i} \\ \mathbf{j}_{p_i} \end{pmatrix}$: velocity "contribution" of joint i
  - $\mathbf{j}_{r_i} = \begin{bmatrix} \frac{d\theta_x}{d\phi_i} & \frac{d\theta_y}{d\phi_i} & \frac{d\theta_z}{d\phi_i} \end{bmatrix}$ (rotational)
  - $\mathbf{j}_{p_i} = \begin{bmatrix} \frac{dx}{d\phi_i} & \frac{dy}{d\phi_i} & \frac{dz}{d\phi_i} \end{bmatrix}$ (translational)

Prismatic Joint

Animation



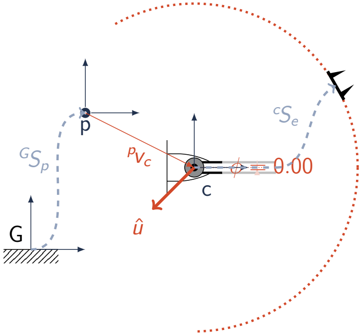
Prismatic Joint Column



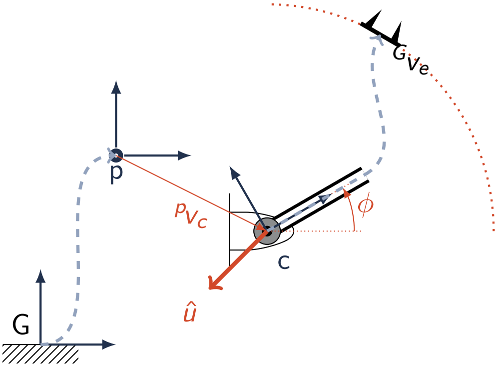
$$\begin{pmatrix} \mathbf{j}_{r_c} \\ \mathbf{j}_{p_c} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ {}^{G}\mathbf{u}_c \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ {}^{G}\hbar_c \otimes \mathbf{u}_c \otimes {}^{G}\hbar_c^* \end{pmatrix}$$

Revolute Joint

Animation

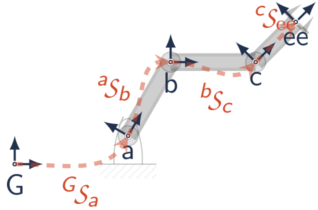


Revolute Joint Column



$$\begin{pmatrix} \mathbf{j}_{r_c} \\ \mathbf{j}_{p_c} \end{pmatrix} = \begin{pmatrix} {}^{G}\mathbf{u}_c \\ {}^{G}\mathbf{u}_c \times \left({}^{G}\mathbf{v}_e - {}^{G}\mathbf{v}_c\right) \end{pmatrix} = \begin{pmatrix} {}^{G}\hbar_c \otimes \mathbf{u}_c \otimes {}^{G}\hbar_c^{*} \\ \left({}^{G}\hbar_c \otimes \mathbf{u}_c \otimes {}^{G}\hbar_c^{*}\right) \times \left({}^{G}\mathbf{v}_e - {}^{G}\mathbf{v}_c\right) \end{pmatrix}$$

Example: Simple Manipulator Jacobian



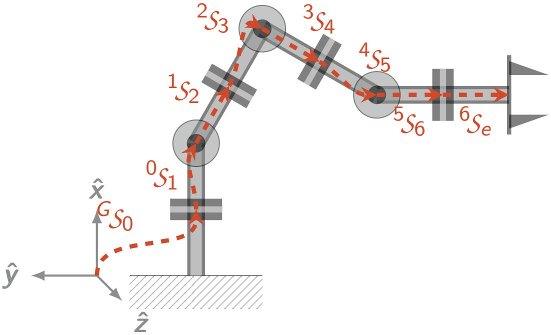
$$\mathbf{J} = \left[\begin{pmatrix} \mathbf{j}_{r_a} \\ \mathbf{j}_{p_a} \end{pmatrix} \quad \begin{pmatrix} \mathbf{j}_{r_b} \\ \mathbf{j}_{p_b} \end{pmatrix} \quad \begin{pmatrix} \mathbf{j}_{r_c} \\ \mathbf{j}_{p_c} \end{pmatrix} \right]$$

- a :
$$\mathbf{j}_{r_a} = {}^G\hslash_a \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_a^*$$
$$\mathbf{j}_{p_a} = \mathbf{j}_{r_a} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_a\right)$$
- b :
$$\mathbf{j}_{r_b} = {}^G\hslash_b \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_b^*$$
$$\mathbf{j}_{p_b} = \mathbf{j}_{r_b} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_b\right)$$
- c :
$$\mathbf{j}_{r_c} = {}^G\hslash_c \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_c^*$$
$$\mathbf{j}_{p_c} = \mathbf{j}_{r_c} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_c\right)$$

$$\mathbf{J} = \begin{bmatrix} \overbrace{{}^G\hslash_a \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_a^*}^{\mathbf{j}_{r_a}} & \overbrace{{}^G\hslash_b \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_b^*}^{\mathbf{j}_{r_b}} & \overbrace{{}^G\hslash_c \otimes \hat{\boldsymbol{k}} \otimes {}^G\hslash_c^*}^{\mathbf{j}_{r_c}} \\ \underbrace{\mathbf{j}_{r_a} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_a\right)}_{\mathbf{j}_{p_a}} & \underbrace{\mathbf{j}_{r_b} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_b\right)}_{\mathbf{j}_{p_b}} & \underbrace{\mathbf{j}_{r_c} \times \left({}^G\mathbf{v}_{\mathrm{ee}} - {}^G\mathbf{v}_c\right)}_{\mathbf{j}_{p_c}} \end{bmatrix}$$

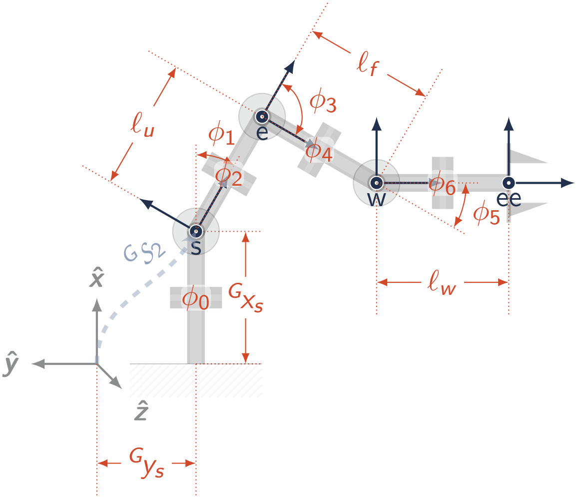
MINES

Exercise: Schunk LWA4D Jacobian



Exercise: Schunk LWA4D Jacobian

continued – transforms



- $^{G}\mathcal{S}_0 = \exp\left(\frac{\phi_0}{2}\hat{\imath}\right) + \frac{1}{2}{}^{G}v_s \otimes \exp\left(\frac{\phi_0}{2}\hat{\imath}\right)\varepsilon$
- $^{0}\mathcal{S}_1 = \exp\left(\frac{\phi_1}{2}\hat{k}\right)$
- $^{1}\mathcal{S}_2 = \exp\left(\frac{\phi_2}{2}\hat{\imath}\right)$
- $^{2}\mathcal{S}_3 = \exp\left(\frac{\phi_3}{2}\hat{k}\right) + \frac{1}{2}\ell_u\hat{\imath} \otimes \exp\left(\frac{\phi_3}{2}\hat{k}\right)\varepsilon$
- $^{3}\mathcal{S}_4 = \exp\left(\frac{\phi_4}{2}\hat{\imath}\right) + \frac{1}{2}\ell_f\hat{\imath} \otimes \exp\left(\frac{\phi_4}{2}\hat{\imath}\right)\varepsilon$
- $^{4}\mathcal{S}_5 = \exp\left(\frac{\phi_5}{2}\hat{k}\right)$
- $^{5}\mathcal{S}_6 = \exp\left(\frac{\phi_6}{2}\hat{\imath}\right)$

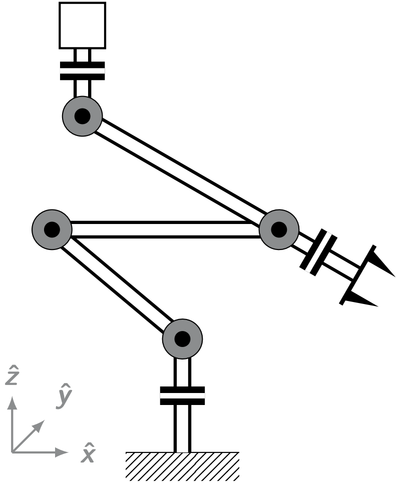
Exercise: Schunk LWA4D Jacobian

continued – Jacobian

-
-
-

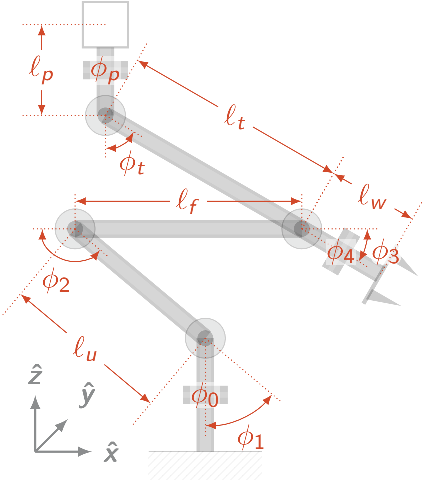
Exercise: EOD 510 Packbot Jacobian

http://endeavorrobotics.com/products



Exercise: EOD 510 Packbot Jacobian

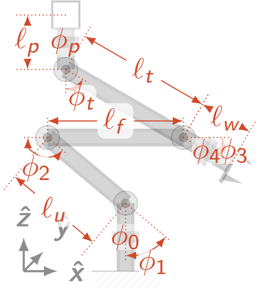
continued – transforms



- $^{G}\mathcal{S}_0 = \exp\left(\frac{\phi_0}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}\,{}^{G}v_0 \otimes \exp\left(\frac{\phi_0}{2}\hat{\boldsymbol{k}}\right)\varepsilon$
- $^{0}\mathcal{S}_1 = \exp\left(\frac{\phi_1}{2}\hat{\boldsymbol{\jmath}}\right) + \frac{1}{2}\,{}^{0}v_1 \otimes \exp\left(\frac{\phi_1}{2}\hat{\boldsymbol{\jmath}}\right)\varepsilon$
- $^{1}\mathcal{S}_2 = \exp\left(\frac{\phi_2}{2}\hat{\boldsymbol{\jmath}}\right) + \frac{1}{2}\ell_u\hat{\boldsymbol{k}} \otimes \exp\left(\frac{\phi_2}{2}\hat{\boldsymbol{\jmath}}\right)\varepsilon$
- $^{2}\mathcal{S}_3 = \exp\left(\frac{\phi_3}{2}\hat{\boldsymbol{\jmath}}\right) + \frac{1}{2}\ell_f\hat{\boldsymbol{k}} \otimes \exp\left(\frac{\phi_3}{2}\hat{\boldsymbol{\jmath}}\right)\varepsilon$
- $^{3}\mathcal{S}_4 = \exp\left(\frac{\phi_4}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}\ell_w\hat{\boldsymbol{k}} \otimes \exp\left(\frac{\phi_4}{2}\hat{\boldsymbol{k}}\right)\varepsilon$
- $^{3}\mathcal{S}_t = \exp\left(\frac{\phi_t}{2}\hat{\boldsymbol{\jmath}}\right) + \frac{1}{2}(-\ell_t)\hat{\boldsymbol{k}} \otimes \exp\left(\frac{\phi_t}{2}\hat{\boldsymbol{\jmath}}\right)\varepsilon$
- $^{t}\mathcal{S}_p = \exp\left(\frac{\phi_p}{2}\hat{\boldsymbol{k}}\right) + \frac{1}{2}(-\ell_p)\hat{\boldsymbol{k}} \otimes \exp\left(\frac{\phi_p}{2}\hat{\boldsymbol{k}}\right)\varepsilon$

Exercise: EOD 510 Packbot Jacobian

continued – Jacobian structure



$$\begin{bmatrix} \omega_4 \\ \dot{\mathbf{v}}_4 \\ \omega_p \\ \dot{\mathbf{v}}_p \end{bmatrix} = \begin{bmatrix} \mathbf{J}_4 \\ \mathbf{J}_p \end{bmatrix} \dot{\phi}$$

$$\begin{bmatrix} \omega_4 \\ \dot{\mathbf{v}}_4 \\ \omega_p \\ \dot{\mathbf{v}}_p \end{bmatrix} = \begin{bmatrix} \begin{pmatrix} \mathbf{j}_{r0} \\ \mathbf{j}_{p0} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{r1} \\ \mathbf{j}_{p1} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{r2} \\ \mathbf{j}_{p2} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{r3} \\ \mathbf{j}_{p3} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{r4} \\ \mathbf{j}_{p4} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{rt} \\ \mathbf{j}_{pt} \end{pmatrix}_4 & \begin{pmatrix} \mathbf{j}_{rp} \\ \mathbf{j}_{pp} \end{pmatrix}_4 \\ \begin{pmatrix} \mathbf{j}_{r0} \\ \mathbf{j}_{p0} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{r1} \\ \mathbf{j}_{p1} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{r2} \\ \mathbf{j}_{p2} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{r3} \\ \mathbf{j}_{p3} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{r4} \\ \mathbf{j}_{p4} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{rt} \\ \mathbf{j}_{pt} \end{pmatrix}_p & \begin{pmatrix} \mathbf{j}_{rp} \\ \mathbf{j}_{pp} \end{pmatrix}_p \end{bmatrix} \dot{\phi}$$

Exercise: EOD 510 Packbot Jacobian

continued – Jacobian Columns

- $\begin{pmatrix} \mathbf{j}_{r0} \\ \mathbf{j}_{p0} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{r1} \\ \mathbf{j}_{p1} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{r2} \\ \mathbf{j}_{p2} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{r3} \\ \mathbf{j}_{p3} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{r4} \\ \mathbf{j}_{p4} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{rt} \\ \mathbf{j}_{pt} \end{pmatrix}_4 =$
- $\begin{pmatrix} \mathbf{j}_{rp} \\ \mathbf{j}_{pp} \end{pmatrix}_4 =$

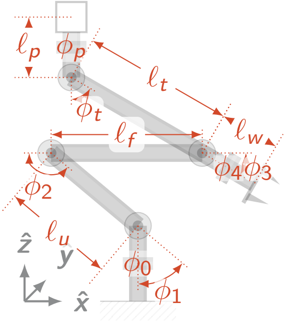
- $\begin{pmatrix} \mathbf{j}_{r0} \\ \mathbf{j}_{p0} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{r1} \\ \mathbf{j}_{p1} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{r2} \\ \mathbf{j}_{p2} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{r3} \\ \mathbf{j}_{p3} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{r4} \\ \mathbf{j}_{p4} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{rt} \\ \mathbf{j}_{pt} \end{pmatrix}_p =$
- $\begin{pmatrix} \mathbf{j}_{rp} \\ \mathbf{j}_{pp} \end{pmatrix}_p =$

Exercise: EOD 510 Packbot Jacobian

continued – Jacobian Matrix

$$\begin{bmatrix}\omega_4\\ \dot{\mathbf{v}}_4\\ \omega_p\\ \dot{\mathbf{v}}_p\end{bmatrix} =$$

$$\begin{bmatrix}\dot\phi_0\\ \dot\phi_1\\ \dot\phi_2\\ \dot\phi_3\\ \dot\phi_4\\ \dot\phi_t\\ \dot\phi_p\end{bmatrix}$$



Outline

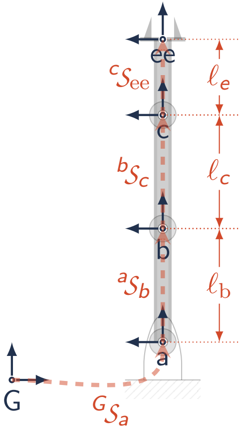
Mathematical Singularity

Definition: Mathematical Singularity

A **singularity** is a point at which a mathematical object is undefined or not "well-behaved."

Example: $f(x) = \frac{1}{x}$ has a singularity at $x = 0$.

Kinematic Singularities



$$
\mathbf{J} = \begin{bmatrix} \mathbf{J}_{\omega,\hat{\imath}} \\ \mathbf{J}_{\omega,\hat{\jmath}} \\ \mathbf{J}_{\omega,\hat{k}} \\ \mathbf{J}_{\dot{v},\hat{\imath}} \\ \mathbf{J}_{\dot{y},\hat{\jmath}} \\ \mathbf{J}_{\dot{z},\hat{k}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \\ -\ell_b - \ell_c - \ell_e & -\ell_c - \ell_e & -\ell_e \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}
$$

No possible instantaneous translation in $\hat{\jmath}$

Damped Least Squares

aka Levenberg–Marquardt

Computation

Pseudoinverse:

- $\mathbf{A}^{+} = \left(\mathbf{A}^{T}\mathbf{A}\right)^{-1}\mathbf{A}^{T}$ (independent cols)
- $\mathbf{A}^{+} = \mathbf{A}^{T}\left(\mathbf{A}\mathbf{A}^{T}\right)^{-1}$ (independent rows)

Damped Pseudoinverse: For constant k:

- $\mathbf{A}^{\dagger} = \left(\mathbf{A}^{T}\mathbf{A} + k\mathbf{I}\right)^{-1}\mathbf{A}^{T}$
- $\mathbf{A}^{\dagger} = \mathbf{A}^{T}\left(\mathbf{A}\mathbf{A}^{T} + k\mathbf{I}\right)^{-1}$

Damped Least Squares:

- $\mathbf{A}\mathbf{x} = \mathbf{b}$
- $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}$

Properties

- $\left(\mathbf{A}^{T}\mathbf{A} + k\mathbf{I}\right)^{-1}$ is invertible (nonsingular)
- But, introduces error
- Typical robots: $k \approx .001$

Singular Value Decomposition (SVD)

Definition (Singular Value Decomposition (SVD))

The Singular Value Decomposition (SVD) of $m \times n$ matrix $\mathbf{A}$ is the factorization:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \begin{bmatrix} \mathbf{u}_1 & \ldots \mathbf{u}_m \end{bmatrix} \begin{bmatrix} \sigma_1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_m \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \ldots \mathbf{v}_n \end{bmatrix}^T$$

Where:

- $\mathbf{U}$ is an $m \times m$ orthogonal matrix whose columns are the *left-singular vectors* of $\mathbf{A}$
- $\mathbf{\Sigma}$ is an $m \times n$ diagonal matrix containing the *singular values* of $\mathbf{A}$
- $\mathbf{V}$ is an $n \times n$ orthogonal matrix whose columns are the *right-singular vectors* of $\mathbf{A}$

SVD Pseudoinverse

Pseudoinverse via SVD

$$\mathbf{A}^+ = \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^T$$

$$= \begin{bmatrix}\mathbf{v}_1 & \ldots \mathbf{v}_n\end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1} & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & \frac{1}{\sigma_m} \end{bmatrix} \begin{bmatrix}\mathbf{u}_1 & \ldots \mathbf{u}_m\end{bmatrix}^T$$

$$= \sum_{i=1}^{m} \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T$$

Damping Factor

$$\mathbf{A}^\dagger = \sum_{i=1}^{m} \frac{\sigma_i^2}{\sigma_i^2 + k^2}\mathbf{v}_i\mathbf{u}_i^T$$

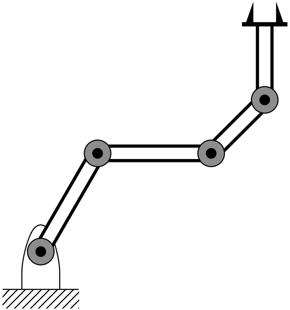
Minimum Singular Value

$$\mathbf{A}^\dagger = \sum_{i=1}^{m} \begin{cases} \frac{1}{\sigma_i}\mathbf{v}_i\mathbf{u}_i^T & \text{when } \sigma_i \geq \epsilon \\ \mathbf{0} & \text{when } \sigma_i < \epsilon \end{cases}$$

Outline

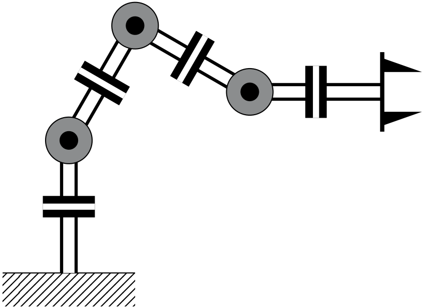
Redundant Manipulator

Planar



(more than 3 joints)

3D



(more than 6 joints)

Matrix Nullspace

Definition: Matrix Nullspace

Then nullspace of matrix **A** is the set of vectors whose product with **A** is zero.

$$\mathrm{null}(\mathbf{A}) = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{0}\}$$

Example

$$\underbrace{\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}}_{\mathbf{x}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T \in \mathrm{null}\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right)$$

Jacobian Nullspace

- General Jacobian relationship:

$$\begin{bmatrix} \omega \\ \dot{v} \end{bmatrix} = \mathbf{J}\dot{\phi}$$

- Vectors in Jacobian Nullspace:

$$\mathbf{0} = \mathbf{J}\dot{\phi}_{\mathrm{null}}$$

Nullspace joint velocities do not affect workspace velocity.

Nullspace Projection

i.e., constructing nullspace vectors

Projection Matrix

$$\dot{\phi}_{\mathrm{null}} = (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\dot{\phi}$$

- $\dot{\phi}$: any configuration velocity
- $(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})$: Nullspace projection matrix
- $\dot{\phi}_{\mathrm{null}}$: Nullspace configuration velocity
- That is: $\mathbf{0} = \mathbf{J}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\dot{\phi}$

Workspace Control

$$\dot{\phi}_{\mathrm{cmd}} = \mathbf{J}^{+} \begin{bmatrix} \omega_{\mathrm{ref}} \\ \dot{\mathbf{v}}_{\mathrm{ref}} \end{bmatrix} + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\dot{\phi}_{\mathrm{ref}}$$

- Variables:
  - ω_{ref}: Reference rotational velocity
  - $\dot{\mathbf{v}}_{\mathrm{ref}}$: Reference translational velocity
  - $\dot{\phi}_{\mathrm{ref}}$: Reference joint velocity
  - $\dot{\phi}_{\mathrm{cmd}}$: Command joint velocity
- Least-squares tracking of $\begin{bmatrix} \omega_{\mathrm{ref}} & \dot{v}_{\mathrm{ref}} \end{bmatrix}^{T}$
- "Best-effort" tracking of $\dot{\phi}_{\mathrm{ref}}$

Workspace Control Variations

Feed-forward Velocity Control:

$$\dot{\phi}_{\mathrm{cmd}} = \mathbf{J}^{+} \begin{bmatrix} \omega_{\mathrm{ref}} \\ \dot{\mathbf{v}}_{\mathrm{ref}} \end{bmatrix}$$

Proportional Position Control:

$$\dot{\phi}_{\mathrm{cmd}} = \mathbf{J}^{+} \left(\begin{bmatrix} \omega_{\mathrm{ref}} \\ \dot{\mathbf{v}}_{\mathrm{ref}} \end{bmatrix} + \mathbf{K}_p \begin{bmatrix} \ln\left(\hbar_{\mathrm{act}} \otimes \hbar^{*}_{\mathrm{ref}}\right) \\ \mathbf{v}_{\mathrm{ref}} - \mathbf{v}_{\mathrm{act}} \end{bmatrix} \right)$$

Nullspace Projection:

$$\dot{\phi}_{\mathrm{cmd}} = \mathbf{J}^{+} \left(\begin{bmatrix} \omega_{\mathrm{ref}} \\ \dot{\mathbf{v}}_{\mathrm{ref}} \end{bmatrix} + \mathbf{K}_p \begin{bmatrix} \ln\left(\hbar_{\mathrm{act}} \otimes \hbar^{*}_{\mathrm{ref}}\right) \\ \mathbf{v}_{\mathrm{ref}} - \mathbf{v}_{\mathrm{act}} \end{bmatrix} \right) + (\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\dot{\phi}_{\mathrm{ref}}$$

Joint Centering:

$$\dot{\phi}_{\mathrm{cmd}} = \mathbf{J}^{+} \left(\begin{bmatrix} \omega_{\mathrm{ref}} \\ \dot{\mathbf{v}}_{\mathrm{ref}} \end{bmatrix} + \mathbf{K}_p \begin{bmatrix} \ln\left(\hbar_{\mathrm{act}} \otimes \hbar^{*}_{\mathrm{ref}}\right) \\ \mathbf{v}_{\mathrm{ref}} - \mathbf{v}_{\mathrm{act}} \end{bmatrix} \right) + \mathbf{K}_{\phi}(\mathbf{I} - \mathbf{J}^{+}\mathbf{J})\left(\phi_{\mathrm{center}} - \phi_{\mathrm{act}}\right)$$

References

Reading: Buss, Samuel R. "Introduction to inverse kinematics with Jacobian transpose, pseudoinverse and damped least squares methods." Technical Report. 2004

Libraries:

- Basic Linear Algebra Subprograms (BLAS): matrix and vector arithmetic
- Linear Algebra Package (LAPACK): matrix decompositions, linear equations, eigenvalues, singular values
- Optimized Implementations
  - Automatically Tuned Linear Algebra System (ATLAS)
  - Intel Math Kernel Library (MKL)
  - OpenBLAS (formerly GotoBLAS)