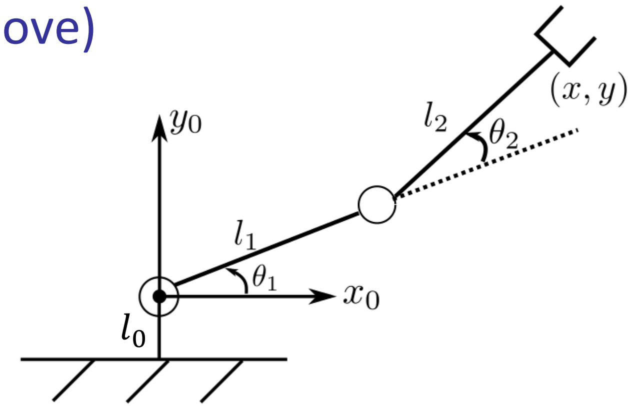
COMS W4733: Computational Aspects of Robotics

Lecture 8: Kinematics Review

Instructor: Tony Dear

Open-Chain Manipulators

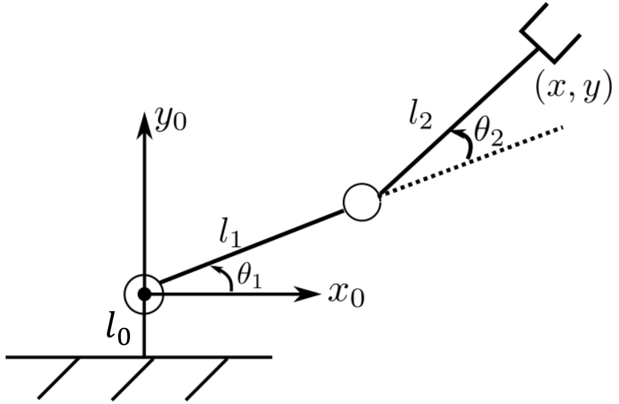
- Fixed base, alternating links and joints connect to end effector
- *Notation*: Links go from 0 to n (link 0 does not move)
  - Possible to have link constants of 0!
- Joint DOFs go from 1 to n (joint i moves link i)
  - Revolute: θ_i
  - Prismatic: d_i
- Kinematics: Mapping between joint variables and end effector



Kinematics

- Nonlinear mapping from joint space to operational space: $\boldsymbol{x}_e = k(\boldsymbol{q})$
  - Position and orientation of end effector frame relative to base frame
  - *Notation*: Frame $i-1$ for joint i; frame i on end effector

- Positions (x, y, z) generally given by single equations
  - May be possible to find geometrically in simple cases



- In 2D, orientation also a single variable, e.g. ϕ
- In 3D, orientation is a rotation matrix!

- General kinematic map given by a homogeneous transformation $\boldsymbol{T}_e^0 = \boldsymbol{A}_e^0$

Forward Kinematics

- Want to find analytical form of FK map, i.e. the homogeneous transform $\boldsymbol{T}_e^0$
- Only unknowns should be joint variables θ_i or d_i

- DH convention provides method for both assigning frames as well as finding transformations between successive frames, summarized as four parameters
- Each transform (DH table row) consists of one joint variable plus three constants

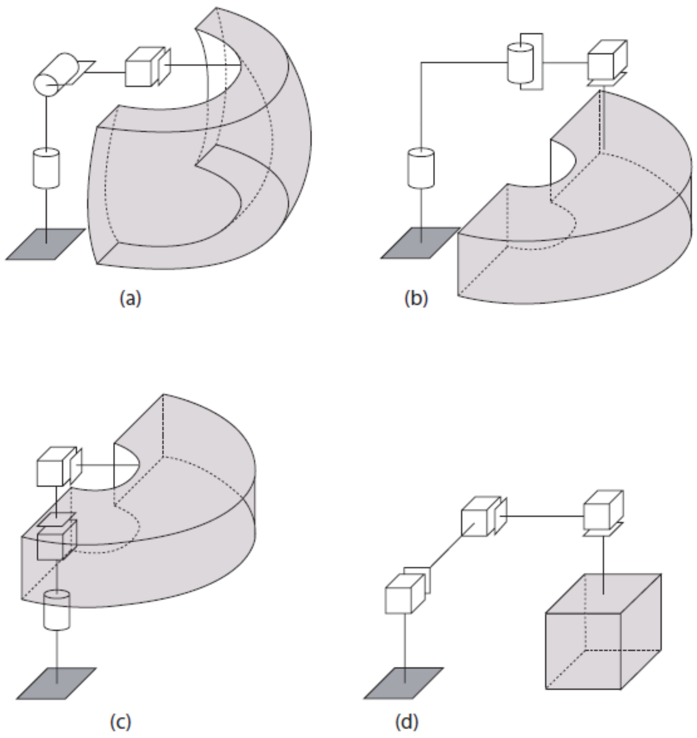
$$\boldsymbol{A}_i^{i-1} = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

- Composing all n transforms provides FK map: $\boldsymbol{T}_n^0 = \boldsymbol{A}_1^0 \boldsymbol{A}_2^1 \cdots \boldsymbol{A}_{n-1}^{n-2} \boldsymbol{A}_n^{n-1}$

Workspaces

Spherical arm

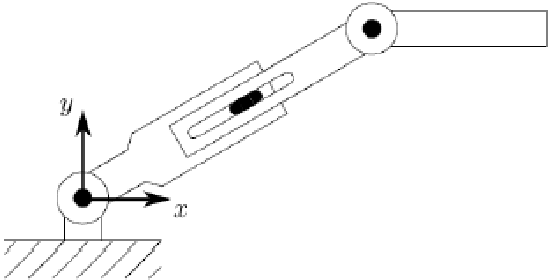
SCARA arm

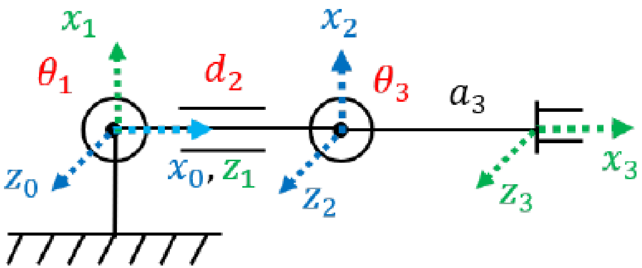


Cylindrical arm

Cartesian arm

Example: Planar RPR



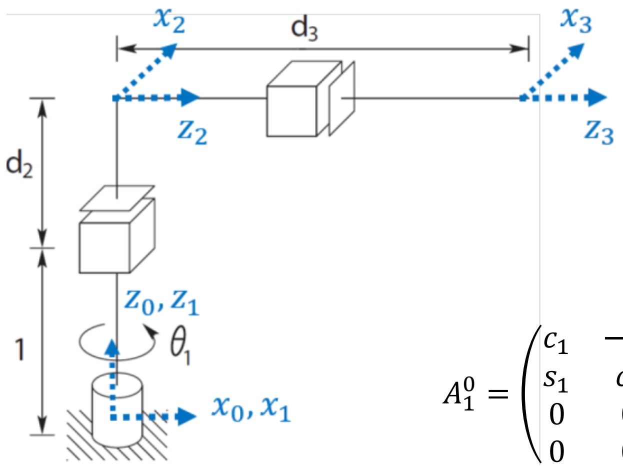


Link	a_i	α_i	d_i	θ_i
1	0	90	0	$\theta_1 + 90$
2	0	-90	d_2	0
3	a_3	0	0	$\theta_3 - 90$

$$A_1^0 = \begin{pmatrix} -s_1 & 0 & c_1 & 0 \\ c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2^1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_3^2 = \begin{pmatrix} s_3 & c_3 & 0 & a_3 s_3 \\ -c_3 & s_3 & 0 & -a_3 c_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2^0 = A_1^0 A_2^1 = \begin{pmatrix} -s_1 & -c_1 & 0 & d_2 c_1 \\ c_1 & -s_1 & 0 & d_2 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{pmatrix} c_{13} & -s_{13} & 0 & a_3 c_{13} + d_2 c_1 \\ s_{13} & c_{13} & 0 & a_3 s_{13} + d_2 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example: Cylindrical Arm



Link	a_i	α_i	d_i	θ_i
1	0	0	0	θ_1
2	0	90	d_2+1	90
3	0	0	d_3	0

$$A_1^0 = \begin{pmatrix} c_1 & -s_1 & 0 & 0 \\ s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_2^1 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & d_2+1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_3^2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_2^0 = A_1^0 A_2^1 = \begin{pmatrix} -s_1 & 0 & c_1 & 0 \\ c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & d_2+1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{pmatrix} -s_1 & 0 & c_1 & d_3 c_1 \\ c_1 & 0 & s_1 & d_3 s_1 \\ 0 & 1 & 0 & d_2+1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Analytical Inverse Kinematics

- *Idea*: We tell the robot where to go and let it figure out how to get there
- Invert the nonlinear FK mapping: $\boldsymbol{q} = k^{-1}(\boldsymbol{x}_e)$
  - Generally an algebraic process, solving system of equations
  - Possibly multiple, infinite, or no solutions, depending on robot's workspace

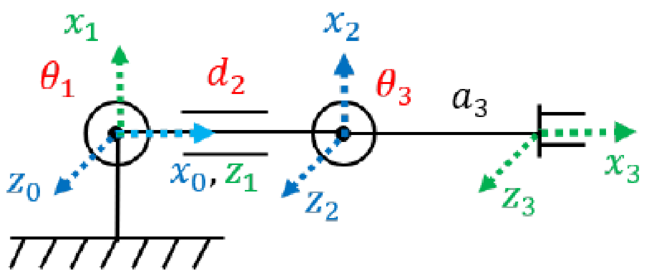
- For positions or 2D orientation, solve FK equations directly
- We have not discussed how to solve for 3D orientations (entire rotation matrices)!

- Common tricks: Exploit trigonometric identities ($\sin^2\theta + \cos^2\theta = 1$), Atan2 function
- Always relate algebraic solutions back to geometric intuition

Example: Planar RPR

$$A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{pmatrix} c_{13} & -s_{13} & 0 & a_3 c_{13} + d_2 c_1 \\ s_{13} & c_{13} & 0 & a_3 s_{13} + d_2 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\phi = \theta_1 + \theta_3$$



- Substitute in known ϕ to get rid of θ_3 unknown
- Isolate θ_1, or c_1 and s_1 simultaneously
- Common factors cancel out with Atan2

$$p_x = a_3 c_\phi + d_2 c_1 \rightarrow c_1 = \frac{1}{d_2}(p_x - a_3 c_\phi)$$

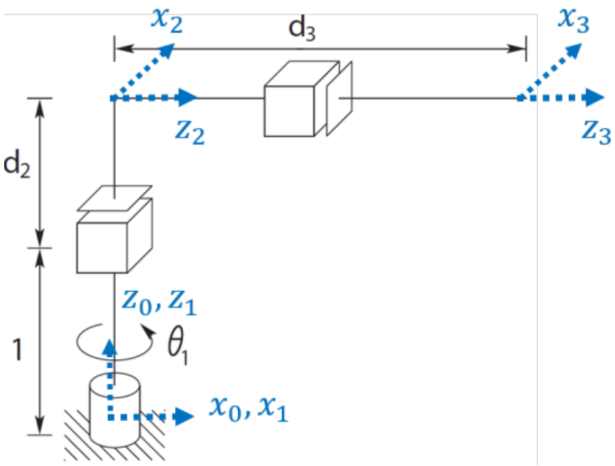
$$p_y = a_3 s_\phi + d_2 s_1 \rightarrow s_1 = \frac{1}{d_2}(p_y - a_3 s_\phi)$$

- Atan2 returns one unique solution
- With one joint solved, others are also specified

$$\theta_1 = \text{Atan2}(p_y - a_3 s_\phi, p_x - a_3 c_\phi)$$

- Why only one solution and not two or more?
- Any edge cases of no solutions?

Example: Cylindrical Arm



$$A_3^0 = A_1^0 A_2^1 A_3^2 = \begin{pmatrix} -s_1 & 0 & c_1 & d_3 c_1 \\ c_1 & 0 & s_1 & d_3 s_1 \\ 0 & 1 & 0 & d_2 + 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- How many solutions for 3D position? (p_x, p_y, p_z)
- What if prismatic joint variables can be negative?

- Is end effector orientation a degree of freedom?

$$p_x = d_3 c_1$$
$$p_y = d_3 s_1$$
$$p_z = d_2 + 1$$

$$\theta_1 = \text{Atan2}(p_y, p_x)$$
$$d_2 = p_z - 1$$
$$d_3 = \frac{p_x}{c_1} = \frac{p_y}{s_1}$$

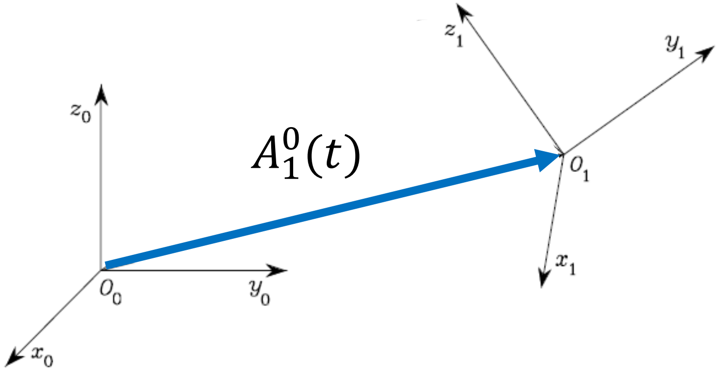
Differential Kinematics

- *Idea*: We want a mapping between joint and end effector velocities
- Algebraically turns out to be differential of the forward kinematics!

- Homogeneous transforms indicate displacements (directions) between frames
- If we also have transform *rates* as functions of time, then we have **velocities**

- Time derivative of translation: linear velocity $\dot{\boldsymbol{p}}_e(\boldsymbol{q})$
- Time derivative of rotation: angular velocity

$$\boldsymbol{S}(\boldsymbol{\omega}(t)) = \dot{\boldsymbol{R}}(t)\boldsymbol{R}^T(t) \qquad \boldsymbol{S}(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$



The Jacobian

- Differential of FK gives us a *linear, configuration-dependent* mapping

$$\boldsymbol{v}_e = \begin{pmatrix} \dot{\boldsymbol{p}}_e \\ \boldsymbol{\omega}_e \end{pmatrix} = \begin{pmatrix} \boldsymbol{J}_P(\boldsymbol{q}) \\ \boldsymbol{J}_O(\boldsymbol{q}) \end{pmatrix} \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$$

- Linear velocity Jacobian

$$\boldsymbol{J}_P(\boldsymbol{q}) = \begin{pmatrix} \frac{\partial p_{e,x}}{\partial q_1} & \cdots & \frac{\partial p_{e,x}}{\partial q_n} \\ \frac{\partial p_{e,y}}{\partial q_1} & \cdots & \frac{\partial p_{e,y}}{\partial q_n} \\ \frac{\partial p_{e,z}}{\partial q_1} & \cdots & \frac{\partial p_{e,z}}{\partial q_n} \end{pmatrix}$$

Or column by column:

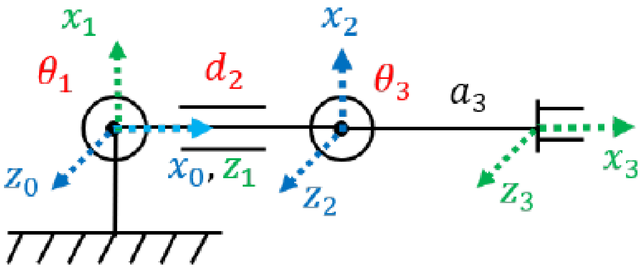
$$[\boldsymbol{J}_{Pi}] = \begin{cases} [\boldsymbol{z}_{i-1}^0] & \text{prismatic} \\ [\boldsymbol{z}_{i-1}^0 \times (\boldsymbol{p}_e - \boldsymbol{p}_{i-1})] & \text{revolute} \end{cases}$$

- Angular velocity Jacobian

$$[\boldsymbol{J}_{Oi}] = \begin{cases} [\boldsymbol{0}] & \text{prismatic} \\ [\boldsymbol{z}_{i-1}^0] & \text{revolute} \end{cases}$$

Example: Planar RPR

- Sanity check: Planar manipulator has FK $z = 0$
- Should expect Jacobian rows $\dot{z}, \omega_x, \omega_y = 0$



- Linear velocity Jacobian: Partial derivatives of $\boldsymbol{p}_3^0$

$$\boldsymbol{J}_P = \begin{pmatrix} -a_3 s_{13} - d_2 s_1 & c_1 & -a_3 s_{13} \\ a_3 c_{13} + d_2 c_1 & s_1 & a_3 c_{13} \\ 0 & 0 & 0 \end{pmatrix}$$

$$\boldsymbol{J}_O = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

- Angular velocity Jacobian: z_{i-1}^0 of revolute joints
  - If first joint revolute, always $(0,0,1)^T$; otherwise $(0,0,0)^T$
  - Try looking for parallel z axes, especially in planar cases!

$$A_1^0 = \begin{pmatrix} -s_1 & 0 & c_1 & 0 \\ c_1 & 0 & s_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

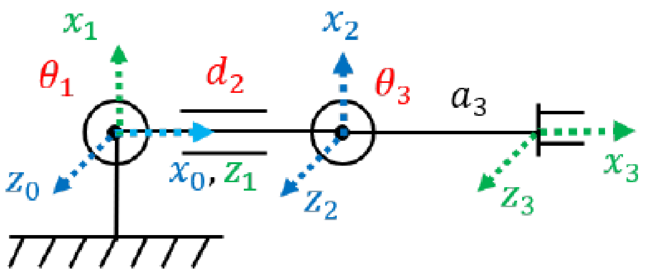
$$A_2^0 = \begin{pmatrix} -s_1 & -c_1 & 0 & d_2 c_1 \\ c_1 & -s_1 & 0 & d_2 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$A_3^0 = \begin{pmatrix} c_{13} & -s_{13} & 0 & a_3 c_{13} + d_2 c_1 \\ s_{13} & c_{13} & 0 & a_3 s_{13} + d_2 s_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Example: Planar RPR

- What are the singularities?

$$J_P = \begin{pmatrix} -a_3 s_{13} - d_2 s_1 & c_1 & -a_3 s_{13} \\ a_3 c_{13} + d_2 c_1 & s_1 & a_3 c_{13} \\ 0 & 0 & 0 \end{pmatrix} \qquad J_O = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

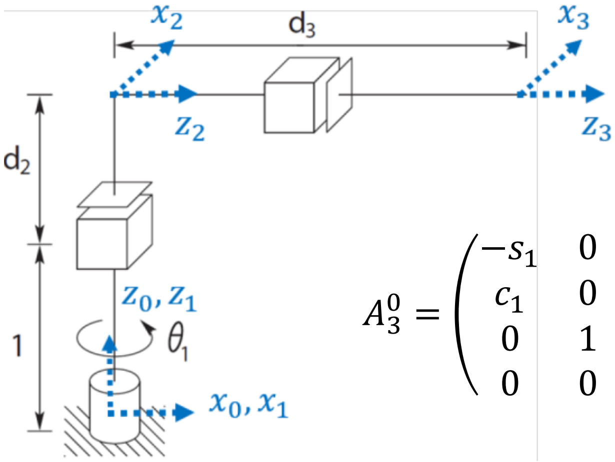


- Since manipulator is planar, we only need to consider "effective" Jacobian consisting of rows $\dot{x}$, $\dot{y}$, ω_z (last one can be thought of as $\dot{\phi}$)

$$\det \begin{pmatrix} -a_3 s_{13} - d_2 s_1 & c_1 & -a_3 s_{13} \\ a_3 c_{13} + d_2 c_1 & s_1 & a_3 c_{13} \\ 1 & 0 & 1 \end{pmatrix} = d_2$$

- Why is there no elbow singularity (stretched-out configuration) like that of the RR arm?

Example: Cylindrical Arm



$$A_3^0 = \begin{pmatrix} -s_1 & 0 & c_1 & d_3 c_1 \\ c_1 & 0 & s_1 & d_3 s_1 \\ 0 & 1 & 0 & d_2 + 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$J_P = \begin{pmatrix} -d_3 s_1 & 0 & c_1 \\ d_3 c_1 & 0 & s_1 \\ 0 & 1 & 0 \end{pmatrix}$$

Partial derivatives of FKs

$$J_O = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Only first joint is revolute!

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} -d_3 s_1 & 0 & c_1 \\ d_3 c_1 & 0 & s_1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{pmatrix}$$

- Are there singularities? Depends on the velocity directions
- Overall no, because we can always get $\dot{z}$ and ω_z

- When do we fail to span the plane ($\dot{x}$ and $\dot{y}$)?
- When $d_3 = 0$, first two rows are linearly dependent

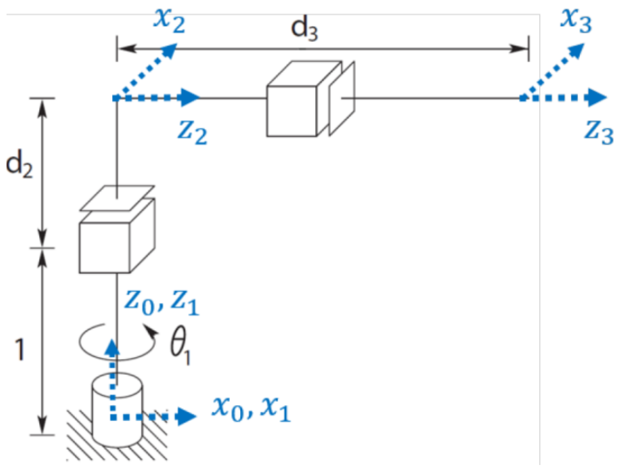
Inverse Differential Kinematics

- Similar idea to inverse pose kinematics: We tell the robot how fast the end effector should go and the robot figures out how to do it
- Since DK is a linear mapping, we can "invert" the Jacobian

- Underconstrained: Many joint velocity solutions for desired end effector velocity
  - Right pseudo-inverse produces solution with the smallest magnitude
  - Can also tweak solution to minimize selected joint velocity components or to push solution toward another criterion using null space mapping

- Overconstrained: No exact solutions since fewer DOFs than desired components
  - Left pseudo-inverse produces solution that minimizes error in actual end effector velocity

Example: Underconstrained Cylindrical Arm



$$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\\ \omega_x\\ \omega_y\\ \omega_z\end{pmatrix} = \begin{pmatrix}-d_3 s_1 & 0 & c_1\\ d_3 c_1 & 0 & s_1\\ 0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\\ 1 & 0 & 0\end{pmatrix}\begin{pmatrix}\dot{\theta}_1\\ \dot{d}_2\\ \dot{d}_3\end{pmatrix}$$

- Suppose $\boldsymbol{q} = (45°, 0,1)^T$ and we want $(\dot{x}, \dot{y})^T = (1,1)$
- Clearly $\dot{d}_2$ is free since $\dot{x}$ and $\dot{y}$ do not depend on it!
- Right pseudo-inverse should minimize it to 0:

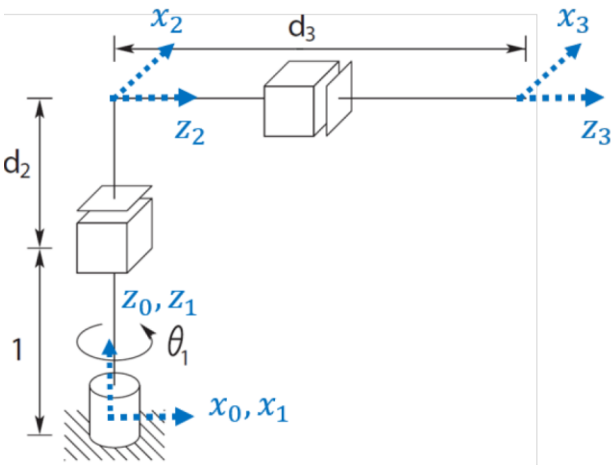
$$\dot{\boldsymbol{q}}^* = \begin{pmatrix}-d_3 s_1 & 0 & c_1\\ d_3 c_1 & 0 & s_1\end{pmatrix}_r^+ \begin{pmatrix}1\\1\end{pmatrix} = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}0\\0\\ \sqrt{2}\end{pmatrix}$$

- We can also add in the homogeneous solution:

$$\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}^* + (\boldsymbol{I} - \boldsymbol{J}_r^+\boldsymbol{J})\dot{\boldsymbol{q}}_0 = \dot{\boldsymbol{q}}^* + \begin{pmatrix}0 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{pmatrix}\dot{\boldsymbol{q}}_0$$

- This solution causes $\dot{d}_2$ to match any specified $(\dot{d}_2)_0$

Example: Overconstrained Cylindrical Arm



$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} = \begin{pmatrix} -d_3 s_1 & 0 & c_1 \\ d_3 c_1 & 0 & s_1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \dot{\theta}_1 \\ \dot{d}_2 \\ \dot{d}_3 \end{pmatrix}$$

- $\boldsymbol{q} = (45°, 0, 1)^T$ and we want $(\dot{x}, \dot{y}, \dot{z}, \omega_z)^T = (1,1,1,1)$
- Not possible to satisfy all four velocities simultaneously
- Left pseudo-inverse minimizes resulting error:

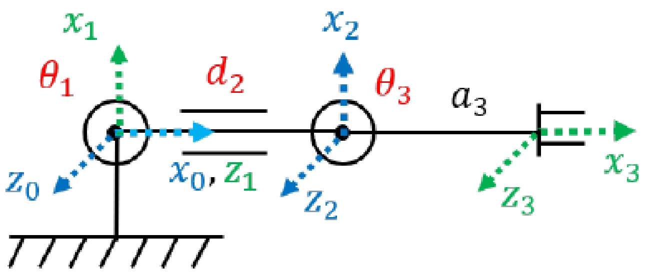
$$\dot{\boldsymbol{q}}^* = \begin{pmatrix} -d_3 s_1 & 0 & c_1 \\ d_3 c_1 & 0 & s_1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}_l^+ \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = (J^T J)^{-1} J^T \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.5 \\ 1 \\ \sqrt{2} \end{pmatrix}$$

- Compare to actual end effector velocities using $\dot{\boldsymbol{q}}^*$

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \omega_z \end{pmatrix} = \begin{pmatrix} -1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0.5 \\ 1 \\ \sqrt{2} \end{pmatrix} = \begin{pmatrix} 0.646 \\ 1.354 \\ 1 \\ 0.5 \end{pmatrix}$$

Example: Singular Planar RPR

- Suppose we want $(\dot{x}, \dot{y}, \omega_z)^T = (1,1,1)^T$ at the configuration $(\theta_1, d_2, \theta_3) = (0,0,0)$, $a_3 = 1$
- $\boldsymbol{J}$ loses rank, rows become linearly dependent



- This particular problem still has an exact solution since $\dot{x} = \omega_z$

$$\boldsymbol{J} = \begin{pmatrix} -a_3 s_{13} - d_2 s_1 & c_1 & -a_3 s_{13} \\ a_3 c_{13} + d_2 c_1 & s_1 & a_3 c_{13} \\ 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

- Damped least-squares pseudo-inverse:

$$\boldsymbol{J}^* = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T + k^2\boldsymbol{I})^{-1}$$

$\boldsymbol{J}^* = (\boldsymbol{J} + k^2\boldsymbol{I})^{-1}$ for square Jacobians

$$k = 0.01\!: \ \dot{q} = \begin{pmatrix} -.0001 \\ 1 \\ 1 \end{pmatrix} \qquad k = 0.1\!: \ \dot{q} = \begin{pmatrix} -.0101 \\ 1.0001 \\ 1.0001 \end{pmatrix} \qquad k = 1\!: \ \dot{q} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \qquad k = 2\!: \ \dot{q} = \begin{pmatrix} 0.4 \\ 0.2 \\ 0.2 \end{pmatrix}$$