Lecture 18, Nov 16, 2023

Manipulator Jacobians

Velocity

- Each joint gives us one degree of freedom $q_i = \begin{cases} \theta_i & \text{joint is revolute} \\ d_i & \text{joint is prismatic} \end{cases}$
- We want to know how, given a desired velocity of the end-effector, we can set the joint rates to achieve that velocity
- The manipulator Jacobian relates the end-effector velocity and angular velocity: $\boldsymbol{v} = \boldsymbol{J}(\boldsymbol{q})\dot{\boldsymbol{q}}$
  - $\boldsymbol{v} = \begin{bmatrix} \boldsymbol{v}_0^{ee} \\ \boldsymbol{\omega}_0^{ee} \end{bmatrix}$ is the velocity (including translational and angular) velocity of the end-effector
    - Note that this is expressed in frame 0, which is our world/inertial frame
  - $\dot{\boldsymbol{q}}$ are the joint rates
  - $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{m \times n}$ where $m \leq 6$ and n is the number of joints; in general it is a function of the current joint states
- Partition the Jacobian as $\boldsymbol{J}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{J}^{(v)}(\boldsymbol{q}) \\ \boldsymbol{J}^{(\omega)}(\boldsymbol{q}) \end{bmatrix}$, where one part is for linear velocity and the other part is for angular
  - Given an expression for the end-effector position we can simply differentiate it to get the translational velocity Jacobian
  - $\boldsymbol{J}^{(v)}(\boldsymbol{q}) = \dfrac{\partial \boldsymbol{r}_0^{ee}}{\partial \boldsymbol{q}^T}$ where $\boldsymbol{r}_0^{ee}$ is the position of the end-effector
  - Angular velocity however is more complicated since it's not the direct derivatives of the orientation variables
- For angular velocity $(\boldsymbol{\omega}_0^{ee})^\times \boldsymbol{C}_{0,n}\dot{\boldsymbol{C}}_{0,n}^T = \sum \boldsymbol{C}_{0,n}\dfrac{\partial \boldsymbol{C}_{0,n}^T}{\partial q_i}\dot{q}_i \equiv \sum_i (\boldsymbol{\nu}_i^{ee})^\times \dot{q}_i$
  - $\boldsymbol{C}_{0,i}$ is the rotation matrix from frame i to the world frame
  - Therefore $\boldsymbol{\omega}_0^{ee} = \sum_i \boldsymbol{\nu}_i^{ee}\dot{q}_i$ and so $\boldsymbol{J}^{(\omega)} = \begin{bmatrix} \boldsymbol{\nu}_1^{ee} & \cdots & \boldsymbol{\nu}_n^{ee} \end{bmatrix}$
- Using DH parameters:
  - Let $\underset{\rightarrow}{\rho}_i^j = \sum_{k=i}^{j-1} \underset{\rightarrow}{\rho}_k^{k+1}$ be the relative position of O_j from O_i
  - Let $\underset{\rightarrow}{\omega}_i^j = \sum_{k=i}^{j-1} \underset{\rightarrow}{\omega}_k^{k+1}$ be the angular velocity of link i with respect to link j
  - Let $\boldsymbol{C}_{ij} = \prod_{k=i}^{j-1} \boldsymbol{C}_{k,k+1}$ be the rotation matrix from frame j to frame i
- Then $\underset{\rightarrow}{v}^{ee} - \dot{\underset{\rightarrow}{\rho}}_0^{n+1}$ and $\underset{\rightarrow}{\omega}^{ee} = \underset{\rightarrow}{\omega}_0^n$
  - Note the velocity is to $n+1$ because we want the velocity of the end-effector (i.e. end of the last link), but the angular velocity is of the last link so it's to n
  - Note $\underset{\rightarrow}{\rho}_i^{i+1} = \underset{\rightarrow}{\boldsymbol{\mathcal{F}}}_i^T \boldsymbol{\rho}_i^{i+1}, \underset{\rightarrow}{\omega}_{i-1}^i = \underset{\rightarrow}{\boldsymbol{\mathcal{F}}}_i^T \omega_{i-1}^i$, i.e. $\boldsymbol{\rho}_i^{i+1}$ and $\boldsymbol{\omega}_{i-1}^i$ are both expressed in frame i
- For the angular velocity part:
  - $\underset{\rightarrow}{\omega}_{i-1}^i = \begin{cases} \dot{\theta}_i \underset{\rightarrow}{z}_i & \text{revolute joint} \\ \underset{\rightarrow}{0} & \text{prismatic joint} \end{cases}$
  - $\underset{\rightarrow}{\omega}^{ee} = \sum_{i=1}^n \varepsilon_i \dot{\theta} \underset{\rightarrow}{z}_i$
    - Note ε_i is 1 if the joint is revolute, otherwise 0
  - $\underset{\rightarrow}{z}_i = \underset{\rightarrow}{\boldsymbol{\mathcal{F}}}_i^T \mathbf{1}_3 \implies \boldsymbol{\omega}_0^{ee} = \sum_{i=1}^n \varepsilon_i \boldsymbol{C}_{0,i}\mathbf{1}_3\dot{\theta}_i$

- The Jacobian is then $\boldsymbol{J}^{(\omega)} = \begin{bmatrix} \boldsymbol{j}_1^{(\omega)} & \cdots & \boldsymbol{j}_n^{(\omega)} \end{bmatrix}$ where $\boldsymbol{j}_i^{(\omega)} = \varepsilon_i \boldsymbol{C}_{0,i} \mathbf{1}_3$
- For the translational velocity part:
  - $\underset{\to}{v}^{ee} = \underset{\to}{\rho}_0^{n+1\,\bullet} = \displaystyle\sum_{i=0}^{n} \underset{\to}{\rho}_i^{i+1\,\bullet}$
  - $\underset{\to}{\rho}_i^{i+1\,\bullet} = \underset{\to}{\rho}_i^{i+1\,\circ} + \underset{\to}{\omega}_0^i \times \underset{\to}{\rho}_i^{i+1}$
    - Recall $\underset{\to}{\rho}_i^{i+1} = d_i \underset{\to}{z}_i + a_i \underset{\to}{x}_i \implies \underset{\to}{\rho}_i^{i+1\,\circ} = (1-\varepsilon_i)\dot{d}_i \underset{\to}{z}_i + d_i \underset{\to}{z}_i^\circ + a_i \underset{\to}{x}_i^\circ$
    - But $\underset{\to}{x}_i^\circ = \underset{\to}{z}_i^\circ = \underset{\to}{0}$
  - Therefore $\underset{\to}{\rho}_i^{i+1\,\bullet} = (1-\varepsilon_i)\dot{d}_i \underset{\to}{z}_i + \underset{\to}{\omega}_0^i \times \underset{\to}{\rho}_i^{i+1}$
  - Substitute $\underset{\to}{\omega}_0^i = \displaystyle\sum_{k=1}^{i} \varepsilon_k \dot{\theta}_k \underset{\to}{z}_k$
  - So $\underset{\to}{v}^{ee} = \displaystyle\sum_{i=1}^{n} \left[(1-\varepsilon_i)\dot{d}_i \underset{\to}{z}_i + \sum_{k=1}^{i} \varepsilon_k \dot{\theta}_k \underset{\to}{z}_k \times \underset{\to}{\rho}_i^{i+1} \right]$
  - This reduces to $\underset{\to}{v}^{ee} = \displaystyle\sum_{i=1}^{n} \left[(1-\varepsilon_i)\dot{d}_i \underset{\to}{z}_i + \varepsilon_i \dot{\theta}_i \underset{\to}{z}_i \times \underset{\to}{\rho}_i^{n+1} \right]$
  - In the world frame, $\boldsymbol{v}_0^{ee} = \displaystyle\sum_{i=1}^{n} \left[(1-\varepsilon_i)\dot{d}_i \boldsymbol{C}_{0,i}\mathbf{1}_3 + \varepsilon_i \dot{\theta}_i \boldsymbol{C}_{0,i} \mathbf{1}_3^\times \boldsymbol{\rho}_i^{n+1} \right]$
  - Therefore $\boldsymbol{J}^{(v)} = \begin{bmatrix} \boldsymbol{j}_1^{(v)} & \cdots & \boldsymbol{j}_n^{(v)} \end{bmatrix}$ where $\boldsymbol{j}_i^{(v)} = (1-\varepsilon_i)\boldsymbol{C}_{0,i}\mathbf{1}_3 + \varepsilon_i \boldsymbol{C}_{0,1}\mathbf{1}_3^\times \boldsymbol{\rho}_i^{n+1}$
- $\boldsymbol{J} = \begin{bmatrix} \boldsymbol{j}_1 & \cdots & \boldsymbol{j}_n \end{bmatrix}$ where $\boldsymbol{j}_i = \begin{bmatrix} \boldsymbol{j}_i^{(v)} \\ \boldsymbol{j}_i^{(\omega)} \end{bmatrix} = \begin{cases} \begin{bmatrix} \boldsymbol{C}_{0,i}\mathbf{1}_3^\times \boldsymbol{\rho}_i^{n+1} \\ \boldsymbol{C}_{0,i}\mathbf{1}_3 \end{bmatrix} & \text{revolute joint} \\ \begin{bmatrix} \boldsymbol{C}_{0,i}\mathbf{1}_3 \\ \mathbf{0} \end{bmatrix} & \text{prismatic joint} \end{cases}$

Force

- Define the joint control force/torque as $\underset{\to}{\eta}_{i-1}^i = \eta_i \underset{\to}{z}_i = \begin{cases} \tau_i \underset{\to}{z}_i & \text{revolute joint} \\ f_i \underset{\to}{z}_i & \text{prismatic joint} \end{cases}$
  - This force or torque is between links $i-1$ and i
- We can obtain the actual control input force by taking the dot product of the joint forces with $\underset{\to}{z}_i$, since only 1 of 6 degrees of freedom of force is due to the input and the other are due to constraints
- How do we relate the control input force to the force delivered at the end-effector?
- Consider a free-body segment between links i and n; assume this is static (i.e. ignoring inertial forces)
  - The interlink force and torque are $\underset{\to}{\tau}_{i-1}^i = \underset{\to}{\tau}^{ee} + \underset{\to}{\rho}_i^{n+1} \times \underset{\to}{f}^{ee}$ and $\underset{\to}{f}_{i-1}^i = \underset{\to}{f}^{ee}$, derived from the FBD
- The control inputs are therefore $\eta_i = \begin{cases} \tau_i = \underset{\to}{z}_i \cdot \underset{\to}{\tau}^{ee} + \underset{\to}{z}_i \cdot \rho_i^{n+1} \times \underset{\to}{f}^{ee} & \text{revolute joint} \\ f_i = \underset{\to}{z}_i \cdot \underset{\to}{f}^{ee} & \text{prismatic joint} \end{cases}$
  - Expressed in world frame: $\eta_i = \begin{cases} \tau_i = (\boldsymbol{C}_{0,i}\mathbf{1}_3)^T \boldsymbol{\tau}_0^{ee} + (\boldsymbol{C}_{0,i}\mathbf{1}_3^\times \boldsymbol{\rho}_i^{n+1})^T \boldsymbol{f}_0^{ee} & \text{revolute joint} \\ f_i = (\boldsymbol{C}_{0,i}\mathbf{1}_3)^T \boldsymbol{f}_0^{ee} & \text{prismatic joint} \end{cases}$
- This gives $\boldsymbol{\eta} = \boldsymbol{J}^T(\boldsymbol{q})\boldsymbol{f}$ where $\boldsymbol{f} = \begin{bmatrix} \boldsymbol{f}_0^{ee} \\ \boldsymbol{\tau}_0^{ee} \end{bmatrix}$, where the Jacobian is the same as before

Acceleration

- $\boldsymbol{a} = \dot{\boldsymbol{v}} = \begin{bmatrix} \dot{\boldsymbol{v}}_0^{ee} \\ \dot{\boldsymbol{\omega}}_0^{ee} \end{bmatrix}$
- So $\boldsymbol{a} = \boldsymbol{J}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q})\dot{\boldsymbol{q}}$
- We can write $\dot{\boldsymbol{J}(\boldsymbol{q})}\dot{\boldsymbol{q}} = \operatorname{col}\left[\displaystyle\sum_{j=1}^{n}\sum_{k=1}^{n} \frac{\partial J_{ik}}{\partial q_j} \dot{q}_j \dot{q}_k \right]$

Kinematics

- Forward kinematics is finding $\boldsymbol{v}$ given $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$; this is easy if we have the Jacobian
- Inverse kinematics is the problem of finding $\dot{\boldsymbol{q}}$ given $\boldsymbol{v}$ (and integrating for $\boldsymbol{q}$); in general this is much more challenging
- If the Jacobian is square and invertible, then we can simply find $\dot{\boldsymbol{q}} = \boldsymbol{J}^{-1}(\boldsymbol{q})\boldsymbol{v}$
- If it is not invertible, assuming $m < n$ (i.e. we have more joints/DoF than spacial dimensions), we can try using the *pseudoinverse*
  - Provided $\operatorname{rank} \boldsymbol{J} = m$, $\boldsymbol{J}\boldsymbol{J}^T$ is invertible
  - Define the *(Moore-Penrose) pseudoinverse* $\boldsymbol{J}^\dagger = \boldsymbol{J}^T(\boldsymbol{J}\boldsymbol{J}^T)^{-1}$, so $\boldsymbol{J}\boldsymbol{J}^\dagger = \mathbf{1} \in \mathbb{R}^{m\times m}$
- Then in general if $\operatorname{rank} \boldsymbol{J} = m$, $\dot{\boldsymbol{q}} = \boldsymbol{J}^\dagger\boldsymbol{v} + (1 - \boldsymbol{J}^\dagger\boldsymbol{J})\boldsymbol{b}$, for any $\boldsymbol{b} \in \mathbb{R}^n$ (i.e. we have an infinite number of solutions)
  - Note that $(\mathbf{1} - \boldsymbol{J}^\dagger\boldsymbol{J})\boldsymbol{b} \in \ker \boldsymbol{J}$
  - Take $\boldsymbol{b} = \mathbf{0}$ if we want to minimize the joint rates
- What about square but non-invertible $\boldsymbol{J}$ or $\operatorname{rank} \boldsymbol{J} < m$?
  - In this case we have a *singularity* – we cannot solve for $\dot{\boldsymbol{q}}$ given an arbitrary $\boldsymbol{v}$
- At singularities, configurations with motion in certain directions may be unattainable
  - These often occur at boundaries of the workspace
  - Finite end-effector rates might imply infinite joint rates
  - Finite joint forces/torques might imply infinite end-effector forces and torques

Definition

A *singularity* occurs at $\boldsymbol{q}$ when $\operatorname{rank} \boldsymbol{J}(\boldsymbol{q}) < m$, or equivalently $\det(\boldsymbol{J}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})^T) = 0$, where m is the dimension of the workspace.

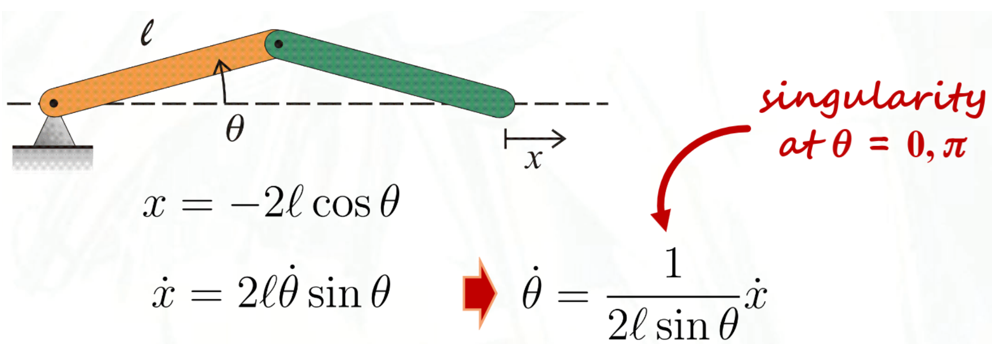

Figure 1: Simple singularity example.

- Singularities for the above arm occur at $\theta_3 = 0, \pi$ or $\theta_3 = -2\theta_2$
  - At $\theta_3 = -2\theta_2$, the end-effector will be on the z axis, so any θ_1 gives us the same end-effector position; therefore we can't solve for θ_1
  - At $\theta_3 = 0$, the links are in a straight line, so we can't get any motion along that line
  - At $\theta_3 = \pi$, the arm is folded back on itself, so again we can't get any motion on that line
- Translation and rotation of an end-effector can be theoretically uncoupled if we have a *wrist-partitioned arm*, if:
  - Last 3 joints are revolute with axes passing through the common centre E
  - Successive axes are not parallel
  - E can be placed arbitrarily in position space
- Practically however the end-effector is always displaced from E

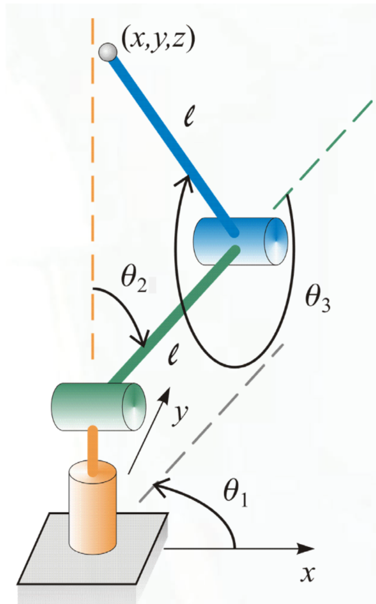

Figure 2: Singularity example.

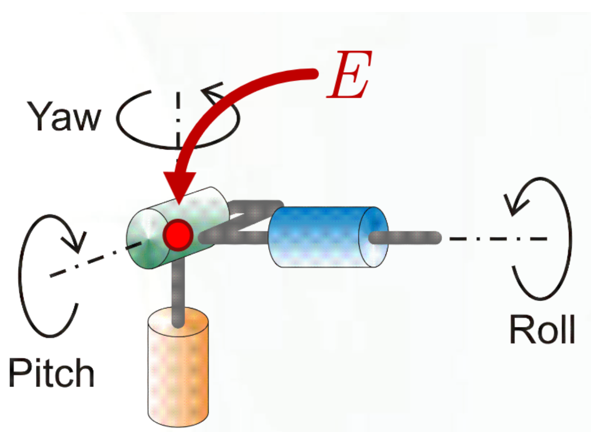

Figure 3: A wrist-partitioned arm.