Matthew Tran
EE106A
MT1

- Rigid Body Motion - preserves distance between points
  - $\|p(t)-q(t)\| = \|p(0)-q(0)\|$ distance
  - $g: O \to \mathbb{R}^3$ - mapping ; $g_*(v) = g(q) - g(p)$
  - $g_*(v \times w) = g_*(v) \times g_*(w)$ cross product
  - we use right-handed frames, $z = x \times y$
- Rotational Motion
  - R_{AB} - axes of B written in terms of A
    - $= [x_{ab} \; y_{ab} \; z_{ab}]$ column vectors
    - orthonormal $R \cdot R^T = R^T R = I$
    - $\det(R) = 1$ // right handed
  - $SO(n) = \{R \in \mathbb{R}^{n \times n} : RR^T = I, \det R = +1\}$
    - group - closure, identity, inverse, associativity
  - R_{AB} - also rotates point from frame B to frame A
  - $q_a = R_{AB} q_b$
  - $R_{AC} = R_{AB} R_{BC}$
  - $\hat{a} = (a)^\wedge = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}$ cross product $a \times b = (a)^\wedge b$, $\hat{\omega}^T = -\hat{\omega}$
  - $R(\omega)^\wedge R^T = (R\omega)^\wedge$
  - $R(\omega, \theta) = e^{\hat{\omega}\theta}$ // rotate about ω by θ (unit)
    - $so(n) = \{S \in \mathbb{R}^{n \times n} : S^T = -S\}$
      - $\hat{a}^2 = aa^T - \|a\|^2 I$
      - $\hat{a}^3 = -\|a\|^2 \hat{a}$
  - Rodrigues' formula
    $e^{\hat{\omega}\theta} = I + \frac{\hat{\omega}}{\|\omega\|}\sin(\|\omega\|\theta) + \frac{\hat{\omega}^2}{\|\omega\|^2}(1-\cos(\|\omega\|\theta))$
  - Euler Angles - (α, β, γ) = ZYZ, rotate about mobile axis, also find in the mobile frame
    - $R_{ba} = R_z(-\gamma)R_y(-\beta)R_z(-\alpha)$
  - ZYX Euler (yaw, pitch, roll) - fixed axis ("lead permutation")
  - 3 dimensional rep of rotation always has singularities (many soln.)
  - Quaternions - $q_0 + q_1 i + q_2 j + q_3 k$
    - $\theta = 2\cos^{-1}(q_0)$ $\omega = \begin{cases} \vec{q}/\sin(\theta/2) & \text{if } \theta \neq 0 \\ 0 \end{cases}$
- Rigid Motion
  - translation and rotation
  - p_{ab} - origin of B in A
  - $SE(3) = \{(p, R) : p \in \mathbb{R}^3, R \in SO(3)\} = \mathbb{R}^3 \times SO(3)$
  - $q_a = p_{ab} + R_{ab} q_b = g_{ab}(q_b)$
    - $g_*(v) = Rv$ // vectors
  - Homogeneous Representation
    - point $\begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ 1 \end{bmatrix}$ vector $\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ 0 \end{bmatrix}$
    - $\bar{q}_a = \begin{bmatrix} q_a \\ 1 \end{bmatrix} = \begin{bmatrix} R_{ab} & p_{ab} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q_b \\ 1 \end{bmatrix} = \bar{g}_{ab} \bar{q}_b$
    - $\bar{g}_{ac} = \bar{g}_{ab} \bar{g}_{bc}$
    - $\bar{g}^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix}$
  - $\hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$ // rotate about ω w/ pt. q
    twist
    - $v = -\omega \times q$
    - $se(3) = \{(v, \hat{\omega}) : v \in \mathbb{R}^3, \hat{\omega} \in so(3)\}$
  - $\begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}^\vee = \begin{bmatrix} v \\ \omega \end{bmatrix}$
    $\begin{bmatrix} v \\ \omega \end{bmatrix}^\wedge = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix}$
  - $e^{\hat{\xi}\theta} \in SE(3)$ proof by cases
  - all $SE(3)$ have $\xi(3)$ proof by cases
  - Screw
    - rotate about axis, then translate
    - θ: angle, translate $h\theta$
    - h: pitch = ∞ if no rotation
    - ℓ: line $\{q + \lambda\omega : \lambda \in \mathbb{R}\}$
    - screw to twist, use unit twists
      - $h = \infty$, $\|\omega\| = 1$, or $\omega = 0 \wedge \|v\| = 1$
        $\hat{\xi} = \begin{bmatrix} 0 & v \\ 0 & 0 \end{bmatrix}$
      - h = finite, $\|\omega\| = 1$
        $\hat{\xi} = \begin{bmatrix} \hat{\omega} & -\omega \times q + h\omega \\ 0 & 0 \end{bmatrix}$

$e^{\hat{\xi}\theta} = \begin{bmatrix} e^{\hat{\omega}\theta} & (I - e^{\hat{\omega}\theta})(\omega \times v) + \omega\omega^T v\theta \\ 0 & 1 \end{bmatrix}$

$g e^{\hat{\xi}\theta} g^{-1} = e^{(Ad_g \xi)^\wedge \theta}$ $Ad_g \xi$ is new axis of ξ after transform by g

- Kinematics
  - Forward Kinematics
    - joint space Q - all possible angles
      - S^1 - angles for revolute
      - $\mathbb{R}$ - for prismatic
    - base frame S
    - tool frame T
    - Product of Exponentials
      - get twist at $\theta = 0$
      - order doesn't matter? depends : think
      - $g_{st}(\theta) = e^{\hat{\xi}_1\theta_1} \cdots e^{\hat{\xi}_n\theta_n}$
      - choice of base frame can simplify things
    - Denavit-Hartenberg Parameters
      - Each joint 4 scalars $\alpha_i, a_i, d_i, \phi_i$
        revolute $\theta_i = \phi_i$ revolute
        prismatic $\theta_i = d_i$ prismatic
      - pick ϵ' cancellation occur
      - twist are relative to previous link
    - Manipulator Workspace
      - reachable workspace $W_R = \{p(\theta) : \theta \in Q\} \subset \mathbb{R}^3$
      - dextrous workspace $W_D = \{p \in \mathbb{R}^3 : \forall R \in SO(3), \exists\theta \text{ w/ } g_{st}(\theta) = (p, R)\} \subset \mathbb{R}^3$
        - not easy to calculate
        - have spherical wrist (3 int. joints) so $W_D = W_R$
  - Inverse Kinematics
    - desired config $g_d = g_{st}(\theta)$, θ?
    - Paden-Kahan subproblems
      - Subproblem 1 - Rotation about 1 axis
        - $e^{\hat{\xi}\theta} p = q$
        - 0 soln, 1 soln, or infinite (if p,q on axis)
      - Subproblem 2 - Rotation about 2 axes
        - $e^{\hat{\xi}_1\theta_1} e^{\hat{\xi}_2\theta_2} p = q$
        - 0, 1, 2 soln
      - Subproblem 3 Rotation to a given distance
        - $\|q - e^{\hat{\xi}\theta} p\| = \delta$
        - 0, 1, 2 soln
    - Tips
      - pick point on axis to remove stuff
      - post multiply by p to do so
      - also subtract point from both sides and take norm
      - careful to not remove things if want them
  - Extra
    - $e^A = \sum_{n=0}^{\infty} \frac{A^n}{n!} = I + A + \frac{1}{2!}A^2 + \cdots$
      $(e^A)^T = e^{A^T}$
      $e^{gAg^{-1}} = g e^A g^{-1}$
      e^{λ_i} also eigenvalue of e^A
      $(e^A)^{-1} = e^{-A}$
      $\det(e^A) = e^{\operatorname{tr} A}$
  - ROS
    - node - executable that runs
    - topics - can publish messages to / read from
    - services - request node to do stuff

Matthew Tran
EE106A
MT2

- Computer Vision
  - Pinhole Camera Model
    - 3D → 2D

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \rightarrow \begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix}$$

$$\lambda \begin{bmatrix} x_{im} \\ y_{im} \\ 1 \end{bmatrix} = \begin{bmatrix} fs_x & fs_\theta & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} I_3 & 0 \end{bmatrix} \begin{bmatrix} R & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix}$$

scale to $d_x X$ — pixel — intrinsic matrix — camera location — 3D

    - also $\lambda x = K\bar{X}$ ← in camera frame (X,Y,Z)
      Z — $(x,y,1)$ — always invertible, nonzero determinant
    - typically "all normalize" to $\lambda x = \bar{X}$ (mult by K^{-1})

  - Correspondences - 2 cameras
    - lots of assumptions about materials
    - Epipolar constraint - applies to pts + transforms
    - assume one camera is world frame $X_2 = RX_1 + T$
    - $x_2^T \hat{T} R x_1 = 0$
    - $\hat{T}R$ - essential matrix
      - has SVD $E = U\Sigma V^T$ w/ $\Sigma = diag(6,6,0)$
    - Computing essential matrix E
      - 8 point linear algorithm
        - write out $x_2^T E x_1 = 0 \rightarrow Ae = 0$
        - need rank 8 to solve
        - if noisy, min($\|Ae\|^2$), A^TA smallest λ
        - E also needs to be in essential space
          - use SVD $F = U\,diag(\lambda_1, \lambda_2, \lambda_3)V^T$
            get $E = U\,diag(6,6,0)\,V^T$
            $6 = (\lambda_1 + \lambda_2)/2$
            minimize $\|E - F\|_F^2$ Frobenius
        - can only get E up to scale factor
          - normalize T to unit vector

$$R = UR_z^T(\pm\tfrac{\pi}{2})V^T$$

$$\hat{T} = UR_z(\pm\tfrac{\pi}{2})\Sigma U^T$$

        - sanity check depths are positive
        - pts can't be in same plane

- Rigid Body Velocities
  - Spatial and body frames A & B
  - Spatial Velocity

$$v_{q_a}(t) = \dot{q}_a(t) = \dot{g}_{ab}(t) q_b = \dot{g}_{ab}(t) g_{ab}^{-1}(t) q_a = \hat{V}_{ab}^s q_a$$

$$\hat{V}_{ab}^s = \dot{g}_{ab} g_{ab}^{-1} = \begin{bmatrix} \dot{R}_{ab} R_{ab}^T & -\dot{R}_{ab} R_{ab}^T p_{ab} + \dot{p}_{ab} \\ 0 & 0 \end{bmatrix}$$

  - Body Velocity - not in body frame, relative to A viewed from B

$$v_{q_b}(t) = g_{ab}^{-1}(t)\dot{g}_{ab}(t) q_b = \hat{V}_{ab}^b q_b$$

$$= g_{ab}^{-1}(t) v_{q_a}(t)$$

$$\hat{V}_{ab}^b = g_{ab}^{-1}\dot{g}_{ab} = \begin{bmatrix} R_{ab}^T \dot{R}_{ab} & R_{ab}^T \dot{p}_{ab} \\ 0 & 0 \end{bmatrix}$$

  - $\hat{V}_{ab}^s = g_{ab} \hat{V}_{ab}^b g_{ab}^{-1}$

$$V_{ab}^s = Ad_{g_{ab}} V_{ab}^b \qquad Ad_g = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix}$$

$$Ad_g^{-1} = Ad_{g^{-1}} = \begin{bmatrix} R^T & -R^T\hat{p} \\ 0 & R^T \end{bmatrix}$$

  - $V_{ac}^s = V_{ab}^s + Ad_{g_{ab}} V_{bc}^s$ composition

$$V_{ac}^b = Ad_{g_{bc}^{-1}} V_{ab}^b + V_{bc}^b$$

  - can interpret as screws too $\begin{bmatrix} v \\ \omega \end{bmatrix}$ angular + linear components

- Manipulator Jacobians
  - Spatial Jacobian
    - $V_{ab}^s = J_{ab}^s(\theta)\dot{\theta}$

$$J_{st}^s(\theta) = \left[\left(\frac{\partial g_{st}}{\partial \theta_1}\right)^\vee \cdots \left(\frac{\partial g_{st}}{\partial \theta_n}\right)^\vee\right]$$

$$= \begin{bmatrix} \xi_1 & \xi_2' & \cdots & \xi_n' \end{bmatrix}$$

$$\xi_i' = Ad_{(e^{\hat{\xi}_1\theta_1} \cdots e^{\hat{\xi}_{i-1}\theta_{i-1}})}\xi_i$$

      ξ basically transforms from reference frame

can do by inspection by noting pt axis in spatial frame
revolute $\begin{bmatrix} -\omega \times q \\ \omega \end{bmatrix}$
prismatic $\begin{bmatrix} v \\ 0 \end{bmatrix}$

  - Body Jacobian
    - $V_{ab}^b = J_{ab}^b(\theta)\dot{\theta}$

$$J_{ab}^b(\theta) = \begin{bmatrix} \xi_1^\dagger & \cdots & \xi_n^\dagger \end{bmatrix}$$

$$\xi_i^\dagger = Ad^{-1}_{(e^{\hat{\xi}_{i+1}\theta_{i+1}} \cdots e^{\hat{\xi}_n\theta_n} g_{st}(0))}\xi_i$$

can also by inspection but in body frame

  - $J_{st}^s(\theta) = Ad_{g_{st}(\theta)} J_{st}^b(\theta)$
  - Singularities
    - when Jacobian drops rank (usually when not full rank) — commutes & collapse
    - when Jacobian drops rank, can't achieve movement in all directions, or force
      - not invertible, can't achieve movement in all directions
    - near singular == need large velocities == undesirable
    - use determinant $\det(A) \neq 0$ if invertible, $rank(A^TA) = rank(A) \leq \min(r,c)$

- Lagrangian Dynamics
  - Pick parameterization q (usually θ)
  - T = kinetic energy
    $\frac{1}{2}mv^2 + \frac{1}{2}I\dot{\theta}^2$ are helpful
    V = potential energy
    mgh, $\frac{1}{2}kx^2$ are helpful

$$L = T - V \text{ (Lagrangian)}$$

$$\Upsilon = \frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q}$$

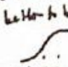
    - treat $\dot{q}$ and q as diff variables
    - Υ are joint forces / torques

$$\Upsilon = M(\theta)\ddot{\theta} + C(\theta,\dot{\theta})\dot{\theta} + G(\theta)$$

manipulator inertia — Coriolis — gravity vector

- Controls
  - desired trajectory $\theta_d(t)$
  - open loop $\tau = M\ddot{\theta}_d + C\dot{\theta}_d + N$
    - but nothing perfect
    - $\tau = M\ddot{\theta}_d + C\dot{\theta} + N$ (servo)
  - closed loop, hence accept (PD controller)
    $\tau = M(\ddot{\theta}_d - K_v\dot{e} - K_pe) + C\dot{\theta} + N$
    $e = \theta - \theta_d$
    $\ddot{e} + K_v\dot{e} + K_pe = 0$ — make to reach out K_v, K_p
  - Feedforward + feedback
    PD part $\tau = -K_v\dot{e} - K_pe$
  - exponential tracking

- path planning
  - w/ obstacles → navigate configuration space
  - cell decomposition - straight + math model
    - refine to avoid obstacles
    - gives via points
    - interpolate to smooth out
    - or approximate (execute)
  - increase interpolation order allows another
    cnt more controllable via (like v, a)
    1 - θ_0, θ_f, order
    2 - $\theta_0, \theta_f, v_{max}$
    3 - $\theta_0, \theta_f, v_0, v_f$
    5 - $\theta_0, \theta_f, v_0, v_f, a_0, a_f$
  - better to blend methods to go faster (LFPB)



poly via pts

  - minimum time trajectory (bang bang)
  - A* path search
    - $c(n) = g(n) + h(n)$
      previous — heuristic to goal
    - but what's around points
      - add extra to obstacles? but obstacles
  - Probabilistic Road Map (PRM)
    - random pnts, remove collisions
    - connect all edges, remove coll lines
    - use A*
  - Rapidly-Exploring Random Tree (RRT)
    - biased towards unexplored (Voronoi)
    - suboptimal af
    - basically keeps searching outward
  - RRT* - rewire nodes, asymptotically optimal
  - other RRT variants

sampling based planners

    - Shooting - like a cannon shot see if hit
      - single vs multiple
    - Direct Transcription ZOH, a
    - Direct Collocation - polynomial/spline

optimization planners (Sub)

  - Labs
    - bag - completed topic
    - point cloud - cloud of points
    - image segmentation
      - thresholding
      - edge detection
      - clustering
    - can synchronize topics, "approximate time"
    - Feature detection
      - descriptor - BRISK
      - nearest neighbor
      - hamming - # diff bits XOR
  - occupancy grid - how SLAM works
    - shooting probes
  - ochoxener - learning on blocking

- launch file
- parameter server