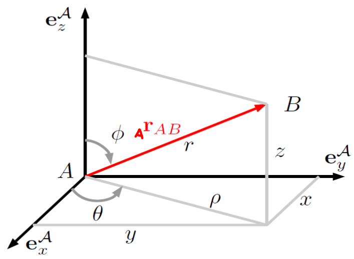
Robot Dynamics

Position

Position: vector ${}_A r_{AB} = (x,y,z)$ in frame A (cartesian) $\longrightarrow$



- **position representations** χ_P
  - cartesian: $\chi_{P_c} = (x,y,z) = (x,y,z) \Leftrightarrow {}_A r_{AB} = (x,y,z)$
  - cylindrical: $\chi_{P_z} = (\rho,\theta,z) = (\sqrt{x^2+y^2}, \operatorname{atan}(y/x), z) \Leftrightarrow {}_A r_{AB} = (r\cdot\cos\theta, r\cdot\sin\theta, z)$
  - spherical: $\chi_{P_s} = (r,\theta,\phi) = (\sqrt{x^2+y^2+z^2}, \operatorname{atan}(y/x), \operatorname{acos}(z/\sqrt{x^2+y^2+z^2})) \Leftrightarrow {}_A r_{AB} = (r\cdot\cos\theta\sin\phi, r\cdot\sin\theta\cdot\sin\phi, r\cdot\cos\phi)$

- **velocity representation** $\dot\chi_P \Leftrightarrow$ velocity vector ${}_A\dot r_{AB}$ ($= v_B$ if A is inertial frame)
  - cartesian: $\dot\chi_{P_c} = (\dot x, \dot y, \dot z)$
  - cylindrical: $\dot\chi_{P_z} = (\dot\rho, \dot\theta, \dot z)$
  - spherical: $\dot\chi_{P_s} = (\dot r, \dot\theta, \dot\phi)$

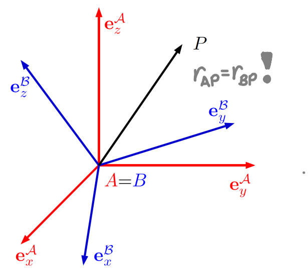
$$\Rightarrow \quad {}_A\dot r_{AB} = E_P(\chi_P)\cdot\dot\chi_P$$
$$\Leftarrow \quad \dot\chi_P = E_P^{-1}(\chi_P)\cdot {}_A\dot r_{AB}$$
$$\left(E_{P[\cdot]}(\chi_{P[\cdot]}) = \frac{\partial r(\chi_{P[\cdot]})}{\partial \chi_{P[\cdot]}}\right)$$

$$E_{P_c}(\chi_{P_c}) = \mathbb{I} \qquad E_{P_z}(\chi_{P_z}) = \begin{bmatrix} \cos\theta & -\rho\sin\theta & 0 \\ \sin\theta & \rho\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_{P_s}(\chi_{P_s}) = \begin{bmatrix} \cos\theta\sin\phi & \sin\phi\sin\theta & \cos\phi \\ -\sin\theta/(r\sin\phi) & \cos\theta/(r\sin\phi) & 0 \\ (\cos\phi\cos\theta)/r & (\cos\phi\sin\theta)/r & -\sin\phi/r \end{bmatrix}$$

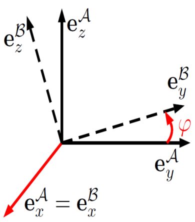
$$E_{P_c}^{-1}(\chi_{P_c}) = \mathbb{I} \qquad E_{P_z}^{-1}(\chi_{P_z}) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta/\rho & \cos\theta/\rho & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_{P_s}^{-1}(\chi_{P_s}) = \begin{bmatrix} \cos\theta\sin\phi & -r\sin\phi\sin\theta & r\cos\phi\cos\theta \\ \sin\phi\sin\theta & r\cos\theta\sin\phi & r\cos\phi\sin\theta \\ \cos\phi & 0 & -r\sin\phi \end{bmatrix}$$

Rotation

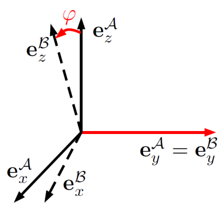
Rotation: matrix $C_{AB} \in SO(3)$ rotating from frame B to A



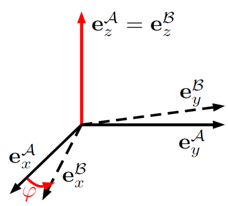
$${}_A r_{AP} = \underbrace{\begin{pmatrix} \top & \top & \top \\ {}_A e_x^B & {}_A e_y^B & {}_A e_z^B \\ \bot & \bot & \bot \end{pmatrix}}_{= C_{AB} = C_{BA}^{-1} = C_{BA}^T} \cdot {}_B r_{AP} \qquad / \qquad {}_B r_{AP} = \underbrace{\begin{pmatrix} \top & \top & \top \\ {}_B e_x^A & {}_B e_y^A & {}_B e_z^A \\ \bot & \bot & \bot \end{pmatrix}}_{= C_{BA} = C_{AB}^{-1} = C_{AB}^T} \cdot {}_A r_{AP}$$



$$C_{AB} = C_x(\varphi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi & \cos\varphi \end{pmatrix}$$



$$C_{AB} = C_y(\varphi) = \begin{pmatrix} \cos\varphi & 0 & \sin\varphi \\ 0 & 1 & 0 \\ -\sin\varphi & 0 & \cos\varphi \end{pmatrix}$$



$$C_{AB} = C_z(\varphi) = \begin{pmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- **rotation representations** χ_R $\quad \downarrow c_{[\cdot][\cdot]}$: entries in C_{AB} $\quad \downarrow c_{[\cdot]}, s_{[\cdot]}$: cos, sin of $[\cdot]$

  - ZYZ Euler angles: (yaw-pitch-yaw)

$$\chi_{R\,eulZYZ} = \begin{pmatrix} z_1 \\ y \\ z_2 \end{pmatrix} = \begin{pmatrix} atan2(c_{23}, c_{13}) \\ atan2\left(\sqrt{c_{13}^2 + c_{23}^2}, c_{33}\right) \\ atan2(c_{32}, -c_{31}) \end{pmatrix} \qquad C_{AB} = \begin{bmatrix} c_y c_{z_1} c_{z_2} - s_{z_1} s_{z_2} & -c_{z_2} s_{z_1} - c_y c_{z_1} s_{z_2} & c_{z_1} s_y \\ c_{z_1} s_{z_2} + c_y c_{z_2} s_{z_1} & c_{z_1} c_{z_2} - c_y s_{z_1} s_{z_2} & s_y s_{z_1} \\ -c_{z_2} s_y & s_y s_{z_2} & c_y \end{bmatrix}$$

  - ZXZ Euler angles: (yaw-roll-yaw)

$$\chi_{R\,eulZXZ} = \begin{pmatrix} z_1 \\ x \\ z_2 \end{pmatrix} = \begin{pmatrix} atan2(c_{13}, -c_{23}) \\ atan2\left(\sqrt{c_{13}^2 + c_{23}^2}, c_{33}\right) \\ atan2(c_{31}, c_{32}) \end{pmatrix} \qquad C_{AB} = \begin{bmatrix} c_{z_1} c_{z_2} - c_x s_{z_1} s_{z_2} & -c_{z_1} s_{z_2} - c_x c_{z_2} s_{z_1} & s_x s_{z_1} \\ c_{z_2} s_{z_1} + c_x c_{z_1} s_{z_2} & c_x c_{z_1} c_{z_2} - s_{z_1} s_{z_2} & -c_{z_1} s_x \\ s_x s_{z_2} & c_{z_2} s_x & c_x \end{bmatrix}$$

  - ZYX Euler angles: (yaw-pitch-roll)

$$\chi_{R\,eulZYX} = \begin{pmatrix} z \\ y \\ x \end{pmatrix} = \begin{pmatrix} atan2(c_{21}, c_{11}) \\ atan2\left(-c_{31}, \sqrt{c_{32}^2 + c_{33}^2}\right) \\ atan2(c_{32}, c_{33}) \end{pmatrix} \qquad C_{AB} = \begin{bmatrix} c_y c_z & c_z s_x s_y - c_x s_z & s_x s_z + c_x c_z s_y \\ c_y s_z & c_x c_z + s_x s_y s_z & c_x s_y s_z - c_z s_x \\ -s_y & c_y s_x & c_x c_y \end{bmatrix}$$

- XYZ Euler angles: (roll-pitch-yaw)

$$\chi_{R\,eulXYZ} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} atan2(-c_{23}, c_{33}) \\ atan2\left(c_{13}, \sqrt{c_{11}^2 + c_{12}^2}\right) \\ atan2(-c_{12}, c_{11}) \end{pmatrix} \qquad C_{AB} = \begin{bmatrix} c_y c_z & -c_y s_z & s_y \\ c_x s_z + c_z s_x s_y & c_x c_z - s_x s_y s_z & -c_y s_x \\ s_x s_z - c_x c_z s_y & c_z s_x + c_x s_y s_z & c_x c_y \end{bmatrix}$$

- angle axis: (axis $\underline{n}$ + angle θ)

$$\chi_{R\,aa} = \begin{pmatrix} \theta \\ n_x \\ n_y \\ n_z \end{pmatrix} = \begin{matrix} \cos^{-1}\left(\frac{c_{11}+c_{22}+c_{33}-1}{2}\right) \\ \frac{1}{2\sin(\theta)} \begin{pmatrix} c_{32}-c_{23} \\ c_{13}-c_{31} \\ c_{21}-c_{12} \end{pmatrix} \end{matrix}$$

$$C_{AB} = \begin{bmatrix} n_x^2(1-c_\theta)+c_\theta & n_x n_y(1-c_\theta) - n_z s_\theta & n_x n_z(1-c_\theta) + n_y s_\theta \\ n_x n_y(1-c_\theta) + n_z s_\theta & n_y^2(1-c_\theta)+c_\theta & n_y n_z(1-c_\theta) - n_x s_\theta \\ n_x n_z(1-c_\theta) - n_y s_\theta & n_y n_z(1-c_\theta) + n_x s_\theta & n_z^2(1-c_\theta)+c_\theta \end{bmatrix}$$

- rotation vector: $(\underline{\varphi} = \underline{n} \cdot \theta)$

$$\chi_{R\,rv} = \begin{pmatrix} \varphi_x \\ \varphi_y \\ \varphi_z \end{pmatrix} = (\text{same as angle axis with } \underline{\varphi} = \underline{n} \cdot \theta) \qquad C_{AB} =$$

- unit quaternions: from angle axis $\underline{n}, \theta$: $\xi_0 = \cos\theta/2$ $\quad \underline{\xi} = \sin\theta/2 \cdot \underline{n}$ $\leftrightarrow$ $\boxed{\xi_0^2 + \xi_1^2 + \xi_2^2 + \xi_3^2 = 1}$

$$\chi_{R,quat} = \begin{pmatrix} \xi_0 \\ \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \sqrt{c_{11}+c_{22}+c_{33}+1} \\ sgn(c_{32}-c_{23})\sqrt{c_{11}-c_{22}-c_{33}+1} \\ sgn(c_{13}-c_{31})\sqrt{c_{22}-c_{33}-c_{11}+1} \\ sgn(c_{21}-c_{12})\sqrt{c_{33}-c_{11}-c_{22}+1} \end{pmatrix} \qquad C_{AB} = \begin{bmatrix} \xi_0^2+\xi_1^2-\xi_2^2-\xi_3^2 & 2\xi_1\xi_2 - 2\xi_0\xi_3 & 2\xi_0\xi_2 + 2\xi_1\xi_3 \\ 2\xi_0\xi_3 + 2\xi_1\xi_2 & \xi_0^2-\xi_1^2+\xi_2^2-\xi_3^2 & 2\xi_2\xi_3 - 2\xi_0\xi_1 \\ 2\xi_1\xi_3 - 2\xi_0\xi_2 & 2\xi_0\xi_1 + 2\xi_2\xi_3 & \xi_0^2-\xi_1^2-\xi_2^2+\xi_3^2 \end{bmatrix}$$

↳ inverse: $C_{AB} \xrightarrow{inv} C_{BA}$ $\Rightarrow$ $\xi = \begin{pmatrix} \xi_0 \\ \underline{\xi} \end{pmatrix} \xrightarrow{inv} \begin{pmatrix} \xi_0 \\ -\underline{\xi} \end{pmatrix} = \xi^T = \xi^{-1}$

↳ multiply: $C_{AC} = C_{AB} \cdot C_{BC}$ $\Rightarrow$ $\xi_{AC} = \xi_{AB} \otimes \xi_{BC} =$

$$\begin{bmatrix} \xi_0 & -\xi_1 & -\xi_2 & -\xi_3 \\ \xi_1 & \xi_0 & -\xi_3 & \xi_2 \\ \xi_2 & \xi_3 & \xi_0 & -\xi_1 \\ \xi_3 & -\xi_2 & \xi_1 & \xi_0 \end{bmatrix}_{AB} \begin{pmatrix} \xi_0 \\ \xi_1 \\ \xi_2 \\ \xi_3 \end{pmatrix}_{BC}$$

↳ apply: ${}_A\vec{r} = C_{AB} \cdot {}_B\vec{r}$ $\Rightarrow$ $\begin{pmatrix} 0 \\ {}_A\vec{r} \end{pmatrix} = \xi_{AB} \otimes \begin{pmatrix} 0 \\ {}_B\vec{r} \end{pmatrix} \otimes \xi_{AB}^T$

angular velocity ${}_A\omega_{AB}$

element wise derivative

$${}_A\omega_{AB} = \begin{pmatrix} \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \leftarrow [{}_A\omega_{AB}]_\times = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} = \dot{C}_{AB} \cdot C_{AB}^T$$

(angular velocity vector at which frame B rotates relative to frame A)

${}_D\omega_{AC} = {}_D\omega_{AB} + {}_D\omega_{BC}$ / ${}_D\omega_{AB} = -{}_D\omega_{BA}$ / ${}_B\omega_{AB} = C_{BA}\,{}_A\omega_{AB}$ / $[{}_B\omega_{AB}]_\times = C_{BA} \cdot [{}_A\omega_{AB}]_\times \cdot C_{AB}$

angular velocity representations $\dot{\chi}_R$ ⇔ angular velocity ${}_A\omega_{AB}$

↳ ${}_A\omega_{AB} = E_R(\chi_R) \cdot \dot{\chi}_R$; $\dot{\chi}_R = E_R^{-1}(\chi_R) \cdot {}_A\omega_{AB}$

$$E_{Reul\,ZYZ} = \begin{bmatrix} 0 & -\sin(z_1) & \cos(z_1)\sin(y) \\ 0 & \cos(z_1) & \sin(z_1)\sin(y) \\ 1 & 0 & \cos(y) \end{bmatrix} \qquad E^{-1}_{Reul\,ZYZ} = \begin{bmatrix} -\frac{\cos(y)\cos(z_1)}{\sin(y)} & -\frac{\cos(y)\sin(z_1)}{\sin(y)} & 1 \\ -\sin(z_1) & \cos(z_1) & 0 \\ \frac{\cos(z_1)}{\sin(y)} & \frac{\sin(z_1)}{\sin(y)} & 0 \end{bmatrix}$$

$$E_{Reul\,ZXZ} = \begin{bmatrix} 0 & \cos(z_1) & \sin(z_1)\sin(x) \\ 0 & \sin(z_1) & -\cos(z_1)\sin(x) \\ 1 & 0 & \cos(x) \end{bmatrix} \qquad E^{-1}_{Reul\,ZXZ} = \begin{bmatrix} -\frac{\cos(x)\sin(z_1)}{\sin(x)} & \frac{\cos(x)\cos(z_1)}{\sin(x)} & 1 \\ \cos(z_1) & \sin(z_1) & 0 \\ \frac{\sin(z_1)}{\sin(x)} & -\frac{\cos(z_1)}{\sin(x)} & 0 \end{bmatrix}$$

$$E_{Reul\,ZYX} = \begin{bmatrix} 0 & -\sin(z) & \cos(y)\cos(z) \\ 0 & \cos(z) & \cos(y)\sin(z) \\ 1 & 0 & -\sin(y) \end{bmatrix} \qquad E^{-1}_{Reul\,ZYX} = \begin{bmatrix} \frac{\cos(z)\sin(y)}{\cos(y)} & \frac{\sin(y)\sin(z)}{\cos(y)} & 1 \\ -\sin(z) & \cos(z) & 0 \\ \frac{\cos(z)}{\cos(y)} & \frac{\sin(z)}{\cos(y)} & 0 \end{bmatrix}$$

$$E_{Reul\,XYZ} = \begin{bmatrix} 1 & 0 & \sin(y) \\ 0 & \cos(x) & -\cos(y)\sin(x) \\ 0 & \sin(x) & \cos(x)\cos(y) \end{bmatrix} \qquad E^{-1}_{Reul\,XYZ} = \begin{bmatrix} 1 & \frac{\sin(x)\sin(y)}{\cos(y)} & -\frac{\cos(x)\sin(y)}{\cos(y)} \\ 0 & \cos(x) & \sin(x) \\ 0 & -\frac{\sin(x)}{\cos(y)} & \frac{\cos(x)}{\cos(y)} \end{bmatrix}$$

$$E_{R\,quat} = 2 \cdot \begin{bmatrix} -\xi_1 & \xi_0 & -\xi_3 & \xi_2 \\ -\xi_2 & \xi_3 & \xi_0 & -\xi_1 \\ -\xi_3 & -\xi_2 & \xi_1 & \xi_0 \end{bmatrix} \qquad E^{-1}_{R\,quat} = \frac{1}{2} \cdot \begin{bmatrix} -\xi_1 & \xi_0 & -\xi_3 & \xi_2 \\ -\xi_2 & \xi_3 & \xi_0 & -\xi_1 \\ -\xi_3 & -\xi_2 & \xi_1 & \xi_0 \end{bmatrix}^T$$

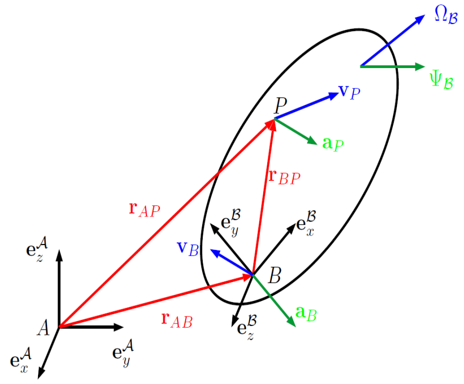
$$E_{R\,aa} = \begin{bmatrix} \mathrm{n} & \sin\theta \mathbb{I}_{3\times3} + (1-\cos\theta)[\mathrm{n}]_\times \end{bmatrix} \qquad E^{-1}_{R\,aa} = \begin{bmatrix} \mathrm{n}^T \\ -\frac{1}{2}\frac{\sin\theta}{1-\cos\theta}[\mathrm{n}]_\times^2 - \frac{1}{2}[\mathrm{n}]_\times \end{bmatrix}$$

Transformations (combine rotation and translation for new frame)

$$T_{AB} = \begin{pmatrix} C_{AB} & {}_A r_{AB} \\ 0\ 0\ 0 & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} {}_A r_{AB} \\ 1 \end{pmatrix} = T_{AB} \cdot \begin{pmatrix} {}_B r_{AB} \\ 1 \end{pmatrix} \quad ; \quad T_{BA} = \begin{pmatrix} C_{AB}^T & {}_B r_{BA} \\ 0\ 0\ 0 & 1 \end{pmatrix}$$

0: vector
1: point

$${}_B r_{BA} = -C_{AB}^T \cdot {}_A r_{AB}$$



Velocity in moving bodies: (rigid body theorem)

$${}_A\dot{r}_{AP} = {}_A\dot{r}_{AB} + {}_A\omega_{AB} \times {}_A r_{BP} \quad ; \quad {}_A\ddot{r}_{AP} = {}_A\ddot{r}_{AB} + {}_A\alpha_{AB} \times {}_A r_{BP} + {}_A\omega_{AB} \times ({}_A\omega_{AB} \times {}_A r_{BP})$$

↳ ${}_A\dot{r}_{AP} = {}_A v_P$ only if frame A not moving !! else ${}_A v_P = {}_A\dot{r}_{IP} + {}_A\omega_{IA} \times {}_A r_{IP}$

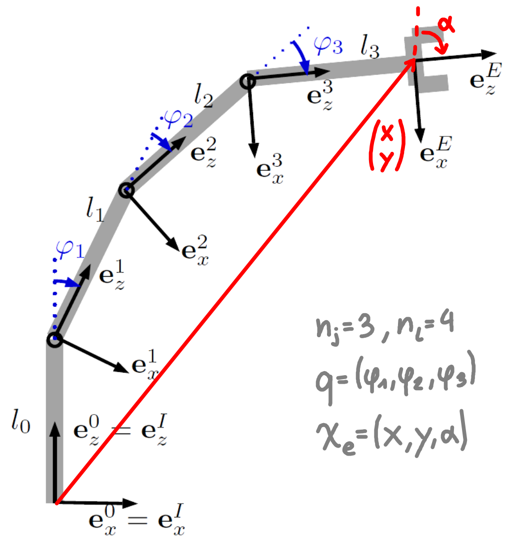
Classical Serial Kinematic Linkages

n_j: # of joints $\begin{cases} \bullet \text{ revolute (1DOF)} & \to q_i: \text{angle} \\ \bullet \text{ prismatic (slider) (1DOF)} & \to q_i: \text{linear dist.} \end{cases}$

$n_L = n_j + 1$: # of links $\begin{cases} \bullet\ n_j \text{ moving links} \\ \bullet\ 1 \text{ fixed link} \end{cases}$

joint space: $q = \begin{pmatrix} q_1 \\ \vdots \\ q_{n_j} \end{pmatrix} \in \mathbb{R}^{n_j}$

task space: $\chi_e = \begin{pmatrix} \text{one of } \chi_P \\ \text{one of } \chi_R \end{pmatrix} \in SE(3)$ {pos. + orient. of end-effector



$n_j = 3$, $n_L = 4$
$q = (\varphi_1, \varphi_2, \varphi_3)$
$\chi_e = (x, y, \alpha)$

$$T_{IE} = T_{I0} \cdot \left(T_{01}(q_1) \cdot T_{12}(q_2) \cdot \ldots \cdot T_{n_j n_j-1}(q_{n_j})\right) \cdot T_{n_j E} = \begin{pmatrix} C_{IE}(q) & {}_I r_{IE}(q) \\ 0_{1\times3} & 1 \end{pmatrix}$$

T_{I0}: const. transform from inertial frame to base of arm
$T_{n_j E}$: const. transform from last link to end-effector
$\chi_{eP}(q)$ from ${}_I r_{IE}(q)$, χ_{eR} from $C_{IE}(q)$: task space

$$\downarrow \quad \chi_e(q) = \begin{pmatrix} L_1\sin(q_1) + L_2\sin(q_1+q_2) + L_3\sin(q_1+q_2+q_3) \\ L_0 + L_1\cos(q_1) + L_2\cos(q_1+q_2) + L_3\cos(q_1+q_2+q_3) \\ q_1+q_2+q_3 \end{pmatrix}$$

- analytic jacobian: $J_{eA}(q, \chi_e) = \begin{pmatrix} \frac{\partial \chi_{e1}}{\partial q_1} & \cdots & \frac{\partial \chi_{e1}}{\partial q_{n_j}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \chi_{em}}{\partial q_1} & \cdots & \frac{\partial \chi_{em}}{\partial q_{n_j}} \end{pmatrix} \Rightarrow \dot{\chi}_e(q) = J_{eA}(q) \cdot \dot{q}$

  ↳ $\Delta\chi_e(q) \approx J_{eA}(q)\Delta q$

- geometric jacobian: ${}_I J_{e0}(q) = E_e(\chi_e) \cdot J_{eA}(q) = \begin{pmatrix} E_P & 0 \\ 0 & E_R \end{pmatrix} \cdot J_{eA}(q) \Rightarrow {}_I w_e = \begin{pmatrix} {}_I v_e \\ {}_I \omega_e \end{pmatrix} = {}_I J_{e0}(q) \cdot \dot{q}$

  $\chi_{de} \to J_{deA} \to J_{de0}$

  ↳ ${}_I w_e = {}_I w_d + {}_I w_{de} \leftrightarrow {}_I J_{e0} = {}_I J_{d0} + {}_I J_{de0}$

- geometric Jacobian from geometry (if q all angles):
  1) find local tra. matrices: $T_{I0}, T_{01}, T_{12}, \ldots, T_{(n_j-1)n_j}, T_{n_j E}$
  2) find local rot. axis: ${}_0 n_1, {}_1 n_2, \ldots, {}_{(n_j-1)} n_{n_j}$
  3) get full tra. matrices: $T_{I0}, T_{I1}, T_{I2}, \ldots, T_{I n_j}, T_{IE}$ ← $T_{IK} = T_{I0} \cdot \prod_{i=0}^{k} T_{(i-1)i}$
  4) get full rot. axis: ${}_I n_1, {}_I n_2, \ldots, {}_I n_{n_j}$ ← $({}_I n_k, 0) = C_{I(k-1)} \cdot ({}_{(k-1)} n_k, 0)$
  5) get pos. joint → end-effector: ${}_I r_{1E}, {}_I r_{2E}, \ldots, {}_I r_{n_j E}$ ← ${}_I r_{kE} = \underbrace{T_{IE} \cdot (0,0,0,1)}_{{}_I r_{IE}} - \underbrace{T_{IK} \cdot (0,0,0,1)}_{{}_I r_{IK}}$

  5) construct ${}_I J_{e0} = \begin{pmatrix} {}_I n_1 \times {}_I r_{1E} & \cdots & {}_I n_{n_j} \times {}_I r_{n_j E} \\ {}_I n_1 & \cdots & {}_I n_{n_j} \end{pmatrix} \begin{matrix} \}J_P \\ \}J_R \end{matrix}$

  $({}_I\omega_{01} = {}_I n_1 \cdot \dot{q}_1, {}_I\omega_{12} = {}_I n_2 \cdot \dot{q}_2, \ldots \quad {}_I\omega_{IE} = {}_I\omega_{I0} + {}_I\omega_{01} + \ldots + {}_I\omega_{n_j E})$

Inverse differential kinematics (find $\dot{q}$ so that w_e^* is reached)

$w_e = \begin{pmatrix} v_e \\ \omega_e \end{pmatrix} = \begin{pmatrix} J_{Pe0} \\ J_{Re0} \end{pmatrix} \cdot \dot{q} = J_{e0}(q) \cdot \dot{q} \Rightarrow \boxed{\dot{q} = J_{e0}^+(q) \cdot w_e^*}$ (J_{e0}^+: pseudo inverse ; w_e^*: desired ; w_e: actual)

- $n \geq m$: $J_{e0}^+ = J_{e0}^T (J_{e0} J_{e0}^T)^{-1}$ or damped $J_{e0}^+ = J_{e0}^T (J_{e0} J_{e0}^T + \lambda^2 \mathbb{I})^{-1}$
- $n \leq m$: $J_{e0}^+ = (J_{e0} J_{e0}^T)^{-1} J_{e0}^T$ or damped $\underbrace{J_{e0}^+ = (J_{e0} J_{e0}^T + \lambda^2 \mathbb{I})^{-1} J_{e0}^T}_{\text{good for case 3,4}}$

- $\dim(w_e) = \text{rank}(J_{e0}) = \dim(\dot{q})$: one on one mapping w_e to $\dot{q} \rightarrow w_e = w_e^*$
- $\dim(w_e) = \text{rank}(J_{e0}) < \dim(\dot{q})$: one w_e can have many $\dot{q}$ $\rightarrow w_e = w_e^*$; J_{e0}^+ minimizes $\|\dot{q}\|$
  - all other solutions: $\dot{q} = J_{e0}^+ w_e^* + N \dot{q}_0$ ($N = \mathbb{I} - J_{e0}^+ \cdot J_{e0}$: null-space of J_{e0}, $\dot{q}_0$: anything)
- $\dim(w_e) > \text{rank}(J_{e0}) = \dim(\dot{q})$: some w_e don't have any $\dot{q}$ $\rightarrow w_e \neq w_e^*$; J_{e0}^+ minimizes $\|w_e - w_e^*\|$
- $\dim(w_e) > \text{rank}(J_{e0}) < \dim(\dot{q})$: some DOF are redundant (e.g. same rot. axis)

multi-task, equal priority:

$\text{task}_i : \{w_i^*, J_i\}$ ($w_i = J_i(q) \cdot \dot{q}$) (like row of above $w_e = J_{e0} \cdot \dot{q}$, but with any $w_i = v_{(i)}$ or $\omega_{(i)}$ and corresponding J_i)

$$\underbrace{\dot{q}}_{\dot{q}} = \underbrace{\begin{pmatrix} - J_1 - \\ \vdots \\ - J_{n_t} - \end{pmatrix}^+}_{\bar{J}^+} \cdot \underbrace{\begin{pmatrix} w_1^* \\ \vdots \\ w_{n_t}^* \end{pmatrix}}_{\bar{w}^*} \Rightarrow \underbrace{\begin{pmatrix} w_1 \\ \vdots \\ w_{n_t} \end{pmatrix}}_{\bar{w}} = \underbrace{\begin{pmatrix} - J_1 - \\ \vdots \\ - J_{n_t} - \end{pmatrix}}_{\bar{J}} \cdot \dot{q}$$

n_t: # of tasks i
w_i^*: desired v, ω of task i
w_i: resulting v, ω of task i
J_i: geom jacobian to w_i

multi-task, weighted priority:

same as above, except: $\bar{J}^+ \rightarrow \bar{J}^{+W} = (\bar{J}^T W \bar{J})^{-1} \bar{J}^T W$ (W = diag matrix with weights of tasks) (weights = importance)

multi-task, ordered priority:

solve task 1 then task 2: $\dot{q} = \underbrace{J_1^+ w_1^*}_{\text{sol. to task 1}} + N_1 \cdot \underbrace{((J_2 N_1)^+ (w_2^* - J_2 J_1^+ w_1^*))}_{\dot{q}_0 \text{ that solves task 2}}$ ($N_k = \mathbb{I} - J_k^+ J_k$: nullspace of J_k)

solve tasks 1 to n_t in order: $\dot{q} = \sum_{i=1}^{n_t} \bar{N}_i \dot{q}_i$; $\dot{q}_i = (J_i \bar{N}_i)^+ (w_i^* - J_i \sum_{k=1}^{i-1} \bar{N}_k \dot{q}_k)$ ($\bar{N}_i$: nullspace of $\begin{pmatrix} - J_1 - \\ \vdots \\ - J_{i-1} - \end{pmatrix}$)

Inverse Kinematics (find q so that χ_e^* is reached)

$q \leftarrow q_0$ — q_0: start config.
while $\|\chi_e^* - \chi_e(q)\| \geq tol$ do — χ_e^*: target pos.
  $J_{eA} \leftarrow J_{eA}(q) = \partial \chi_e / \partial q (q)$ — J_{eA}: analytic jacobian
  $J_{eA}^+ \leftarrow (J_{eA}(q))^+$ — J_{eA}^+: pseudo inverse ← if J_{eA} missing ranks, maybe use damped pseudo inverse (see diff. inv. kin.) or $J_{eA}^+ \approx \alpha J_{eA}^T$?!
  $\Delta \chi_e \leftarrow \chi_e^* - \chi_e(q)$ — $\Delta \chi_e$: target error
  $q \leftarrow q + k \cdot J_{eA}^+ \Delta \chi_e$ — q: improved config. ← if $\Delta \chi_e$ large, maybe add scaling $0 < k < 1$

---

trajectory of orientation is affected by choice of χ_R for $\chi_e^* = \begin{pmatrix} \text{some } \chi_P \\ \text{some } \chi_R \end{pmatrix}$. for fastest path do:
- replace rotation error $\Delta \chi_{eR}$ in $\Delta \chi_e$ with $\Delta \varphi$:
  - find rot. matrix $C_{IE^*} = C(\chi_{eR}^*)$ and $C_{IE} = C(\chi_{eR}(q)) \rightarrow C_{EE^*} = C_{IE}^T \cdot C_{IE^*} \rightarrow \Delta \varphi$ from $\chi_{R,rv}$ of C_{EE^*}
- replace rot. part of J_{eA} with rot part of J_{e0} (since $\Delta \varphi / \Delta q \approx \omega$) (change k maybe)

Trajectory control (find $\dot{q}$ that tracks trajectory χ_e^*)

- **position trajectory control**

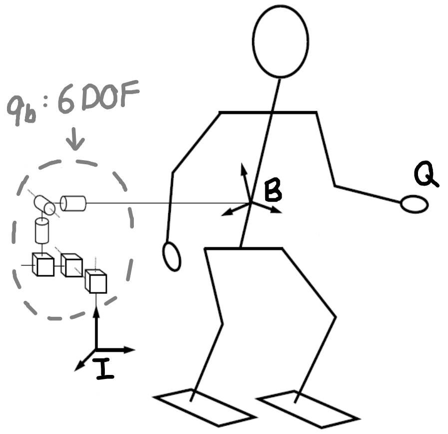
$\Delta r_e(t) = r_e^*(t) - r_e(q(t)) \to 0 \quad \Rightarrow \quad \dot{q} = J_{e0P}^+(q(t)) \cdot (\dot{r}_e^*(t) + k_{pP} \cdot \Delta r_e(t))$

- **orientation trajectory control**

$\Delta\varphi(t) = $ see chapter above $\to 0 \quad \Rightarrow \quad \dot{q} = J_{e0R}^+(q(t)) \cdot (\omega_e^*(t) + k_{pR}\,\Delta\varphi(t))$

combined: $\dot{q} = J_{e0}^+(q(t)) \cdot \begin{pmatrix} \dot{r}_e^*(t) + k_{pP} \cdot \Delta r_e(t) \\ \omega_e^*(t) + k_{pR}\,\Delta\varphi(t) \end{pmatrix}$

Floating base kinematics



joint space $q = \begin{pmatrix} q_b \\ q_j \end{pmatrix}$ ← virtual un-actuated base joints $= \begin{pmatrix} \chi_P \\ \chi_R \end{pmatrix}$; ← normal actuated joints

joint velocity, acceleration: $u = \begin{pmatrix} {}_I v_B \\ {}_B\omega_{IB} \\ \dot{q}_j \end{pmatrix} \neq \dot{q} \qquad \dot{u} = \begin{pmatrix} {}_I a_B \\ {}_B\psi_{IB} \\ \ddot{q}_j \end{pmatrix} \neq \ddot{q}$

↳ $u = \begin{pmatrix} E_P & \cdot & \cdot \\ \cdot & E_R & \cdot \\ \cdot & \cdot & \mathbb{I}_{n_j \times n_j} \end{pmatrix} \dot{q}$

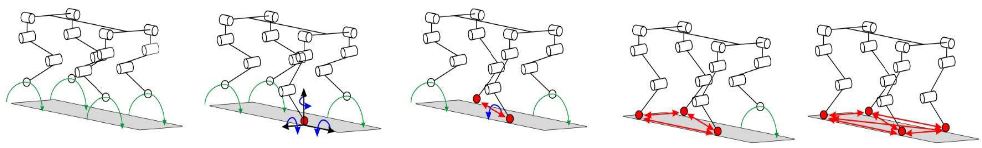
position : ${}_I r_{IQ}(q) = {}_I r_{IB}(q) + C_{IB}(q) \cdot {}_B r_{BQ}(q)$

velocity : ${}_I w_{IQ}(q) = \begin{pmatrix} {}_I v_Q \\ {}_I\omega_{IQ} \end{pmatrix} = \begin{bmatrix} \mathbb{I}_{3\times3} & -C_{IB} \cdot [{}_B r_{BQ}]_\times & C_{IB} \cdot {}_B J_{P_{q_j}}(q_j) \\ 0_{3\times3} & C_{IB} & C_{IB} \cdot {}_B J_{R_{q_j}}(q_j) \end{bmatrix} \cdot u = {}_I J_Q(q) \cdot u$

- contact constraint: ${}_I r_{IC_i} = $ const , ${}_I\dot{r}_{IC_i} = {}_I\ddot{r}_{IC_i} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ $\quad$ (C_i: $1 \ldots n_c$ contact points)

↳ ${}_I J_{C_i} \cdot u = 0$, ${}_I J_{C_i} \cdot \dot{u} + {}_I\dot{J}_{C_i} u = 0 \quad \to \quad J_c = \begin{pmatrix} J_{c1} \\ \vdots \\ J_{c n_c} \end{pmatrix} = (J_{cb} \;\; J_{cj})$ (contact jacobian)

- rank(J_c): # of independent constraints (3 per contact point)
- rank(J_{cq}): # of DOF to move base with actuators



total constraints	: rank(J_c)	0	3	6	9	12
base constraints	: rank(J_{cb})	0	3	5	6	6
internal constraints	: rank(J_c) − rank(J_{cb})	0	0	1	3	6
uncontrollable DoF	: 6 − rank(J_{cb})	6	3	1	0	0

- dynamics $\quad M(q) \cdot \dot{u} + b(q,u) + g(q) = \tau + J_c^T F_c \qquad \tau = \begin{pmatrix} 0 \\ \tau_{act} \end{pmatrix}$ $\}$ body frame B (unactuated!) $\}$ joint space (actuated!)

Dynamics

(find relation of input force/torque τ and output motion $\ddot{q}$)

EoM equation of motion: $M(q)\ddot{q} + b(q,\dot{q}) + g(q) = \tau + J_c^T \cdot F_c$

- $q, \dot{q}, \ddot{q}$: generalized coordinates
- $M(q)$: mass matrix
- $b(q,\dot{q})$: centrifugal, coriolis forces
- $g(q)$: gravity forces
- τ: generalized joint forces
- $J_c^T F_c$: ext. forces $J_{c0,P}^T \cdot F_c$, moments $J_{c0,R}^T \cdot M_c$

cornerstone: principle of virtual work (rigid body, r_s=CoG):

$\dot{p}_s = d/dt(m \cdot v_s) = m \cdot a_s = F_{ext}$ — r_s, v_s, a_s: linear pos., speed, acc.

$\dot{N}_s = d/dt(\Theta_s \Omega) = \Theta_s \Psi + \Omega \times \Theta_s \Omega = T_{ext}$ — Φ, Ω, Ψ: rotation pos., speed, acc.

method 1: Newton-Euler for single bodies:

1) cut all bodies free + add link forces
2) write 6 eq. for each body $\dot{p}_s = F_{ext}$, $\dot{N}_s = T_{ext}$ for x,y,z axis
3) eliminate all link forces
4) write $r_s, v_s, a_s, \Phi, \Omega, \Psi$ as functions of $q, \dot{q}, \ddot{q}$

method 2: Lagrange II:

Kinetic E.: $\mathcal{T} = \sum_{i=1}^{n_b} \left(\frac{1}{2} \dot{r}_{s_i}^T m_i \dot{r}_{s_i} + \frac{1}{2} \Omega_{s_i}^T \Theta_{s_i} \Omega_{s_i} \right) = \frac{1}{2} \dot{q}^T M(q) \cdot \dot{q}$

Potential E.: $\mathcal{U} = \sum_{i=1}^{n_b} \left(- r_{s_i}^T F_{g_i} \right)$ + springs etc..

Lagrangian: $\mathcal{L} = \mathcal{T} - \mathcal{U}$

↳ EoM: $\frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}}\right) - \left(\frac{\partial \mathcal{L}}{\partial q}\right) = \tau + J_c^T F_c$

method 3: projected Newton-Euler:

$$\begin{pmatrix} v_s \\ \Omega \end{pmatrix} = \begin{pmatrix} J_s \\ J_R \end{pmatrix} \cdot \dot{q}, \quad \begin{pmatrix} a_s \\ \Psi \end{pmatrix} = \begin{pmatrix} J_s \\ J_R \end{pmatrix} \ddot{q} + \begin{pmatrix} \dot{J}_s \\ \dot{J}_R \end{pmatrix} \dot{q} \qquad \begin{pmatrix} J_s \\ J_R \end{pmatrix} = {}_I J_{s0}$$

(m_i: mass of link i, ${}_i\Theta_{s_i}$: inertia mat. of link i in body frame)

$M(q) = \sum_{i=0}^{n_b} {}_I J_{S_i}^T m_i \, {}_I J_{S_i} + {}_I J_{R_i}^T \, {}_I \Theta_{S_i} \, {}_I J_{R_i}$ $\quad ({}_I\Theta_{S_i} = C_{Ii} \cdot {}_i\Theta_{S_i} \cdot C_{Ii}^T)$

$b(q,\dot{q}) = \sum_{i=0}^{n_b} {}_I J_{S_i}^T m_i \, {}_I \dot{J}_{S_i} \dot{q} + {}_I J_{R_i}^T \left({}_I\Theta_{S_i} \, {}_I\dot{J}_{R_i} \dot{q} + {}_I\Omega_{S_i} \times ({}_I\Theta_{S_i} \, {}_I\Omega_{S_i}) \right)$ $\quad ({}_I\Omega_{S_i} = {}_I J_{R_i} \cdot \dot{q})$

$g(q) = \sum_{i=0}^{n_b} - {}_I J_{S_i}^T \, {}_I F_{g_i}$ $\quad ({}_I F_{g_i} = m_i \cdot g \cdot {}_I e_g)$

$J_c^T \cdot F_c = \sum_{j=0}^{n_{f ext}} J_{P_j}^T F_{ext,j} + \sum_{k=0}^{n_{m ext}} J_{Rk}^T T_{ext,k}$ $\quad$ (J_{P_i} like J_{S_i} but for F_{ext} attack point instead of CoG)

Joint-space dynamic control

(find τ^* to track desired trajectory $q^*, \dot{q}^*$)

joint impedance regulation: (PD-control, ignoring system dynamics)

$\tau^* = k_p(q^* - q) + k_d(\dot{q}^* - \dot{q}) + g(q)$

τ^*: target torques at each joint; k_p: "spring" at joints; k_d: "dampener" at joints; $g(q)$: gravity compensation (from EoM)

inverse dynamic control: (PD control with dynamics)

$\tau^* = M(q)\,\ddot{q}^* + b(q,\dot{q}) + g(q) - J_c^T F_c \leftarrow \ddot{q}^* = k_p(q^* - q) + k_d(\dot{q}^* - \dot{q})$

($\omega = \sqrt{k_p}$: eigenfreq., $D = k_d / 2\sqrt{k_p}$: damping)

Task-space dynamic control

(find τ^* to track desired trajectory x_e^*, w_e^* or directly $\dot w_e^*$)

$\dot w_e^* = k_p \begin{pmatrix}\Delta r\\ \Delta\varphi\end{pmatrix} + k_d(w_e^* - w_e)$ (see inv. kinem. chapter for $\Delta r, \Delta\varphi$) ($\omega = \sqrt{k_p}$: eigenfreq, $D = k_d / 2\sqrt{k_p}$: damping)

$\ddot q^* = J_{e0}^+(q)\cdot(\dot w_e^* - \dot J_{e0}(q)\cdot\dot q)$ (from: $\dot w_e = J_{e0}(q)\cdot\ddot q + \dot J_{e0}(q)\cdot\dot q$)

$\tau^* = M(q,\dot q)\ddot q^* + b(q,\dot q) + g(q) - J_c^T F_c$ (from equation of motion)

- **multi-task, equal priority:**

$$\ddot q^* = \begin{pmatrix} \vdash J_1 \dashv \\ \vdots \\ \vdash J_{n_t} \dashv \end{pmatrix}^+ \cdot \left(\begin{pmatrix} \dot w_1^* \\ \vdots \\ \dot w_{n_t}^* \end{pmatrix} - \begin{pmatrix} \vdash \dot J_1 \dashv \\ \vdots \\ \vdash \dot J_{n_t} \dashv \end{pmatrix} \cdot \dot q \right)$$

(n_t : # of tasks i; $\dot w_i^*$: desired v, ω of task i; J_i : geom jac. to w_i $J_{i0,P}, J_{i0,R}$)

- **multi-task, ordered priority:**

solve task 1 then task 2 : $\ddot q^* = J_1^+(\dot w_1^* - \dot J_1 \dot q) + N_1\left((J_2 N_1)^+(\dot w_2^* - \dot J_2 \dot q - J_2 J_1^+ \dot w_1^*)\right)$ ($N_a = \mathbb{I} - J_a^+ J_a$: nullspace of J_a)

solve task 1 to n_t in order: $\ddot q^* = \sum_{i=1}^{n_t} N_i \ddot q_i$, $\ddot q_i = (J_i \bar N_i)^+(\dot w_i^* - \dot J_i \dot q - J_i \sum_{k=1}^{i-1} \bar N_k \ddot q_k)$ ($\bar N_a$: nullspace of $\begin{pmatrix} \vdash J_1 \dashv \\ \vdash J_{a-1} \dashv \end{pmatrix}$)

- **end-effector control:** (simplify above formula for end-effector acc. + forces)

$\Lambda_e \dot w_e + \mu + p = (J_e^T)^{-1}\tau + F_e$ ($\Lambda_e = (J_e M^{-1} J_e^T)^{-1}$, $\mu = \Lambda_e J_e M^{-1} b - \Lambda_e \dot J_e \dot q$, $p = \Lambda_e J_e M^{-1} g$)

control position: $\tau^* = J_e^T(\Lambda_e \dot w_e^* + \mu + p)$ (with $\dot w_e^* = k_p \begin{pmatrix} r_e^* - r_e \\ \Delta\phi_e \end{pmatrix} + k_d(w_e^* - w_e)$)

control pos. + forces: $\tau^* = J_e^T(\Lambda_e \dot w_e^* + F_e^* + \mu + p)$ (take care that dir. of $\dot w_e^*$ and F_c^* allowable!)

internal force control: $\tau^* = J_e^T(\dots) + N(J_e^T)\cdot\tau_0^*$ ($N(J_e^T) = \mathbb{I} - J_e^T J_e^{T+}$ nullspace ; τ_0^* any value to change τ^*)

- **quadratic task optimization:**

write all tasks stacked in $\underline{\underline{A}}$ and $\underline{b}$ as: $\underline{\underline{A}} \cdot \begin{pmatrix} \ddot q \\ F_c \\ \tau \end{pmatrix} - \underline{b} = \underline{0}$ → $\min \left\| \underline{\underline{A}} \begin{pmatrix} \ddot q \\ F_c \\ \tau \end{pmatrix} - \underline{b} \right\|$: $\begin{pmatrix} \ddot q \\ F_c \\ \tau \end{pmatrix} = \underline{\underline{A}}^+ \underline{b} + N(\underline{\underline{A}}) \cdot \underline{x_0}$

- respect eq. of motion : $\underline{\underline{A}} = (M(q), -J_c^T(q), -\mathbb{I})$, $\underline{b} = -b(q,\dot q) - g(q)$
- motion task $\dot w_e^*$: $\underline{\underline{A}} = (J_e, 0, 0)$, $\underline{b} = -\dot J_e \dot q + \dot w^*$
- force task F_i^* : $\underline{\underline{A}} = (0, \mathbb{I}, 0)$, $\underline{b} = F_i^*$
- min torque task : $\underline{\underline{A}} = (0, 0, \mathbb{I})$, $\underline{b} = 0$

if ordered priority $A_1, b_1 \to A_2, b_2$:

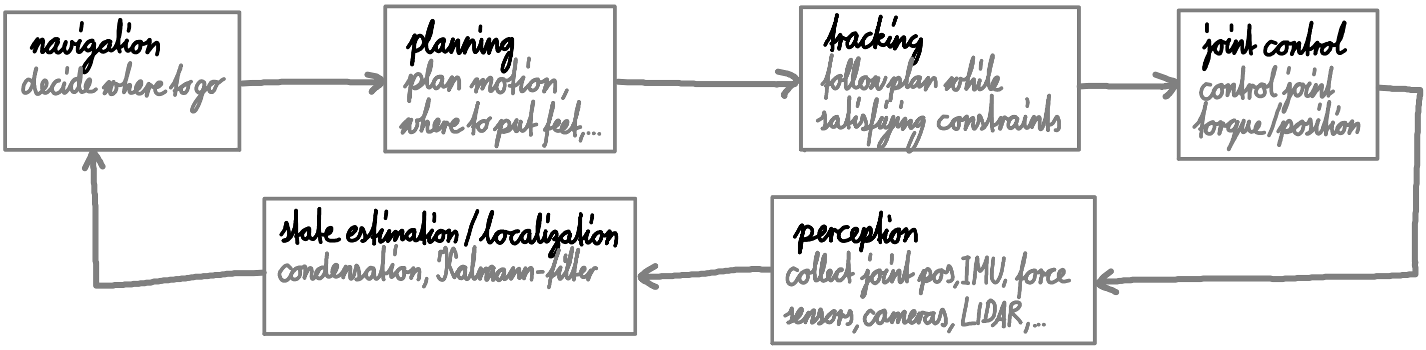
$\begin{pmatrix} \ddot q \\ F_c \\ \tau \end{pmatrix} = A_1^+ b_1 + N(A_1)\, x_0$

$x_0 = (A_2 N(A_1))^+ (b_2 - A_2 A_1^+ b_1)$

casestudies

- **legged robot:** (see floating base chapter!)

actuators need to be: fast, high torque, small+light, efficient, robust, cheap, large motion range
- high-geared motor + torque sensor (+spring on output) : cons. speed, efficiency, robustness
- low-geared high torque motor : cons. big+heavy, expensive
- hydraulic actuator (piston) : cons. big+heavy pump, inefficient
- pneumatic actuator ("muscle"): cons. big+heavy pump, only contraction, hard to control
- other: shape memory aloy, piezo-electric (-polymer) : cons. weak, little travel



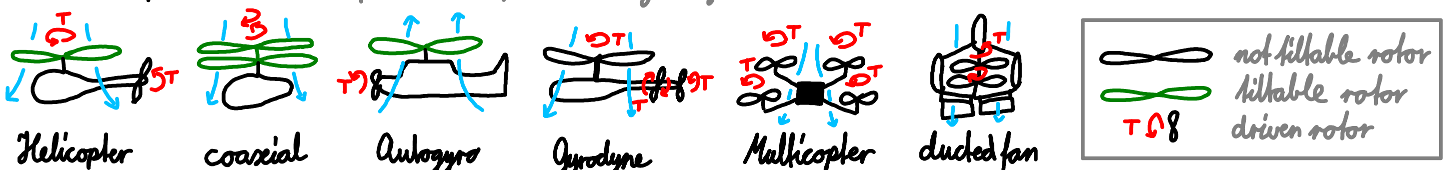
compliant system + force control > position control! (robust, energy storage, power/speed amplification)

tracking constraints: 1) equation of motion 2) no contact motion 3) torque limits 4) friction cone ($F_R < \mu F_N$) 5) desired torso position 6) swing foot tracking 7) desired torso orientation 8) contact force optimization

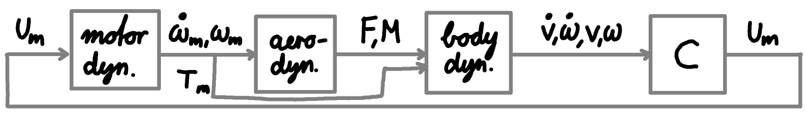
planning methods:
- use simplified model or real observations to generate "realistic" footholds + trajectories
- use complex model + nonlinear MPC to find best footholds + trajectories

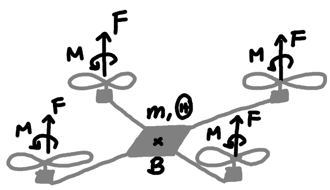
navigation: estimate traversability of terrain by giving cost to steps, slopes, bad terrain. → find path

- **rotorcraft** (aircraf that produce lift from a rotary wing):



quadrorotor example:



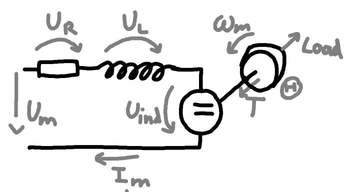


motor dyn.: $\Theta\dot{\omega}_m = T_m - \text{Load}$ ($T_m = \kappa \cdot I_m$) $\quad L\dot{I}_m = U_m - RI_m - U_{ind}$ ($U_{ind} = \kappa\,\omega_m$)

aero dyn.: ${}_BF_i = b \cdot \omega_{m,i}^2 \quad {}_BM_i = d \cdot \omega_{m,i}^2$ (thrust force, drag moment of motor i)

${}_BF_{aero} = \sum_i {}_BF_i \quad {}_BM_{aero} = \sum_i {}_BM_i + r_{Bi} \times F_i$ (" , " of body B)

(hub forces, rolling moments ???)



body dyn.: $\frac{d}{dt}(m \cdot {}_Iv) = {}_IF \rightarrow m \, {}_B\dot{v} + {}_B\omega \times m \cdot {}_Bv = C_{IB}^T \cdot (0,0,m \cdot g) + {}_BF_{aero}$

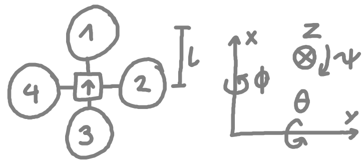
$\frac{d}{dt}({}_I\Theta \cdot {}_I\omega) = {}_IM \rightarrow {}_B\Theta \cdot {}_B\dot{\omega} + {}_B\omega \times {}_B\Theta \cdot {}_B\omega = {}_BM_{aero} + {}_BM_{motors}$

$\left({}_Bv = \dot{\chi}_{PB} \text{ (cartesian!)}, \quad {}_B\omega = E_{R\,eulerXYZ}(\chi_{RB})\,\dot{\chi}_{RB} \right)$

$\hookrightarrow M(\varphi)\ddot{\varphi} + b(\varphi,\dot{\varphi}) + g(\varphi) + J_{ext}^T (F|M)_{ext} = 0$ $\quad \left(\varphi = \binom{\chi_{PB}}{\chi_{RB}}\right.$ body coord. cartesian + roll-pitch-yaw euler.)

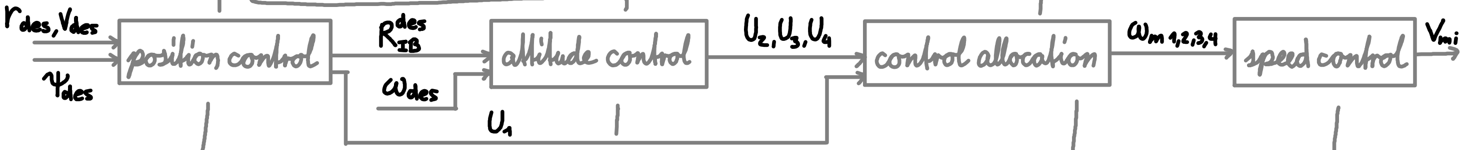
control C: assume motors fast, light: $U_m \sim \omega_m, T_m \approx 0$; drone symmetric: $\Theta = \begin{pmatrix} \Theta_{xx} & & \\ & \Theta_{yy} & \\ & & \Theta_{zz} \end{pmatrix}$; hover: ${}_b\omega \approx \dot{\chi}_{\text{Reuler}XYZ}$

virtual control input:
$$\begin{cases} U_1 = l \cdot b(\omega_{m4}^2 - \omega_{m2}^2) & = \text{roll torque} \\ U_2 = l \cdot b(\omega_{m1}^2 - \omega_{m3}^2) & = \text{pitch torque} \\ U_3 = d(-\omega_{m1}^2 + \omega_{m2}^2 - \omega_{m3}^2 + \omega_{m4}^2) & = \text{yaw torque} \\ U_4 = b(\omega_{m1}^2 + \omega_{m2}^2 + \omega_{m3}^2 + \omega_{m4}^2) & = \text{thrust force} \end{cases}$$



→ EoM:
$$m\dot{v}_x = m(\omega_z v_y - \omega_y v_z) - \sin\theta\, mg$$
$$m\dot{v}_y = m(\omega_x v_z - \omega_z v_x) + \sin\phi\cos\theta\, mg$$
$$m\dot{v}_z = m(\omega_y v_x - \omega_x v_y) + \cos\phi\cos\theta\, mg - U_4$$
$$\Theta_{xx}\dot{\omega}_x = \omega_y\omega_z(\Theta_{yy} - \Theta_{zz}) + U_1$$
$$\Theta_{yy}\dot{\omega}_y = \omega_z\omega_x(\Theta_{zz} - \Theta_{xx}) + U_2$$
$$\Theta_{zz}\dot{\omega}_z = \omega_x\omega_y(\cancel{\Theta_{xx} - \Theta_{yy}}) + U_3$$

body x-axis: not directly controlable
body y-axis: not directly controlable
body z-axis: controlable
body roll: controlable
body pitch: controlable
body yaw: hard to control since U_3 small



position control:
$U_{1des} = (-K_p(r - r_{des}) - K_d(v - v_{des}) - mg)$
$U_1 = U_{1des} \cdot {}_Be_z$ (project des. thrust on body z)
${}_Be_{desz} = \text{normalize}(U_{1des})$
${}_Be_{desY} = {}_Be_{desz} \times (\cos\psi_{des}, \sin\psi_{des}, 0)^T$
${}_Be_{desx} = {}_Be_{desy} \times {}_Be_{desz}$
↳ $R_{IB}^{des} = ({}_Be_{desx}, {}_Be_{desy}, {}_Be_{desz})$

attitude control:
$err_R = \frac{1}{2}\,\text{invskew}(R_{IB}^{des\,T} \cdot R_{IB} - R_{IB}^T \cdot R_{IB}^{des})$
$err_\omega = \omega - R_{IB}^T R_{IB}^{des} \cdot \omega_{des}$
$(U_2, U_3, U_4) = -K_R \cdot err_R - K_\omega \cdot err_\omega$

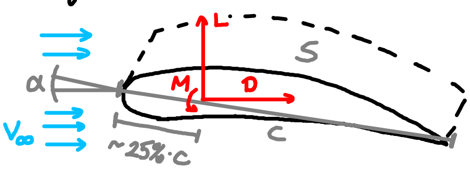
control allocation: invert virtual control inputs

speed control: get voltage for motors $\sim \omega_{mi}^2$

max $\frac{C_L}{C_d}$ → max flight range (dist.)
max $\frac{C_L^3}{C_d^2}$ → max flight endurance (time)

fixed wing (lift trough wings and their shape)

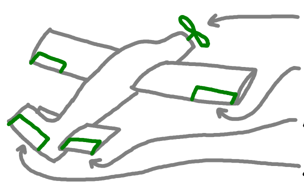
- wing model:



lift force: $L = c_L \cdot \frac{1}{2}\varrho V_\infty^2 S$ {perpendicular to flow V_∞!
drag force: $D = c_d \cdot \frac{1}{2}\varrho V_\infty^2 S$ {parallel to flow V_∞!
moment: $M = c_m \cdot \frac{1}{2}\varrho V_\infty^2 S c$
↑ (attack at aero-center (≈25%·c) then c_m independent of α!)

S: wing area
c: mean chord length
ϱ: density of air
V_∞: air speed
C_L: lift coefficient
C_d: drag coefficient
C_m: moment coefficient

- aircraft model + actuators:



propeller (thrust)
ailerons (rolling)
elevators (pitching)
rudder (yawing)

every wing produces forces L, D, M according to above.
↳ actuators (▭) can change c_L, c_d, c_m

→ L: total lift (⊥ to V_∞)
D: total drag (∥ to V_∞)
Y: side force (⊥ to L, D)
F_G: weight (inertial z-axis)
T: thrust (see multicopter)

L_m: rolling moment (body x-axis)
M_m: pitching moment (body y-axis)
N_m: yawing moment (body z-axis)

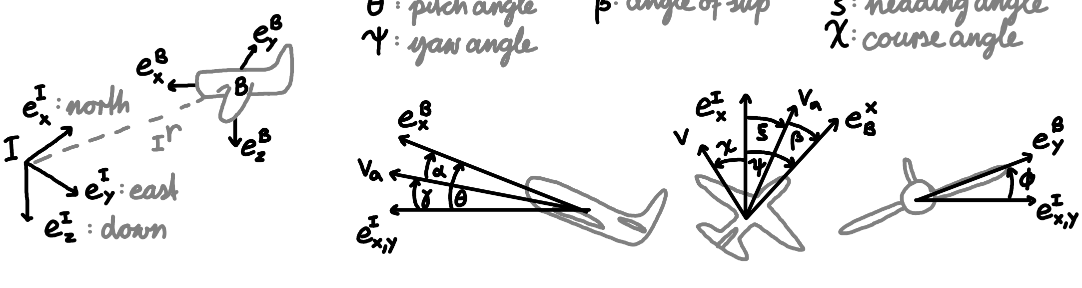
- reference axes + angles:

ϕ: roll angle
θ: pitch angle
ψ: yaw angle

α: angle of attack
β: angle of slip

γ: flight-path angle
ξ: heading angle
χ: course angle

$V_a \hat{=} V_\infty$: airspeed
$V = V_a + \text{wind}$: groundspeed



- *equations of motion:*

$$m \cdot {}_B\dot{V}_a = {}_B\omega \times m \cdot {}_BV_a + \left(T \begin{pmatrix} \cos\varepsilon \\ 0 \\ \sin\varepsilon \end{pmatrix} - D \begin{pmatrix} \cos\alpha \\ 0 \\ \sin\alpha \end{pmatrix} + L \begin{pmatrix} \sin\alpha \\ 0 \\ -\cos\alpha \end{pmatrix} + Y \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} + m \cdot g \begin{pmatrix} \sin\theta \\ 0 \\ \cos\phi\cos\theta \end{pmatrix} \right)$$

$${}_B\Theta \cdot {}_B\dot{\omega} = {}_B\omega \times {}_B\Theta \cdot {}_B\omega + \begin{pmatrix} L_m \\ M_m \\ N_m \end{pmatrix} + (\text{motor-offset torque}) \qquad \left({}_B\omega = E_{R\,eul\,ZYX} \cdot \begin{pmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{pmatrix} \right)$$

$$\boxed{\dot\psi = V_a/R = g \cdot \tan(\phi)/V_a}$$

coordinated turn force balance

- *control example* L1/TECS + PID: (assumes $\psi = \xi$)