Robot Dynamics

Silvan Stadelmann - 30. Januar 2026 - v0.1.0

github.com/silvasta/summary-rodyn

Contents

Kinematics

Exam strategy: 30% DH/forward, 40% Jacobians/singularities, 30% inverse/control. Practice 3R examples; link to dynamics (e.g., Jacobian in torque control).

1 Vectors and Positions

Position vectors: Parametrize P in frame $\mathcal{A}$ as ${}_{\mathcal{A}}\mathbf{r}_{AP} = \mathbf{r}(\chi)$, where χ are parameters (e.g., Cartesian coords)

1.1 Linear Velocity

$$r = r(\chi) \qquad \dot{r} = \frac{\partial r}{\partial \chi}\dot{\chi} = E_p(\chi)\dot{\chi}$$

Exam tip: Basis for Jacobians

Compute for end-effector task velocities in control problems.

1.2 Rotations

Rotation matrix $\mathbf{C}_{\mathcal{AB}}$ transforms vectors from frame $\mathcal{B}$ to $\mathcal{A}$:

$${}_{\mathcal{A}}\mathbf{r}_{AP} = \mathbf{C}_{\mathcal{AB}} \cdot {}_{\mathcal{B}}\mathbf{r}_{AP}$$

Properties Orthogonal ($\mathbf{C}^T = \mathbf{C}^{-1}$), det=1 for proper rotations

Elementary rotations (about x,y,z axes by angle θ):

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

$$\mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Composition: $\mathbf{C}_{\mathcal{AC}} = \mathbf{C}_{\mathcal{AB}} \cdot \mathbf{C}_{\mathcal{BC}}$

Homogeneous transformations (4x4 for position + orientation):

$$\mathbf{T}_{\mathcal{AB}} = \begin{bmatrix} \mathbf{C}_{\mathcal{AB}} & {}_{\mathcal{A}}\mathbf{r}_{AB} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix}$$

Passive: Rotate frame **Active**: Rotate vector

Exam pitfall: Confuse active/passive in inverse kinematics, always specify frames.

1.3 Angular Velocity

Angular velocity $\boldsymbol{\omega}$ satisfies $\dot{\mathbf{C}} = \boldsymbol{\omega} \times \mathbf{C}$, or $\dot{\mathbf{C}} = \mathbf{S}(\boldsymbol{\omega})\mathbf{C}$

$$\mathbf{S}(\mathbf{a}) = \begin{bmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix}.$$

Composition: ${}^{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AC}} = {}^{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} + \mathbf{C}_{\mathcal{AB}}{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{BC}}$

Exam note: Use skew-symmetric for deriving velocity Jacobians

1.4 Parametrization of 3d Rotations

Minimal params: 3 (due to SO(3) manifold)
Common for avoiding singularities in kinematics/control.

- **Rotation matrix** 9 params, 6 orthonormality constraints. Direct but redundant.
- **Euler angles** (e.g., ZYZ): $\mathbf{C} = \mathbf{R}_z(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)$
  3 params, **singularities** at $\theta = 0, \pi$ (gimbal lock).
  Exam: Derive matrix; convert to/from quaternions.
- **Angle-axis** No singularities but multi-valued.
  $\mathbf{C} = \exp(\mathbf{S}(\mathbf{k}\theta)) = \mathbf{I} + \sin\theta\mathbf{S}(\mathbf{k}) + (1-\cos\theta)\mathbf{S}^2(\mathbf{k})$ (Rodrigues). $\mathbf{k}$ unit vector, θ angle.
- **Rotation vector** $\boldsymbol{\rho} = \mathbf{k}\theta$, Similar to angle-axis.
- **Unit quaternions** 4 params, 1 constraint. No singularities, efficient for interpolation/composition.
  $\mathbf{q} = (q_0, \mathbf{q}_v) = (\cos(\theta/2), \mathbf{k}\sin(\theta/2))$, $\|\mathbf{q}\| = 1$.

1.5 Unit Quaternions

To rotation matrix:

$$\mathbf{C}(\mathbf{q}) = \mathbf{I} + 2\mathbf{S}^2(\mathbf{q}_v) + 2q_0\mathbf{S}(\mathbf{q}_v).$$

From matrix Extract $\theta = \arccos((\mathrm{trace}(\mathbf{C}) - 1)/2)$

$\mathbf{q}_1 \circ \mathbf{q}_2 = (q_{10}q_{20} - \mathbf{q}_{1v}\cdot\mathbf{q}_{2v}, q_{10}\mathbf{q}_{2v} + q_{20}\mathbf{q}_{1v} + \mathbf{q}_{1v}\times\mathbf{q}_{2v})$

Rotate vector $\mathbf{v}' = \mathbf{q} \circ (0, \mathbf{v}) \circ \mathbf{q}^{-1}$ (pure quaternion)

Time derivative $\dot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \circ (0, \boldsymbol{\omega})$

Exam: Use for singularity-free integration, common in control velocity loops.

2 Multi Body Kinematics

Generalized coordinates Joint variables $\mathbf{q} = (q_1, \ldots, q_n)^T$ (angles for revolute, displacements for prismatic).

End-effector configuration $\boldsymbol{\chi}_e = (\boldsymbol{\chi}_{eP}, \boldsymbol{\chi}_{eR})^T$ (position + orientation params)

Operational/task space Subset $\boldsymbol{\chi}_o$ for specific tasks (e.g., position only)

2.1 Forward Kinematics

End-effector configuration $\boldsymbol{\chi}_e = f(\mathbf{q})$. For serial chains: Product of homogeneous transforms $\mathbf{T}_{0n} = \mathbf{T}_{01}\mathbf{T}_{12}\cdots\mathbf{T}_{(n-1)n}$

Denavit-Hartenberg (DH) params Standard for link modeling

Exam hint: For revolute, θ_i variable; prismatic, d_i variable.

Common pitfall: Wrong x_i alignment—check perpendicularity.

Transform $\mathbf{T}_{i-1,i} =$

$$\begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Link length: a_i| Link twist: α_i| Link offset: d_i| Joint angle: θ_i

Rules Align z_i with joint axis $i+1$; x_i perpendicular to z_{i-1} and z_i; origin at intersection.

Exam: Assign DH for 3-6 DOF arms (e.g., SCARA, PUMA); compute pose; analyze reachable workspace (volume, boundaries).

2.2 Workspace Analysis

Reachable: All positions end-effector can reach (ignore orientation). **Dexterous**: All poses.

For planar 2R: Annulus with radii $|l_1 - l_2|$ to $l_1 + l_2$. For 3R: Adds redundancy for orientation.

Exam pattern: Sketch workspace for given arm; identify voids/holes due to joint limits.

2.3 Jacobians

Jacobi-Box

Differential map: $\dot{\chi}_e = J_{eA}(q)\dot{q}$

Differential map: $\dot{\boldsymbol{\chi}}_e = \mathbf{J}_{eA}(\mathbf{q})\dot{\mathbf{q}}$

$$J_{eA} = \frac{\partial \chi_e}{\partial q} = \begin{bmatrix} \frac{\partial \chi_1}{\partial q_1} & \cdots & \frac{\partial \chi_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \chi_m}{\partial q_1} & \cdots & \frac{\partial \chi_m}{\partial q_n} \end{bmatrix}$$

Analytical Jacobian For orientation params (Euler rates,..)

Geometric Jacobian Direct velocity map $\mathbf{t} = \mathbf{J}_G\dot{\mathbf{q}}$ differs in orientation (angular velocity vs. rates)

Prismatic Position $\mathbf{J}_{P,i} = \mathbf{z}_{i-1}$, Rotation $\mathbf{J}_{O,i} = \mathbf{0}$
Revolute Pos. $\mathbf{J}_{P,i} = \mathbf{z}_{i-1} \times \mathbf{r}_{ie}$, Rotation $\mathbf{J}_{O,i} = \mathbf{z}_{i-1}$

Singularity: $\det(\mathbf{J}) = 0 \to$ DOF loss. Types: Boundary (workspace edge), internal (e.g., aligned links).
Manipulability: $\mu = \sqrt{\det(\mathbf{J}\mathbf{J}^T)}$; ellipsoid for velocity/-force transmission.

Real-world tip: Use manipulability index for path planning; damp near singularities ($\lambda \propto 1/\mu$) to prevent instability.

Exam: Compute J for 2-4 DOF, find singularities (e.g., $\theta_2 = 0$ in 3R), condition number $\kappa = \sigma_{\max}/\sigma_{\min}$.

Exam tip: For 3R planar (pos only), $J =$

$$\begin{bmatrix} -l_1s_1 - l_2s_{12} - l_3s_{123} & -l_2s_{12} - l_3s_{123} & -l_3s_{123} \\ l_1c_1 + l_2c_{12} + l_3c_{123} & l_2c_{12} + l_3c_{123} & l_3c_{123} \end{bmatrix}$$

det=0 when collinear

2.4 Velocity in Moving Bodies

Rigid Body Formulation Point P velocity in frame A:

$${}^{\mathcal{A}}\mathbf{v}_P = {}^{\mathcal{A}}\mathbf{v}_B + {}^{\mathcal{A}}\boldsymbol{\omega}_{\mathcal{AB}} \times {}^{\mathcal{A}}\mathbf{r}_{BP} + \mathbf{C}_{\mathcal{AB}}{}^{\mathcal{B}}\mathbf{v}^{\text{rel}}_{BP}$$

Twist vector $\mathbf{t} = (\boldsymbol{\omega}, \mathbf{v})^T$ propagates via adjoint:

$$\mathbf{t}_i = \mathrm{Ad}_{\mathbf{T}_{i-1,i}}\mathbf{t}_{i-1} + \mathbf{e}_i\dot{q}_i \quad \text{with unit twist: } \mathbf{e}_i$$

Exam: Recursive forward vel. for chains, links to Newton-Euler dynamics

3 Inverse Kinematics

Main idea: Solve $\mathbf{q} = f^{-1}(\boldsymbol{\chi}_e^\star)$

Analytical for low DOF, e.g. for 2R planar:
$\theta_2 = \pm\arccos((x^2 + y^2 - l_1^2 - l_2^2)/(2l_1l_2))$,
$\theta_1 = \arctan(y, x) - \arctan(l_2\sin\theta_2, l_1 + l_2\cos\theta_2))$
For 6R: Decouple position/orientation (spherical wrist)

Numerical Newton-Raphson: $\mathbf{q}_{k+1} = \mathbf{q}_k + \mathbf{J}^{-1}(\boldsymbol{\chi}^\star - f(\mathbf{q}_k))$

Velocity level $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathcal{X}}_e + (\mathbf{I} - \mathbf{J}^\dagger \mathbf{J})\dot{\mathbf{q}}_0$
(pseudo-inverse for redundancy, nullspace optimization)

Singularities Damped least-squares: $\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2 \mathbf{I})^{-1}$

Redundancy: $n > m$; min-norm or secondary tasks (e.g., joint limit avoidance).

Exam: Solve inverse for 3R arm, handle multiple solutions/elbow configs.

Exam pattern: For PUMA-like (spherical wrist), solve position first (joints 1-3), then orientation (4-6); multiple wrist configs.

3.1 Multi-task control

Single task: $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathcal{X}}^\star$.

Stacked: Combine Jacobians $\mathbf{J}_s = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix}$, solve if consistent.

Prioritized: $\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathcal{X}}_1^\star + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\mathbf{J}_2^\dagger(\dot{\mathcal{X}}_2^\star - \mathbf{J}_2 \mathbf{J}_1^\dagger \dot{\mathcal{X}}_1^\star)$.

3.2 Multi-Task Control

Single task: $\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathcal{X}}^\star$ **Stacked** use: $\mathbf{J}_s = \begin{bmatrix} \mathbf{J}_1 \\ \mathbf{J}_2 \end{bmatrix}$

Prioritized $\dot{\mathbf{q}} = \mathbf{J}_1^\dagger \dot{\mathcal{X}}_1^\star + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\mathbf{J}_2^\dagger(\dot{\mathcal{X}}_2^\star - \mathbf{J}_2 \mathbf{J}_1^\dagger \dot{\mathcal{X}}_1^\star)$

Error Analysis and Trajectories

Task error $\mathbf{e} = \mathcal{X}^\star - \mathcal{X}$

Control $\dot{\mathcal{X}}^\star = \dot{\mathcal{X}}_d + \mathbf{K}\mathbf{e}$ (resolved rate)

Joint trajectory Interpolate $\mathbf{q}(t)$ (cubic poly: $q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3$; match vel/acc)

Exam patterns: Design inverse control loop for redundant arm; avoid singularities via damping/nullspace.

Dynamics

4 Rigid-body Manipulators - Fixed Base

Equation of Motion

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

$M(q)$: Mass/inertia matrix (symmetric, positive definite)
$b(q,\dot{q})$: Coriolis/centrifugal vector $= C(q,\dot{q})\dot{q}$
$g(q)$: Gravity vector
τ : Joint torques/forces (actuators)
$J_c^T F_c$: External contact forces mapped to joint space

Properties: $\dot{M} - 2C$ skew-symmetric
Passivity: $\dot{q}^T(\dot{M} - 2C)\dot{q} = 0$, M bounded/invertible, linear in parameters (for identification)

Exam tip: Derive for 2-3 DOF arms; compute components numerically.

Exam tip: Derive EoM for 2-3 DOF (e.g., 2R planar: compute M as 2x2, C via Christoffel, g from potentials) numerical torque calc at given $q, \dot{q}$

4.1 Principle of Virtual Work (D'Alembert's Principle)

For dynamic equilibrium: Virtual work $\delta W = 0$ for all δq.

$$\sum_i (F_i - m_i \ddot{r}_i)\cdot \delta r_i + \sum_j (T_j - I_j \dot{\omega}_j - \omega_j \times I_j \omega_j)\cdot \delta\theta_j = 0$$

Applies Newton's laws in directions of possible motion, extends to constraints via multipliers.

Exam: Derive EoM for constrained systems (e.g., closed-chain); relate to projected dynamics.

4.2 Single Rigid Body Dynamics

Translational $m\ddot{r} = F$ (Newton)

Rotational $I\dot{\omega} + \omega \times I\omega = T$ (Euler)

Moving frame: Include Coriolis/centrifugal vel. ${}^I v = {}^I \dot{r} + \omega \times r$

Exam: Compute for link in chain, propagate to next (e.g., NE forward pass)

4.3 Newton-Euler Method

Recursive for serial chains ($O(n)$ efficiency)
Forward: Velocities/accelerations base→EE
Backward: Forces/torques EE→base, yields τ_i.
For link i (revolute):

$${}^i\omega_i = {}^i R_{i-1}^{i-1}\omega_{i-1} + \dot{q}_i^i z_i,$$
$${}^i v_i = {}^i R_{i-1}({}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1} p_i) + \dot{q}_i^i z_i,$$
$${}^i \dot{v}_i = {}^i R_{i-1}^{i-1}\dot{v}_{i-1} + {}^i\dot{\omega}_i \times {}^i p_{c,i} + \dots \text{ (full acc inc. Coriolis)}$$

Force: $f_i = m_i \dot{v}_{c,i} + \omega_i \times (\omega_i \times m_i r_{c,i})$

Exam: Apply to 3R arm; compare Lagrangian (same EoM, NE faster for high DOF).

4.4 Projected Newton-Euler

Applies virtual work to multi-body systems.
Project dynamics into joint space.
EoM: $\tau = \sum$ projected inertias/forces (via Jacobians).
Mass: $M_{ij} = \sum_k \text{trace}(J_{v,k}^T m_k J_{v,k} + J_{\omega,k}^T I_k J_{\omega,k})$.
Coriolis/gravity similarly projected.

Exam: Derive M for 2DOF; use trace identity for efficiency.

4.5 Lagrange Formulation

EoM from $\frac{d}{dt}\frac{\partial L}{\partial \dot{q}} - \frac{\partial L}{\partial q} = \tau$ where $L = T - V$.

Energy - **Kinetic** $T = \frac{1}{2}\dot{q}^T M(q)\dot{q}$ **Potential** $V = \sum m_i g^T r_i$

$$M_{kl} = \sum_{i=\max(k,l)}^{n} \text{trace}\left(\frac{\partial T_i}{\partial q_k} J_i \frac{\partial T_i^T}{\partial q_l}\right) + m_i \frac{\partial r_i^T}{\partial q_k}\frac{\partial r_i}{\partial q_l}$$

$$C_{kj} = \sum_i \Gamma_{kji}\dot{q}_i \quad \text{(Coriolis)}$$

$$\Gamma_{kji} = \frac{1}{2}(\partial_k M_{ji} + \partial_j M_{ki} - \partial_i M_{kj}) \quad \text{(Christoffel)}$$

Exam: Full derivation for planar 2R, identify M, C, g M symmetric, C skew contrib, ID params via regression $Y(q,\dot{q},\ddot{q})\theta = \tau$.

Exam: For 2R arm, derive M,C,g via Lagrange (energy) vs NE (recursion); verify same τ at sample $q = (0, \pi/2), \dot{q} = (1,1)$

4.6 External Forces and Torques

Map to joints: $\tau_{ext} = J_P^T F_{ext} + J_R^T T_{ext}$

Forces **Torques**

$$\tau_{F_{ext}} = \sum_{j=1}^{n_f} J_{P,j}^T F_{ext,j} \qquad \tau_{T_{ext}} = \sum_{k=1}^{n_m} J_{R,k}^T T_{ext,k}$$

Actuators

$$\tau_a = \sum_k (J_{S_k} - J_{S_{k-1}})^T F_{a,k} + (J_{R_k} - J_{R_{k-1}})^T T_{a,k}$$

J_P, J_R: Position/rotation Jacobians.

Exam: Compute for EE force, add to EoM.

4.7 Velocity in Moving Bodies

Linear ${}^i v = {}^i \dot{r} + {}^i\omega \times {}^i r$ **Angular** ${}^i\omega$ (Velocity in frame i)

Twist vector: $V = \begin{bmatrix} v \\ \omega \end{bmatrix}$

Propagation ${}^i V_i = {}^i A_{i-1}^{i-1} V_{i-1} + {}^i \dot{q}_i e_i$ (A: adjoint).

Exam: Use in NE, relate to Jacobian columns

4.8 Jacobians for Prismatic/Revolute Joints

Jacobian $J = [J_v \quad J_\omega]$ maps $\dot{q} \to$ task velocity, $\dot{x} = J\dot{q}$

Singularity $\det(JJ^T) = 0$

Prismatic $J_{v,i} = z_{i-1}$, $J_{\omega,i} = 0$

Revolute$i$: $J_{v,i} = z_{i-1} \times (p - p_{i-1})$, $J_{\omega,i} = z_{i-1}$.

Exam: 2R planar Jacobian, singularity when aligned/extended

Exam: Compute J for RRP (SCARA); singularities at det(J)=0 (e.g., arm folded); manipulability $\sqrt{\det(JJ^T)}$.

5 Dynamic Control

Control loops
Position (inner velocity/torque), Torque (feedforward dynamics).

Exam: Block diagrams for PD + gravity comp: $\tau = g(q) + K_p e + K_d \dot{e}$, error $\ddot{e} + K_d\dot{e} + K_p e = M^{-1}\delta\tau$.

5.1 Joint Impedance Control

$\tau = g(q) + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) + K_i \int e\, dt + J^T F_{ext}$

As mass-spring-damper: $\omega_n = \sqrt{K_p/m}$, $\zeta = K_d/(2\sqrt{mK_p})$

Exam: Lyapunov stability $V = \frac{1}{2}\dot{e}^T M \dot{e} + \frac{1}{2} e^T K_p e \to \dot{V} \le 0$

Exam: Prove stability for PD control using Lyapunov $V = \frac{1}{2}\dot{q}^T M \dot{q} + \frac{1}{2} e^T K_p e$, $\dot{V} = -\dot{q}^T K_d \dot{q} \le 0$ (LaSalle for convergence)

5.2 Inverse Dynamics Control (Computed Torque)

$\tau = M(q)(\ddot{q}_d + K_d\dot{e} + K_p e) + b(q,\dot{q}) + g(q)$

Decouples: $\ddot{e} + K_d\dot{e} + K_p e = 0$, crit damp $K_d = 2\sqrt{K_p}$

Exam: Derive error dynamics; choose gains for crit. damping, f.e. overshoot <5%

5.3 Task-Space Dynamic Control

EoM

$$\Lambda(x)\ddot{x} + \mu(x,\dot{x}) + p(x) = F + J^{-T}\tau_{ext}$$

with $\Lambda = (JM^{-1}J^T)^{-1}$

Control $F = \Lambda(\ddot{x}_d + K_d\dot{e}_x + K_p e_x) + \mu + p$

Redundancy weighted psd-inv: $J^\dagger = W^{-1}J^T(JW^{-1}J^T)^{-1}$

Null-space projector $N = I - J^\dagger J$

Multiple tasks Stack Jacobians, project secondary to N

Exam: Prioritize (eg. EE motion > joint limits) compute Λ for 3R

5.4 End-Effector Dynamics

As above and with feedforward $\ddot{x}_d$ from trajectory planning.

Exam: Hybrid with selection S (diag, 0=force,1=motion)

6 Interaction Control

6.1 Operational Space Control

$$\tau = J^T\Lambda(\ddot{x}_d + K_d\dot{e}_x + K_p e_x - JM^{-1}(b+g)) + (I - J^T\bar{J}^T)\tau_0$$
$$\bar{J} = \Lambda^{-1}JM^{-1}$$

Exam: Formula for hybrid: $\tau = J^T(SF_m + (I-S)F_f)$

6.2 Selection Matrix

S: Diagonal, separates DOFs (e.g., force in z, motion in x-y)
Control: Blend impedances.

Exam: Hybrid f-m via S (e.g., $S = I$ motion, $S = 0$ force)

6.3 Inverse Dynamics as QP

Formulation $\min_u \|Au - b\|_W^2$ s.t. constraints (torque limits,...)

Hierarchical Solve primary, project secondary to null

Exam: Formulate for redundancy; weighted pseudo-inv for LS

Exam: Formulate QP for 7DOF arm: min $\|\dot{q}\|$ s.t. $J\dot{q} = \dot{x}_d$, torque bounds; null for secondary (e.g., obstacle avoid).

7 Floating Base Dynamics

For mobile/legged robots: Unactuated base.

7.1 Generalized coordinates

$$q = \begin{pmatrix} q_b \\ q_j \end{pmatrix} \quad \text{with} \quad q_b = \begin{pmatrix} q_{b_P} \\ q_{b_R} \end{pmatrix} \in \mathbb{R}^3 \times SO(3)$$

7.2 Generalized Velocities/Accelerations

Twist-based: $u = [{}^I v_b^T \; {}^b\omega_b^T \; \dot{q}_j^T]^T \in \mathbb{R}^{6+n_j}$; $\dot{u}$ similar. Map: $u = E_{fb}\dot{q}$ (E_{fb} handles rot param, e.g., quats to ang vel).

Exam: Note $\dot{q} \neq u$ due to SO(3); use for non-holonomic systems.

7.3 Generalized velocities and accelerations

$$u = \begin{pmatrix} {}_I v_B \\ {}_B\omega_{IB} \\ \dot{\varphi}_1 \\ \vdots \\ \dot{\varphi}_{n_j} \end{pmatrix} \quad \dot{u} = \begin{pmatrix} {}_I a_B \\ {}_B\psi_{IB} \\ \ddot{\varphi}_1 \\ \vdots \\ \ddot{\varphi}_{n_j} \end{pmatrix} \in \mathbb{R}^{6+n_j} = \mathbb{R}^{n_u}$$

$$u = E_{fb}\cdot\dot{q} \quad \text{with} \quad E_{fb} = \begin{bmatrix} \mathbb{1}_{3\times3} & 0 & 0 \\ 0 & E_{\chi R} & 0 \\ 0 & 0 & \mathbb{1}_{n_j \times n_j} \end{bmatrix}$$

E_{fb} maps quaternions/Euler to twists.

Exam: Note $\dot{q} \neq u$ due to SO(3).

7.4 Differential Kinematics

Floating: $J = [J_b \; J_j]$, $\dot{x} = J(q)u$ (task vel from gen. vel)

7.5 Contacts and Constraints

Hard $J_c u = 0$ (no-slip) **Const. acc.** $J_c\dot{u} + \dot{J}_c u = 0$

Soft $F_c = k\delta + d\dot{\delta}$ **Friction** Cone $|F_t| \le \mu F_n$

Exam: Enforce via multipliers $\lambda = -F_c$
impacts $\Delta u = -(J_c M^{-1} J_c^T)^{-1} J_c u^-$

8 Dynamics of Floating Base Systems

EoM $M(q)\dot{u} + h(q,u) = S^T \tau + J_c^T F_c$

where $S = \begin{bmatrix} 0 & I \end{bmatrix}$ (underactuated base), $h = Cu + g$.

Centroidal: $A_G \dot{u} + \dot{A}_G u = \sum F_{ext} + g_G$
(CMM A_G for momentum).

Exam: Project to constraint null; CoM control for balance

Exam: Derive centroidal momentum for quadruped; control CoM vel via $A_G u$ for balance under disturbances.

8.1 Constraint-Consistent Dynamics

Project to null-space of constraints: $\bar{M}\dot{\bar{u}} + \bar{h} = \bar{S}^T \tau$.

Exam: For legged, balance via CoM control

8.2 Contact Dynamics

Impacts Instant velocitiy change:
$\Delta u = -(J_c M^{-1} J_c^T)^{-1} J_c u^-$ (pre-impact)

Soft Spring-damper $F_c = k\delta + d\dot{\delta}$

8.3 Dynamic Control Methods

Multi-task CoM, feet as priorities.

Inv dyn: $\tau = S^+(M\dot{u}_d + h - J_c^T F_{c,d}) + N\tau_0$

Exam: $\dot{u}_d$ from tasks; QP for torque opt w/ cones.