Robot Dynamics

Silvan Stadelmann - 6. Februar 2026 - v0.2.0

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):

$$\mathbf{T}_{\mathcal{AB}} = \begin{bmatrix} \mathbf{C}_{\mathcal{AB}} & {}_{\mathcal{A}}\mathbf{r}_{AB} \\ \mathbf{0}_{1\times 3} & 1 \end{bmatrix}$$

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

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 $\theta = \arccos((\text{trace}(\mathbf{C}) - 1)/2)$

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

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 $\chi_e = (\chi_{eP}, \chi_{eR})^T$ (position + orientation params)

Operational/task space Subset χ_o for specific tasks (e.g., position only)

2.1 Forward Kinematics

End-effector configuration $\chi_e = f(\mathbf{q})$. 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: $\det(\mathbf{J}) = 0 \rightarrow$ 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), $J =$

$$\begin{bmatrix} -l_1 s_1 - l_2 s_{12} - l_3 s_{123} & -l_2 s_{12} - l_3 s_{123} & -l_3 s_{123} \\ l_1 c_1 + l_2 c_{12} + l_3 c_{123} & l_2 c_{12} + l_3 c_{123} & l_3 c_{123} \end{bmatrix}$$

det=0 when collinear

2.4 Velocity in Moving Bodies

Rigid Body Formulation Point P velocity in frame A:

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

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

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

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

3 Inverse Kinematics

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

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

Velocity level $\dot{\mathbf{q}} = \mathbf{J}^\dagger\dot{\chi}_e + (\mathbf{I} - \mathbf{J}^\dagger\mathbf{J})\dot{\mathbf{q}}_0$ (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 pattern: For PUMA-like (spherical wrist), solve position first (joints 1-3), then orientation (4-6); multiple wrist configs.

3.1 Analytical Inverse Kinematics

For 2R planar arm (lengths l_1, l_2, target (x, y)):

$\theta_2 = \pm\,\text{arccos}\left(\frac{x^2+y^2-l_1^2-l_2^2}{2l_1l_2}\right)$,

$\theta_1 = \text{arctan}(y, x) - \text{arctan}(l_2 \sin\theta_2, l_1 + l_2\cos\theta_2)$

For 3R: Solve for θ_3 via circle intersection, then reduce to 2R (multiple solutions; check workspace).

Solve inverse for 3R, handle multiple solutions/elbow configs

3.2 Multi-Task Control

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

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

Error Analysis and Trajectories
Task error $\mathbf{e} = \chi^\star - \chi$
Control $\dot{\chi}^\star = \dot{\chi}_d + \mathbf{K}\mathbf{e}$ (resolved rate)
Joint trajectory Interpolate $\mathbf{q}(t)$ (cubic poly: $q(t) = a_0 + a_1t + a_2t^2 + a_3t^3$; 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 $\quad m\ddot{r} = F$ (Newton)

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

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

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):
${}^i\omega_i = {}^iR_{i-1}^{i-1}\omega_{i-1} + \dot{q}_i^i z_i$,
${}^iv_i = {}^iR_{i-1}({}^{i-1}v_{i-1} + {}^{i-1}\omega_{i-1} \times {}^{i-1}p_i) + \dot{q}_i^i z_i$,
${}^i\dot{v}_i = {}^iR_{i-1}^{i-1}\dot{v}_{i-1} + {}^i\dot{\omega}_i \times {}^ip_{c,i} + \ldots$ (full acc inc. Coriolis)

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

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

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

Propagation ${}^iV_i = {}^iA_{i-1}^{i-1}V_{i-1} + {}^i\dot{q}_ie_i$ (A: adjoint).

Exam: Use in NE, relate to Jacobian columns

4.8 Jacobians for Prismatic/Revolute Joints

Jacobian $J = \begin{bmatrix} J_v & J_\omega \end{bmatrix}$ maps $\dot{q} \rightarrow$ 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: $\tau = g(q) + K_pe + K_d\dot{e}$, error $\ddot{e} + K_d\dot{e} + K_pe = M^{-1}\delta\tau$.

5.1 Joint Impedance Control

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

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

Exam: Lyapunov stability $V = \frac{1}{2}\dot{e}^T M\dot{e} + \frac{1}{2}e^TK_pe \rightarrow \dot{V} \leq 0$

Exam: Prove stability for PD control using Lyapunov $V = \frac{1}{2}\dot{q}^TM\dot{q} + \frac{1}{2}e^TK_pe, \dot{V} = -\dot{q}^TK_d\dot{q} \leq 0$ (LaSalle for convergence)

5.2 Inverse Dynamics Control (Computed Torque)

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

Decouples: $\ddot{e} + K_d\dot{e} + K_pe = 0$, 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 $F = \Lambda(\ddot{x}_d + K_d\dot{e}_x + K_pe_x) + \mu + p$

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

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

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

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

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

E_{fb} maps quaternions/Euler to twists.

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

Note: Slides often simplify to $\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{S}^\top \boldsymbol{\tau} + \mathbf{J}_c^\top \mathbf{F}_c$ for $\mathbf{h} = \mathbf{b} + \mathbf{g}$. Important property: $\dot{\mathbf{M}} - 2\mathbf{C}$ is skew-symmetric (passivity for control proofs).

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 $|F_t| \leq \mu F_n$

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 $S = [0 \quad I]$ (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.

8.4 Lyapunov Stability for PD Control

For $\tau = -K_p\tilde{q} - K_d\dot{q} + g(q)$:

Candidate $V = \frac{1}{2}\dot{q}^T M\dot{q} + U_g + \frac{1}{2}\tilde{q}^T K_p \tilde{q}$,

$\dot{V} = -\dot{q}^T K_d \dot{q} \leq 0$ (asymptotically stable if $K_p, K_d > 0$)

Application in Robotics

9 Legged Robotics

9.1 Generalized Coordinates & DOF Counting

$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{bmatrix}$ (dim: 6 base + n_j joints, e.g., 18 for 12-joint quad)

- Actuated: $\mathbf{q}_j$ (dim: n_j, controlled by torques $\boldsymbol{\tau}$).
- Unactuated: $\mathbf{q}_b$ (floating base, 6 DOF).
- Contact constraints (point feet, no slip): 3 per stance foot (holonomic: $\mathbf{J}_c\dot{\mathbf{q}} = 0$).

Exam trick: Controllable DOF = total DOF − constraints (e.g., 3 stance legs: 18 − 9 = 9; 3 for swing → 6 internal/force DOF).
- Underactuation degree: unactuated DOF − constraints (if > 0, system underactuated).

9.2 Dynamics Equations

Core equation for floating-base system:

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{b}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{J}_c^T \mathbf{F}_c = \mathbf{S}^T \boldsymbol{\tau}$$

Trick: For support-consistent dynamics, project into null-space of $\mathbf{J}_c$ (removes $\mathbf{F}_c$ dependency): $\mathbf{N}_c = \mathbf{I} - \mathbf{J}_c^+ \mathbf{J}_c$.

Centroidal Momentum (Balance Control)

Centroidal momentum matrix $A_G(q)$:

$\dot{h}_G = A_G\dot{q} = \sum m_i (J_{P,i}^T v_i + J_{R,i}^T (I_i\omega_i + \omega_i \times I_i\omega_i))$

CoM tasks: Control $\dot{h}_G$ via contacts (underactuated base)

9.3 Contact Constraints

- Velocity level: $\mathbf{J}_c\dot{\mathbf{q}} = 0$ (no motion at contact points).
- Acceleration level: $\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = 0$.
- Friction: cone constraints on $\mathbf{F}_c$ (e.g., $\|\mathbf{F}_{c,\perp}\| \leq \mu \mathbf{F}_{c,\|}$).

Exam hint: Always enforce as highest priority in hierarchical control to avoid slippage.

9.4 Inverse Differential Kinematics (Swing Leg Task)

Given constraints $\mathbf{J}_c\dot{\mathbf{q}} = 0$ and task $\mathbf{J}_{\text{swing}}\dot{\mathbf{q}} = \dot{\mathbf{r}}_{\text{swing}}^{\text{des}}$

- Stacked (may be singular): $\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_c \\ \mathbf{J}_{\text{swing}} \end{bmatrix}^+ \begin{bmatrix} 0 \\ \dot{\mathbf{r}}_{\text{swing}}^{\text{des}} \end{bmatrix}$
- Hierarchical/null-space (preferred, robust):

$$\dot{\mathbf{q}} = \mathbf{J}_c^+ \cdot 0 + \mathbf{N}_c\dot{\mathbf{q}}_0 \quad \dot{\mathbf{q}}_0 = (\mathbf{J}_{\text{swing}}\mathbf{N}_c)^+ \dot{\mathbf{r}}_{\text{swing}}^{\text{des}}$$

Trick: Use for tasks like swing foot tracking; extend to acceleration level for dynamics.

9.5 Control Strategies

Common approaches (by robustness/exam frequency):

1. High-gain joint PD/PID: $\boldsymbol{\tau} = \mathbf{K}_p(\mathbf{q}^* - \mathbf{q}) + \mathbf{K}_d(\dot{\mathbf{q}}^* - \dot{\mathbf{q}})$ (poor for impacts due to instability on uneven terrain, prefer torque-based for compliance)
2. Inverse dynamics + low-gain: $\boldsymbol{\tau} = \boldsymbol{\tau}_{FB} + \boldsymbol{\tau}_{FF}$ (model-based feedforward)
3. Support-consistent ID: Project desired $\ddot{\mathbf{q}}^*$ into $\mathbf{N}_c$ null-space.
4. Task-space control: Regulate tasks (e.g., CoM, feet) via QP.
5. Hierarchical QP: Strict priorities (dynamics/contact highest).

Hint: High-gain fails due to impacts/unmodeled terrain; prefer torque control for compliance.

9.6 Actuator Types

Type	Torque Method	Pros/Cons
High-gear + SEA/sensor	Elasticity/sensor	Robust but slower response
Low-gear + current	Current ≈ torque	Fast, backdrivable impact-resistant
Hydraulic	Pressure valve	High power, hard to scale

Trick: Low-gear best for dynamic locomotion (e.g.Mini Cheetah)

9.7 Whole-Body Control (WBC) Hierarchy

Typical priority order:

1. Dynamics: $\mathbf{M}\ddot{\mathbf{q}} + \mathbf{h} = \mathbf{J}_c^T\mathbf{f}_c + \mathbf{S}^T\boldsymbol{\tau}$.
2. Contact stability: $\mathbf{J}_c\ddot{\mathbf{q}} + \dot{\mathbf{J}}_c\dot{\mathbf{q}} = 0$.
3. Tasks (CoM/base/swing): e.g., $\mathbf{J}_{com}\ddot{\mathbf{q}} = \ddot{\mathbf{x}}_{com}^{ref} - \dot{\mathbf{J}}_{com}\dot{\mathbf{q}}$.
4. Regularization: min $\|\boldsymbol{\tau}\|$ or $\|\mathbf{f}_c\|$.

Torque command:

$$\boldsymbol{\tau}^d = \mathbf{M}_j(\mathbf{q})\ddot{\mathbf{q}}^* + \mathbf{h}_j(\mathbf{q}, \dot{\mathbf{q}}) - \mathbf{J}_{s,j}^T(\mathbf{q})\boldsymbol{\lambda}^*$$

Final: $\boldsymbol{\tau}^{ref} = \boldsymbol{\tau}^d + k_p\tilde{\mathbf{q}} + k_d\dot{\tilde{\mathbf{q}}}$.

9.8 Optimization-Based Control (QP/SLQ)

QP for WBC: min $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ s.t. constraints (tasks, limits)

Finite-time OCP (SLQ/DDP):

$$\min_{u(\cdot)} \Phi(x(T)) + \int_0^T L(x(t), u(t), t)\, dt \quad \text{s.t. constraints}$$

Trick: SLQ linear in horizon length (faster than DDP); use for MPC in legged systems.

9.9 Exam Tricks & Hints

- Why torque control? Compliance for rough terrain (vs. stiff position control).
- Underactuation: Torque can't directly control base; use contacts.
- Learning (RL): Robust to sim2real gaps in contacts/perception.
- Common Q: Derive controllable DOF for given stance; formulate QP for multi-task control.

10 Rotorcrafts

10.1 Key Assumptions and Modeling

- Core assumptions: CoG at body frame origin; rigid, symmetric structure; rigid propellers; neglect fuselage drag; near-hover (hub forces & rolling moments ≈ 0).
- Generalized coordinates for aerial robots:

$$q = [C_{BW}, v, \omega, \alpha_i, \omega_i]^T \in \text{SO(3)} \times \mathbb{R}^{6+10}$$

(incl. base orientation, velocities, arm angles, rotor speeds)

- Newton-Euler equations (full dynamics):

$$\sum F_{\text{ext}} = m(\alpha_i, \omega_i)\,\dot{v} + b(v, \omega, \alpha_i, \omega_i) + g(C_{BW}, \alpha_i, \omega_i)$$

$$\sum T_{\text{ext}} = I_B(\alpha_i, \omega_i)\,\dot{\omega} + b(v, \omega, \alpha_i, \omega_i) + g(C_{BW}, \alpha_i, \omega_i)$$

- Simplified for quadrotor: $q = [\Theta, v, \omega]^T \in \mathbb{R}^3 \times \mathbb{R}^6$.
- Derivation methods: Newton-Euler momentum theory, Lagrange, or virtual work principle.

10.2 Propulsion and Thrust Models

- Thrust and drag torque (hover/low-speed): $T_i = b\omega_{p,i}^2$, $Q_i = d\omega_{p,i}^2$
- Classical momentum theory (ideal hover):

$$T = 2\rho A v_i^2, \quad P_{\text{ideal}} = T v_i = \frac{T^{3/2}}{\sqrt{2\rho A}}.$$

- Figure of Merit: $FM = P_{\text{ideal}}/P_{\text{actual}} < 1$.
- Dependencies: Thrust $T \propto \omega^2$ (independent of forward speed near hover); drag torque $Q \propto \omega^2$.

10.3 Control Allocation (Quadrotor, X-Configuration)

- Virtual inputs:

$$\begin{aligned} u_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) && \text{(collective thrust)}, \\ u_2 &= lb(\omega_4^2 - \omega_2^2) && \text{(roll moment)}, \\ u_3 &= lb(\omega_1^2 - \omega_3^2) && \text{(pitch moment)}, \\ u_4 &= d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) && \text{(yaw moment)}. \end{aligned}$$

- Allocation matrix (solve for ω_i^2):

$$\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} \frac{1}{b} & 0 & \frac{1}{lb} & -\frac{1}{d} \\ \frac{1}{b} & -\frac{1}{lb} & 0 & \frac{1}{d} \\ \frac{1}{b} & 0 & -\frac{1}{lb} & -\frac{1}{d} \\ \frac{1}{b} & \frac{1}{lb} & 0 & \frac{1}{d} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}.$$

Note: For + configuration, adjust signs (e.g., $u_2 = lb(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)$) exams often specify X-config.

- For over-actuated systems (e.g., hexacopter): Multiple solutions; choose minimum-energy (min rotor-speed norm) or min max(ω_i).

10.4 Attitude Dynamics (Linearized near Hover)

- Linearized equations:

$$\ddot{\phi} \approx \frac{u_2}{I_{xx}} = \frac{lb}{I_{xx}}(\omega_4^2 - \omega_2^2), \quad \ddot{\theta} \approx \frac{u_3}{I_{yy}}, \quad \ddot{\psi} \approx \frac{u_4}{I_{zz}}.$$

- PD attitude controller (decoupled double integrators):

$$\begin{aligned} u_2 &= k_{p\phi}(\phi_d - \phi) - k_{d\phi}\dot{\phi}, \\ u_3 &= k_{p\theta}(\theta_d - \theta) - k_{d\theta}\dot{\theta}, \\ u_4 &= k_{p\psi}(\psi_d - \psi) - k_{d\psi}\dot{\psi}. \end{aligned}$$

- Tuning hint: k_p sets rise time/natural frequency; k_d sets damping (avoid large k_d to prevent saturation).

10.5 Position and Altitude Control

- Altitude (small-angle approx.):

$$\ddot{z} \approx g - u_1 m^{-1} \cos\phi \cos\theta$$

$$u_1 = (m(g - k_{p_z}(z_d - z) - k_{d_z}\dot{z}))(\cos\phi\cos\theta)^{-1}$$

- Thrust vector control (desired ${}^E\mathbf{T} = [T_x, T_y, T_z]^T$):

$$u_1 = \|{}^E\mathbf{T}\|, \theta_d = \arcsin\left(\frac{T_x}{u_1}\right), \phi_d = -\arcsin\left(\frac{T_y}{u_1}\right)$$

- Forward acceleration: $\propto \sin\theta$ or $\sin\phi$ (indep. of yaw ψ)

Tricks and Hints

- Singularities: Euler angles singular at $\theta = \pm 90°$; use quaternions for aggressive maneuvers.
- Derivatives: Near hover, $\dot{\phi} \approx p, \dot{\theta} \approx q, \dot{\psi} \approx r$.
- Linearization: Small-angle approx. $\sin\phi \approx \phi$, $\cos\phi \approx 1$.
- Power: $\propto \omega^3$; minimize max(ω_i) over sum ω_i^2.
- Underactuation: Position/attitude coupled; check config. (X vs. +) for allocation.
- Common pitfalls: Division by $\cos\phi\cos\theta$ (numerical issues); neglect drag in hover.
- Thrust vector: arcsin can cause singularities at high tilts; use quaternions for large angles (as in slides).

11 Fixed-Wing Aircraft

Key Assumptions and Simplifications

- **Rigid symmetric structure** Constant, diagonal inertia matrix
- **Constant mass** Neglect fuel burn (electric/short duration)
- **No stall** Operating strictly in the linear lift domain
- **Lumped Aerodyn.** Lift/drag of wing and fuselage combined
- **No Side-Force** Sideslip β is regulated to 0
- **No Interference effects** e.g. prop wash on control surfaces
- **CoM Origin** Body frame origin placed at Center of Mass

11.1 Reference Frames and Kinematics

- **Inertial Frame E (NED)** North (x_E), East (y_E), Down (z_E)
- **Body Frame B** x_B (nose), y_B (right wing), z_B (down)
- **Wind Frame W** x_W aligned with air-mass velocity vector $\mathbf{V}_a$

Body Velocity in B(ody frame): $\mathbf{V}_a^B = (u, v, w)^T$
Airspeed: $V = \|\mathbf{V}_a^B\| = \sqrt{u^2+v^2+w^2}$ (always positive)
Body Rates (in B): $\boldsymbol{\omega}^B = (p, q, r)^T$

Airflow Angles:

$$\alpha = \arctan(w/u) \quad \text{(Angle of Attack)}$$
$$\beta = \arcsin(v/V) \quad \text{(Sideslip Angle)}$$

Euler Angles: $\mathbf{X} = (\phi, \theta, \psi)^T$ (roll, pitch, yaw)

Rotation Matrix C_{EB} (ZYX sequence):

$$\mathbf{V}_a^E = C_{EB}\,\mathbf{V}_a^B, \quad C_{EB} = C_z(\psi)\,C_y(\theta)\,C_x(\phi)$$

Angular Rates Relation:

$$\begin{pmatrix} p \\ q \\ r \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{pmatrix} \begin{pmatrix} \dot\phi \\ \dot\theta \\ \dot\psi \end{pmatrix}.$$

Singularity at $\theta = \pm\pi/2$ (gimbal lock, use quaternions)

Polar Coordinates (No Wind Assumption)

Longitudinal θ (pitch), γ (flight path), α, Relation: $\theta = \alpha + \gamma$
Lateral ψ (yaw/heading), ξ (course)

Why no wind? If wind = 0, ground velocity equals airspeed, simplifying position derivatives.

With Steady Wind $\mathbf{v}_g = \mathbf{v}_a + \mathbf{v}_w$ (Wind Triangle) Difference between Heading ψ and Course angle χ (direction of $\mathbf{v}_g$) χ is the *Crab Angle*.

11.2 Aerodynamics

Bernoulli Equation (Incompressible) $\frac{1}{2}v^2 + gh + \frac{P}{\rho} = \text{const}$

Dynamic Pressure: $q = \frac{1}{2}\rho V^2$ (ρ: air density, V: airspeed)
Aerodynamic Forces & Moments

Type	2D Airfoil (per unit span)	3D Whole Aircraft
Lift ($\perp$)	$dL = q\,c\,C_l\,dy$	$L = q\,S\,C_L$
Drag ($\parallel$)	$dD = q\,c\,C_d\,dy$	$D = q\,S\,C_D$
Moment	$dM = q\,c^2\,C_m\,dy$	$M = q\,S\,\bar{c}\,C_M$

q: dynamic pressure, S: wing area, $c/\bar{c}$: chord/mean
$C_{l,d,m}$ and $C_{L,D,M}$ functions of: α, Re, Ma (Mach)
Reynolds Number: $\text{Re} = \frac{VL}{\nu}$ (L: char. length, ν: kin. viscosity)

Stall: Occurs at high α; C_L drops post-max. Avoid in operation!

- Coefficients from CFD (Xfoil, Javafoil), wind tunnel, flight data sysid or linear approximations ($C_L \approx C_{L0} + C_{L\alpha}\alpha$)
- Control surfaces modify C_L/C_M:
  Ailerons (Roll), Elevator (Pitch), Rudder (Yaw)
- **Steady Level Flight:** $L = mg$, $D = T$ (thrust)

11.3 Equation of Motion and Dynamics

Model as single rigid body (Newton-Euler):

$$m\dot{\mathbf{v}}^B = \mathbf{F}^B - \boldsymbol{\omega}^B \times m\mathbf{v}^B$$
$$\mathbf{I}\dot{\boldsymbol{\omega}}^B = \mathbf{T}^B - \boldsymbol{\omega}^B \times \mathbf{I}\boldsymbol{\omega}^B$$

Forces in B: $\mathbf{F}^B = \mathbf{F}^B_{\text{aero}} + \mathbf{F}^B_{\text{prop}} + \mathbf{F}^B_{\text{grav}}$

Simplified Aero Forces (in wind frame, then rotate):

- Assume no side force ($Y = 0$).
- Lift/Drag lumped.

11.4 Key Flight Conditions

Steady Level Flight

$$C_L = \frac{2mg}{\rho V^2 S}, \quad T = D.$$

Note: Drag $D \propto m$ (since $V \propto \sqrt{2mg/(\rho S C_L)}$ at fixed α)

Coordinated Turn (no sideslip, constant altitude/speed)

$$\tan\phi = \frac{V^2}{gR_{\text{turn}}}, \quad L = \frac{mg}{\cos\phi}$$

Stall speed increases by $\sqrt{n}$ where $n = \frac{1}{\cos\phi} = \frac{L}{W}$ load Factor (check $C_L < C_{L,\text{max}}$ for margin)

Gliding (Thrust $T = 0$)

$$\gamma = -\arctan(D/L) \approx -C_D/C_L \text{(for small angles)}$$

Note: Max range occurs at max L/D

- **Best: Range** max C_L/C_D **Endurance** max $C_L^{3/2}/C_D$
- **Stall Speed** $V_{\text{stall}} = \sqrt{\frac{2mg}{\rho S C_{L,\text{max}}}}$
- **Stall Margin** Required C_L increases in turn, check $C_{L,\text{max}}$
- **Guidance** Use Ground speed for navigation, Airspeed for control loop (aerodynamic stability)

11.5 Control Strategies

- **Cascaded Control** Position (Outer) $\rightarrow$ Velocity/Attitude (Mid) $\rightarrow$ Rates (Inner).
- **Attitude Control** PID often sufficient.
- **Guidance** Feedforward is useful for wind rejection.
- **Underactuated** Cannot control all 6 DOFs independently (e.g., rolling creates yaw/sideslip).

12 Recipes - Steps to Solve Common Tasks

12.1 1. System Analysis & DoF Counting (General/-Legged Systems)

Start every problem with this to identify underactuation and constraints (extends legged.tex DoF concepts).

- **Generalized Coordinates** (n): Size of q (e.g., fixed base: $n = n_{joints}$; 3D floating base: $n = 6 + n_{joints}$; planar: $n = 3 + n_{joints}$).
- **Constraints** (n_c): Per contact (e.g., point no-slip: 3 in 3D, 2 in 2D; surface: 6 in 3D).
- **Uncontrollable DoFs**: dim(Base) $-$ rank($J_{contact}$) (e.g., planar base with 1 point contact: $3 - 2 = 1$ uncontrollable).
- **Controllable DoFs**: Total $n - n_c$ (subtract unactuated base if underactuated).

12.2 2. Geometric Jacobian Derivation (Serial Chains)

To relate $\dot q$ to end-effector twist (merges both lists; extends kinematics.tex with step-by-step construction).

- Assign DH parameters/frames.
- For revolute joint i: $J_{P,i} = z_{i-1} \times (r_E - r_{i-1})$, $J_{R,i} = z_{i-1}$.
- For prismatic: $J_{P,i} = z_{i-1}$, $J_{R,i} = 0$.
- Total $J = \begin{bmatrix} J_P \\ J_R \end{bmatrix}$.
- Singularity: $\det(JJ^T) = 0$; manipulability $\mu = \sqrt{\det(JJ^T)}$.

12.3 3. Inverse Kinematics (Velocity Level, Single/Multi-Task)

To find $\dot q$ for desired task velocity (merges both lists; extends kinematics.tex with hierarchical details).

- Single task: $\dot q = J^\dagger v_{task}$.
- Hierarchical (null-space): Primary J_1, v_1: $\dot q = J_1^\dagger v_1 + (I - J_1^\dagger J_1)\dot q_{null}$, where $\dot q_{null} = J_2^\dagger(v_2 - J_2 J_1^\dagger v_1)$.
- Multi-task: $\dot q = \sum N_i \dot q_i$, $\dot q_i = (J_i N_i)^+(v_i^* - J_i \sum_{k<i} \dot q_k)$.
- Near singularity: Damped LS $J^* = J^T(JJ^T + \lambda I)^{-1}$.

12.4 4. Derive EoM (Lagrange for Fixed-Base Arms)

For 2R/3R manipulators (from List 1; extends dynamics.tex with explicit computation steps).

- Kinetic $T = \sum \frac{1}{2} m_i v_i^2 + \frac{1}{2}\omega_i^T I_i \omega_i$.
- Potential $U = \sum m_i g h_i$.
- $\frac{d}{dt}(\partial L/\partial \dot q_k) - \partial L/\partial q_k = \tau_k$ ($L = T - U$).
- Compute M, C (Christoffel: $C_{kj} = \frac{1}{2}\sum_i(\partial M_{kj}/\partial q_i + \partial M_{ki}/\partial q_j - \partial M_{ij}/\partial q_k)\dot q_i$), g.

12.5 5. Inverse Dynamics (Joint/Task Space)

To compute τ for desired motion (from List 2; extends dynamics.tex with force and task handling).

- Joint space: $\tau = M(q)(\ddot q^* + K_p e + K_d \dot e) + h(q, \dot q)$ ($h = C\dot q + g$).
- With external force: $\tau_{total} = \tau_{motion} + J^T F_{ext}$.
- Task space: Solve $\ddot q_{des} = J^\dagger(\ddot x^* - \dot J \dot q)$, then plug into EoM.

12.6 6. Floating Base Dynamics & Contact Consistency

For underactuated systems with contacts (from List 2; extends dynamics.tex and legged.tex with projection steps).

- Full EoM: $M\dot u + h = S^T\tau + J_c^T F_c$.
- Constraint: $J_c \dot u = -\dot J_c u$ (no slip).
- Projected dynamics: Use projector P ($PJ_c^T = 0$): $P(M\dot u + h) = PS^T\tau$.
- Note: Eliminates F_c for feasibility checks.

12.7 7. Hierarchical Optimization (QP) Formulation

For legged/floating-base control (merges both lists; extends legged.tex and dynamics.tex with matrix setup).

$\min_{\ddot q, \tau, f_c} \|\tau\|^2 + \epsilon\|\ddot q\|^2 + \epsilon\|f_c\|^2$ s.t.
- Dynamics: $M\ddot q + h = S^T\tau + J_c^T f_c$.
- Contact: $J_c \ddot q + \dot J_c \dot q = 0$.
- Task (e.g., CoM): $J_{com}\ddot q + \dot J_{com}\dot q = \ddot x^*_{com}$.
- Bounds: $|\tau| \le \tau_{max}$, friction cone $\|f_{c,\perp}\| \le \mu f_{c,z}$.

Matrix form: Stack into $Ax = b$ (e.g., $x = [\dot u^T, \tau^T, F_c^T]^T$).

12.8 8. Multirotor Control Allocation & Hover

For quad/hexacopters (merges both lists; extends rotor.tex with tilt and over-actuated cases).

- Hover: Total $T = mg$, per rotor $\omega_i = \sqrt{T/(4b)}$ (quad).
- Allocation matrix: Solve $A\omega^2 = [T, M_x, M_y, M_z]^T$ (pseudo-inverse for over-actuated; min $\sum\omega^2$).
- Tilt for accel: $\theta_d = \arcsin(a_x/g)$, check drag $D \propto \omega^2$.
- Yaw signs: Positive d for CCW rotors (CW torque on body).

12.9 9. Fixed-Wing Performance & Steady States

For level flight and turns (merges both lists; extends wing.tex with optimal velocities and safety checks).

- Steady level: $L = mg, T = D = \frac{1}{2}\rho v^2 S C_D$; $C_L = \frac{2mg}{\rho v^2 S}$.
- Optimal: Max range at max C_L/C_D; max endurance at max $C_L^{1.5}/C_D$; gliding angle $\tan\gamma = (L/D)^{-1}$.
- Coordinated turn: $\tan\phi = \frac{v^2}{gR}$, load $n = 1/\cos\phi$; required $C_L = n(2mg/(\rho v^2 S))$.
- Safety: $C_L < C_{L,max}$ (stall check); min R at max bank without stall.