1 Frame Transformations

Points: $p^0 = O_0^0 + \begin{bmatrix} a & b & c \end{bmatrix}^T = O_0 + ax_0^0 + by_0^0 + cz_0^0$

Vectors: $v^0 = \begin{bmatrix} d & e & f \end{bmatrix}^T = dx_0^0 + ey_0^0 + fz_0^0$

1.1 Rotations

Rotation Matrices: R_1^0 denotes the rotation of frame 1 from frame 0.

$$R_1^0 = \begin{bmatrix} x_1^0 & y_1^0 & z_1^0 \end{bmatrix} = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix}$$

$R \in SO(3)$ where $SO(3) = \{\, R \in \mathbb{R}^{3\times 3} \mid R^T = R^{-1}, \det(R) = 1 \,\}$.

Changing between frames: $v^0 = R_1^0 v^1$. Compose rotations: $R_2^0 = R_1^0 R_2^1$. R_1^0 is the rotational transformation R we apply to x_0, y_0, z_0 to get x_1, y_1, z_1.

Rotational Transformations: If R is expressed in frame 0, then in frame 1, $R' = (R_1^0)^T R (R_1^0)$. i.e. $w^0 = Rv^0 \implies w^1 = R'v^1 = (R_1^0)^T R (R_1^0) v^1$.

When composing rotations each one should be expressed in the new frame after the previous rotation, otherwise a similarity transformation is needed.

Elementary Rotations:

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \quad R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \quad R_{z,\theta} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

zyz **Euler Angles:** $R_1^0 = R_{z,\phi} R_{y,\theta} R_{z,\psi} =$

$$\begin{bmatrix} \cos\phi\cos\theta\cos\psi - \sin\phi\sin\psi & -\cos\phi\cos\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\sin\theta \\ \sin\phi\cos\theta\cos\psi + \cos\phi\sin\psi & -\sin\phi\cos\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\sin\theta \\ -\sin\theta\cos\psi & \sin\theta\sin\psi & \cos\theta \end{bmatrix}$$

Assuming $\sin\theta > 0$:

$$\begin{aligned} \theta &= \cos^{-1}(r_{33}) \\ &= \operatorname{atan2}\left(\sqrt{1 - r_{33}^2}, r_{33}\right) \\ \psi &= \operatorname{atan2}(r_{32}, -r_{31}) \\ \phi &= \operatorname{atan2}(r_{23}, r_{13}) \end{aligned}$$

Assuming $\sin\theta < 0$:

$$\begin{aligned} \theta &= -\cos^{-1}(r_{33}) \\ &= \operatorname{atan2}\left(-\sqrt{1 - r_{33}^2}, r_{33}\right) \\ \psi &= \operatorname{atan2}(-r_{32}, r_{31}) \\ \phi &= \operatorname{atan2}(-r_{23}, -r_{13}) \end{aligned}$$

Axis-Angle: $\theta = \cos^{-1}\left(\frac{\operatorname{tr}(R) - 1}{2}\right) \qquad k = \frac{1}{2\sin\theta}\begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix}$

1.2 Rigid Transformations

Rigid Motions: $T(p^0) = Rp^0 + d^0, R \in SO(3), d^0 \in \mathbb{R}^3 \qquad p^0 = O_1^0 + R_1^0 p^1$

Homogeneous Transformations: $P^0 = \begin{bmatrix} p^0 \\ 1 \end{bmatrix} = H_1^0 \begin{bmatrix} p^1 \\ 1 \end{bmatrix}, H_1^0 \in SE(3)$

$$SE(3) = \left\{ \begin{bmatrix} R & d \\ 0_{1\times 3} & 1 \end{bmatrix} \;\middle|\; R \in SO(3), d \in \mathbb{R}^3 \right\} \qquad H^{-1} = \begin{bmatrix} R^T & -R^T d \\ 0_{1\times 3} & 1 \end{bmatrix}$$

$\text{Rot}_{x,\alpha}$ denotes pure rotation, $\text{Trans}_{x,a}$ denotes pure translation.

1.3 Velocity Transformations

Skew-Symmetric Form:

$$S(w) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \qquad w \times v = S(w)v \qquad x^T S(w) x = 0$$

$$S(w_1 + \lambda w_2) = S(w_1) + \lambda S(w_2) \qquad RS(w)R^T = S(Rw), \forall R \in SO(3)$$

Cross Product: $a \times b = -b \times a \quad R(a \times b) = (Ra) \times (Rb), \forall R \in SO(3)$

Angular Velocity: $\dot{R}_1^0 = S(w_1^0) R_1^0 \iff S(w_1^0) = \dot{R}_1^0 (R_1^0)^T$ where w_1^0 is the angular velocity of frame 1 with respect to frame 0 (expressed in frame 0). Compose through addition: $w_2^0 = w_1^0 + R_1^0 w_2^1$.

Shorthand: $w_{i,j}^k = R_i^k w_j^i \qquad w_n^0 = w_{0,1}^0 + \cdots + w_{n-1,n}^0 = \sum_{i=1}^n R_{i-1}^0 w_i^{i-1}$

Linear velocity of a point on a rotating body is $v = w \times p$ where p is a vector from the axis of rotation.

Velocity in Another Frame: For point p with velocity $\dot{p}^1$ and position p^1 in frame 1, moving relative to 0, $\dot{p}^0 = \dot{O}_1^0 + w_1^0 \times (R_1^0 p^1) + R_1^0 \dot{p}^1$.

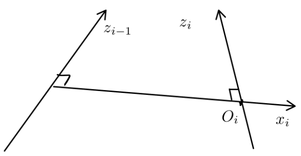
For frames, $\dot{O}_2^0 = \dot{O}_1^0 + w_1^0 \times (R_1^0 O_2^1) + R_1^0 \dot{O}_2^1$.

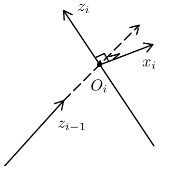
2 Denavit-Hartenberg Parameters

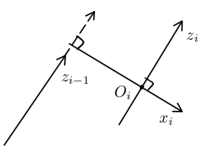
DH Frame Rules: Frame 0 is the inertial frame; frame i is rigidly attached to link i, so when joint i is actuated, frame i moves with link i. Axis rules: x_i is orthogonal to and intersects z_{i-1}.

DH Frame Assignment:

1. Assign z_i to the axis of actuation of joint $i+1$ and assign x_0 arbitrarily.
2. Assign x_i for $i = 1, \ldots, n-1$ according to cases:
   (a) z_{i-1}, z_i not coplanar: x_i is in the direction of the shortest segment joining the axes.
   (b) z_{i-1}, z_i intersect: x_i is normal to the plane formed by z_{i-1}, z_i.
   (c) z_{i-1}, z_i parallel: x_i is any vector orthogonal and intersecting z_{i-1}.
3. Assign the end-effector x_n orthogonal and intersecting z_{i-1}, then z_n arbitrarily.

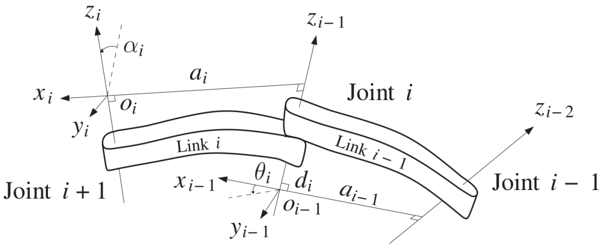






DH Parameters:

1. Link length a_i: the signed distance between z_{i-1} and z_i, along x_i
2. Link twist α_i: the signed angle between z_{i-1} and z_i, about x_i
3. Link offset d_i: the signed distance between O_{i-1} and O_i, along z_{i-1}
4. Joint angle θ_i: the signed angle between x_{i-1} and x_i, about z_{i-1}



DH Homo. Transform: $H_i^{i-1} = \text{Rot}_{z,\theta_i} \text{Trans}_{z,d_i} \text{Trans}_{x,a_i} \text{Rot}_{x,\alpha_i}$

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

Recovering Manipulator From DH Table:

1. Start at frame 0 and move to next frame in sequence.
2. Identify direction of x_{i+1} by rotating x_i by θ_i around z_i.
3. Identify O_{i+1} by moving d_i along z_i and a_i along x_{i+1}.
4. Identify direction of z_{i+1} by rotating α_i along x_{i+1}.

3 Kinematics

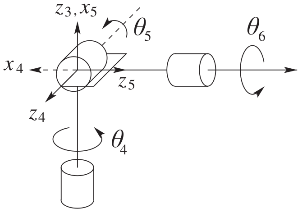
3.1 Forward Kinematics

Given joint angles q, $H_n^0(q) = H_1^0(q_1) H_2^1(q_2) \cdots H_n^{n-1}(q_n)$.

3.2 Inverse Kinematics

If $n < 6$, no solutions in general. If $n = 6$ there are finite solutions. If $n > 6$ there are infinite solutions.

Kinematic Decoupling: Assuming $n = 6$ and last 3 joints form a spherical wrist, $x_4 = -z_3 \times z_4$. Let $O_c^0 = O_d^0 - R_d^0 \begin{bmatrix} a_6 & 0 & d_6 \end{bmatrix}^T$. Solve for q_1, q_2, q_3 given desired $O_c = O_4 = O_5$. Then using $R_6^3 = (R_3^0)^T R_d^0$ solve for q_4, q_5, q_6 as zyz Euler angles ($q_4 = \phi, q_5 = \theta, q_6 = \psi$).



Spherical Wrist DH Table:

Link	a	α	d	θ
4	0	$-\pi/2$	d_4	θ_4^*
5	0	$\pi/2$	0	θ_5^*
6	a_6	0	d_6	θ_6^*

3.3 Forward Velocity Kinematics

End-Effector Velocity: $\dot{O}_i^0 = \dot{O}_{i-1}^0 + w_{i-1}^0 \times (R_{i-1}^0 O_i^{i-1}) + R_{i-1}^0 \dot{O}_i^{i-1}$ where $O_i^0 = O_{i-1}^0 + R_{i-1}^0 O_i^{i-1}, w_i^0 = w_{i-1}^0 + R_{i-1}^0 w_i^{i-1}$.

Revolute joint: $w_i^{i-1} = \begin{bmatrix} 0 & 0 & \dot{q}_i \end{bmatrix}^T, \dot{O}_i^{i-1} = w_i^{i-1} \times O_i^{i-1}$.

Prismatic joint: $w_i^{i-1} = 0, \dot{O}_i^{i-1} = \begin{bmatrix} 0 & 0 & \dot{q}_i \end{bmatrix}^T$

Strategy: If forward kinematic expression is simple, get $\dot{O}_n^0$ by direct differentiation, and get w_n^0 by $S(w_n^0) = \dot{R}_n^0 (R_n^0)^T$ and directly differentiating the rotation matrix. Use Jacobian formulas otherwise.

Indicator Function: $\rho_i = 0$ for prismatic joints, 1 for revolute joints.

Angular Velocity Jacobian: $w_n^0 = J_w(q)\dot{q}, J_w(q) = \frac{\partial w_n(q)}{\partial q} \in \mathbb{R}^{3\times n}$

$$J_w(q) = \begin{bmatrix} \rho_1 z_0^0 & \rho_2 z_1^0 & \cdots & \rho_n z_{n-1}^0 \end{bmatrix}$$

Linear Velocity Jacobian: $\dot{O}_n^0 = J_v(q)\dot{q}, J_v(q) = \frac{\partial O_n^0(q)}{\partial q} \in \mathbb{R}^{3\times n}$

$$J_v(q) = \begin{bmatrix} J_{v,1}(q) & \cdots \end{bmatrix}, J_{v,i}(q) = \begin{cases} z_{i-1}^0 & i \text{ prismatic} \\ z_{i-1}^0 \times (O_n^0 - O_{i-1}^0) & i \text{ revolute} \end{cases}$$

The complete Jacobian is $J(q) = \begin{bmatrix} J_v(q) \\ J_w(q) \end{bmatrix}$.

3.4 Inverse Velocity Kinematics

If $n < 6$, no solutions in general. If $n = 6$, (unique) solution exists iff $J(q)$ is invertible. If $n > 6$, infinite solutions exist, iff $J(q)$ has rank 6.

Pseudoinverse: $J^\dagger = J^T(JJ^T)^{-1}$, satisfies $JJ^\dagger = I$.

$n > 6$ **Solution:** $\dot{q} = J^\dagger(q)\xi + (I - J^\dagger(q)J(q))b$ where $b \in \mathbb{R}^n$ is arbitrary and $\xi = (\dot{O}_n^0, w_n^0)$ is the desired velocity.

4 Other Applications

Basic Motion Planning: Define quadratic potential $U(q) = \frac{1}{2}\|O_d^0 - O_n^0(q)\|^2$. Use velocity $\dot{q} = -\gamma \nabla_q \mathcal{U}(q) = \gamma \left(\dfrac{\partial O_n^0(q)}{\partial q}\right)^T (O_d^0 - O_n^0(q))$.

Numerical Inverse Kinematics: Let $x = \begin{bmatrix}(O_n^0)^T & \phi & \theta & \psi\end{bmatrix}^T \in \mathbb{R}^6$ describe the state (end-effector pose). The analytic Jacobian $J_a(q)$ contains derivatives of x with respect to q, and satisfies $\dot{x} = J_a(q)\dot{q}, \dot{q} = J_a^\dagger(q)\dot{x}$. Form ODE $\dot{q} = J_a^\dagger(q)K(x_d - x(q)) = J_a^\dagger(q)Ke$ and solve using numerical methods, and q should converge to a solution.

End-Effector Force and Torque: Given desired wrench at the end-effector $F^0 = \begin{bmatrix}f^0 & n^0\end{bmatrix}^T$ where f^0 is a force and n^0 is a torque, each joint's force/torque is $\tau = J^T(q)F^0$.

4.1 Potential Field Path Planning

Potential Field Method: Given q^s, q^f, convex obstacles, initialize $q^0 = q^s$,

$$q^{k+1} = q^k + \alpha_k \sum_{i=1}^{n} J_{v,O_i}^T(q^k)\left(F_{i,att}(O_i^0(q^k)) + F_{i,rep}(O_i^0(q^k))\right)$$

until convergence $\|q^{k+1} - q^f\| < \varepsilon$, outputs a set of waypoints in joint space.

Attractive Potential: $\mathcal{U}_{i,att}(O_i^0) = \dfrac{1}{2}c_i\|O_i^0 - \bar{O}_i^0\|^2$ for weights $c_i > 0$. Gradient: $F_{i,att}(O_i^0) = -\nabla\mathcal{U}_{i,att}(O_i^0) = -c_i(O_i^0 - \bar{O}_i^0)$.

Repulsive Potential: For weights $\eta_i > 0$, region of influence ρ_0, and closest obstacle point $\pi(O_i^0)$, when $\rho(O_i^0) = \|O_i^0 - \pi(O_i^0)\| > \rho_0$:

$$\mathcal{U}_{i,rep} = \frac{\eta_i}{2}\left(\frac{1}{\rho(O_i^0)} - \frac{1}{\rho_0}\right)^2 \qquad F_{i,rep}(O_i^0) = \eta_i\left(\frac{1}{\rho(O_i^0)} - \frac{1}{\rho_0}\right)\frac{(O_i^0 - \pi(O_i^0))}{\rho(O_i^0)^3}$$

Jacobians: $J_{v,O_i}(q) = \frac{\partial O_i^0(q)}{\partial q} = \begin{bmatrix}J_{v,O_i,1} & \cdots & J_{v,O_i,i} & 0_{3\times(n-i)}\end{bmatrix}$ where $J_{v,O_i,j} = z_{j-1}^0 \times (O_i^0 - O_{j-1}^0)$ for revolute, $J_{v,O_i,j} = z_{j-1}^0$ for prismatic.

Spline Interpolation: Find $q : \mathbb{R} \mapsto \mathbb{R}^n \in \mathcal{C}^2$ such that $q(0) = q^0$ and $q(t_i) = q_i$. Each segment is $P_i(t) = a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0^i, t \in [t_i, t_{i+1}]$ for $i = 0, \ldots, N-1$, total $4N$ unknowns in $\mathbb{R}^n$, with constraints:

1. Interpolation: $P_i(t_i) = q^i$ for $i = 0, 1, \ldots, N-1$ and $P_{N-1}(t_N) = q^N$.
2. Twice-differentiability: $P_i(t_{i+1}) = P_{i+1}(t_{i+1})$, $\dot{P}_i(t_{i+1}) = \dot{P}_{i+1}(t_{i+1})$, $\ddot{P}_i(t_{i+1}) = \ddot{P}_{i+1}(t_{i+1})$ for $i = 0, 1, \ldots, N-2$.
3. Endpoint acceleration: $\ddot{P}_0(t_0) = \ddot{P}_{N-1}(t_N) = 0$.

5 Robot Modelling

Euler-Lagrange: For Lagrangian $\mathcal{L} = T - \mathcal{U}$, n degrees of freedom ($n = 3N - l$ for l constraints and N particles in 3 dimensions), generalized coordinates q:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right) - \frac{\partial \mathcal{L}}{\partial q_j} = \tau_j \text{ where } \tau_j = \sum_{i=1}^{N}(f_i^l)^T\frac{\partial r_i}{\partial q_j} \text{ for applied forces } f^l.$$

Holonomic Constraints: Expressed as $g(r_1, \ldots, r_N) = 0 \in \mathbb{R}^l$, assuming independence: rank $\left(\frac{\partial g}{\partial r}\right) = l$.

Virtual Displacements: $\delta r = \begin{bmatrix}\delta r_1^T & \cdots & \delta r_N^T\end{bmatrix}^T \in \mathbb{R}^{3N}$ must satisfy $\frac{\partial g}{\partial r}\delta r = 0$. All possible virtual displacements are described by $\delta r = \frac{\partial r}{\partial q}\mathrm{d}q$.

Canonical Robot Model: $D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau$ where:

1. $D(q) = \sum_{i=1}^n \left[m_i J_{v_i}^T(q)J_{v_i}(q) + J_{w_i}^T(q)R_i^0\bar{I}_i(R_i^0)^T J_{w_i}^T(q)\right]$
2. $[C(q,\dot{q})]_{kj} = \sum_{i=1}^n c_{ijk}(q)\dot{q}_i$ where $c_{ijk} = \dfrac{1}{2}\left(\dfrac{\partial d_{kj}}{\partial q_i} + \dfrac{\partial d_{ki}}{\partial q_j} - \dfrac{\partial d_{ij}}{\partial q_k}\right)$
3. $G(q) = \nabla_q\mathcal{U}(q)$ where $\mathcal{U}(q) = -\sum_{i=1}^n m_i\bar{g}^T r_i^0(q)$ is the potential energy.

for body-fixed inertias $\bar{I}_i$, COMs r_i^0, gravity $\bar{g} = \begin{bmatrix}0 & 0 & -g\end{bmatrix}^T$. For non-constant inertias transform as $I_i = R_i^0\bar{I}_i(R_i^0)^T$.

Kinetic Energy: $T = \displaystyle\sum_{i=1}^{N}\frac{1}{2}m_i\|\dot{r}_i\|^2 + \frac{1}{2}(w_i^0)^T R_i^0\bar{I}_i(R_i^0)^T w_i^0 = \frac{1}{2}\dot{q}^T D(q)\dot{q}$

Partial Jacobians:

$$J_{v_i} = \begin{cases}\begin{bmatrix}z_0^0 \times (r_i^0 - O_0^0) & \cdots & z_{i-1}^0 \times (r_i^0 - O_{i-1}^0) & 0_{3\times(n-i)}\end{bmatrix} & \text{revolute} \\ \begin{bmatrix}z_0^0 & z_1^0 & \cdots & z_{i-1}^0 & 0_{3\times(n-i)}\end{bmatrix} & \text{prismatic}\end{cases}$$

$$J_{w_i} = \begin{bmatrix}\rho_1 z_0^0 & \rho_2 z_1^0 & \cdots & \rho_i z_{i-1}^0 & 0_{3\times(n-i)}\end{bmatrix}$$

6 Lyapunov Theory

Consider general nonlinear systems $\dot{x} = f(x), x \in \mathbb{R}^n, f : \mathbb{R}^n \mapsto \mathbb{R}^n \in \mathcal{C}^1$, with equilibria $\bar{x} \in \mathbb{R}^n$ where $f(\bar{x}) = 0$.

Lyapunov Stability: $\forall\varepsilon > 0, \exists\delta > 0$ s.t. $\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon, \forall t \geq 0$, i.e. we can stay arbitrarily close to the equilibrium forever by starting close enough to the equilibrium.

Asymptotic Stability: Stable and $\exists\delta_0 > 0$ s.t. $\|x(0)\| < \delta_0 \implies \lim\limits_{t\to\infty} x(t) = 0$, i.e. solutions converge to the equilibrium within some radius.

Definiteness: $U : \mathbb{R}^n \mapsto \mathbb{R}$ is positive definite at $x = 0$ if $U(0) = 0$ and $x \neq 0 \implies U(x) > 0$; positive semidefinite if $U(x) = 0$ and $U(x) \geq 0, \forall x$.

Lyapunov's Theorem: If there exists a Lyapunov function $V : \mathbb{R}^n \mapsto \mathbb{R} \in \mathcal{C}^1$ which is positive definite at $\bar{x}$, then if $\dot{V}(x) = \frac{\partial V}{\partial x}f(x)$ is negative semidefinite, $\bar{x}$ is stable. If $\dot{V}(x)$ is negative definite, then $\bar{x}$ is asymptotically stable.

LaSalle Invariance Principle: If V is positive definite at $\bar{x}$, and $\dot{V}(x) = \dfrac{\partial V}{\partial x}f(x) \leq 0$, then $\lim\limits_{t\to\infty}\dot{V}(x(t)) = 0$. If $\dot{V}(x) \equiv 0, \forall t \implies x(t) \equiv \bar{x}, \forall t$, i.e. the only way to have $\dot{V} = 0$ for all time is to be at the equilibrium, then $\bar{x}$ is asymptotically stable.

7 Robot Control

Simplified Motor Model: $J\ddot{\theta}_m + B\dot{\theta}_m = u - \tau_l$ for voltage u, load τ_l.

Augmented Robot Model: $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B(q)\dot{q} + G(q) = u$ for motor voltages u, back-EMF $B(q)$ (positive definite), $M(q) = D(q) + J$ where $J = \operatorname{diag}\{\, J_{m_i} \,\}$. Note $\dot{M}(q,\dot{q}) - 2C(q,\dot{q})$ is skew-symmetric.

Decentralized (Independent Joint) Control: $u = J\ddot{q}^r + B\dot{q}^r + K_p e + K_d\dot{e}$ for $K_p, K_d > 0$, where $e(t) = q^r(t) - q(t)$, assuming each $\theta_m = q \in \mathbb{R}$. Each joint has dynamics $\ddot{e} + \frac{B}{J}\dot{e} = \ddot{q}^r + \frac{B}{J}\dot{q}^r - \frac{1}{J}u$, closed-loop dynamics $\ddot{e} + \left(\frac{B}{J} + \frac{K_D}{J}\right)\dot{e} + \frac{K_p}{J}e = 0$. Treats robot dynamics as disturbances.

Feedback Linearization: $u = M(q)v + C(q,\dot{q})\dot{q} + B(q)\dot{q} + G(q)$ where $v = \ddot{q}^r + K_p\tilde{q} + K_d\dot{\tilde{q}}$, for error $\tilde{q} = q^r - q$ and diagonal, positive definite K_p, K_d. Closed-loop dynamics $\ddot{q} = v \iff \ddot{\tilde{q}} + K_d\dot{\tilde{q}} + K_p\tilde{q} = 0$. This requires perfect knowledge of system parameters so does not work well in practice.

PD Control With Gravity Compensation: $u = K_p\tilde{q} + K_d\dot{\tilde{q}} + G(q)$ where $\tilde{q} = q^r - q$ and reference q^r is constant, for positive definite K_p, K_d. Closed-loop dynamics $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B(q)\dot{q} = K_p\tilde{q} + K_d\dot{\tilde{q}}$. Lyapunov function $V(q,\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}(q - q^r)^T K_p(q - q^r) \implies \dot{V} = -\dot{q}^T(B(q) + K_d)\dot{q}$. Can only track a constant reference but only requires knowledge of gravity terms.

Passivity-Based Control: Define $r = \dot{\tilde{q}} + \Lambda\tilde{q}$ where $\tilde{q} = q^r - q$ and Λ diagonal, $u = M(q)(\ddot{q}^r + \Lambda\dot{\tilde{q}}) + C(q,\dot{q})(\dot{q}^r + \Lambda\tilde{q}) + B(q)\dot{q} + G(q) + K(\dot{\tilde{q}} + \Lambda\tilde{q})$ with closed-loop dynamics $M(q)\dot{r} + (C(q,\dot{q}) + K)r = 0$. Use Lyapunov function $V = \frac{1}{2}r^T M(q)r + \tilde{q}^T\Lambda K\tilde{q} \implies \dot{V} = -\dot{\tilde{q}}^T K\dot{\tilde{q}} - \tilde{q}^T\Lambda K\Lambda\tilde{q}$.

Linear Parametrization: $M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B(q)\dot{q} + G(q) = Y(q,\dot{q},\ddot{q})\Phi$ where $Y(q,\dot{q},\ddot{q})$ is a known regressor matrix for the structure of the robot, and Φ is an unknown vector containing all the physical parameters of the robot.

Adaptive Passivity-Based Control: $u = -Kr + Y\hat{\Theta}$ where $r = \dot{\tilde{q}} + \Lambda\tilde{q}, v = \dot{q}^r - \Lambda\tilde{q}, a = \ddot{q}^r - \Lambda\dot{\tilde{q}} = \dot{v}$, with error $\tilde{q} = q - q^r$ (reversed!), for positive definite K, Λ, Γ, regressor Y such that $M(q)a + C(q,\dot{q})v + B(q)\dot{q} + G(q) = Y(q,\dot{q},a,v)\Theta$, and parameter estimate $\hat{\Theta}$, with adaptation law $\dot{\hat{\Theta}} = -\Gamma Y^T r$ for learning rate Γ. Closed-loop dynamics $M(q)\dot{r} + (C(q,\dot{q}) + K)r = Y(q,\dot{q},a,v)\tilde{\Theta}$ (substitute using $\dot{q} = r + v, \ddot{q} = \dot{r} + a$) where $\tilde{\Theta} = \hat{\Theta} - \Theta$. Use Lyapunov function $V = \frac{1}{2}r^T M(q)r + \tilde{q}^T\Lambda K\tilde{q} + \frac{1}{2}\tilde{\Theta}^T\Gamma^{-1}\tilde{\Theta} \implies \dot{V} = -\dot{\tilde{q}}^T K\dot{\tilde{q}} - \tilde{q}^T\Lambda K\Lambda\tilde{q}$.

This tracks the reference but does not guarantee asymptotic convergence of $\hat{\Theta}$, which requires a persistence of excitation condition. This controller does not require knowledge of system parameters but results in aggressive transient behaviour.

8 Useful Identities

$$\sin\left(x + \tfrac{\pi}{2}\right) = \cos(x) \quad \sin\left(x - \tfrac{\pi}{2}\right) = -\cos(x) \quad \sin\left(\tfrac{3\pi}{2} - x\right) = -\cos(x)$$

$$\cos\left(x + \tfrac{\pi}{2}\right) = -\sin(x) \quad \cos\left(x - \tfrac{\pi}{2}\right) = \sin(x) \quad \cos\left(\tfrac{3\pi}{2} - x\right) = \sin(x)$$

$$\sin(x + y) = \sin x\cos y + \cos x\sin y \quad \sin(x - y) = \sin x\cos y - \cos x\sin y$$

$$\cos(x + y) = \cos x\cos y - \sin x\sin y \quad \cos(x - y) = \cos x\cos y + \sin x\sin y$$

$$\operatorname{atan2}(-y, x) = -\operatorname{atan2}(y, x) \qquad \operatorname{atan2}(y, -x) = \pi - \operatorname{atan2}(y, x)$$

$$\forall a > 0,\ \operatorname{atan2}(ay, ax) = \operatorname{atan2}(y, x)$$

$$\operatorname{atan2}\left(x, \sqrt{1 - x^2}\right) = \sin^{-1} x \qquad \operatorname{atan2}\left(\pm\sqrt{1 - x^2}, x\right) = \pm\cos^{-1} x$$

$$\operatorname{atan2}\left(x, -\sqrt{1 - x^2}\right) = \begin{cases}\pi - \sin^{-1} x & x > 0 \\ -\pi - \sin^{-1}(x) & x < 0\end{cases}$$

$$\nabla_x f(x) = \left(\frac{\partial f(x)}{\partial x}\right)^T \qquad \frac{\partial\|x\|}{\partial x} = \frac{1}{\|x\|}x^T$$

$$v_1 \cdot v_2 = \|v_1\|\|v_2\|\cos\theta \qquad v_1 \times v_2 = \|v_1\|\|v_2\|\sin(\theta)\hat{n}$$