Coordinate Transformation

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

$R_x=\begin{bmatrix}1&0&0\\0&c&-s\\0&s&c\end{bmatrix}$ • $R_1R_2 \neq R_2R_1$ unless are about same axis

$R_y=\begin{bmatrix}c&0&s\\0&1&0\\-s&0&c\end{bmatrix}$ • $R^T=R^{-1}$ • Vector norms are invariant under rotation

$R_z=\begin{bmatrix}c&-s&0\\s&c&0\\0&0&1\end{bmatrix}$

Properties:

$R(\theta_1)R(\theta_2)=R(\theta_1+\theta_2)$

$R(-\theta)=R(\theta)^T=R(\theta)^{-1}$

$\|R(\theta)\,x\|=\|x\|$

Homogeneous Transformation

2D $T\vec{x}=\begin{bmatrix}R & \begin{matrix}T_x\\T_y\end{matrix}\\ 0\,0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\1\end{bmatrix}$

3D $T\vec{x}=\begin{bmatrix}R & \begin{matrix}T_x\\T_y\\T_z\end{matrix}\\ 0\,0\,0 & 1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}$

- Distance of 2 pts do not change after being transformed
- Composition of rigid transform are still rigid transform
- 2D, area of any triangle does not change
- 3D, area of tetrahedron does not change

${}^AT_B$: Homogeneous transform from B to A

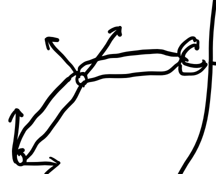
${}^AT_C={}^AT_B\,{}^BT_C$

Translation

$({}^AT_B)^{-1}={}^BT_A=\begin{bmatrix}({}^AR_B)^T & -({}^AR_B)^T\,{}^AP_B\\ 0 & 1\end{bmatrix}$

Example in 2D RR

$^{Base}T_{EE}={}^BT_{joint\,1}\,{}^{joint\,1}T_{joint\,2}\,\vec{x}$



$\begin{bmatrix}\cos\theta_1 & -\sin\theta_1 & L_1\cos\theta_1\\ \sin\theta_1 & \cos\theta_1 & L_1\sin\theta_1\\ 0&0&1\end{bmatrix}\begin{bmatrix}\cos\theta_2 & -\sin\theta_2 & L_2\cos\theta_2\\ \sin\theta_2 & \cos\theta_2 & L_2\sin\theta_2\\ 0&0&1\end{bmatrix}\vec{x}$

$X_{ee}=L_1\cos\theta_1+L_2\cos(\theta_1+\theta_2)$

$Y_{ee}=L_1\sin\theta_1+L_2\sin(\theta_1+\theta_2)$

IK: Task space → Joint space

① Do FK on the robot

② Set up equation between final homogeneous matrix & Desired position + orientation

Example $X=L_1\cos\theta_1+L_2\cos(\theta_1+\theta_2)$

$y=L_1\sin\theta_1+L_2\sin(\theta_1+\theta_2)$

$\tan\theta_2=\frac{y}{x}=\frac{r\sin\theta_2}{r\cos\theta_2}=\frac{\sin\theta_2}{\cos\theta_2}$

$x^2+y^2=L_1^2+L_2^2+2L_1L_2\cos(\theta_2)$

$\cos\theta_2=\frac{x^2+y^2-(L_1^2+L_2^2)}{2L_1L_2}$ $\sin\theta_2=\pm\sqrt{1-\cos^2\theta_2}$

$\theta_2=\operatorname{atan2}(\pm\sqrt{1-\cos^2\theta_2},\ \cos\theta_2)$

$\begin{bmatrix}x\\y\end{bmatrix}=R(\theta_1)\begin{bmatrix}L_1\\0\end{bmatrix}+R(\theta_1)R(\theta_2)\begin{bmatrix}L_2\\0\end{bmatrix}$

$=R(\theta_1)\left(\begin{bmatrix}L_1\\0\end{bmatrix}+R(\theta_2)\begin{bmatrix}L_2\\0\end{bmatrix}\right)$

$\begin{bmatrix}x\\y\end{bmatrix}=R(\theta_1)\begin{bmatrix}a\\b\end{bmatrix}$ $a=L_1+L_2\cos\theta_2$

$b=L_2\sin\theta_2$

$B=\operatorname{atan2}(b,a)$

$\alpha=\operatorname{atan2}(y,x)$

$\theta_1+B=\alpha$

$\theta=\alpha-B$

$\theta_1=\operatorname{atan2}(y,x)-\operatorname{atan2}(L_2\sin\theta_2,\ L_1+L_2\cos\theta_2)$

Jacobian joint → ee

$\dot{X}=J(\theta)\,\dot{\theta}$

$J(q)=\begin{bmatrix}\frac{\partial x}{\partial q_1}&\frac{\partial x}{\partial q_2}&\frac{\partial x}{\partial q_3}\\ \frac{\partial y}{\partial q_1}&\frac{\partial y}{\partial q_2}&\frac{\partial y}{\partial q_3}\\ \frac{\partial z}{\partial q_1}&\frac{\partial z}{\partial q_2}&\frac{\partial z}{\partial q_3}\end{bmatrix}$ Specific to configuration

PID Controller

$e(t)=X_{set}-X$

$$u(t)=K_pe(t)+K_i\int_0^t e(\tau)d\tau+K_d\frac{de(t)}{dt}$$

(1) Proportional (P)
- Reacts to current error
- large error → large correction

Trade offs: Responsiveness vs. Oscillation/overshoot

(2) Integral (I)
- Reacts to accumulated past error
- Eliminate steady state offset (friction, wind drag etc.)

Accuracy vs. slower response

(3) Derivative (D)
- Reacts to rate of change of error
- Damping to avoid overshoot
- Error suddenly growing → respond immediately
- Error dropping too fast → damp the correction to prevent overshoot

Stability / Sensitivity to noise

C-Space

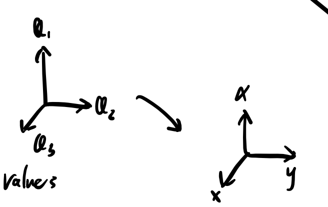
SO(n) ⇒ group of rotation matrices

SE(n) ⇒ set of (n+1)×(n+1) homogeneous matrix

Dof $\dim(SO(2))=1$ $\dim(SO(3))=3$

$SE(2)=3$ $SE(3)=6$

FK: Joint space → Operation space



Comparison between spaces

① Joint space → Space of all possible joint values

② Configuration space → ≈ Joint space but people sometimes include base position

③ Task space (operational space) → Space of all possible poses (position + orientation) of the end-effector in the world frame

④ Workspace → Subset of the Task space that the robot can physically reach (Determined by joint limits, mechanical constraints)

PRM

① Randomly generate robot configurations and add as nodes to the roadmap if collision-free

② Attempt to connect nearby nodes with local paths

To perform queries, add q_{init} and q_{goal} to the roadmap using local planar

OBPRM When we sample $\alpha(i)$ in an obstacle region, sample a number of random directions away from $\alpha(i)$ to find free configuration q

Guards & Connectors
- Generate a new point P
  - → Cannot see existing guard P = new guard
  - → See 1 guard discard P
  - → See 2+ guards, add it as a connector

RRT

RRT: Probabilistically complete, not optimal

RRT*: Probabilistically complete & optimal

1. Initialize tree $T = \{q_{start}\}$
2. For $i = 1$ to N:
   - Sample a random configuration $q_{rand} \in C_{free}$
   - Find the nearest node $q_{near} = Nearest(T, q_{rand})$
   - Move from q_{near} toward q_{rand} by step size δ. $\rightarrow q_{new} = q_{near} + \delta \frac{(q_{rand} - q_{near})}{\| q_{rand} - q_{near} \|}$
   - If collision free, add q_{new} to T and a edge $q_{near} \rightarrow q_{new}$
3. If q_{new} is close enough to q_{goal}, return the path

RRT*

After q_{new} is added:

- Find all existing nodes within radius r
- Choose a parent from these that gives the lowest total cost

$$q_{parent} = \underset{q_i \in Near}{\mathrm{argmin}} \left[C(q_i) + Cost(q_i, q_{new}) \right]$$

- Add edge $q_{parent} \rightarrow q_{new}$ to the tree
- Rewire $\Rightarrow$ For each $q_i \in Near$

  if going through q_{new} lowers its cost $C(q_{new}) + Cost(q_{new}, q_i) < C(q_i)$

  Set $parent(q_i) = q_{new}$

Camera Calibration

Camera $\rightarrow$ pixel Coordinates

$$u = f_x \frac{x_c}{z_c} + O_x$$

$$v = f_y \frac{y_c}{z_c} + O_y$$

(O_x, O_y) = Principle point (coordinates of image center)

f = focal length

$m_x\ m_y$ = Pixel/length along each axis

$x_c\ y_c\ z_c$ = 3D point coordinate in camera frame

$f_x f_y = (m_x f, m_y f)$

Extrinsic Matrix

C_w: Camera $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$ in World

$$X_c = R(X_w - C_w) = R X_w + t, \quad t = -R C_w$$

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{array}{l} \leftarrow \text{Direction of } \hat{x}_c \text{ in world coordinates} \\ \leftarrow \text{Direction of } \hat{y}_c \text{ in world coordinates} \\ \leftarrow \text{Direction of } \hat{z}_c \text{ in world coordinates} \end{array}$$

$$M_{ext} = \begin{bmatrix} R & t \\ 0\,0\,0 & 1 \end{bmatrix} \qquad X_c = M_{ext} X_w$$

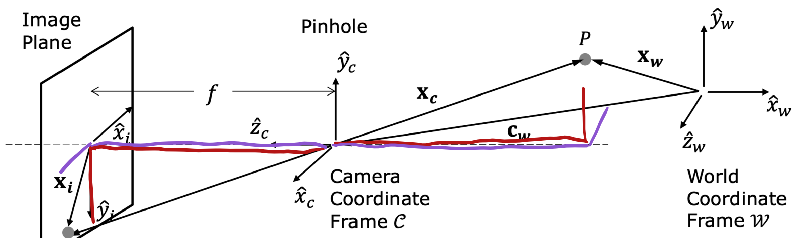
Intrinsic Matrix

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \equiv \begin{bmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{bmatrix} \equiv \begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & O_x & 0 \\ 0 & f_y & O_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

Where $(u, v) = (\tilde{u}/\tilde{w}, \tilde{v}/\tilde{w})$

Full projection matrix P: World $\rightarrow$ image plane

$$\tilde{u} = M_{int} M_{ext} X_w$$



Least-Squares for Camera Calibration

$$\min_{K, \{R_j t_j\}} J(K, \{R_j, t_j\}) = \sum_{j=1}^{M} \sum_{i=1}^{N_j} \left\| \begin{bmatrix} \hat{u}_{ij} - u_{ij} \\ \hat{v}_{ij} - v_{ij} \end{bmatrix} \right\|^2$$

K: intrinsic camera params

$R_j\ t_j$: extrinsic rotation/translation for image j

M: total number of images

N_j: Number of detected points in image j

$$\begin{bmatrix} \hat{u}_{ij} \\ \hat{v}_{ij} \end{bmatrix} = K \left[R_j \mid t_j \right] \begin{bmatrix} P_w^i \\ 1 \end{bmatrix} \quad \text{For point } i \text{ on image } j$$

Position & Impedence Control

Mass-spring damper System (natural response to equilibrium)

$$m\ddot{x} + c\dot{x} + kx = 0$$

W external input

$$m\ddot{x} + c\dot{x} + kx = u(t)$$

Damping ratio

$$L = \frac{c}{2\sqrt{mk}} = \frac{K_d}{2\sqrt{mK_p}}$$

$L < 1$	Underdamped
$L = 1$	Critically damped
$L > 1$	Overdamped

Implementation of Impedance Control

1. Measure current end effector pose $x, \dot{x}$ $\qquad W_n = \sqrt{\frac{K_p}{m}}$
2. Compute desired response using

$$F_{cmd} = M_d(\ddot{x} - \ddot{x}_d) + D_d(\dot{x} - \dot{x}_d) + K_d(x - x_d)$$

matrices 6x6 — Vectors 6x1

3. Map desired force cmd to joint torques: $F_{cmd} \in \mathbb{R}^6$ = Force xyz torque xyz

$$\tau = J^T F_{cmd}$$

Admittance Ctrl: Measure external force & command corresponding displacement/velocity

DFS	$O(V+E)$	$O(V)$
BFS	$O(V+E)$	$O(V)$
Dijkstra	$O((V+E)\log V)$	$O(V)$
A*	$O(b^d)$	$O(b^d)$

b = branching factor

d = Depth of optimal solution

With $h(n) \le h^*(n)$ = True Cost