Spatial Velocity

$$V_{q_a} = \dot{q}_a(t) = \hat{V}^s_{ab}\, q_a = \dot{g}_{ab}(\cancel{t})\, g^{-1}_{ab}(t)\, q_a$$

$$\hat{V}^s_{ab} = \hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \dot{R}R^T & -\dot{R}R^T p + \dot{p} \\ 0 & 0 \end{bmatrix}$$

$$V^s_{ab} = \xi = \begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} -\dot{R}R^T p + \dot{p} \\ (\dot{R}R^T)^\vee \end{bmatrix}$$

A point attached to the body moving around, determine the rate of position change respect to the spatial frame.

Body Velocity

$$V_{q_b} = g^{-1}_{ab}(t)\, V_{q_a} = g^{-1}_{ab}(t)\, \dot{g}_{ab}(\cancel{t})\, q_b$$

$$\hat{V}^b_{ab} = \hat{\xi} = \begin{bmatrix} \hat{\omega} & v \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} R^T\dot{R} & R^T\dot{p} \\ 0 & 0 \end{bmatrix}$$

$$V^b_{ab} = \xi = \begin{bmatrix} R^T\dot{p} \\ (R^T\dot{R})^\vee \end{bmatrix}$$

Still the same time derivative (velocity) of point q relative to spatial frame, but expressed in body frame

Solution Tips

1. when determine spatial/body velocities,

① get $g_{ab}(t)$ and $g^{-1}_{ab}(t)$ first;

② then, ☆ derivative $\underline{\dot{g}_{ab}(t)}$;

note that $y = \sin\theta(t)$ $\quad \frac{dy}{dt} = \frac{dy}{d\theta}\frac{d\theta}{dt} = (\cos\theta)\underline{\dot{\theta}}$

2. $$\hat{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}$$

3. Circle method to validate

That's how joints affect frames.

非 robot arm/joints, 直接看, 注意拆分 rotate & translate

Adjoint Transformation

$$Ad_{g_{ab}} = \begin{bmatrix} R_{ab} & \hat{p}_{ab}R_{ab} \\ 0 & R_{ab} \end{bmatrix}$$

$$Ad_{g^{-1}_{ab}} = \begin{bmatrix} R^T_{ab} & -R^T_{ab}\hat{p}_{ab} \\ 0 & R^T_{ab} \end{bmatrix}$$

$$V^a_{ab} = Ad_{g_{ab}} V^b_{ab} = \begin{bmatrix} R_{ab} & \hat{p}_{ab}R_{ab} \\ 0 & R_{ab} \end{bmatrix} \begin{bmatrix} v^b_{ab} \\ w^b_{ab} \end{bmatrix} = \begin{bmatrix} Rv + \hat{p}Rw \\ Rw \end{bmatrix}$$

$$\hat{V}^a_{ab} = g_{ab}\, \hat{V}^b_{ab}\, g^{-1}_{ab}$$

$$V^a_{ac} = V^a_{ab} + Ad_{g_{ab}} V^b_{bc}$$

$$V^c_{ac} = Ad_{g^{-1}_{bc}} V^b_{ab} + V^c_{bc}$$

$$(Ad_g)^{-1} = Ad_{g^{-1}}$$

$$Ad_{g_1 g_2} = Ad_{g_1} Ad_{g_2}$$

Useful formula

$$\hat{\xi}_a = g_{ab}\, \hat{\xi}_b\, g^{-1}_{ab}$$

$$\xi_a = Ad_{g_{ab}}\, \xi_b$$

$$g^{-1}_{ab} = \begin{bmatrix} R^T_{ab} & -R^T_{ab}p \\ 0 & 1 \end{bmatrix}$$

Spatial Jacobian

$$V_{ST}^{S} = J_{ST}^{S}(\theta)\,\dot{\theta}$$

$$J_{ST}^{S}(\theta) = \left[\left(\frac{\partial g_{ST}}{\partial \theta_1} \cdot g_{ST}^{-1} \right)^{\vee}, \ \dots, \ \left(\frac{\partial g_{ST}}{\partial \theta_n} \cdot g_{ST}^{-1} \right)^{\vee} \right]$$

$$= [\ \xi_1,\ \xi_2',\ \dots\ \xi_n'\ ]$$

$$\xi_i' = Ad_{(e^{\hat{\xi}_1\theta_1} \dots e^{\hat{\xi}_{i-1}\theta_{i-1}})}\,\xi_i$$

Convert

$$J_{ST}^{S}(\theta) = Ad_{g_{ST}(\theta)}\, J_{ST}^{T}(\theta)$$

Tips

Tool frame is fixed to the tool, it will change when tool changes.

Body Jacobian

$$V_{ST}^{T} = J_{ST}^{T}(\theta)\,\dot{\theta}$$

$$J_{ST}^{T}(\theta) = \left[\left(g_{ST}^{-1} \cdot \frac{\partial g_{ST}}{\partial \theta_1} \right)^{\vee}, \ \dots \ \left(g_{ST}^{-1} \frac{\partial g_{ST}}{\partial \theta_n} \right)^{\vee} \right] = [\ \xi_1^{+},\ \xi_2^{+},\ \dots\ \xi_n^{+}\ ]$$

$$\xi_i^{+} = Ad_{(e^{\hat{\xi}_{i+1}\theta_{i+1}} \dots e^{\hat{\xi}_n\theta_n} g_{ST}(0))}\,\xi_i$$

Singularity

rank (J) < joint numbers

① 2 collinear revolute joints
② 3 parallel coplanar r joints
③ 4 intersected revolute joints

1. Rotation R is achieved by intrinsic rotations about the z, y, x
   The spatial Jacobian for the rotation has a singularity when $\theta_y = \frac{\pi}{2}$

2. Any rotation about the linearly independent axes will also have a singularity.
   when rotate about w_3, w_1 will not change, only rotate w_3.

3. When 4 revolute joint axes are coplanar, any six degree of freedom manipulator is at a singular configuration.

4. manipulability measure: how close we're encountering singularities.

$$\mu(\theta) = \prod_{i=1} \sigma_i(\theta)$$ singular value of J^S

J^S singular $\Rightarrow$ non-trival null space (K dimensional) $\Rightarrow (J^S)^T J^S$ singular, null space k-d

$\Downarrow$

at least one eigenvale $= 0$

$\Downarrow$

$\sigma_i = 0$

计算 singular value.
① Calculate A^TA
② 求 A^TA eigenvalue
③ singular value = A^TA 的非负 eigenvalue
降序.

Dynamics

Steps:

① Pick generalized coordinates q ← position / angle

② Find KE, PE.

③ Take Lagrangian $L = T - V$ and find its derivatives.

④ $\tau = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q}$

③

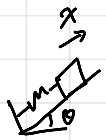
$q = [\theta_1\ \theta_2\ \theta_3]^T$

$L = 0 + \frac{1}{2} I \dot{\theta}_1^2 + \frac{1}{2} m_2 V_2^2 + \frac{1}{2} m_2 \dot{\theta}_2^2 + \frac{1}{2} m_3 V_3^2 + \frac{1}{2} m_3 \dot{\theta}_3^2$ $\quad V = 0$

$V_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = (L \cdot \dot{\cos\theta_1})^2 + (L \cdot \dot{\sin\theta_1})^2$

$= (-L\sin\theta_1 \cdot \dot{\theta}_1)^2 + (L\cos\theta_1\, \dot{\theta}_1)^2 = L^2 \dot{\theta}_1^2$

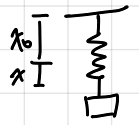
②



$q = [x]$

$L = \frac{1}{2} m \dot{x}^2$

$V = \frac{1}{2} k x^2 + mgx \cdot \sin\theta$

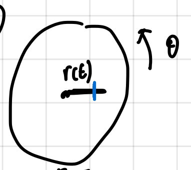


$q = [x]$

$L = \frac{1}{2} m \dot{x}^2$

$V = \frac{1}{2} k x^2 - mgx$

④



$q = \begin{bmatrix} r \\ \theta \end{bmatrix}$

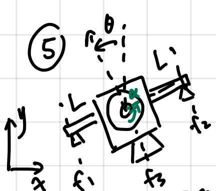
$L = 0 + \frac{1}{2} I \dot{\theta}_1^2 + \frac{1}{2} m V^2$

$V = \dot{x}^2 + \dot{y}^2 = (r(t)\dot{\cos\theta})^2 + (r(t)\sin\theta)^2$

$= (\dot{r}\cos\theta - r(t)\sin\theta\,\dot{\theta})^2 + (\dot{r}\sin\theta + r(t)\cos\theta\,\dot{\theta})^2$

$= \dot{r}^2 + r(t)^2\dot{\theta}^2$

⑤



$q = \begin{bmatrix} x \\ y \\ \theta \\ \alpha \end{bmatrix}$

$L = \frac{1}{2} m_b (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_b \dot{\theta}^2 + \frac{1}{2} m_r (\dot{x}^2 + \dot{y}^2) + \frac{1}{2} I_r (\dot{\theta} + \dot{\alpha})^2$

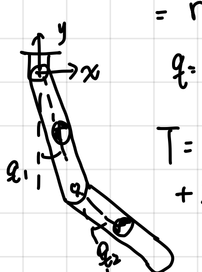
$\tau_x = -(f_1 + f_2 + f_3)\sin\theta$

$\tau_y = (f_1 + f_2 + f_3)\cos\theta$

$\tau_\theta = (f_2 - f_1)L - \tau$

$\tau_\alpha = \tau$

⑥



$q = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$

$T = \frac{1}{2} m V_1^2 + \frac{1}{2} I_1 \dot{q}_1^2 + \frac{1}{2} m V_2^2 + \frac{1}{2} I_2 (\dot{q}_1 + \dot{q}_2)^2$

$M = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix}$

$V_1^2 = (-\frac{L}{2}\dot{q}_1 \sin q_1)^2 + (\frac{L}{2}\dot{q}_1 \cos q_1)^2 = \frac{L^2}{4}\dot{q}_1^2$

$V_2^2 = (L\dot{\sin q_1} + \frac{L}{2}\dot{\sin(q_1+q_2)})^2 + (-L\dot{\cos q_1} - \frac{L}{2}\dot{\cos(q_1+q_2)})^2$

$M_{box} = \begin{bmatrix} \frac{m}{12}(w^2+h^2) & 0 & 0 \\ 0 & \frac{m}{12}(l^2+h^2) & 0 \\ 0 & 0 & \frac{m}{12}(l^2+w^2) \end{bmatrix}$

$V = -\frac{L}{2} mg \cdot \cos q_1 - (L \cdot \cos q_1 + \frac{L}{2}\cos(q_1+q_2))$

$T = \frac{1}{2}[v^T\ w^T]\begin{bmatrix} mI_3 & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} v \\ w \end{bmatrix}$

$= \frac{1}{2}[v^T\ w^T]\begin{bmatrix} mv \\ Iw \end{bmatrix} = \underbrace{\frac{1}{2} m v^T v}_{\text{trans KE}} + \underbrace{\frac{1}{2} w^T I w}_{\text{rotation KE}}$

$V_g = mgh \qquad V_s = \frac{1}{2} k x^2$

☆ $I_{total} = I_{rotation} + m \cdot l^2$

ROS: provide service expected from OS

- Query the camera sensing loop for a single image
- Use a Vision algo to compute the location of obj
- Compute joint angles to move the arm to the location
- Send position commands to each joint control loops
- Signal the gripper control loop to grab

each control loop = node

rospack find [package name]
catkin_make
catkin_create_pkg [name]
rosrun [pkg. name] [exe_name]

node $\xrightarrow{msg}$ topic → node

rosnode list
rosnode info /[name]

Rqt
/node $\xrightarrow{/topic}$ /node

rosservice call [name] [argv]

```
rospy.init_node('name')
try:
    talker()
def talker():
    pub = rospy.publisher('topic', [DataTy], qsize=10)
    r = rospy.rate(10)
    while not down:
        pubstring = 'string' (rospy.get_time()) < pub.publish(pubstring)
def listener():
    rospy.subscriber('topic', [DataTy], callback)
def callback(message):
    print(msg)
```

---

request	float32	geometry_msgs/Pose[]
response	string	

Stability: $\det(A-\lambda I)=0$

$$\dot{x} = Ax(t)$$

if $re(\lambda(A)) < 0$, then $x(t) \to 0$ as $t \to \infty$

equilibrium point: $\dot{x} = 0$

Controllability:

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$Q = [B \;\; AB \;\; A^2B \; \cdots \; A^{n-1}B]$$

$\text{rank}(Q) \geq n \Rightarrow$ completely controlable

Stabilizable?

$\lambda_i > 0$ $\underbrace{\text{rank}([A-\lambda_i I \;\; B])}_{\text{check}} \overset{?}{=} n$

Stabilize @ $u = -Kx$

$$\dot{x} = (A-BK)x$$

$$\det([A-BK] - \lambda I) = 0$$

$$\det([\;\cdots\;]) = 0 \to \lambda^2 + a\lambda + c = 0$$

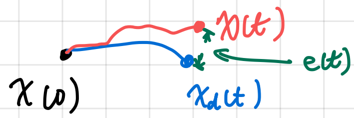
$\underline{a>0 \;\&\; c>0}$ ✓ n

---

error $e(t) = x(t) - x_d(t)$ how far off we are from our desired point.

Choose Control Input

$$\dot{e} = Ae(t) + Bu + (Ax_d(t) - \dot{x}_d)$$



choose u so that $e(t)$ goes to 0, $u(t) = -K_p e(t)$

$$\Rightarrow \dot{e} = Ae(t) + B(-K_p e(t)) + Ax_d(t) - \dot{x}_d$$

$$= \underline{(A-BK_p)}e(t) + Ax_d(t) - \dot{x}_d$$

eigenvalue $re(\lambda(A-BK_p)) < 0 \Rightarrow e(t) \to 0$ as $t \to \infty$

PID: P: K for spring

稳态误差

波动大

(阻尼) D: 消除波动

I: 消除稳态误差

Linearization

① $x = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

② $\dot{x} = \begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} x_2 \\ \text{formula} \end{bmatrix} = \begin{bmatrix} f_1(x,u) \\ f_2(x,u) \end{bmatrix}$

③ $f(x,u) = f(x_0, u_0) + \frac{\partial f}{\partial x_i}(x - x_0)\Big|_{\substack{x= \\ u=}} + \frac{\partial f}{\partial u}\Big|_{x=,\,u=}(u - u_0)$

$\dot{x} = f(x,u)$

④ $\underbrace{\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix}}_{\dot{x}} = \begin{bmatrix} \cdots & \cdots \\ \cdots & \cdots \end{bmatrix} \underbrace{\begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}}_{x} + \begin{bmatrix} \vdots \end{bmatrix} u$

$$\frac{\partial \vec{f}}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \cdots & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u \\ 0 & f_y & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix}$$

$$x_{pixel} = f_x \cdot \frac{X_c}{Z_c} + u$$

estimated g_{21} must satisfy $\underline{\text{epipolar constraints}}$

$g_{21} = \begin{bmatrix} R_{21} & T_{21} \\ 0 & 1 \end{bmatrix}$

$x_2^T E x_1 = 0$ where

$E = \hat{T}_{21} R_{21}$