Polynomial interpolation

~~[illegible]~~

Represent trajectory as linear combination of phase-dependent terms.

or $p(t) = C_n t^n + C_{n-1} t^{n-1} + C_{n-2} t^{n-2} \cdots C_0$

x/c (decision variable in QP)

" $[C_{1n}\ C_{1n-1} \cdots C_{10}\ C_{2n} \cdots C_{m1}\ C_{m0}]$

where

n is order of trajectory polynomial

m is # of pieces in traj

# of knot points $= m+1$

must be $2r-1$ at least, where r is ~~order~~ derivative to minimize

- Cubic hermite interpolation

$$p(t) = h_{00}(t)p_0 + h_{10}(t)m_0 + h_{01}(t)p_1 + h_{11}(t)m_1$$

"Hermite form" → start/stop constraints

slope/gradient at start/end points

- Legendre polynomial

orthogonal on $[-1,-1]$

i.e. $\int_{-1}^{1} P_m(\tau)P_n(t)d\tau = 0$ if $m \neq n$

n	$P_n(x)$
0	1
1	x
2	$\frac{1}{2}(3x^2-1)$
3	$\frac{1}{2}(5x^3-3x)$
$\vdots$	$\vdots$

In "Mixed Integer"

$k_r = 4,\ n_p = 9 \sim 15$

differential flatness

input space — waypoints

↓

trajectory generation ~~[illegible]~~ QP

↓

calculate $u : [F, \ddot{p}, \ddot{r}, \ddot{y}]$

↓

Invert allocation matrix to find rotor speeds

In this case, all equality constraints $Ax=b$

Constraints

Endpoint constraints

ex. $p_0(0) = [x_0\ y_0\ z_0\ \psi_0]$

$x_0 = c_x(0)^T t$

$$= \underbrace{[C_n\ C_{n-1} \cdots C_0]}_{1\times n} \underbrace{\begin{bmatrix} t^n = 0 \\ t^{n-1} = 0 \\ \vdots \\ t^0 = 1 \end{bmatrix}}_{n\times 1}$$

Note that by default, only start and end points are fixed. Every other knot point can move, as long as dynamics are consistent.

simple controller gain tuning strategy: increment by error between "observed" gain values ~~and~~ current.

ie. If you just look at the trajectory, what would you calculate gain to be?

continuity constraints

need $x_j(t) = x_{j+1}(t)$ at all knot points

↓

$x_j(t) - x_{j+1}(t) = 0$

$[t^n, t^{n-1}, \ldots, t^0]^T$

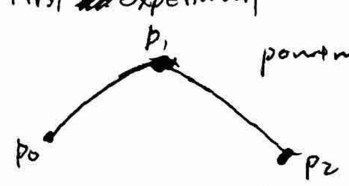
$c_j^T t - c_{j+1}^T t = 0$

$(c_j - c_{j+1})^T t = 0$

$t^T(c_j - c_{j+1}) = [0]$

$Ax = b$

- First experiment



point mass: minimum acceleration?

because that's where u is

Cost

$c^T H c$

$$H = \begin{bmatrix} H_1 & & \\ & \ddots & \\ & & H_m \end{bmatrix}$$

p $p(t) = c_3 t^3 + c_2 t^2 + c_1 t + c_0$

v $\dot{p}(t) = 3c_3 t^2 + 2c_2{}^2 + c_1 + 1$

ac $\ddot{p}(t) = 6c_3 t + 4c_2 + 1$

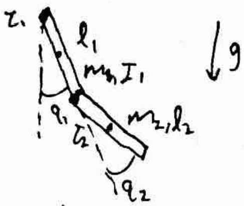
j $p^{(3)}(t) = 6c_3 + 4$

s $p^{(4)}(t) = 6$

higher and higher terms because:
- coefficients accumulated
- obviously high jerk means high cost for velocity ex.

$$H_i = \begin{array}{c} \begin{bmatrix} \ddots & - & \\ & & 0 \\ 0 & 0 & \end{bmatrix} \end{array} \begin{matrix} c_n \\ c_{n-1} \\ \vdots \\ c_0 \end{matrix}$$

Two-link arm / acrobot



$q = [\theta_1, \theta_2]$

$x = [q, \dot{q}]$

Target: $x = [\pi, 0, 0, 0]$

$x_1 = [l_1 s_1, -l_1 c_1]$

$x_2 = x_1 + \begin{bmatrix} l_2 s_{1+2} \\ -l_2 c_{1+2} \end{bmatrix}$

l_{c_1}; assume half
l_{c_2}
|
COM

To derive dynamics: write down equations for KE and PE.

$s_i = \sin(i)$
$c_i = \cos(i)$

Plug into Lagrangian

$$L = KE - PE$$

$$\frac{\partial L}{\partial q_i} - \frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} = 0$$

$I = ml^2$
const

$\cancel{H(q) = [I_1 + I_2 + m_2 l_1^2 + 2 m_2 l_1 l_{c_2}}$

Manipulator equation form

$$H(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = B(q)u$$

$$H(q) = \begin{bmatrix} I_1 + I_2 + m_2 l_1^2 + 2 m_2 l_1 l_{c_2} c_2 & I_2 + m_2 l_1 l_{c_2} c_2 \\ I_2 + m_2 l_1 l_{c_2} c_2 & I_2 \end{bmatrix}$$

$$\ddot{q} = H^{-1}(Bu - C\dot{q} - G)$$

$$C(q,\dot{q}) = \begin{bmatrix} -2 m_2 l_1 l_{c_2} s_2 \dot{q}_2 & -m_2 l_1 l_{c_2} s_2 \dot{q}_2 \\ m_2 l_1 l_{c_2} s_2 \dot{q}_1 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} m_1 g l_{c_1} s_1 + m_2 g (l_1 s_1 + l_{c_2} s_{1+2}) \\ m_2 g l_{c_2} s_{1+2} \end{bmatrix}$$

$B = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ $B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

$\cancel{B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ or } \begin{bmatrix} 1 \\ 1 \end{bmatrix}}$

acrobot twolinkarm