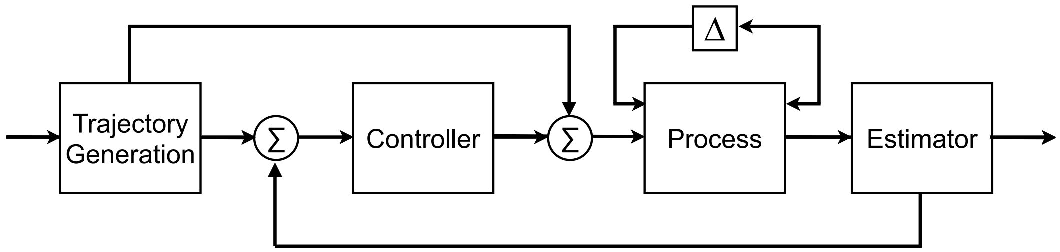
NCS Lecture 4: Trajectory Generation and Differential Flatness, 17 Mar 08

Modern Control System Design



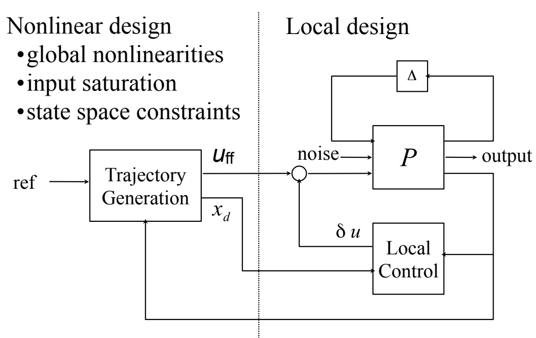
Traditional Control System: controller + process

- Corresponds to “inner loop” of most control system designs

Modern Control System: optimization-based design + robust analysis

- Replace reference with reference trajectory (Weeks 1-4)
- Replace process output with estimated output (Weeks 5-8)
- Replace “inner loop” controller with robust controller (Week 9-10 + CDS 212/213)

Two Degree of Freedom Control Design



Case 1: stabilize equilibrium point

- Given equilibrium point (x_d, u_{ff}), find a controller $u = \alpha(x, x_d, u_{ff})$ that stabilizes x_d
- State space: $u = -K(x - x_d) + u_{ff}$
- Variation: $u = -K(x - x_d) + k_r\, r$

Case 2: track a reference trajectory

- Given a reference trajectory r(t), find a controller $u = \alpha(x, r(\cdot))$ such that

$$\lim_{t\to\infty} (r(t) - y(t)) = 0$$

- Controller can often depend on r at future times (anticipation)

Case 3: optimize performance criterion

- Find controller (trajectory + feedback) that minimizes a cost function:

$$\int_0^T L(x,u)\,dt + V(x(T), u(T))$$

Key idea

- Separate problems of generating a *feasible* trajectory from local *tracking*
- Use linear control theory for generating local control laws based on the *error*
- Use nonlinear and optimal control for generating feasible trajectories
- Recompute the trajectory during operation ⇒ *receding horizon control*

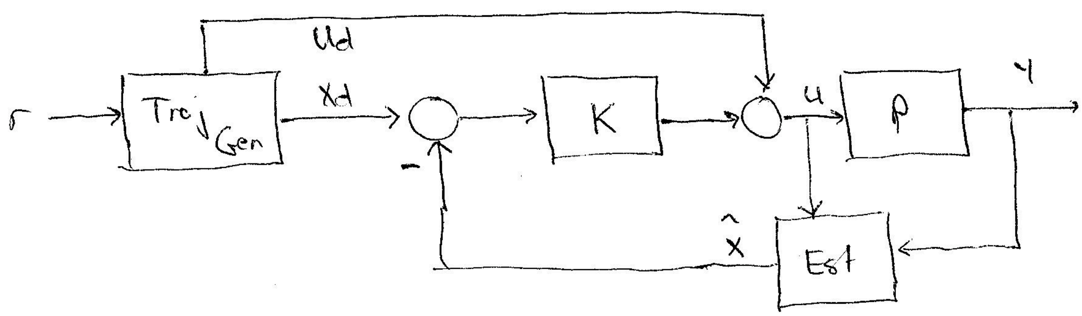
<u>Trajectory tracking problem</u>: Given a nonlinear control system

$$\dot{x} = f(x,u) \qquad x \in \mathbb{R}^n,\ u \in \mathbb{R}^p$$

$$y = h(x) \qquad y \in \mathbb{R}^q$$

and a reference trajectory $r(t) \in \mathbb{R}^q$, find a control law $u = \alpha(x, r)$ such that $\lim_{t \to \infty} (y(t) - r(t)) = 0$.

Approach: two degree of freedom design



Trajectory generation: find <u>feasible</u> trajectory (satisfies dynamics)

$$\left.\begin{aligned} \dot{x}_d &= f(x_d, u_d) \\ r &= h(x_d) \end{aligned} \quad \longrightarrow \quad x_d(r),\ u_d(r) \right\}$$

Will discuss this problem on <u>Wed</u>

Tracking: choose $u = \alpha(\hat{x}, x_d, u_d)$ ~~&~~ such that closed loop system is stable, with desired performance

Estimation: determine $\hat{x}$ from y, u

Gain scheduling

Nonlinear system with feasible trajectory:

$$\dot{x} = f(x, u) \qquad \dot{x}_d = f(x_d, u_d)$$
$$y = h(x) \qquad r(t) = h(x_d)$$

Solve via optimal control or other means (more on Wed)

To stabilize the reference trajectory, look at error $e = x - x_d$

$$\dot{x} = f(x, u) \qquad e = x - x_d$$
$$\dot{x}_d = f(x_d, u_d) \qquad v = u - u_d$$

$$\dot{e} = f(x, u) - f(x_d, u_d) =: F(e, v, x_d, u_d)$$

$$= f(e + x_d, v + u_d) - f(x_d, u_d)$$

Nonlinear, time-varying system

We can now treat (x_d, u_d) as parameters in the controller and linearize around $(e, v) = (0, 0)$

$$\dot{e} \approx A_d e + B_d v \qquad A_d = \left.\frac{\partial F}{\partial e}\right|_{(0,0)} = \left.\frac{\partial f}{\partial x}\right|_{(x_d, u_d)}$$

$$B_d = \left.\frac{\partial F}{\partial v}\right|_{(0,0)} = \left.\frac{\partial f}{\partial u}\right|_{(x_d, u_d)}$$

Now stabilize $e = 0$ by choosing K_d such that $(A_d - B_d K_d)$ stable.

$$u = -K_d e + u_d = -K_d(x - x_d) + u_d$$

Note that K_d depends on (x_d, u_d). Eq

$$u = -K_d(x_d, u_d) \cdot (x - x_d) + u_d$$

EECI 2008: skip this page an the next if everyone already knows about gain scheduling

RMM 11 Feb 06

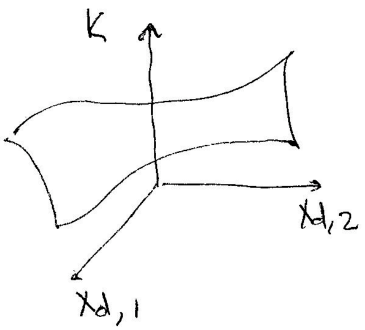
Remarks

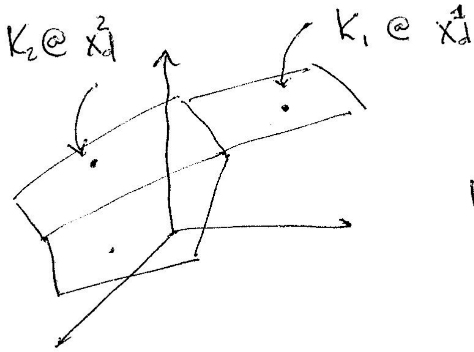
1. Does not assume (x_d, u_d) are equilibrium values, just that they satisfy dynamics. Eq pt is a special case (r = constant)

2. More generally, can schedule gains on any parameter, or even the state: $u = -K(x,\mu)\cdot(x - x_d) + u_d$. Use with care (NL)

3. Problem: linearizing about desired eq pt $\Rightarrow$ linearization may not work well if you are far from this point.

4. In practice, implement gain scheduling through interpolation





$K_d = \sum \alpha_i(x) K_i$

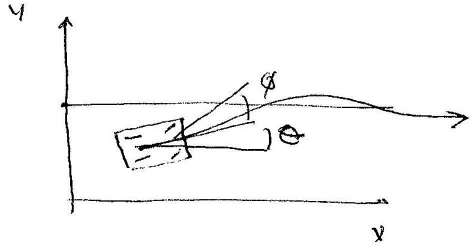
interpolation fcns

Sample implementation: "falcon" library (RMM web page)

5. Theory: not much can be said about when this works

   - OK if (x_d, u_d) are varying sufficiently slowly
   - Works well in practice, even if they aren't

Example: steering control with velocity scheduling



$$\dot{x} = \cos\theta\, v \qquad \dot{y} = \sin\theta\, v \qquad \dot{\theta} = \frac{v}{l}\tan\phi$$

$$\left.\begin{matrix} x_d = V_r t \\ y_d = y_{ref} \\ \theta_d = 0 \end{matrix}\right] x_d \qquad \left.\begin{matrix} V_d = V_r \\ \phi_d = 0 \end{matrix}\right] u_d$$

Note that $(x_d, y_d, \theta_d; V_d, \phi_d)$ are not at equilibrium, but they do satisfy the equations of motion.

$$A_d = \left.\frac{\partial f}{\partial x}\right|_{(x_d,u_d)} = \left.\begin{bmatrix} 0 & 0 & -\sin\theta \\ 0 & 0 & \cos\theta \\ 0 & 0 & 0 \end{bmatrix}\right|_{(x_d,u_d)} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

$$B_d = \left.\frac{\partial f}{\partial u}\right|_{(x_d,u_d)} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & V_r/l \end{bmatrix} \longleftarrow \text{depends on } u_d \text{ through } V_r$$

Error dynamics: $e = x - x_d \qquad w = u - u_d$

$$\begin{aligned} \dot{e}_x &= w_1 & w_1 &= -\lambda_1 e_x \\ \dot{e}_y &= e_\theta & w_2 &= -\frac{l}{V_r}(a_1 e_y + a_2 e_\theta) \\ \dot{e}_\theta &= \frac{V_r}{l} w_2 \end{aligned}$$

pole at λ_1 and $s^2 + a_1 s + a_2 = 0$

Controller in original coordinates

$$\begin{bmatrix} v \\ \phi \end{bmatrix} = -\underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \frac{a_1 l}{V_r} & \frac{a_2 l}{V_r} \end{bmatrix}}_{K_d} \underbrace{\begin{bmatrix} x - V_r t \\ y - y_r \\ \theta \end{bmatrix}}_{e} + \underbrace{\begin{bmatrix} V_r \\ 0 \end{bmatrix}}_{u_d}$$

Intuition: at slower speeds, turn wheel harder to get same time response. Note that gains blow up if $V_r = 0$ (not moving)

Lecture 1-2, Trajectory generation and differential flatness
CDS 110b, 9 Jan 08

RMM 13 Feb 07

1

Trajectory generation: find $x_d(t), u_d(t)$ that satisfy

$$(*) \qquad \dot{x}_d = f(x_d, u_d) \qquad x_d(0) = x_0 \quad x_d(T) = x_f$$

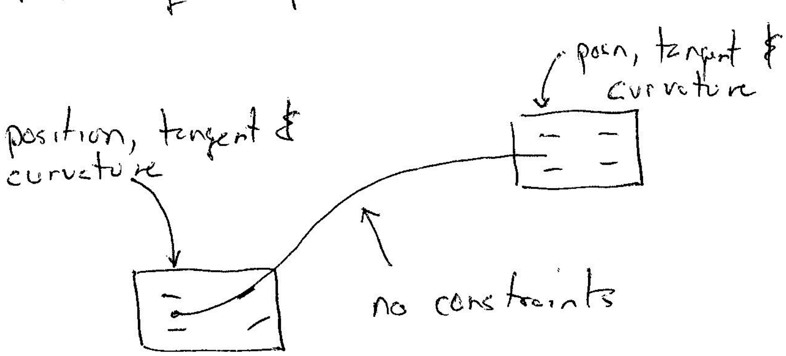
In addition, we may wish to satisfy additional constraints

- Input saturation: $|u(t)| < M$
- State constraints: $g(x) \leq 0$
- Tracking: $h(x) = r(t)$
- Optimization: $\min \int_0^T L(x,u)\,dt + V(x(T), u(T))$

  Special case: $L(x,u) = \| h(x) - r(t) \|^2 \Rightarrow$ tracking

In principle, we can use maximum principle to solve this, but often difficult to work out <u>explicit</u> solutions for x_d, u_d from two point boundary value problem $\Rightarrow$ look for other solutions.

Motivating example: kinematic car



$$\dot{x} = \cos\theta\, v$$
$$\dot{y} = \sin\theta\, v$$
$$\dot{\theta} = \frac{v}{\ell} \tan\phi$$

Given x, y, we can solve for $\theta = \operatorname{atan}(\dot{y}/\dot{x})$, $v = \dot{x}/\cos\theta$, $\phi = \operatorname{atan}\left(\frac{\ell}{v\dot{\theta}}\right) \Rightarrow$ get trajectory for full state. Initial &

Differential Flatness

Defn A nonlinear system (*) is <u>differentially flat</u> if there exists a function α such that

$$z = \alpha(x, u, \dot{u}, \ldots, u^{(q)})$$

and we can write the solutions of the differential equation as functions of z and a finite number of derivatives

$$x = \beta(z, \dot{z}, \ldots, z^{(p)})$$
$$u = \gamma(z, \dot{z}, \ldots, z^{(p)})$$

Example: for kinematic car $z = \alpha(\vec{x}) = (x, y)$

Remarks

1. For a differentially flat system, the <u>flat outputs</u>, z, completely define the feasible trajectories of the system
2. The number of flat outputs = number of system inputs
3. General theory for determing if a system is flat is hard; usually guess & check
4. General classes of systems that are flat:
   - Reachable linear systems
   - Mechanical systems with m configuration variables and m inputs
   - Feedback linearizable ~~system~~ nonlinear systems

RMM 13 Feb 07

Using Flatness to plan trajectories

3

Suppose we wish to generate a feasible trajectory for NL system

$$\dot{x} = f(x, u) \qquad x(0) = x_0 \qquad x(T) = x_f$$

If system is differential flat then

$$x(0) = \beta\left(z(0), \dot{z}(0), \ldots, z^{(p)}(0)\right) = x_0$$
$$x(T) = \gamma\left(z(T), \dot{z}(T), \ldots, z^{(p)}(T)\right) = x_f$$

Find any $z(0), \dot{z}(0), \ldots$ & $z(T), \dot{z}(T), \ldots$ that satisfy these constraints (often unique). Choose

$$z(t) = \sum_{i=1}^{N} \alpha_i \psi_i(t)$$

$\alpha_i \in \mathbb{R}$
ψ^i smooth basis fcns

$$\dot{z}(t) = \sum_{i=1}^{N} \alpha_i \dot{\psi}_i(t)$$

$\vdots$

$$z^{(p)}(t) = \sum_{i=1}^{N} \alpha_i \psi_i^{(p)}(t)$$

$$\begin{bmatrix} \psi_1(0) & \psi_2(0) & \cdots & \psi_N(0) \\ \dot{\psi}_1(0) & \dot{\psi}_2(0) & \cdots & \dot{\psi}_N(0) \\ \vdots & & & \\ \psi_1^{(p)}(0) & \psi_2^{(p)}(0) & \cdots & \psi_N^{(p)}(0) \\ \psi_1(T) & & \cdots & \psi_N(T) \\ \vdots & & & \vdots \\ \psi_1^{(p)}(T) & & & \psi_N^{(p)}(T) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix} = \underbrace{\begin{bmatrix} z(0) \\ \dot{z}(0) \\ \vdots \\ z^{(p)}(0) \\ z(T) \\ \vdots \\ z^{(p)}(T) \end{bmatrix}}_{b}$$

RMM 13 Feb 07

④

<u>Example</u> Nonholonomic integrator

$$\dot{x}_1 = u_1 \qquad \dot{x}_2 = u_2 \qquad \dot{x}_3 = x_2 u_1$$

Differentially flat w/ $z = (x_1, x_3)$

$$x_1 = z_1 \qquad x_2 = \dot{x}_3/\dot{x}_1 = \dot{z}_2/\dot{z}_1$$

$$x_3 = z_2 \qquad u_1 = \dot{z}_1$$

$$u_2 = \dot{x}_2 = \frac{d}{dt}\left(\dot{z}_2/\dot{z}_1\right) = \frac{\ddot{z}_2}{\dot{z}_1} - \frac{\dot{z}_2 \ddot{z}_1}{\dot{z}_1^2}$$

Note: intuition says taking derivatives is "noisy". Here we are taking derivatives of basis functions ⇒ noise isn't an issue.

$$z_1(t) = \sum a_{1,i} \psi_{1,i}(t) \qquad z_2(t) = \sum a_{2,i} \psi_{2,i}(t)$$

Generate a trajectory from origin to a point $x = (x_{1f}, x_{2f}, x_{3f})$

Choose basis functions as simple polynomials

$$\psi_{1,1}(t) = 1 \qquad \psi_{1,2}(t) = t \qquad \psi_{13}(t) = t^2 \qquad \psi_{14}(t) = t^3$$

$$\psi_{2,1}(t) = 1 \qquad \psi_{2,2}(t) = t \qquad \psi_{2,3}(t) = t^2 \qquad \psi_{24}(t) = t^3$$

$$\begin{array}{c} z_1 \\ \dot{z}_1 \\ z_2 \\ \dot{z}_2 \\ \\ \\ \\ \\ \end{array}
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & T & T^2 & T^3 & 0 & 0 & 0 & 0 \\
0 & 1 & 2T & 3T^2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & T & T^2 & T^3 \\
0 & 0 & 0 & 0 & 0 & 1 & 2T & 3T^2
\end{bmatrix}
\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{21} \\ a_{22} \\ a_{23} \end{bmatrix}
=
\begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ x_{1f} \\ 1 \\ x_{3f} \\ x_{2f} \end{bmatrix}$$

8x8 matrix
Invertible for most T
Doesn't depend on x_0, x_f

Some freedom here

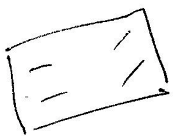
<u>Remarks</u>

1. There are several remaining degrees of freedom (T, 1, etc) that have to be specified ⇒ solutions are not unique
2. Can also over parameterize and solve least squares

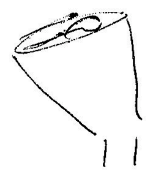
RMM 13 Feb 06

6

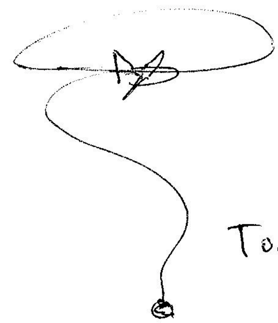
Examples of flat systems



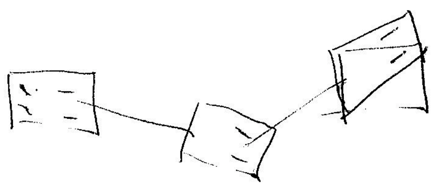
Kin car



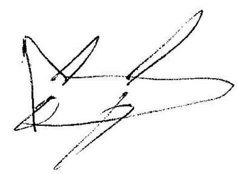
Ducted fan



Towed cable



N-trailer



Lift (elevator) $\ll$ Lift (wings)

Additional discussion topics for EECI:
* Other approaches to trajectory generation (shooting, optimal control)
* Known conditions for flatness (triangular, feedback linearizable, reachable linear, ...)
* Comparison between flatness and feedback linearization (if people know fbklin)

Skip this topic if short on time; will cover in lecture on Tuesday

RMM 13 Feb 07

Using flatness for constrained, sub-optimal trajectory generation

Return to full problem

$\cancel{\dot{x} = Ax}$, $\min \int_0^T L(x,u)\,dt + V(x(T), u(T))$

subject to

$\dot{x} = f(x,u) \qquad g(x,u) \le 0 \leftarrow$ state + input constraints

If system is differentially flat, can write $z = \sum a_i \psi_i(t)$

$$x = \beta(z, \dot{z}, \dots, z^{(q)}) = \beta(a,t)$$
$$u = \gamma(\qquad\qquad) = \gamma(a,t)$$

Can rewrite entire problem in terms of α

$$\min \int_0^T L(\beta(\alpha,t), \gamma(a,t))\,dt + V(\beta(\alpha,T), \gamma(\alpha,T))$$

$$g(\beta(\alpha,t), \gamma(\alpha,t)) \le 0 \qquad \text{No dynamics}$$

This is an optimization problem in fixed set of parameters: given a, can compute all quantities. Very good numerical tools available.

Remarks

1. $g(\beta(\alpha,t), \gamma(\alpha,t))$ give constraint for each $t \Rightarrow$ infinite # of constraints $\Rightarrow$ hard. Can replace with $g(\beta(a,t_i), \gamma(a,t_i))$ for $t_1, t_2, \dots, t_M \Rightarrow$ finite approx.
2. Can extend to non-flat, etc (more next week)
3. Sub-optimal, but fast