Lecture notes: Trajectory generation and tracking

Jie Fu

Department of Electrical and Computer Engineering
Robotics Engineering Program
Worcester Polytechnic Institute

RBE502

Outline

This lecture note is based on

- Karl Johan Aström Richard M. Murray, *Feedback Systems, An introduction to Scientists and Engineers*. Chapter 6-7. http://www.cds.caltech.edu/~murray/amwiki/index.php/Second_Edition

Linearity

In general, a system is given by

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

where $f(x,u), h(x,u)$ is a nonlinear function.
Assume the system has an equilibrium x_e, u_e,

Deviation variables

$$\delta_x = x(t) - x_e; \quad \delta_u = u(t) - u_e.$$

$\delta y = y(t) - y_e$

$y_e = h(x_e, u_e)$

and rewrite the equation of motion in new variable.:

$$\dot{\delta}_x + \dot{x}_e = f(\delta_x + x_e, \delta_u + u_e)$$

and

$$\delta_y + y_e = h(\delta_x + x_e, \delta_u + u_e)$$

Jacobian Linearization

- $\delta_x, \delta_u, \delta_y$ are all close to zero when we are near the equilibrium point.
- Eliminate the higher-order terms in the taylor series expansion of the vector fields $f()$ and $h()$

Jacobian linearization of the nonlinear system is

$$\dot{\delta}_x \doteq A\delta_x + B\delta_u, \quad , \delta_y \doteq C\delta_x + D\delta_u$$

where

$$A = \frac{\partial f}{\partial x}\Big|_{x_e, u_e}, \quad B = \frac{\partial f}{\partial u}\Big|_{x_e, u_e},$$

$$C = \frac{\partial h}{\partial x}\Big|_{x_e, u_e}, \quad D = \frac{\partial h}{\partial u}\Big|_{x_e, u_e},$$

The system only approximates the original system around the equilbrium point.

$$\delta_y + y_e = h(\delta_x + x_e, \delta_u + u_e) \qquad h(x, u)$$

$$\delta_y + \cancel{h(x_e, u_e)} \doteq \cancel{h(x_e, u_e)} + \left.\frac{\partial h}{\partial x}\right|_{x_e, u_e} (x - x_e) + \left.\frac{\partial h}{\partial u}\right|_{x_e, u_e} (u - u_e) + \underbrace{\cdots}_{\text{higher order terms}}$$

$$y_e = h(x_e, u_e)$$

$$\delta_y \doteq \underbrace{\left.\frac{\partial h}{\partial x}\right|_{x_e, u_e}}_{C} \delta_x + \underbrace{\left.\frac{\partial h}{\partial u}\right|_{x_e, u_e}}_{D} \delta_u$$

Example: Jacobian matrix

- Suppose you have two dim function:

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix}$$

- The gradient generation

$$\Delta_x = \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \dots \frac{\partial}{\partial x_n} \end{bmatrix}^T$$

- The jacobian is defined by

$$J_f = \begin{bmatrix} f_1(x) \\ f_2(x) \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} & \dots \frac{\partial}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \end{bmatrix}$$

Consider a nonlinear system

$$
\begin{aligned}
\dot{x}_1 &= x_1 \sin x_2 + x_2 u = f_1(x, u) \\
\dot{x}_2 &= x_1 e^{-x_2} + u^2
\end{aligned}
$$

with output

$$y = 2x_1 x_2 + x_2^2$$

Assuming the system is given a desired trajectory and input

$$x^d(t), u^d(t)$$

What is the linearized state space equation of this nonlinear system?

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix}\Bigg|_{x^d, u^d} = \begin{bmatrix} \sin x_2 & x_1 \cos x_2 + u \\ e^{-x_2} & -x_1 e^{-x_2} \end{bmatrix}\Bigg|_{x^d, u^d}$$

Gain scheduling control

gain scheduling: is used to describe any controller that depends on a set of measured parameters in the system.
Consider the stabilizing control of a nonlinear system with jacobian linearization:
Given x_d and u_d are constants, then

$$\dot{\delta}_x = A\delta_x + B\delta_u$$

where A and B are constant matrix. $\lim_{t\to\infty} \delta_x \to 0 \Rightarrow \delta_u = -K\delta_x$

A-BK is stable.

The feedback control for stabilizing is

$$u = \delta_x + u_e = -K\delta_x + u_e$$

$$= -K(x - x_e) + u_e$$

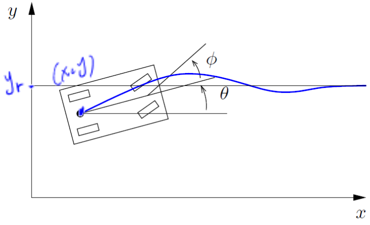
Gain scheduling control

General form of Gain scheduling:

$$u = -K(x, \mu)(x - x_d) + u_d$$

where $-K(x, \mu)$ depends on the current system state and an external parameter μ.

Example: Steering control with velocity scheduling.



(a)

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

output $\vec{y} = [0 \ 1 \ 0]\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ $\qquad \vec{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$

Control objective: to follow a straight line in the x direction at lateral position y_r and fixed velocity v_r,
a feasible trajectory:

- desired state $x_d = (v_r t, \ y_r, \ 0)$
- desired input $u_d = (v_r, \ 0)$

Linearize the system around the desired trajectory and obtain the error dynamics. $\delta x, \ \delta y$

$$A = \frac{\partial f}{\partial \vec{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} & \frac{\partial f_1}{\partial \theta} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} & \frac{\partial f_2}{\partial \theta} \\ \frac{\partial f_3}{\partial x} & \frac{\partial f_3}{\partial y} & \frac{\partial f_3}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -v\sin\theta \\ 0 & 0 & v\cos\theta \\ 0 & 0 & 0 \end{bmatrix}$$

$$\dot{x} = v\cos\theta$$

$$\dot{y} = v\sin\theta$$

$$\dot{\theta} = \frac{v}{\ell}\tan\phi$$

$$B = \frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial v} & \frac{\partial f_1}{\partial \phi} \\ \frac{\partial f_2}{\partial v} & \frac{\partial f_2}{\partial \phi} \\ \frac{\partial f_3}{\partial v} & \frac{\partial f_3}{\partial \phi} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ \frac{1}{\ell}\tan\phi & \frac{v}{\ell}\sec^2\phi \end{bmatrix}$$

$$\vec{u} = \begin{bmatrix} v \\ \phi \end{bmatrix}$$

subsitute $\vec{x}_d$, $\vec{u}_d$

$$A\Big|_{\substack{x_d \\ u_d}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} ; \quad B = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & \frac{v_r}{\ell} \end{bmatrix}$$

$$\dot{\vec{\delta}}_x = A\vec{\delta}_x + B\underline{\delta_u}$$

$$\vec{\delta}_x = \begin{bmatrix} \delta_x \\ \delta_y \\ \delta_\theta \end{bmatrix}$$

$$\vec{\delta}_y = C\vec{\delta}_x = [0 \quad 1 \quad 0]\begin{bmatrix} \delta_x \\ \delta_y \\ \delta_\theta \end{bmatrix} = \delta_y$$

$$\vec{\delta}_u = \begin{bmatrix} \delta_r \\ \delta_\phi \end{bmatrix}$$

decoupled

$$\boxed{\begin{aligned} \dot{\delta}_x &= \delta_r \\ \dot{\delta}_y &= V_r\delta_\theta \\ \dot{\delta}_\theta &= \frac{V_r}{l}\underline{\delta_\phi} \end{aligned}}$$

$$[B \quad AB \quad A^2B] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{V_r^2}{l} & 0 & 0 \\ 0 & \frac{V_r}{l} & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\dot{\delta}_x = \delta_r$$

pick $\delta_r = -\lambda_1\delta_x \Rightarrow \dot{\delta}_x = -\lambda_1\delta_x \quad \lambda_1 > 0$

$$\begin{bmatrix} \dot{\delta}_y \\ \dot{\delta}_\theta \end{bmatrix} = \begin{bmatrix} 0 & V_r \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \delta_y \\ \delta_\theta \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{V_r}{l} \end{bmatrix}\delta_\phi$$

$$\delta_\phi = [-k_1 \quad -k_2]\begin{bmatrix} \delta_y \\ \delta_\theta \end{bmatrix} \quad \text{such that} \quad \underbrace{\begin{bmatrix} 0 & V_r \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{V_r}{l} \end{bmatrix}[k_1 \; k_2]}_{\text{stable}}$$

$$k_1 = \frac{l}{V_r} a_1 \quad ; \quad k_2 = \frac{l}{V_r} a_2$$

$\delta v, \quad \delta\phi$

$$v = V_r + \delta v$$

$$\phi = \phi_d + \delta\phi = \delta\phi = -\frac{l}{V_r} a_1 \underbrace{(y - y_r)}_{\delta y} - \frac{l}{V_r} a_2 \, \cancel{\delta}\theta$$

$\theta - \theta_d$

$\to 0$

$$\begin{bmatrix} 0 & V_r \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ a_1 & a_2 \end{bmatrix} = \begin{bmatrix} 0 & V_r \\ -a_1 & -a_2 \end{bmatrix}$$

$$\det\left(\lambda I - \begin{bmatrix} 0 & V_r \\ -a_1 & -a_2 \end{bmatrix} \right) = 0$$

$$\det \begin{bmatrix} \lambda & -V_r \\ a_1 & \lambda + a_2 \end{bmatrix} = 0$$

$$\lambda(\lambda + a_2) + a_1 V_r = \lambda^2 + a_2 \lambda + a_1 V_r$$

$$\frac{-a_2 \pm \sqrt{a_2^2 - 4 a_1 V_r}}{2}$$

① V_r small $\to$ ϕ large

② V_r large $\to$ ϕ small

③ $V_r = 0 \to \phi$ infinite.

$$\begin{bmatrix} v \\ \phi \end{bmatrix} = -\underbrace{\begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \frac{a_1 l}{V_r} & \frac{a_2 l}{V_r} \end{bmatrix}}_{K(x,u) = K_d} \underbrace{\begin{bmatrix} x - v_r t \\ y - y_r \\ \theta \end{bmatrix}}_{d_x} + \underbrace{\begin{bmatrix} v_r \\ 0 \end{bmatrix}}_{u_d}$$

Trajectory generator

$$\dot{x}_d = f(x_d, u_d)$$

where

- x_d is the desired state trajectory.
- u_d is the feedforward signal: Given the system $x(0) = x_d(0)$, the input u_d, if **no disturbances**, **no initial state error**, and **no modeling error**, the system $x(t) = x_d(t)$ for all t.

We also call $x_d(t)$, for $t \geq 0$ a **feasible** trajectory of the system.

Trajectory generator

Consider a chain of integrator

$$\dot{x} = v, \quad \dot{v} = a$$

control input $u = a$.
Design the controller that the system trajectory tracks $x_d(t) = 1 + t^2 + 3t^3$ for $t \in [0, T]$.

$\dot{x}_d = v_d = 2t + 9t^2$

$\dot{v}_d = a = 2 + 18t$

the feedforward signal is $u_d =$. $2 + 18t$

Trajectory Generation and Differential Flatness

Additional Performance Requirement:

- input saturation constraints $|u(t)| < M$, state constraints $g(x) \leq 0$ and tracking constraints $y(t) = r(t)$, each of which gives an algebraic constraint on the states or inputs at each instant in time.
- Also to optimize a function by choosing $(x_d(t), u_d(t))$ to minimize

$$\int_{t=0}^{T} L(x,u)dt + V(x(T), u(T)).$$

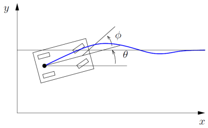
  where $L(x,u)$ is the running cost, and $V(x(T), u(T))$ is the terminal cost.

Examples:

1. minimum effort: $\int_{t=0}^{T} |u|^2 dt$.
2. minimum jerk: x is the position, $x^{(3)}$ is the jerk: $\int_{t=0}^{T} |x^{(3)}|^2 dt$.
3. etc.

Differential flatness

Example: Consider a Dubin's car dynamical system:



(a)

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

$\dot{x}_d = V_d \cos\theta_d \quad \dot{y}_d = V_d \sin\theta_d \quad \phi_d = \tan^{-1}\left(\frac{\ell\dot{\theta}_d}{V_d}\right)$

Suppose that we are given a trajectory for the rear wheels of the system, $x_d(t)$ and $y_d(t)$:

Can we solve for ...

- $\theta_d(t)$ $= \tan^{-1}\left(\dot{y}_d / \dot{x}_d\right) = \tan^{-1}\left(\dot{\alpha}_2(t) / \dot{\alpha}_1(t)\right) = \tan^{-1}\left(\frac{\dot{z}_2}{\dot{z}_1}\right)$
- $v_d(t)$ $= \dot{x}_d / \cos\theta_d$
- $\phi_d(t)$ $= \tan^{-1}\left(\frac{\ell\dot{\theta}_d}{v_d}\right)$

$\Big\}\ f_1(z, \dot{z}, \ldots)$

$\vdots$

Differentially flatness

Definition 1.1 (Differential flatness). A nonlinear system (1.1) is *differentially flat* if there exists a function α such that

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

and we can write the solutions of the nonlinear system as functions of z and a finite number of derivatives

$$\begin{aligned} \vec{x} &= \beta(z, \dot{z}, \ldots, z^{(q)}), \\ \vec{u} &= \gamma(z, \dot{z}, \ldots, z^{(q)}). \end{aligned} \tag{1.4}$$

- z: Flat output.
- in Dubins car: $z = [x_d, y_d]^T$.

$$z = \alpha(x, u \cdots u^{(p)}) = \begin{bmatrix} \alpha_1(\cdot) \\ \alpha_2(\cdot) \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix}$$

$$z = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_d \\ y_d \end{bmatrix}$$

Trajectory Generation and Differential Flatness

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

If the system is differentially flat then

$$x(0) = \beta\big(z(0), \dot{z}(0), \ldots, z^{(q)}(0)\big) = x_0,$$
$$x(T) = \gamma\big(z(T), \dot{z}(T), \ldots, z^{(q)}(T)\big) = x_f,$$

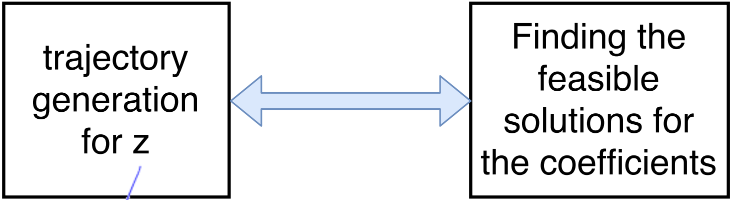
Suppose we aims to generate a trajectory:

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

Any trajectory for z that satisfies these boundary conditions will be a feasible trajectory for the system.

Trajectory generation

Let $z(t) = \sum_{i=1}^{N} \alpha_i \phi_i(t)$ where $\phi_i(t)$ is a basis function with variable t.



$x = \beta(z, \dot{z} \cdots z^{(p)})$

$u = \gamma(z, \dot{z} \cdots z^{(p)})$

$z(t) = \alpha_1 + \alpha_2 t + \alpha_3 t^2 + \alpha_4 t^3$

$\phi_i = \underbrace{1, t, t^2}_{4}, t^3 \cdots$

Formulate the linear matrix equations for trajectory generation.

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

$$\begin{array}{l} z(0) \\ \dot{z}(0) \\ \vdots \\ z^{(p)}(0) \\ \vdots \\ z(T) \\ \dot{z}(T) \\ \vdots \\ z^{(p)}(T) \end{array}$$

$$\left\{ \begin{array}{l} X(0) = \beta(z(0), \dot{z}(0) \cdots z^{(p)}(0)) \\ X(T) = \beta(z(T), \dot{z}(T) \cdots z^{(p)}(T)) \end{array} \right\}$$

unknowns $\alpha_i,\ i = 1, \dots N.$

$$\left. \begin{array}{l} \dot{z}(t) = \sum_{i=1}^{N} \alpha_i \dot{\phi}_i(t) \\ \ddot{z}(t) = \sum_{i=1}^{N} \alpha_i \ddot{\phi}_i(t) \\ \vdots \\ z^{(p)}(t) = \sum_{i=1}^{N} \alpha_i \phi_i^{(p)}(t) \end{array} \right\} \quad \begin{array}{l} t = 0. \\ t = T \end{array}$$

$$\boxed{2p \times N}$$

$$\begin{bmatrix} \phi_1(0) & \phi_2(0) & \cdots & \phi_N(0) \\ \dot{\phi}_1(0) & \dot{\phi}_2(0) & \cdots & \dot{\phi}_N(0) \\ \vdots & & & \vdots \\ \phi_1(T) & \phi_2(T) & \cdots & \phi_N(T) \\ \vdots & \vdots & & \vdots \\ \phi_1^{(p)}(T) & \phi_2^{(p)}(T) & \cdots & \phi_N^{(p)}(T) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} = \begin{bmatrix} z(0) \\ \dot{z}(0) \\ \vdots \\ z^{(p)}(0) \\ z(T) \\ \vdots \\ z^{(p)}(T) \end{bmatrix}$$

Example: nonholonomic integrator

$$\dot{x}_1 = u_1, \quad \dot{x}_2 = u_2 \quad \dot{x}_3 = x_2 u_1.$$

It is differentially flat with flat output $z = (x_1, x_3)$.

$x_1 = z_1$

$x_3 = z_2$

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

$u_1 = \dot{x}_1$

$u_2 = \dot{x}_2$

$\dot{x}_3 = x_2 u_1$

$= x_2 \dot{x}_1$

$z = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

$x_3 = \ ?$

$x_1(0) = z_1(0)$

$x_3(0) = z_2(0)$

$x_2(0) = \dfrac{\dot{z}_2(0)}{\dot{z}_1(0)}$

$x_1(T) = z_1(T)$

$x_3(T) = z_2(T)$

$x_2(T) = \dfrac{\dot{z}_2(T)}{\dot{z}_1(T)}$

$z_1 = \sum_{i=1}^{N} \alpha_i \phi_i(t) = \alpha_{11} + \alpha_{12} t + \alpha_{13} t^2$

$[1, t, t^2, t^3]$ $\quad z_2 = \alpha_{21} + \alpha_{22} t + \alpha_{23} t^2$

using simple polynomials of time as basis:

$$\phi_{1,i}(t) = t^i.$$

Given $x(0) = [1, 2, 10]$ and $x(T) = [2, 12, 20]$. Solve for the feasible trajectories.

Matlab example: Vehicle tracking

Consider a dubins car:

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

where x, y, θ are position and turning angle of the vehicle. v and w are linear and angular velocity.
The system is "differentially flat" and fully controllable. in matlab, we demonstrate

- Trajectory planning.
- Jacobian Linearization.
- Feedback control for trajectory tracking.

$$x_d = \sum \alpha_i \phi_i(t)$$

$$y_d = \sum b_i \phi_i(t)$$

$$\theta_d = \tan^{-1}\left(\frac{\dot{y}_d}{\dot{x}_d}\right)$$

$T=10$

$x_d(0) \quad\quad x_d(T)$

$y_d(0) \quad \rightarrow \quad y_d(T)$

$\theta_d(0) \quad\quad \theta_d(T)$

$$\left.\begin{aligned} \dot{y} &= v\sin\theta \\ \dot{x} &= v\cos\theta \end{aligned}\right\} \Rightarrow \boxed{\underbrace{\dot{y}\cos\theta - \dot{x}\sin\theta}_{v\sin\theta\cos\theta \,-\, v\cos\theta\sin\theta} = 0}$$

$$t=0: \quad \underset{\Delta}{\dot{y}_d(0)}\,\underbrace{\cos\theta_d(0)}_{\text{const}} - \underset{\Delta}{\dot{x}_d(0)}\,\underbrace{\sin\theta_d(0)}_{\text{const.}} = 0$$

$$\sum \alpha_i \dot{\phi}_i(0)\cdot \text{const}_1 - \sum a_i \dot{\phi}_i(0)\cdot \text{const}_2 = 0$$

$v_d(0) = 0$

$v_d(T) = 0$

$$\dot{y}\,\underbrace{\sin\theta}_{\text{const}} + \dot{x}\,\underbrace{\cos\theta}_{\text{const}} = v\sin^2\theta + v\cos^2\theta = v$$

$v_d(0)$

$v_d(T)$

Conclusion

- A nominal trajectories and inputs that satisfy the equations of motion for a differentially flat system can be computed in a computationally efficient way (solving **a set of algebraic equations**).
- Constraints of the system — the constraints for the flat output:
  - Bounds on the inputs, can be transformed into the flat output space and (typically) become limits on the curvature or higher order derivative properties of the curve.
  - Performance index can be transformed and becomes a functional depending on the flat outputs and their derivatives up to some order.
- Unfortunately, general conditions for flatness are not known,
- but many important class of nonlinear systems, including feedback linearizable systems, are differential flat.