# Control and Estimation

Liangchun Xu

Department of Mechanical Engineering, Tufts University
574 Boston Avenue, Medford, 02155, US

*Email:* liangchun.xu@tufts.edu

*March 21, 2019*

## Table of contents

# CONTROL

### State and Control Input

1. $\vec{x}$ = State = Differential Equation: how $\vec{x}$ changes is given;
2. $\vec{u}$ = Control input = Algebraic Equation: $\vec{u}$ can be changed arbitrarily.

### Laplace Transform

The Laplace transform is defined as

$$F(s) = \int_0^\infty f(t)e^{-st}dt = \mathcal{L}(f) \tag{1}$$

the inverse Laplace transform is

$$f(t) = \frac{1}{2\pi i}\lim_{T\to\infty}\int_{\gamma-iT}^{\gamma+iT} e^{st}F(s)ds \tag{2}$$

### Z-Transform

The unilateral Z-transform is the Laplace transform with

$$z \overset{\text{def}}{=\!=} e^{sT} \tag{3}$$

where $T = 1/f_s$ is the sampling period. Let

$$\triangle_T(t) \overset{\text{def}}{=\!=} \sum_{n=0}^{\infty}\delta(t-nT) \tag{4}$$

$$\begin{aligned} x_q(t) &\overset{\text{def}}{=\!=} x(t)\triangle_T(t) \\ &= x(t)\sum_{n=0}^{\infty}\delta(t-nT) \\ &= \sum_{n=0}^{\infty}x(nT)\delta(t-nT) \end{aligned} \tag{5}$$

$$x[n] \overset{\text{def}}{=\!=} x(nT) \tag{6}$$

The Laplace transform of the sampled signal $x_q(t)$ is

$$\begin{aligned} X_q(s) &= \int_0^\infty\sum_{n=0}^{\infty}x[n]\delta(t-nT)e^{-st}dt \\ &= \sum_{n=0}^{\infty}x[n]\int_0^\infty\delta(t-nT)e^{-st}dt \\ &= \sum_{n=0}^{\infty}x[n]e^{-nsT} \\ &= \sum_{n=0}^{\infty}x[n]z^{-n} \end{aligned} \tag{7}$$

The precise definition of the unilateral Z-transform of the discrete function $x[n]$ is

$$X(z) = \sum_{n=0}^{\infty}x[n]z^{-n} \tag{8}$$

### Fourier Transform

The Fourier transform is equivalent to evaluating the bilateral Laplace transform with imaginary argument $s = i\omega$ or $s = 2\pi fi$,

$$f(\omega) = F(s = i\omega) = \int_{-\infty}^{\infty}f(t)e^{-i\omega t}dt \tag{9}$$

the inverse FT is

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(\omega)e^{i\omega t}d\omega \tag{10}$$

Discrete Fourier transform (DFT) is

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n e^{-i2\pi kn/N} \\ &= \sum_{n=0}^{N-1} x_n(\cos(2\pi kn/N) - i\sin(2\pi kn/N)) \end{aligned} \tag{11}$$

Inverse DFT is

$$x_n = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{i2\pi kn/N} \tag{12}$$

**Fast Fourier Transform**

Divide DFT into a sum over the even-numbered indices and a sum over the odd-numbered indices

$$\begin{aligned} X_k &= \sum_{m=0}^{N/2-1} x_{2m}e^{-i2\pi k(2m)/N} + \sum_{m=0}^{N/2-1} x_n e^{-i2\pi k(2m+1)/N} \\ &= \sum_{m=0}^{N/2-1} x_{2m}e^{-i2\pi k(2m)/N} + e^{-i2\pi k/N}\sum_{m=0}^{N/2-1} x_n e^{-i2\pi k(2m)/N} \\ &\overset{\text{def}}{=\!=} E_k + e^{-i2\pi k/N}O_k \end{aligned} \tag{13}$$

The periodictiy of the DFT indicates

$$E_{k+\frac{N}{2}} = E_k \tag{14}$$

$$O_{k+\frac{N}{2}} = O_k \tag{15}$$

Consider $X_{k+\frac{N}{2}}$,

$$\begin{aligned} X_{k+\frac{N}{2}} &= E_{k+\frac{N}{2}} + e^{-i2\pi\left(k+\frac{N}{2}\right)/N}O_{k+\frac{N}{2}} \\ &= E_k + (e^{-\pi i}e^{-i2\pi k/N})O_k \\ &= E_k - e^{-i2\pi k/N}O_k \end{aligned} \tag{16}$$

Based on (13) and (16), FFT algorithm can reuse $E_k$, and $O_k$ to reduce DFT algorithm complexity from $O(N^2)$ to $O(N\log(N))$.

**First-Order Systems**

Consider a simplified closed-loop block diagram [4], in which the input-output relationship is

$$\frac{C(s)}{R(s)} = \frac{1}{Ts+1} \tag{17}$$

1. Unit-Step Response
   Subsitituting $R(s)=\frac{1}{s}$ into (17), we obtain,

   $$C(s)=\frac{1}{Ts+1}\frac{1}{s}=\frac{1}{s}-\frac{1}{s+(1/T)} \tag{18}$$

   Take the inverse Laplace transform,

   $$c(t)=1-e^{-t/T} \tag{19}$$

2. Unit-Ramp Response
   Subsitituting $R(s)=\frac{1}{s^2}$ into (17), we obtain,

   $$C(s)=\frac{1}{Ts+1}\frac{1}{s^2}=\frac{1}{s^2}-\frac{T}{s}+\frac{T^2}{Ts+1} \tag{20}$$

Take the inverse Laplace transform,

$$c(t) = t - T + Te^{-t/T} \tag{21}$$

The error signal is

$$e(t) = r(t) - c(t) = T(1 - e^{-t/T}) \tag{22}$$

$$\lim_{t\to\infty} e(t) = T \tag{23}$$

3. Unit-Impulse Response

   Subsitituting $R(s) = \frac{1}{s^2}$ into (17), we obtain,

$$C(s) = \frac{1}{Ts+1} \times (1) = \frac{1}{Ts+1} \tag{24}$$

Take the inverse Laplace transform,

$$c(t) = \frac{1}{T} e^{-t/T} \tag{25}$$

**Second-Order Systems**

Consider a simplified closed-loop block diagram, in which the input-output relationship is

$$\frac{C(s)}{R(s)} = \frac{K}{Js^2 + Bs + K} \tag{26}$$

Write $\frac{K}{J} = \omega_n^2$, $\frac{B}{J} = 2\zeta\omega_n = 2\sigma$, then

$$\begin{aligned} \frac{C(s)}{R(s)} &= \frac{\frac{K}{J}}{\left(s + \frac{B}{2J} + \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right)\left(s + \frac{B}{2J} - \sqrt{\left(\frac{B}{2J}\right)^2 - \frac{K}{J}}\right)} \\ &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \end{aligned} \tag{27}$$

1. Underdamped case $(0 < \zeta < 1)$:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s + \zeta\omega_n + j\omega_d)(s + \zeta\omega_n - j\omega_d)} \tag{28}$$

where $\omega_d$ is called the damped natural frequency and defined as $\omega_d = \omega_n\sqrt{1-\zeta^2}$.

For a unit-step input, $C(s)$ can be written

$$\begin{aligned} C(s) &= \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}\frac{1}{s} \\ &= \frac{1}{s} - \frac{s + 2\zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \\ &= \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} \\ &= \frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2} - \frac{\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{(s + \zeta\omega_n)^2 + \omega_d^2} \end{aligned}$$

Apply inverse Laplace transform on both sides,

$$\begin{aligned} c(t) &= 1 - \mathcal{L}^{-1}\left(\frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_d^2}\right) - \frac{\zeta}{\sqrt{1-\zeta^2}}\mathcal{L}^{-1}\left(\frac{\omega_d}{(s + \zeta\omega_n)^2 + \omega_d^2}\right) \\ &= 1 - e^{-\zeta\omega_n t}\cos(\omega_d t) - \frac{\zeta}{\sqrt{1-\zeta^2}} e^{-\zeta\omega_n t}\sin(\omega_d t) \\ &= 1 - \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}}\sin\left(\omega_d t + \tan^{-1}\frac{\sqrt{1-\zeta^2}}{\zeta}\right) \end{aligned} \tag{29}$$

The error signal is

$$\begin{aligned} e(t) &= r(t) - c(t) \\ &= e^{-\zeta\omega_n t}\left(\cos(\omega_d t) + \frac{\zeta}{\sqrt{1-\zeta^2}}\sin(\omega_d t)\right) \\ \lim_{t\to\infty} e(t) &= 0 \end{aligned} \tag{30}$$

If the damping ratio $\zeta = 0$,

$$c(t) = 1 - \cos(\omega_n t) \tag{31}$$

2. Critically damped case ($\zeta = 1$):
   Two poles of $C(s)/R(s)$ are equal,

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{(s+\omega_n)^2} \tag{32}$$

For a unit-step input, $C(s)$ can be written

$$C(s) = \frac{\omega_n^2}{(s+\omega_n)^2}\frac{1}{s} \tag{33}$$

Apply inverse Laplace transform on both sides,

$$c(t) = 1 - e^{-\omega_n t}(1+\omega_n t) \tag{34}$$

3. Overdamped case ($\zeta > 1$):
   Two poles of $C(s)/R(s)$ are negative real and unequal,

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{\left(s+\zeta\omega_n+\omega_n\sqrt{\zeta^2-1}\right)\left(s+\zeta\omega_n-\omega_n\sqrt{\zeta^2-1}\right)} \tag{35}$$

For a unit-step input, $C(s)$ can be written

$$C(s) = \frac{\omega_n^2}{\left(s+\zeta\omega_n+\omega_n\sqrt{\zeta^2-1}\right)\left(s+\zeta\omega_n-\omega_n\sqrt{\zeta^2-1}\right)}\frac{1}{s} \tag{36}$$

Apply inverse Laplace transform on both sides,

$$\begin{aligned} c(t) &= 1 + \frac{e^{-\left(\zeta+\sqrt{\zeta^2-1}\right)\omega_n t}}{2\sqrt{\zeta^2-1}\left(\zeta+\sqrt{\zeta^2-1}\right)} - \frac{e^{-\left(\zeta-\sqrt{\zeta^2-1}\right)\omega_n t}}{2\sqrt{\zeta^2-1}\left(\zeta-\sqrt{\zeta^2-1}\right)} \\ &= 1 + \frac{\omega_n}{2\sqrt{\zeta^2-1}}\left(\frac{e^{-s_1 t}}{s_1} - \frac{e^{-s_2 t}}{s_2}\right) \end{aligned} \tag{37}$$

where $s_1 = \left(\zeta+\sqrt{\zeta^2-1}\right)\omega_n$, and $s_2 = \left(\zeta-\sqrt{\zeta^2-1}\right)\omega_n$.

## PID Controller

Definition. *A proportional-integral-derivative controller (PID controller) continuously calculates an error value $e(t)$ as the difference between a desired setpoint (SP) and a measured process variable (PV) and applies a correction based on proportional, integral, and derivative terms.*

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{38}$$

**Example 1.** PD controller is a PID controller with $K_i = 0$,

$$u = -K_p(x - x_R) - K_d(\dot{x} - \dot{x}_R) \tag{39}$$

where $e = x - x_R$, and $\dot{x}_R$ is usually 0. Consider the case when applying force to move an object,

$$\begin{aligned} u &= m\ddot{x} \\ -K_p(x - x_R) - K_d(\dot{x} - \dot{x}_R) &= m\ddot{x} \end{aligned} \tag{40}$$

then the controller gains are

$$K_p = m\omega_n^2 \tag{41}$$
$$K_d = 2m\zeta\omega_n \tag{42}$$

**LQR Controller**

DEFINITION. *A linear–quadratic regulator (LQR) is essentially an automated way of finding an appropriate state-feedback controller by minimizing a cost function with weighting factors. The meaning of regulator is to maintain a fixed point.*

The cost function is [6]

1. Infinite-horizon LQR

$$J = \int_0^\infty (x^T Q x + u^T R u + 2x^T N u) dt \tag{43}$$

where $u = -kx$. The cost with optimal control is defined as

$$J^* = x^T \mathrm{sx} \tag{44}$$

Then the solution of LQR is

$$u^* = -(R^{-1}B^T S)x = -kx \tag{45}$$

$S$ satisfies the associated Riccati equation ATS+SA−(SB+N)R−1(BTS+NT)+Q=0

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = 0 \tag{46}$$

2. Finite-horizon LQR

$$J = x^T(T)Q_F x(T) + \int_0^T (x^T Q x + u^T R u + 2x^T N u) dt \tag{47}$$

where $x^T(T)Q_F x(T)$ is called terminal cost, $Q_F$ is the solution of (46), and $S$ satisfies

$$A^T S + SA - (SB + N)R^{-1}(B^T S + N^T) + Q = -\dot{S} \tag{48}$$
$$S(T) = Q_F \tag{49}$$

Above is a final value problem.

**Runge-Kutta Simulation**

Initial value problem:

1. Euler's method

$$\dot{x} = f(t, x(t)) \tag{50}$$
$$\begin{aligned} x_{k+1} &= x_k + h f(t_k, x_k) \\ &= x_k + h f_k \end{aligned} \tag{51}$$

2. Midpoint method

$$x_{k+1} = x_k + h f\left(t_k + \frac{1}{2}h, x_k + \frac{1}{2}h f(t_k, x_k)\right) \tag{52}$$

3. Heun's method

$$\begin{aligned} \tilde{x}_{k+1} &= x_k + h f(t_k, x_k) \\ x_{k+1} &= x_k + \frac{1}{2}h(f(t_k, x_k) + f(t_k + h, \tilde{x}_{k+1})) \end{aligned} \tag{53}$$

4. Second-order general Runge-Kutta method
5. The forth-order Runge-Kutta mehod
   The family of explicit Runge-Kutta methods is

$$x_{k+1} = x_k + h\sum_{i=1}^{s} b_i k_i \tag{54}$$

where

$$\begin{aligned} k_1 &= f(t_k, y_k) \\ k_2 &= f(t_k + c_2 h, y_k + h(a_{21}k_1)) \\ k_3 &= f(t_k + c_3 h, y_k + h(a_{31}k_1 + a_{32}k_2)) \\ &\vdots \\ k_s &= f(t_k + c_s h, y_k + h(a_{s1}k_1 + a_{s2}k_2 + ... + a_{s,s-1}k_{s-1})) \end{aligned}$$

| | | | | | |
|---|---|---|---|---|---|
| 0 | | | | | |
| $c_2$ | $a_{21}$ | | | | |
| $c_3$ | $a_{31}$ | $a_{32}$ | | | |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | | |
| $c_s$ | $a_{s1}$ | $a_{s2}$ | ... | $a_{s,s-1}$ | |
| | $b_1$ | $b_2$ | ... | $b_{s-1}$ | $b_s$ |

**Table 1.** *Butcher tableau*

| | | | | |
|---|---|---|---|---|
| 0 | | | | |
| 1/2 | 1/2 | | | |
| 1/2 | 0 | 1/2 | | |
| 1 | 0 | 0 | 1 | |
| | 1/6 | 1/3 | 1/3 | 1/6 |

**Table 2.** *Rk4 method*

**Trajectory Optimization**

Boundary value problem (bvp4c in Matlab)

$$\begin{aligned} \dot{\vec{x}} &= f(t, \vec{x}) \\ s.t. \qquad A\vec{x}(a) + B\vec{x}(b) &= \vec{c} \end{aligned} \tag{55}$$

Boundary value optimization (fmincon in Matlab)

$$\begin{aligned} \min \quad & J(\vec{x}) \\ s.t. \qquad \dot{\vec{x}} &= f(t, \vec{x}) \\ A\vec{x}(a) + B\vec{x}(b) &= \vec{c} \end{aligned} \tag{56}$$

Transcription methods:

$$\begin{aligned} \hat{J}_0 &= 0 \\ \hat{J}_{k+1} &= \hat{J}_k + h\,w(t_k, \vec{x}_k, u_k) \end{aligned} \tag{57}$$

Augmented State

$$X = \begin{pmatrix} \vec{x} \\ \hat{J} \end{pmatrix} \qquad F = \begin{pmatrix} \vec{f}(t_k, \vec{x}_k, \vec{u}_k) \\ \vec{w}(t_k, \vec{x}_k, \vec{u}_k) \end{pmatrix} \tag{58}$$

1. Single shooting
   Use the simulator to find objective functions; evaluate path constraints at grid points and boundary constraints at $x_0, x_N$.
   Don't use ode45(), instead use fixed step methods.
   Objective function is

$$\vec{J} = h\sum_{k=0}^{N-1} g(\vec{x}_k, \vec{u}_k) \tag{59}$$

   Decision variables are

$$\vec{Z} = (\vec{u}_0, \vec{u}_1, ..., \vec{u}_{N-1}) \tag{60}$$

   The defect is

$$\vec{C} = \vec{x}_N - \vec{x}_F = \vec{0} \tag{61}$$

where $\vec{x}_F$ is given, $\vec{x}_N$ is the output of simulation. It's an equality constraint.

$$\begin{aligned}\frac{\partial \vec{C}}{\partial \vec{Z}} &= \frac{\partial(\vec{x}_N - \vec{x}_F)}{\partial \vec{Z}} \\ &= \frac{\partial \vec{x}_N}{\partial \vec{Z}} \\ &= \begin{pmatrix} \frac{\partial \vec{x}_N}{\partial \vec{u}_0} & \frac{\partial \vec{x}_N}{\partial \vec{u}_1} & \cdots & \frac{\partial \vec{x}_N}{\partial \vec{u}_{N-1}} \end{pmatrix} \end{aligned} \tag{62}$$

start with $N = k+1$,

$$\begin{aligned}\frac{\partial \vec{x}_{k+1}}{\partial \vec{u}_p} &= \frac{\partial}{\partial \vec{u}_p}(\vec{x}_k + h f(\vec{x}_k, \vec{u}_k)) \\ &= \frac{\partial}{\partial \vec{x}_k}(\vec{x}_k + h f(\vec{x}_k, \vec{u}_k))\frac{\partial \vec{x}_k}{\partial \vec{u}_p} + \frac{\partial}{\partial \vec{u}_k}(\vec{x}_k + h f(\vec{x}_k, \vec{u}_k))\frac{\partial \vec{u}_k}{\partial \vec{u}_p} \\ &= (1 + h f_x(k))\frac{\partial \vec{x}_k}{\partial \vec{u}_p} + h f_u(k)\frac{\partial \vec{u}_k}{\partial \vec{u}_p} \end{aligned} \tag{63}$$

Recursive equationo while $k > p$,

$$\begin{cases} \frac{\partial \vec{x}_k}{\partial \vec{u}_p} = 0, & k \leqslant p \\ \frac{\partial \vec{u}_k}{\partial \vec{u}_p} = 1, & k = p \end{cases} \tag{64}$$

because current state is not affected by current or future controls; current control is only dependent on itself. In terms of the cost function

$$\begin{aligned}\frac{\partial \vec{J}}{\partial \vec{u}_p} &= h\sum_{k=0}^{N-1}\left(\frac{\partial}{\partial \vec{u}_p} g(\vec{x}_k, \vec{u}_k)\right) \\ &= h\sum_{k=0}^{N-1}\left(\frac{\partial g(k)}{\partial \vec{x}_k}\cdot\frac{\partial \vec{x}_k}{\partial \vec{u}_p} + \frac{\partial g(k)}{\partial \vec{u}_k}\cdot\frac{\partial \vec{u}_k}{\partial \vec{u}_p}\right) \\ &= h\sum_{k=0}^{N-1}\left(g_x(k)\cdot\frac{\partial \vec{x}_k}{\partial \vec{u}_p} + g_u(k)\cdot\frac{\partial \vec{u}_k}{\partial \vec{u}_p}\right) \end{aligned} \tag{65}$$

2. Multiple shooting

   Decision variables are

$$\vec{Z} = (\vec{x}_0, \vec{x}_1, \bar{x}_2, ..., \bar{x}_{N-1}, \vec{u}_0, \vec{u}_1, ..., \vec{u}_{N-1}) \tag{66}$$

   Assume Euler's method is used,

$$\vec{C} = \begin{pmatrix} \vec{x}_0 + h f(\vec{x}_0, \vec{u}_0) - \vec{x}_1 \\ \vec{x}_1 + h f(\vec{x}_1, \vec{u}_1) - \vec{x}_2 \\ \vdots \\ \vec{x}_{N-1} + h f(\vec{x}_{N-1}, \vec{u}_{N-1}) - \vec{x}_N \end{pmatrix} \tag{67}$$

3. Trapezoidal collocation method

   Approximating the control trajectory and the system dynamics as piecewise linear splines [2], the state trajectory becomes piecewise quadratic splines, and the knot points of the spline are coincident with the collocation points.

$$\vec{C}_k = \vec{x}_k + \frac{h_k}{2}(f(\vec{x}_k, \vec{u}_k) + f(\vec{x}_{k+1}, \vec{u}_{k+1})) - \vec{x}_{k+1} \tag{68}$$

   The objecive function is

$$J = \sum_{k=0}^{N-1}\frac{h_k}{2}(g_k + g_{k+1}) \tag{69}$$

4. Hermite-Simpson collocation method

Approximating the control trajectory and the system dynamics as piecewise quadratic splines, the state trajectory is a cubic Hermite spline, which has a continuous first derivative.

$$\vec{C}_k = \vec{x}_k + \frac{h_k}{6}\Big( f(\vec{x}_k, \vec{u}_k) + f\Big(\vec{x}_{k+\frac{1}{2}}, \vec{u}_{k+\frac{1}{2}}\Big) + f(\vec{x}_{k+1}, \vec{u}_{k+1})\Big) - \vec{x}_{k+1} \tag{70}$$

$$\vec{C}_{k+\frac{1}{2}} = \vec{x}_{k+\frac{1}{2}} - \frac{1}{2}\Big(\vec{x}_k + \vec{x}_{k+\frac{1}{2}}\Big) - \frac{h_k}{8}(f(\vec{x}_k, \vec{u}_k) - f(\vec{x}_{k+1}, \vec{u}_{k+1})) \tag{71}$$

The objecive function is

$$J = \sum_{k=0}^{N-1} \frac{h_k}{6}\Big( g_k + 4g_{k+\frac{1}{2}} + g_{k+1}\Big) \tag{72}$$

5. Orthogonal collocation method
   Decision variables are

$$\underset{\text{Initialstate}}{x_0} \quad , \quad \underset{\text{Stateatcollocationpoints}}{x_1, x_2, ..., x_N} \quad , \quad \underset{\text{Finalstate}}{x_{N+1}}$$

   Time grid (grid or knot points) is then

$$\underset{\text{Initialtime}}{t_0} \quad , \quad \underset{\text{Collocationpoints}}{t_1, t_2, ..., t_N} \quad , \quad \underset{\text{Finaltime}}{t_{N+1}}$$

   QUESTION. *How to find collocation points?*

   ***Answer.*** *Roots of an orthogonal polynomial, eg. Legendre, chebyshev.*

   There are three types of grid points:

   i. Gauss: un-collocated end points $|\cdots\cdots|$ roots of $P_n(\tau)$

   ii. Radav: one collocated end points $|\cdot\cdots\cdots|$ roots of $P_n(\tau) + P_{n-1}(\tau)$

   iii. Lobotto: two collocated end points $|\cdot\cdots\cdots\cdot|$ roots of $\dot{P}_n(\tau)$, [Bnd points]

   Collocation constraints are

$$\begin{cases} 0 = DX - f(X^{LG}) \\ 0 = \underset{\text{changeinstate}}{(x_{N+1} - x_0)} - \underset{\text{integral}\int_0^T f(t)\text{dt}}{W^T f(X^{LG})} \end{cases} \tag{73}$$

   which indicates that change in position is integral of dynamics. In (73),

$$X = \begin{pmatrix} x_0 \\ X^{LG} \end{pmatrix} \quad X^{LG} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}$$

   $W$ is quadrature weights which is computed from collocation points. $D$ is a $N \times (N+1)$ differentiation matrix. To determine $W$, find $\int_0^T f(t)\text{dt}$ (Quadrature), select order and find $[t_i, w_i]$ that satisfies

$$\int_0^T f(t)\text{dt} = W^T F = \sum_{i=1}^{N} w_i f(t_i) \tag{74}$$

Error analysis
   Sources of error are

- NLP: feasibility $(1e-12)$ and optimality $(1e-6)$
- Transcription:
  How good is the discrete model? (function approximation)
  How good is our function approximation? (tolerance for transcription)
- Compute discrepancy between:

Analytic derivative of the spline
Dynamics function along path

$$\varepsilon(t) = \frac{d}{dt}x(t) - f(t, x(t), u(t)) \tag{75}$$

where $\frac{d}{dt}x(t)$ is calculated via ppDer function in Matlab code. $|\varepsilon(t_k)| \leqslant$ tol (fmincon feasible tolerance). For overall estimate,

$$\eta_k = \int_{t_k}^{t_{k+1}} |\varepsilon(t)| dt \tag{76}$$

It indicates how much integrator drift on Segment $k$. In Matlab, use integral() function.

# NONLINEAR CONTROL

## Nonlinear Phenomena

1. Finite escape time
2. Multiple isolated equilibria
3. Limit cycles
4. Subharmonic, harmonic, or almost-periodic oscillations
5. Chaos
6. Multiple modes of behavior

In particular [3], if $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ are analytic functions in a neighborhood of the equilibrium point ($f_1$ and $f_2$ have convergent Taylor series representations), then it's true that if the origin of the linearized state equation is a stable (unstable) node, then, in a small neighborhood of the equilibrium point, the trajectories of the nonlinear state equation will behave like a stable (unstable) node whether or not the eigenvalues of the linearization are distinct.

## Second-Order Systems

A second-order autonomous system is

$$\begin{aligned} \dot{x_1} &= f_1(x_1, x_2) \\ \dot{x_2} &= f_2(x_1, x_2) \end{aligned}$$

## Qualitative Behavior of Linear Systems

The solution of the linear time-invariant system $\dot{x} = Ax$ is

$$x(t) = Me^{J_r t}M^{-1}x_0 \tag{1}$$

where $J_r$ is the real Jordan form of $A$. $M$ satisfies

$$M^{-1}AM = J_r = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}, \quad \begin{pmatrix} \lambda & k \\ 0 & \lambda \end{pmatrix}, \quad \text{or} \quad \begin{pmatrix} \alpha & -\beta \\ \beta & \alpha \end{pmatrix} \tag{2}$$

where $k = 0$ or $1$.

**Case 1 Both eigenvalues are real: $\lambda_1 \neq \lambda_2 \neq 0$.**

In this case, $M = [v_1, v_2]$, where $v_1$ and $v_2$ are the real eigenvectors associated with $\lambda_1$ and $\lambda_2$. The change of coordinates $z = M^{-1}x$ transforms the system into two decoupled equations,

$$\dot{z_1} = \lambda_1 z_1, \quad \dot{z_2} = \lambda_2 z_2 \tag{3}$$

The solution given initial state $(z_{10}, z_{20})$ is

$$z_1(t) = z_{10}e^{\lambda_1 t}, \quad z_2(t) = z_{20}e^{\lambda_2 t} \tag{4}$$

when $\lambda_2 < \lambda_1 < 0$, both $z_1$, and $z_2$ tend to zero as $t \to \infty$. The equilibrium point $x = 0$ is called a stable node.

when $\lambda_2 > \lambda_1 > 0$, both $z_1$, and $z_2$ grow exponentially as $t$ increases. The equilibrium point $x = 0$ is called an unstable node.

when $\lambda_2 < 0 < \lambda_1$, the equilibrium point $x = 0$ is called a saddle.

**Case 2. Complex eigenvalues: $\boldsymbol{\lambda_{1,2} = \alpha \pm j\beta}$.**

The change of coordinates $z = M^{-1}x$ transforms the system into the form,

$$\dot{z_1} = \alpha z_1 - \beta z_2, \quad \dot{z_2} = \beta z_1 + \alpha z_2 \tag{5}$$

Change variables in the polar coordinates

$$r = \sqrt{z_1^2 + z_2^2}, \quad \theta = \tan^{-1}\left(\frac{z_2}{z_1}\right) \tag{6}$$

Now we have two uncoupled first-order differential equations

$$\dot{r} = \alpha r, \quad \dot{\theta} = \beta \tag{7}$$

The solution for a given initial state $(r_0, \theta_0)$ is given by

$$r(t) = r_0 e^{\alpha t}, \quad \theta(t) = \theta_0 + \beta t \tag{8}$$

which define a logarithmic spiral in the $z_1 - z_2$ plane. The equilibrium point $x = 0$ is referred to as a stable focus if $\alpha < 0$, unstable focus if $\alpha > 0$, and center if $\alpha = 0$.

**Case 3. Nonzero multiple eigenvalues: $\boldsymbol{\lambda_1 = \lambda_2 = \lambda \neq 0}$.**

The change of coordinates $z = M^{-1}x$ transforms the system into the form,

$$\dot{z_1} = \lambda z_1 + k z_2, \quad \dot{z_2} = \lambda z_2 \tag{9}$$

whose solution, for a given initial state $(z_{10}, z_{20})$, is given by

$$z_1(t) = e^{\lambda t}(z_{10} + k z_{20} t), \quad z_2(t) = z_{20} e^{\lambda t} \tag{10}$$

the trajectory equation

$$z_1 = z_2\left(\frac{z_{10}}{z_{20}} + \frac{k}{\lambda}\ln\left(\frac{z_2}{z_{20}}\right)\right) \tag{11}$$

The equilibrium point $x = 0$ is a stable node if $\lambda < 0$ and unstable node if $\lambda > 0$.

**Case 4. One or both eigenvalues are zero.**

When $\lambda_1 = 0$ and $\lambda_2 \neq 0$, the change of variable $z = M^{-1}x$ results in

$$\dot{z_1} = 0, \quad \dot{z_2} = \lambda_2 z_2 \tag{12}$$

whose solution, for a given initial state $(z_{10}, z_{20})$, is given by

$$z_1(t) = z_{10}, \quad z_2(t) = z_{20} e^{\lambda_2 t} \tag{13}$$

All trajectories converge to the equilibrium subspace when $\lambda_2 < 0$, and diverge away from it when $\lambda_2 > 0$.

When $\lambda_1 = \lambda_2 = 0$, this is a trivial case where every point in the plane is an equilibrium point.

$$\dot{z_1} = z_2, \quad \dot{z_2} = 0 \tag{14}$$

whose solution is

$$z_1(t) = z_{10} + z_{20} t, \quad z_2(t) = z_{20} \tag{15}$$

Trajectories starting off the equilibrium subspace move parallel to it.

The node, focus, and saddle equilibrium points are said to be structurally stable because they maintain their qualitative behavior under infinitesimally small perturbations $A + \triangle A$, while the center equilibrium point is not structurally stable.

### Limit Cycles

A system oscillates when it has a nontrivial periodic solution

$$x(t+T)=x(t), \quad \forall \quad t\geqslant 0 \tag{16}$$

The linear oscillator is not structurally stable, and the amplitude of oscillation is dependent on the initial conditions. It's possible to build pysical nonlinear oscillators such that

- The nonlinear oscillator is structurally stable.
- The amplitude of oscillation (at steady state) is independent of initial conditions.

### Bifurcation

DEFINITION. *Bifurcation is a change in the equilibrium points or periodic orbits, or in their stability properties, as a parameter is varied. The parameter is called a bifurcation parameter, and the values at which changes occur are called bifurcation points.*

**Saddle-node bifurcation.**

**Transcritical bifurcation.**

**Supercritical/subcritical pitchfork bifurcation.**

**Supercritical/subcritical Hopf bifurcation.**

### Lyapunov Stability

## ESTIMATION

### Scalar Root Solvers

Bracketed methods in the span $[a,b]$:

1. Bisection search

$$c_k = \frac{a_k+b_k}{2} \tag{1}$$

2. False position

$$c_k = b_k - f(b_k)\frac{b_k-a_k}{f(b_k)-f(a_k)} \tag{2}$$

3. Ridder's method

$$\begin{aligned} c_1 &= \frac{a+b}{2} \\ f(a)-2f(c_1)e^Q+f(b)e^{2Q} &= 0 \\ e^Q &= \frac{f(c_1)+\operatorname{sign}(f(b))\sqrt{f(c_1)^2-f(a)f(b)}}{f(b)} \\ c_2 &= c_1+(b-a) \end{aligned}$$

4. Brent's method

Unbounded methods:

1. Newton's method

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{3}$$

2. Secant method

$$\begin{aligned} x_{k+1} &= x_k - f(x_k)\frac{x_k-x_{k-1}}{f(x_k)-f(x_{k-1})} \\ &= \frac{x_{k-1}f(x_k)-x_kf(x_{k-1})}{f(x_k)-f(x_{k-1})} \end{aligned} \tag{4}$$

**Scalar Optimization**

Bracketed methods:

1. Golden section
2. Brent's method

Unbounded methods:

1. Newton's method

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} \tag{5}$$

**Unconstrained Minimization**

Most of this chapter is from [1].

$$min \quad f(x)$$

where $f: R^n \to R$ is convex and twice continuously differentiable (dom $f$ is open). The optimal point $x^*$ satisfies $\nabla f(x^*) = 0$.

**Example.** The general convex quadratic minimization is

$$min \quad \frac{1}{2}x^T P x + q^T x + r \tag{6}$$

One special case of the quadratic minimization problem is the least-square problem

$$min \quad \|Ax - b\|^2 = x^T (A^T A)\, x - 2(A^T b)^T x + b^T b \tag{7}$$

The optimality condition is

$$(A^T A)\, x^* = A^T b \tag{8}$$

DEFINITION. *The objective function is strongly convex on $S$, which means that there exists an $m > 0$ such that $\nabla^2 f(x) \geqslant mI$ for all $x \in S$.*

For $x, y \in S$ we have

$$\begin{aligned} f(y) &= f(x) + \nabla f(x)^T (y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x) \\ &\geqslant f(x) + \nabla f(x)^T (y - x) + \frac{m}{2}\|y - x\|^2 \end{aligned}$$

If we have

$$mI \leqslant \nabla^2 f(x) \leqslant MI \tag{9}$$

the ratio $\kappa = M/m$ is an upper bound on the conditional number of the matrix $\nabla^2 f(x)$, which is the ratio of its largest eigenvalue to its smallest eigenvalue.

**Descent Method**

Descent methods use updates

$$x_{k+1} = x_k + t_k \Delta x_k \tag{10}$$

such that $f(x_{k+1}) < f(x_k)$. Gradient descent method sets $\Delta x_k = -\nabla f(x_k)$, $t_k$ can be determined by

1. Exact line search

$$t_k = \underset{s \geqslant 0}{\operatorname{argmin}} f(x_k + s\Delta x_k) \tag{11}$$

2. Backtracking line search

   Choose $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$, $t = 0$,

$$\text{while } f(x_k + t\Delta x_k) > f(x_k) + \alpha t \nabla f(x_k)^T \Delta x_k, \quad t = \beta t$$

Newton's method set $\Delta x_k = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$, check whether

$$\lambda^2 = \frac{1}{2} \nabla f(x_k)^T \nabla^2 f(x_k)^{-1} \nabla f(x_k) \leqslant \epsilon \tag{12}$$

Choose step size $t_k$ by backtracking line search; update $x_{k+1} = x_k + t_k \Delta x_k$.

**Graph Theory**

DEFINITION. *Adjacency matrix for an* $n-$node *graph* $G = (V, E)$ *where* $V = \{v_1, v_2, ..., v_n\}$ *is an* $n \times n$ *matrix* $A = \{a_{ij}\}$ *where*

$$a_{ij} = \begin{cases} 1 & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

*For a weighted graph with edge weights,*

$$a_{ij} = \begin{cases} w(v_i, v_j) & \text{if } \{v_i, v_j\} \in E \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

DEFINITION. *If* $s$ and $t$ *are the node IDs of the source and target nodes of the* $j$th *edge in* $G$*, then the incidence matrix is defined as*

$$\begin{aligned} I_{sj} &= -1 \\ I_{tj} &= 1 \end{aligned} \tag{15}$$

DEFINITION. *The weighted Laplacian matrix of an undirected graph is defined as*

$$L = \sum_{i=1}^{m} I \operatorname{diag}(w) I^T \tag{16}$$

Non-negativity of the weights implies $L \succeq 0$. Denote the eigenvalues of the Laplacian $L$ as

$$\lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$$

The minimum eigenvalue $\lambda_1 = 0$, while the second smallest eigenvalue $\lambda_2$ is called the algebraic connectivity of $G$. The larger $\lambda_2$ is, the better connected the graph is. $\lambda_2 > 0$ if and only if the graph is connected. The eigenvector $v_2$ associated with $\lambda_2$ is often called the Fiedler vector and widely used in spectral partitioning. Finally, $\lambda_2$ is closely related to a quantity called the isoperimetric number or Cheeger constant of $G$, which measures the degree to which a graph has a bottleneck.

DEFINITION. *A directed graph is weakly connected (or just connected) if the undirected underlying graph by replacing directed edges of the graph with undirected edges is a connected graph (a directed path from* $v_i$ to $v_j$ *or from* $v_j$ to $v_i$ *for every pair of vertices* $\{v_i, v_j\}$*). A directed graph is strongly connected or strong if it contains both a directed path from* $v_i$ to $v_j$ *and a directed path from* $v_j$ to $v_i$*. The strong components are the maximal strongly connected subgraphs.*

The matrix sum $S_k$ is defined to be

$$S_k = I + A + A^2 + ... + A^k \tag{17}$$

If there is a positive integer $k$ such that $S_k$ is positive, then the graph is strongly connected.

DEFINITION. *The indegree of* $i$ *is the number of edges for which* $i$ *is a head. Similarly, the outdegree of* $i$ *is the number of edges for which* $i$ *is a tail. The transition matrix can be obatained by assigning weight to an edge from* $v_i$ to $v_j$ *with the outdegree of vertex* $v_i$ *as* $d_i$

$$w(v_i, v_j) = \frac{1}{d_i} \tag{18}$$

For a strongly connected graph, the transition matrix is column-stochastic. If the weights are assigned with the indegree of vertices, the transition matrix becomes row-stochastic.

**Distributed Gradient Descent**

The goal is to minimize

$$\min \quad \sum_{i=1}^{n} f_i(x) \tag{19}$$

The main iteration is

$$x_i(k+1) = \sum_{j=1}^{n} w_{ij} x_j(k) - \alpha \nabla f_i(x_i(k)), \quad i = 1, 2, ..., n \tag{20}$$

For undirected graphs,

$$x_i(k+1) = \sum_{j=1}^{n} w_{ij} x_j(k) - \eta y_i(k) \tag{21}$$

$$y_i(k+1) = \sum_{j=1}^{n} w_{ij} y_j(k) + (\nabla f_i(k+1) - \nabla f_i(k)) \tag{22}$$

where we can define $r_i(k) = \nabla f(k+1) - \nabla f(k)$ to simplify the equation. In a compact matrix form,

$$x_{k+1} = W x_k - \eta y_k \tag{23}$$

$$y_{k+1} = W y_k + (\nabla f_{k+1} - \nabla f_k) \tag{24}$$

the initial condition is $y_0 = \nabla f_0 = \nabla f(x_0)$; $W = \{w_{\mathrm{ij}}\}$ is a doubly-stochastic matrix. For a directed graph, this algorithm can be modified as

$$x_{k+1} = A x_k - \eta y_k \tag{25}$$

$$y_{k+1} = B\,(y_k + (\nabla f_{k+1} - \nabla f_k)) \tag{26}$$

where $A = \bar{A} \otimes I$, $\bar{A}$ is a row-stochastic matrix, and $B = \bar{B} \otimes I$, $\bar{B}$ is a column-stochastic matrix.

**Proof.** $A$ is irreducible, row-stochastic with positive diagonals then

$$\begin{aligned}
A_\infty &= \lim_{k\to\infty} A^k \\
&= \lim_{k\to\infty} \bar{A}^k \otimes I \\
&= (\mathbf{1}_n \pi_r^T) \otimes I \\
A A_\infty &= (\bar{A} \otimes I)((\mathbf{1}_n \pi_r^T) \otimes I) \\
&= A_\infty \\
A_\infty A_\infty &= ((\mathbf{1}_n \pi_r^T) \otimes I)((\mathbf{1}_n \pi_r^T) \otimes I) \\
&= A_\infty
\end{aligned}$$

We also have

$$\begin{aligned}
(\mathbf{1}_n^T \otimes I) y_k &= (\mathbf{1}_n^T \otimes I)(\bar{B} \otimes I)(y_{k-1} + (\nabla f_k - \nabla f_{k-1})) \\
&= (\mathbf{1}_n^T \bar{B}) \otimes (II)(y_{k-1} + (\nabla f_k - \nabla f_{k-1})) \\
&= (\mathbf{1}_n^T \otimes I) y_{k-1} + (\mathbf{1}_n^T \otimes I)(\nabla f_k - \nabla f_{k-1}) \\
&= (\mathbf{1}_n^T \otimes I)(\bar{B} \otimes I)(y_{k-2} + (\nabla f_{k-1} - \nabla f_{k-2})) + (\mathbf{1}_n^T \otimes I)(\nabla f_k - \nabla f_{k-1}) \\
&= (\mathbf{1}_n^T \otimes I) y_{k-2} + (\mathbf{1}_n^T \otimes I)(\nabla f_k - \nabla f_{k-2}) \\
&= ... \\
&= (\mathbf{1}_n^T \otimes I)(y_0 + \nabla f_k - \nabla f_0)
\end{aligned}$$

because $y_0 = \nabla f_0$, then

$$(\mathbf{1}_n^T \otimes I) y_k = (\mathbf{1}_n^T \otimes I) \nabla f_k \tag{27}$$

The state equation can be expanded as

$$\begin{aligned} x_{k+1} &= Ax_k - \eta y_k \\ &= A\left(Ax_{k-1} - \eta y_{k-1}\right) - \eta y_k \\ &= A^2 x_{k-1} - \eta(A+1)y_k \\ &= \ldots \\ &= A^{k+1}x_0 - \eta(A^k + A^{k-1} + \ldots + A^0)y_k \\ &= A^{k+1}x_0 - \eta\left(\frac{I - A^{k+1}}{I - A}\right)y_k \end{aligned} \tag{28}$$

When $k \to \infty$, sum the geometric series and (28) becomes

$$\begin{aligned} \lim_{k\to\infty} x_{k+1} &= A_\infty x_0 + \ldots \\ &= ((\mathbf{1}_n \pi_r^T) \otimes I)x_0 \\ &= (\mathbf{1}_n(\pi_r^T x_0)) \otimes I \end{aligned} \tag{29}$$

□

**Primal-dual Interior-point Method**

The modified KKT condition is

$$r_t(x,\lambda,\nu) = \begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} = \begin{pmatrix} \nabla f_0(x) + Df(x)^T\lambda + A^T\nu \\ -\text{diag}(\lambda)f(x) - \left(\frac{1}{t}\right)\mathbf{1} \\ Ax - b \end{pmatrix} = 0 \tag{30}$$

where three components are called dual residual, centrality residual, and primal residual, respectively. Use Newton's method to solve these nonlinear equations,

$$\begin{aligned} r_t(y + \Delta y) &\approx r_t(y) + Dr_t(y)\Delta y \\ &= 0 \\ Dr_t(y)\Delta y &= -r_t(y) \end{aligned}$$

In terms of $(x,\lambda,\nu)$, we have

$$\begin{pmatrix} \nabla^2 f_0(x) + \sum_{i=1}^m \lambda_i \nabla^2 f_i(x) & Df(x)^T & A^T \\ -\text{diag}(\lambda)\text{Df}(x) & -\text{diag}(f(x)) & 0 \\ A & 0 & 0 \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta\lambda \\ \Delta\nu \end{pmatrix} = -\begin{pmatrix} r_{\text{dual}} \\ r_{\text{cent}} \\ r_{\text{pri}} \end{pmatrix} \tag{31}$$

**Kalman Filter**

QUESTION. (DISCRETIZATION OF KALMAN FILTER)
*Given the continuous time system*

$$\dot{x} = Ax + Bu \tag{32}$$
$$y = Cx + Du \tag{33}$$

*where A,B,C,D are constant and x,u are functions of time t, obtain a discrete form of the system*

$$x_{k+1} = Fx_k + Gu_k \tag{34}$$
$$y_k = Hx_k + Ju_k \tag{35}$$

*where F,G,H,J are constant.*

**Answer.** Considering the first-order linear ordinary differential equation

$$\dot{y} + p(x)y = q(x) \tag{36}$$

the general solution is

$$y(x) = e^{-\int p(x)dx}\left(\int e^{\int p(x)dx}q(x)dx + C\right) \tag{37}$$

If $y(a)=b$, then the solution is

$$y(x) = e^{-\int_a^x p(\xi)d\xi}\left(\int_a^x e^{\int_a^\xi p(\zeta)d\zeta}q(\xi)d\xi + b\right) \tag{38}$$

Assuming the initial condition of (32) is $x(t=0)=x_0$, then the solution can be represented as

$$\begin{aligned} x(t) &= e^{-\int_0^t p(\xi)d\xi}\left(\int_0^t e^{\int_0^\xi p(\zeta)d\zeta}q(\xi)d\xi + x_0\right) \\ &= e^{-\int_0^t (-A)d\xi}\left(\int_0^t e^{\int_0^\xi (-A)d\zeta}Bu(\xi)d\xi + x_0\right) \\ &= e^{At}\left(\int_0^t e^{-A\xi}Bu(\xi)d\xi + x_0\right) \\ &= e^{At}x_0 + e^{At}\int_0^t e^{-A\xi}Bu(\xi)d\xi \end{aligned} \tag{39}$$

Discretize (39) by setting $t=kT$, and $t=(k+1)T$, where $k\in \boldsymbol{Z}^+$,

$$x(kT) = e^{AkT}x_0 + e^{AkT}\int_0^{kT} e^{-A\xi}Bu(\xi)d\xi \tag{40}$$

$$\begin{aligned} x((k+1)T) &= e^{A(k+1)T}x_0 + e^{A(k+1)T}\int_0^{(k+1)T} e^{-A\xi}Bu(\xi)d\xi \\ &= e^{AT}\left(e^{AkT}x_0 + e^{AkT}\int_0^{kT} e^{-A\xi}Bu(\xi)d\xi\right) + e^{A(k+1)T}\int_{kT}^{(k+1)T} e^{-A\xi}Bu(\xi)d\xi \\ &= e^{AT}x(kT) + \int_{kT}^{(k+1)T} e^{A((k+1)T-\xi)}Bu(\xi)d\xi \end{aligned} \tag{41}$$

where $\xi\in[kT,(k+1)T]$. Applying zero order hold (ZOH) here, which means $u(\xi)=u(kT)$, plus B is constant, then it's obtained

$$x((k+1)T) = e^{AT}x(kT) + \left(\int_{kT}^{(k+1)T} e^{A((k+1)T-\xi)}\,d\xi\right)Bu(kT) \tag{42}$$

Changing variable $\lambda=(k+1)T-\xi$, then $d\xi=-d\lambda$, and (42) is turned to be

$$\begin{aligned} x((k+1)T) &= e^{AT}x(kT) - \left(\int_{-T}^{0} e^{A\lambda}\,d\lambda\right)Bu(kT) \\ &= e^{AT}x(kT) + \left(\int_0^T e^{A\lambda}\,d\lambda\right)Bu(kT) \end{aligned} \tag{43}$$

Define $x_{k+1}=x((k+1)T), x_k=x(kT)$, (43) is rewritten as

$$x_{k+1} = e^{AT}x_k + \left(\int_0^T e^{A\lambda}\,d\lambda\right)Bu_k \tag{44}$$

which means

$$F = e^{AT} \tag{45}$$

$$G = \left(\int_0^T e^{A\lambda}\,d\lambda\right)B \tag{46}$$

where $T$ is the sampling interval.

Further, if A is invertible and constant, (46) can be simplified as

$$\begin{aligned} G &= \left(A^{-1}\int_0^T Ae^{A\lambda}\,d\lambda\right)B \\ &= A^{-1}(e^{AT}-e^{A\cdot 0}) \\ &= A^{-1}(e^{AT}-I)B \end{aligned} \tag{47}$$

The Kalman filter is given by the following equations [5]

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} && (48)\\ K_k &= P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1} && (49)\\ &= P_k^+H_k^TR_k^{-1} \\ x_k^- &= F_{k-1}x_{k-1}^+ + G_{k-1}u_{k-1} && (50)\\ x_k^+ &= x_k^- + K_k(y_k - H_kx_k^-) && (51)\\ P_k^+ &= (I-K_kH_k)P_k^-(I-K_kH_k)^T + K_kR_kK_k^T \\ &= ((P_k^-)^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\ &= (I-K_kH_k)P_k^- && (52) \end{aligned}$$

**Example.** If $Q_{r_1}, Q_{r_2} \sim N(\mu, \delta^2)$, consider two cases:

$$\begin{aligned} y_1 &= 2r_1 \\ y_2 &= r_1 + r_2 \end{aligned}$$

then $Q_{y_1} \sim N(2\mu, 4\delta^2)$, $Q_{y_2} \sim N(2\mu, 2\delta^2)$.

**Partical Filter**

For each particle $x_{k-1}^{(m)}$, sample $q_{k-1}^{(m)}$ from $p_k(q_{k-1})$, then predict

$$x_k^{(m)} = f_{k-1}\big(x_{k-1}^{(m)}, u_{k-1}, q_{k-1}^{(m)}\big)$$

Correction is to first compute likelihood (weight):

$$w_{k+1}^{(m)} = p(y_k - h_k(x_k, u_k))w_k$$

normalize to compute probability:

$$p_k^{(m)} = w_{k+1}^{(m)} / \sum_{m=1}^{M} w_{k+1}^{(m)}$$

This won't work because one particle's weight would converge to 1 and others' to 0. A resampling process is needed. Based on the cumulated probability function of the weights, resample to obtain $M$ particles of uniform probability (Duplicate the particles with high weights and discard the ones with the least weights). Iterate.

**Controllability and Observability**

QUESTION. (CONTROLLABILITY)

*Can you change the value of all of the states of a system independently by changing the system inputs (actuator values)?*

**Answer.** Iterate the dynamics equation

$$\begin{aligned} x_k &= Fx_{k-1} + Gu_{k-1} \\ &= F(Fx_{k-2} + Gu_{k-2}) + Gu_{k-1} \\ &= \dots \\ &= F^kx_0 + F^{k-1}Gu_0 + F^{k-2}Gu_1 + \dots + FGu_{k-2} + Gu_{k-1} \\ &= F^kx_0 + \sum_{i=0}^{k-1} F^iGu_{k-i-1} \\ &= F^kx_0 + \begin{pmatrix} G & \dots & F^{k-2}G & F^{k-1}G \end{pmatrix} \begin{pmatrix} u_{k-1} \\ \vdots \\ u_1 \\ u_0 \end{pmatrix} \end{aligned}$$

# APPENDIX

**Taylor Series**

$$f(x)=\sum_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n=f(a)+\frac{f'(a)}{1!}(x-a)+\frac{f''(a)}{2!}(x-a)^2+... \tag{1}$$

for multiple variables:

$$f(\vec{x}) = f(\vec{a})+(\vec{x}-\vec{a})^T\nabla f(\vec{a})+\frac{1}{2}(\vec{x}-\vec{a})^T\mathbf{H}(\vec{a})(\vec{x}-\vec{a})+... \tag{2}$$

where the Hessian matrix $\mathbf{H}(x)$ is

$$\mathbf{H}(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

$$\mathbf{H}_{i,j} = \frac{\partial^2 f}{\partial x_i \partial x_j} \tag{3}$$

the gradient $\nabla f(x)$ is

$$\nabla f = \frac{\partial f}{\partial x}\mathbf{i}+\frac{\partial f}{\partial y}\mathbf{j}+\frac{\partial f}{\partial z}\mathbf{k} \tag{4}$$

**Kronecker Product**

$$A\otimes B=\begin{pmatrix} a_{11}B & ... & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & ... & a_{mn}B \end{pmatrix} \tag{5}$$

properties:

$$(A\otimes B)(C\otimes D) = (AC)\otimes(BD) \tag{6}$$
$$(A\otimes B)^{-1} = A^{-1}\otimes B^{-1} \tag{7}$$
$$(A\otimes B)^T = A^T\otimes B^T \tag{8}$$

**Determinant and Trace**

Determinant properties:

$$\det(I_n) = 1 \tag{9}$$
$$\det(A^T) = \det(A) \tag{10}$$
$$\det(A^{-1}) = \frac{1}{\det(A)}=\det(A)^{-1} \tag{11}$$
$$\det(AB) = \det(A)\det(B) \tag{12}$$
$$\det(cA) = c^n\det(A) \tag{13}$$

For a triangular matrix ($a_{ij}=0$ whenever $i>j$ or alternatively whenever $i<j$), then

$$\det(A)=\prod_{i=1}^{n}a_{ii} \tag{14}$$

Relation to eigenvalues and trace:

$$\det(A) = \prod_{i=1}^{n}\lambda_i \tag{15}$$
$$\det(A-\lambda I) = 0 \tag{16}$$

For complex matrices $A$,

$$\det(e^A)=e^{\mathrm{tr}(A)} \tag{17}$$

For real matrices $A$,

$$\mathrm{tr}(A) = \ln(\det(e^A)) \tag{18}$$

For a positive definite matrix $A$,

$$\mathrm{tr}(I - A^{-1}) \leqslant \log\det(A) \leqslant \mathrm{tr}(A - I) \tag{19}$$

If $A$ or $D$ is invertible, then block matrices satisfy,

$$\begin{aligned} \det\begin{pmatrix} A & B \\ C & D \end{pmatrix} &= \det(A)\det(D - CA^{-1}B) && (20)\\ &= \det(D)\det(A - BD^{-1}C) && (21)\\ &= \det(AD - BC) && (22) \end{aligned}$$

Trace

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \ldots + a_{\mathrm{nn}} \tag{23}$$

properties:

$$\begin{aligned} \mathrm{tr}(A+B) &= \mathrm{tr}(A) + \mathrm{tr}(B) && (24)\\ \mathrm{tr}(cA) &= c\,\mathrm{tr}(A) && (25)\\ \mathrm{tr}(AB) &= \mathrm{tr}(BA) \\ &\neq \mathrm{tr}(A)\mathrm{tr}(B) && (26)\\ \mathrm{tr}(A \otimes B) &= \mathrm{tr}(A)\mathrm{tr}(B) && (27)\\ \mathrm{tr}(P^{-1}AP) &= \mathrm{tr}(A) && (28)\\ \mathrm{tr}(A^k) &= \sum_{i=1}^{n} \lambda_i^k && (29) \end{aligned}$$

**Norm**

General vector norm:

$$\|v\|_p = \left(\sum_{k=1}^{N} |v_k|^p\right)^{\frac{1}{p}} \tag{30}$$

where $p$ is any positive real value, Inf, or $-$ Inf.

$$\begin{aligned} \|v\|_1 &= \sum_{k=1}^{N} (|v_k|) && (31)\\ \|v\|_\infty &= \max_{1 \leqslant k \leqslant N} (|v_k|) && (32)\\ \|v\|_{-\infty} &= \min_{1 \leqslant k \leqslant N} (|v_k|) && (33) \end{aligned}$$

The maximum absolute column sum of a matrix is defined by

$$\|X\|_1 = \max_{1 \leqslant j \leqslant n} \left(\sum_{i=1}^{m} |a_{ij}|\right) \tag{34}$$

The second-order norm (spectral norm) is defined by

$$\begin{aligned} \|X\|_2 &= \max(\mathrm{svd}(X)) \\ &= \sqrt{\lambda_{\max}(X^H X)} \\ &= \sigma_{\max}(A) && (35) \end{aligned}$$

The maximum absolute row sum of a matrix is defined by

$$\|X\|_\infty = \max_{1 \leqslant i \leqslant m} \left(\sum_{j=1}^{n} |a_{ij}|\right) \tag{36}$$

The Frobenius norm of a matrix is defined by

$$\begin{aligned} \|X\|_F &= \sqrt{\mathrm{tr}(XX^H)} \\ &= \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{ij}|^2} \\ &= \sqrt{\sum_{i=1}^{\min\{m,n\}}\sigma_i^2(A)} \end{aligned} \tag{37}$$

where $X^H$ is the conjugate transpose, $\sigma_i(A)$ are the singualr values of $A$.

DEFINITION. *Let $\lambda_1,...,\lambda_n$ be the eigenvalues of a matrix $A\in\mathbb{C}^{n\times n}$. Then its spectral radius $\rho(A)$ is*

$$\rho(A)=\max\{|\lambda_1|,|\lambda_2|,...,|\lambda_n|\} \tag{38}$$

*The condition number of $A$ can be expressed as $\kappa(A)=\rho(A)\rho(A^{-1})$.*

LEMMA. *If $\|\cdot\|$ is a matrix norm on $M_n$, then, for any $A\in M_n$,*

$$\rho(A)\leqslant\|A\| \tag{39}$$

**Proof.** Let $\lambda$ be an eigenvalue of $A$, $x\neq 0$ be a corresponding eigenvector, we have

$$\begin{aligned} AX &= \lambda X \\ |\lambda|\|X\| &= \|\lambda X\| \\ &= \|AX\| \\ &\leqslant \|A\|\|X\| \\ |\lambda| &\leqslant \|A\| \\ \rho(A) &\leqslant \|A\| \end{aligned}$$

Taking the maximum over all eigenvalues $\lambda$ gives the result. □

LEMMA. *Given $A\in M_n$ and $\varepsilon>0$, there exists a matrix norm $\|\cdot\|$ such that*

$$\|A\|\leqslant\rho(A)+\varepsilon \tag{40}$$

LEMMA. *The spectral norm of a matrix satisfies*

$$\rho(A) = \inf\{\|A\|\} \tag{41}$$

$$\rho(A) = \lim_{k\to\infty}\|A^k\|^{\frac{1}{k}} \tag{42}$$

**Singular Value Decomposition**

SVD is

$$\underset{m\times n}{A} = \underset{m\times m}{U}\,\underset{m\times n}{\Sigma}\,\underset{n\times n}{V^H} \tag{43}$$

where the singular values $\sigma$ are always real and nonnegative even if $A$ is complex. $\sigma$ are on the diagonal of a diagonal matrix $\Sigma$ and the corresponding singular vectors form the columns of two orthogonal matrices $U$ and $V$.

**Example.** Apply SVD to a given matrix $A$

$$A=\begin{pmatrix}1&2&3\\7&4&9\\5&8&2\end{pmatrix}=U\begin{pmatrix}14.8512&0&0\\0&5.5485&0\\0&0&1.2864\end{pmatrix}V^H$$

The eigenvalues of $A$ are

$$A = Q\begin{pmatrix} 13.8397 & 0 & 0 \\ 0 & -1.4108 & 0 \\ 0 & 0 & -5.4289 \end{pmatrix}Q^{-1}$$

Thus we have

$$\rho(A) = 13.8397 < 14.8512 = \|A\|_2$$

## BIBLIOGRAPHY


**[1]** Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

**[2]** Matthew Kelly. An introduction to trajectory optimization: how to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.

**[3]** Hassan Khalil. *Nonlinear Systems*. Prentice Hall, 2001.

**[4]** Katsuhiko Ogata and Yanjuan Yang. *Modern control engineering*, volume 4. Prentice hall India, 2002.

**[5]** Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

**[6]** Russ Tedrake, Ian R Manchester, Mark Tobenkin, and John W Roberts. Lqr-trees: feedback motion planning via sums-of-squares verification. *The International Journal of Robotics Research*, 29(8):1038–1052, 2010.