State-Space – Analysis: Controllability & Observability & Stability

http://elec3004.org

ELEC 3004: **Digital Linear Systems:** Signals & Controls

Dr. Surya Singh

Lecture 11

elec3004@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~elec3004/

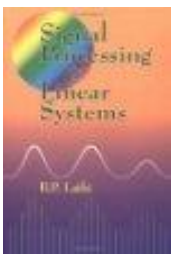
May 18, 2015

Schedule

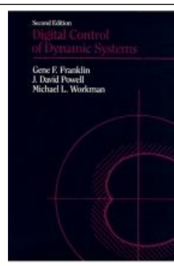
Week	Date	Lecture Title
1	2-Mar	Introduction
	3-Mar	Systems Overview
2	9-Mar	Signals as Vectors & Systems as Maps
	10-Mar	[Signals]
3	16-Mar	Sampling & Data Acquisition & Antialiasing Filters
	17-Mar	[Sampling]
4	23-Mar	System Analysis & Convolution
	24-Mar	[Convolution & FT]
5	30-Mar	Discrete Systems & Z-Transforms
	31-Mar	[Z-Transforms]
6	13-Apr	Frequency Response & Filter Analysis
	14-Apr	[Filters]
7	20-Apr	Digital Filters
	21-Apr	[Digital Filters]
8	27-Apr	Discrete Systems Analysis
	28-Apr	[Feedback]
9	4-May	Introduction to (Digital) Control
	5-May	[Digitial Control]
10	11-May	Digital Control Design
	12-May	[Introduction to State-Space]
11	18-May	State-Space - Analysis
	19-May	[Stability]
12	25-May	Digitial Control Systems: Shaping the Dynamic Response
	26-May	[Applications in Industry]
13	1-Jun	System Identification & [Summary and Course Review]
	2-Jun	Information Theory + Communications

Outline:
(1) Review: PID
(2) Expand on State-Space Representations
(3) Controllability
(4) Observability
(5) Stability

** Additional Reading Materials **

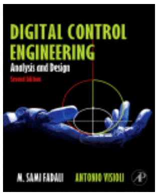


- Chapter 13
  State Space Analysis
  - “s-plane”
  - `tf2ss`



- Chapter 6
  Design of Digital Control Systems Using State-Space Methods
  - Extends to “z-plane”
  - `c2d`

Online Reading Materials

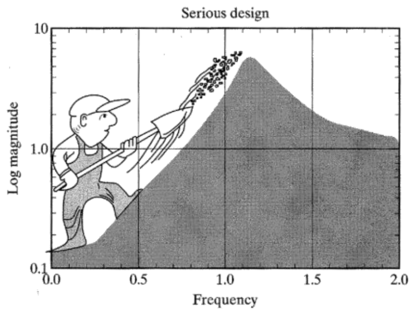


M. Sami Fadali and Antonio Visioli
Digital Control Engineering Analysis & Design (2nd Edition), 2012
Online via ScienceDirect

Chapter 7: State Space Representation
Chapter 9: State Feedback Control

Seeing PID – No Free Lunch

- The energy (and sensitivity) moves around (in this case in "frequency")



- Sensitivity reduction at low frequency unavoidably leads to sensitivity increase at higher frequencies.

Source: Gunter Stein's interpretation of the water bed effect – G. Stein, *IEEE Control Systems Magazine*, 2003.

Another way to see P I|D

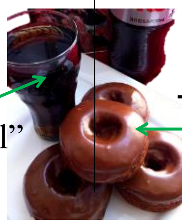
- Derivative

  D provides:
  - High sensitivity
  - Responds to change
  - Adds “damping” & ∴ permits larger K_P
  - Noise sensitive
  - Not used alone (∵ its on rate change of error – by itself it wouldn’t get there)

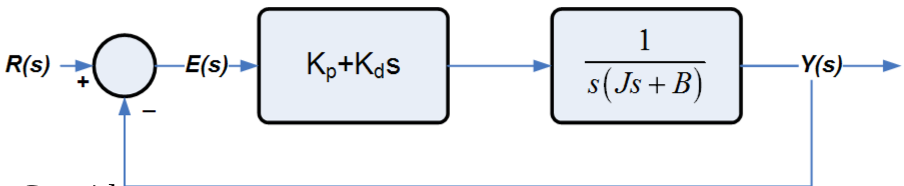
→ “**D**iet Coke of control”

- Integral
  - Eliminates offsets (makes regulation ☺)
  - Leads to Oscillatory behaviour
  - Adds an “order” but instability (Makes a 2nd order system 3rd order)

→ “**I**nteresting cake of control”



PD for 2nd Order Systems



- Consider:

$$\frac{Y(s)}{R(s)} = \frac{(K_P + K_D s)}{Js^2 + (B + K_D)s + K_P}$$

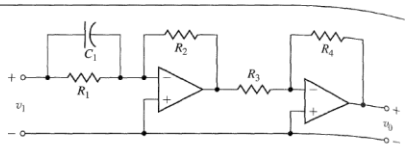
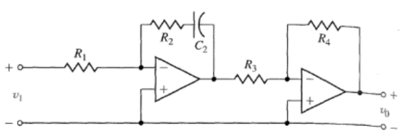
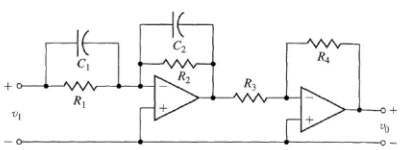
- Steady-state error: $e_{ss} = \frac{B}{K_P}$
- Characteristic equation: $Js^2 + (B + K_D)s + K_P = 0$
- Damping Ratio: $\zeta = \frac{B+K_D}{2\sqrt{K_P J}}$

➔It is possible to make e_{ss} and overshoot small (↓) by making B small (↓), K_P large ↑, K_D such that ζ:between [0.4 – 0.7]

Can Be Used to Make (& Get Over) Speed Bumps

Source: xkcd -- http://what-if.xkcd.com/61/
"How fast can you hit a speed bump while driving and live?"

Operational Amplifier Circuits for Compensators

Type of Controller	$G_c(s) = \frac{V_0(s)}{V_1(s)}$	
PD	$G_c = \frac{R_4R_2}{R_3R_1}(R_1C_1s + 1)$	
PI	$G_c = \frac{R_4R_2(R_2C_2s + 1)}{R_3R_1(R_2C_2s)}$	
Lead or lag Lead if $R_1C_1 > R_2C_2$ Lag if $R_1C_1 < R_2C_2$	$G_c = \frac{R_4R_2(R_1C_1s + 1)}{R_3R_1(R_2C_2s + 1)}$	

- (Yet Another Way to See PID)

Source: Dorf & Bishop, *Modern Control Systems*, p. 828

State-Space Control

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x}$$

(That can not be all of it? There has to be more to it than this…)

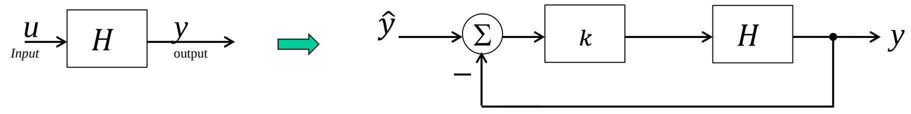
State-Space Control

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}$$

Benefits:

- Characterises the process by systems of coupled, first-order differential equations
- More general mathematical model
  - MIMO, time-varying, nonlinear
- Mathematically esoteric (who needs practical solutions)
- Yet, well suited for digital computer implementation
  - That is: based on vectors/matrices (think LAPACK → MATLAB)

Difference Equations & Feedback



- Start with the Open-Loop:

$$y = Hu$$

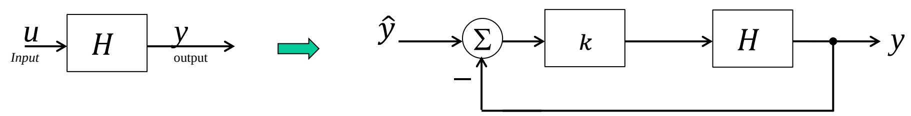
- Close the loop:

$$u = ke = k(\hat{y} - y) \rightarrow y = H[k(\hat{y} - y)]$$

$$\rightarrow \mathrm{y} = \frac{Hk}{1+Hk}\hat{y}$$

- All easy! (yesa!)

Difference Equations & Feedback



- Now add delay (image the plant is a replica with a delay τ)

$$y(t) = u(t - \tau)$$

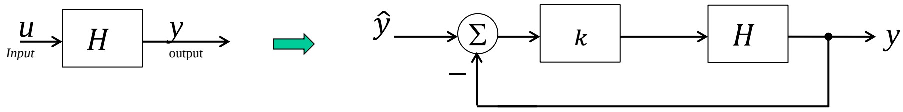
- Close the loop:

$$u(t-\tau) = ke(t-\tau) = k\,[\hat{y}(t-\tau) - \mathrm{y}(t-\tau)]$$

$$\rightarrow \mathrm{y(t)} = k\,[\hat{y}(t-\tau) - \mathrm{y}(t-\tau)]$$

- Notice we have a difference equation!

Difference Equations & Feedback



- What happens with a single delay and a unit step?

$$u(t) = k \text{ for } 0<t<\tau$$

$$y(t) = u(t-\tau) \text{ for } \tau<t<2\tau$$

- Then with feedback we get:

$$u(t) = k(1-k) = k - k^2$$

$$y(t) = k - k^2 + k^3 + \cdots + (-1)^{n-1}k^{n-1}$$

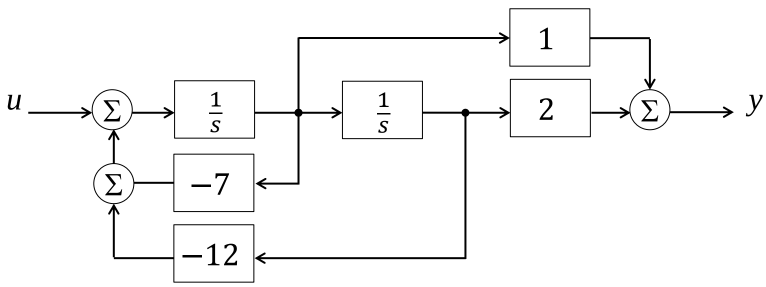
- If k<1: then:

$$\rightarrow \lim y(t) = \frac{k}{1+k}$$

Introduction to state-space

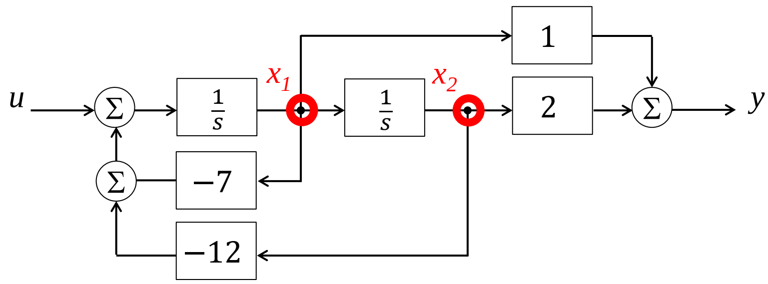
- Linear systems can be written as networks of simple dynamic elements:

$$H = \frac{s+2}{s^2+7s+12} = \frac{2}{s+4} + \frac{-1}{s+3}$$



Introduction to state-space

- We can identify the nodes in the system
  - These nodes contain the integrated time-history values of the system response
  - We call them "states"



Linear system equations

- We can represent the dynamic relationship between the states with a linear system:

$$\begin{aligned} \dot{x_1} &= -7x_1 - 12x_2 + u \\ \dot{x_2} &= x_1 + 0x_2 + 0u \\ y &= x_1 + 2x_2 + 0u \end{aligned}$$

State-space representation

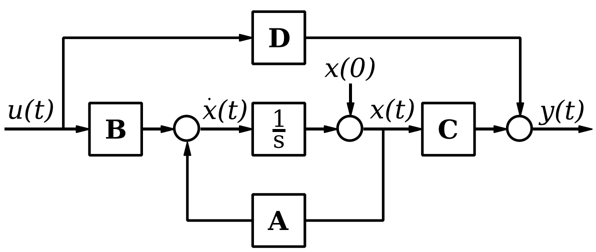
- We can write linear systems in matrix form:

$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix}-7 & 12\\ 1 & 0\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}1\\0\end{bmatrix}u\\ \boldsymbol{y} &= \begin{bmatrix}1 & 2\end{bmatrix}\boldsymbol{x} + 0u\end{aligned}$$

Or, more generally:

$$\left.\begin{aligned}\dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}u\\ y &= \mathbf{C}\boldsymbol{x} + Du\end{aligned}\right\} \text{"State-space equations"}$$

State-Space Terminology



$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t)$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

LTI State-Space

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t)$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

- **If** the system is **linear and time invariant**, **then** A,B,C,D are constant coefficient

$$\rightarrow \dot{x} = Ax + Bu$$
$$\rightarrow y = Cx + Du$$

Discrete Time State-Space

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t)$$
$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

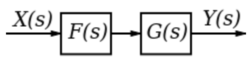
- **If** the system is **discrete**, **then** x and u are given by difference equations

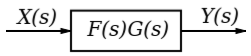
$$\rightarrow x[k+1] = A[k]\,x[k] + B[k]\,u[k]$$
$$y[k] = C[k]\,x[k] + D[k]\,u[k]$$

$$\rightarrow x^{+} = Ax + Bu$$
$$y = Cx + Du$$

Block Diagram Algebra in State Space

- Series:





System 1:

$$x'_F = A_F x_F + B_F u$$
$$y_F = C_F x_F + D_F u$$

System 2:

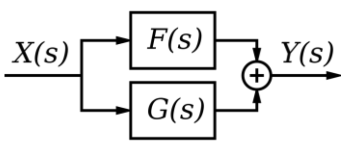
$$x'_G = A_G x_G + B_G y_F$$
$$y_G = C_G x_G + D_G y_F$$

$$\begin{bmatrix} x'_G \\ x'_F \end{bmatrix} = \begin{bmatrix} A_G & B_G C_F \\ 0 & A_F \end{bmatrix} \begin{bmatrix} x_G \\ x_F \end{bmatrix} + \begin{bmatrix} B_G D_F \\ B_F \end{bmatrix} u$$

$$\begin{bmatrix} y_G \\ y_F \end{bmatrix} = \begin{bmatrix} C_G & D_G C_F \\ 0 & C_F \end{bmatrix} \begin{bmatrix} x_G \\ x_F \end{bmatrix} + \begin{bmatrix} D_G D_F \\ D_F \end{bmatrix} u$$

Block Diagram Algebra in State Space

- Parallel:



$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & A_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u$$

$$y = \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + (D_1 + D_2) u$$

State-space representation

- State-space matrices are not necessarily a unique representation of a system
  - There are two common forms

- Control canonical form
  - Each node – each entry in $\boldsymbol{x}$ – represents a state of the system (each order of s maps to a state)

- Modal form
  - Diagonals of the state matrix **A** are the poles (“modes”) of the transfer function

Why is this “Kind of awesome”?

- The controllability of a system depends on the particular set of states you chose

- You can’t tell just from a transfer function whether all the states of $\boldsymbol{x}$ are controllable

- The poles of the system are the Eigenvalues of **F**, (p_i).

State evolution

- Consider the system matrix relation:

$$\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$$
$$y = \mathbf{H}\boldsymbol{x} + Ju$$

The time solution of this system is:

$$\boldsymbol{x}(t) = e^{\mathbf{F}(t-t_0)}\,\boldsymbol{x}(t_0) + \int_{t_0}^{t} = e^{\mathbf{F}(t-\tau)}\,\mathbf{G}u(\tau)d\tau$$

If you didn't know, the matrix exponential is:

$$e^{\mathbf{K}t} = \mathbf{I} + \mathbf{K}t + \frac{1}{2!}\mathbf{K}^2t^2 + \frac{1}{3!}\mathbf{K}^3t^3 + \cdots$$

Controllability matrix

- To convert an arbitrary state representation in **F**, **G**, **H** and *J* to control canonical form **A**, **B**, **C** and *D*, the "controllability matrix"

$$\mathcal{C} = [\mathbf{G} \quad \mathbf{FG} \quad \mathbf{F}^2\mathbf{G} \quad \cdots \quad \mathbf{F}^{n-1}\mathbf{G}]$$

must be nonsingular.

Why is it called the "controllability" matrix?

Controllability matrix

- If you can write it in CCF, then the system equations must be linearly independent.
- Transformation by any nonsingular matrix preserves the controllability of the system.
- Thus, a nonsingular controllability matrix means $\boldsymbol{x}$ can be driven to any value.

Great, so how about control?

- Given $\dot{\boldsymbol{x}} = \mathbf{F}\boldsymbol{x} + \mathbf{G}u$, if we know $\mathbf{F}$ and $\mathbf{G}$, we can design a controller $u = -\mathbf{K}\boldsymbol{x}$ such that
$$\mathrm{eig}(\mathbf{F} - \mathbf{GK}) < 0$$
- In fact, if we have full measurement and control of the states of $\boldsymbol{x}$, we can position the poles of the system in arbitrary locations!

(Of course, that never happens in reality.)

Solving State Space…

- Recall:

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

- For Linear Systems:

$$\dot{x}(t) = A(t)\,x(t) + B(t)\,u(t)$$

$$y(t) = C(t)\,x(t) + D(t)\,u(t)$$

- For LTI:

$$\rightarrow \dot{x} = Ax + Bu$$

$$\rightarrow y = Cx + Du$$

➔ Solutions to State Equations

$$\dot{x} = Ax + Bu$$
$$sX(s) - x(0) = AX(s) + BU(s)$$
$$X(s) = (sI - A)^{-1}x(0) + (sI - A)^{-1}BU(s)$$

$$X(s) = \mathcal{L}[e^{At}]x(0) + \mathcal{L}[e^{At}]BU(s)$$

$$x(t) = \int_0^t e^{At}\,Bu(\tau)d\tau$$

$$\Rightarrow\ e^{At}$$

➔ State-Transition Matrix Φ

- $\Phi(t) = e^{At} = \mathcal{L}^{-1}[(sI - A)^{-1}]$

- It contains all the information about the free motions of the system described by $\dot{x} = Ax$

LTI Properties:

- $\Phi(0) = e^{0t} = I$
- $\Phi^{-1}(t) = \Phi(-t)$
- $\Phi(t_1 + t_2) = \Phi(t_1)\Phi(t_2) = \Phi(t_2)\Phi(t_1)$
- $[\Phi(t)]^n = \Phi(nt)$

➔ The closed-loop poles are the eignvalues of the system matrix

Example: PID control

- Consider a system parameterised by three states:
  - x_1, x_2, x_3
  - where $x_2 = \dot{x}_1$ and $x_3 = \dot{x}_2$

$$\dot{\boldsymbol{x}} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & -2 \end{bmatrix} \boldsymbol{x} - \mathbf{K}u$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \boldsymbol{x} + 0u$$

$x_2$is the output state of the system;
$x_1$is the value of the integral;
x_3 is the velocity.

Example: PID control [2]

- We can choose **K** to move the eigenvalues of the system as desired:

$$\det\begin{bmatrix} 1-K_1 & & \\ & 1-K_2 & \\ & & -2-K_3 \end{bmatrix} = \mathbf{0}$$

All of these eigenvalues must be positive.

It's straightforward to see how adding derivative gain K_3 can stabilise the system.

Digital State Space:

- Difference equations in state-space form:

$$x[n+1] = Ax[n] + Bu[n]$$
$$y[n] = Cx[n] + Du[n]$$

- Where:
  - u[n], y[n]: input & output (scalars)
  - x[n]: state vector

Digital Control Law Design

In Chapter 2, we saw that the state-space description of a continuous system is given by (2.43),

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u, \tag{6.1}$$

and (2.44),

$$y = \mathbf{H}\mathbf{x}. \tag{6.2}$$

We assume the control is applied from the computer by a ZOH as shown in Fig. 1.1. Therefore, (6.1) and (6.2) have an exact discrete representation as given by (2.57),

$$\mathbf{x}(k+1) = \mathbf{\Phi}\mathbf{x}(k) + \mathbf{\Gamma}u(k),$$
$$y(k) = \mathbf{H}\mathbf{x}(k), \tag{6.3}$$

where

$$\boxed{\mathbf{\Phi} = e^{\mathbf{F}T},} \tag{6.4a}$$

$$\mathbf{\Gamma} = \int_0^T e^{\mathbf{F}\eta} d\eta \mathbf{G}, \tag{6.4b}$$

Discretisation FTW!

- We can use the time-domain representation to produce difference equations!

$$\boldsymbol{x}(kT+T) = e^{\mathbf{F}T}\, \boldsymbol{x}(kT) + \int_{kT}^{kT+T} e^{\mathbf{F}(kT+T-\tau)} \mathbf{G} u(\tau) d\tau$$

Notice $\boldsymbol{u}(\tau)$ is not based on a discrete ZOH input, but rather an integrated time-series.

We can structure this by using the form:

$$u(\tau) = u(kT), \qquad kT \le \tau \le kT + T$$

State-space z-transform

We can apply the z-transform to our system:

$$(z\mathbf{I} - \boldsymbol{\Phi})\boldsymbol{X}(z) = \boldsymbol{\Gamma} U(k)$$
$$Y(z) = \mathbf{H}\boldsymbol{X}(z)$$

which yields the transfer function:

$$\frac{Y(z)}{\boldsymbol{X}(z)} = G(z) = \mathbf{H}(z\mathbf{I} - \boldsymbol{\Phi})^{-1}\boldsymbol{\Gamma}$$

State-space control design
¿¿¿Que pasa????

- Design for discrete state-space systems is just like the continuous case.
  - Apply linear state-variable feedback:

$$u = -\mathbf{K}\boldsymbol{x}$$

such that $\det(z\mathbf{I} - \boldsymbol{\Phi} + \boldsymbol{\Gamma}\mathbf{K}) = \alpha_c(z)$

where $\alpha_c(z)$ is the desired control characteristic equation

Predictably, this requires the system controllability matrix

$\boldsymbol{\mathcal{C}} = [\boldsymbol{\Gamma} \quad \boldsymbol{\Phi\Gamma} \quad \boldsymbol{\Phi}^2\boldsymbol{\Gamma} \quad \cdots \quad \boldsymbol{\Phi}^{n-1}\boldsymbol{\Gamma}]$ to be full-rank.

Can you use this for more than Control?

- **Yes**

Frequency Response in State Space

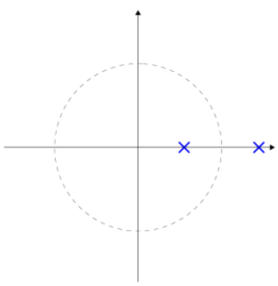
$$H(z) = C(zI - A)^{-1}B + D = \frac{1}{100z^2 - 200z + 80}$$

Poles at $\approx 0.55, 1.45$.

Eigenvalues of A:

$1, 1, 1.45, .55$

What are the (physical) implications?

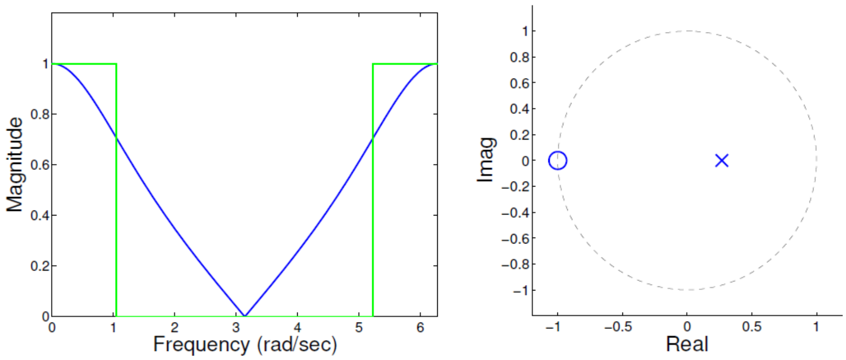


The Approach:

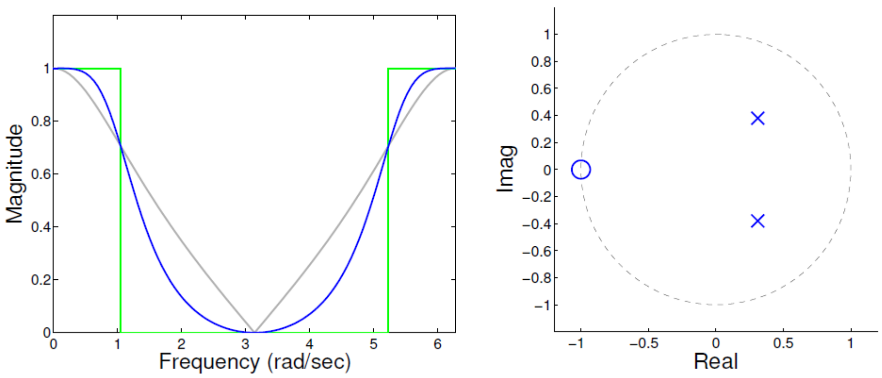
- Formulate the goal of control as an **optimization** (e.g. minimal impulse response, minimal effort, ...).
- You've already seen some examples of optimization-based design:
  - Used least-squares to obtain an FIR system which matched (in the least-squares sense) the desired frequency response.
  - Poles/zeros lecture: Butterworth filter

Discrete Time Butterworth Filters

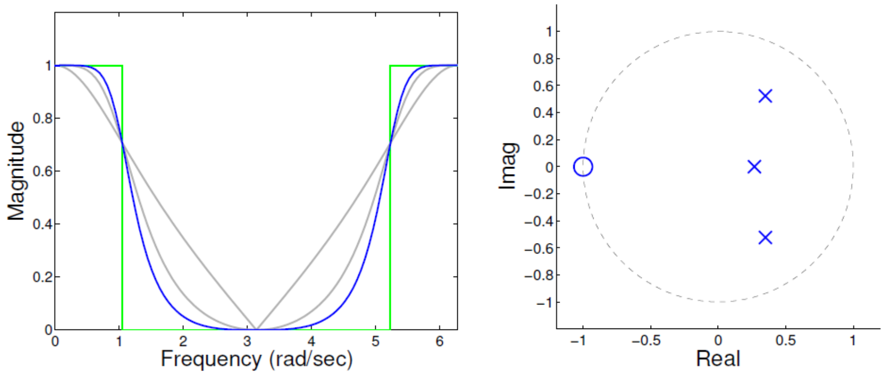
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



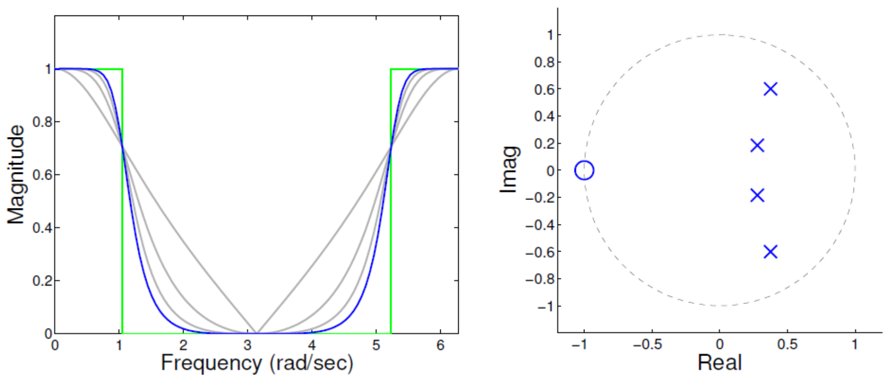
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



ELEC 3004: **Systems**

18 May 2015 - 45

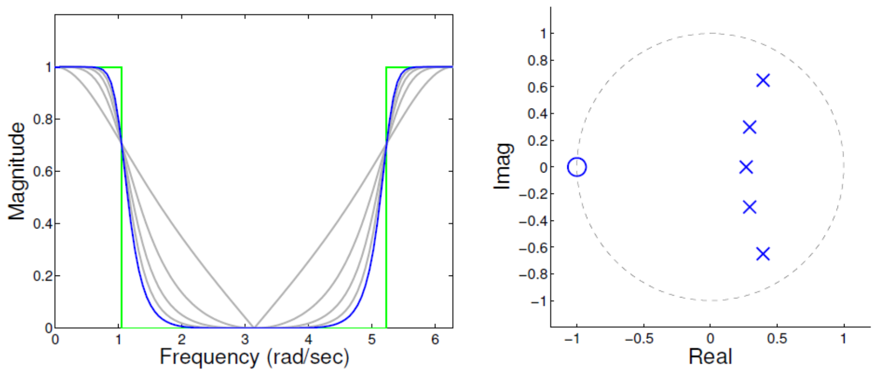
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



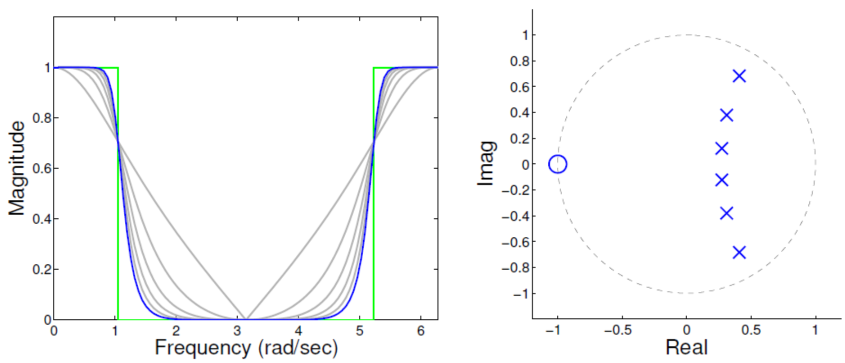
ELEC 3004: **Systems**

18 May 2015 - 46

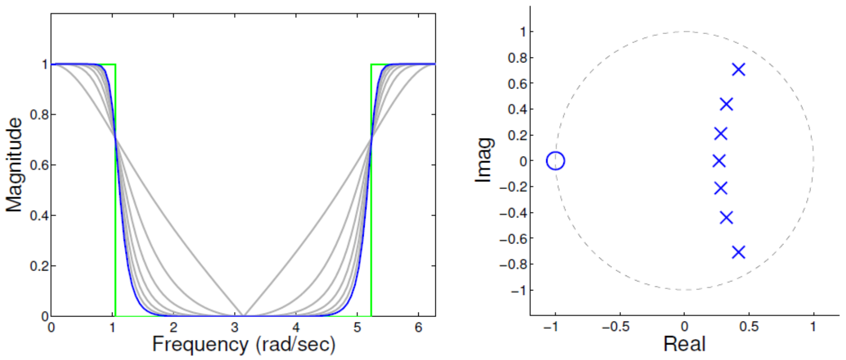
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



“Maximally-flat filter”. Sacrifice sharpness to have flat response in pass band and stop band.



How?

- Constrained Least-Squares …

One formulation: Given $x[0]$

$$\underset{u[0],u[1],\ldots,u[N]}{\text{minimize}} \quad ||\vec{u}||^2, \quad \text{where } \vec{u} = \begin{bmatrix} u[0] \\ u[1] \\ \vdots \\ u[N] \end{bmatrix}$$

$$\text{subject to} \quad x[N] = 0.$$

Note that

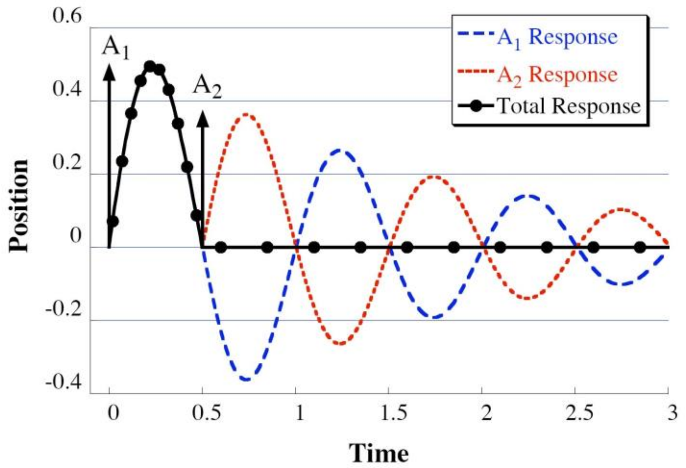
$$x[n] = A^n x[0] + \sum_{k=0}^{n-1} A^{(n-1-k)} B u[k],$$

so this problem can be written as

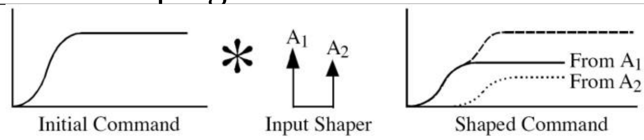
$$\underset{x_{ls}}{\text{minimize}} \, ||A_{ls} x_{ls} - b_{ls}||^2 \quad \text{subject to} \quad C_{ls} x_{ls} = D_{ls}.$$

Example 2:
Command Shaping

Command Shaping



Command Shaping



- Zero Vibration (ZV)

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \dfrac{1}{1+K} & \dfrac{K}{1+K} \\ 0 & \dfrac{T_d}{2} \end{bmatrix} \qquad K = e^{\left(\frac{-\zeta\pi}{\sqrt{1-\zeta^2}}\right)}$$

- Zero Vibration and Derivative (ZVD)

$$\begin{bmatrix} A_i \\ t_i \end{bmatrix} = \begin{bmatrix} \dfrac{1}{(1+K)^2} & \dfrac{2K}{(1+K)^2} & \dfrac{K^2}{(1+K)^2} \\ 0 & \dfrac{T_d}{2} & T_d \end{bmatrix}$$