http://elec3004.com

State-Space

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 20, 2014

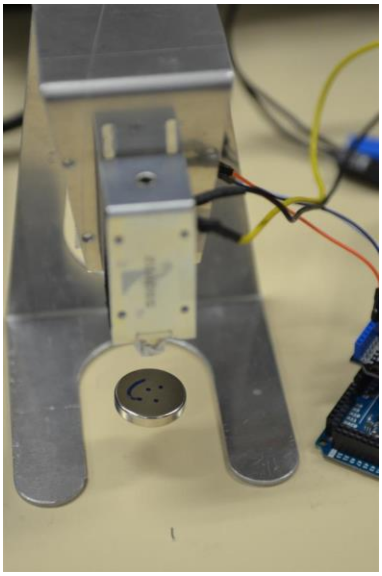
Lecture Schedule:

Week	Date	Lecture Title
1	4-Mar	Introduction & Systems Overview
	6-Mar	[Linear Dynamical Systems]
2	11-Mar	Signals as Vectors & Systems as Maps
	13-Mar	[Signals]
3	18-Mar	Sampling & Data Acquisition & Antialiasing Filters
	20-Mar	[Sampling]
4	25-Mar	System Analysis & Convolution
	27-Mar	[Convolution & FT]
5	1-Apr	Frequency Response & Filter Analysis
	3-Apr	[Filters]
6	8-Apr	Discrete Systems & Z-Transforms
	10-Apr	[Z-Transforms]
7	15-Apr	Introduction to Digital Control
	17-Apr	[Feedback]
8	29-Apr	Digital Filters
	1-May	[Digital Filters]
9	6-May	Digital Control Design
	8-May	[Digitial Control]
10	13-May	Stability of Digital Systems
	15-May	[Stability]
11	20-May	State-Space
	22-May	Controllability & Observability
12	27-May	PID Control & System Identification
	29-May	Digitial Control System Hardware
13	3-Jun	Applications in Industry & Information Theory & Communications
	5-Jun	Summary and Course Review

Announcements:

- Final Exam Logistics:
  - Saturday 21/6/2014 at 4:30pm
  - Location: TBA
  - Closed-book
  - Practice exam will be posted by the end of the week.

Lab 3



Or more aptly…

Welcome to

State-Space!

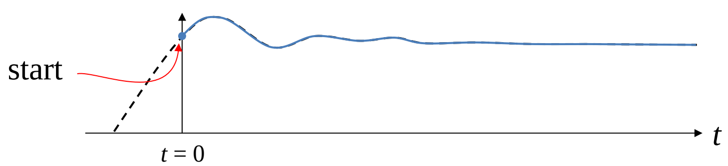
(It be stated -- Hallelujah !)

- More general mathematical model
  - MIMO, time-varying, nonlinear
- Matrix notation (think LAPACK ➔ MATLAB)
- Good for discrete systems
- More design tools!

Affairs of state

- Introductory brain-teaser:
  - If you have a dynamic system model with history (ie. integration) how do you represent the instantaneous state of the plant?

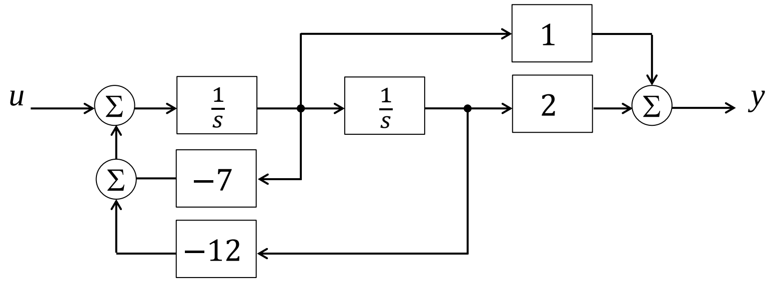
Eg. how would you setup a simulation of a step response, mid-step?



Introduction to state-space

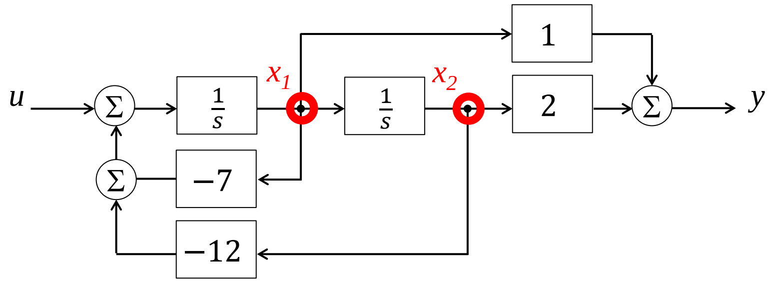
- Linear systems can be written as networks of simple dynamic elements:

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



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:

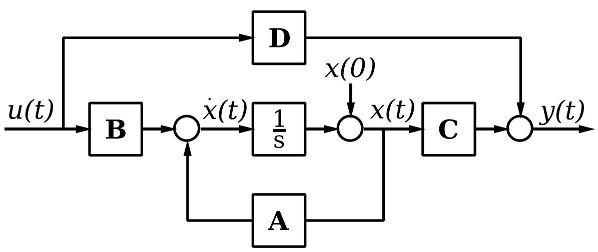
$$\left.\begin{aligned} \dot{\boldsymbol{x}} &= \mathbf{A}\boldsymbol{x} + \mathbf{B}u \\ y &= \mathbf{C}\boldsymbol{x} + Du \end{aligned}\right\} \text{“State-space equations”}$$

State-Space Terminology

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

- **u**: Input: $u : [0, \infty) \to \mathbb{R}^k$
- **x**: State: $x : [0, \infty) \to \mathbb{R}^n$
- **y**: Output $y : [0, \infty) \to \mathbb{R}^m$

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

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

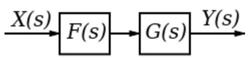
- **If** the system is **discrete**, **then** x and u are given by difference equations

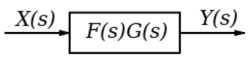
$$\rightarrow x[k+1] = A[k]\,x[k] + B[k]\,u[k]$$
$$y[k] = C[k]\,x[k] + D[k]\,u[k]$$

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

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:

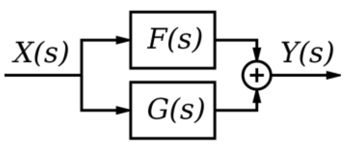
$$x_G' = A_G x_G + B_G y_F$$
$$y_G = C_G x_G + D_G y_F$$

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

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

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 ***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

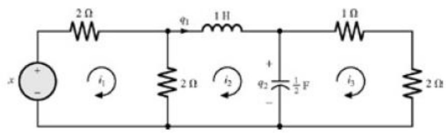
A Procedure for Determining State Equations in **Electrical Circuits**

1. Choose all independent capacitor voltages and inductor currents to be the state variables.

2. Choose a set of loop currents; express the state variables and their first derivatives in terms of these loop currents.

3. Write loop equations, and eliminate all variables other than state variables (and their first derivatives) from the equations derived in steps 2 and 3.

A Quick Example



1. The inductor current q_1 and the capacitor voltage $q2$ as the state variables.

2. $q_1 = i_2$
   $\frac{1}{2}\dot{q}_2 = i_2 - i_3$

3. $4i_1 - 2i_2 = x$
   $2(i_2 - i_1) + \dot{q}_1 + q_2 = 0$
   $-q_2 + 3i_3 = 0$

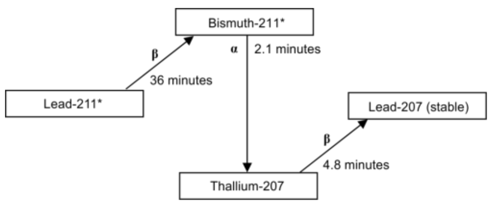
$$\dot{q}_1 = 2(i_1 - i_2) - q_2$$

$$\dot{q}_1 = -q_1 - q_2 + \tfrac{1}{2}x$$

$$\dot{q}_2 = 2q_1 - \tfrac{2}{3}q_2$$

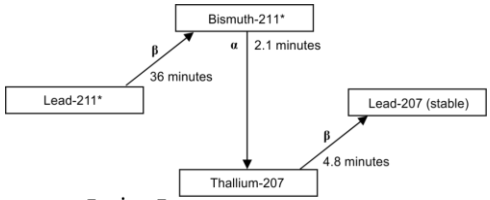
$$\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} = \begin{bmatrix}-1 & -1\\ 2 & -\frac{2}{3}\end{bmatrix}\begin{bmatrix}q_1\\ q_2\end{bmatrix} + \begin{bmatrix}\frac{1}{2}\\ 0\end{bmatrix}x$$

Another Example



- $\frac{dN1(t)}{dt} = -\lambda_1 N1(t)$
- $\frac{dN2(t)}{dt} = -\lambda_2 N2(t) + \lambda_1 N1(t)$
- $\frac{dN3(t)}{dt} = -\lambda_3 N3(t) + \lambda_2 N2(t)$
- $\frac{dN4(t)}{dt} = \lambda_3 N3(t)$

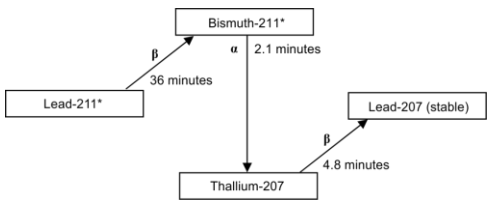
Another Example



$$X = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} \rightarrow \dot{X} = \begin{bmatrix} \dot{N}_1 \\ \dot{N}_2 \\ \dot{N}_3 \\ \dot{N}_4 \end{bmatrix}$$

$$\dot{X} = FX \rightarrow \begin{bmatrix} \dot{N}_1 \\ \dot{N}_2 \\ \dot{N}_3 \\ \dot{N}_4 \end{bmatrix} = \begin{bmatrix} -\lambda_1 & 0 & 0 & 0 \\ \lambda_1 & -\lambda_2 & 0 & 0 \\ 0 & \lambda_2 & -\lambda_3 & 0 \\ 0 & 0 & \lambda_3 & 0 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix}$$

Another Example



- $N_1(t)=N_1(0)\exp(-\lambda_1 t)$
- $N2(t) = N2(0)exp(-\lambda_2 t) - N1(0)\frac{\lambda_1}{\lambda_2-\lambda_1}(exp(-\lambda_2 t) - exp(-\lambda_1 t))$
- $N3(t) = \lambda_1\lambda_2 N1(0)\left[\frac{exp(-\lambda_1 t)}{(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)} + \frac{exp(-\lambda_2 t)}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)} + \frac{exp(-\lambda_3 t)}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)}\right]$
- $N4(t) = \lambda_1\lambda_2\lambda_3 N1(0)\left[\frac{exp(-\lambda_1 t)}{(\lambda_2-\lambda_1)(\lambda_3-\lambda_1)(-\lambda_1)} + \frac{exp(-\lambda_2 t)}{(\lambda_1-\lambda_2)(\lambda_3-\lambda_2)(-\lambda_2)} + \frac{exp(-\lambda_3 t)}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_3)(-\lambda_3)} + \frac{1}{(\lambda_1\lambda_2\lambda_3)}\right]$

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

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 ***x*** can be driven to any value.

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 ***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:

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

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.)

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.

- 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.

Just scratching the surface

- There is a lot of stuff to state-space control

- One lecture (or even two) can't possibly cover it all in depth

Go play with Matlab and check it out!

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:

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

Discretisation FTW!

- Put this in the form of a new variable:

$$\eta = kT + T - \tau$$

Then:

$$\boldsymbol{x}(kT+T) = e^{\boldsymbol{F}T}\boldsymbol{x}(kT) + \left(\int_{kT}^{kT+T} e^{\boldsymbol{F}\eta}\,d\eta\right)\boldsymbol{G}u(kT)$$

Let's rename $\boldsymbol{\Phi} = e^{\boldsymbol{F}T}$ and $\boldsymbol{\Gamma} = \left(\int_{kT}^{kT+T} e^{\boldsymbol{F}\eta}\,d\eta\right)\boldsymbol{G}$

Discrete state matrices

So,

$$\boldsymbol{x}(k+1) = \boldsymbol{\Phi}\boldsymbol{x}(k) + \boldsymbol{\Gamma}u(k)$$
$$y(k) = \mathbf{H}\boldsymbol{x}(k) + \mathbf{J}\boldsymbol{u}(k)$$

Again, $\boldsymbol{x}(k+1)$ is shorthand for $\boldsymbol{x}(kT+T)$

Note that we can also write $\boldsymbol{\Phi}$ as:

$$\boldsymbol{\Phi} = \mathbf{I} + \mathbf{F}T\boldsymbol{\Psi}$$

where

$$\boldsymbol{\Psi} = \mathbf{I} + \frac{\mathbf{F}T}{2!} + \frac{\mathbf{F}^2T^2}{3!} + \cdots$$

State-space z-transform

We can apply the z-transform to our system:

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

which yields the transfer function:

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

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

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

Predictably, this requires the system controllability matrix $\boldsymbol{\mathcal{C}} = [\mathbf{\Gamma} \quad \mathbf{\Phi}\mathbf{\Gamma} \quad \mathbf{\Phi}^2\mathbf{\Gamma} \quad \cdots \quad \mathbf{\Phi}^{n-1}\mathbf{\Gamma}]$ to be full-rank.

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:

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

Solving State Space

- In the conventional, frequency-domain approach the differential equations are converted to transfer functions as soon as possible
  - The dynamics of a system comprising several subsystems is obtained by combining the transfer functions!
- With the state-space methods, on the other hand, the description of the system dynamics in the form of differential equations is retained throughout the analysis and design.

State-transition matrix Φ(t)

- Describes how the state *x(t)* of the system at some time *t* evolves into (or from) the state x(τ) at some other time *T*.

$$x(t) = \Phi(t, \tau)\, x(\tau)$$

Solving State Space…

Time-invariant dynamics The simplest form of the general differential equation of the form (3.1) is the "homogeneous," i.e., unforced equation

$$\dot{x} = Ax \tag{3.2}$$

where A is a constant k by k matrix. The solution to (3.2) can be expressed as

$$x(t) = e^{At}c \tag{3.3}$$

where e^{At} is the matrix exponential function

$$e^{At} = I + At + A^2\frac{t^2}{2} + A^3\frac{t^3}{3!} + \cdots \tag{3.4}$$

and c is a suitably chosen constant vector. To verify (3.3) calculate the derivative of $x(t)$

$$\frac{dx(t)}{dt} = \frac{d}{dt}(e^{At})c \tag{3.5}$$

and, from the defining series (3.4),

$$\frac{d}{dt}(e^{At}) = A + A^2t + A^3\frac{t^2}{2!} + \cdots = A\left(I + At + A^2\frac{t^2}{2!} + \cdots\right) = A\,e^{At}$$

Thus (3.5) becomes

$$\frac{dx(t)}{dt} = Ae^{At}c = Ax(t)$$

Solving State Space

which was to be shown. To evaluate the constant c suppose that at some time τ the state $x(\tau)$ is given. Then, from (3.3),

$$x(\tau) = e^{A\tau}c \tag{3.6}$$

Multiplying both sides of (3.6) by the inverse of $e^{A\tau}$ we find that

$$c = (e^{A\tau})^{-1}x(\tau)$$

Thus the general solution to (3.2) for the state $x(t)$ at time t, given the state $x(\tau)$ at time τ, is

$$x(t) = e^{At}(e^{A\tau})^{-1}x(\tau) \tag{3.7}$$

The following property of the matrix exponential can readily be established by a variety of methods—the easiest perhaps being the use of the series definition (3.4)—

$$e^{A(t_1+t_2)} = e^{At_1}e^{At_2} \tag{3.8}$$

for any t_1 and t_2. From this property it follows that

$$(e^{A\tau})^{-1} = e^{-A\tau} \tag{3.9}$$

and hence that (3.7) can be written

$$x(t) = e^{A(t-\tau)}x(\tau) \tag{3.10}$$

Solving State Space

The matrix $e^{A(t-\tau)}$ is a special form of the *state-transition matrix* to be discussed subsequently.

We now turn to the problem of finding a "particular" solution to the nonhomogeneous, or "forced," differential equation (3.1) with A and B being constant matrices. Using the "method of the variation of the constant,"[1] we seek a solution to (3.1) of the form

$$x(t) = e^{At}c(t) \tag{3.11}$$

where $c(t)$ is a function of time to be determined. Take the time derivative of $x(t)$ given by (3.11) and substitute it into (3.1) to obtain:

$$Ae^{At}c(t) + e^{At}\dot{c}(t) = Ae^{At}c(t) + Bu(t)$$

or, upon cancelling the terms $A\,e^{At}c(t)$ and premultiplying the remainder by e^{-At},

$$\dot{c}(t) = e^{-At}Bu(t) \tag{3.12}$$

Thus the desired function $c(t)$ can be obtained by simple integration (the mathematician would say "by a quadrature")

$$c(t) = \int_T^t e^{-A\lambda}Bu(\lambda)\,d\lambda$$

The lower limit T on this integral cannot as yet be specified, because we will need to put the particular solution together with the solution to the

Solving State Space

homogeneous equation to obtain the complete (general) solution. For the present, let T be undefined. Then the particular solution, by (3.11), is

$$x(t) = e^{At}\int_T^t e^{-A\lambda}Bu(\lambda)\,d\lambda = \int_T^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.13}$$

In obtaining the second integral in (3.13), the exponential e^{At}, which does not depend on the variable of integration λ, was moved under the integral, and property (3.8) was invoked to write $e^{At}e^{-A\lambda} = e^{A(t-\lambda)}$.

The complete solution to (3.1) is obtained by adding the "complementary solution" (3.10) to the particular solution (3.13). The result is

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_T^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.14}$$

We can now determine the proper value for lower limit T on the integral. At $t = \tau$ (3.14) becomes

$$x(\tau) = x(\tau) + \int_T^\tau e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.15}$$

Thus, the integral in (3.15) must be zero for any $u(t)$, and this is possible only if $T = \tau$. Thus, finally we have the complete solution to (3.1) when A and B are constant matrices

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_\tau^t e^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.16}$$

Solving State Space

This important relation will be used many times in the remainder of the book. It is worthwhile dwelling upon it. We note, first of all, that the solution is the sum of two terms: the first is due to the "initial" state $x(\tau)$ and the second—the integral—is due to the input $u(\tau)$ in the time interval $\tau \leqq \lambda \leqq t$ between the "initial" time τ and the "present" time t. The terms initial and present are enclosed in quotes to denote the fact that these are simply convenient definitions. There is no requirement that $t \geqq \tau$. The relationship is perfectly valid even when $t \leqq \tau$.

Another fact worth noting is that the integral term, due to the input, is a "convolution integral": the contribution to the state $x(t)$ due to the input u is the convolution of u with $e^{At}B$. Thus the function $e^{At}B$ has the role of the impulse response[1] of the system whose output is $x(t)$ and whose input is $u(t)$.

If the output y of the system is not the state x itself but is defined by the observation equation

$$y = Cx$$

then this output is expressed by

$$y(t) = Ce^{A(t-\tau)}x(t) + \int_\tau^t Ce^{A(t-\lambda)}Bu(\lambda)\,d\lambda \tag{3.17}$$

Solving State Space

and the impulse response of the system with y regarded as the output is $Ce^{A(t-\lambda)}B$.

The development leading to (3.16) and (3.17) did not really require that B and C be constant matrices. By retracing the steps in the development it is readily seen that when B and C are time-varying, (3.16) and (3.17) generalize to

$$x(t) = e^{A(t-\tau)}x(\tau) + \int_\tau^t e^{A(t-\lambda)}B(\lambda)u(\lambda)\,d\lambda \tag{3.18}$$

and

$$y(t) = C(t)\,e^{A(t-\tau)}x(\tau) + \int_\tau^t C(t)\,e^{A(t-\lambda)}B(\lambda)u(\lambda)\,d\lambda \tag{3.19}$$

Digital State Space: Recall from the Last Episode …

- 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

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

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

Can you use this for more than Control?

- Yes

Frequency Response in State Space

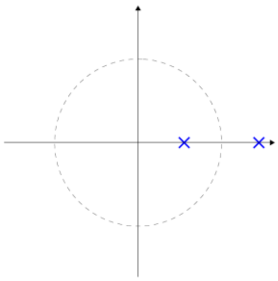
$$H(z) = C(zI - A)^{-1}B + D = \frac{1}{100z^2 - 200z + 80}$$

Poles at $\approx 0.55, 1.45$.

Eigenvalues of A:

$1, 1, 1.45, .55$

What are the (physical) implications?

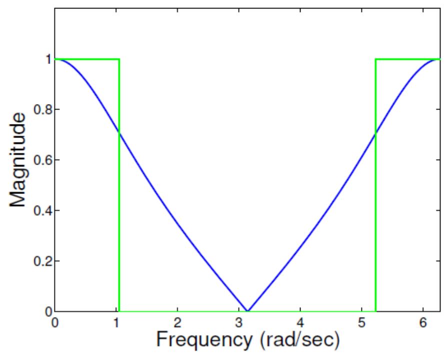


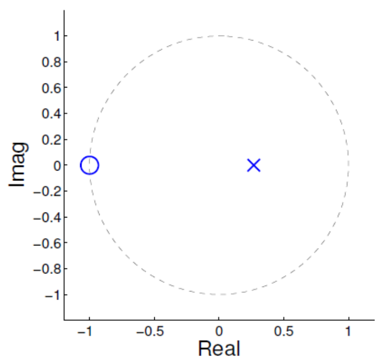
The Approach:

- 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

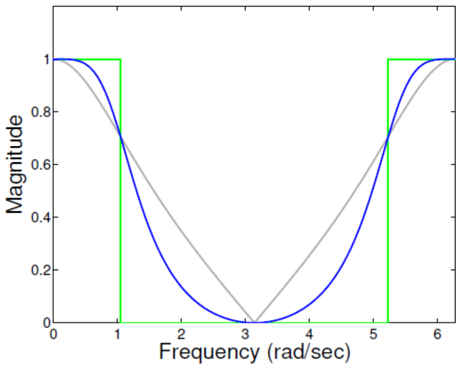
Discrete Time Butterworth Filters

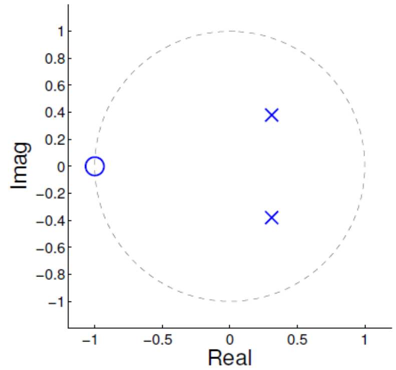
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



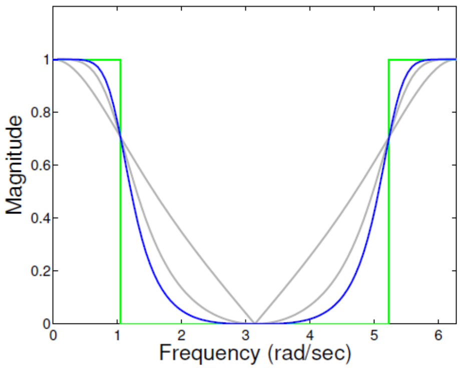


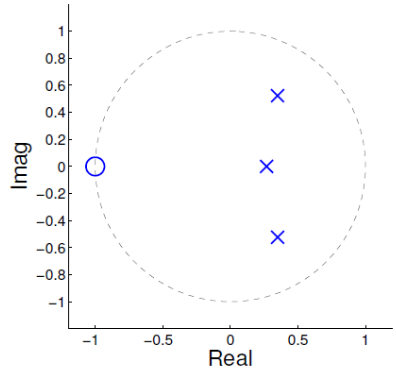
“Maximally-flat filter”. Sacrifice sharpness to have flat response in pass band and stop band.



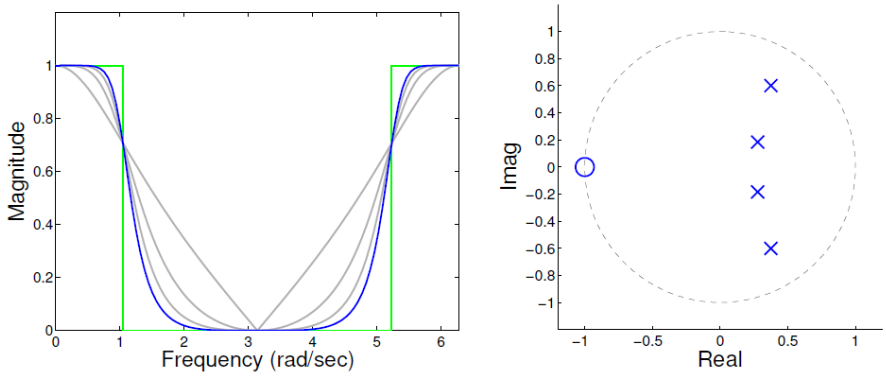


“Maximally-flat filter”. Sacrifice sharpness to have flat response in pass band and stop band.

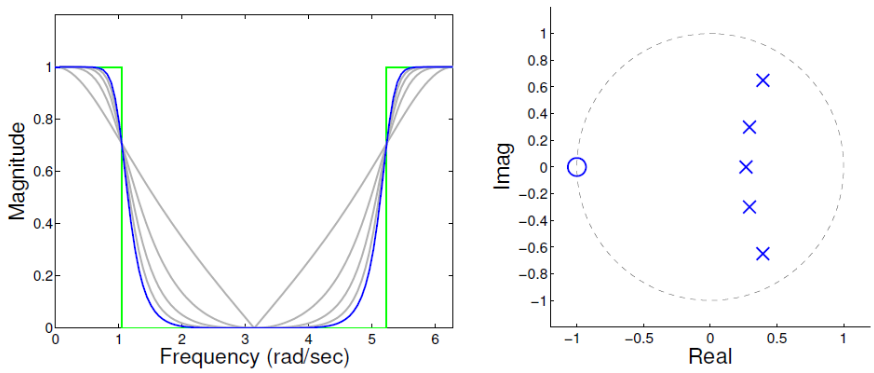




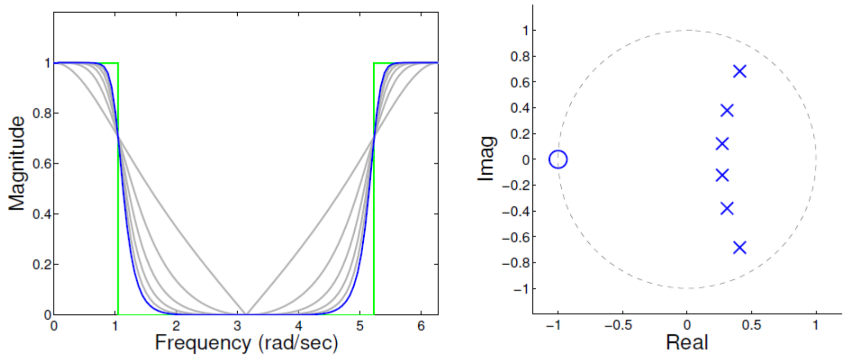
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



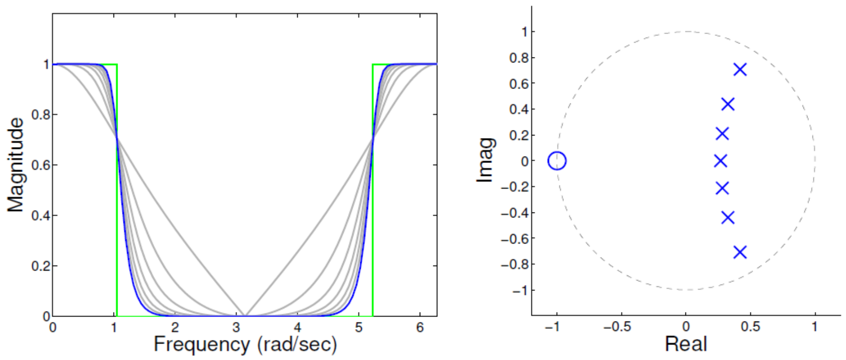
"Maximally-flat filter". Sacrifice sharpness to have flat response in pass band and stop band.



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



“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

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