http://elec3004.com

# Digital Control Design & State-Space


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

Dr. Surya Singh


Lecture 19

(with material from FPW and Cannon (*Discrete Systems*))


elec3004@itee.uq.edu.au

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

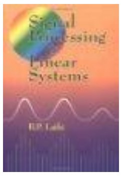

May 16, 2016



## Lecture Schedule:

| Week | Date | Lecture Title |
|---|---|---|
| 1 | 29-Feb | Introduction |
| | 3-Mar | Systems Overview |
| 2 | 7-Mar | Systems as Maps & Signals as Vectors |
| | 10-Mar | Data Acquisition & Sampling |
| 3 | 14-Mar | Sampling Theory |
| | 17-Mar | Antialiasing Filters |
| 4 | 21-Mar | Discrete System Analysis |
| | 24-Mar | Convolution Review |
| | 28-Mar | Holiday |
| | 31-Mar | |
| 5 | 4-Apr | Frequency Response & Filter Analysis |
| | 7-Apr | Filters |
| 6 | 11-Apr | Digital Filters |
| | 14-Apr | Digital Filters |
| 7 | 18-Apr | Digital Windows |
| | 21-Apr | FFT |
| 8 | 25-Apr | Holiday |
| | 28-Apr | Introduction to Feedback Control |
| 9 | 3-May | Holiday |
| | 5-May | Feedback Control & Regulation |
| 10 | 9-May | Servoregulation/PID |
| | 12-May | Introduction to (Digital) Control |
| **11** | 16-May | Digital Control Design & State-Space |
| | 19-May | Observability, Controllability & Stability of Digital Systems |
| 12 | 23-May | Digital Control Systems: Shaping the Dynamic Response & Estimation |
| | 26-May | Applications in Industry |
| 13 | 30-May | System Identification & Information Theory |
| | 2-Jun | Summary and Course Review |

## Follow Along Reading:




**B. P. Lathi**
***Signal processing and linear systems***
**1998**
**TK5102.9.L38 1998**


**G. Franklin,**
**J. Powell,**
**M. Workman**
*Digital Control of Dynamic Systems*
1990

**TJ216.F72 1990**
[Available as UQ Ebook]

Today

➔ **State-space** 🡐 **[A stately idea! ☺]**

- FPW
  - Chapter 4: Discrete Equivalents to Continuous
  - Transfer Functions: The Digital Filter

- Lathi Ch. 13
  - § 13.2 Systematic Procedure for Determining State Equations
  - § 13.3 Solution of State Equations

***Next Time***



## Lab 4 – LeviLab II:
## [AKA "Revenge of the Tuning!"]

- *"not a long time ago in a lab down, down the way"*

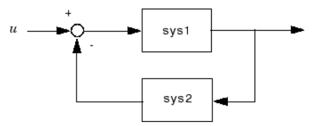
## Effects of Increasing Gain

**Effects of *increasing* a parameter independently**

| Parameter | Rise time | Overshoot | Settling time | Steady-state error | Stability[11] |
|---|---|---|---|---|---|
| $K_p$ | Decrease | Increase | Small change | Decrease | Degrade |
| $K_i$ | Decrease | Increase | Increase | Eliminate | Degrade |
| $K_d$ | Minor change | Decrease | Decrease | No effect in theory | Improve if $K_d$ small |

Matlab helps with PID tuning:

```
G_system=[???]; H=[1];
D_compensator = pidtune(G_system, 'PIDF')
CL_system = feedback(series(D_compensator,G_system), H)
step(CL_system)
```






# PID Review

## PID Control

A continuous PID controller has transfer function:

$$D(s) = \frac{U(s)}{E(s)} = K\left(1 + \frac{1}{T_I s} + T_D s\right)$$

In the time domain, $u(t)$ and $e(t)$ are related by a differential equation:

$$\frac{\mathrm{d}u}{\mathrm{d}t} = K\left[\frac{\mathrm{d}e}{\mathrm{d}t} + \frac{1}{T_I}e + T_D\frac{\mathrm{d}^2 e}{\mathrm{d}t^2}\right]$$

Using Euler's approximation (for 1st and 2nd derivatives)
gives approximate discrete time controller:

$$\frac{u_k - u_{k-1}}{T} = K\left[\frac{e_k - e_{k-1}}{T} + \frac{1}{T_I}e_k + T_D\frac{e_k - 2e_{k-1} + e_{k-2}}{T^2}\right]$$

$$\Longrightarrow \quad u_k = u_{k-1} + K\left[\left(1 + \frac{T}{T_I} + \frac{T_D}{T}\right)e_k - \left(1 + \frac{2T_D}{T}\right)e_{k-1} + \frac{T_D}{T}e_{k-2}\right]$$

i.e. a linear recurrence equation: $u_k = -a_1 u_{k-1} + b_0 e_k + b_1 e_{k-1} + b_2 e_{k-2}$



## PID Control

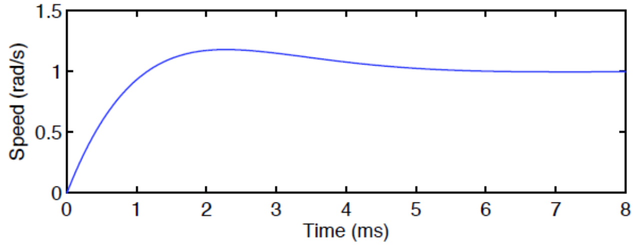
**Example**

Servo-motor speed: $y$ rad s$^{-1}$
input: $u$ Volts

$$\frac{Y(s)}{U(s)} = G(s) = \frac{3.6 \times 10^5}{(s+60)(s+600)}$$

PID controller parameters: $K = 5$, $T_D = 8 \times 10^{-4}$, $T_I = 3 \times 10^{-3}$

Continuous controller step response:




Bandwidth $\approx 1/(4\text{ ms}) = 250$ Hz

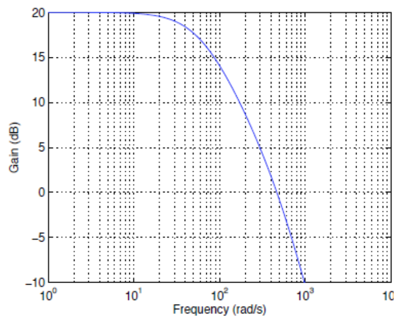
$\Longrightarrow$ need to sample at $30 \times 250 = 7.5$ kHz

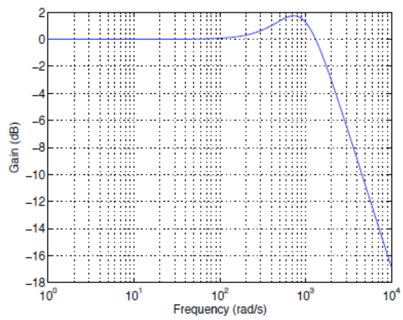
# PID Control

Check the bandwidth estimate . . .

open loop bode magnitude plot, $20 \log|G(j\omega)|$:

closed loop bode magnitude plot, $20 \log\left|\dfrac{G(j\omega)D(j\omega)}{1+G(j\omega)D(j\omega)}\right|$:



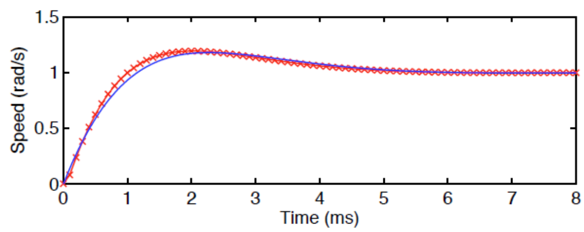




. . . closed loop system bandwidth $= 1.5 \times 10^3 \ \mathrm{rad\,s^{-1}}$
$\approx 250$ Hz

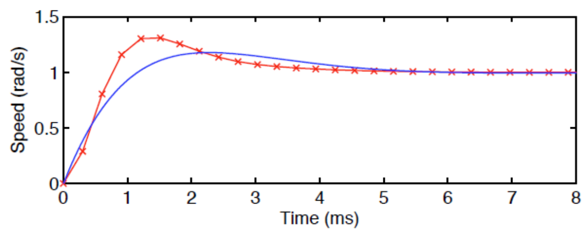


# PID Control

Discrete controller with 10 kHz sample rate:




Discrete controller with 3.3 kHz sample rate:



## PID Control

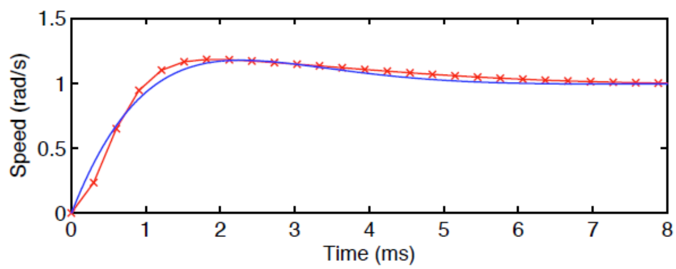
Try to improve response at 3.3 kHz sample rate by changing $K$, $T_D$ & $T_I$

Increase damping by:

- ⋆ reducing $K$
- ⋆ increasing $T_D$

$\Big\}$ e.g. $K = 3.2$, $T_D = 1.1 \times 10^{-3}$ (trial and error)

New response at 3.3 kHz sample rate:






# State-Space

## Or more aptly…

Welcome to

# State-Space!

( It be stated -- Hallelujah ! )

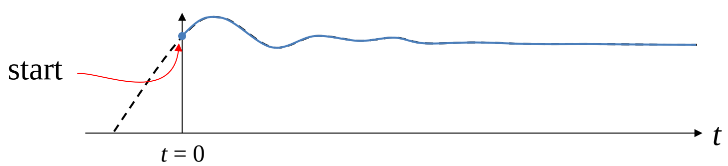
- More general mathematical model
  - MIMO, time-varying, nonlinear
- Matrix notation (think LAPACK ➔ MATLAB)
- Good for discrete systems
- More design tools!



## 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?



## 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…)



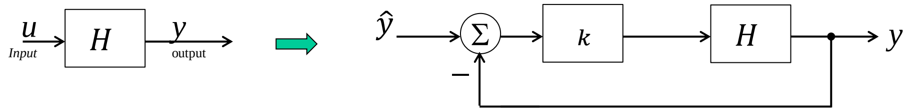
## State-Space Control

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

**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:

$$y = kHu$$

- Close the loop:

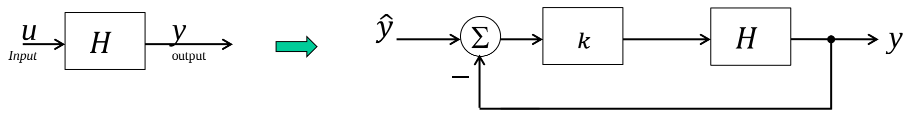
$$u = ke = k(\hat{y} - y) \rightarrow y = H[k(\hat{y} - y)]$$

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

- All easy!   (yesa!)



## Difference Equations & Feedback




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

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

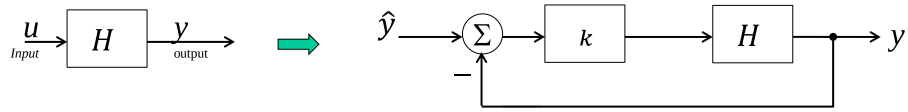
- Close the loop:

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

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

- 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:

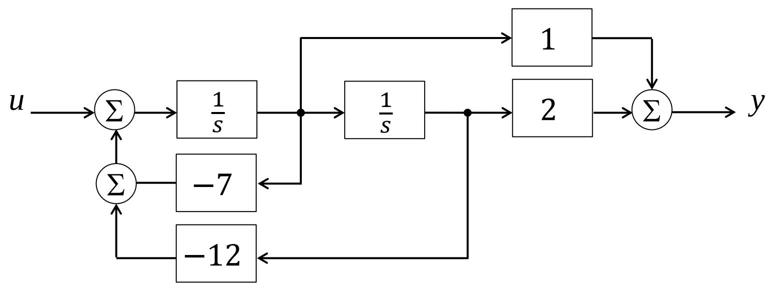
$$\rightarrow \lim y(t) = \frac{k}{1+k}$$



## Introduction to state-space

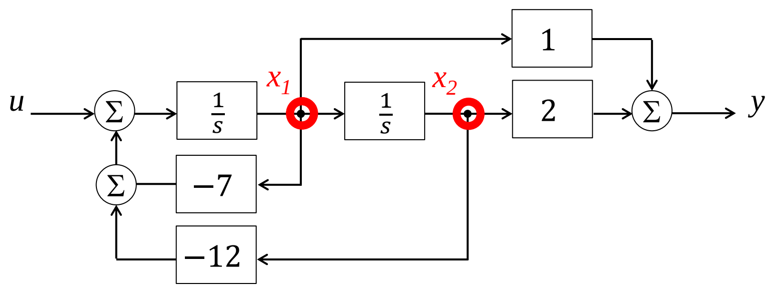
- Linear systems can be written as networks of simple dynamic elements:

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



## 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:

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

## State-space representation

- We can write linear systems in matrix form:

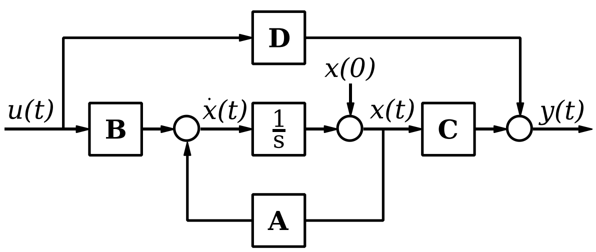
$$\begin{aligned}\dot{\boldsymbol{x}} &= \begin{bmatrix}-7 & 12\\ 1 & 0\end{bmatrix}\boldsymbol{x} + \begin{bmatrix}1\\0\end{bmatrix}u\\ \boldsymbol{y} &= [1 \quad 2]\boldsymbol{x} + 0u\end{aligned}$$

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

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

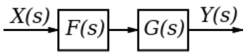
- **If** the system is **discrete**, **then** x and u are given by difference equations

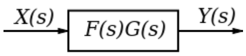
$$\rightarrow x[k+1] = A[k]\,x[k] + B[k]\,u[k]$$
$$y[k] = C[k]\,x[k] + D[k]\,u[k]$$

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

# 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:

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

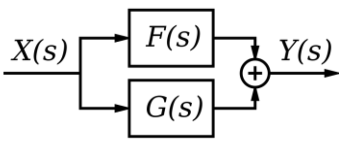
$$\begin{bmatrix} x'_G \\ x'_F \end{bmatrix} = \begin{bmatrix} A_G & B_G C_F \\ 0 & A_F \end{bmatrix} \begin{bmatrix} x_G \\ x_F \end{bmatrix} + \begin{bmatrix} B_G D_F \\ B_F \end{bmatrix} u$$

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



# 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:

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



## 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 $\dot{x} = Ax$

**<u>LTI</u>** 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 $x_3 = \dot{x}_2$

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

$$y = [0 \quad 1 \quad 0]\boldsymbol{x} + 0u$$

$x_2$ 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 $\mathbf{K}$ 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),

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

where

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

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



## A Systematic Procedure for Determining State Eqs.

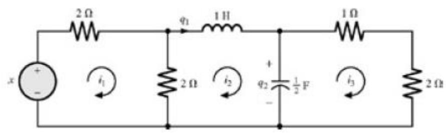
1. Choose all independent capacitor voltages and inductor currents to be the state variables.

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

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

**See also:** Lathi § 13.2-1 (p. 788)

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

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

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

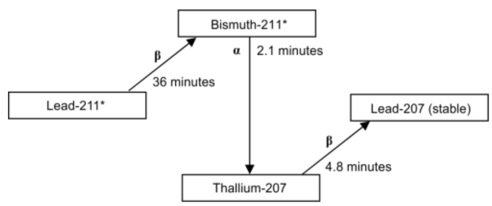
$$\dot{q}_2 = 2q_1 - \tfrac{2}{3}q_2$$

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

**See also:** Fig. 13.2, Lathi p. 789

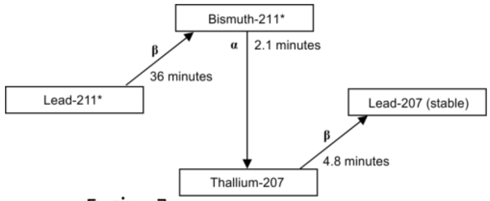


# Another Example




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

# Another Example

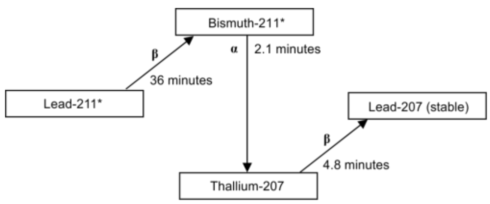



$$X = \begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \end{bmatrix} \rightarrow \dot{X} = \begin{bmatrix} \dot{N}_1 \\ \dot{N}_2 \\ \dot{N}_3 \\ \dot{N}_4 \end{bmatrix}$$

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



# Another Example




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

## Discretisation (FPW!)

- 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:

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



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

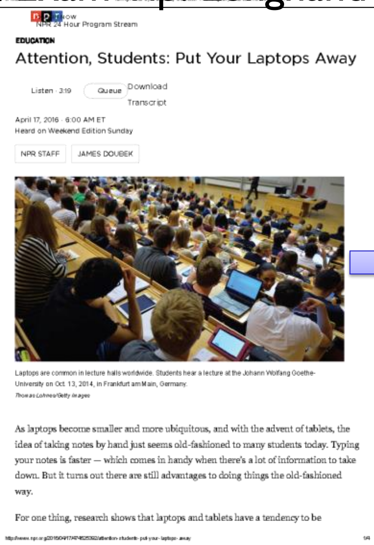
where $\alpha_c(z)$ is the desired control characteristic equation

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.



## Final Exam Tip: Longhand Notes > Typed


NPR 24 Hour Program Stream

EDUCATION

Attention, Students: Put Your Laptops Away

Listen · 3:19 Queue Download Transcript

April 17, 2016 · 6:00 AM ET
Heard on Weekend Edition Sunday

NPR STAFF JAMES DOUBEK

Laptops are common in lecture halls worldwide. Students hear a lecture at the Johann Wolfang Goethe-University on Oct. 13, 2014, in Frankfurt am Main, Germany.
*Thomas Lohnes/Getty Images*

As laptops become smaller and more ubiquitous, and with the advent of tablets, the idea of taking notes by hand just seems old-fashioned to many students today. Typing your notes is faster — which comes in handy when there's a lot of information to take down. But it turns out there are still advantages to doing things the old-fashioned way.

For one thing, research shows that laptops and tablets have a tendency to be


Psychological Science OnlineFirst, published on May 22, 2014 as doi:10.1177/0956797614524581

*Research Article*

**The Pen Is Mightier Than the Keyboard: Advantages of Longhand Over Laptop Note Taking**

Pam A. Mueller[1] and Daniel M. Oppenheimer[2]
[1]Princeton University and [2]University of California, Los Angeles

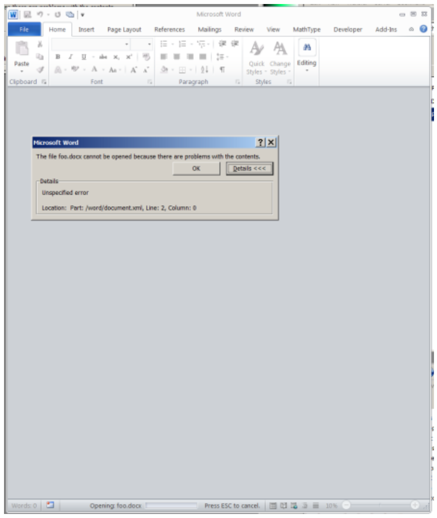
**Abstract**
Taking notes on laptops rather than in longhand is increasingly common. Many researchers have suggested that laptop note taking is less effective than longhand note taking for learning. Prior studies have primarily focused on students' capacity for multitasking and distraction when using laptops. The present research suggests that even when laptops are used solely to take notes, they may still be impairing learning because their use results in shallower processing. In three studies, we found that students who took notes on laptops performed worse on conceptual questions than students who took notes longhand. We show that whereas taking more notes can be beneficial, laptop note takers' tendency to transcribe lectures verbatim rather than processing information and reframing it in their own words is detrimental to learning.

**Keywords**
academic achievement, cognitive processes, memory, educational psychology, open data, open materials

Received 5/11/13; Revision accepted 1/16/14

- http://www.npr.org/2016/04/17/474525392/attention-students-put-your-laptops-away
- doi: 10.1177/0956797614524581

## Friendly computing tale…




- Please save (as) often ☺



## Next Time…

- **Digital Control via Emulation!**

- Review:
  - PID notes online
  - Chapter 5 of FPW

- Deep Pondering??