State-Space Control Design

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 21

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

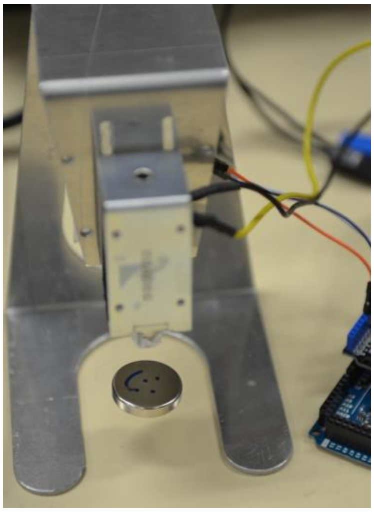
May 17, 2019

http://elec3004.com

Lecture Schedule:

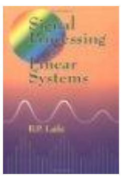
Week	Date	Lecture Title
1	27-Feb	Introduction
	1-Mar	Systems Overview
2	6-Mar	Systems as Maps & Signals as Vectors
	8-Mar	Systems: Linear Differential Systems
3	13-Mar	Sampling Theory & Data Acquisition
	15-Mar	Aliasing & Antialiasing
4	20-Mar	Discrete Time Analysis & Z-Transform
	22-Mar	Second Order LTID (& Convolution Review)
5	27-Mar	Frequency Response
	29-Mar	Filter Analysis
6	3-Apr	Digital Filters (IIR) & Filter Analysis
	5-Apr	PS 1: Q & A
7	10-Apr	Digital Windows
	12-Apr	Digital Filter (FIR)
8	17-Apr	Active Filters & Estimation
	19-Apr	**Holiday**
	24-Apr	
	26-Apr	
9	1-May	Introduction to Feedback Control
	3-May	Servoregulation & PID Control
10	8-May	State-Space Control
	10-May	Guest Lecture: FFT
11	15-May	Advanced PID & & FFT Processes
	17-May	**State Space Control System Design**
12	22-May	Shaping the Dynamic Response
	24-May	Stability and Examples
13	29-May	System Identification & Information Theory & Information Space
	31-May	Summary and Course Review

Next WEEK: Lab 5 – LeviLab *Encore!*

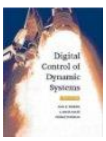


- We will run a combined Lab/Tutorial Session ☺

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]

Today

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

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

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

Next Time

Even More Online Reading Materials


CHAPTER
SIX
SHAPING THE DYNAMIC RESPONSE

6.1 INTRODUCTION


CHAPTER
THREE
DYNAMICS OF LINEAR SYSTEMS

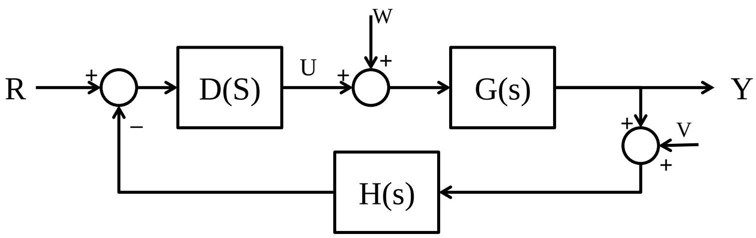
3.1 DIFFERENTIAL EQUATIONS REVISITED

- Friedland, *Control System Design* Ch. 6 and 3

➔ http://robotics.itee.uq.edu.au/~elec3004/tutes.html

State Space

Basic Closed-loop Block Diagram



$$TF(s) = \frac{Y(s)}{R(s)} = \frac{D(s)\,G(s)}{1 + D(s)\,G(s)\,H(s)} = \frac{DG}{1 + DGH}$$

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

➔ Solutions to State Equations

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

$$sX(s) - x(0) = AX(s) + BU(s)$$

$\Phi(s)$

$$X(s) = \underbrace{(sI - A)^{-1}}_{\Phi(s)}x(0) + (sI - A)^{-1}BU(s)$$

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

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

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

➔ 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

Solving State Space

- In the **conventional, frequency-domain approach** [METR4201] 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.

Solving State Space [Extended Reading…]

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

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

Source: Friedland, *Control System Design*

Solving State Space [Extended Reading…]

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

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

Source: Friedland, *Control System Design*

Solving State Space [Extended Reading…]

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 [Extended Reading…]

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 [Extended Reading…]

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

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

Source: Friedland, *Control System Design*

Solving State Space [Extended Reading…]

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

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

Source: Friedland, *Control System Design*

Digital State Space

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

Discretization

- 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:

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

Discretization

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

Digital Control Law Design [Extended Reading…]

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

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

Source: FPW

Discrete state matrices

So,

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

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

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

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

where

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

Simplifying calculation

- We can also use $\mathbf{\Psi}$ to calculate $\mathbf{\Gamma}$
  - Note that:

$$\Gamma = \sum_{k=0}^{\infty} \frac{\mathbf{F}^k T^k}{(k+1)!} T\mathbf{G}$$
$$= \mathbf{\Psi} T \mathbf{G}$$

$\mathbf{\Psi}$ itself can be evaluated with the series:

$$\mathbf{\Psi} \cong \mathbf{I} + \frac{\mathbf{F}T}{2}\left\{\mathbf{I} + \frac{\mathbf{F}T}{3}\left[\mathbf{I} + \cdots \frac{\mathbf{F}T}{n-1}\left(\mathbf{I} + \frac{\mathbf{F}T}{n}\right)\right]\right\}$$

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:

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

∴ State-space Control Design

- 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

Note for Next Class…

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.

State-Space Control: An example!

Given:

$$y(k+1) - y(k) + Ty(k) = x(k+1) - x(k) + 2Tx(k)$$

Simplify:

$$y(k+1) + (T-1)y(k) = x(k+1) + (2T-1)x(k)$$

$$y(k+1) = x(k+1) + (2T-1)x(k) - (T-1)y(k)$$

A quick note on causality!

- Calculating the "(k+1)th" value of a signal using

$$y(k+1) = \underbrace{x(k+1)}_{\text{future value}} + \underbrace{Ax(k) - By(k)}_{\text{current values}}$$

relies on also knowing the next (future) value of $x(t)$!

- Thus, shift it to run with a delay:

$$y(k) = x(k) + Ax(k-1) - By(k-1)$$

"Digital" (Computer) Control Version:

```
T = 0.02;  //period of 50 Hz, a number pulled from thin air
A = 2*T-1; //pre-calculated control constants
B = T-1;

...

while(1)
   {
      if(interrupt_flag)          //this triggers every 20 ms
      {
         x0 = x;                  //save previous values
         y0 = y;
         x = update_input();      //get latest x value
         y = x + A*x0 - B*y0;     //do the difference equations
         update_output(y);        //write out current value
      }
   }
```

(The actual calculation)

Break ☺

Solving State Space: Tustin's Method (Analog Emulation)

Tustin's method

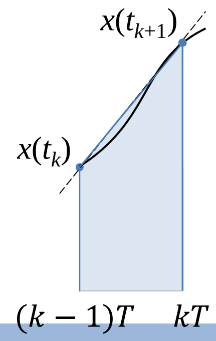
- Tustin uses a trapezoidal integration approximation (compare Euler's rectangles)
- Integral between two samples treated as a straight line:

$$u(kT) = \frac{T}{2}\left[x(k-1) + x(k)\right]$$

Taking the derivative, then z-transform yields:

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

which can be substituted into continuous models



Matched pole-zero

- If $z = e^{sT}$, why can't we just make a direct substitution and go home?

$$\frac{Y(s)}{X(s)} = \frac{s+a}{s+b} \Rightarrow \frac{Y(z)}{X(z)} = \frac{z-e^{-aT}}{z-e^{-bT}}$$

- Kind of!
  - Still an approximation
  - Produces quasi-causal system (hard to compute)
  - Fortunately, also very easy to calculate.

Matched pole-zero

The process:

1. Replace continuous poles and zeros with discrete equivalents:

$$(s + a) \Rightarrow (z - e^{-aT})$$

2. Scale the discrete system DC gain to match the continuous system DC gain

3. If the order of the denominator is higher than the enumerator, multiply the numerator by $(z + 1)$ until they are of equal order*

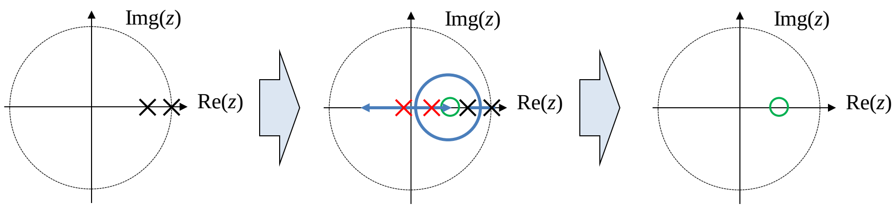
   * This introduces an averaging effect like Tustin's method

Modified matched pole-zero

- We're prefer it if we didn't require instant calculations to produce timely outputs
- Modify step 2 to leave the dynamic order of the numerator one less than the denominator
  - Can work with slower sample times, and at higher frequencies

Discrete design process

1. Derive the dynamic system model ODE
2. Convert it to a discrete transfer function
3. Design a digital compensator
4. Implement difference equations in software
5. Platypus Is Divine!



Discrete design process

- Handy rules of thumb:
  - Sample rates can be as low as twice the system bandwidth
    - but 5 to 10× for “stability”
    - 20 to 30 × for better performance
  - A zero at $z = -1$ makes the discrete root locus pole behaviour more closely match the s-plane
  - Beware “dirty derivatives”
    - dy/dt terms derived from sequential digital values are called ‘dirty derivatives’ – these are especially sensitive to noise!
    - Employ actual velocity measurements when possible

Control Systems Design: `tf2ss`

TF 2 SS – Control Canonical Form)

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1} s + b_n}{s^n + a_1 s^{n-1} + \cdots + a_{n-1} s + a_n}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ \cdot & \cdot & \cdot & & \cdot \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} u$$

$$y = \left[b_n - a_n b_0 \;\vdots\; b_{n-1} - a_{n-1} b_0 \;\vdots\; \cdots \;\vdots\; b_1 - a_1 b_0 \right] \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + b_0 u$$

Modal Form

$$\frac{Y(s)}{U(s)} = \frac{b_0 s^n + b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{(s+p_1)(s+p_2)\cdots(s+p_n)}$$

$$= b_0 + \frac{c_1}{s+p_1} + \frac{c_2}{s+p_2} + \cdots + \frac{c_n}{s+p_n}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} -p_1 & & & & & 0 \\ & -p_2 & & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ 0 & & & & & -p_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} + b_0 u$$

Matlab's tf2ss

- Given: $\frac{Y(s)}{U(s)} = \frac{25.04s+5.008}{s^3+5.03247s^2+25.1026s+5.008}$

  Get a state space representation of this system

- Matlab:

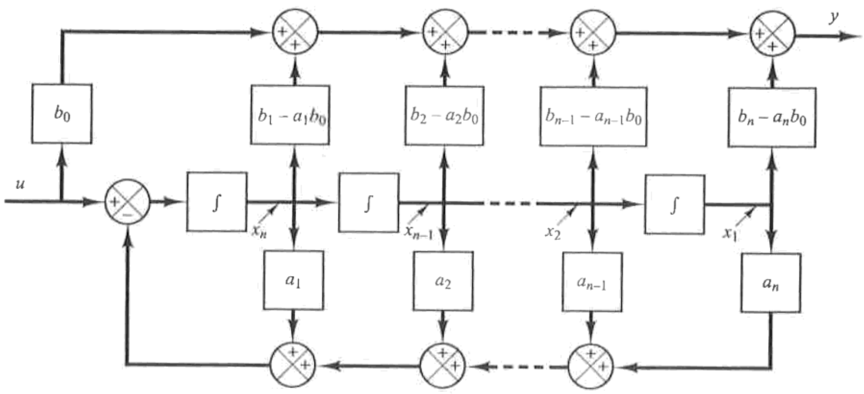
```
num = [25.04 5.008];
den = [1 5.03247 25.1026 5.008];
[A,B,C,D] = tf2ss(num/den);
```

- Answer:

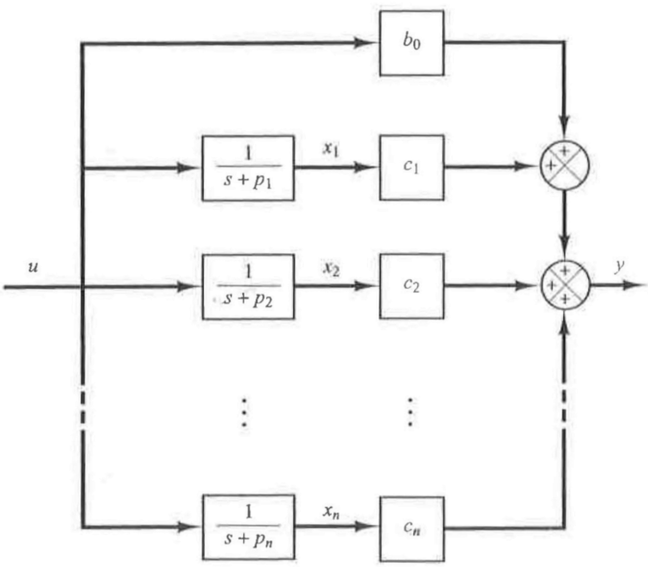
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -5.0325 & -25.1026 & -5.008 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 25.04 & 5.008 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + [0]u$$

Control Canonical Form as a Block Diagram



Modal Form Block Diagram



Control System Design: Obtaining a Time Response

From SS to Time Response — Impulse Functions

- Given: $\dot{x} = Ax + Bu$
- Solution:

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

  – Substituting $t_0 = 0$ into this:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0-) + \int_{0-}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{u}(\tau)\,d\tau$$

  – Write the impulse as: $\mathbf{u}(t) = \delta(t)\mathbf{w}$

  – where **w** is a vector whose components are the magnitudes of **r** impulse functions applied at t=0

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0-) + \int_{0-}^{t} e^{\mathbf{A}(t-\tau)}\mathbf{B}\delta(\tau)\mathbf{w}\,d\tau$$

$$= e^{\mathbf{A}t}\mathbf{x}(0-) + e^{\mathbf{A}t}\mathbf{B}\mathbf{w}$$

From SS to Time Response — Step Response

- Given: $\dot{x} = Ax + Bu$
- Start with $u(t) = \boldsymbol{k}$
  Where **k** is a vector whose components are the magnitudes of *r* step functions applied at t=0.

$$\begin{aligned}\mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\mathbf{k}\,d\tau \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + e^{\mathbf{A}t}\left[\int_0^t \left(\mathbf{I} - \mathbf{A}\tau + \frac{\mathbf{A}^2\tau^2}{2!} - \cdots\right) d\tau\right]\mathbf{B}\mathbf{k} \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + e^{\mathbf{A}t}\left(\mathbf{I}t - \frac{\mathbf{A}t^2}{2!} + \frac{\mathbf{A}^2t^3}{3!} - \cdots\right)\mathbf{B}\mathbf{k}\end{aligned}$$

  – Assume A is non-singular

$$\begin{aligned}\Rightarrow \quad \mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + e^{\mathbf{A}t}\left[-\left(\mathbf{A}^{-1}\right)\left(e^{-\mathbf{A}t} - \mathbf{I}\right)\right]\mathbf{B}\mathbf{k} \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + \mathbf{A}^{-1}\left(e^{\mathbf{A}t} - \mathbf{I}\right)\mathbf{B}\mathbf{k}\end{aligned}$$

From SS to Time Response — Ramp Response

- Given: $\dot{x} = Ax + Bu$
- Start with $u(t) = t\boldsymbol{v}$
  Where **v** is a vector whose components are magnitudes of ramp functions applied at $t = 0$

$$\begin{aligned}\mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}\tau\mathbf{v}\,d\tau \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + e^{\mathbf{A}t}\int_0^t e^{-\mathbf{A}\tau}\tau\,d\tau\,\mathbf{B}\mathbf{v} \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + e^{\mathbf{A}t}\left(\frac{\mathbf{I}}{2}t^2 - \frac{2\mathbf{A}}{3!}t^3 + \frac{3\mathbf{A}^2}{4!}t^4 - \frac{4\mathbf{A}^3}{5!}t^5 + \cdots\right)\mathbf{B}\mathbf{v}\end{aligned}$$

  – Assume A is non-singular

$$\begin{aligned}\Rightarrow \quad \mathbf{x}(t) &= e^{\mathbf{A}t}\mathbf{x}(0) + \left(\mathbf{A}^{-2}\right)\left(e^{\mathbf{A}t} - \mathbf{I} - \mathbf{A}t\right)\mathbf{B}\mathbf{v} \\ &= e^{\mathbf{A}t}\mathbf{x}(0) + \left[\mathbf{A}^{-2}\left(e^{\mathbf{A}t} - \mathbf{I}\right) - \mathbf{A}^{-1}t\right]\mathbf{B}\mathbf{v}\end{aligned}$$

Example: Obtain the Step Response

- Given:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} u, \quad \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad u(t) = 1(t)$$

- Solution:

$$\mathbf{A} = \begin{bmatrix} -1 & -0.5 \\ 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \qquad \mathbf{\Phi}(t) = e^{\mathbf{A}t} = \mathcal{L}^{-1}\left[(s\mathbf{I} - \mathbf{A})^{-1}\right]$$

$$(s\mathbf{I} - \mathbf{A})^{-1} = \begin{bmatrix} s+1 & 0.5 \\ -1 & s \end{bmatrix}^{-1} = \frac{1}{s^2 + s + 0.5} \begin{bmatrix} s & -0.5 \\ 1 & s+1 \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{s + 0.5 - 0.5}{(s+0.5)^2 + 0.5^2} & \dfrac{-0.5}{(s+0.5)^2 + 0.5^2} \\ \dfrac{1}{(s+0.5)^2 + 0.5^2} & \dfrac{s + 0.5 + 0.5}{(s+0.5)^2 + 0.5^2} \end{bmatrix}$$

$$\mathbf{\Phi}(t) = e^{\mathbf{A}t} = \mathcal{L}^{-1}\left[(s\mathbf{I} - \mathbf{A})^{-1}\right] = \begin{bmatrix} e^{-0.5t}(\cos 0.5t - \sin 0.5t) & -e^{-0.5t}\sin 0.5t \\ 2e^{-0.5t}\sin 0.5t & e^{-0.5t}(\cos 0.5t + \sin 0.5t) \end{bmatrix}$$

  - Set k=1, x(0)=0:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \mathbf{A}^{-1}(e^{\mathbf{A}t} - \mathbf{I})\mathbf{B}k$$

$$= \mathbf{A}^{-1}(e^{\mathbf{A}t} - \mathbf{I})\mathbf{B}$$

$$= \begin{bmatrix} 0 & 1 \\ -2 & -2 \end{bmatrix} \begin{bmatrix} 0.5e^{-0.5t}(\cos 0.5t - \sin 0.5t) - 0.5 \\ e^{-0.5t}\sin 0.5t \end{bmatrix}$$

$$= \begin{bmatrix} e^{-0.5t}\sin 0.5t \\ -e^{-0.5t}(\cos 0.5t + \sin 0.5t) + 1 \end{bmatrix}$$

$$y(t) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1 = e^{-0.5t}\sin 0.5t$$

Example II: Obtain the Step Response

- Given:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \qquad u(t) = 1(t)$$

- Solution:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\mathbf{\Phi}(t) = e^{\mathbf{A}t} = \mathcal{L}^{-1}\left[(s\mathbf{I} - \mathbf{A})^{-1}\right]$$

$$\mathbf{\Phi}(t) = e^{\mathbf{A}t} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix}$$

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}(0) + \int_0^t \begin{bmatrix} 2e^{-(t-\tau)} - e^{-2(t-\tau)} & e^{-(t-\tau)} - e^{-2(t-\tau)} \\ -2e^{-(t-\tau)} + 2e^{-2(t-\tau)} & -e^{-(t-\tau)} + 2e^{-2(t-\tau)} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} [1]\, d\tau$$

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} \frac{1}{2} - e^{-t} + \frac{1}{2}e^{-2t} \\ e^{-t} - e^{-2t} \end{bmatrix}$$

  - Assume x(0)=0:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} + \begin{bmatrix} \frac{1}{2} - e^{-t} + \frac{1}{2}e^{-2t} \\ e^{-t} - e^{-2t} \end{bmatrix}$$

The Direct Method of Digital Controls –

NOT to be confused with
Controller Emulation
(e.g., Tustin's Method)

Direct Design Method Of Ragazzini (See also: FPW 5.7 pp.216-222)

Start with 3 Discrete Transfer Functions:

- **G(z)**: TF[1] of a **plant + a hold** (e.g., from a ZOH)
- **D(z)**: A **controller** TF to do the job (what we want here)
- **H(z)**: The final desired TF between **R** (reference) and **Y** (output)
- Thus[2]:

$$\boxed{\begin{aligned} H(z) &= \frac{DG}{1+DG} \\ \rightarrow D(z) &= \frac{1}{G}\frac{H}{1-H} \end{aligned}}$$

- This calls for a **D(z)** that will **cancel the plant effects** and that will **add whatever is necessary to give the desired result.** The problem is to discover and implement constraints on **H(z)** so that we do not ask for the impossible.
  - This implies that we **need some constraints** on both **H(z)** and **D(z)**

1: Transfer Function

2: Mental Quiz: What does **1+DG** say about the sign of the feedback (positive or negative)? That is, what is the characteristic equation for a system with positive feedback?

Direct Design Method Of Ragazzini [2]: Design Constraints: I. Causality

- Remember/Recall an Interesting Point:
  - From z-transform theory we know that **if D(z)** is causal, **then** as $z \to \infty$ its transfer function is well behaved & it **does not** have a pole at infinity.

- $D(z) = \frac{1}{G}\frac{H}{1-H}$ implies that **if** G**(z)** = 0 (at ∞), **then D(z)** would have a pole (at ∞) **unless H(z)** cancels it.

∴

H(z) must have a zero (at ∞) of the same order as G**(z)**'s 0s (at ∞)

→ Which means: **If** there is a lag in the plant (**G(z) starts with** z^{-l}) **then** causality requires that the delay of **H(z)** is that the closed-loop system must be at least as long a delay of the plant.

(Whoa! It might sound deep, but it's rather intuitive ☺)

Direct Design Method Of Ragazzini [3]: Design Constraints: II. Stability

- The characteristic equation and the closed loop roots:

$$1 + D(z)G(z) = 0$$

- Define[3] $D = \frac{c}{d}$ and G= $\frac{b}{a}$ ➔ $ad + bc = 0$
- Define $z - \alpha$ as a pole of **G(z)** and a common factor in DG that represents **D(z)** cancelling a pole/zero of **G(z)**.
- **Then** this common factor *remains a factor of the characteristic polynomial.*
- **If** this factor is outside the unit circle, **then** the system is unstable!

∴

1-H(z) must contain as zeros
all the poles of **G(z)** that are outside the unit circle &
H(z) must contain as zeros
all the zeros of G(z) that are outside the unit circle

3: Note the switching of the "alphabetical-ness" of these two fractions

Direct Design Method Of Ragazzini [4]: Design Constraints: III. Steady State Accuracy

- The error from **H(z)** is given by:

$$\mathrm{E(z)} = \mathrm{R(z)}\big(1 - \mathrm{H(z)}\big)$$

- **If** the system is “Type 1” (with a constant velocity/first derivative (K_v)
  - **Then**[4] $E_{ss}^{Step} = 0$ and $E_{ss}^{Ramp} = {}^{1}/_{K_v}$

$$\therefore$$

$$\mathrm{H(z)} = 1$$

$$\&$$

$$-\mathrm{T_s}\left.\frac{dH(z)}{d(z)}\right|_{z=1} = \frac{1}{K_v}\mathrm{H(z)} = 1$$

4: E_{ss}: steady-state error

Direct Design Method Of Ragazzini [5]: An Example

- Consider the plant: $s^2 + s + 1 = 0$

With T_s=1 → z-Transform: $z^2 + 0.786z + 0.368$=0

- Let’s design this **system** such that
  - $K_v = 1$
  - Poles at the roots of the plant equation & additional poles as needed

➔$\mathrm{H(z)} = \frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+\cdots}{1-0.786z^{-1}+0.368\,z^{-2}}$

I. Causality: $\mathrm{H(z)}|_{z=\infty} = 0 \rightarrow b_0 = 0$

II. Stability: All poles/zeros of G(z) are in the unit circle
  - except for b_0, which is taken care of by $b_0 = [Const] = 0$

III. Tracking:
  - $\mathrm{H(1)} = b_1 + b_2 + b_3 + \cdots = 1 \cdot (1 - 0.786 + 0.368)$ &
  - $-\{1\}\left.\frac{dH(z)}{d(z^{-1})}\right|_{z=1} = \frac{1}{\{1\}}$ ➔ $\frac{b_1+2b_2+3b_3+\cdots\ -[-.05014]}{(1-0.786+0.368)}$ (note the z^{-1})
  - Truncate the number of unknowns to 2 “zeros” … thus solve for b_1 and b_2 (& set b_3,b_4,…=0)

$\therefore\ \mathrm{H(z)} = \frac{b_1z+b_2}{z^2-0.786z+0.368}$ ➔

$$D(z) = \frac{(z-1)(z-0.9048)(0.6321)}{(0.04837)(z+0.9672)}\,\frac{(z-0.07932)}{(z-1)(z-0.4180)}$$

$$= 13.07\,\frac{(z-0.9048)}{(z+0.9672)}\,\frac{(z-0.07932)}{(z-0.4180)}.$$

Next Time…

- **Example of Digital Feedback Control**

- Review:
  - Chapter 5 of FPW

- More Pondering??