Model Predictive Control: Background

B. Wayne Bequette

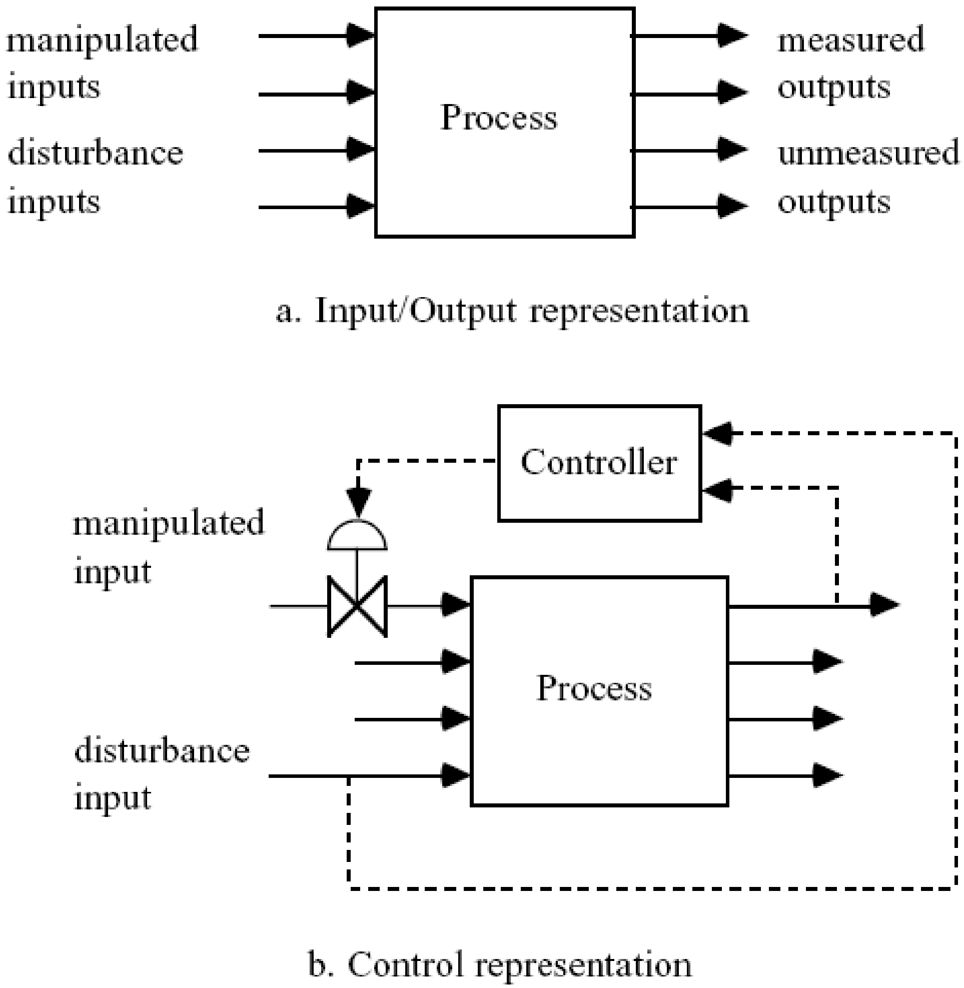
- Concise Review of Undergraduate Process Control
- Introduction to Discrete-time Systems
- Digital PID
- Discrete Internal Model Control (IMC)

NTNU – Trondheim
Norwegian University of
Science and Technology

January 2015

Rensselaer
Chemical and Biological Engineering

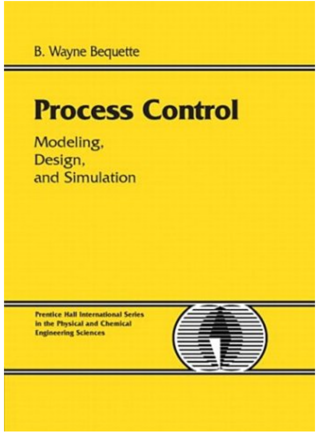
Automation and Control



a. Input/Output representation

b. Control representation

Figure 1–1 Conceptual process input/output block diagram.

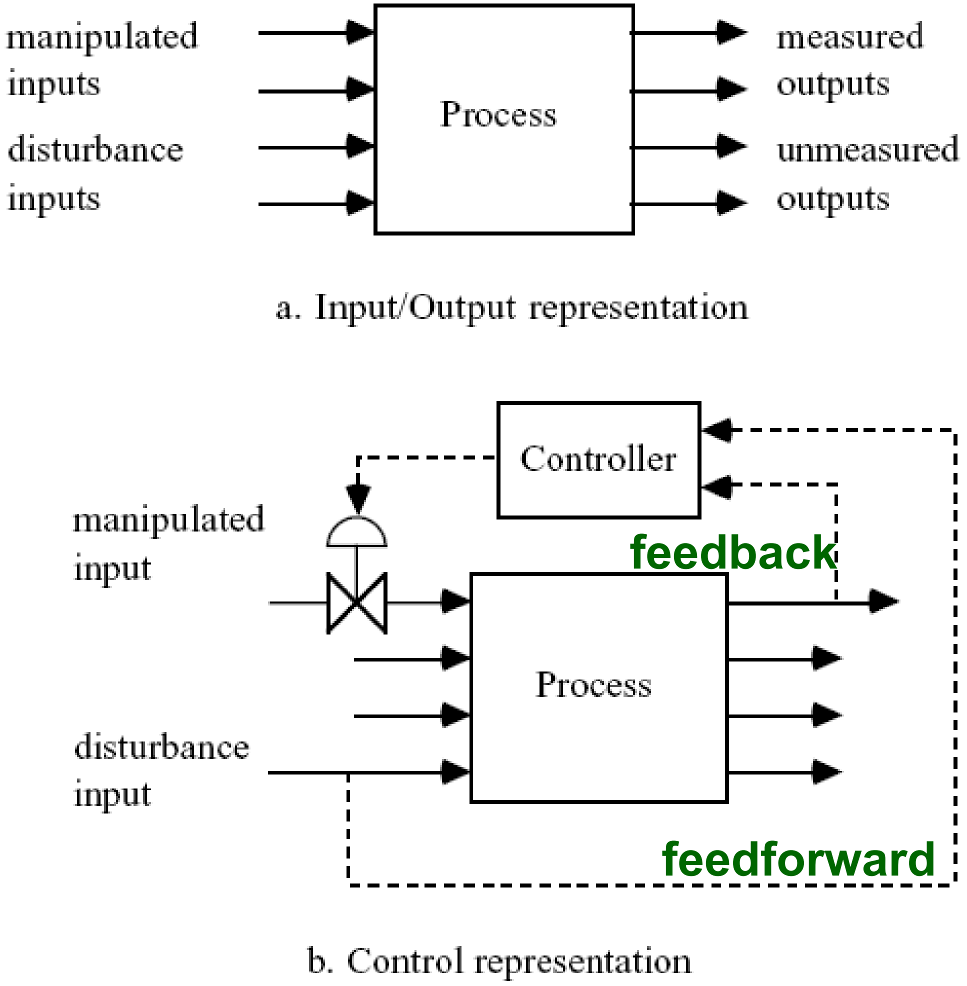


B. Wayne Bequette

Process Control

Modeling,
Design,
and Simulation

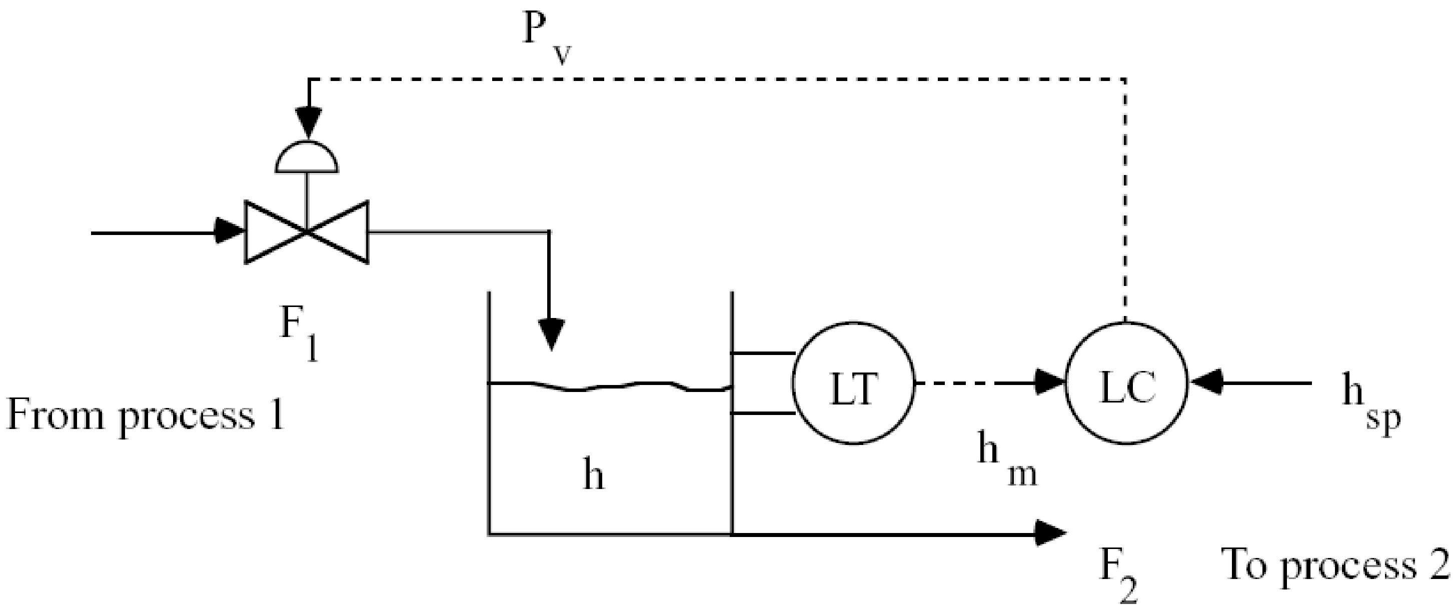
Prentice Hall International Series
in the Physical and Chemical
Engineering Sciences

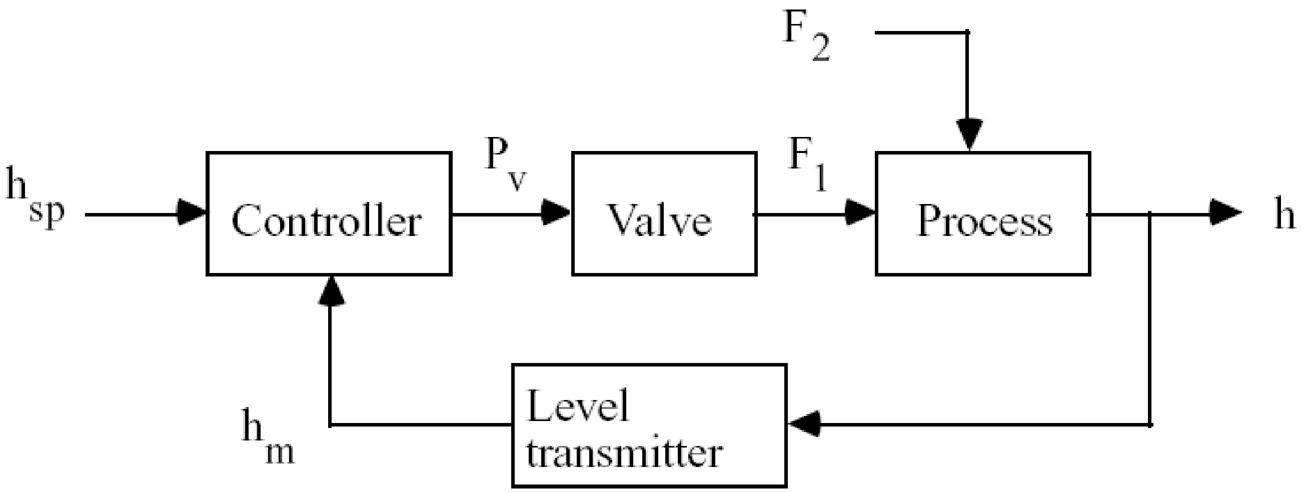


a. Input/Output representation

b. Control representation

Figure 1–1 Conceptual process input/output block diagram.


P_v
F_1
From process 1
h
LT
LC
h_m
h_{sp}
F_2
To process 2


F_2
h_{sp}
Controller
P_v
Valve
F_1
Process
h
h_m
Level transmitter

Control Algorithm

- Compares measured process output with desired setpoint, and calculates a manipulated input (often a flowrate)

- Proportional-integral-derivative (PID)
  - Long history, “workhorse”, lower-level control loops

- Model predictive control (MPC)
  - Most widely applied “advanced” control algorithm
  - Constraints, multivariable systems

Common notation

r = setpoint (desired value of process output)

y = measured process output

e = error (setpoint – output, $r - y$)

u = manipulated input (often a flowrate)

Proportional-Integral-Derivative (PID) Control

Error = setpoint − measured output

$$e(t) = r(t) - y(t)$$

$$u(t) = u_0 + k_c \left[e(t) + \frac{1}{\tau_I} \int_0^t e(t)\,dt + \tau_D \frac{de(t)}{dt} \right]$$

Manipulated Input

Proportional gain

Integral time

Derivative time

- Single-input, single-output design
  - Problems with one controller can impact another controller
- Constraints
  - Can cause “windup” problems
- Does not explicitly require a process model

Continuous Linear Models

State Space and Transfer Function

B. Wayne Bequette

- Linearization
- State Space Form
- Transfer Function
- Step Responses
- MV Properties

Rensselaer
Chemical and Biological Engineering

Nonlinear ODE Models

- General "Lumped Parameter" Form

$$\dot{x} = f(x, u, p)$$ differential state equations

$$y = g(x, u, p)$$ algebraic output equations

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \dot{x} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \vdots \\ \frac{dx_n}{dt} \end{bmatrix} \qquad u = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_r \end{bmatrix} \qquad p = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_q \end{bmatrix}$$

states

state derivatives (from accumulation term)

inputs

outputs

parameters

Linearization

$$\left.\begin{aligned} \dot{x} &= f(x,u,p) \\ y &= g(x,u,p) \end{aligned}\right\} \quad \text{steady-state solution, } u_s,\ x_s,\ y_s$$

‘perturbation’ or ‘deviation’ variables

$$\begin{aligned} \dot{x}' &= Ax' + Bu' \\ y' &= Cx' + Du' \end{aligned} \qquad x' = \begin{bmatrix} x_1 - x_{1s} \\ x_2 - x_{2s} \\ \vdots \\ x_n - x_{ns} \end{bmatrix} \qquad u' = \begin{bmatrix} u_1 - u_{1s} \\ u_2 - u_{2s} \\ \vdots \\ u_m - u_{ms} \end{bmatrix} \qquad y' = \begin{bmatrix} y_1 - y_{1s} \\ y_2 - y_{2s} \\ \vdots \\ y_r - y_{rs} \end{bmatrix}$$

Where:

$$A_{ij} = \left.\frac{\partial f_i}{\partial x_j}\right|_{x_s,u_s} \qquad B_{ij} = \left.\frac{\partial f_i}{\partial u_j}\right|_{x_s,u_s} \qquad C_{ij} = \left.\frac{\partial g_i}{\partial x_j}\right|_{x_s,u_s} \qquad D_{ij} = \left.\frac{\partial g_i}{\partial u_j}\right|_{x_s,u_s}$$

eqn i state j eqn i input j

Example: Van de vuuse Reaction

$$\frac{dC_A}{dt} = \frac{F}{V}\left(C_{Af} - C_A\right) - k_1 C_A - k_3 C_A^2 \qquad = \mathrm{f}_1$$

$$\frac{dC_B}{dt} = -\frac{F}{V}C_B + k_1 C_A - k_2 C_B \qquad = \mathrm{f}_2$$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} C_A - C_{As} \\ C_B - C_{Bs} \end{bmatrix}, \quad u = \begin{bmatrix} F/V - F_s/V \\ C_{Af} - C_{Afs} \end{bmatrix}, \quad y = x_2 = \left[C_B - C_{Bs}\right]$$

$$A_{11} = \left.\frac{\partial f_1}{\partial x_1}\right|_{x_s, u_s} = \left.\frac{\partial f_1}{\partial C_A}\right|_{C_{As}, etc}$$

$$A = \begin{bmatrix} -\frac{F_s}{V} - k_1 - 2k_3 C_{As} & 0 \\ k_1 & -\frac{F_s}{V} - k_2 \end{bmatrix}, \quad B = \begin{bmatrix} C_{Afs} - C_{As} & \frac{F_s}{V} \\ -C_{Bs} & 0 \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 1 \end{bmatrix}, \quad D = \begin{bmatrix} 0 & 0 \end{bmatrix}$$

“hidden slide” provided for additional background

Continuous State Space Model

$$\left.\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}\right\} \text{Assumes deviation variable form}$$

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \cdots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}}_{\text{n by n}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} b_{11} & \cdots & b_{1m} \\ \vdots & \cdots & \vdots \\ b_{n1} & \cdots & b_{nm} \end{bmatrix}}_{\text{n by m}} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_r \end{bmatrix} = \underbrace{\begin{bmatrix} c_{11} & \cdots & c_{1n} \\ \vdots & \cdots & \vdots \\ c_{r1} & \cdots & c_{rn} \end{bmatrix}}_{\text{r by n}} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \underbrace{\begin{bmatrix} d_{11} & \cdots & d_{1m} \\ \vdots & \cdots & \vdots \\ d_{r1} & \cdots & d_{rm} \end{bmatrix}}_{\text{r by m}} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix}$$

Stability

- Eigenvalues of A
  - n x n A matrix = n eigenvalues = n states
  - If all eigenvalues have a negative real portion – stable
  - If any eigenvalue has a positive real portion – unstable
  - Complex – generally ‘oscillatory’

- Characteristic Polynomial (n roots):
  - Must have at least 2 states to oscillate (be complex)

Identity matrix

$$\det(\lambda I - A) = 0$$

eigenvalues are the roots of the equation

Example: CSTR at 2 Operating Points

Operating condition 1

$$A_1 = \begin{bmatrix} -1.1680 & -0.0886 \\ 2.0030 & -0.2443 \end{bmatrix}$$

Operating condition 2

$$A_2 = \begin{bmatrix} -1.8124 & -0.2324 \\ 9.6837 & 1.4697 \end{bmatrix}$$

$$\lambda I - A_1 = \begin{bmatrix} \lambda + 1.1680 & 0.0886 \\ -2.0030 & \lambda + 0.2443 \end{bmatrix}$$

$$\det(\lambda I - A_1) = (\lambda + 1.1680)(\lambda + 0.2443) - (0.0886)(-2.003) = 0$$

$$= \lambda^2 + 1.4123\lambda + 0.4628 = 0$$

$\lambda = -0.8955 \text{ hr}^{-1}$ and $\lambda = -0.5168 \text{ hr}^{-1}$ → Operating Point 1 = Stable

Can show that the Eigenvalues for Operating Point 2 are:

$\lambda = -0.8366 \text{ hr}^{-1}$ and $\boxed{\lambda = 0.4939 \text{ hr}^{-1}}$ → Operating Point 2 = Unstable

“hidden slide” provided for additional background

Laplace Transform

- Convert Differential Equations to Algebraic Equations
  - Design controllers using algebra rather than differential equations
- Easy Analysis of “Block Diagrams”

$$L\left[f(t)\right] = F(s) = \int_0^\infty f(t)e^{-st}dt$$ Definition of Laplace Transform

$$L[1] = \frac{1}{s}$$ Unit step

$$L\left[\frac{df(t)}{dt}\right] = sF(s) - f(0)$$ Derivative

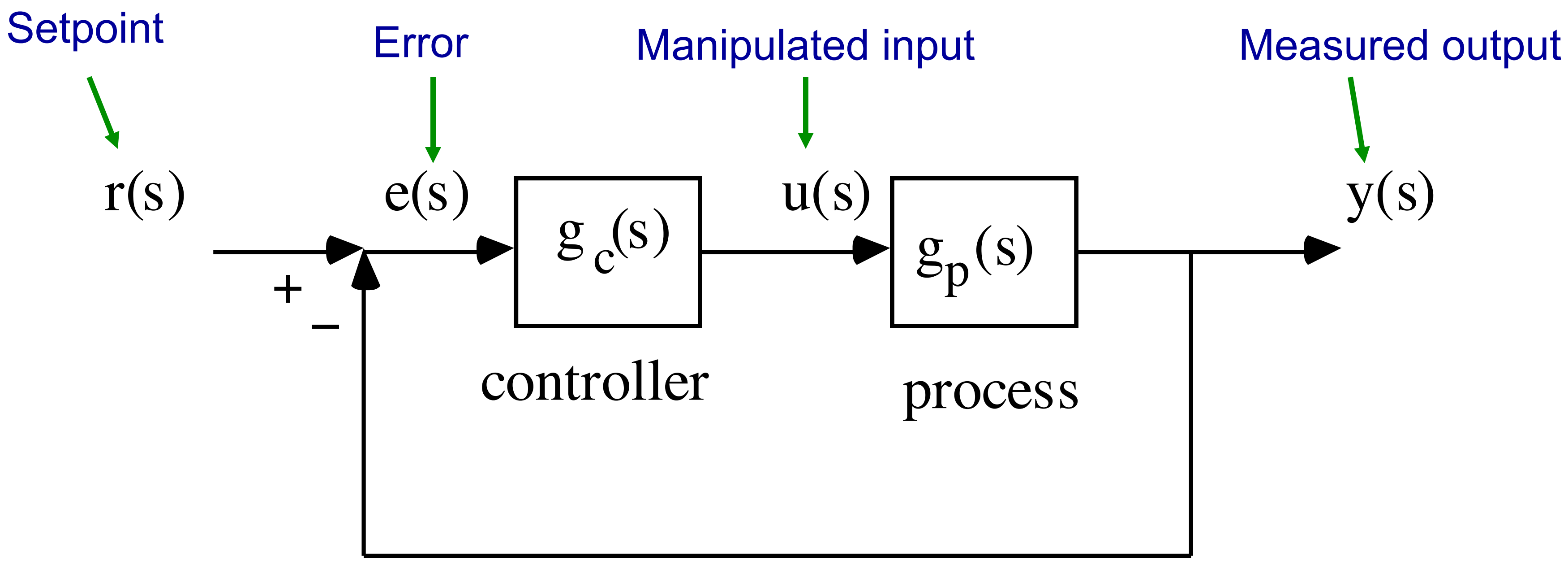
Laplace Transforms, cont'd

- Most Undergraduate Courses
  - Much (perhaps **too much**) time is spent:
    - Taking Laplace transform of process differential equation
    - Laplace transform of "forcing function" (typically a step)
    - Multiply, then perform partial fraction expansion
    - Invert each term back to the time domain for an analytical expression

Very painful, many nightmares?

- **In Practice**
  - **Step response behavior for process understanding**
  - **Main use of transfer function is for "controller synthesis"**
  - **Can easily convert from differential equations ("state space") to transfer function form using MATLAB, etc.**
  - **Closed-loop block diagram analysis**

Block Diagram Analysis



$$y(s) = \frac{g_c(s)g_p(s)}{1 + g_c(s)g_p(s)} \cdot r(s)$$

Closed-loop characteristic equation (roots determine stability)

Routh Array, etc.

State Space to Transfer Function Form

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

$$y = Cx + Du$$

Take the Laplace Transform (s = transform variable)

$$sx(s) = Ax(s) + Bu(s)$$

$$y(s) = Cx(s) + Du(s)$$

usually, D = 0

$$x(s) = (sI - A)^{-1} Bu(s)$$

$$y(s) = \underbrace{\left[C(sI - A)^{-1} B + D\right]} u(s)$$

$$G(s) = \left[C(sI - A)^{-1} B\right]$$

$$y(s) = G(s) \quad u(s)$$

Transfer function matrix

Matrix Transfer Function Form

$$y(s) = G(s)u(s)$$

$$\begin{bmatrix} y_1(s) \\ \vdots \\ y_r(s) \end{bmatrix} = \underbrace{\begin{bmatrix} g_{11}(s) & \cdots & g_{1m}(s) \\ \vdots & & \\ g_{r1}(s) & \cdots & g_{rm}(s) \end{bmatrix}}_{\text{r rows by m columns}} \begin{bmatrix} \left.\begin{matrix} u_1(s) \\ \vdots \\ u_m(s) \end{matrix}\right\} \end{bmatrix}$$

r outputs

m inputs

$$y_i(s) = g_{ij}(s)u_j(s)$$

output i

input j

Transfer Function Matrix

First subscript = output

Second subscript = input

Dynamic Behavior: SISO

Relative order

= n-m

$$g_p(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \cdots + b_1 s + b_0}{a_n s^n + a_{m-1} s^{n-1} + \cdots + a_1 s + a_0}$$

Polynomial

$$g_p(s) = \frac{k_{pz}(s - z_1)(s - z_2)\cdots(s - z_m)}{(s - p_1)(s - p_2)\cdots(s - p_n)}$$

Pole-zero

$$g_p(s) = \frac{k_p(\tau_{n1} s + 1)(\tau_{n2} s + 1)\cdots(\tau_{nm} s + 1)}{(\tau_{p1} s + 1)(\tau_{p2} s + 1)\cdots(\tau_{pn} s + 1)}$$

Gain-time constant

Zeros = roots of numerator

Poles = roots of denominator = determine stability

$\tau_{p1} = -1/p_1$

Large time constant = small, negative, pole

Example of Pole-Zero Cancellation

- Number of states = Number of poles (order of numerator of transfer function), except when some poles and zeros 'cancel'

$$\left.\begin{aligned} A &= \begin{bmatrix} 0 & 0.9056 \\ -0.75 & -2.5640 \end{bmatrix} \qquad B = \begin{bmatrix} -1.5301 \\ 3.8255 \end{bmatrix} \\ C &= \begin{bmatrix} 1 & 0 \end{bmatrix} \qquad\qquad\qquad\quad D = \begin{bmatrix} 0 \end{bmatrix} \end{aligned}\right\} \text{ Bioreactor model}$$

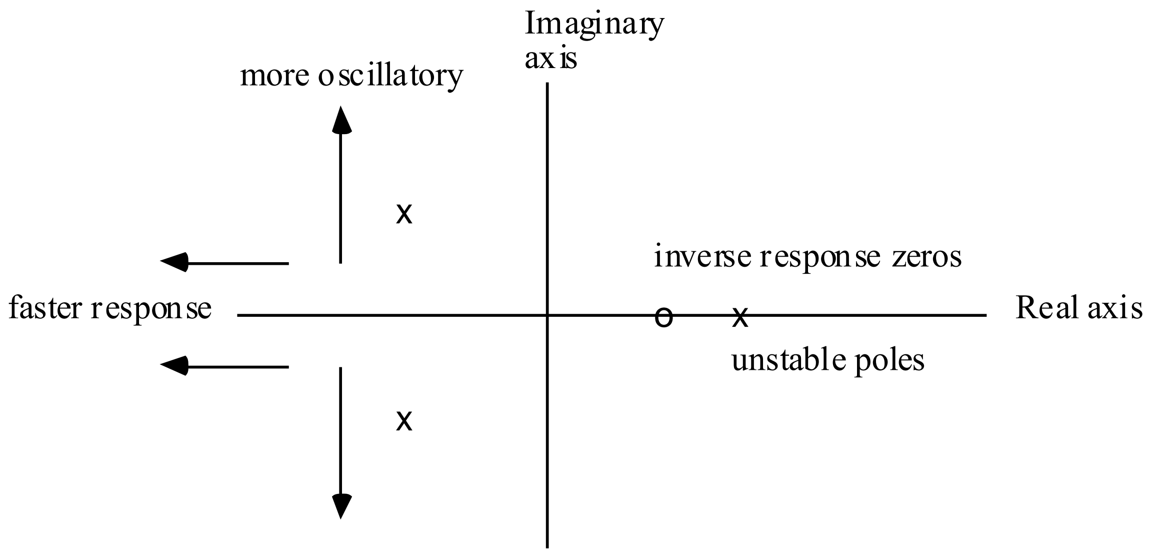
$$g_p(s) = \frac{-1.5302s - 0.4590}{s^2 + 2.564s + 0.6792} = \frac{-1.5302(s + 0.3)}{(s + 0.3)(s + 2.2640)}$$

$$g_p(s) = \frac{-0.6758}{0.4417s + 1}$$

"hidden slide" provided for additional background

Chemical Process Engineers

- More familiar with gain-time constant form
- Most chemical processes are stable
  - Exceptions: Exothermic or bioreactors, closed-loop systems (mistuned)



x = pole

o = zero

“hidden slide” provided for additional background

First-Order

$$\tau_p \frac{dy}{dt} + y = k_p u$$

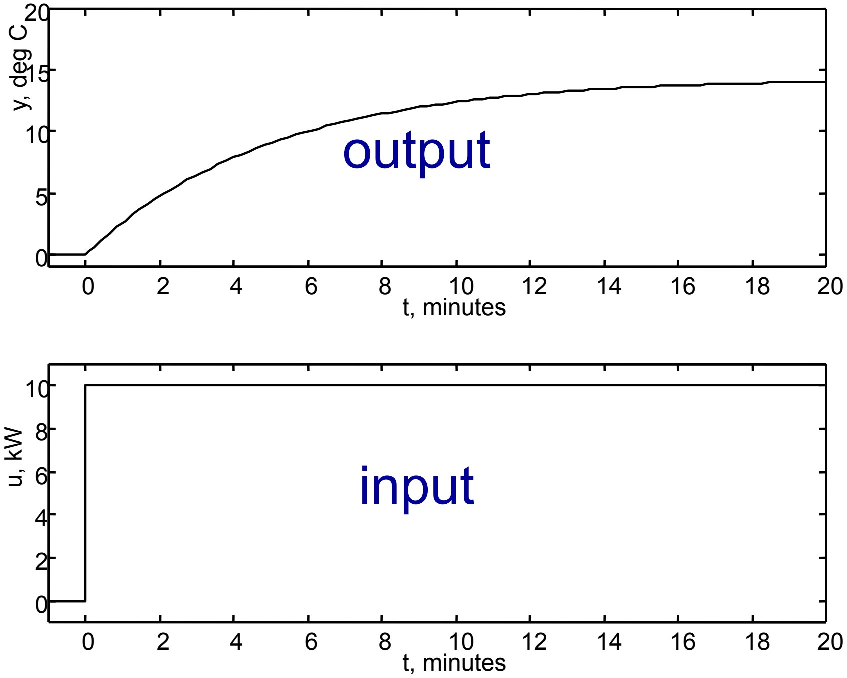
or

$$\frac{dy}{dt} = -\frac{1}{\tau_p} y + \frac{k_p}{\tau_p} u$$

gain

$$y(s) = \frac{k_p}{\tau_p s + 1} u(s)$$

time constant



Example: gain = 1.43 °C/kW

time constant = 5 minutes

First Order + Time-delay

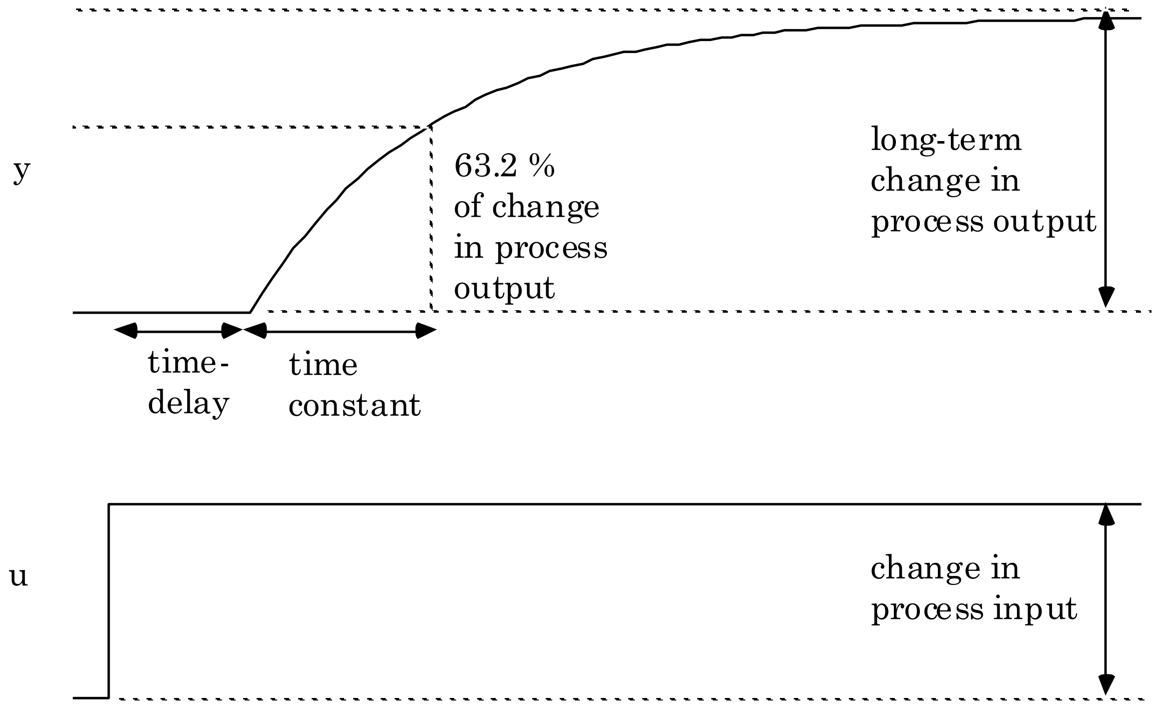
$$\tau_p \frac{dy}{dt} + y = k_p u(t-\theta)$$

gain

time-delay

$$y(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1} u(s)$$

time constant



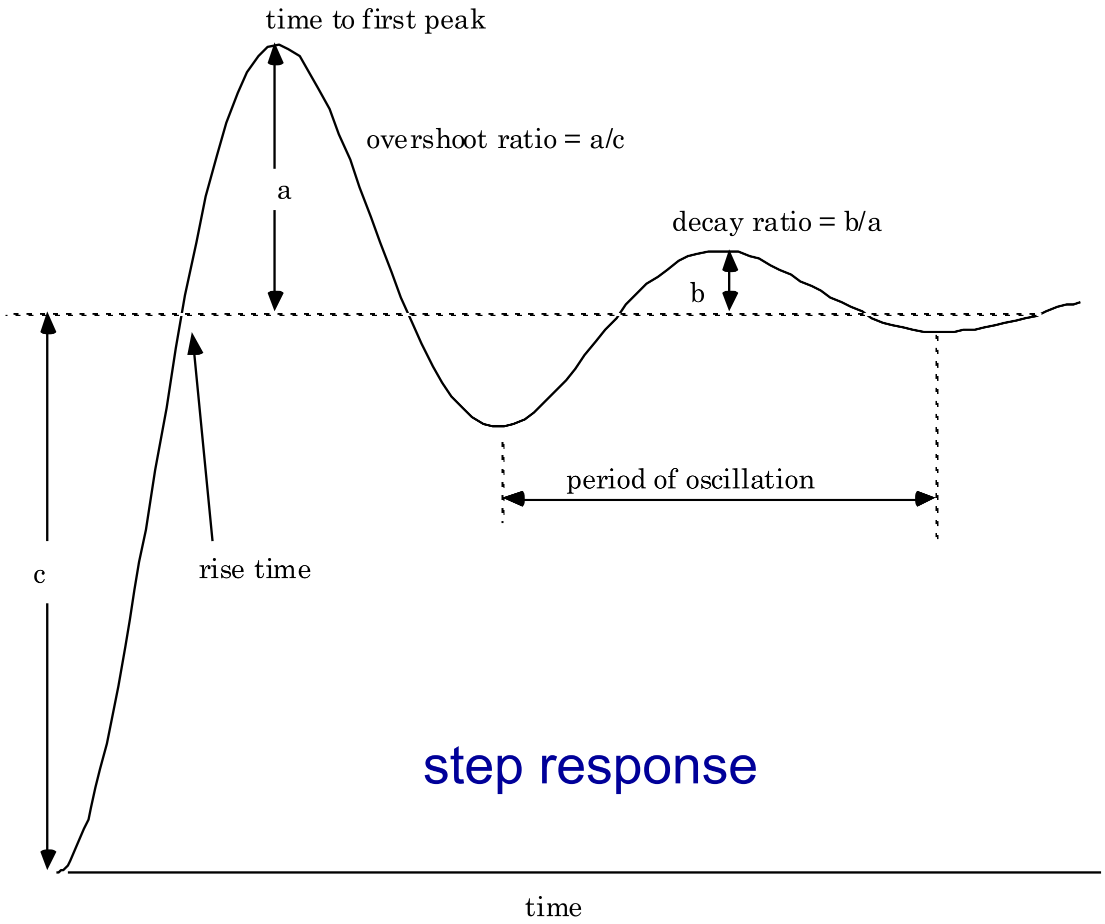
Second-order Underdamped

$$y(s) = \frac{k}{\tau^2 s^2 + 2\zeta\tau s + 1} \cdot u(s)$$

$$\zeta < 1$$

$$p_{1,2} = -\frac{\zeta}{\tau} \pm \frac{\sqrt{\zeta^2 - 1}}{\tau}$$

$$p_{1,2} = \mathrm{Re} \pm j\,\mathrm{Im}$$



step response

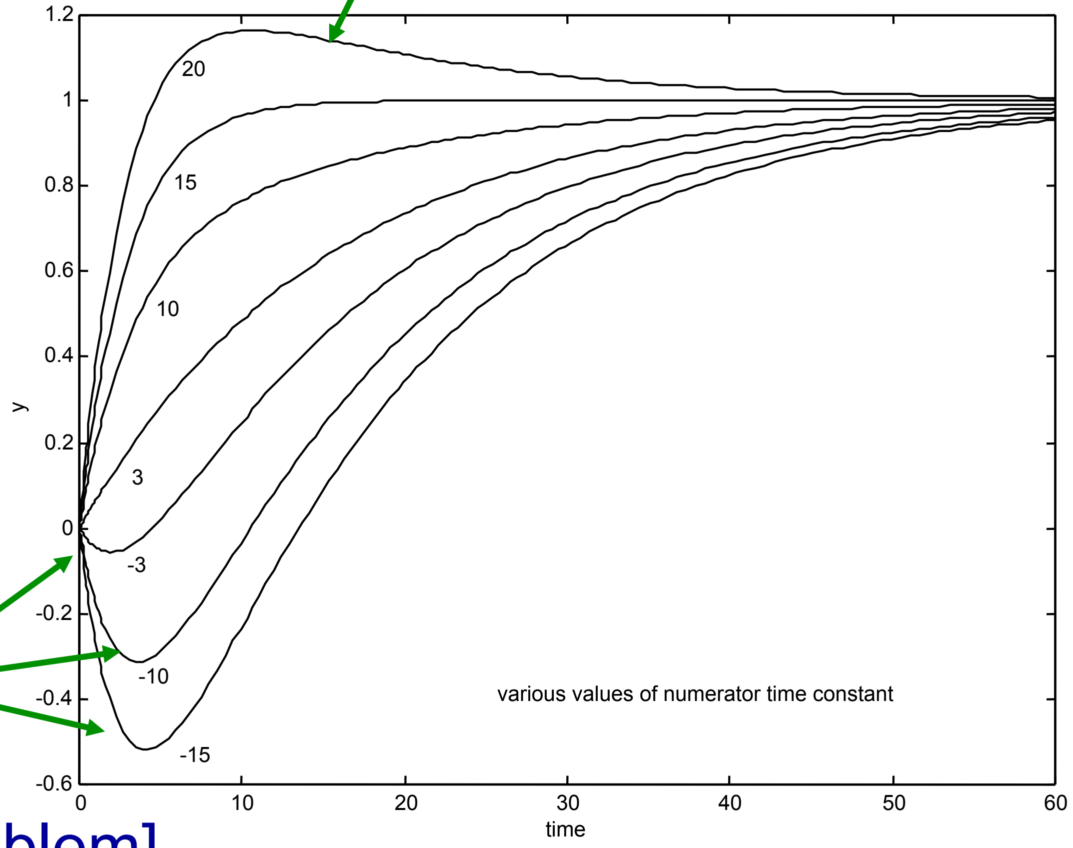
Numerator Dynamics

$$g_p(s) = \frac{(\tau_n s + 1)}{(3s+1)(15s+1)}$$

$$z = -1/\tau_n$$

$$\tau_n < 0 \rightarrow z > 0$$

τ_n = large → faster response



"inverse response"

(right-half-plane zeros)

[challenging control problem]

Steady-state Gain

- For stable systems, the steady-state gain is found from the long-term response

$$y(s)=\frac{k_p(\tau_{n1}s+1)(\tau_{n2}s+1)\cdots(\tau_{nm}s+1)}{(\tau_{p1}s+1)(\tau_{p2}s+1)\cdots(\tau_{pn}s+1)}\cdot\frac{\Delta u}{s} \qquad \longleftarrow \text{Step input}$$

Final value theorem

$$\lim_{t\to\infty}y(t)=\lim_{s\to 0}sy(s)=s\cdot\frac{k_p(\tau_{n1}s+1)(\tau_{n2}s+1)\cdots(\tau_{nm}s+1)}{(\tau_{p1}s+1)(\tau_{p2}s+1)\cdots(\tau_{pn}s+1)}\cdot\frac{\Delta u}{s}$$

$$\lim_{t\to\infty}y(t)=\lim_{s\to 0}sy(s)=s\cdot k_p\cdot\frac{\Delta u}{s}=k_p\Delta u$$

$$k_p=\frac{\lim_{t\to\infty}y(t)}{\Delta u}=\frac{\Delta y}{\Delta u}$$

“hidden slide” provided for additional background

Process Gain: Controller Implications

Long-term behavior from steady-state information

$$\Delta y = k_p \Delta u$$

Controller really serves as “inverse” of process

$$\Delta u = \frac{1}{k_p} \Delta y$$

Process with higher gain is generally easier to control, all else being equal…

Process Zero: Controller Implications

For “tight” control, controller is “inverse” of process

$$y(s) = g_p(s)u(s)$$

$$u(s) = \frac{1}{g_p(s)} y(s)$$

Inverse of $g_p(s)$ is unstable if $g_p(s)$ has a right-half-plane zero

Transfer Function to State Space

- An infinite number of state space models can yield a given transfer function model

- Two different state space “realizations” are normally used
    - Controllable canonical form
    - Observable canonical form

Multivariable Systems: Properties

2 input – 2 output Example

$$y_1(s) = g_{11}(s)u_1(s) + g_{12}(s)u_2(s)$$
$$y_2(s) = g_{21}(s)u_1(s) + g_{22}(s)u_2(s)$$

Matrix-vector Notation

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} g_{11}(s) & g_{12}(s) \\ g_{21}(s) & g_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix}$$

$$y(s) = G_p u(s)$$

Similar to SISO, controller serves as “process inverse”

$$u(s) = G_p^{-1} y(s)$$

Steady-state Implications

System 1

$$y_1 = 1u_1 + 1u_2$$
$$y_2 = -1u_1 + 1u_2$$

System 2

$$y_1 = 1u_1 + 1u_2$$
$$y_2 = 1u_1 + 1u_2$$

$$u = K_p^{-1} y$$

$$K_{p1} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$K_{p2} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Which system will be more difficult to control?

Steady-state Implications

System 1

$$y_1 = 1u_1 + 1u_2$$
$$y_2 = -1u_1 + 1u_2$$

System 2

$$y_1 = 1u_1 + 1u_2$$
$$y_2 = 1u_1 + 1u_2$$

$$u = K_p^{-1} y$$

$$K_{p1} = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

$$K_{p2} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Inverse does not exist

(Gain Matrix is singular, rank = 1)

Cannot independently control both outputs

Another Example

$$y_1 = 1u_1 + 0.95u_2$$
$$y_2 = 1u_1 + 1u_2$$

$$K_{p3} = \begin{bmatrix} 1 & 0.95 \\ 1 & 1 \end{bmatrix}$$

Inverse exists – No longer singular

(“gut feeling” that there may still be a problem…)

Directional Sensitivity: Singular Value Decomposition (SVD)

$$G = U\Sigma V^T$$

Gain matrix

Left singular Vector matrix

Matrix of singular values

Right singular Vector matrix

Example

Max singular value

$$\underbrace{\begin{bmatrix}1 & 0.95\\ 1 & 1\end{bmatrix}}_{G} = \underbrace{\begin{bmatrix}-0.70 & -0.72\\ -0.72 & 0.70\end{bmatrix}}_{U}\underbrace{\begin{bmatrix}1.98 & 0\\ 0 & 0.0253\end{bmatrix}}_{\Sigma}\underbrace{\begin{bmatrix}-0.72 & -0.70\\ -0.70 & 0.72\end{bmatrix}^T}_{V}$$

Strongest output direction

Strongest input direction

Example, cont'd

$$\underbrace{\begin{bmatrix} 1 & 0.95 \\ 1 & 1 \end{bmatrix}}_{G} = \underbrace{\begin{bmatrix} -0.70 & -0.72 \\ -0.72 & 0.70 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 1.98 & 0 \\ 0 & 0.0253 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} -0.72 & -0.70 \\ -0.70 & 0.72 \end{bmatrix}}_{V}{}^{T}$$

Input in Strongest Direction

Output in Strongest Direction

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 0.72 \\ 0.70 \end{bmatrix} \Longrightarrow \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.95 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0.72 \\ 0.70 \end{bmatrix} = \begin{bmatrix} 1.38 \\ 1.41 \end{bmatrix}$$

$$\mathbf{y} = \mathbf{K_p} \quad \mathbf{u}$$

Input in Weakest Direction

Output in Weakest Direction

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} -0.70 \\ 0.72 \end{bmatrix} \Longrightarrow \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0.95 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -0.70 \\ 0.72 \end{bmatrix} = \begin{bmatrix} -0.018 \\ 0.018 \end{bmatrix}$$

Same magnitude input = very different magnitude output

Example, cont'd

$$\underbrace{\begin{bmatrix} 1 & 0.95 \\ 1 & 1 \end{bmatrix}}_{G} = \underbrace{\begin{bmatrix} -0.70 & -0.72 \\ -0.72 & 0.70 \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} 1.98 & 0 \\ 0 & 0.0253 \end{bmatrix}}_{\Sigma} \underbrace{\begin{bmatrix} -0.72 & -0.70 \\ -0.70 & 0.72 \end{bmatrix}^{T}}_{V}$$

High condition number indicates problems

Condition number = σ_1/σ_2 = 1.98/0.0253 = 78

Note: For SVD analysis it is important to properly scale the inputs and outputs

MV Dynamic Properties: Quadruple Tank

Operating Point 1 – Minimum Phase

$$G_1(s) = \begin{bmatrix} \dfrac{2.6}{62s+1} & \dfrac{1.5}{(23s+1)(62s+1)} \\ \dfrac{1.4}{(30s+1)(90s+1)} & \dfrac{2.8}{(90s+1)} \end{bmatrix}$$

"Transmission zeros" are both negative

$$z = -0.060 \text{ and } -0.018 \text{ sec}^{-1}$$

Operating Point 2 – Nonminimum Phase (RHPT zero)

$$G_2(s) = \begin{bmatrix} \dfrac{1.5}{63s+1} & \dfrac{2.5}{(39s+1)(63s+1)} \\ \dfrac{2.5}{(56s+1)(91s+1)} & \dfrac{1.6}{(91s+1)} \end{bmatrix}$$

$$z = -0.057 \text{ and } +0.013 \text{ sec}^{-1}$$

Matrix inverse is unstable

Multivariable Systems

- Can have right-half-plane "transmission zeros" even when no individual transfer function has a RHP zero

- Can have individual RHP zeros yet not have a RHPT zero
  - Fine performance when constraints are not active
  - May fail when one constraint becomes active or a loop is "opened"

- Can exhibit "directional sensitivity" – with some setpoint directions much easier to achieve than others

- Some of these MV properties cause challenges *independent of control strategy selected*

Summary

- Nonlinear Model
  - Solve for steady-state, then linearize (Taylor series expansion)
- State Space (linear) Model
  - Deviation variable form (perturbations from steady-state)
- Dynamics
  - Eigenvalues of A matrix = poles of transfer function matrix
    - Pole-zero cancellation may reduce number of poles
  - Right-half-plane zeros = inverse response
  - MV: Transmission Zeros
- Long term behavior from steady-state gain
- Singular Value Decomposition (SVD)

Suggestion

- Work through **Workshop on the Three Tank Model**

- Derive state space model
- MATLAB Commands
  - Eigenvalues (eig)
  - State Space model (ss), step input (step)
- Integration using ode45
  - Single integration
  - In a “loop,” integrating from time step to time step
- Convert state space to transfer function (ss2tf)

Discrete Linear Models

B. Wayne Bequette

- Sampling Rules/Assumptions
- Continuous to Discrete
- Z-transform
- Dynamic Properties of Discrete Systems

Rensselaer

Chemical and Biological Engineering

Discrete Models

Input held constant between sample times (zero-order hold)

Sample time is constant

Rule of thumb for discrete control – select sample time roughly 1/10 of dominant time constant

Discrete Models

$$x_{k+1} = \Phi x_k + \Gamma u_k$$

$$y_k = C x_k + D u_k$$

State Space

Some texts/papers have different sign conventions

$$y_k = -a_1 y_{k-1} - a_2 y_{k-2} - \cdots - a_n y_{k-n} +$$

$$b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots + b_m u_{k-m}$$

Input-Output (ARX)

usually *b₀* = 0

$$y(z) = Z[y_k]$$

Z-transform

$$Z[y_{k-1}] = z^{-1} y(z)$$

Backwards shift operator

$$\Downarrow$$

$$\left(1 + a_1 z^{-1} + \ldots + a_{n-1} z^{-n+1} + a_n z^{-n}\right) y(z) =$$

$$\left(b_0 + b_1 z^{-1} + \ldots + b_{m-1} z^{-m+1} + b_m z^{-m}\right) u(z)$$

Solve for y(z)

Discrete Models

$$y(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{m-1} z^{-m+1} + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_{n-1} z^{-n+1} + a_n z^{-n}} u(z)$$

$$g_p(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_{m-1} z^{-m+1} + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_{n-1} z^{-n+1} + a_n z^{-n}}$$

Discrete

Transfer Function

Multiply by $\frac{z^n}{z^n}$

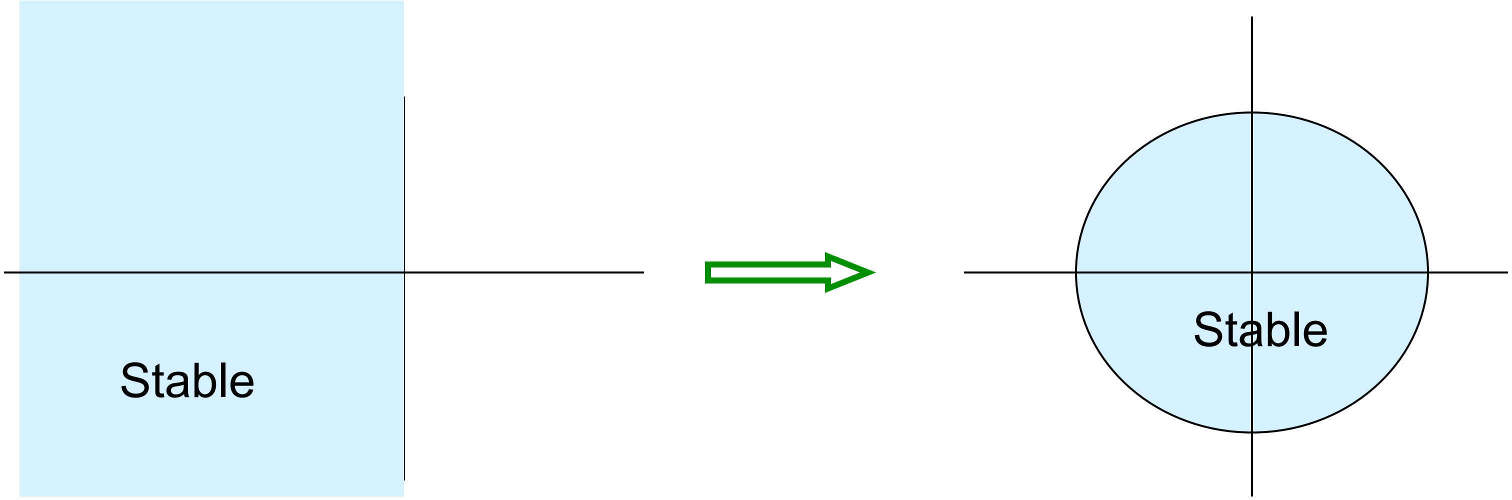
$$g_p(z) = \frac{b_0 z^n + b_1 z^{n-1} + \dots + b_m z^{n-m}}{z^n + a_1 z^{-1} + \dots + a_{n-1} z^{-n+1} + a_n}$$

$$g_p(z) = K_{pz} \cdot \frac{(z - z_1)(z - z_2)\cdots(z - z_m)}{(z - p_1)(z - p_2)\cdots(z - p_n)}$$

zeros = roots of numerator polynomial

poles = roots of denominator polynomial

Stability



Continuous

Discrete

poles in LHP are stable

poles inside unit circle are stable

Example

$$y(k) = -a_1 y(k-1) + b_1 u(k-1) \Longrightarrow g_p(z) = \frac{b_1 z^{-1}}{1 + a_1 z^{-1}} = \frac{b_1}{z + a_1}$$

Value of the pole $\longrightarrow$ $p = -a_1$

$$a_1 = 0.5, -0.5, 1.5, -1.5 \Longrightarrow p = -0.5, 0.5, -1.5, 1.5$$

Let y(0) = 1, u(k) = 0

a_1	0.5	-0.5	1.5	-1.5
p	-0.5	0.5	-1.5	1.5
$y(1)$	-0.5	0.5	-1.5	1.5
$y(2)$	0.25	0.25	2.25	2.25
$y(3)$	-0.125	0.125	-3.375	3.375
$y(4)$	0.0625	0.0625	5.0625	5.0625
Characteristic behavior	Oscillatory, stable	Monotonic, stable	Oscillatory, unstable	Monotonic, unstable

Simple Example: Results

- Poles inside circle
  - stable
- Poles outside circle
  - unstable
- Negative poles
  - oscillate
- First-order discrete systems can oscillate
  - **This cannot happen with continuous systems**

Discrete zeros

- Zeros outside unit circle
  - Inverse is unstable (not necessarily inverse response)
- **Any continuous system with relative order >2 will have an unstable inverse with a small enough sample time**

Astrom, KJ, P Hagander & J Sternby "Zeros of Sampled Systems," *Proceedings of the 1984 American Control Conference*, 1077-1081

Relative order = 3

Relative order = 1

$$g_p(s) = \frac{1}{(s+1)^3} \xrightarrow{\text{Sample time = 1}} g_p(z) = \frac{0.0803\boxed{(z+1.7990)}(z+0.1238)}{(z-0.3679)^3}$$

poles = -1 (multiplicity 3)

poles = 0.3679 (multiplicity 3)

zeros = $\boxed{-1.7990}$ & -0.1238

Unstable inverse

Final & Initial Value Theorems

Final value theorem $\lim_{n\to\infty} y(n\Delta t) = \lim_{z\to 1}\left(1-z^{-1}\right)y(z)$

Long-term step response $y(z) = g_p(z)u(z) = g_p(z)\cdot\overset{\text{unit step}}{\frac{1}{1-z^{-1}}}$

$$\lim_{t\to\infty} y(t) = \lim_{z\to 1}\left(1-z^{-1}\right)y(z) = \lim_{z\to 1}\left(1-z^{-1}\right)g_p(z)\frac{1}{1-z^{-1}} = \lim_{z\to 1} g_p(z)$$

So, simply set z = 1 in $g_p(z)$ for long-term unit step response

Initial value theorem $\lim_{t\to 0} y(t) = \lim_{z\to\infty}\left(1-z^{-1}\right)y(z)$

“hidden slide” provided for additional background

Final Value Theorems

Long-term outputs for unit step inputs

continuous

$$g_p(s) = \frac{1}{(s+1)^3}$$

$$g_p(s \to 0) = \frac{1}{(0+1)^3} = 1$$

discrete

$$g_p(z) = \frac{0.0803(z+1.7990)(z+0.1238)}{(z-0.3679)^3}$$

$$g_p(z \to 1) = \frac{0.0803(1+1.7990)(1+0.1238)}{(1-0.3679)^3} = \frac{0.2525}{0.2525} = 1$$

“hidden slide” provided for additional background

State Space Models

$$\begin{aligned}\dot{x} &= Ax + Bu \\ y &= Cx + Du\end{aligned} \quad \xrightarrow{?} \quad \begin{aligned}x_{k+1} &= \Phi x_k + \Gamma u_k \\ y_k &= Cx_k + Du_k\end{aligned}$$

Continuous

Discrete

Finite differences approximation for derivative

$$\dot{x} \approx \frac{x\left(\left(k+1\right)\Delta t\right) - x\left(k\Delta t\right)}{\Delta t} = \frac{x_{k+1} - x_k}{\Delta t}$$

$$\frac{x_{k+1} - x_k}{\Delta t} \approx Ax_k + Bu_k$$

$$x_{k+1} = \underbrace{\left[I + A\Delta t\right]}_{\Phi} x_k + \underbrace{B\Delta t}_{\Gamma} u_k$$

How good are the approximations?

Exact Discretization

$$x(t_k + \Delta t) = e^{A\Delta t} x(t_k) + e^{A\Delta t} \int_{t_k}^{t_k + \Delta t} e^{-A\sigma} d\sigma B u(t_k)$$

$$x_{k+1} = e^{A\Delta t} x_k + \left(e^{A\Delta t} - I\right) A^{-1} B u_k$$

$$\Phi = e^{A\Delta t}$$

$$\Gamma = \left(e^{A\Delta t} - I\right) A^{-1} B$$

Exact

$$\Phi = \left[I + A\Delta t\right]$$

$$\Gamma = B\Delta t$$

Approximate

Example Discretization

$$\left.\begin{aligned}
\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} &= \begin{bmatrix}-0.1 & 0\\ 0.04 & -0.04\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0.1\\ 0\end{bmatrix}u\\
y &= \begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}
\end{aligned}\right\}\qquad g_p(s) = \frac{1}{(10s+1)(25s+1)}$$

$$\Delta t = 3$$

$$\Phi = e^{A\Delta t} = \exp\begin{bmatrix}-0.3 & 0\\ 0.12 & -0.12\end{bmatrix} = \boxed{\begin{bmatrix}0.7408 & 0\\ 0.0974 & 0.8869\end{bmatrix}}$$

Exact

$$\Gamma = (\Phi - I)\begin{bmatrix}-0.1 & 0\\ 0.04 & -0.04\end{bmatrix}^{-1}\begin{bmatrix}0.1\\ 0\end{bmatrix} = \boxed{\begin{bmatrix}0.2592\\ 0.0157\end{bmatrix}}$$

$$\Phi = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} + \begin{bmatrix}-0.3 & 0\\ 0.12 & -0.12\end{bmatrix} = \boxed{\begin{bmatrix}0.7 & 0\\ 0.12 & 0.88\end{bmatrix}}$$

Approximate

$$\Gamma = \boxed{\begin{bmatrix}0.3\\ 0\end{bmatrix}}$$

Example, Continued

Discrete Transfer Function

$$g_p(z) = C(zI - \Phi)^{-1}\Gamma + D$$

Exact
$$g_p(z) = \frac{0.0157z + 0.0136}{z^2 - 1.6277z + 0.65702} = \frac{0.0157z^{-1} + 0.0136z^{-2}}{1 - 1.6277z^{-1} + 0.65702z^{-2}}$$

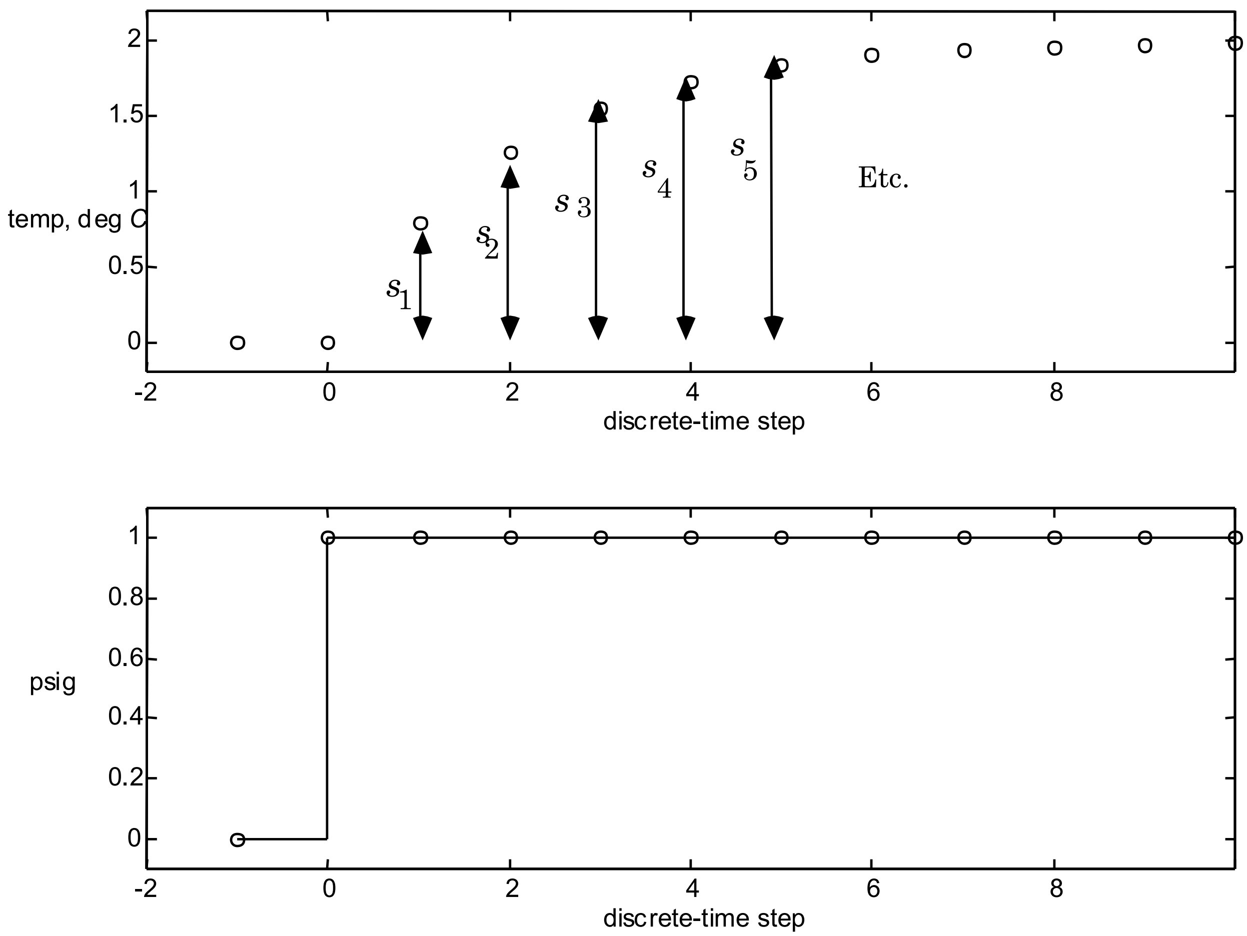
Poles/eigenvalues = 0.7408 & 0.8869

Approximate
$$g_p(z) = \frac{0.036}{z^2 - 1.58z + 0.616} = \frac{0.036z^{-2}}{1 - 1.58z^{-1} + 0.616z^{-2}}$$

Poles/eigenvalues = 0.7 & 0.88

“hidden slide” provided for additional background

Example Step Response Model



$$S = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & \cdots & s_N \end{bmatrix}^T$$

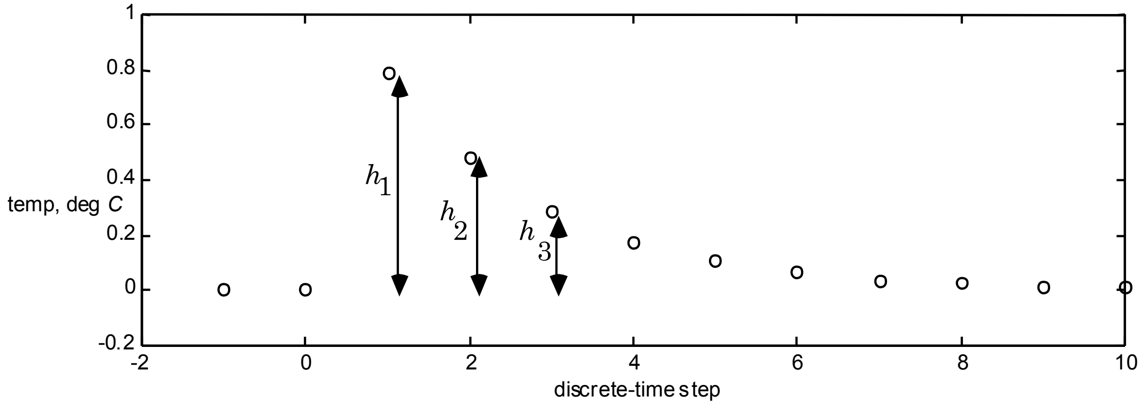
Step Response Model

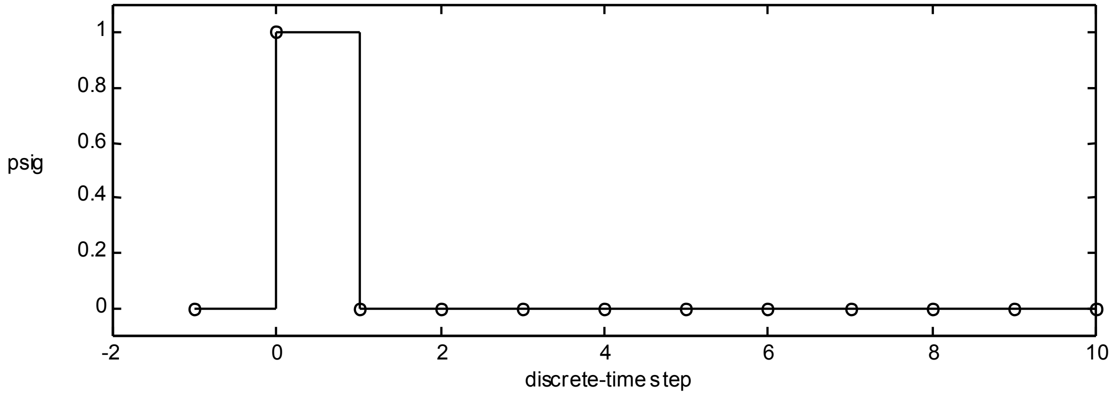
$$
\begin{aligned}
y_k &= \sum_{i=1}^{\infty} s_i \Delta u_{k-i} \\
&= s_1 \Delta u_{k-1} + \cdots + s_N \Delta u_{k-N} + s_{N+1} \Delta u_{k-N-1} + \cdots + s_{N+\infty} \Delta u_{k-\infty}
\end{aligned}
$$

$$
\begin{aligned}
y_k &= s_1 \Delta u_{k-1} + \cdots + s_{N-1} \Delta u_{k-N+1} + s_N \Delta u_{k-N} + \cdots + s_N \Delta u_{k-\infty} \\
&= s_1 \Delta u_{k-1} + \cdots + s_{N-1} \Delta u_{k-N+1} + s_N \underbrace{\left(\Delta u_{k-N} + \cdots + \Delta u_{k-\infty}\right)}_{u_{k-N}},
\end{aligned}
$$

$$
y_k = s_N u_{k-N} + \sum_{i=1}^{N-1} s_i \Delta u_{k-i}.
$$

Example Impulse Response Model





Impulse and step response coefficients are related

$$h_i = s_i - s_{i-1}$$

$$s_i = \sum_{j=1}^{i} h_j$$

Parameter Estimation, ARX Models

model output

measured output

known (measured) input

$$\hat{y}_1 = -a_1 y_0 - a_2 y_{-1} + b_1 u_0 + b_2 u_{-1}$$

$$\hat{y}_2 = -a_1 y_1 - a_2 y_0 + b_1 u_1 + b_2 u_0$$

$$\vdots$$

$$\hat{y}_N = -a_1 y_{N-1} - a_2 y_{N-2} + b_1 u_{N-1} + b_2 u_{N-2}$$

time step index

N pts

$$\underbrace{\begin{bmatrix} \hat{y}_1 \\ . \\ . \\ \hat{y}_N \end{bmatrix}}_{\hat{Y}} = \underbrace{\begin{bmatrix} y_0 & y_{-1} & u_0 & u_{-1} \\ . & . & . & . \\ . & . & . & . \\ y_{N-1} & y_{N-2} & u_{N-1} & u_{N-2} \end{bmatrix}}_{\Phi} \underbrace{\begin{bmatrix} -a_1 \\ -a_2 \\ b_1 \\ b_2 \end{bmatrix}}_{\Theta}$$

$$\hat{Y} = \Phi \quad \Theta$$

could change sign on columns 1 & 2 of Φ

Optimization/Parameter Solution

$$\min_{a_1,a_2,b_1,b_2} \sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2$$

Objective Function

$$\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2 = \left(Y - \hat{Y}\right)^T \left(Y - \hat{Y}\right) = \left(Y - \Phi\Theta\right)^T \left(Y - \Phi\Theta\right)$$

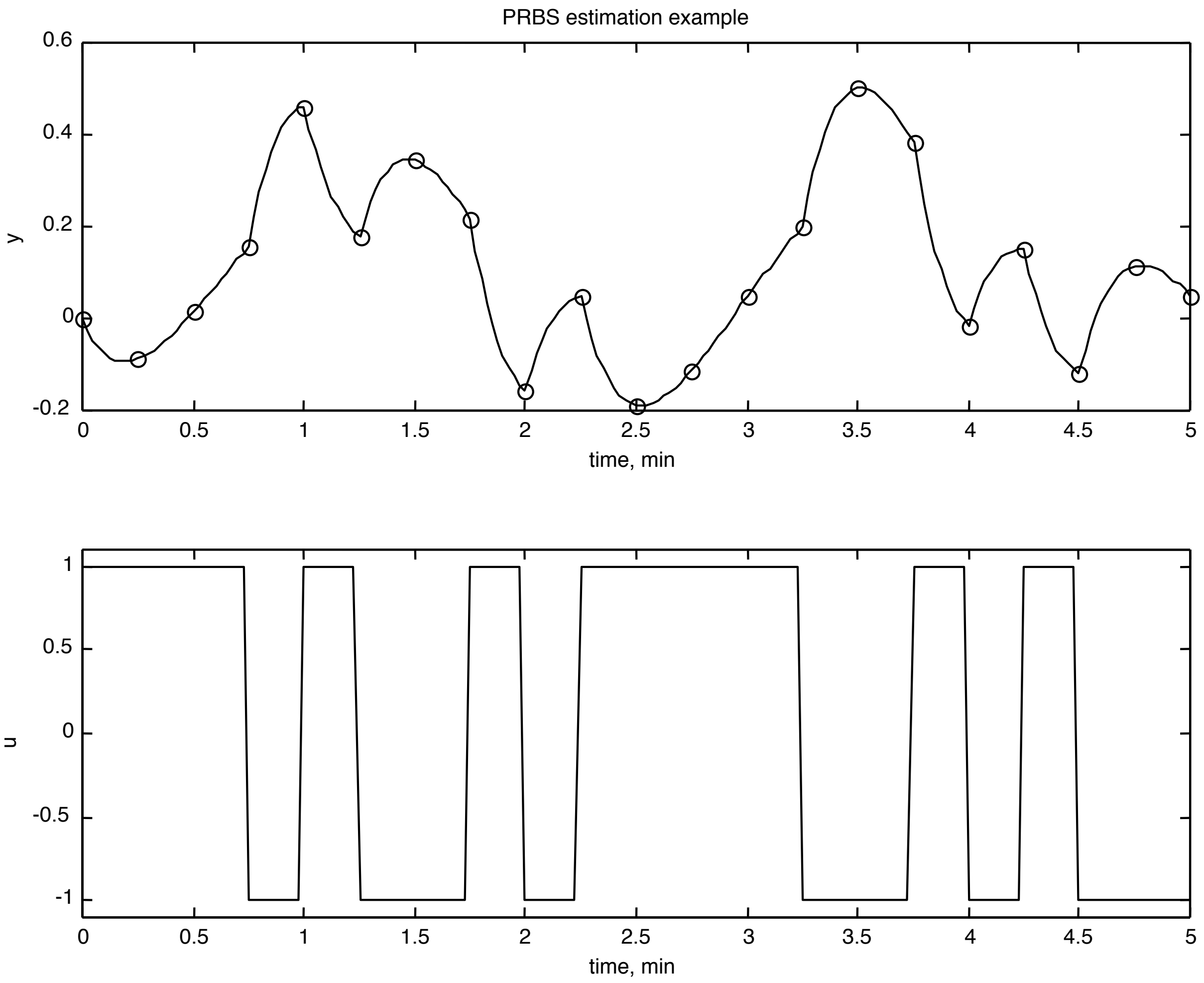
$$\Theta = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y$$

Solution

parameter
estimate
vector

measured
output
vector

Example



PRBS input

Result

$$\Phi = \begin{bmatrix} -0.0889 & 0 & 1 & 1 \\ 0.0137 & -0.0889 & 1 & 1 \\ 0.1564 & 0.0137 & -1 & 1 \\ 0.4618 & 0.1564 & 1 & -1 \\ 0.1771 & 0.4618 & -1 & 1 \\ 0.3446 & 0.1771 & -1 & -1 \\ 0.2171 & 0.3446 & 1 & -1 \\ -0.1558 & 0.2171 & -1 & 1 \\ 0.0485 & -0.1558 & 1 & -1 \\ -0.1879 & 0.0485 & 1 & 1 \\ -0.1123 & -0.1879 & 1 & 1 \\ 0.0463 & -0.1123 & 1 & 1 \\ 0.2003 & 0.0463 & -1 & 1 \\ 0.5007 & 0.2003 & -1 & -1 \\ 0.3846 & 0.5007 & 1 & -1 \\ -0.0172 & 0.3846 & -1 & 1 \\ 0.1513 & -0.0172 & 1 & -1 \\ -0.1162 & 0.1513 & -1 & 1 \\ 0.1134 & -0.1162 & -1 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} 0.0137 \\ 0.1564 \\ 0.4618 \\ 0.1771 \\ 0.3446 \\ 0.2171 \\ -0.1558 \\ 0.0485 \\ -0.1879 \\ -0.1123 \\ 0.0463 \\ 0.2003 \\ 0.5007 \\ 0.3846 \\ -0.0172 \\ 0.1513 \\ -0.1162 \\ 0.1134 \\ 0.0502 \end{bmatrix}$$

$$\Theta = \left(\Phi^T \Phi\right)^{-1} \Phi^T Y$$

$$\Theta = \begin{bmatrix} 1.1196 \\ -0.3133 \\ -0.0889 \\ 0.2021 \end{bmatrix} = \begin{bmatrix} -a_1 \\ -a_2 \\ b_1 \\ b_2 \end{bmatrix}$$

$$g_p(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} = \frac{-0.0889z + 0.2021}{z^2 - 1.1196z + 0.3133}$$

$$= \frac{-0.0889z^{-1} + 0.2021z^{-2}}{1 - 1.1196z^{-1} + 0.3133z^{-2}}$$

$$= \frac{-0.0889(z - 2.274)}{(z - 0.5716)(z - 0.5481)}$$

Identified Model

$$g_p(z) = \frac{b_1 z + b_2}{z^2 + a_1 z + a_2} = \frac{-0.0889z + 0.2021}{z^2 - 1.1196z + 0.3133}$$

$$y(z) = \frac{-0.0889z^{-1} + 0.2021z^{-2}}{1 - 1.1196z^{-1} + 0.3133z^{-2}} \cdot u(z)$$

$$y_k = 1.1196y_{k-1} - 0.3133y_{k-2} - 0.0889u_{k-1} + 0.2021u_{k-2}$$

Subspace Identification

- Subspace ID can be used to develop discrete state space models from input-output data

Summary

- Discrete models
  - State space, ARX, discrete transfer function
- Zeros & poles
  - Poles inside unit circle are stable (negative = oscillate)
  - Zeros inside unit circle have stable inverses
- Parameter estimation
  - Example with PRBS input
- Step and impulse response models

Discrete (Digital) Control

B. Wayne Bequette

- Review of Digital PID
- Review of Model-based Digital Control
- Discrete Internal Model Control
- Examples
- Summary

Rensselaer

Chemical and Biological Engineering

Proportional-Integral-Derivative (PID) Control

Error = setpoint – measured output

$$e(t) = r(t) - y(t)$$

$$u(t) = u_0 + k_c\left[e(t) + \frac{1}{\tau_I}\int_0^t e(t)dt + \tau_D \frac{de(t)}{dt}\right]$$

Manipulated Input **Proportional gain** **Integral time** **Derivative time**

$$e(k) = e(t_k)$$

$$\int_0^{t_k} e(t)dt \approx e(t_1)\cdot\Delta t + e(t_2)\cdot\Delta t + \cdots + e(t_k)\cdot\Delta t \approx \sum_{i=1}^{k} e(t_i)\Delta t = \sum_{i=1}^{k} e(i)\Delta t$$

$$\frac{de(t_k)}{dt} \approx \frac{e(k) - e(k-1)}{\Delta t}$$

Substituting each term, we find the discrete controller

$$u(k) = u_0 + k_c\left[e(k) + \frac{\Delta t}{\tau_I}\sum_{i=0}^{k} e(i) + \frac{\tau_D}{\Delta t}\big(e(k) - e(k-1)\big)\right]$$

It is convenient to work with the “velocity form.” First, generate the equation for step k-1

$$u(k-1) = u_0 + k_c\left[e(k-1) + \frac{\Delta t}{\tau_I}\sum_{i=0}^{k-1} e(i) + \frac{\tau_D}{\Delta t}\big(e(k-1) - e(k-2)\big)\right]$$

Then subtract u(k-1) from u(k) to find

$$u(k) = u(k-1) + k_c\left[\left(1 + \frac{\Delta t}{\tau_I} + \frac{\tau_D}{\Delta t}\right)e(k) + \left(-1 - \frac{2\tau_D}{\Delta t}\right)e(k-1) + \frac{\tau_D}{\Delta t}e(k-2)\right]$$

$$u(k) = u(k-1) + k_c\left[\left(1 + \frac{\Delta t}{\boldsymbol{\tau}_I} + \frac{\boldsymbol{\tau}_D}{\Delta t}\right)e(k) + \left(-1 - \frac{2\boldsymbol{\tau}_D}{\Delta t}\right)e(k-1) + \frac{\boldsymbol{\tau}_D}{\Delta t}e(k-2)\right]$$

$$b_2 = k_c\left(1 + \frac{\Delta t}{\boldsymbol{\tau}_I} + \frac{\boldsymbol{\tau}_D}{\Delta t}\right),$$

$$b_1 = -k_c\left(1 + \frac{2\boldsymbol{\tau}_D}{\Delta t}\right),$$

$$b_0 = \frac{k_c\boldsymbol{\tau}_D}{\Delta t}$$

$$u(k) = u(k-1) + b_2 e(k) + b_1 e(k-1) + b_0 e(k-2)$$

$$u(k) - u(k-1) = b_2 e(k) + b_1 e(k-1) + b_0 e(k-2)$$

$$\mathrm{Z}[u(k)] = u(z),$$
$$\mathrm{Z}[u(k-1)] = z^{-1}u(z),$$
$$\mathrm{Z}[u(k-2)] = z^{-2}u(z).$$

$$\mathrm{Z}[e(k)] = e(z),$$
$$\mathrm{Z}[e(k-1)] = z^{-1}e(z),$$
$$\mathrm{Z}[e(k-2)] = z^{-2}e(z).$$

$$\left(1 - z^{-1}\right)u(z) = \left(b_2 + b_1 z^{-1} + b_0 z^{-2}\right)e(z)$$

$$u(z) = \frac{\left(b_2 + b_1 z^{-1} + b_0 z^{-2}\right)}{1 - z^{-1}}e(z)$$

$$u(z) = \frac{\left(b_2 z^2 + b_1 z + b_0\right)}{z^2 - z}e(z) = g_c(z)e(z)$$

PID Tuning

- Usually based on continuous-time methods, including:
- Ziegler-Nichols closed-loop oscillations
- Cohen-Coon
- IMC-based PID
- Skogestad's tuning method
- Frequency response
- SWAG (most common)

IMC-Based PID

- Continuous Model

$$\tilde{g}_p(s) = \frac{k_p e^{-\theta s}}{\tau_p s + 1}$$

- PID Parameters

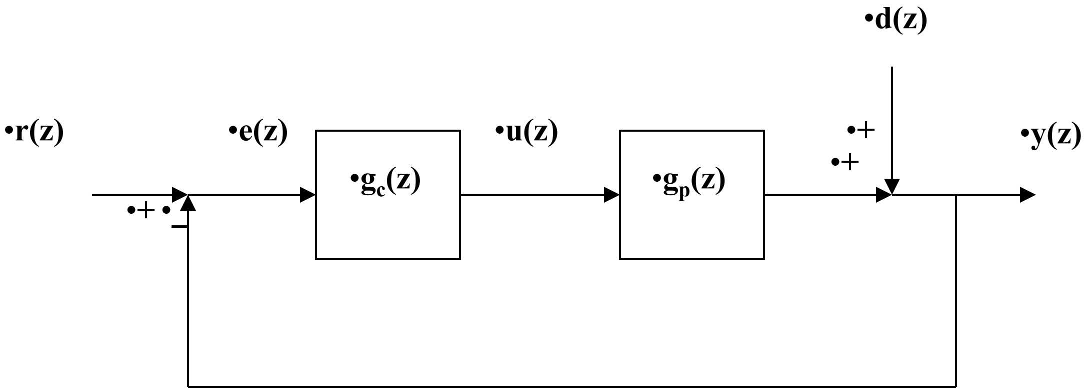
$$k_c = \frac{\left(\tau_p + 0.5\theta\right)}{k_p\left(\lambda + 0.5\theta\right)},$$

$$\tau_I = \tau_p + 0.5\theta,$$

$$\tau_D = \frac{\tau_p \theta}{2\tau_p + \theta}.$$

λ is the tuning parameter – related to the closed-loop time constant

Digital Control: Block Diagram



- Stability analysis
- Poles of CLTF must be inside the unit circle

$$y(z)=\frac{g_p(z)g_c(z)}{\underbrace{1+g_p(z)g_c(z)}}\cdot r(z)$$

$$y(z)=\quad g_{CL}(z)\quad r(z)$$

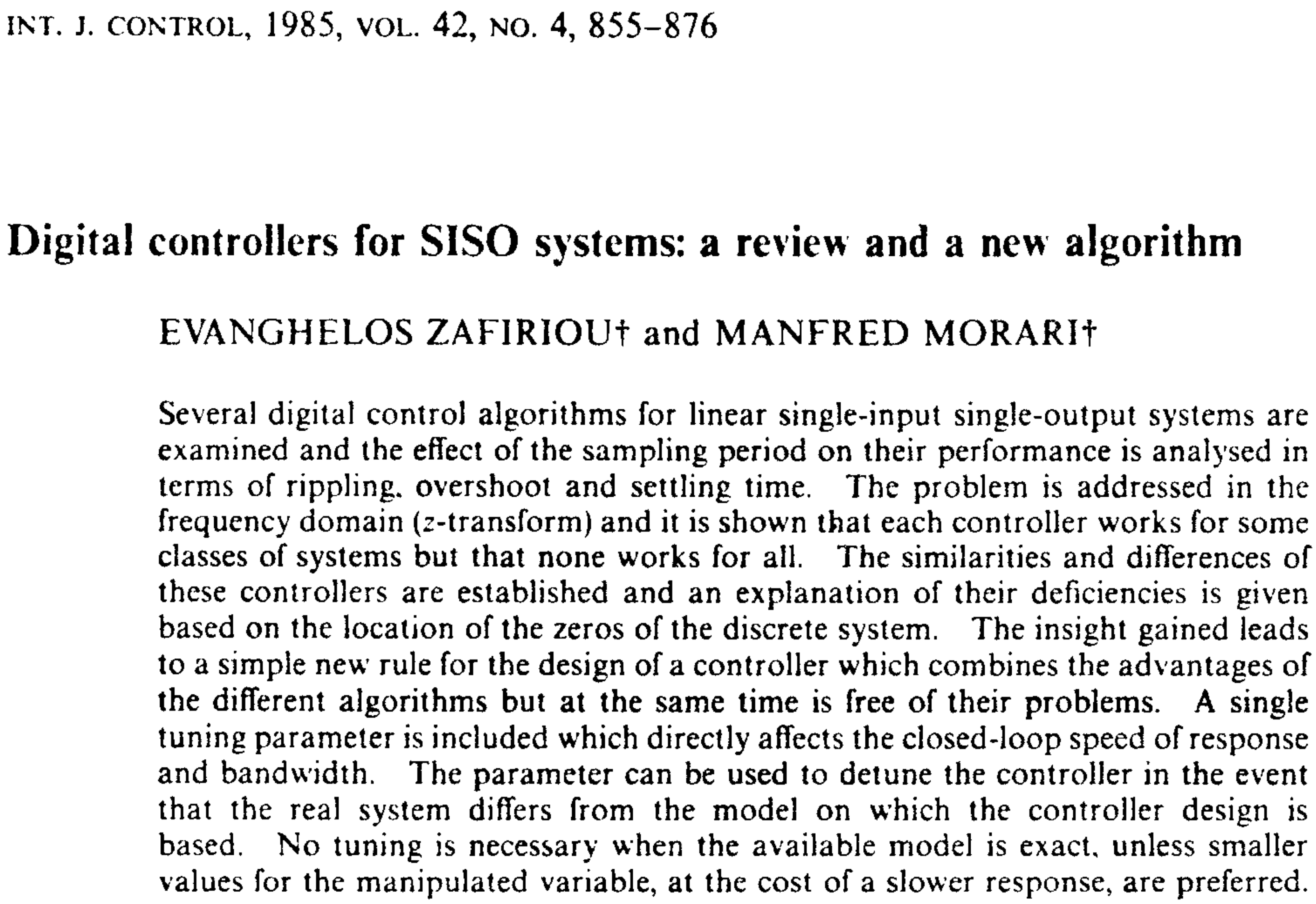
Closed-loop transfer function

Suggestion

- Work through **Workshop on Digital PID Control**

- Find discrete time model
- Digital PID
- Tune for Verge of Instability (continuous oscillations)
  - Find closed-loop poles
- Increase gain for unstable closed-loop
  - Find closed-loop poles

Next Topic: Internal Model Control (following slides)



INT. J. CONTROL, 1985, VOL. 42, NO. 4, 855–876

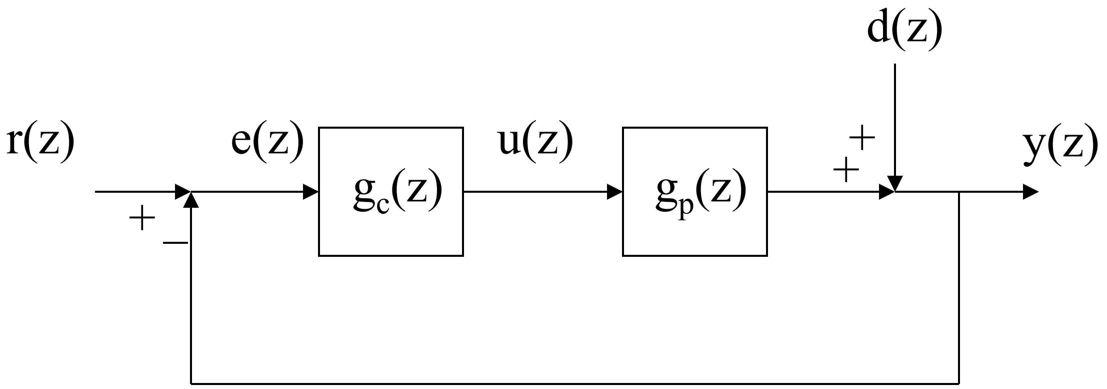
Digital controllers for SISO systems: a review and a new algorithm

EVANGHELOS ZAFIRIOU† and MANFRED MORARI†

Several digital control algorithms for linear single-input single-output systems are examined and the effect of the sampling period on their performance is analysed in terms of rippling, overshoot and settling time. The problem is addressed in the frequency domain (z-transform) and it is shown that each controller works for some classes of systems but that none works for all. The similarities and differences of these controllers are established and an explanation of their deficiencies is given based on the location of the zeros of the discrete system. The insight gained leads to a simple new rule for the design of a controller which combines the advantages of the different algorithms but at the same time is free of their problems. A single tuning parameter is included which directly affects the closed-loop speed of response and bandwidth. The parameter can be used to detune the controller in the event that the real system differs from the model on which the controller design is based. No tuning is necessary when the available model is exact, unless smaller values for the manipulated variable, at the cost of a slower response, are preferred.

Manfred Morari

Digital Feedback Control



- Digital Controller Design
  - Response specification-based

$$y(z) = \underbrace{\frac{g_p(z)g_c(z)}{1+g_p(z)g_c(z)}} \cdot r(z)$$

$$y(z) = g_{CL}(z) \quad r(z)$$

Desired response

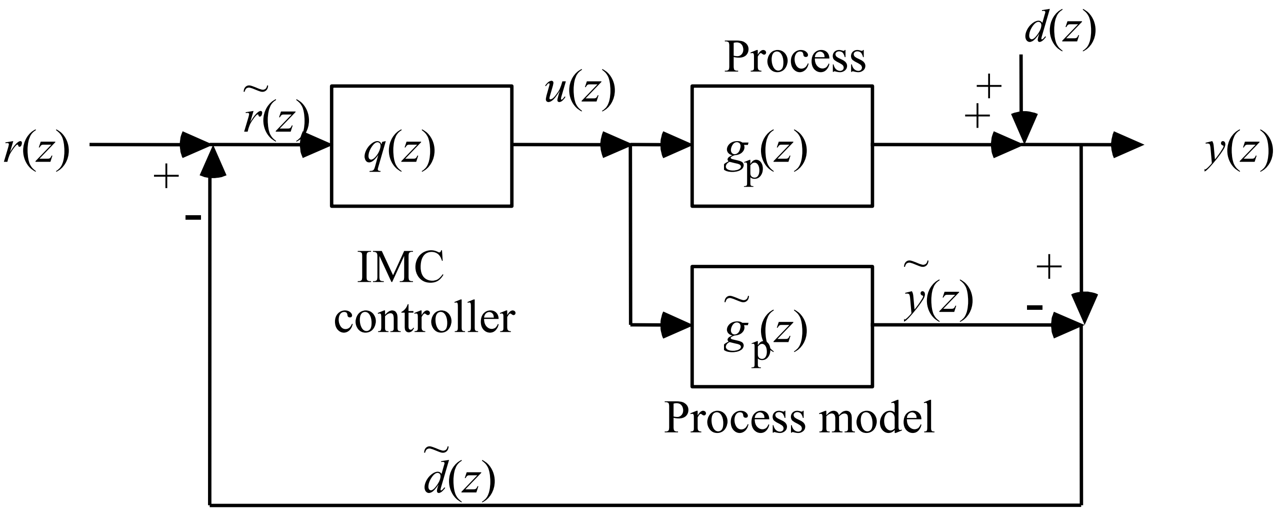
Controller Design

Desired response $$g_{CL}(z) = \frac{g_p(z)g_c(z)}{1 + g_p(z)g_c(z)}$$

$$g_c(z) = \frac{1}{g_p(z)} \cdot \frac{g_{CL}(z)}{1 - g_{CL}(z)}$$

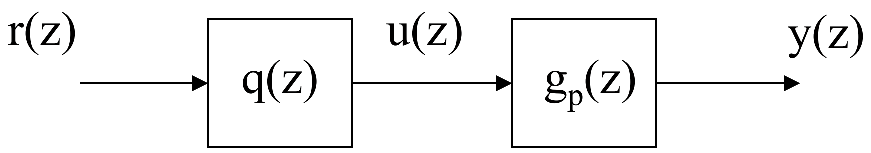
solve for controller

Internal Model Form



For **controller design**, consider perfect model and no disturbances

IMC Design



$$y(z) = \underbrace{g_p(z)\ q(z)}\ r(z)$$

$$y(z) = g_{CL}(z)\ r(z)$$

Desired response

solve for controller

$$q(z) = \frac{g_{CL}(z)}{g_p(z)}$$

Really based on the model

$$q(z) = \frac{g_{CL}(z)}{\tilde{g}_p(z)} = g_{CL}(z)\tilde{g}_p^{-1}(z)$$

Can implement IMC in Classic Feedback Form

$$g_c(z) = \frac{q(z)}{1 - q(z)\tilde{g}_p(z)}$$

Digital Controller Design

John Ragazzini

- Deadbeat

  BERGEN, A. R., and RAGAZZINI, J. R., 1954, *A.I.E.E. Trans.*, **73**, 236.

- Dahlin's Controller

  DAHLIN, E. B., 1968, *Instrum. Control Syst.*, **41**, 77.

- State Deadbeat

  BERGEN, A. R., and RAGAZZINI, J. R., 1954, *A.I.E.E. Trans.*, **73**, 236.

- State Deadbeat with Filter (Vogel-Edgar)

  VOGEL, E. F., 1982, Adaptive control of chemical processes with variable dead time. Ph.D. dissertation, University of Texas, Austin.

- Modified Dahlin's Controller

- NEW (IMC) DESIGN

Deadbeat

- Achieve setpoint in the minimum time (if no time-delay, then one time step)

$$y_{k+1} = r_k$$

$$y(z) = z^{-1} r(z)$$

Backwards shift operator

$$g_{CL}(z) = z^{-1}$$

$$q(z) = z^{-1} \tilde{g}_p^{-1}(z)$$

IMC Form

$$g_c(z) = \tilde{g}_p^{-1}(z) \cdot \frac{z^{-1}}{1 - z^{-1}}$$

Classic Feedback Form

Example 0, FO Process

$$\tilde{g}_p(s) = \frac{1}{10s+1} \xrightarrow[\Delta t = 1]{} \tilde{g}_p(z) = \frac{0.0952z^{-1}}{1-0.9048z^{-1}}$$

$$q(z) = \frac{z^{-1}\left(1-0.9048z^{-1}\right)}{0.0952z^{-1}} = \frac{z-0.9048}{0.0952z+0}$$

$u(z) = q(z)r(z)$ Control action

$let\ \ r(z) = \dfrac{1}{1-z^{-1}} = \text{unit step}$ setpoint change

$$u(z) = \frac{z^{-1}-0.9048z^{-2}}{0.0952z^{-1}\left(1-z^{-1}\right)} = \frac{10.504-9.504z^{-1}}{1-z^{-1}}$$

$u(z) = \dfrac{10.504}{1-z^{-1}} + \dfrac{-9.504z^{-1}}{1-z^{-1}}$ Large control action up, then down

Classic Feedback Form for Deadbeat Design

$$g_c(z) = \tilde{g}_p^{-1}(z) \cdot \frac{z^{-1}}{1 - z^{-1}} = \tilde{g}_p^{-1}(z) \cdot \frac{1}{z - 1}$$

$$\tilde{g}_p(z) = \frac{0.0952 z^{-1}}{1 - 0.9048 z^{-1}} = \frac{0.0952}{z - 0.9048}$$

$$\tilde{g}_p^{-1}(z) = \frac{1 - 0.9048 z^{-1}}{0.0952 z^{-1}} = \frac{z - 0.9048}{0.0952}$$

$$g_c(z) = \frac{z - 0.9048}{0.0952} \cdot \frac{1}{z - 1} = \left(\frac{1}{0.0952} \right) \cdot \frac{z - 0.9048}{z - 1}$$

$$= \left(\frac{1}{0.0952} \right) \cdot \frac{1 - 0.9048 z^{-1}}{1 - z^{-1}} = 10.5 \cdot \frac{1 - 0.9048 z^{-1}}{1 - z^{-1}}$$

Classic Feedback Implementation

$$g_c(z) = 10.5 \cdot \frac{1 - 0.9048z^{-1}}{1 - z^{-1}}$$

$$u(z) = g_c(z)e(z)$$

$$\left(1 - z^{-1}\right)u(z) = 10.5 \cdot \left(1 - 0.9048z^{-1}\right)e(z)$$

$$u_k - u_{k-1} = 10.5 \cdot \left(e_k - 0.9048e_{k-1}\right)$$

$$\boxed{u_k = u_{k-1} + 10.5 \cdot \left(e_k - 0.9048e_{k-1}\right)}$$

Note that this is a PI controller

$$u_k = u_{k-1} + 10.5 \cdot \left(e_k - 0.9048 e_{k-1}\right)$$

$$u(k) = u(k-1) + b_2 e(k) + b_1 e(k-1) + b_0 e(k-2)$$

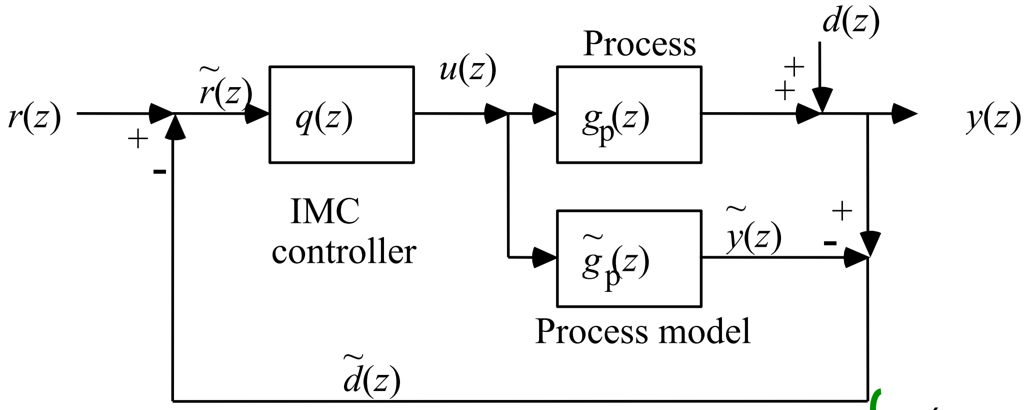
$$b_2 = k_c \left(1 + \frac{\Delta t}{\tau_I} + \frac{\tau_D}{\Delta t}\right) = 10.5$$

$$b_1 = -k_c \left(1 + \frac{2\tau_D}{\Delta t}\right) = -10.5 * 0.9048 = -9.5$$

$$b_0 = \frac{k_c \tau_D}{\Delta t} = 0$$

The PI parameters are then $k_c = 9.5$ $\tau_I = 9.5$

IMC Implementation



$$\tilde{y}(z)=\tilde{g}_p(z)u(z) \longrightarrow \left\{\begin{array}{l} (1-0.9048z^{-1})\tilde{y}(z)=0.0952z^{-1}u(z) \\ \tilde{y}_k = 0.9048\tilde{y}_{k-1}+0.0952u_{k-1} \end{array}\right.$$

$$\tilde{d}(z)=y(z)-\tilde{y}(z) \longrightarrow \tilde{d}_k = y_k - \tilde{y}_k$$

$$\tilde{r}(z)=r(z)-\tilde{d}(z) \longrightarrow \tilde{r}_k = r_k - \tilde{d}_k$$

$$u(z)=q(z)\tilde{r}(z) \longrightarrow \left\{\begin{array}{l} 0.0952z^{-1}u(z)=z^{-1}(1-0.9048z^{-1})\tilde{r}(z) \\ u(z)=\left(\dfrac{1}{0.0952}\right)(1-0.9048z^{-1})\tilde{r}(z) \\ \boxed{u_k = \left(\dfrac{1}{0.0952}\right)(\tilde{r}_k - 0.9048\tilde{r}_{k-1})} \end{array}\right.$$

Discussion

Standard feedback control clearly has the current control action as a function of the previous control action. Why doesn't IMC?

Standard Feedback

$$e_k = r_k - y_k$$

$$u_k = u_{k-1} + 10.5 \cdot \left(e_k - 0.9048 e_{k-1}\right)$$

IMC

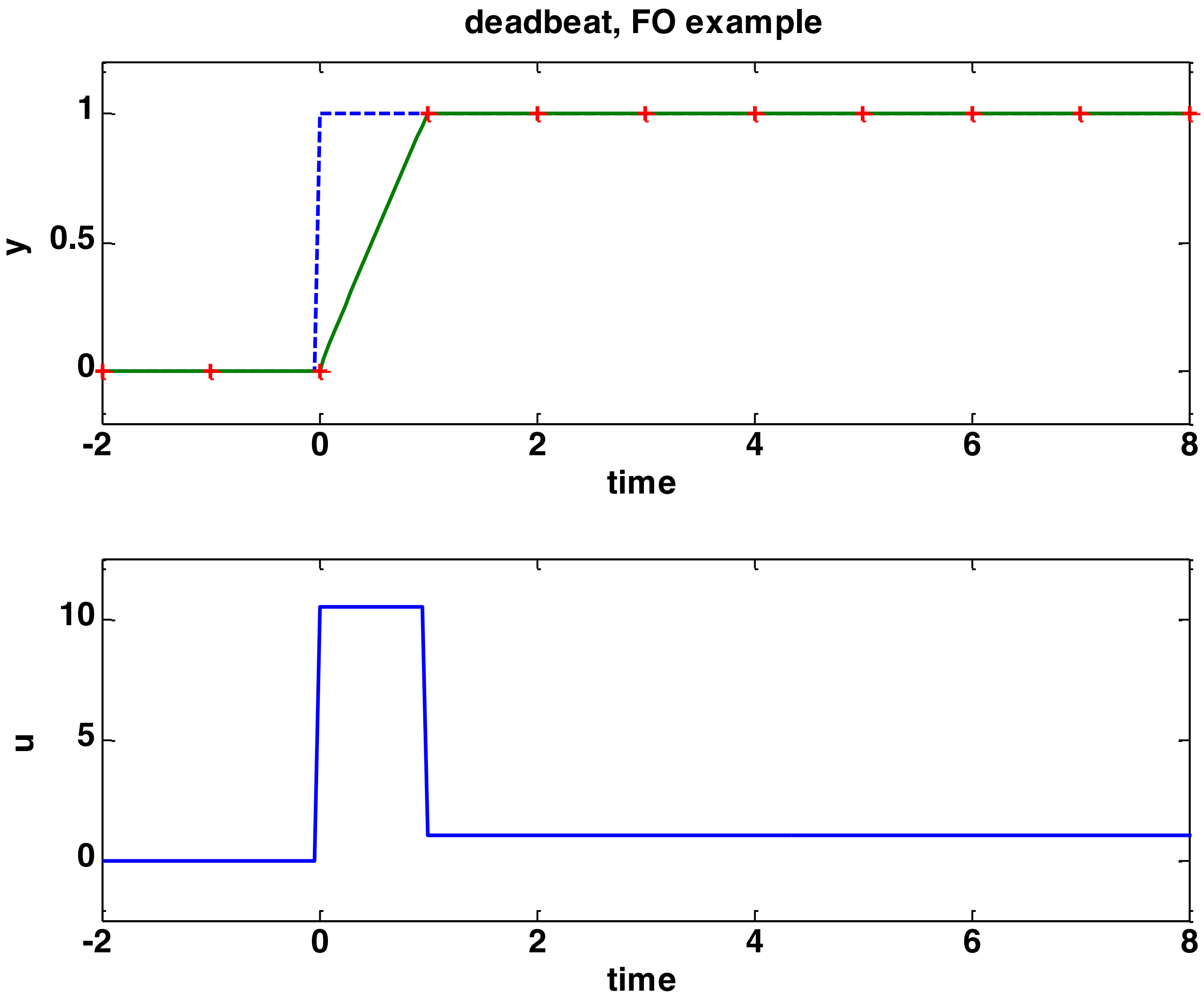
$$\tilde{y}_k = 0.9048 \tilde{y}_{k-1} + 0.0952 u_{k-1}$$

$$\tilde{d}_k = y_k - \tilde{y}_k$$

$$\tilde{r}_k = r_k - \tilde{d}_k$$

$$u_k = \left(\frac{1}{0.0952}\right)\left(\tilde{r}_k - 0.9048 \tilde{r}_{k-1}\right)$$

Example 0 (First-order), Deadbeat Design



Example 1, Second-Order Process

$$\tilde{g}_p(s) = \frac{1}{(10s+1)(25s+1)} \xrightarrow[\Delta t = 3]{} \tilde{g}_p(z) = \frac{0.0157z^{-1} + 0.0136z^{-2}}{1 - 1.6277z^{-1} + 0.65702z^{-2}}$$

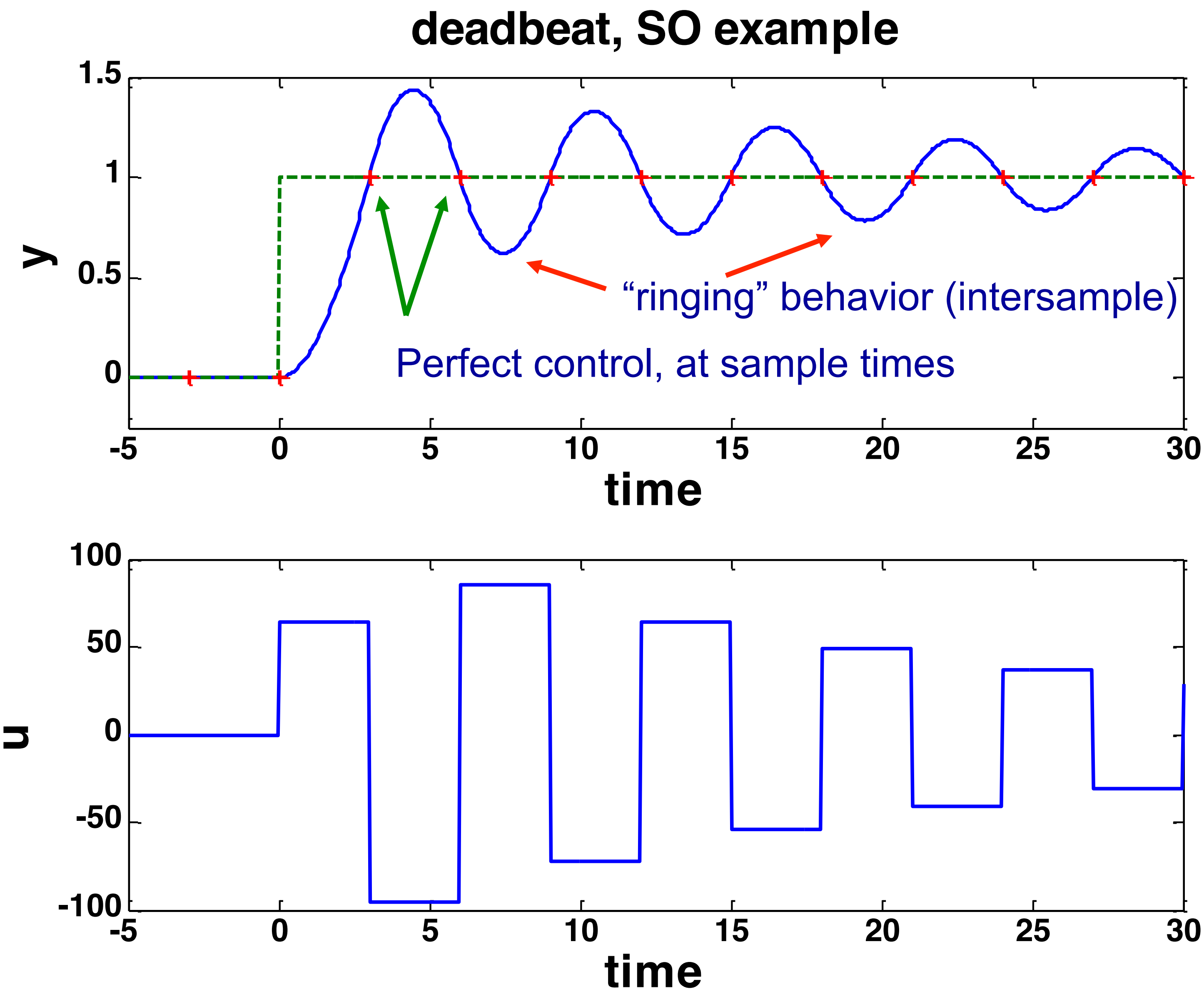
zero = -0.0136/0.0157 = -0.8694

$$q(z) = \frac{z^{-1} - 1.6277z^{-2} + 0.65702z^{-3}}{0.0157z^{-1} + 0.0136z^{-2}} = \frac{z^2 - 1.6277z + 0.65702}{0.0157z^2 + 0.0136z + 0}$$

controller has pole = -0.8694, which is stable, but oscillates

“ringing” behavior as shown next

Example 1 (Second-order): Deadbeat Design



"ringing" behavior (intersample)

Perfect control, at sample times

Dahlin’s Controller

- Desired first-order + time-delay response to setpoint change

For no time-delay: $g_{CL}(z) = \dfrac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}}$ where $\alpha = \exp(-\Delta t/\lambda)$

$$q(z) = \frac{g_{CL}(z)}{\tilde{g}_p(z)} = \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}} \cdot \tilde{g}_p^{-1}(z) = \frac{(1-\alpha)}{(z-\alpha)} \cdot \tilde{g}_p^{-1}(z)$$

For the second-order example:

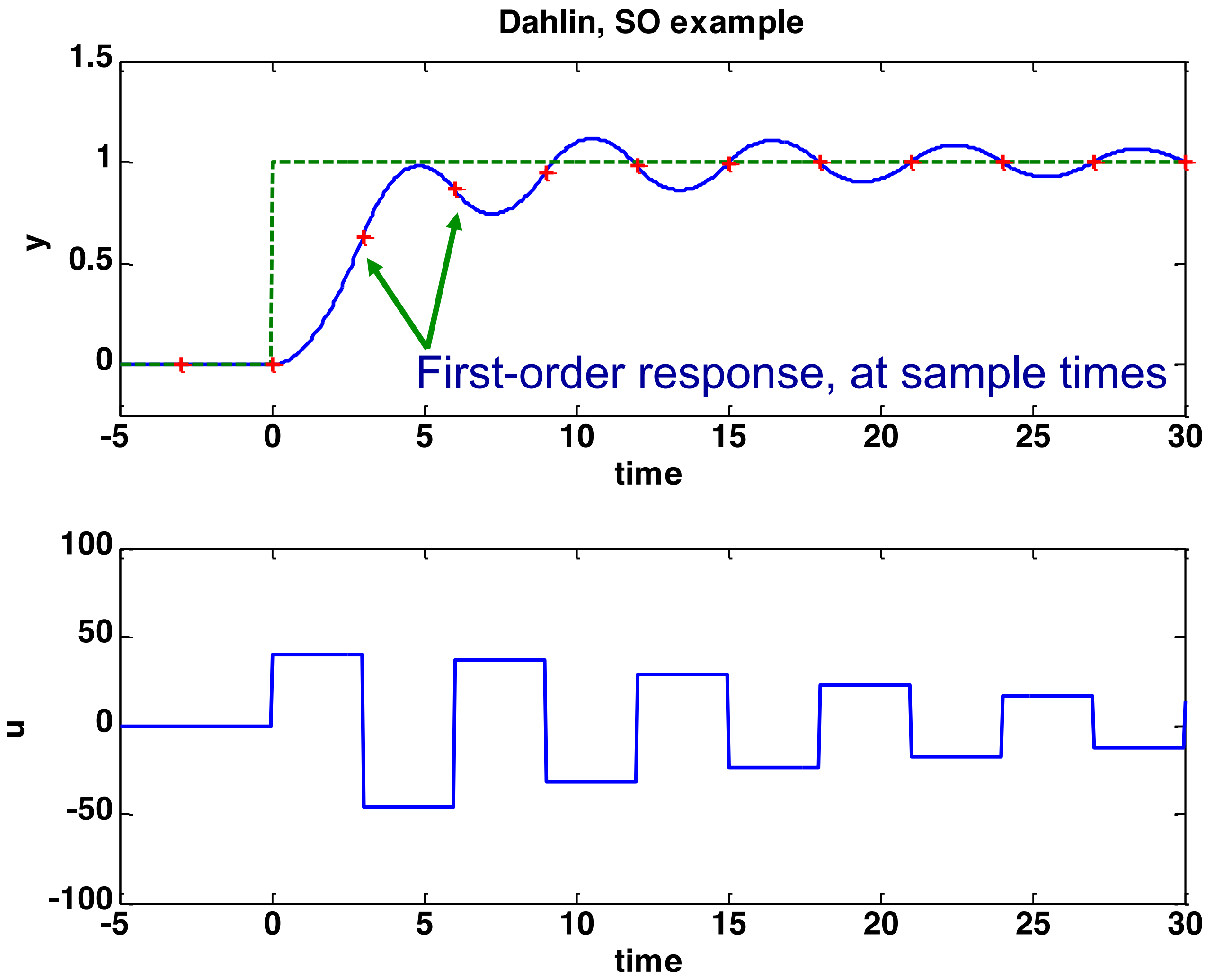
$$q(z) = \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}} \cdot \frac{1-1.6277z^{-1}+0.65702z^{-2}}{0.0157z^{-1}+0.0136z^{-2}}$$

$$= \frac{(1-\alpha)}{(z-\alpha)} \cdot \frac{z^2-1.6277z+0.65702}{0.0157z+0.0136}$$

Second-order Process: Dahlin’s Controller

For $\lambda = 3$,

$\alpha = 0.3679$



First-order response, at sample times

Still “ringing”, but more damped than deadbeat

Dahlin’s Modified Controller

- Substitute zeros at origin for unstable (|zero|>1) or ringing (|zero|<1 but negative) zeros. Also, keep the gain the same

- Problem: Dahlin applied this to $g_c(z)$, but should have applied it to the IMC form, q(z)!

“hidden slide” provided for additional background

State Deadbeat Controller Design

- Brings outputs & inputs to new steady-state in the minimum number of time steps
- Does not invert zeros at all

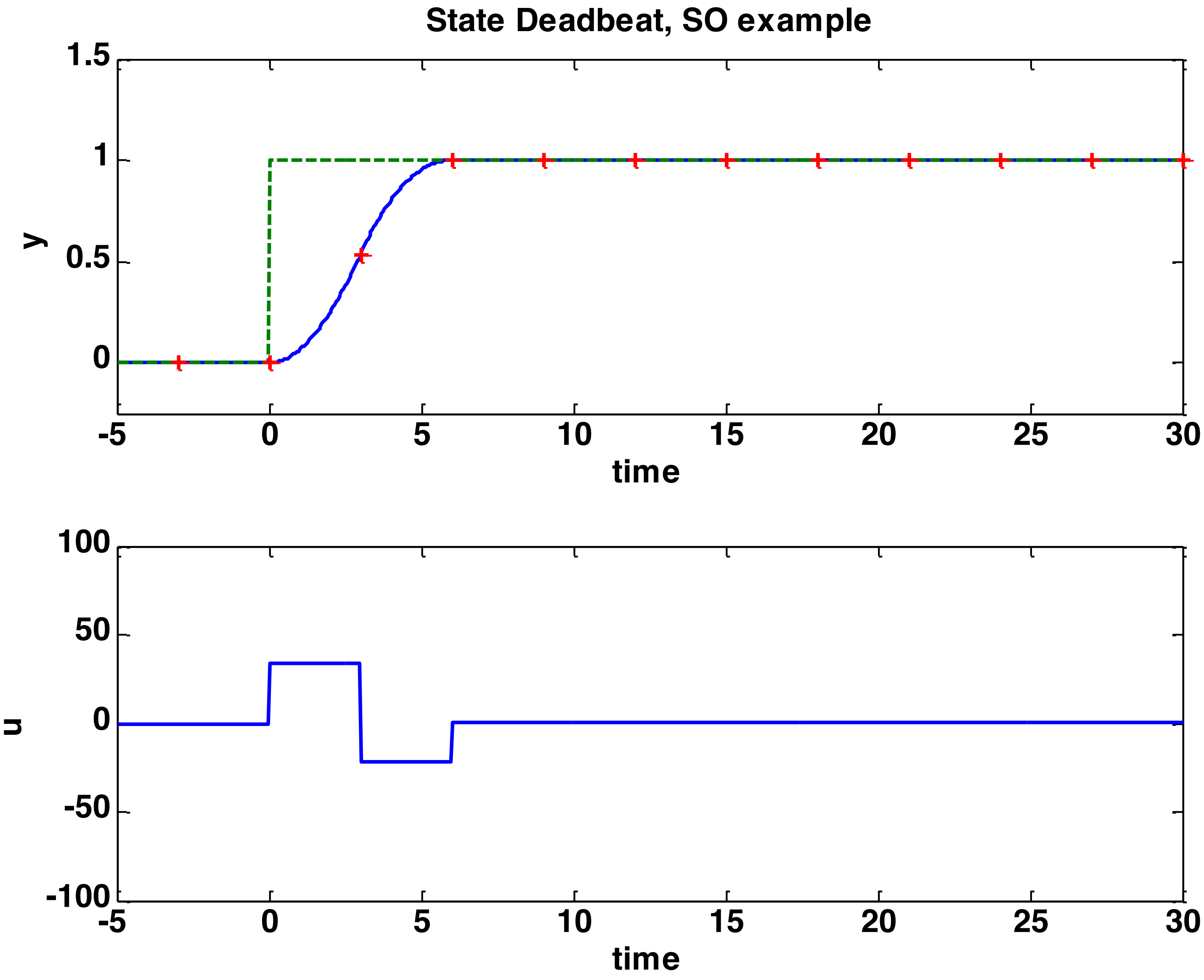
$$\tilde{g}_p(z) = \frac{0.0157z^{-1} + 0.0136z^{-2}}{1 - 1.6277z^{-1} + 0.65702z^{-2}}$$

second-order example

$$= \underbrace{\frac{0.0157 + 0.0136}{1 - 1.6277z^{-1} + 0.65702z^{-2}}}_{\tilde{g}_{p-}(z)} \cdot \underbrace{\frac{0.0157z^{-1} + 0.0136z^{-2}}{0.0157 + 0.0136}}_{\tilde{g}_{p+}(z)}$$

$$q(z) = \tilde{g}_{p-}^{-1} = \frac{1 - 1.6277z^{-1} + 0.65702z^{-2}}{0.0293} = \frac{z^2 - 1.6277z + 0.65702}{0.0293z^2 + 0z + 0}$$

Second-order Example: State Deadbeat Design



Controller Forms

zero inside unit circle = a_i^-

a_i^+= zero outside unit circle

$$\tilde{g}_p(z) = G(z) = \frac{K_{pz}(z-a_1^-)\cdots(z-a_k^-)(z-a_{k+1}^+)\cdots(z-a_{n-1}^+)}{(z-p_1)\cdots(z-p_n)}$$

Zafiriou & Morari notation

Assumes stable poles

State Deadbeat:

$$q_{SD}(z) = \frac{(z-p_1)\cdots(z-p_n)}{K_{pz}(1-a_1^-)\cdots(1-a_k^-)z^n}$$

State Deadbeat with Filter (Vogel-Edgar):

$$q_{VE}(z) = q_{SD}F(z) = \frac{(z-p_1)\cdots(z-p_n)}{K_{pz}(1-a_1^-)\cdots(1-a_k^-)z^n} \cdot \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}}$$

Notice that neither the state deadbeat nor Vogel-Edgar controllers try to invert zeros (even good ones!)

IMC Design Summary

- Model Factorization ("good stuff" and "bad stuff")

$$\tilde{g}_p(z) = \tilde{g}_{p-}(z)\tilde{g}_{p+}(z)$$

- numerator one order less than denominator

- zeros outside unit circle
- zeros inside unit circle that are negative
- "all-pass" by including pole at $1/z_i$ for each positive z_i outside the unit circle (but not the negative ones)

- Controller: Invert "good stuff" part of model

$$q(z) = \tilde{g}_{p-}^{-1}(z)F(z)$$

- "Filter" for desired response, often first-order

Example 2: Third-Order (3-tank)

$$\tilde{g}_p(s) = \frac{1}{(s+1)^3} \xrightarrow{\textbf{Sample time = 1}} \tilde{g}_p(z) = \frac{0.0803(z+1.7990)(z+0.1238)}{(z-0.3679)^3}$$

zeros at -1.7990 (outside unit circle)

-0.1238 (inside, but negative)

$$\tilde{g}_p(z) = \qquad \tilde{g}_{p-}(z) \qquad \qquad \tilde{g}_{p+}(z)$$

$$\tilde{g}_p(z) = \frac{0.0803(2.7990)(1.1238)z^2}{(z-0.3679)^3} \cdot \underbrace{\frac{(z+1.7990)(z+0.1238)}{(2.7990)(1.1238)z^2}}_{\text{Note gain is 1 (value at z = 1)}}$$

$$q(z) = \tilde{g}_{p-}^{-1}(z)F(z) = \frac{(z-0.3679)^3}{0.2526z^2} \cdot \frac{(1-\alpha)z^{-1}}{1-\alpha z^{-1}}$$

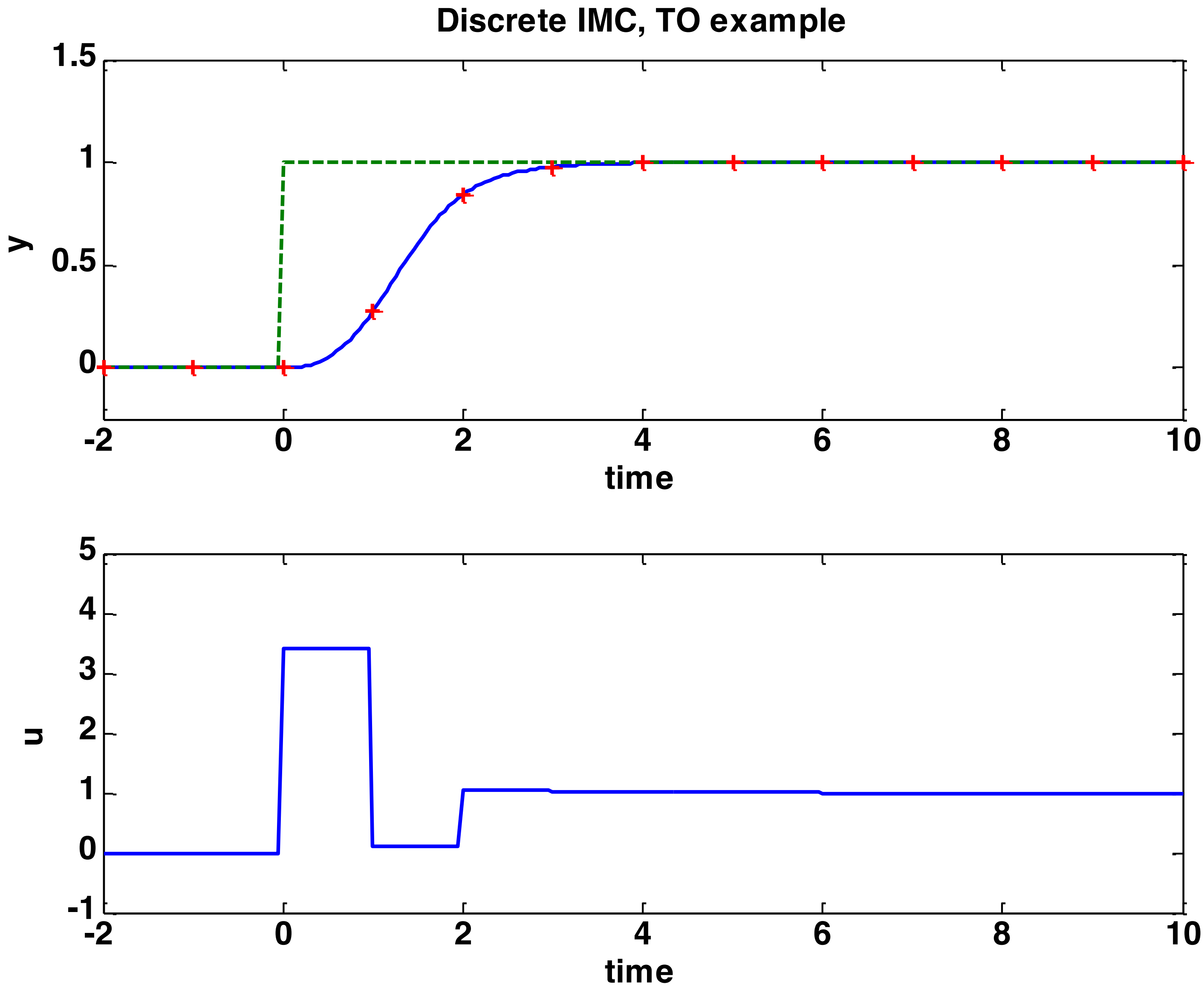
$$= \frac{(z-0.3679)^3}{0.2526z^2} \cdot \frac{(1-\alpha)}{z-\alpha}$$

where $\alpha = \exp(-\Delta t/\lambda)$

Example 2 (Third-Order) IMC Design

For $\lambda = 1$,

$\alpha = 0.135$



Third-Order, Inverse Response (Ex. 3, Z&M, 1985)

$$\tilde{g}_p(s) = \frac{3.333(-s+1.5)}{(s+1)(s+2)(s+2.5)} \xRightarrow{\Delta t = 0.1} \tilde{g}_p(z) = \frac{-0.01316(z-1.162)(z+0.792)}{(z-0.905)(z-0.819)(z-0.779)}$$

zeros at 1.162 (outside unit circle)
-0.792 (inside, but negative)

$$\tilde{g}_p(z) = \frac{(-0.01316)(-0.162)(1.792)\boxed{\left(z-\frac{1}{1.162}\right)}z}{\left(1-\frac{1}{1.162}\right)(z-0.905)(z-0.819)(z-0.779)} \cdot \frac{(z-1.162)(z+0.792)\left(1-\frac{1}{1.162}\right)}{\boxed{\left(z-\frac{1}{1.162}\right)}z(-0.162)(1.792)}$$

$$\tilde{g}_{p-}(z) = \frac{0.02741(z-0.8606)z}{(z-0.905)(z-0.819)(z-0.779)}$$

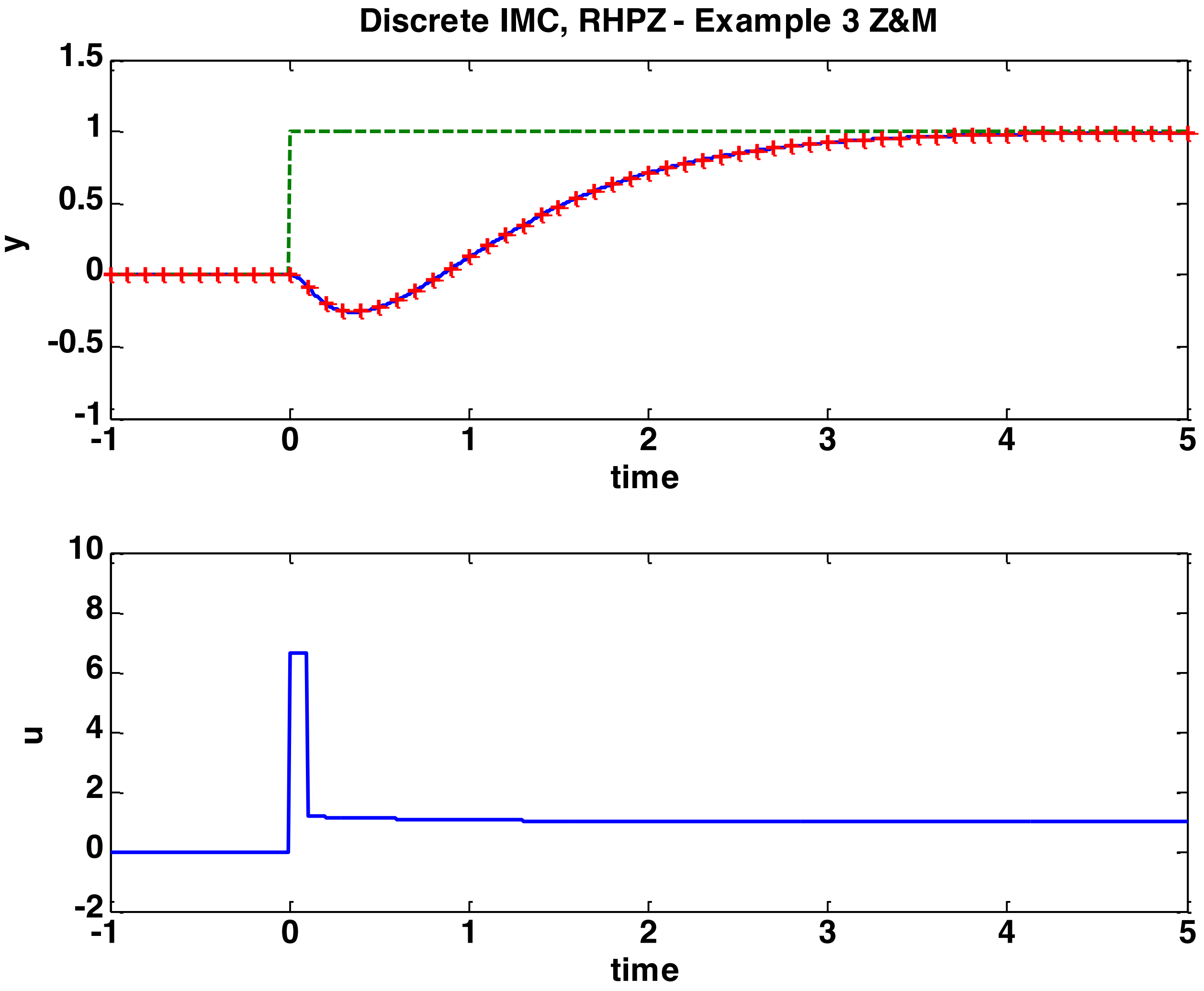
$$q(z) = \tilde{g}_{p-}^{-1}(z)F(z) = \frac{(z-0.905)(z-0.819)(z-0.779)}{0.02741(z-0.8606)z} \cdot \frac{(1-\alpha)}{z-\alpha}$$

where $\alpha = \exp(-\Delta t/\lambda)$

Response (Ex. 3, Z&M ’85)

For λ = 0.5,

α= 0.8187



“Real-World” Discussion

- Have assumed a “perfect model” for these simulations. In practice, the real-world “plant” is not perfectly modeled (indeed, there are usually nonlinearities involved).

- Can approximate real-world challenges by having the “plant” be different than the model used for controller design. Also, it is important to incorporate constraints and noise in the simulations.

Summary

- Digital Control Techniques
  - Deadbeat
  - Dahlin’s
  - State Deadbeat, State Deadbeat w/filter (Vogel-Edgar)
  - Modified Dahlin’s (mis-developed by Dahlin!)
- Internal Model Control
  - Factorization of zeros outside unit circle and negative zeros inside the circle
  - Form “all-pass”(reflection of positive zeros outside the unit circle)
  - Invert “good stuff” and use first-order filter

Suggestion

- Work through Workshop on Discrete IMC