Lectures 9 & 10

Compensator Design via Discrete Equivalent and Direct Design

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6095/4121
Dynamic Modeling and Control of Mechatronic Systems

Compensator Design via Discrete Equivalent and Direct Design Methods

1. Stability Analysis of Discrete-time Systems
   - Jury test, Stability with respect to design parameter(s), Examples
2. H(z) Design via Discrete Equivalent to H(s)
   - Different forms for discrete integration $1/s \rightarrow F(z)$ and different H(s) equivalents
   - Tustin equivalent and Tustin equivalent with prewarping
3. Example of Discrete Equivalent Design
   - H(s) design to meet specs and Discrete equivalent computations
   - Evaluation of CL discrete system
4. Root Locus Design of H(z)
   - Example of design approach, Evaluation, redesign
5. W-Plane Design of H(z)
   - $z \rightarrow w$ and $w \rightarrow z$ mappings
   - Example of design approach, Time and frequency domain evaluation
6. PID, IMC and Pole placement Controllers with Examples
7. Time Delay Systems
   - Smith predictor with Example
8. Implementation of High-Order Compensators

Stability of Discrete Systems

- We need a technique to ascertain stability of the closed-loop system, i.e., whether roots of the CL characteristic polynomial p(z) all lie within the unit circle.

$$p(z) = \begin{cases} \text{denominator of} \quad T(z) = \dfrac{\tilde{G}(z)H(z)}{1 + \tilde{G}(z)H(z)} \\ \left| zI - \Phi + \Gamma K \right| \end{cases}$$

$$p(z) = a_0 z^n + a_1 z^{n-1} + \cdots + a_n \quad (\text{generally } a_0 = 1)$$

- The technique must be simple and involve $\{a_i\}$ only.

  Applicable to <u>any</u> polynomial in z.

- Continuous-time systems analysis has Routh-Hurwitz to determine whether a polynomial p(s) has its roots in LHP. $p(s) = a_0 s^n + a_1 s^{n-1} + \cdots + a_n$
- A way to use Routh-Hurwitz test:
  (1) Map unit circle into left half-plane by replacing z with some suitable function. ($z \to e^{sh}$ will not work here since resulting p(s) will not be a polynomial.)
  (2) One possibility:

$$z = \frac{1 + wh/2}{1 - wh/2}$$

  (3) Substitute for z in p(z), multiply through by $(1 - wh/2)^n$ to obtain $\tilde{p}(w)$ = n-th order polynomial in w.
  (4) Apply Routh-Hurwitz test to $\tilde{p}(w)$.

Messy!

Jury/Raible Test for $p(z)=a_0z^n+a_1z^{n-1}+\cdots+a_n$

Set up Jury array -

	Row							
(n)	1:	a_0	a_1	a_2	$\cdots$	a_{n-1}	a_n	let $r_n = \dfrac{a_n}{a_0}$
	2:	a_n	a_{n-1}	a_{n-2}	$\cdots$	a_1	a_0	
(n-1)	3:	$a_0^{(n-1)}$	$a_1^{(n-1)}$	$a_2^{(n-1)}$	$\cdots$	$a_{n-1}^{(n-1)}$		let $r_{n-1} = \dfrac{a_{n-1}^{(n-1)}}{a_0^{(n-1)}}$
	4:	$a_{n-1}^{(n-1)}$	$a_{n-2}^{(n-1)}$		$\cdots$	$a_0^{(n-1)}$		
	$\vdots$							
(0)	2n+1:	$a_0^{(0)}$						

where
$$r_k = \frac{a_k^{(k)}}{a_0^{(k)}} \qquad k = n, n-1, \cdots, 1$$

$$a_i^{(k-1)} = a_i^{(k)} - r_k a_{k-i}^{(k)} \qquad i = 0, 1, \cdots, k-1 \quad \text{where initially } a_i^{(n)} = a_i$$

In "English" -

- Each odd row = previous odd row – r_k * previous even row.
- Each even row = preceding odd row in reverse order.
- First row has coefficients of p(z).
- Last row has 1 element.

Criteria:

(1) If $a_0 > 0$, then all roots of p(z) lie in unit circle if and only if $a_0^{(k)} > 0$, $k = n-1, n-2, \ldots, 0$.

(2) The no. of negative $a_0^{(k)}$ = no. of roots of p(z) outside unit circle.

Applications of Jury Test

- Test if first entry in each odd row > 0.
- If obtain any $a_0^{(k)} \leq 0$, stop; $p(z)$ has root(s) $|\lambda| \geq 1$.
- Simple computer program, need 2 scratch vectors.

Example 1 : $p(z) = z^2 - z + 0.5$

(2)	1.0 0.5	–1 –1	0.5 1.0	r = 0.5
(1)	1–0.25 = 0.75 –0.5	–1+0.5 = –0.5 0.75		r = –0.5/0.75 = –0.67
(0)	0.75–0.33 = 0.42			

All $a_0^{(k)} > 0$ => system is stable (all roots in unit ⊙).

Applications of Jury Test (Cont'd)

Example 2 : $p(z) = z^2 - z + 2$

(2)	1 2	– 1 – 1	2 1	r = 2
(1)	1– 4 = –3	– 1+2 = 1		$a_0^{(1)} < 0$ ==> system is unstable.
	1	–3		r = – 1/3
(0)	–3–(–1/3) = –8/3			=> 2 roots outside unit ⊙ .

Example 3 : $p(z) = z^3 - 0.15z^2 - 0.59$

(3)	1.00 -0.59	– 0.15 0.00	0.00 – 0.15	– 0.59 1.00	r = – 0.59
(2)	0.65 -0.09	– 0.15 – 0.15	– 0.09 0.65		r = – 0.14
(1)	0.64 -0.13	– 0.13 0.64			r = – 0.20
(0)	0.61				

All $a_0^{(k)} > 0$ => system is stable.

Application to SVFB Example

The equivalent discrete system
$$\underline{x}(k+1)=\underbrace{\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}}_{\Phi}\underline{x}(k)+\underbrace{\begin{bmatrix}0\\ 1\end{bmatrix}}_{\Gamma}u(k)$$

is to be controlled using the algorithm, $u(k) = r(k) - [\underbrace{1 \quad 3}_{K}]\,\underline{x}(k)$

Check if closed-loop system is stable.

- Closed-loop system matrix $\bar{\Phi}=\Phi-\Gamma K$

$$\bar{\Phi}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}-\begin{bmatrix}0\\ 1\end{bmatrix}\begin{bmatrix}1 & 3\end{bmatrix}=\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}-\begin{bmatrix}0 & 0\\ 1 & 3\end{bmatrix}=\begin{bmatrix}1 & 1\\ -1 & -2\end{bmatrix}$$

- Closed-loop characteristic polynomial

$$p(z)=\left|zI-\bar{\Phi}\right|=\begin{vmatrix}z-1 & -1\\ 1 & z+2\end{vmatrix}=z^2+z-1$$

- Jury array

(2)	1	1	–1	r = –1
	–1	1	1	
(1)	0	STOP		

- CL system is unstable, but roots are <u>not</u> on unit circle.
  Roots of p(z) are $z_1 = 0.618$, $z_2 = -1.618$, so $a_0^{(k)} = 0$ does not necessarily imply roots on unit circle. (Note $|z_1 z_2| = 1$ here, corresponding to roots λ and $1/\lambda$.)
- If some $a_0^{(k)} = 0$, can replace $0 \rightarrow +\varepsilon$ and continue further, e.g. as in Routh-Hurwitz test.

Stability with Respect to a Parameter

If system (or controller) has a free parameter, β, wish to determine range of values for which system is stable.

Example 1 -

The system $G(s) = a/(s+a)$, $a = 1$, is to be controlled using series compensation with algorithm $u(k) = Ke(k) + u(k-1)$ and time step $h = 0.69$ sec. For what range of K is CL system stable?

$$\tilde{G}(z) = \left.\frac{1-e^{-ah}}{z-e^{-ah}}\right|_{ah=0.69} = \frac{0.5}{z-0.5};\quad u(z) = Ke(z) + z^{-1}u(z) \Rightarrow \frac{u(z)}{e(z)} = H(z) = \frac{K}{1-z^{-1}} = \frac{Kz}{z-1}$$

$$1+\tilde{G}(z)H(z) = \frac{Kz/2}{(z-1/2)(z-1)} + 1$$

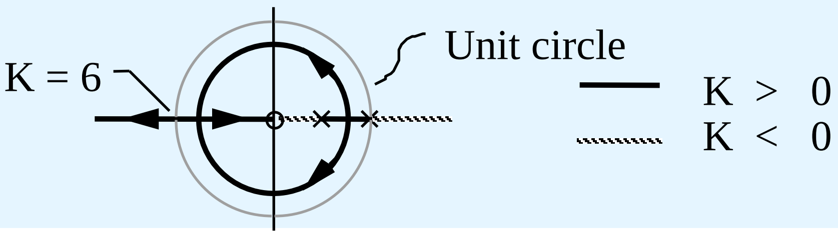
$$p(z) = (z-1/2)(z-1) + Kz/2 = z^2 + \left[(K-3)/2\right]z + 1/2$$

(2)	1	(K–3)/2	1/2	r = 1/2
	1/2	(K–3)/2	1	
(1)	3/4	(K–3)/4		r = (K–3)/3
	(K–3)/4	3/4		
(0)	3/4–(K–3)^2/12			

Jury criterion
=> $3/4 > (K-3)^2/12$
=> $(K-3)^2 < 9$
=> $-3 < K-3 < 3$
=> $0 < K < 6$

- Reconcile with root locus:

$$1+\frac{K}{2}\frac{z}{(z-1/2)(z-1)} = 1+\tilde{G}(z)H(z)$$



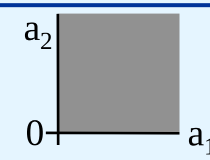
Stability with Respect to Multiple Parameters

Can determine constraints that must be satisfied among a set of parameters.

Example 2 -

Determine region in the $a_1 - a_2$ plane for which $p(z) = z^2 + a_1 z + a_2$ has its roots in the unit circle.

Recall stability conditions for $p(s) = s^2 + a_1 s + a_2$ to have roots in LHP is $a_1, a_2 > 0$.



Jury array:

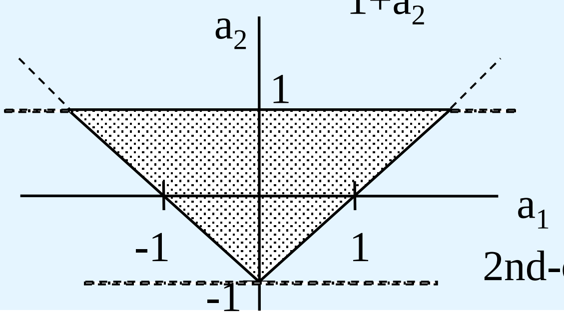
(2)	1	a_1	a_2	$r = a_2$
	a_2	a_1	1	
(1)	$1-a_2^2$	$a_1(1-a_2)$		$r = \dfrac{a_1}{1+a_2}$
	$a_1(1-a_2)$	$1-a_2^2$		
(0)	$1-a_2^2 - \dfrac{a_1^2(1-a_2)}{1+a_2}$			

Jury criteria: $1 - a_2^2 > 0 \Rightarrow \underline{-1 < a_2 < 1}$

$$1 - a_2^2 - \frac{a_1^2(1-a_2)}{1+a_2} > 0 \Rightarrow (1+a_2)^2 - a_1^2 > 0$$

[since $1-a_2 > 0$ and $1+a_2 > 0$]

$$\Rightarrow \underline{-(1+a_2) < a_1 < 1+a_2}$$



2nd-order p(z) stability region

A More Complicated, State –Space Example

The open-loop unstable continuous system defined by

$$\dot{\underline{x}}(t) = \begin{bmatrix} 0 & 1 & -1 \\ 3 & -2 & 1 \\ 0 & 2 & -1 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} u(t); \quad y(t) = \begin{bmatrix} 1 & 0 & 2 \end{bmatrix} \underline{x}(t)$$

is to be controlled using a digital computer with h = 0.05.

Investigate CL stability using the SVFB algorithm

$$u(k) = r(k) - 0.5\ x_1(k) - 2\ x_2(k) - x_3(k)$$

$$= r(k) - \underbrace{\begin{bmatrix} 0.5 & 2 & 1 \end{bmatrix}}_{K} \underline{x}(k) \qquad (K_r = 1)$$

(1) Obtain equivalent discrete system $\underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma u(k)$ using c2d,

$$\Phi = \begin{bmatrix} 1.0035 & 0.0453 & -0.0477 \\ 0.1430 & 0.9105 & 0.0429 \\ 0.0071 & 0.0930 & 0.9535 \end{bmatrix}; \quad \Gamma = \begin{bmatrix} 0.0512 \\ 0.0513 \\ 0.0025 \end{bmatrix}$$

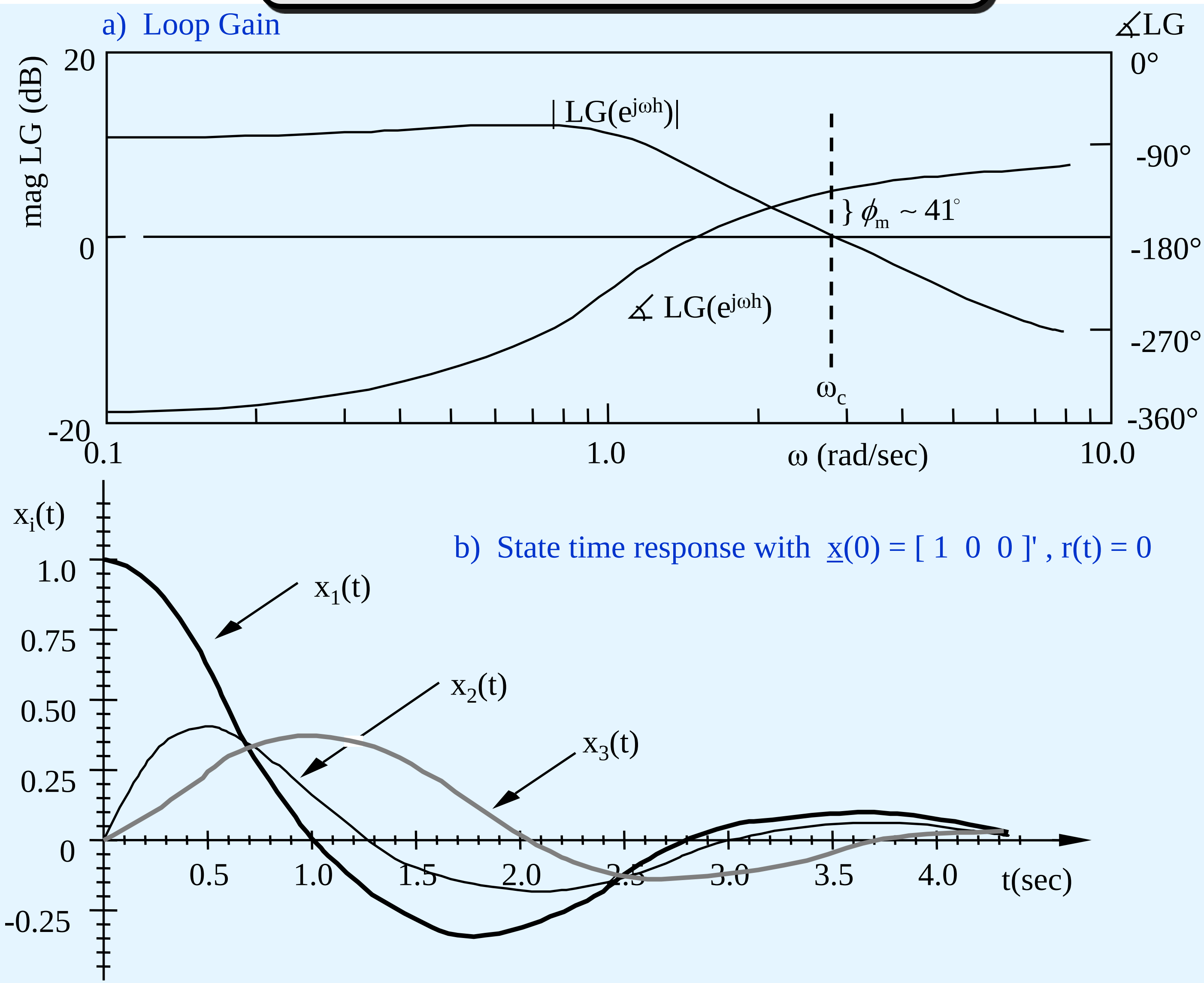
(2) Form CL system matrix, $\bar{\Phi} = \Phi - \Gamma K$, then use ss2tf to obtain CL transfer function $T(z) = C(zI - \bar{\Phi})^{-1}\Gamma$. Need only to obtain $p(z) = | zI - \bar{\Phi} |$ for closed-loop stability test.

$$p(z) = z^3 - 2.737z^2 + 2.497z - 0.758$$

(3) Apply Jury test $\rightarrow$ p(z) has all roots in $\odot$ ==> CL stable

(4) Phase margin can be evaluated by using ss2tf to obtain $K(zI - \Phi)^{-1}\Gamma$, then using Bode (option 2) to plot $LG(z)\big|_{z = e^{j\omega h}}$. ==> Obtain $\omega_c \approx 2.8$ rad/sec, $\phi_m \approx 41^o$

State-Space Example Plots



Fundamentals of Digital Compensator Design

"Given a G(s), or $\tilde{G}(z)$, design a series compensator H(z) so that the closed-loop system meets specs."

Design Approaches

- H(z) design via discrete equivalent
  - Idea is to use continuous time design methods to construct H(s) given G(s), then obtain from H(s) a suitable discrete compensator $\tilde{H}(z)$.
  - Scheme might be expected to be useful provided,

$$\tilde{G}(z)\Big|_{z = e^{j\omega h}} \approx G(j\omega) \quad \Rightarrow \quad h \sim \text{small}$$

  - Alternately, an analog H(s) compensator often exists and we desire to replace the "older" analog system with a digital, µ-processor controller.

<u>Problem</u>: Given H(s) how do we obtain an $\tilde{H}(z)$?

- Direct design of H(z) given $\tilde{G}(z)$.

Evaluation Tools:

- stability tests
- loop gain analysis
- root locus
- simulation

$\vdots$

H(z) Design via Discrete Equivalent: $H(s) \rightarrow \tilde{H}(z)$

Goals:

- Simplicity

  Hold equivalence methods [viz $G(s) \rightarrow \tilde{G}(z)$], and impulse transformation methods [$Z\{L^{-1}\{H(s)\}\}$] are not simple.

- $\tilde{H}(z)$ = rational transfer function

  $\tilde{H}(z) = A(z) / B(z)$ A(z), B(z) = polynomials

  [Thus the "obvious" inverse relation $s = \frac{1}{h}\log(z)$ is NG.]

- If H(s) = m-th order transfer function then $\tilde{H}(z)$ = m-th order transfer function.

  Typically, $H(s) = \dfrac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^m + a_1 s^{m-1} + \cdots + a_m}$ $b_0 \neq 0$

  i.e., H(s) will invariably contain a pure gain, (and state-variable model of H(s) will have $d \neq 0$). Require

  $$\tilde{H}(z) = \frac{\beta_0 z^m + \beta_1 z^{m-1} + \cdots + \beta_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m} \qquad \beta_0 \neq 0$$

- Accuracy

  Desire $\tilde{H}(z)\big|_{z=e^{j\omega h}} \approx H(j\omega)$ over the frequency range of interest/importance.

<u>Idea</u>: Replace s with some suitable rational F(z).

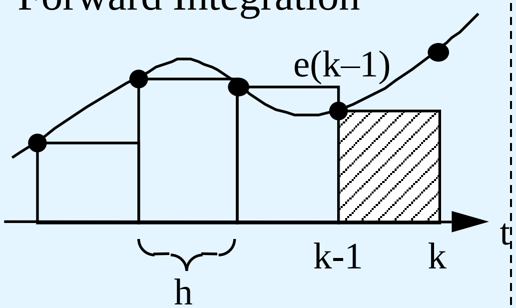
- A given H(s) can be synthesized as an interconnection of integrators = 1/s elements (recall elementary signal flow diagram) => replace 1/s = continuous time integrator by F(z) = transfer function of a <u>discrete</u> integrator.

Forms of Discrete Integration

$$e(k) \longrightarrow \boxed{F(z)} \longrightarrow g(k) \qquad F(z) = \frac{g(z)}{e(z)}$$

$$g(k-1) = \text{approximate value of } \int_{-\infty}^{(k-1)h} e(t)\,dt \;; \qquad g(k) = \text{approximate value of } \int_{-\infty}^{kh} e(t)\,dt$$

1. Forward Integration



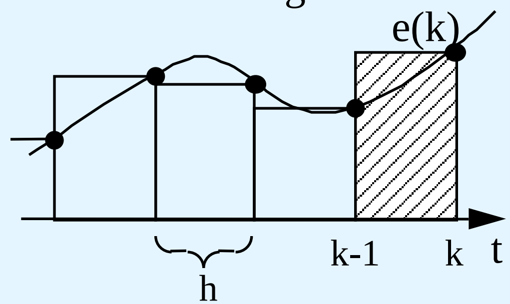
$$g(k) = g(k-1) + he(k-1)$$

$$g(z) = z^{-1}g(z) + z^{-1}he(z)$$

$$\Rightarrow\; F(z) = \frac{h}{z-1} \sim \frac{1}{s}$$

$$\text{Replacement } s \rightarrow \frac{z-1}{h}$$

2. Backward Integration



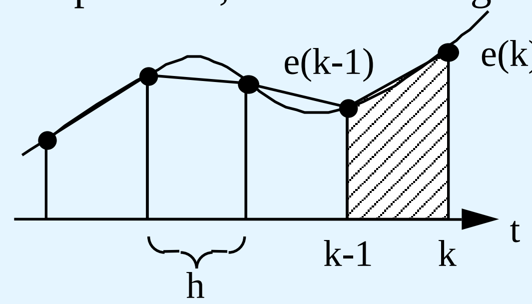
$$g(k) = g(k-1) + he(k)$$

$$g(z) = z^{-1}g(z) + he(z)$$

$$\Rightarrow\; F(z) = \frac{h}{1-z^{-1}} = \frac{zh}{z-1} \sim \frac{1}{s}$$

$$\text{Replacement } s \rightarrow \frac{z-1}{zh}$$

3. Trapezoidal, or Tustin Integration



$$g(k) = g(k-1) + h/2\,[e(k) + e(k-1)]$$

$$g(z) = z^{-1}g(z) + h/2\,(1+z^{-1})\,e(z)$$

$$\Rightarrow\; F(z) = \frac{h}{2}\left(\frac{1+z^{-1}}{1-z^{-1}}\right) = \frac{h}{2}\left(\frac{z+1}{z-1}\right) \sim \frac{1}{s}$$

$$\text{Replacement } s \rightarrow \frac{2}{h}\left(\frac{z-1}{z+1}\right)$$

Relationship to True s→z Map

Each method corresponds to a different rational approximation of e^{sh}

(1) Forward integration:

$$z = e^{sh} \doteq 1 + sh \qquad \text{gives } s = \frac{z-1}{h}$$

(2) Backward integration:

$$z = \frac{1}{e^{-sh}} \doteq \frac{1}{1-sh} \qquad \text{gives } s = \frac{z-1}{zh}$$

(3) Tustin integration:

$$z = \frac{e^{sh/2}}{e^{-sh/2}} \doteq \frac{1+sh/2}{1-sh/2} \qquad \text{gives } s = \frac{2}{h}\frac{z-1}{z+1}$$

Note:

- The above replacements maintain transfer function order

$$\text{if } H(s) = \frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^m + a_1 s^{m-1} + \cdots + a_m} \rightarrow \tilde{H}(z) = \frac{b_0 (z-1)^m + \cdots}{(z-1)^m + \cdots}$$

- Forward integration ◄──► Euler method to predict g(k)

$$\overbrace{\frac{g(k)-g(k-1)}{h}}^{\dot{g}(t) = e(t)} \Rightarrow g(k) = g(k-1) + he(k-1)$$

- Even if $H(s) = \frac{\text{r-th order}}{\text{m-th order}}$, $\tilde{H}(z) = \frac{\text{m-th order}}{\text{m-th order}}$ for (2) and (3)

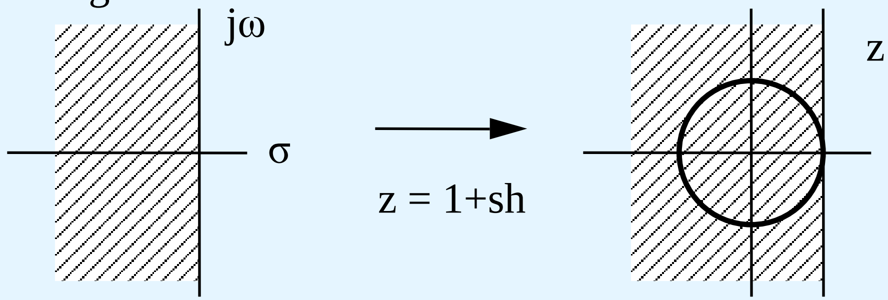
  [OK since H(s) is almost always m-th order/m-th order].

- Tustin ~ 1st order Pade approximation to z^{-1}

Mapping of LHP to Unit Circle

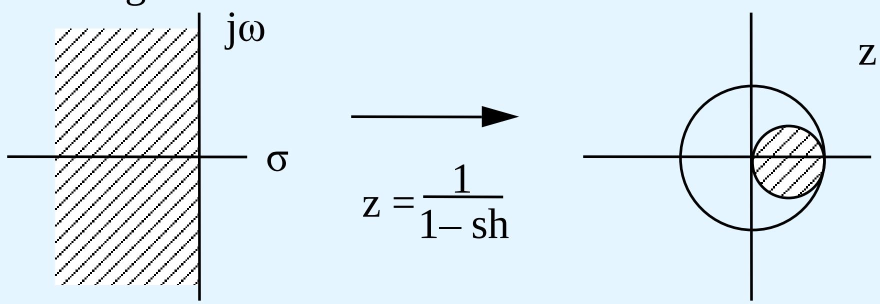
- Useful as a criterion for selecting integration scheme:

(1) Forward integration



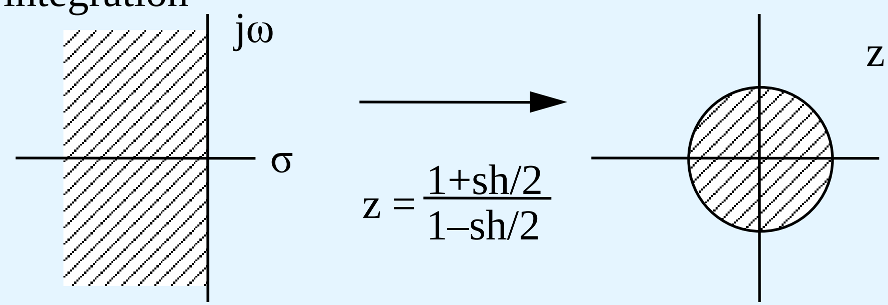
A stable H(s) can yield an unstable $\tilde{H}(z)$! NOT GOOD

(2) Backward integration



Stable H(s) yields stable $\tilde{H}(z)$; some unstable H(s) can yield stable $\tilde{H}(z)$.

(3) Tustin integration



Preferable map since stability areas are mapped 1:1.

Computing $\tilde{H}(z)$ via Tustin Equivalent

- Since any H(s) can be decomposed (via PF expansion) into either a cascade or a sum of first and second-order terms, equivalence can be done on a term-by-term basis.

(1) Simple Lag, $H(s) = K\frac{1}{\tau s + 1} \quad \left(\text{or } K\frac{a_1}{s+a_1} \text{ with } a_1 = \tau^{-1} \right)$

$$\tilde{H}(z) = K\left[\frac{1}{\frac{2\tau}{h}\left(\frac{z-1}{z+1}\right) + 1} \right] = \underbrace{\frac{Kh/2\tau}{1 + h/2\tau}}_{\tilde{K}} \left[\frac{z+1}{z - \underbrace{\frac{1-h/2\tau}{1+h/2\tau}}_{\alpha_1 \sim e^{-h/\tau}}} \right]$$

(2) General First-order factor

$$H(s) = K\frac{b_0 s + b_1}{s + a_1} \quad \rightarrow \quad \tilde{H}(z) = \tilde{K}\frac{z-\beta_1}{z-\alpha_1}$$

$$\frac{b_1}{b_0} < a_1 \quad \Rightarrow \quad \text{lead}; \qquad \frac{b_1}{b_0} > a_1 \quad \Rightarrow \quad \text{lag}$$

$$\beta_1 = \frac{b_0 - b_1 h/2}{b_0 + b_1 h/2}, \quad \alpha_1 = \frac{1 - a_1 h/2}{1 + a_1 h/2}, \quad \tilde{K} = K\frac{b_0 + b_1 h/2}{1 + a_1 h/2}$$

Computing H(z) via Tustin Equivalent (Cont'd)

(3) General Second-order factor

$$H(s) = K\frac{b_0 s^2 + b_1 s + b_2}{s^2 + a_1 s + a_2} \rightarrow \tilde{H}(z) = \tilde{K}\frac{z^2 - \beta_1 z + \beta_2}{z^2 - \alpha_1 z + \alpha_2}$$

$$\alpha_2 = \frac{1 - a_1 h/2 + a_2 h^2/4}{1 + a_1 h/2 + a_2 h^2/4}, \qquad \alpha_1 = \frac{2 - a_2 h^2/2}{1 + a_1 h/2 + a_2 h^2/4}$$

$$\beta_2 = \frac{b_0 - b_1 h/2 + b_2 h^2/4}{b_0 + b_1 h/2 + b_2 h^2/4}, \qquad \beta_1 = \frac{2b_0 - b_2 h^2/2}{b_0 + b_1 h/2 + b_2 h^2/4}$$

$$\tilde{K} = K\frac{b_0 + b_1 h/2 + b_2 h^2/4}{1 + a_1 h/2 + a_2 h^2/4}$$

General Algorithm for Tustin Transformation

$$H(s) = K\frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^m + a_1 s^{m-1} + \cdots + a_m} = \frac{u(s)}{e(s)}$$

Augmented System: $\begin{bmatrix} \underline{x}(k) \\ e(k) \end{bmatrix}$

$\begin{bmatrix} \tilde{A}_a & \tilde{B}_a \\ \tilde{C}_a & 0 \end{bmatrix}; \tilde{A}_a = \begin{bmatrix} \tilde{A} & \tilde{B} \\ 0 & 0 \end{bmatrix};$

$\tilde{B}_a = \begin{bmatrix} \tilde{B} \\ 1 \end{bmatrix}; \tilde{C}_a = \begin{bmatrix} C & d \end{bmatrix}$

Multiply numerator by z

(1) Write a state variable model for H(s) in SOF with K = 1.

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + Be(t)\,; \quad u(t) = C\,\underline{x}(t) + de(t)$$

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & 1 \\ -a_m & 0 & \cdots & & 0 \end{bmatrix}; \quad B = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_m \end{bmatrix}; \quad \tilde{b}_i = b_i - a_i b_0; \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}; \quad d = b_0$$

(2) Take L ==>$s\underline{x}(s) = A\underline{x}(s) + Be(s)$ and replace $s \to \frac{2}{h}\left(\frac{z-1}{z+1}\right)$. $\Rightarrow \frac{2}{h}\left(\frac{z-1}{z+1}\right)\underline{x}(z) = A\underline{x}(z) + Be(z)$

(3) Solve above for $\underline{x}(z)$ and form: $u(z) = C\underline{x}(z) + de(z)$

$$u(z) = \underbrace{\left\{C\left(zI - \tilde{A}\right)^{-1}\tilde{B}(z+1) + d\right\}}_{\tilde{H}(z)} e(z); \qquad \tilde{A} = \left(I - (h/2)A\right)^{-1}\left(I + (h/2)A\right)$$

$$\tilde{B} = \left(I - (h/2)A\right)^{-1} B\, h/2$$

(4) Use ss2tf to obtain coefficients $\bar{a}_i, \bar{b}_i$, of denominator and numerator of $C\left(zI - \tilde{A}\right)^{-1}\tilde{B}$

(5) Form final:

$$\tilde{H}(z) = K\frac{\beta_0 z^m + \beta_1 z^{m-1} + \cdots + \beta_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m}$$

where $\beta_i = \bar{b}_i + \bar{b}_{i+1} + d\bar{a}_i \qquad i = 0, 1, 2, \cdots, m-1$

$\beta_m = \bar{b}_m + d\bar{a}_m; \quad \alpha_i = \bar{a}_i; i = 1, 2, .., m$

General Algorithm for Forward Integration

$$H(s) = K\frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^m + a_1 s^{m-1} + \cdots + a_m} = \frac{u(s)}{e(s)}$$

(1) Write a state variable model for H(s) in SOF with K = 1.

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + Be(t)\,; \quad u(t) = C\,\underline{x}(t) + de(t)$$

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & 1 \\ -a_m & 0 & \cdots & & 0 \end{bmatrix}; \quad B = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_m \end{bmatrix}; \quad \tilde{b}_i = b_i - a_i b_0; \quad C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}; \quad d = b_0$$

(2) Take L ==>$s\underline{x}(s) = A\underline{x}(s) + Be(s)$ and replace $s \rightarrow \left(\frac{z-1}{h}\right)$. $\left(\frac{z-1}{h}\right)\underline{x}(z) = A\underline{x}(z) + Be(z)$

(3) Solve above for $\underline{x}(z)$ and form: $u(z) = C\underline{x}(z) + de(z)$

$$u(z) = \underbrace{\left\{C\left(zI - \tilde{A}\right)^{-1}\tilde{B} + d\right\}}_{\tilde{H}(z)} e(z); \qquad \tilde{A} = (I + hA)$$

$$\tilde{B} = B\,h$$

(4) Use Leverier algorithm to obtain coefficients $\bar{a}_i$, $\bar{b}_i$, of denominator and numerator of $C\left(zI - \tilde{A}\right)^{-1}\tilde{B}$

(5) Form final:

$$\tilde{H}(z) = K\frac{\beta_0 z^m + \beta_1 z^{m-1} + \cdots + \beta_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m}$$

where $\beta_i = \bar{b}_i + d\bar{a}_i; i = 0, 1, 2, \cdots, m$

$\alpha_i = \bar{a}_i; i = 1, 2, .., m$

General Algorithm for Backward Integration

$$H(s) = K\frac{b_0 s^m + b_1 s^{m-1} + \cdots + b_m}{s^m + a_1 s^{m-1} + \cdots + a_m} = \frac{u(s)}{e(s)}$$

(1) Write a state variable model for H(s) in SOF with K = 1.

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + Be(t)\,; \quad u(t) = C\,\underline{x}(t) + de(t)$$

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & 1 & \cdots & 0 \\ \vdots & & & \ddots & 1 \\ -a_m & 0 & \cdots & & 0 \end{bmatrix}; \quad B = \begin{bmatrix} \tilde{b}_1 \\ \tilde{b}_2 \\ \vdots \\ \tilde{b}_m \end{bmatrix}; \quad \tilde{b}_i = b_i - a_i b_0; \quad C = [1 \quad 0 \quad \cdots \quad 0]; \quad d = b_0$$

(2) Take L ==>s$\underline{x}$(s) = A$\underline{x}$(s) + Be(s) and replace $s \to \frac{1}{h}\left(\frac{z-1}{z}\right)$. $\frac{1}{h}\left(\frac{z-1}{z}\right)\underline{x}(z) = A\underline{x}(z) + Be(z)$

(3) Solve above for $\underline{x}$(z) and form: u(z) = C$\underline{x}$(z) + de(z)

$$\underbrace{u(z) = \left\{C\left(zI - \tilde{A}\right)^{-1}\tilde{B}z + d\right\}}_{\tilde{H}(z)} e(z); \qquad \tilde{A} = (I - hA)^{-1}$$

$$\tilde{B} = (I - hA)^{-1} B\,h$$

(4) Use Leverier algorithm to obtain coefficients $\bar{a}_i$, $\bar{b}_i$, of denominator and numerator of $C\left(zI - \tilde{A}\right)^{-1}\tilde{B}$

(5) Form final:

$$\tilde{H}(z) = K\frac{\beta_0 z^m + \beta_1 z^{m-1} + \cdots + \beta_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m}$$

where $\beta_i = \bar{b}_{i+1} + d\bar{a}_i \qquad i = 0, 1, 2, \cdots, m-1$

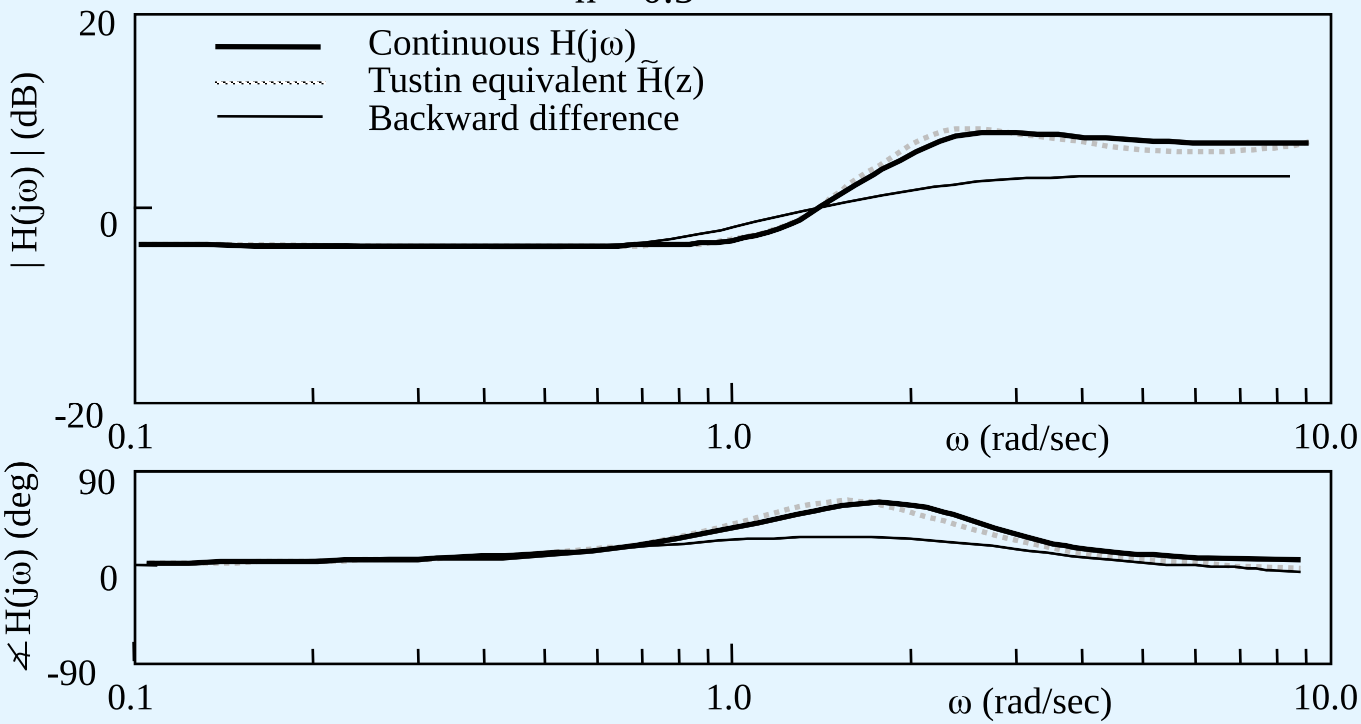
$\beta_m = d\bar{a}_m; \quad \alpha_i = \bar{a}_i; i = 1, 2, .., m$

Bode Plot Comparisons

Usually $\tilde{H}(z)\big|_{z=e^{j\omega h}} \equiv H(s)\big|_{s=\frac{2}{h}\left(\frac{e^{j\omega h}-1}{e^{j\omega h}+1}\right)} \approx H(j\omega)$ for Tustin equivalence.

$\left(\text{Include option 3 in Bode plot program } x = \frac{z-1}{zh}, \text{ and option 4, } x = \frac{2}{h}\left(\frac{z-1}{z+1}\right) \text{ where } z = e^{j\omega h}\right)$

Example 1: $H(s) = \frac{2s^2+3s+4}{s^2+2s+6} \xrightarrow[h=0.5]{\text{Tustin}} \tilde{H}(z) = \frac{1.6z^2 - 1.867z + 0.8}{z^2 - 0.667z + 0.467}$



Tustin equivalence is usually superior to backward difference equivalent when comparing $\tilde{H}(z)\big|_{z=e^{j\omega h}}$ to $H(j\omega)$.

Tustin Equivalence with Frequency Prewarping

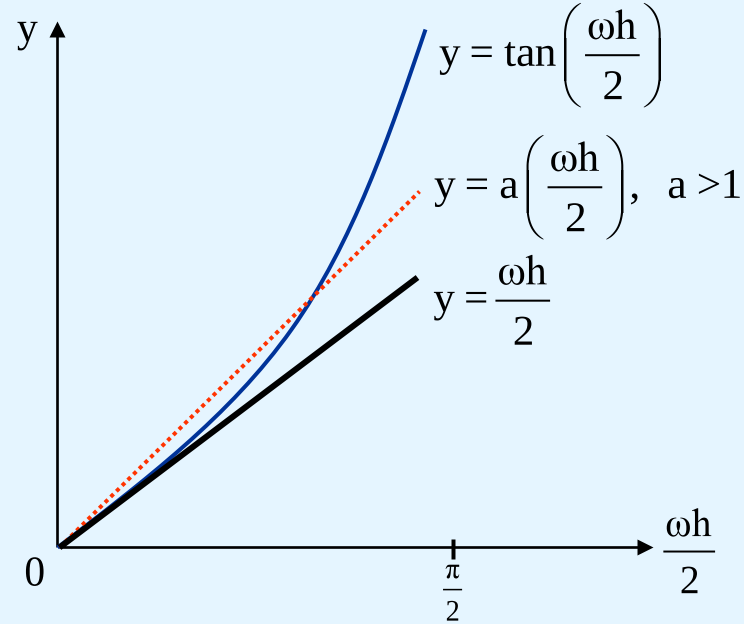
- Is it possible to improve the match between Tustin $\tilde{H}(z)$ at $z = e^{j\omega h}$ and original $H(j\omega)$?
- At which frequencies, ω, does equality hold?

Tustin $\quad H(z)\big|_{z=e^{j\omega h}} = H(s)\big|_{s=j\omega}$

if and only if $\quad \frac{2}{h}\left(\frac{e^{j\omega h}-1}{e^{j\omega h}+1}\right) = j\omega$

or $\quad \frac{e^{j\omega h/2}-e^{-j\omega h/2}}{j\left(e^{j\omega h/2}+e^{-j\omega h/2}\right)} \equiv \tan\left(\frac{\omega h}{2}\right) = \frac{\omega h}{2}$

- For $0 \le \omega < \pi/h$ equality holds only at $\omega = 0$.



- Can obtain equality at <u>one</u> other $\omega \neq 0$ if we have

$\tan\left(\frac{\omega h}{2}\right) = a\frac{\omega h}{2}; \quad a > 1$

This corresponds to replacement $\quad s \to \frac{2}{ah}\left(\frac{z-1}{z+1}\right)$

- For equality at $\omega=\omega_1$, usually some important frequency, $\quad a = \frac{\tan(\omega_1 h/2)}{(\omega_1 h/2)}$

$\Rightarrow$ Tustin with prewarp (include as option 5 in Bode plot)

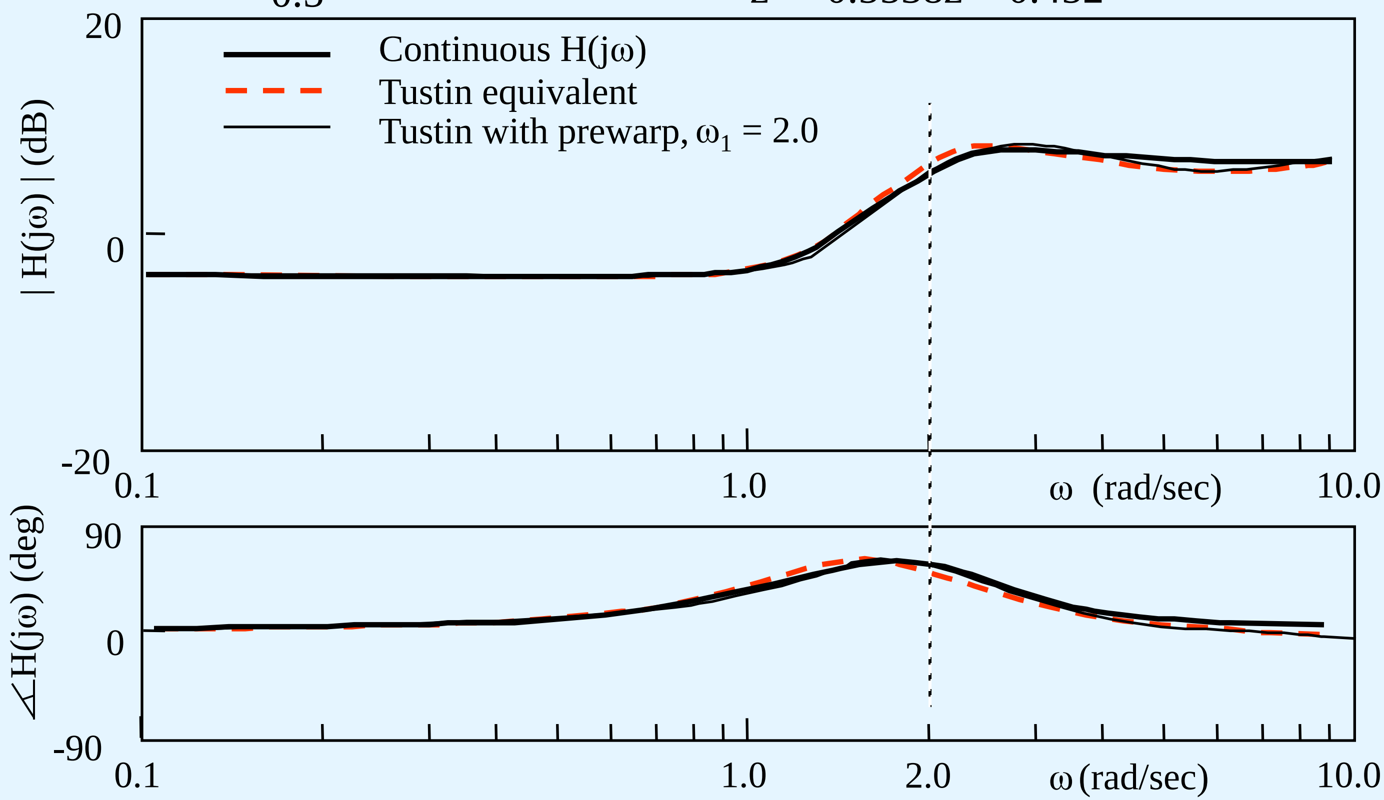
$$s \to \frac{2}{h}\frac{(\omega_1 h/2)}{\tan(\omega_1 h/2)}\cdot\left(\frac{z-1}{z+1}\right) \qquad \text{(like a "modified" } h \to ah\text{)}$$

Example 2 – Tustin Equivalence with Prewarping

$$H(s) = \frac{2s^2 + 3s + 4}{s^2 + 2s + 6}\ ; \quad h = 0.5$$

Require $\tilde{H}(z)\big|_{z=e^{j\omega h}} = H(s)\big|_{s=j\omega}$ at $\omega = 2$ (corresponds approximately to where $\measuredangle H(j\omega)$ is max).

$$a = \frac{\tan 0.5}{0.5} = 1.093\ ; \qquad \tilde{H}(z) = \frac{1.563z^2 - 1.706z + 0.742}{z^2 - 0.5538z + 0.452}$$



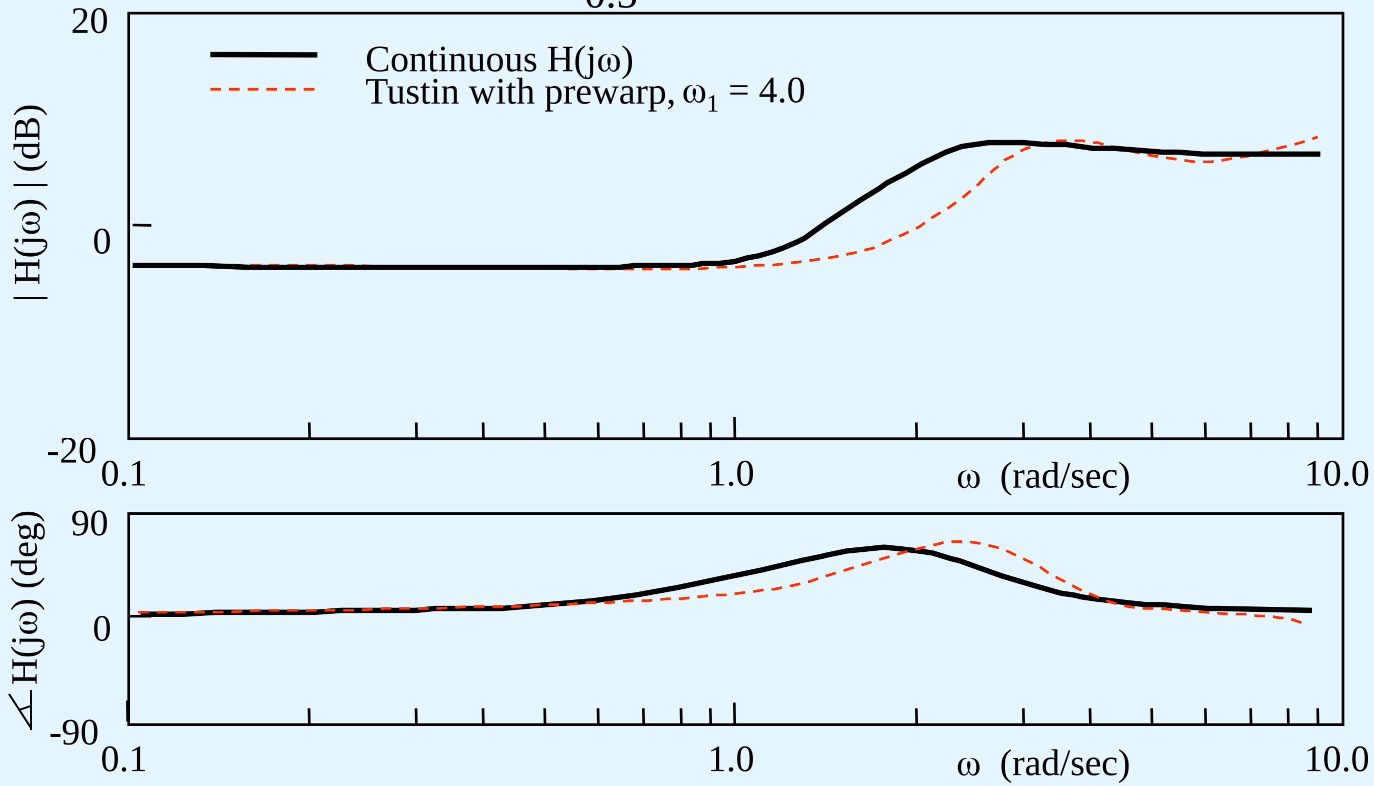
- Gives better match in region $\omega \approx [1.2, 3]$.

Example 3 – Tustin Equivalence with Prewarping

$$H(s) = \frac{2s^2 + 3s + 4}{s^2 + 2s + 6}; \quad h = 0.5$$

- A poor choice of ω_1 can result in substantial $H(j\omega)$ vs. $\tilde{H}(e^{j\omega h})$ mismatch for $\omega \neq \omega_1$.

$$\text{e.g., } \omega_1 = 4, \quad a = \frac{\tan 0.5}{0.5} = 1.558$$



=> To avoid problems keep $\omega_1 \leq 1/h < \pi/h$ and examine Bode plot comparisons of $\tilde{H}(e^{j\omega h})$ vs. $H(j\omega)$.

Other Techniques for $H(s) \rightarrow \tilde{H}(z)$ Equivalence

- Pole-zero mapping

$$H(s) = K \frac{\prod_{i=1}^{p}(s-\delta_i)}{\prod_{i=1}^{m}(s-\lambda_i)} \rightarrow \tilde{H}(z) = \tilde{K} \frac{\prod_{i=1}^{m}(z-\tilde{\delta}_i)}{\prod_{i=1}^{m}(z-\tilde{\lambda}_i)}$$

where

1. If $H(s)$ has a pole at $s = \lambda_i$, then $\tilde{H}(z)$ has a pole at $z = \tilde{\lambda}_i = e^{\lambda_i h}$.
2. If $H(s)$ has a zero at $s = \delta_i$, then $\tilde{H}(z)$ has a zero at $z = \tilde{\delta}_i = e^{\delta_i h}$.
3. Pick $\tilde{K}$ such that $H(s)|_{s=0} = \tilde{H}(z)|_{z=1}$. (use $s = \dfrac{2\pi}{1000h}$ if $H(0) = 0$)

- Zero-order hold

  Write state model (SOF) for $H(s)$, then $\tilde{H}(z) = C(zI - \Phi)^{-1}\Gamma + d$

  (Has "effective" h/2 sec delay due to hold equivalence)

- Higher-order polynomial approximations to 1/s

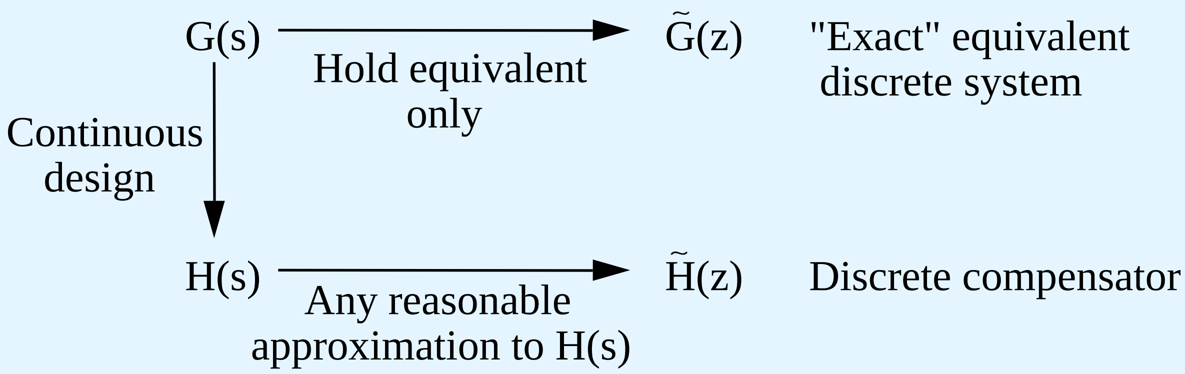
  Tustin ~ 1st order polynomial through $e(k-1)$, $e(k)$

  Simpson ~ 2nd order polynomial through $e(k-2)$, $e(k-1)$, $e(k)$

$$\frac{1}{s} \rightarrow \frac{h(z^2+4z+1)}{3(z^2-1)} \Rightarrow g(k) = g(k-2) + \frac{h}{3}[e(k) + 4e(k-1) + e(k-2)]$$

  Gives a better equivalence in $\tilde{H}(e^{j\omega h})$ vs. $H(j\omega)$ but order of $\tilde{H}(z)$ is <u>2m</u>.

Summary of Discrete Equivalence Methods



- Tustin equivalence, $s \rightarrow \frac{2}{h}\left(\frac{z-1}{z+1}\right)$, gives a good approximation with a minimum of effort.

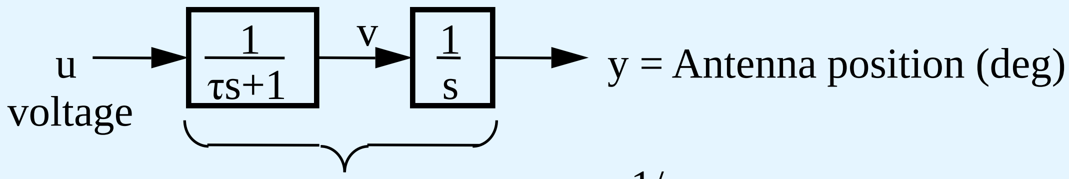
  This is the most commonly used scheme.

$$H(s)\Big|_{s=\frac{2}{h}\left(\frac{z-1}{z+1}\right)} = \tilde{H}(z)$$

- Consider use of prewarping if there is a frequency ω_1, or frequency region about ω_1, where it is important that $\tilde{H}(e^{j\omega h}) \approx H(j\omega)$; e.g., in vicinity of ω_{max} for lead network, or around crossover frequency ω_c.
- Pole-zero mapping is frequently used (very similar in results to Tustin), but does not permit frequency prewarping.
- $H(s) \rightarrow \tilde{H}(z)$ equivalent transformations are very frequently used in digital filtering and Digital filter design.

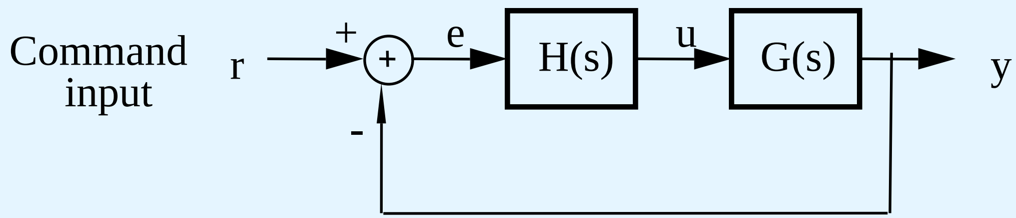
Example of Discrete Equivalent Design

- Radar positioning system (Franklin and Powell, 1980)



Drive motor, $G(s) = \dfrac{1/\tau}{s(s + 1/\tau)}$; $\tau = 10$ sec

- Closed-loop requirements



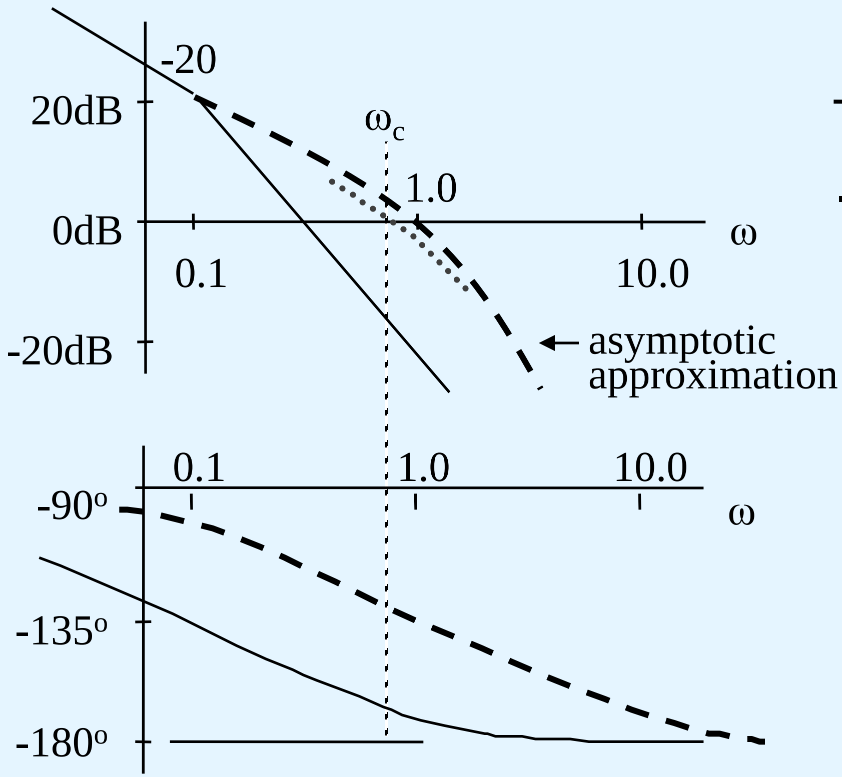
$$G(s) = \frac{0.1}{s(s + 0.1)}$$

Desire ~ 15% overshoot to a step command input (=> $\zeta \sim 0.5$) and

$t_s(1\%)$ ~ 10 sec (=> $\zeta\omega_n \sim 0.5$) with a phase margin $\phi_m \geq 50^o$.

Example of Discrete Equivalent Design (Cont'd)

- "Solution", H(s) = lead NW = $\frac{10s+1}{s+1}$ $(\omega_2 = 0.1, \beta = 10, K = 1)$



Bode plot of G(s)

Bode plot of compensated G(s)H(s)

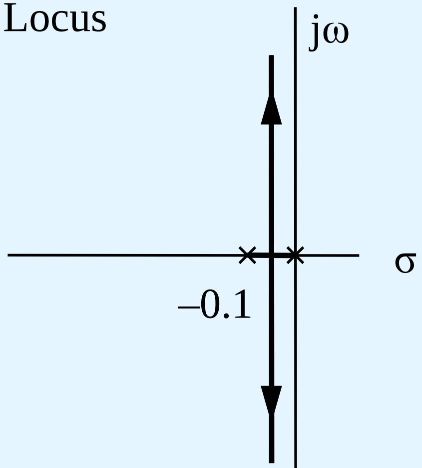
Closed-loop
$\omega_c \approx 0.8$ rad/sec
$\phi_m \approx 51^\circ$

$$G(s)H(s) = \frac{1}{s(s+1)}$$

- Not a good CL design - not a large enough region of –20dB slope around crossover, $\omega_1 \neq \omega_2\sqrt{\beta}$, etc.

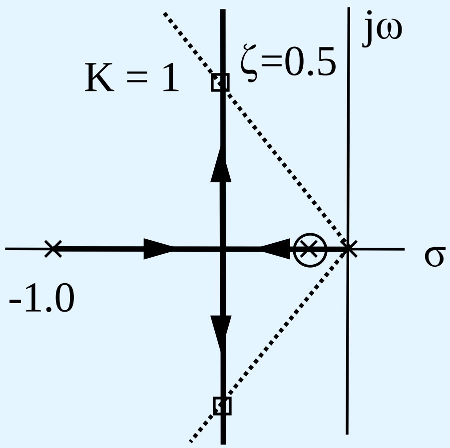
Time Domain Response of Continuous Design

- Root Locus



(a) Root locus of uncompensated system

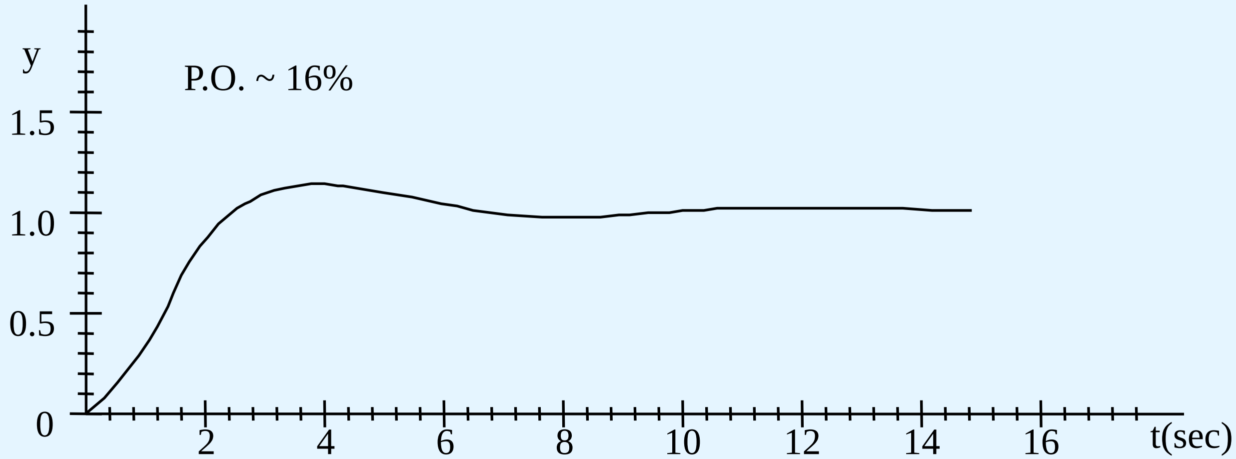
$$1 + KG(s) = 1 + \frac{0.1K}{s(s+0.1)}$$



(b) Root locus of compensated system

$$1 + KG(s)H(s) = 1 + \frac{K}{s(s+1)}$$

- CL Step response



Discrete Equivalent Computations

- Select time step h = 1.0 sec.

  Note: State model of system with $x_1 = v$, $x_2 = y$:

$$\dot{\underline{x}}(t) = \begin{bmatrix} -0.1 & 0 \\ 1.0 & 0 \end{bmatrix} \underline{x}(t) + \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} u(t) \; ; \quad y = \begin{bmatrix} 0 & 1 \end{bmatrix} \underline{x}(t)$$

$$\|A\| = \sqrt{1.01/2} \doteq 0.7 \; ; \quad |\lambda_{max}(A)| = 0.1$$

  so h = 1.0 is compatible with criterion $h < \dfrac{0.5 \rightarrow 1.0}{\|A\|}$.

- Zero-order hold equivalent, $\tilde{G}(z)$

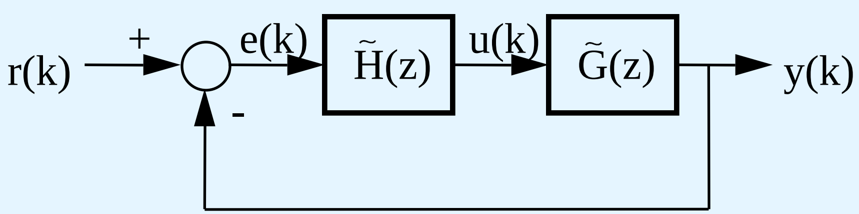
$$\tilde{G}(z) = 0.048 \, \frac{z + 0.967}{(z-1)(z-0.905)}$$

- Tustin equivalent

$$\tilde{H}(z) = H(s)\Big|_{s=2\left(\frac{z-1}{z+1}\right)} = 7\left(\frac{z-0.905}{z-0.333}\right) = 7\left(\frac{1-0.905z^{-1}}{1-0.333z^{-1}}\right) = \frac{u(z)}{e(z)}$$

  - Algorithm

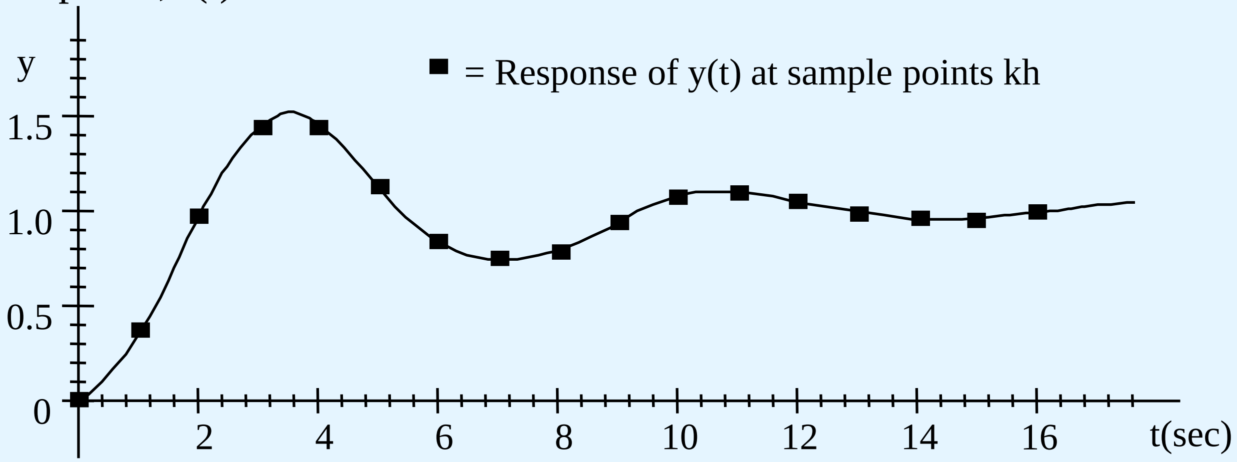
$$u(k) = 7e(k) - 6.335e(k-1) + 0.333u(k-1)$$



Examine CL step response, $LG_{ain}(z)$, etc., for <u>discrete</u> system.

Evaluation of Digital Control Performance

- Step response, $r(t) = 1$.

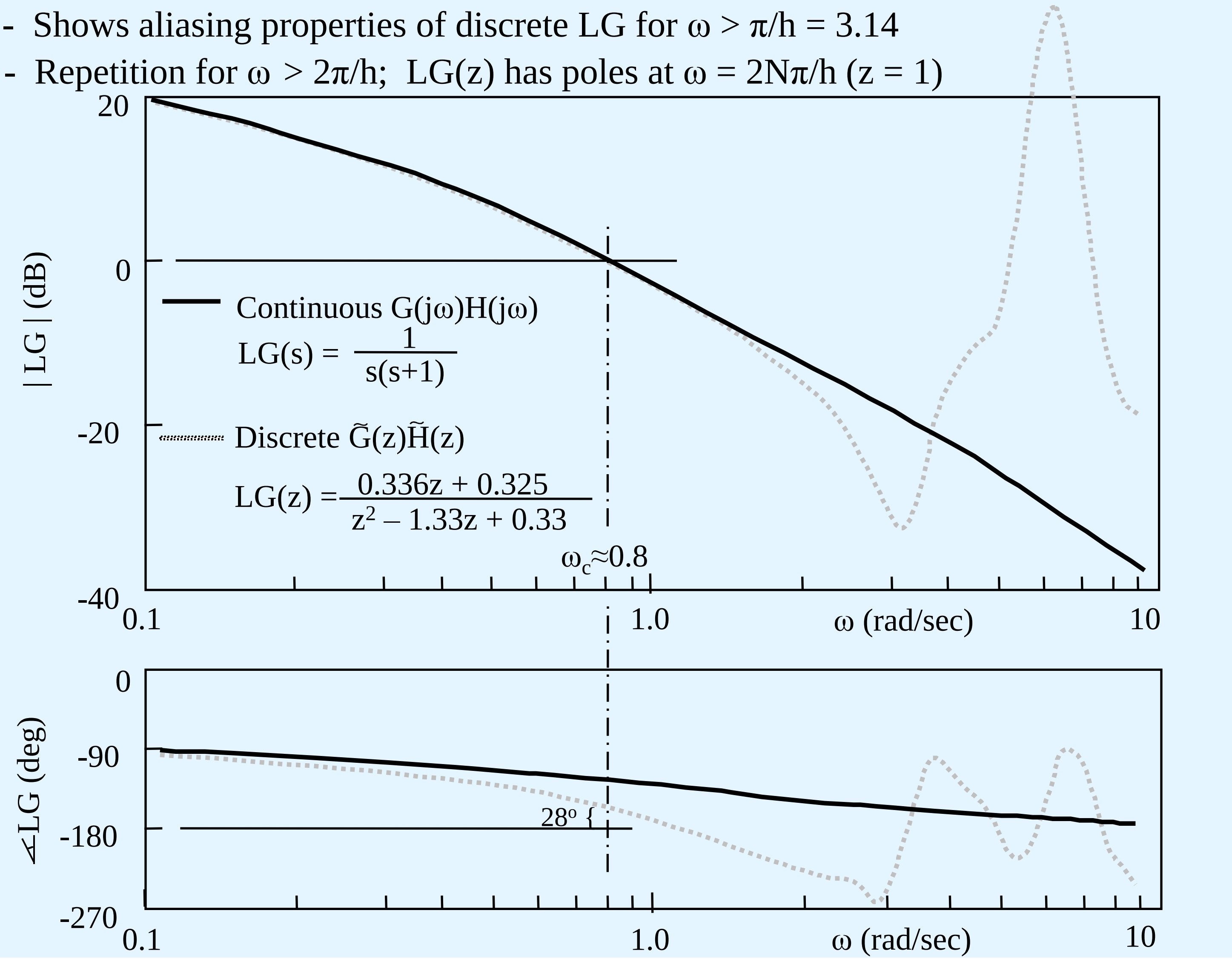


% overshoot is ~ 50%! ($y_{max} \approx 1.5$) This corresponds to $\zeta \sim 0.22$; continuous design had $\zeta \sim 0.5$.

- What happened ?
  - Clearly, there has been a decrease in ϕ_m.
  - $\tilde{H}(e^{j\omega h}) \approx H(j\omega)$, at least in ω_c crossover region.
  - Problem is that $\tilde{G}(e^{j\omega h}) \neq G(j\omega)$ in crossover region.
- Heuristic analysis
  - to a first (crude) approximation $\tilde{G}(e^{j\omega h}) \approx e^{-j\omega h/2}G(j\omega)$, i.e., sampling introduces a delay of h/2 sec.
  - at ω_c get a decrease in ϕ_m of $57.3\omega_c h/2$ deg. => 23^o loss of phase margin here!
  - ϕ_m of discrete system ~ $51^o - 23^o = 28^o$ corresponds to $\zeta \sim 0.25$ (for a 2nd order continuous system).

Continuous vs. Discrete System Loop Gain

- Shows aliasing properties of discrete LG for $\omega > \pi/h = 3.14$
- Repetition for $\omega > 2\pi/h$; LG(z) has poles at $\omega = 2N\pi/h$ ($z = 1$)



Methods to Improve Discrete CL Performance

- Pick the time step, h, so as not to reduce the phase margin much:

$$\Delta\phi_m = 57.3\,(\omega_c h/2)\ \text{deg} < 5 - 10^o$$

  Choosing h in this manner will generally be smaller than when you select h $\approx 0.2/\|A\|$, especially for a lead NW (but not necessarily a lag). But note that very small h may cause CPU timing and other problems.

- Use Tustin with prewarp

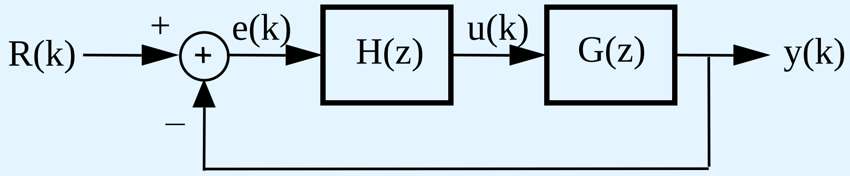
  Not particularly useful here, but could be used to assure $\tilde{H}(z)$ gives little or no magnitude and/or phase distortion in the crossover region.

- Redesign H(s) to give additional positive phase
  - Precompensate for eventual phase decrease in $\tilde{G}(z)$.
  - For given h = 1.0, need a continuous system phase margin of ~ 70^o! : an unreasonable H(s) design.
  - Good approach if $\Delta\phi_m < 15^o$.

- Design H(z) directly in the z-plane
  - $\tilde{G}(z)$ is fundamentally different than G(s).
  - Avoids small time step constraints needed to make Tustin equivalent $\tilde{H}(z)$ perform satisfactorily
  - Less guesswork to modify design.
  - May be possible to use $\tilde{H}(z)$ as a starting point.

=> Use Tustin if $\omega_c h$ is small, otherwise consider direct design of H(z).

Direct Design Compensation Methods



- These schemes work directly with $\tilde{G}(z)$ to design H(z) and so are not limited by the requirement that h ~ small.
  - (i) Root locus design methods
    Compensator design in z-plane using standard root locus design procedures to move CL poles.
  - (ii) w-plane design methods
    This is the equivalent to classical frequency (ω) domain design procedures where w is a rational approximation to (1/h)ln(z).
  - (iii) Fixed-form parametric design
    Assumes a structural form for H(z), e.g., PID, and adjusts free parameters.
  - (iv) Miscellaneous approaches
- Closed-loop transfer function

$$T(z) = \frac{\tilde{G}(z)H(z)}{1+\tilde{G}(z)H(z)} = \frac{y(z)}{r(z)}$$

(1) Zeros of T(z) are the zeros of $\tilde{G}(z)H(z)$ = zeros of $\tilde{G}(z)$ plus those added by H(z).

(2) Poles of T(z) are the roots of $1+\tilde{G}(z)H(z)$.

Root Locus Design of H(z)

$$H(z) = K\frac{(z-\delta_1)(z-\delta_2)\cdots(z-\delta_m)}{(z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_m)} = KH_0(z)$$

- Pick poles and zeros of H(z) so that roots locus of $1+K\tilde{G}(z)H_0(z)$ with respect to gain K passes through the region in z-plane where damping, ζ, and natural frequency, ω_n, are suitable.
  - Do plot on z-plane with ζ, ω_n overlay.
  - Pick δ_i, λ_i, real, generally with $|\lambda_i| \le 1$.
  - Any added zeros δ_i must have an associated pole (no free zeros).
- Generally a first or second order H(z) suffices, e.g.,

$$H(z) = K\frac{z-\delta_1}{z-\lambda_1} = KH_0(z)$$

  - if $\lambda_1 < \delta_1$ => lead compensator
  - if $\lambda_1 > \delta_1$ => lag compensator

> Remember
> $s = 0 \Rightarrow z=1$

- Then pick K so that (dominant) closed-loop poles are at some desired location on the root locus and specs are met.

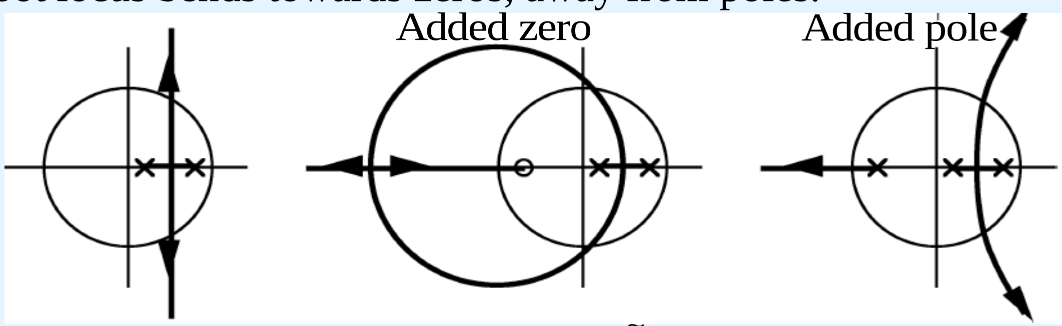
$$K = \left.\frac{-1}{\tilde{G}(z)H_0(z)}\right|_{z = z_{des}}$$

- Next evaluate time response, loop gain $K\tilde{G}(z)H_0(z)$ at $z = e^{j\omega h}$, etc.
- Adjust λ_i, δ_i (and K) until system meets specs.

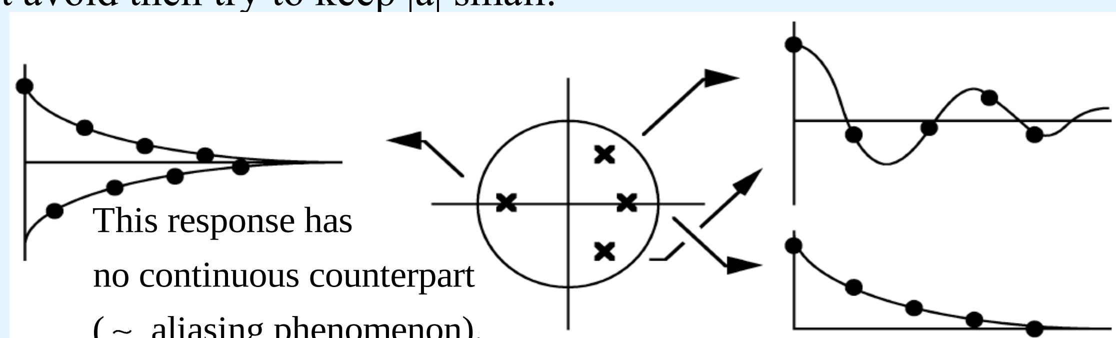
  => trial and error design

Some Helpful Hints for RL Design

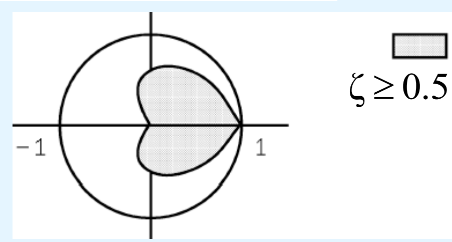
- Recall – Root locus bends towards zeros, away from poles.



- If zero ss error to a constant input is required, $\tilde{G}(z)H(z)$ must have a pole at $z = 1$.
- Try not to have CL poles on $-1 < z < 0$. If there is a pole at $z = -a$ then $y(k)$ or $u(k)$ has a term of the form $(-a)^k \rightarrow 1, -a, +a^2, -a^3, \ldots$ point-to-point oscillation.
  - If can't avoid then try to keep $|a|$ small.



- "Nice" region in z-plane, especially for dominant pair.

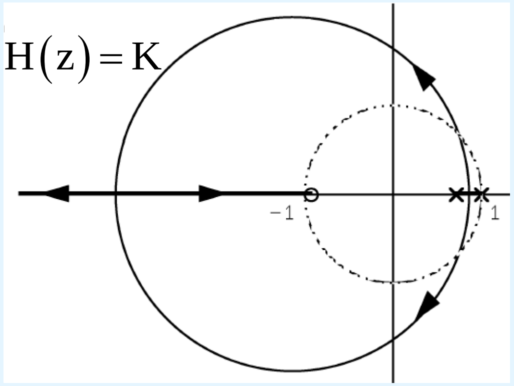


Antenna Positioning Control

$$G(s)=\frac{0.1}{s(s+0.1)} \quad \overset{h=1}{\rightarrow} \quad \tilde{G}(z)=0.048\frac{z+0.97}{(z-1)(z-0.905)}$$

Specs: PO to step input $\approx$ 15%, $t_s|_{1\%} \approx 10\text{sec}$, $\phi_m \geq 50^\circ$

- Uncompensated root locus (pole already at z = 1 via $\tilde{G}$)



Not very good!

- Need a zero on [0, 1] to bend RL inward more.
- Place associated pole on [–1, 0], away from added zero.

To have $t_s|_{1\%} = 5/\zeta\omega_n \approx 10 \quad \Rightarrow \quad$ need $\zeta\omega_n \sim 0.5$ for dominant CL poles

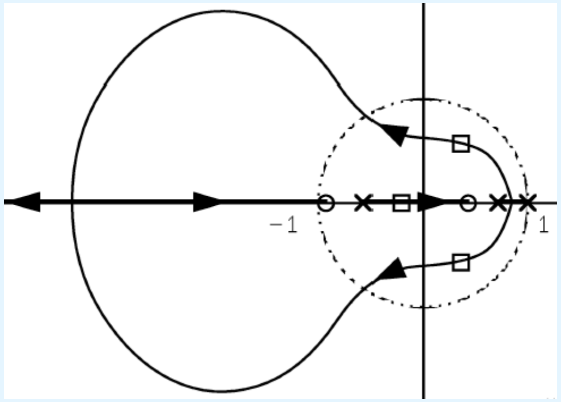
$\Rightarrow$ need $|z| = e^{-\zeta\omega_n h} \approx 0.6$, with ζ=0.5 to PO $\approx$ 15%

- First design trial; $H(z) = K\dfrac{z-0.5}{z+0.6}$

Not too bad –– With K≈20 obtain a dominant CL pair with $\zeta \sim 0.5$.

- Also get a CL pole at z = –0.2 (will this give a problem?)
- Examine CL response via simulation.

$u(k) = 20e(k) - 10e(k-1) - 0.6u(k-1)$

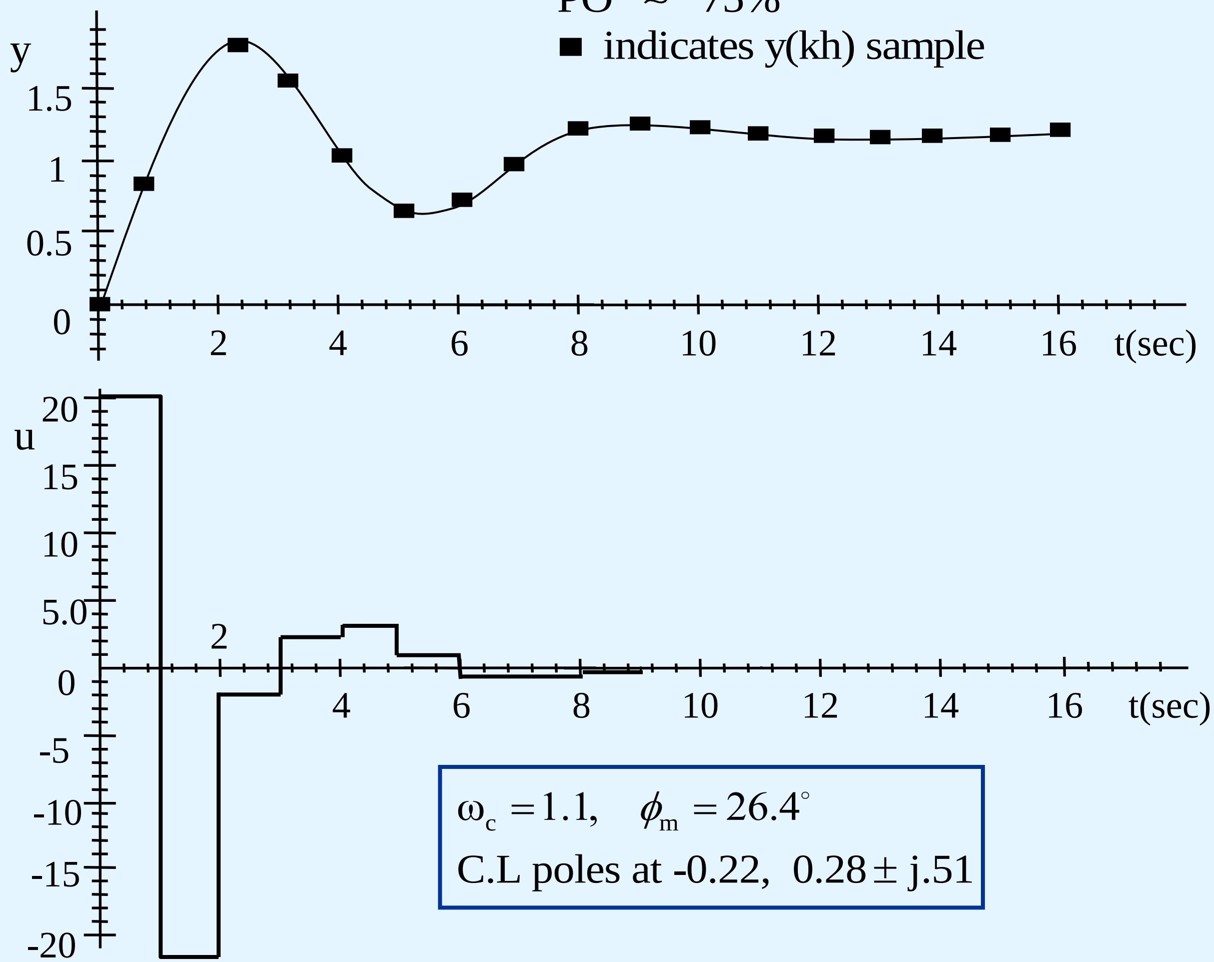


Time Response

$$H(z) = 20\,\frac{z - 0.5}{z + 0.6}$$

PO ~ 75%

■ indicates y(kh) sample



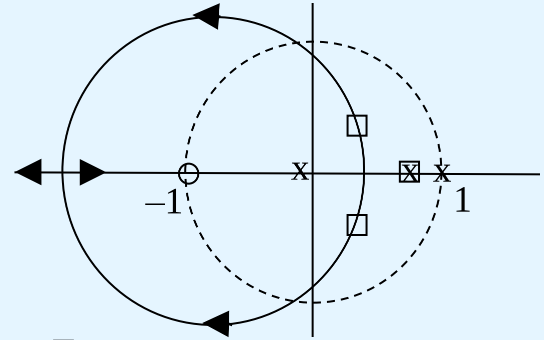
$\omega_c = 1.1, \quad \phi_m = 26.4^\circ$

C.L poles at -0.22, $0.28 \pm j.51$

- Need to reduce gain, move zero at 0.5 closer to pole at z = 0.905.
- Requires movement of pole at –0.6 closer to z = 0.

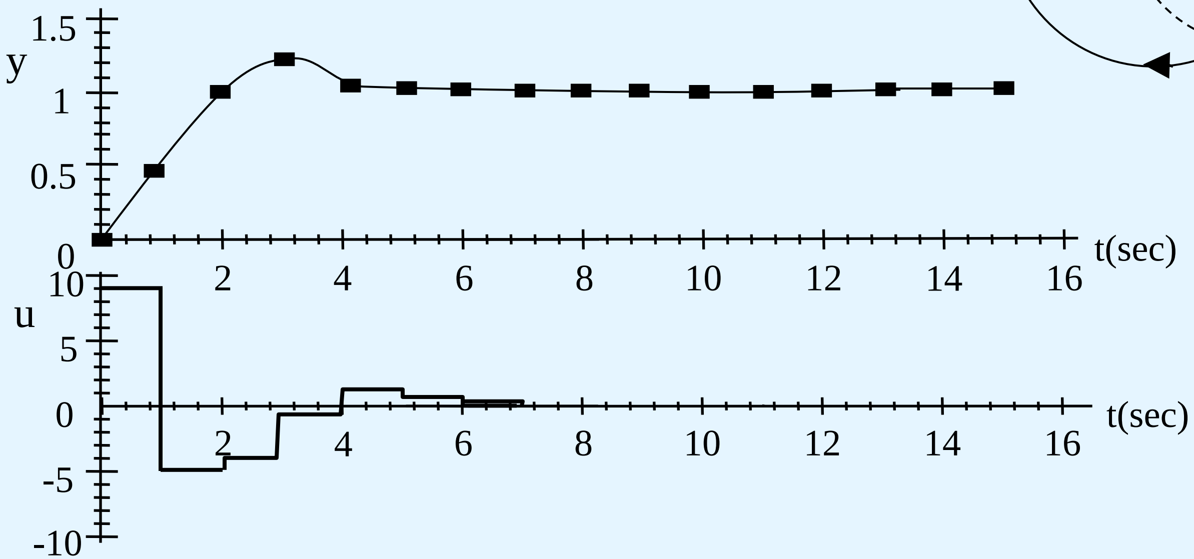
Root Locus Re-Design (After much trial and error)

- Use zero to cancel pole at z = 0.905. Place pole so that root locus goes through nice region ($|z| \le 0.6$, $\zeta \approx 0.5$).



$$H(z) = K\frac{z-0.905}{z+0.2}$$

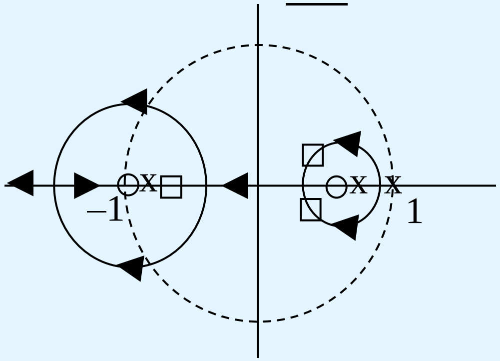
K ~ 9 gives CL poles at $0.18 \pm j0.44$ => $\zeta \approx 0.54$.



- Good design, but K_v has gone from 1.0 (continuous design) to 0.71, since $\frac{1}{h}(1-z^{-1})\tilde{G}(z)H(z)\big|_{z=1} = K_v = 0.71.$
- Bode plot of LG(z) = $\frac{0.432z+0.418}{(z-1)(z+0.2)}$ $\Rightarrow$ ω_c ~ 0.71 rad/sec, ϕ_m ~ 56°

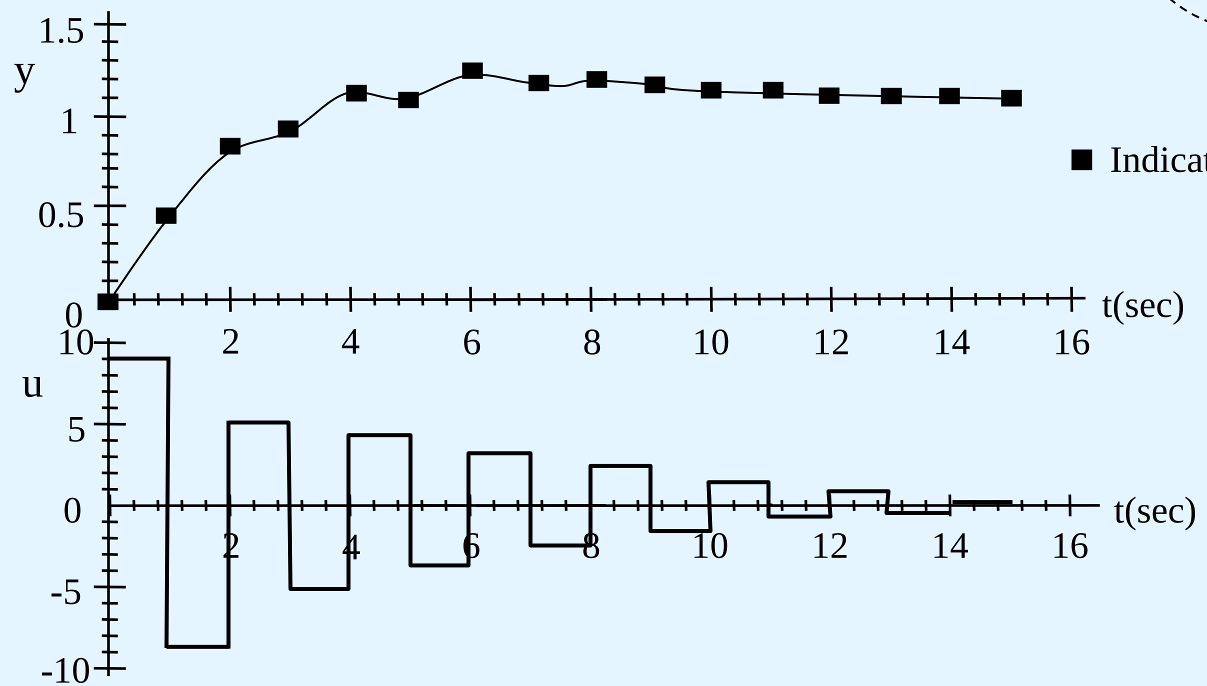
An Example of a Poor Design Choice

Reduce PO further we can move the zero of H(z) close to the pole at z = 0.905 and move the pole of H(z) further out towards z = –1.



$$H(z) = K\frac{z-0.8}{z+0.8}$$

with K = 9 obtain a highly damped system with CL poles:
$z \doteq 0.7 \pm j0.1$ and $z = -0.75$ (ω_c=0.5 rad/sec ,ϕ_m = 62°)



- Intersample "ripples" in y(t) and oscillatory u(k) are indicative of CL poles on negative real axis.

w – Plane Design

- Attempt to use Bode design techniques to obtain H(z) starting with $\tilde{G}(z)$.
- Cannot go into s-plane to design H(s) and then get H(z).
  - Map from z → s plane not rational
  - Need a rational approximation to $z = e^{sh}$
- Define "w – plane" with w ~ s

$$z = \frac{1 + wh/2}{1 - wh/2} \qquad\qquad w = \frac{2}{h}\left(\frac{z-1}{z+1}\right) = \mu + j\nu$$

  - On unit circle, $\nu = \frac{2}{h}\tan\left(\frac{\omega h}{2}\right) \approx \omega$ when $\omega h << 1$
- Rational mapping
  - $\tilde{G}(z) = \frac{b_0 z^n + b_1 z^{n-1} + \cdots + b_n}{z^n + a_1 z^{n-1} + \cdots + a_n} \rightarrow \tilde{G}(w) = \frac{c_0 w^n + c_1 w^{n-1} + \cdots + c_n}{w^n + d_1 w^{n-1} + \cdots + d_n}$
  - $\tilde{G}(w)$ will always be n-th order/n-th order
  - Unit disk $|z| \le 1$ mapped into LHP $\mathrm{Re}(w) \le 0$

$$\tilde{G}(w)\Big|_{w=j\nu} \approx G(z)\Big|_{z=e^{j\omega h}} \quad \text{if } \omega h << 1$$

  - To first approximation ($\omega << \pi/h$)

$$\tilde{G}(w)\Big|_{w=j\nu} \doteq G(s)e^{-sh/2}\Big|_{s=j\omega}$$

- Can include as an additional option in Bode plot subroutine

Design Approach

$G(s) \longrightarrow \tilde{G}(z)$

$\tilde{G}(z) \downarrow \quad z = \dfrac{1 + wh/2}{1 - wh/2}$

$\tilde{G}(w) \longrightarrow H(w)$

$H(w) \uparrow \quad w = \dfrac{2}{h}\left(\dfrac{z-1}{z+1}\right) \quad H(z)$

Bode Plot Design Techniques

- $z - to - w$ Transformation

  (i) if

$$\tilde{G}(z) = \frac{K \prod_{i=1}^{m} (z + \delta_i)}{(z-1)^k \prod_{j=1}^{n-k} (z + \lambda_j)}$$

then,

$$\tilde{G}(w) = \frac{K \prod_{i=1}^{m} (1+\delta_i) \left(1 - \dfrac{w}{2/h}\right)^{n-m} \prod_{i=1}^{m} \left(1 + \dfrac{w}{(2/h)\left[(1+\delta_i)/(1-\delta_i)\right]}\right)}{\left[\prod_{j=1}^{n-k} (1+\lambda_j)\right] h^k w^k \prod_{j=1}^{n-k} \left(1 + \dfrac{w}{(2/h)\left[(1+\lambda_j)/(1-\lambda_j)\right]}\right)}$$

  - Useful formula when $\tilde{G}(z)$ has only real poles and zeros

  (ii) Stat-space approach in general case

  - Need a general technique that is computer-oriented

- $w - to - z$ Transformation

  - Identical to Tustin transform on H(w)

General z – to – w Plane Mapping

- Given $\dot{\underline{x}}(t) = A\,\underline{x}(t) + Bu(t)$, $y(t) = C\,\underline{x}(t)$ } System to be controlled

  determine $\tilde{G}(w)$

  1. Obtain equivalent discrete system in usual manner

     $$\left.\begin{aligned} \underline{x}(k+1) &= \Phi\,\underline{x}(k) + \Gamma u(t) \\ y(k) &= C\,\underline{x}(k) \end{aligned}\right\} \quad \tilde{G}(z) = C(zI - \Phi)^{-1}\Gamma$$

  2. z – transform: $z\underline{x}(z) = \Phi\,\underline{x}(z) + \Gamma u(z)$
  3. Let $z = \dfrac{1 + wh/2}{1 - wh/2}$

     $$(1 + wh/2)\,\underline{x}(w) = (1 - wh/2)\Phi\underline{x}(w) + (1 - wh/2)\Gamma u(w)$$
     $$y(w) = C\,\underline{x}(w)$$

  4. Solve for y(w)

     $$y(w) = \underbrace{C\Big[wI - \overbrace{\tfrac{2}{h}(\Phi + I)^{-1}(\Phi - I)}^{\tilde{\Phi}}\Big]^{-1}\overbrace{(\Phi + I)^{-1}\Gamma}^{\tilde{\Gamma}}\left(\frac{2}{h} - w\right)}_{\tilde{G}(w)}u(w)$$

  Augmented System: $\begin{bmatrix} x \\ \underline{u} \end{bmatrix}$; input $= \dot{\underline{u}}$

  $\begin{bmatrix} \Phi_a & \Gamma_a \\ C_a & 0 \end{bmatrix}$; $\Phi_a = \begin{bmatrix} \tilde{\Phi} & \frac{2}{h}\tilde{\Gamma} \\ 0 & 0 \end{bmatrix}$;

  $\Gamma_a = \begin{bmatrix} -\tilde{\Gamma} \\ I \end{bmatrix}$; $C_a = [C \quad 0]$

  $\tilde{\Phi} = \frac{2}{h}(\Phi + I)^{-1}(\Phi - I)$

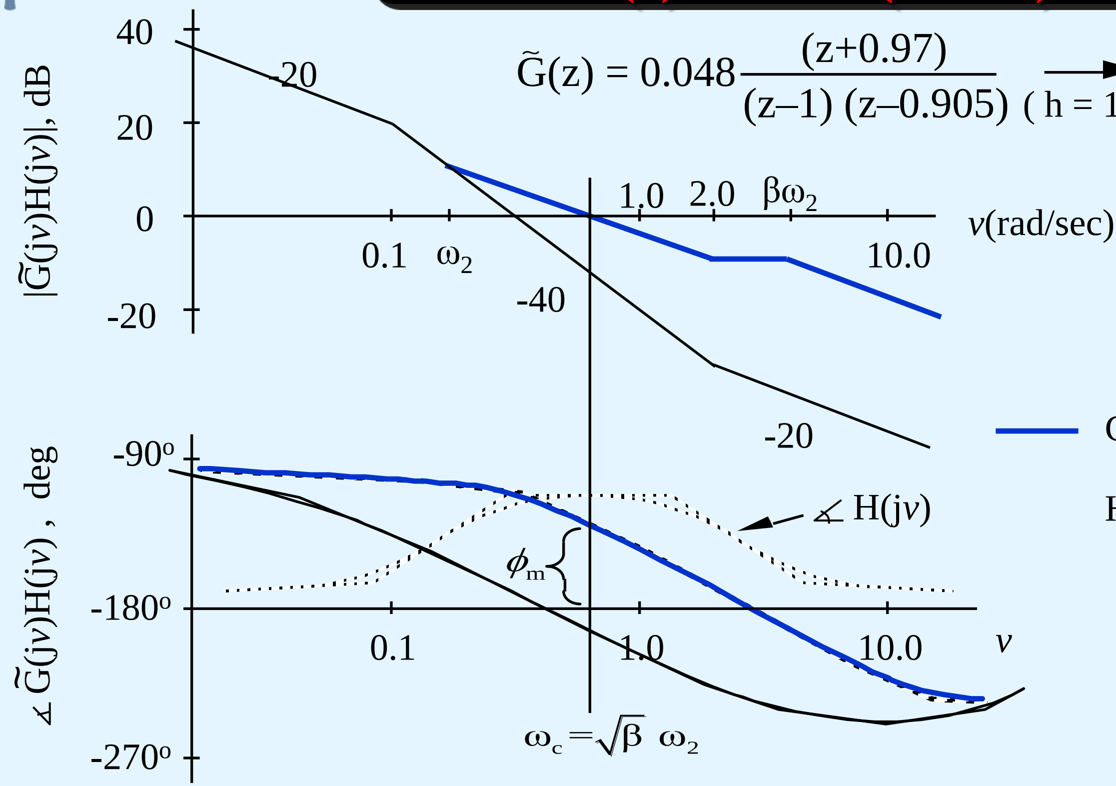
  $\tilde{\Gamma} = (\Phi + I)^{-1}\Gamma$

  - Use Leverier with $\tilde{\Phi}$ and $\tilde{\Gamma}$ to obtain $C(wI - \tilde{\Phi})^{-1}\tilde{\Gamma}$, then include (2/h – w) factor.
  - Note non-minimum phase zero at w = 2/h.

- Follow general Tustin state-space approach for w – to – z plane $H(w) \rightarrow H(z)$.

Antenna Positioning Controller: G(s) = 0.1/s(s+0.1)

$$\tilde{G}(z) = 0.048\frac{(z+0.97)}{(z-1)(z-0.905)} \xrightarrow{(h=1.0)} \tilde{G}(w) = \frac{0.1\,(1+w/120)\,(1-w/2)}{w(1+w/0.1)}$$



$$H(w) = 1.0\,\frac{1+w/0.14}{1+w/4.2} = K\,\frac{1+w/\omega_2}{1+w/\beta\omega_2} = KH_0(w)$$

- Use lead compensation to keep bandwidth up.
- Make ω_c as large as possible with a $\phi_m \sim 55^o$.
- Use limit value of $\beta = 30$ (corresponds to $\Delta\phi_{max} \approx 69^o$).
  - $\Rightarrow \omega_c \sim 0.77 \left(\text{where } \measuredangle G(jv) = -180^\circ + \phi_m - 69^\circ = -194^\circ\right)$
  - $\Rightarrow \omega_2 = \omega_c / \sqrt{\beta} = 0.14,\ \ \beta\omega_2 = 4.2$
- Pick K so that $K\left|\tilde{G}(w)H_0(w)\right|_{w=j0.77} = 1 \Rightarrow K \approx 1.0$

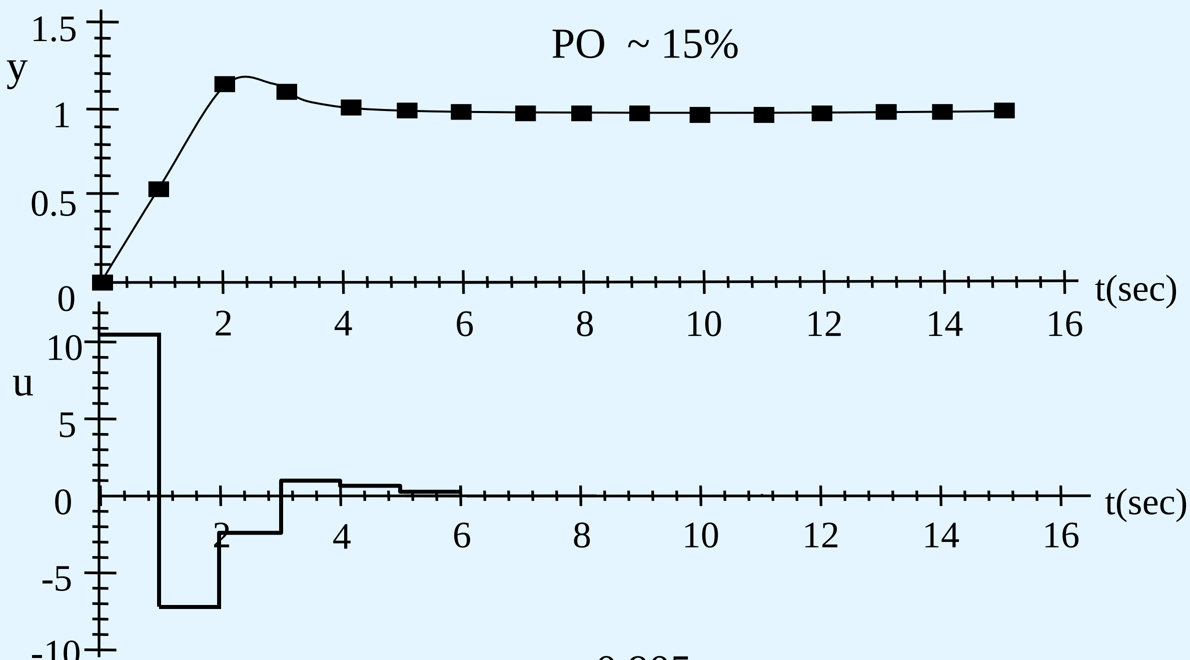
w – to – z (Backward) Transformation

$$H(z) = 1.0\frac{1+w/0.14}{1+w/4.2}\bigg|_{w=\frac{2}{h}\left(\frac{z-1}{z+1}\right)} = 10.5\frac{z-0.87}{z+0.35}$$

- $H(1) \approx 1$ => No reduction in low frequency gain

$$K_v = \left(\frac{z-1}{zh}\right)\tilde{G}(z)H(z)\Big|_{z=1} = 1.0 \quad \text{(same as continuous design)}$$

- Time response



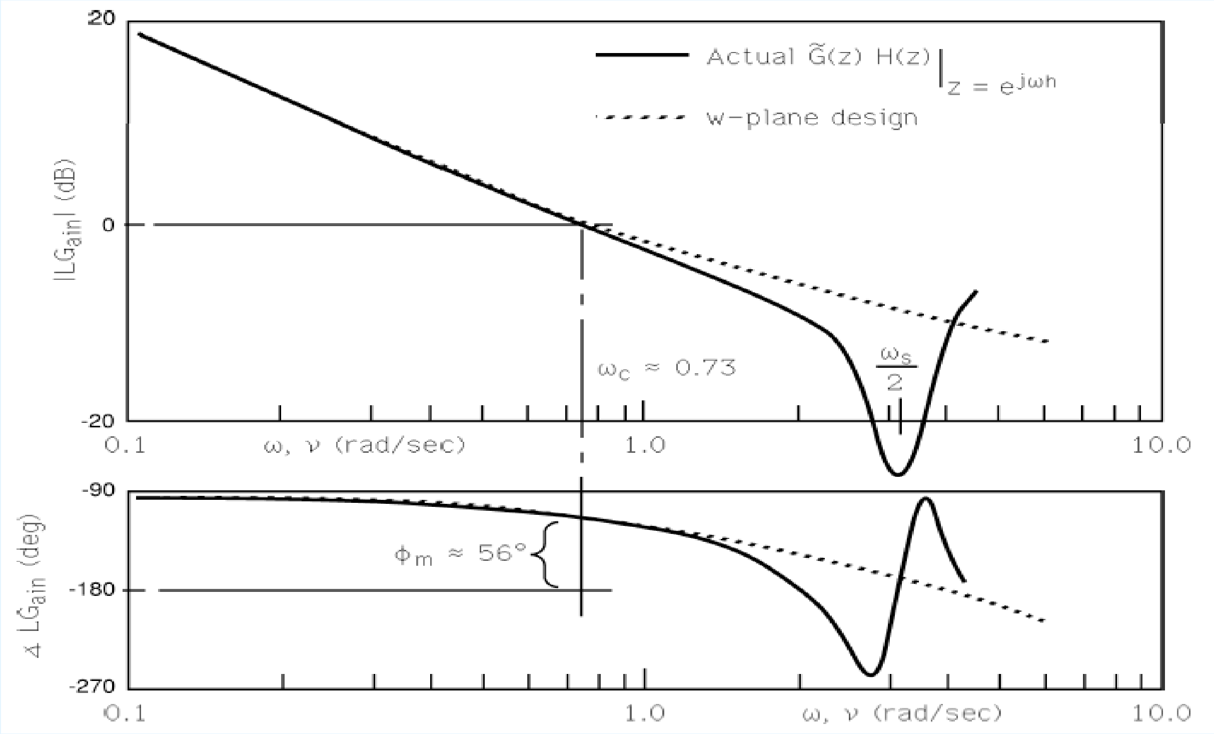
- Very similar to RL design, $H(z) = 9\frac{z-0.905}{z+0.2}$ (a bit faster/better)

Frequency Domain Evaluation

- Examine actual $LG(z)|_{z=e^{j\omega h}}$ to find true ω_c, ϕ_m

$$LG(z) = \tilde{G}(z)H(z) = 0.504\frac{(z+0.967)(z-0.87)}{(z-1)(z-0.905)(z+0.35)}$$

- Compare with $\tilde{G}(w)H(w)$, $w = j\nu$



- Discrete loop gain is very similar to root locus design with ~3dB higher very low frequency gain.
- w – plane design approximation is OK for $\nu \sim \omega < 1/h$
- Actual $\phi_m \approx 56^o$, $\omega_c \approx 0.73$ (system will tolerate a maximum loop delay $\tau_{max} = \phi_m/\omega_c = 1.34$ sec)

Root Locus vs. w – Plane Design Comparison

- Either approach, used correctly, will give a good design.

Root Locus Design

- RL plot more difficult to draw than Bode plot
- Hard to see where to place poles and zeros of H(z) to properly shape RL as desired.
- Seems to require more trial and error than does Bode approach.
- Need overlay of $\zeta - \omega_n$ contours on RL plot.
- Difficult to make engineering approximations.
- If h ~ small, the RL tends to crowd into region around z = 1.

Bode/ w – Plane Design

- Easier to work with and to modify than is RL.
- Requires $z \rightarrow w$ mapping on $\tilde{G}$, then reverse map on H.
- Still need to evaluate frequency plot of LG in z-domain, since $w \neq s$.
- No guarantee that a good w – plane design will yield a good z – plane design (unless $v < 1/h$).
- Gives no explicit knowledge of CL pole locations.

Digital PID Controller

- Discrete equivalent obtained from backward difference (other methods are also used), $s \rightarrow (z-1)/hz$:

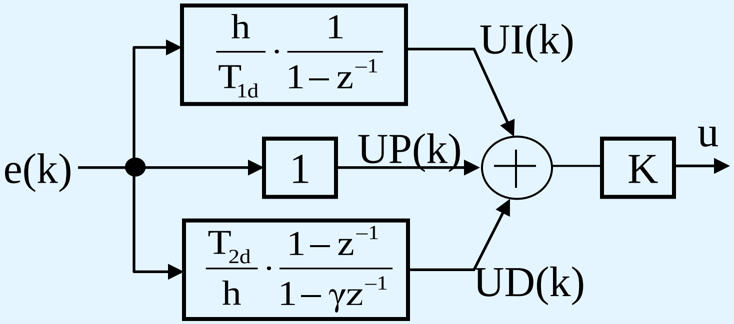
$$u(z) = K\left[1 + \frac{hz}{T_1(z-1)} + \frac{T_2}{\left(h + \frac{T_2}{N}\right)} \cdot \frac{(z-1)}{\left(z - \underbrace{\frac{T_2}{Nh+T_2}}_{\gamma}\right)}\right] e(z)$$

- General parametric form

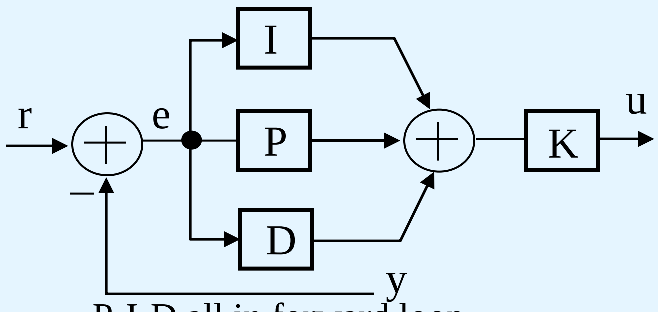
$$u(z) = \underbrace{K\left[1 + \frac{h}{T_{1d}} \cdot \frac{1}{1-z^{-1}} + \frac{T_{2d}}{h}\frac{1-z^{-1}}{1-\gamma z^{-1}}\right]}_{H(z)} e(z)$$

to be determined: K, T_{1d}, T_{2d}, and possibly γ, ($\gamma = T_{2d}/Nh$), $T_{2d} = \dfrac{T_2 Nh}{T_2 + Nh}$

- Implementation – "Textbook"   Sum up 3 parts separately:



≡



P-I-D all in forward loop

$UI(k) = (h/T_{1d})e(k) + UI(k-1)$; $UP(k) = e(k)$; $UD(k) = (T_{2d}/h)[e(k) - e(k-1)] + \gamma UD(k-1)$

then $u(k) = K[UI(k) + UP(k) + UD(k)]$

PID Algorithm Implementation

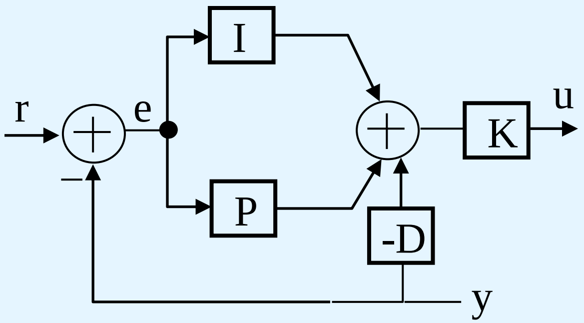
- Algorithm at step k

$$e = r - y$$
$$UI = (h/T_{1d})e + UI$$
$$UP = e$$
$$UD = (T_{2d}/h)[e - e_{last}] + \gamma UD$$
$$e_{last} = e$$
$$u = K(UI + UP + UD)$$

- Derivative on output
  - If r suddenly changes from time k–1 to time k, e.g., a step change, then e(k) – e(k–1) may be large and UD will have a “spike” at step k: This is undesirable.
  - Modify UD computation to use only $\Delta y = y(k) - y(k-1)$,

$$UD(k) = -(T_{2d}/h)[y(k) - y(k-1)] + \gamma UD(k-1)$$

This is “derivative of output form”. Since y(k) cannot change very much from step k–1 to k, UD will be OK.



  - CL stability is unaffected (stability not a function of r).
- “Set-point on I” structure
  - Move P to act only on y also, UP = –y(k)
  - Only integral compensation uses error signal.
  - Popular structure in process control (keeps control signal very smooth).

Integral Windup Modifications

- A problem that arises when u is limited, e.g.,

$$B^{-} \leq u(k) \leq B^{+}$$

(symmetric limits are most common, $B^{-} = -B^{+}$)

- Limits are imposed by the system under control, e.g., actuator constraints.
  - Match these limits in controller software:

$$\text{if } (u \geq B^{+}) \text{ set } u = B^{+}, \ \text{flag} = +1$$
$$\text{if } (u \leq B^{-}) \text{ set } u = B^{-}, \ \text{flag} = -1$$
$$\text{else flag} = 0$$

- The control probably saturated because e(k) was large.
  - Because u is limited the error e will not be reduced to zero as fast (slower system).
  - This is not indicative of a steady-state e.

=> Turn off/skip the integration of e(k) in UI if the last control value was at a limit

$$\text{if (flag} = 0) \text{ UI} = \text{UI} + (h/T_{1d})e$$
$$\text{if (flag} \neq 0) \text{ UI} = \text{UI}$$

- Integral protection
  - Value of UI does not change if/when u is saturated.
- Include PID structure in Cntrl routine as an option during evaluation

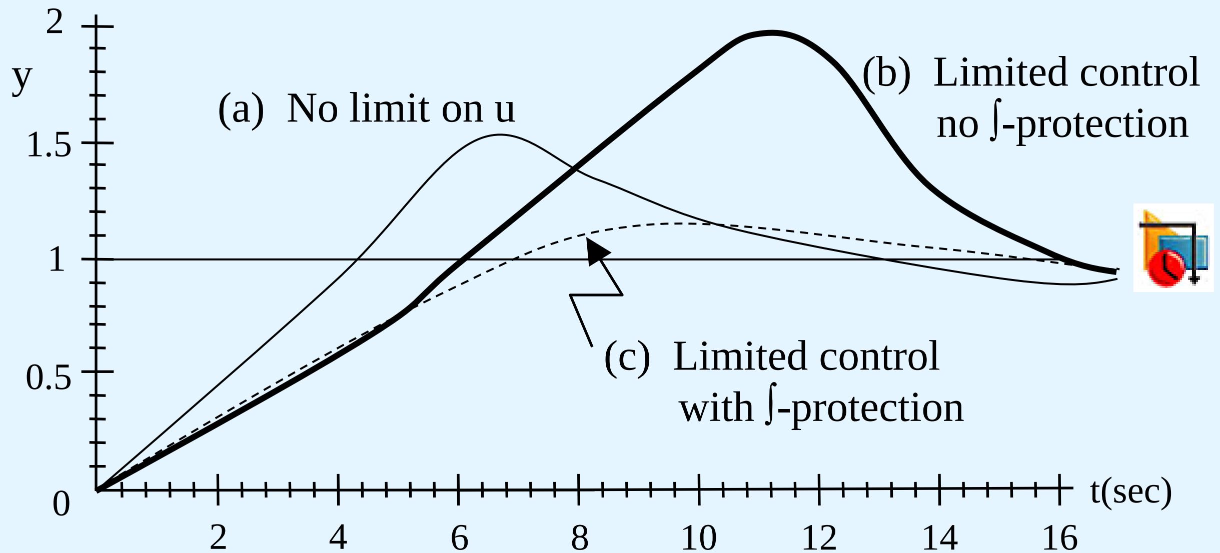
Example (Aström and Wittenmark)

- Lack of integral protection will often result in large overshoots in system response.
  - Since long periods of + (or –) e will cause UI to build up large values. Then e reverses…
- Ex. A motor with transfer function G(s) = 1/s(s+1) is to be controlled using a digital PI controller*

$$u(z) = K\left[1 + \frac{h}{T_{1d}} \cdot \frac{1}{1 - z^{-1}}\right] e(z)$$

with $K = 0.4$, $T_{1d} = 5$ sec, $h = 0.5$ sec.

- Examine step response when $|u(k)| \leq 0.2$, with and without integral windup protection.



* Note: The I part of the controller is not really needed here since G(s) contains a 1/s. But it is only an example.

Unified PID for Various Approximations

- Parameters for different approximations

$e = r - y$
$UI = UI + \alpha_1 e + \alpha_2 e_{past}$
$UP = e$
$UD = -\delta_d [y - y_{past}] + \gamma UD$
$e_{past} = e$
$u = K(UI + UP + UD)$

$$u(s) = K[1 + \frac{1}{T_1 s} + \frac{T_2 s}{1 + T_2 s / N}]e(s)$$

$$T_{1d} = T_1$$

$$T_{2d} = \frac{T_2 Nh}{T_2 + Nh}$$

Parameter`	Forward	Backward	Tustin	Ramp
α_1	0	h/T_{1d}	$h/2T_{1d}$	$h/2T_{1d}$
α_2	h/T_{1d}	0	$h/2T_{1d}$	$h/2T_{1d}$
γ	$1-Nh/T_2$ $=2-Nh/T_{2d}$	T_{2d}/Nh	$(2T_2-Nh)/(2T_2+Nh)=$ $(3T_{2d}-Nh)/(T_{2d}+Nh$	e^{-Nh/T_2} $=e^{1-Nh/T_{2d}}$
δ_d	N	T_{2d}/h	$2N/(1+Nh/T_{2d})$	$T_2(1-e^{-Nh/T_2})/h=$ $(\frac{T_{2d}N}{Nh-T_{2d}})(1-e^{(1-Nh/T_{2d})})$

- Velocity algorithm (compute Δu)

$e = r - y$
$\Delta UI = \alpha_1 e + \alpha_2 e_{past}$
$\Delta UP = e - e_{past}$

$\Delta UD = -\delta_d [y - 2y_{past} + y_{pastpast}] + \gamma \Delta UD$
$e_{past} = e$; $y_{pastpast} = y_{past}$; $y_{past} = y$
$\Delta u = K(\Delta UI + \Delta UP + \Delta UD)$

IMC Design Approach - 1

- IMC design approaches for stable **and possibly non-minimum** phase systems
  - Step 1: Split $\tilde{G}(z) = z^{-k}\frac{b(z)}{a(z)} = z^{-k}\frac{b^{+s}(z)b^{-}(z)b^{nm+}(z)}{a(z)}$ as follows:

    Here b^{+s} = Part of $b(z)$ *with* zeros having positive real parts <u>and</u> inside unit circle

    b^{-} = Part of $b(z)$ with zeros having negative real parts (inside <u>and</u> outside unit •)

    b^{nm+} = Part of $b(z)$ with zeros having positive real parts <u>and</u> outside unit circle
  - Step 2: (i) Replace part with zeros having negative real part with a DC gain (set z=1)

    (ii) Replace non-minimum phase zeros with their reciprocals

    (iii) Add filters of the form $F(z) = (\frac{1-\alpha}{1-\alpha z^{-1}})^{k}; k \geq 1$ so that $Q(z) = \tilde{G}^{-1}(z)F(z)$ is proper
  - Step 3: $H(z) = Q(z)[1-\tilde{G}(z)Q(z)]^{-1}$

- **Example**

$$G(s) = \frac{-s+3}{s^2+5s+6}; h = 0.05\sec \Rightarrow \tilde{G}(z) = \frac{-0.040678(z-1.163)}{(z-0.9048)(z-0.8607)} = \frac{-0.040678z^{-1}(1-1.163z^{-1})}{(1-0.9048z^{-1})(1-0.8607z^{-1})} = z^{-1}\frac{b(z)}{a(z)}$$

$$b^{+s} = -0.040678; b^{-} = 1; b^{nm+}(z) = (1-1.163z^{-1}) \Rightarrow \textit{replace by } (1-1.163z) = z\,(z^{-1}-1.163)$$

$$So, Q(z) = \frac{a(z)}{b(z)}F(z) = \frac{(1-0.9048z^{-1})(1-0.8607z^{-1})}{-0.040678(z^{-1}-1.163)}F(z) = \frac{(z-0.9048)(z-0.8607)}{0.0473(z-0.8598)}\frac{1-\alpha}{z-\alpha}$$

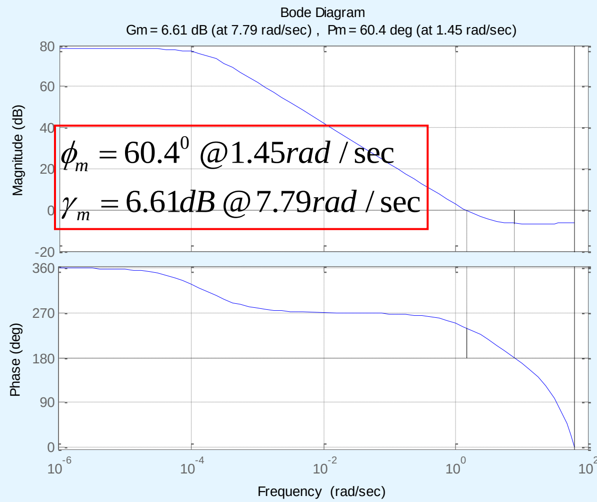
$$= \frac{10.5708(z-0.9048)(z-0.8607)}{(z-0.8598)(z-0.5)}; \alpha = 0.5$$

Select α based on other criteria, e.g., phase margin, settling time

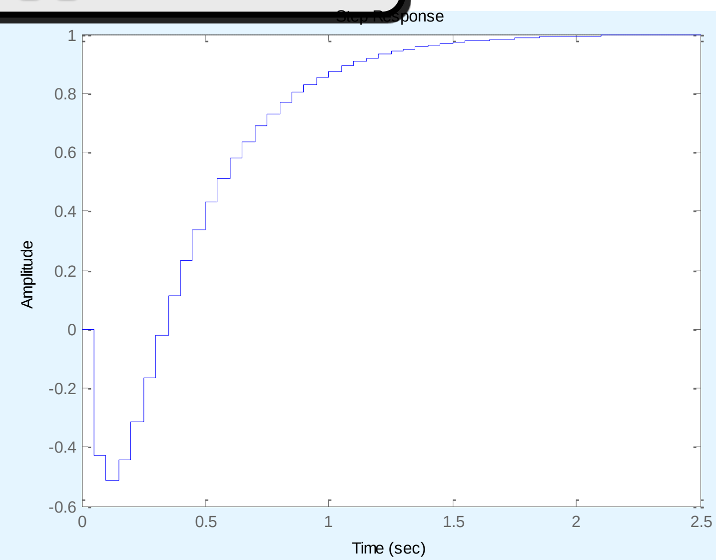
$$H(z) = Q(z)[1-\tilde{G}(z)Q(z)]^{-1} = \frac{10.5708(z-0.9048)(z-0.8607)}{(z+0.07013)(z-1)}$$

- Step response exhibits an undershoot as one would expect from a non-minimum phase system

IMC Design Approach - 2

- 

$$\phi_m = 60.4^0 \ @1.45 rad/\sec$$
$$\gamma_m = 6.61dB \ @7.79 rad/\sec$$



- **Example 2**

$$G(s) = \frac{1}{(10s+1)(25s+1)}; h = 2\sec \Rightarrow \tilde{G}(z) = \frac{0.00729(z+0.9109)}{(z-0.9231)(z-0.8187)}$$

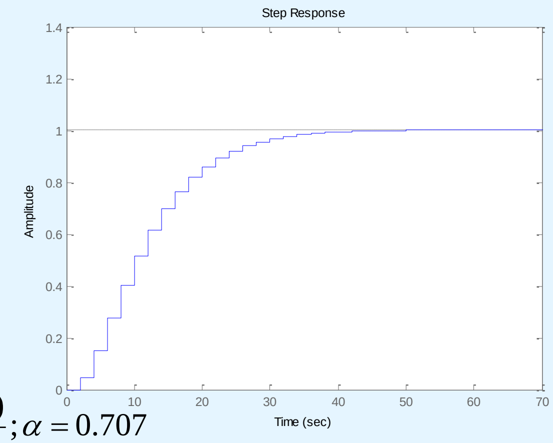
$$so, \tilde{G}(z) = \frac{0.00729z^{-1}(1+0.9109z^{-1})}{(1-0.9231z^{-1})(1-0.8187z^{-1})} = z^{-1}\frac{b(z)}{a(z)}$$

$$b^{+s} = 0.00729; b^{-} = (1+0.9109z^{-1}) \Rightarrow replace\ by\ 1.9109; b^{nm+}(z) = 1$$

$$So, Q(z) = \frac{a(z)}{b(z)}F(z) = \frac{(1-0.9231z^{-1})(1-0.8187z^{-1})}{0.0139}F(z)$$

$$= \frac{71.94(z-0.9231)(z-0.8187)}{1}\frac{(1-\alpha)^2}{(z-\alpha)^2} = \frac{6.17(z-0.9231)(z-0.8187)}{(z-0.707)^2}; \alpha = 0.707$$

$$H(z) = Q(z)[1-\tilde{G}(z)Q(z)]^{-1} = \frac{6.17(z-0.9231)(z-0.8187)}{(z-0.4588)(z-1)}$$



$$\phi_m = 73.8^0 \ @0.0778 rad/\sec$$
$$\gamma_m = 22.4dB \ @0.562 rad/\sec$$

Pole Placement Method: Shaping T(z) - 1

- What is feasible if you have unstable and non-minimum phase systems?

Suppose $\tilde{G}(z) = z^{-k}\dfrac{b(z)}{a(z)}$ and want $T(z) = K_r\, z^{-k}\dfrac{b_r(z)}{d(z)}$. DC gain $=1 \Rightarrow K_r = \dfrac{d(1)}{b_r(1)}$. What is feasible for $b_r(z)$?

Let $b(z) = \underbrace{b^{+s}(z)}_{good}\overbrace{b^{-}(z)b^{nm+}(z)}^{bad};\quad a(z) = \underbrace{a^{+s}(z)}_{good}\;\overbrace{a^{-}(z)a^{us+}(z)}^{bad.\ also\ close\ to\ unit\ circle}$

Consider a control scheme given by $d_h(z)u(z) = K_r p(z)r(z) - q(z)y(z)$; $d_h(z), p(z), q(z)$ are polynomials in z^{-1}

$p(z) = q(z) \Rightarrow$ single DOF controller

So, $z^{-k}b(z)[d_h(z)u(z) + q(z)y(z)] = K_r z^{-k}p(z)b(z)r(z)$. Recall $a(z)y(z) = z^{-k}b(z)u(z)$

$$\Rightarrow [d_h(z)a(z) + z^{-k}b(z)q(z)]y(z) = K_r z^{-k}p(z)b(z)r(z) \Rightarrow \frac{y(z)}{r(z)} = \frac{K_r z^{-k}p(z)b(z)}{d_h(z)a(z) + z^{-k}b(z)q(z)}$$

$$\Rightarrow \frac{K_r z^{-k}p(z)b^{+s}(z)b^{-}(z)b^{nm+}(z)}{d_h(z)a^{+s}(z)a^{-}(z)a^{us+}(z) + z^{-k}b^{+s}(z)b^{-}(z)b^{nm+}(z)q(z)} = K_r\, z^{-k}\frac{b_r(z)}{d(z)}$$

What if we select $d_h(z) = b^{+s}(z)d_1(z); q(z) = a^{+s}(z)q_1(z); p(z) = a^{+s}(z)p_1(z)$

$$then\ \frac{K_r p_1(z)b^{-}(z)b^{nm+}(z)}{d_1(z)a^{-}(z)a^{us+}(z) + z^{-k}b^{-}(z)b^{nm+}(z)q_1(z)} = K_r\frac{b_r(z)}{d(z)}$$

$$\underbrace{d_1(z)}_{Y(z)}\underbrace{a^{-}(z)a^{us+}(z)}_{M(z)} + \underbrace{z^{-k}b^{-}(z)b^{nm+}(z)}_{N(z)}\underbrace{q_1(z)}_{X(z)} = \underbrace{d(z)}_{D(z)}$$

$$X(z)N(z) + Y(z)M(z) = D(z)$$

$\Rightarrow b_r(z) = p_1(z)b^{-}(z)b^{nm+}(z) \Rightarrow$ keep "bad" zeros in the closed-loop system

$\Rightarrow \dfrac{d_1(z)a^{-}(z)a^{us+}(z)}{d(z)} + \dfrac{z^{-k}b^{-}(z)b^{nm+}(z)q_1(z)}{d(z)} = 1 \sim$ Bezout Identity. Get $d_1(z)$ and $q_1(z)$ by equating coefficients.

Solving N(z) X(z) +M(z) Y(z) = D(z)

$$\overbrace{\begin{bmatrix} X(z) & Y(z) \end{bmatrix}}^{V(z)} \underbrace{\begin{bmatrix} N(z) \\ M(z) \end{bmatrix}}_{F(z)} = D(z) \Rightarrow V(z)F(z) = D(z)$$

$$N(z) = n_1 z^{-1} + n_2 z^{-2} + ... + n_k z^{-k}; X(z) = x_0 + x_1 z^{-1} + x_2 z^{-2} + ... + x_p z^{-p}$$

$$M(z) = 1 + m_1 z^{-1} + m_2 z^{-2} + ... + m_l z^{-l}; Y(z) = y_0 + y_1 z^{-1} + y_2 z^{-2} + ... + y_p z^{-p}$$

$$D(z) = 1 + d_1 z^{-1} + d_2 z^{-2} + ... + d_n z^{-n}$$

$$F_0 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; F_i = \begin{bmatrix} n_i \\ m_i \end{bmatrix}; i \geq 1$$

$$F(z) = F_0 + F_1 z^{-1} + ... + F_m z^{-m}; m = \max(k,l); F_i = 2 - vector$$

$V(z) = V_0 + V_1 z^{-1} + ... + V_p z^{-p}$; don't know p a priori. Each V_i is a 2-row vector.

$$V_i = \begin{bmatrix} x_i & y_i \end{bmatrix}; i \geq 0$$

$$\begin{bmatrix} V_0 & V_1 & . & V_p \end{bmatrix} \begin{bmatrix} F_0 & F_1 & . & F_m & 0 \\ 0 & F_0 & F_1 & . & F_m \\ 0 & 0 & F_0 & F_1 & \\ 0 & 0 & 0 & F_0 & \end{bmatrix} = \begin{bmatrix} 1 & d_1 & . & d_n \end{bmatrix} \Rightarrow VF = D$$

Deconvolution Problem

- Find p such that rows of F are independent.
- Get $X(z)$ from odd coefficients of V
- Get $Y(z)$ from even coefficients of V

Pole Placement Method: Shaping T(z) - 2

- **Example**
- $G(s)=\dfrac{-280.14}{s^3+100s^2-981s-98100}$; poles at $31.321, -31.321, -100; h=0.002$ sec

  Want $t_s \le 0.5$ sec for 2% error, $\%OS \le 5\%$ and steady state error ≤ 0.02

  $\Rightarrow K_P \ge 50 \Rightarrow T(1)=K_P/(K_P+1)=0.9804$

  $\Rightarrow \zeta\omega_n \ge 8$ *and* $\zeta=0.69 \Rightarrow \omega_n=11.6$ *rad* / sec $\Rightarrow$ *poles* $: 0.984 \pm j0.0165 \Rightarrow d(z)=1-1.968z^{-1}+0.9685z^{-2}=0$
- $\tilde{G}(z)=\dfrac{-3.56\text{x}10^{-7}(z+3.554)(z+0.255)}{(z-1.065)(z-0.9393)(z-0.8187)}=\dfrac{-3.56\text{x}10^{-7}z^{-1}(1+3.554z^{-1})(1+0.255z^{-1})}{(1-1.065z^{-1})(1-0.9393z^{-1})(1-0.8187z^{-1})}$
- $a^{+s}(z)=(1-0.9393z^{-1})(1-0.8187z^{-1}); a^{-}(z)=1; a^{un+}(z)=(1-1.065z^{-1})$

  $b^{+s}(z)=-3.56\text{x}10^{-7}; b^{-}(z)=(1+3.554z^{-1})(1+0.255z^{-1}); b^{nm+}(z)=1$
- *Bezout (Aryabhatta, Diophantine) identity:*

  $(1-1.065z^{-1})d_1(z)+z^{-1}(1+3.554z^{-1})(1+0.255z^{-1})q_1(z)=1-1.968z^{-1}+0.9685z^{-2}=d(z)$
- Solve for $d_1(z)$ *and* $q_1(z) \Rightarrow d_1(z)=1-0.9042z^{-1}+0.001z^{-2}; q_1(z)=0.0012$
- $d_h(z)=b^{+s}(z)d_1(z)=-3.56\text{x}10^{-7}(1-0.9042z^{-1}+0.001z^{-2})$
- $q(z)=a^{+s}(z)q_1(z)=0.0012(1-0.9393z^{-1})(1-0.8187z^{-1})$
- Select $p_1(z)=1 \Rightarrow p(z)=a^{+s}(z) \Rightarrow T(z)=\dfrac{K_r b^{-}(z)b^{nm+}(z)z^{-1}}{d(z)} \Rightarrow K_r=\dfrac{0.9804d(1)}{b^{-}(1)b^{nm+}(1)}=8.577\text{x}10^{-5}$
- Alternately, $p(z)=q(z) \Rightarrow H(z)=\dfrac{q(z)}{d_h(z)}=\dfrac{-3.371\text{x}10^{3}(1-0.9393z^{-1})(1-0.8187z^{-1})}{(1-0.9042z^{-1}+0.001z^{-2})} \Rightarrow K_r=0.0846$

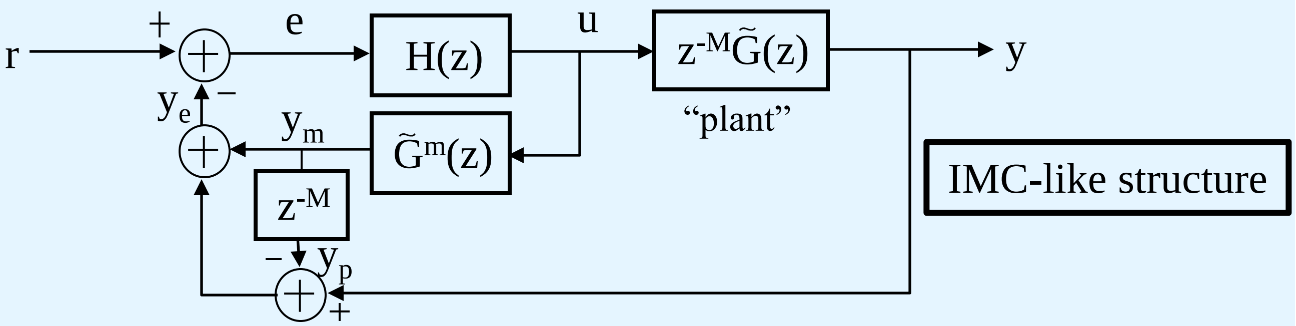
A Technique for Control of Systems with Time Delay, $\tau = Mh + \varepsilon$

$$\tilde{G}(z) \rightarrow z^{-M}\tilde{G}(z), \quad M = \text{integer}$$

(Will consider mods for "fractional" delay part $0 \le \varepsilon < h$ later.)

Smith Predictor/Compensator

- Design $H(z)$ using $\tilde{G}^m(z)$ = "model" of $\tilde{G}(z)$ (usually $\tilde{G}^m \equiv \tilde{G}$).
- Implementation:



$y_p(k)$ = "predicted" value of $y(k)$

$$y_p = z^{-M}\tilde{G}^m(z)u(z)$$

- Nominally $y(k) - y_p(k)$ = prediction error should be small
- Control is based primarily on $r - y_m$

$$y_m(z) = \tilde{G}^m(z)u(z) \sim M\text{ – step ahead prediction of } y$$

$$u(z) = H(z)\{r(z) - \underbrace{[y_m(z) + (y(z) - y_p(z))]}_{y_e \,\sim\, \text{"effective" output}}\}$$

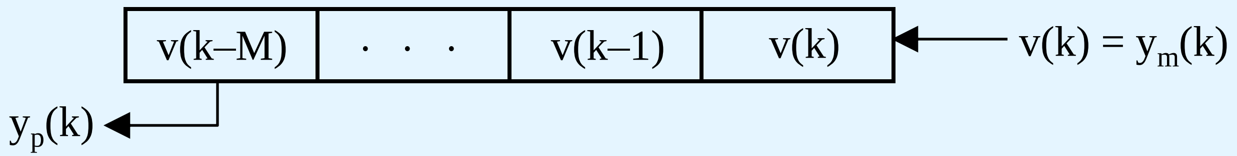
- Basic idea is to build a control that approximates

$$u(z) = H(z)z^{+M}[r(z) - y(z)] \qquad \text{(need to know/estimate future r if it is changing).}$$

Smith Compensator Application

- Model of system in feedback loop
  - Possible numerical problems if $\tilde{G}(z)$ is unstable
  - Initialize $\tilde{G}^m$ to rest condition ($\equiv 0$)

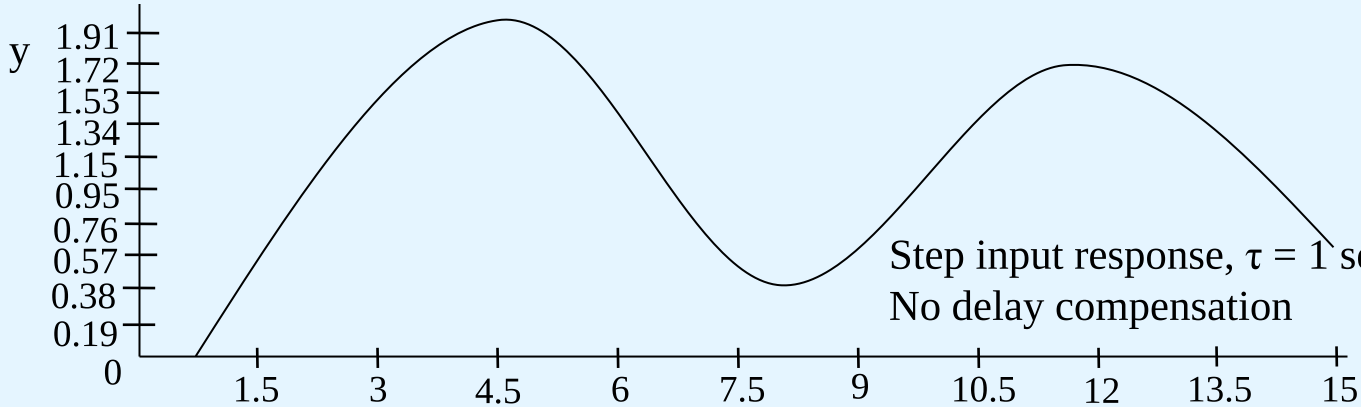
  Implement z^{-M} = M – step delay line by an (M+1) – dimensional push – down stack.



  - Initialize stack with $v(k-j) = y(k)$ for all j at k = 0
- Motor-positioning example with τ = 1sec, h = 1 sec (i.e., M = 1)

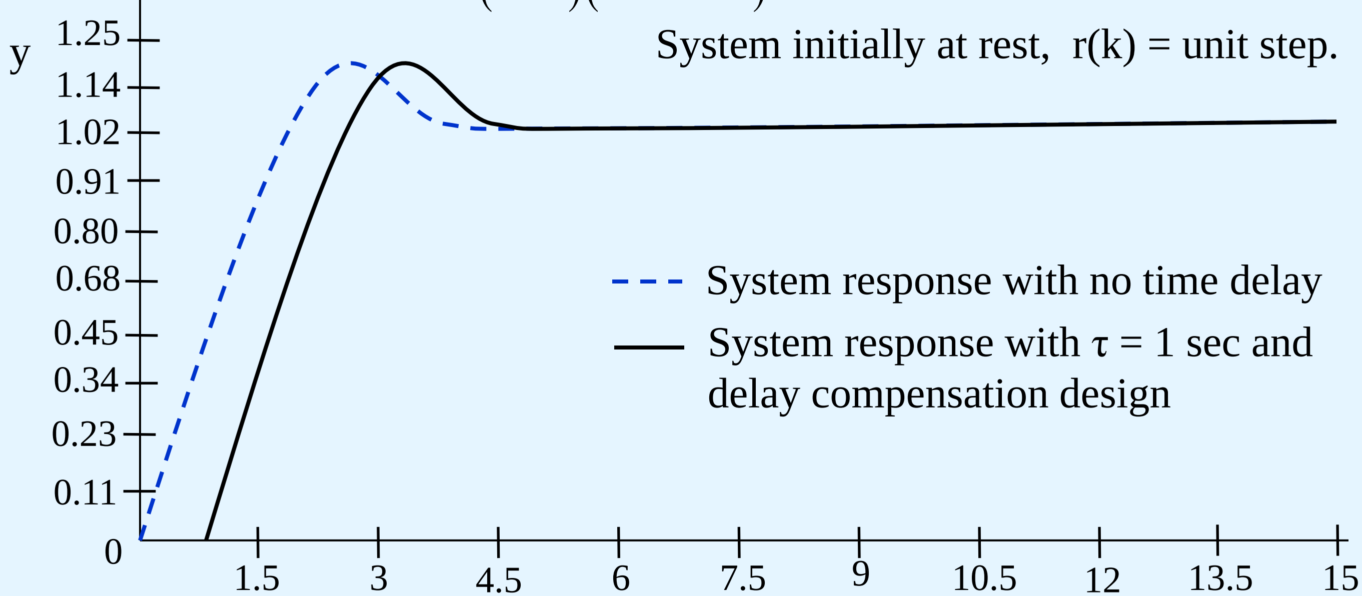
$$H(z) = 10.5\frac{z-0.87}{z+0.35} \quad \text{(from w-plane design)}$$

  - Recall $\phi_m \sim 56^o$, $\omega_c \sim 0.73 \Rightarrow \tau_{max} \sim 1.34$ sec, so expect poor performance with no delay compensation as ϕ_m would drop to ~ 14^o.



Results with Delay Compensation

- $M = 1,\ \tilde{G}^m(z) = \tilde{G}(z) = 0.048\,\dfrac{(z+0.97)}{(z-1)(z-0.905)}$



- CL response is identical to undelayed case, with a time-shift of M steps.
  - If system is not initially at rest, output response would "drift" for first M steps until the first control begins to affect response.
- As M increases the need for $G^m(z) \sim G(z)$ becomes more critical.

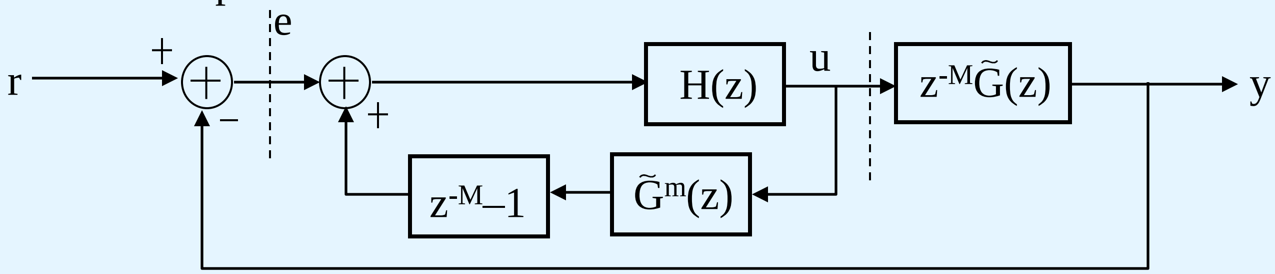
Modifications for non-integer $\tau = Mh + \varepsilon,\ \varepsilon \neq 0$

y_m = propagation of y through $\tilde{G}(z)$, remains unchanged.

y_p = prediction of current y(k). Obtain via model discussed in Lecture 4.

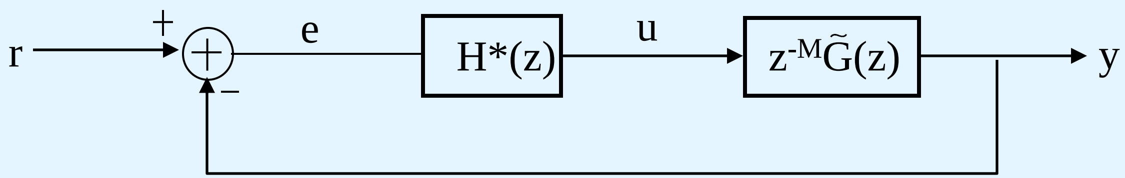
Alternate Implementation of Smith Compensator

- Consolidate FB loops



$$u(z) = H(z)\, e(z) - H(z)\tilde{G}^m(z)(1 - z^{-M})u(z)$$

- Consolidate inner loop, between e and u



$$H^*(z) = \frac{H(z)}{1 + H(z)\tilde{G}^m(z)(1 - z^{-M})}$$

- Typically, H*(z) will be a high-order compensator

  >> 1 – 2 usually associated with lag, lead, and PID.

  - Implementation methods are critical

    Speed/timing for real-time

    Accuracy

Implementation of High-Order Digital Compensators

$$H(z) = \frac{\beta_0 z^m + \beta_1 z^{m-1} + \cdots + \beta_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m}$$

- Direct form

$$u(k) = \beta_0 e(k) + \underbrace{\left[\beta_1 e(k-1) + \cdots + \beta_m e(k-m)\right]}_{SE} - \underbrace{\left[\alpha_1 u(k-1) + \cdots + \alpha_m u(k-m)\right]}_{SU}$$

  - SE and SU for time k: computed at step k–1
  - Needs storage of last m e(i) and u(i)
  - Very poor numerical properties!
    - Small changes in α_i, β_i coefficients (especially α_m, β_m) can cause large changes in roots = poles and zeros of H(z).
    - Errors in e(k), u(k) "hang around" for m steps
- Decomposition Approach
  - Decompose H(z) into a sum of low-order subparts (e.g., as in PID) and then add up parts

$$H(z) = \beta_0 + \frac{\tilde{\beta}_1 z^{m-1} + \cdots + \tilde{\beta}_m}{z^m + \alpha_1 z^{m-1} + \cdots + \alpha_m} \quad ; \qquad \tilde{\beta}_i = \beta_i - \beta_0 \alpha_i$$

PF expansion (assume no repeated roots):

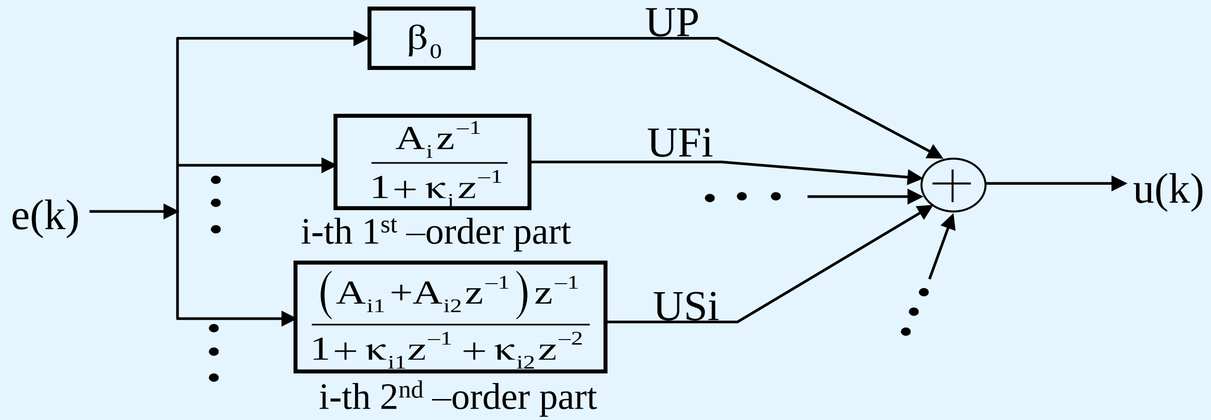
$$H(z) = \beta_0 + \underbrace{\sum_{i=1}^{N_f} \frac{A_i}{z + \kappa_i}}_{N_f \text{ First-order Factors}} + \underbrace{\sum_{i=1}^{N_s} \frac{A_{i1} z + A_{i2}}{z^2 + \kappa_{i1} z + \kappa_{i2}}}_{N_s \text{ Second-order Factors}}$$

Implementation Structure of H(z)

$$H(z) = \beta_0 + \sum_{i=1}^{N_f} \frac{A_i z^{-1}}{1 + \kappa_i z^{-1}} + \sum_{i=1}^{N_s} \frac{\left(A_{i1} + A_{i2} z^{-1}\right) z^{-1}}{1 + \kappa_{i1} z^{-1} + \kappa_{i2} z^{-2}}$$

Note 1 –step delay in all first, second-order parts => can compute these at step k – 1 for use at time k.

Structure



- Algorithm (initiate R, e_1, USi_1, USi_2, UFi_1 = 0)

obtain e(k) = e = r(k) – y(k)

$U = \beta_0 e + R$

output u(k) U

Do for each 1st-order part:
$\left\{ \begin{array}{l} UFi = A_i e_1 - \kappa_i UFi_1 \\ UFi_1 = UFi \end{array} \right\}$ Obtain next value of UFi save it for next time

Do for each 2nd-order part:
$\left\{ \begin{array}{l} USi = A_{i1} e_1 + A_{i2} e_2 - \kappa_{i1} USi_1 - \kappa_{i2} USi_2 \\ USi_2 = USi_1 \\ USi_1 = USi \end{array} \right\}$ Obtain next USi, save last two values of USi

$$R = \sum_{i=1}^{N_f} UFi + \sum_{i=1}^{N_s} USi$$

=> $e_2 = e_1$

$e_1 = e$

- Include in Cntrl subroutine, OPT = 3.

Summary of Compensator Design Methods

- Indirect design $H(s) \rightarrow \tilde{H}(z)$ by discrete equivalent
  - Generally requires small h
  - Easy and straightforward
- Direct design methods
  - Root locus, w – plane, PID
  - Only have Nyquist restriction on h

  => <u>Advantages</u>

  - Generally easy to design H(z)
  - A low-order design, easily realized, is found
  - Higher order dynamics in G(s) accommodated with little extra dffort
  - Universally used techniques, time-tested

  => <u>Disadvantages</u>

  - Low-order compensator designs do not always work
  - Does not use all available information about system behavior (e.g., y instead of $\underline{x}$)
  - Measures used are not 1:1 with time response (requires trial and error with CL simulation)
  - Limited by human insight
  - Extremely difficult for MIMO systems