بسم ا... الرحمن الرحیم

سیستم های کنترل دیجیتال

Digital Control Systems

دکتر محمدرضا رمضانی

مراجع

1. K. Ogata, Discrete-time control systems, Prentice Hall, 1995

2- اسلایدهای درس کنترل دیجیتال دانشگاه علم و صنعت دکتر بلندی و دکتر اسمعیل زاده

- سرفصل مطالب
- ساختار و تحقق کنترل کننده های دیجیتال
- تحقق های مستقیم، استاندارد، سری، موازی و نردبانی
- ارتباط بین صفحه مختلط S و Z
- تحلیل پایداری سیستم های حلقه بسته در صفحه مختلط Z

Structure of the Digital Controller

Discrete-Time Control System, Ogata Ch 3

Why are we doing this?

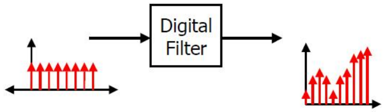
- We will obtain the block diagram structure of the pulse-transfer function in different forms.
- We will also look at the Impulse Response form of the block diagram realization.

Realization of Digital Controller

- Realization means to implement with software (algorithm) or with hardware circuit.
- We will look at the computation components (adders, multipliers, shift registers) of the pulse transfer function.

Realization of Digital Controller

- In digital signal processing (DSP) field, a digital filter is a computational algorithm that converts sequence of numbers into another sequence of numbers.



Realization of Digital Controller

- The signal processing applications can be done off-line, while the control application must be done in real-time.
- We will now look at the block-diagram realization of digital filters.

Realization of Digital Controller

- Once the block-diagram realization is completed, the physical realization in hardware or software is straight forward.

Realization of Digital Controller

- The general form of the pulse transfer function between the output *Y(z)* and input *X(z)* is given by

$$G(z)=\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_mz^{-m}}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}}, \qquad n\geq m$$

Realization of Digital Controller

- The a's and b's are real coefficients of the pulse transfer function.

- Some of the coefficients may be zero.

Block-diagram realization

- The block-diagram structure that has the coefficients a and b appear directly as multiplier is called *direct structure*.

- The direct structure
  - Direct Programming
  - Standard Programming

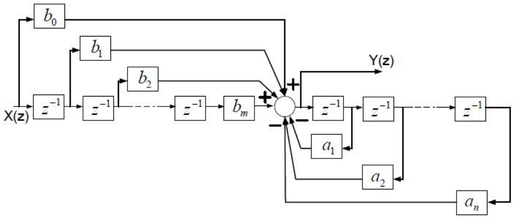
Direct Programming

- Rewrite the pulse transfer function as

$$\begin{aligned}Y(z)=-a_1z^{-1}Y(z)-a_2z^{-2}Y(z)-\cdots-a_nz^{-n}Y(z)\\+b_0X(z)+b_1z^{-1}X(z)+\cdots+b_mz^{-m}X(z)\end{aligned}$$

Direct Programming

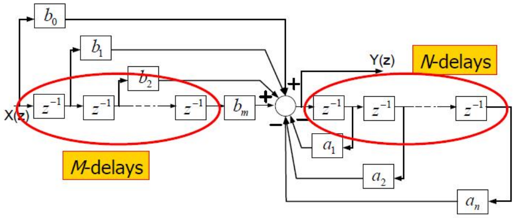
- The structure is



Direct Programming

- The direct programming uses separate set of delay elements for the forward and the feedback path.
- The total number of the delay elements is *n+m*.

Direct Programming



Direct Programming

- In real implementation, a delay elements is the memory buffer that can hold a value. It was very expensive.
- The Standard Programming method minimize the numbers of delay elements to *N* delay elements.

Standard Programming

- Rewriting the pulse transfer function as

$$G(z)=\frac{Y(z)}{X(z)}=\frac{Y(z)}{H(z)}\frac{H(z)}{X(z)}$$

$$=b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_mz^{-m}\frac{1}{\left(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}\right)}$$

$$\frac{Y(z)}{H(z)} \qquad \frac{H(z)}{X(z)}$$

Standard Programming

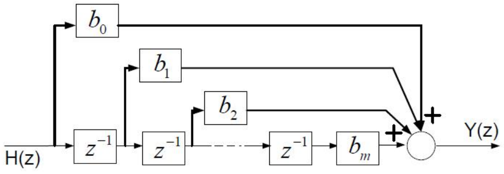
- Write the block diagram for the first portion as

$$\frac{Y(z)}{H(z)}=b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_mz^{-m}$$

$$Y(z)=\left(b_0+b_1z^{-1}+b_2z^{-2}+\cdots+b_mz^{-m}\right)H(z)$$

Standard Programming

$$Y(z)=b_0H(z)+b_1z^{-1}H(z)+b_2z^{-2}H(z)+\cdots+b_mz^{-m}H(z)$$



Standard Programming

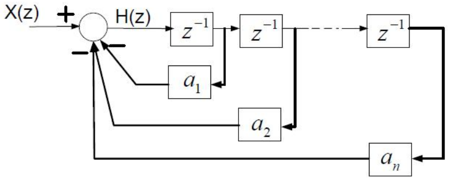
- The second part is

$$\frac{H(z)}{X(z)}=\frac{1}{1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}}$$

$$H(z)\left(1+a_1z^{-1}+a_2z^{-2}+\cdots+a_nz^{-n}\right)=X(z)$$

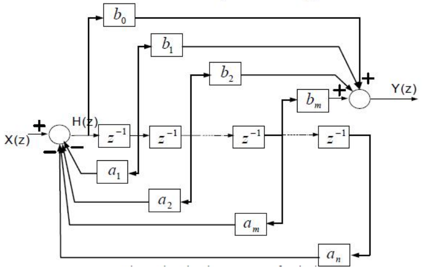
Standard Programming

$$H(z) = X(z) - a_1 z^{-1} H(z) - a_2 z^{-2} H(z) - \cdots - a_n z^{-n} H(z)$$



Standard Programming

- Combine the two parts together



Standard Programming

- Again, the coefficients a's and b's are the multipliers in the block diagram.
- The Standard Programming uses only n delay elements.
- But required 2 summing elements.

Accuracy

- There are three sources of errors affect the accuracy
  - The error due to quantization of the input signal (quantization noise)
  - The error due to accumulation of round-off error in the arithmetic operation
  - The error due to quantization of the coefficients a_i and b_i

Accuracy

- These three errors are due to the finite bits (8-bit) that represent signal samples and coefficients.

- The last source of error is larger as the order of the pulse transfer function increases.

Accuracy

- For higher order impulse transfer function Direct Programming Structure, the small errors in coefficients a_i and b_i causes large errors in the pole locations and zero locations.

Accuracy

- We reduce the order of the impulse response function to reduce the sensitivity to coefficient inaccuracies.

- The approaches are
  - Series Programming
  - Parallel Programming
  - Ladder Programming

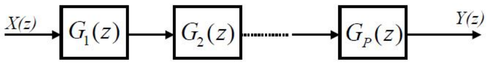
Accuracy

- The first approach is the Series Programming structure.

Series Programming

- Break down the pulse transfer function into series connection.

$$G(z) = G_1(z)G_2(z)\cdots G_P(z)$$



Series Programming

- We try to group the poles and zeros of G(z) into first-order and second-order sub-transfer function.

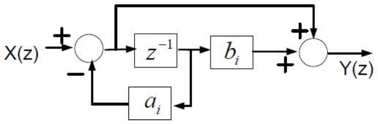
$$G(z) = G_1(z)G_2(z)\cdots G_p(z)$$

$$= \underbrace{\prod_{i=1}^{j} \frac{1+b_i z^{-1}}{1+a_i z^{-1}}}_{\text{1st-order terms}} \underbrace{\prod_{i=j+1}^{p} \frac{1+e_i z^{-1}+f_i z^{-2}}{1+c_i z^{-1}+d_i z^{-2}}}_{\text{2nd-order terms}}$$

Series Programming

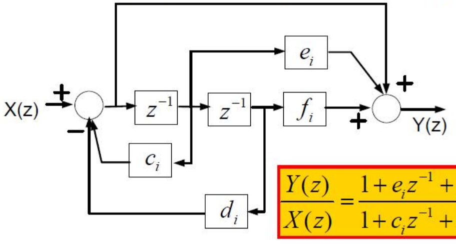
- The first-order term block-diagram is

$$\frac{Y(z)}{X(z)} = \frac{1+b_i z^{-1}}{1+a_i z^{-1}}$$



Series Programming

- The second-order term block-diagram is



$$\frac{Y(z)}{X(z)} = \frac{1+e_i z^{-1}+f_i z^{-2}}{1+c_i z^{-1}+d_i z^{-2}}$$

Accuracy

- The second approach is the Parallel Programming structure.

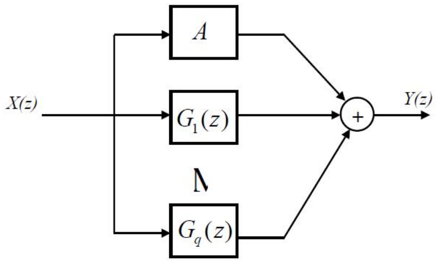
Parallel Programming

- The parallel programming method applies the partial fraction expansion method to G(z).

$$G(z) = A + G_1(z) + G_2(z) + \cdots + G_q(z)$$

constant (DC gain)

Parallel Programming



Parallel Programming

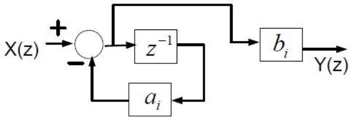
- This structure allows the 1st order terms and 2nd order terms to be a little simpler.

$$\begin{aligned} G(z) &= A + G_1(z) + G_2(z) + \cdots + G_q(z) \\ &= A + \sum_{i=1}^{j} G_i(z) + \sum_{i=j+1}^{q} G_i(z) \\ &= A + \sum_{i=1}^{j} \frac{b_i}{1 + a_i z^{-1}} + \sum_{i=j+1}^{q} \frac{e_i + f_i z^{-1}}{1 + c_i z^{-1} + d_i z^{-2}} \end{aligned}$$

Parallel Programming

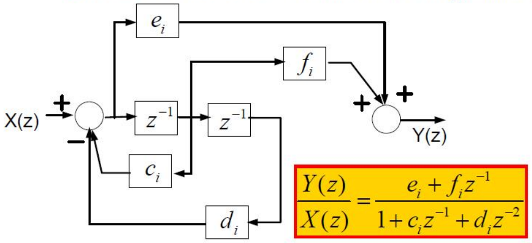
- The first-order term block-diagram is

$$\frac{Y(z)}{X(z)} = \frac{b_i}{1 + a_i z^{-1}}$$



Parallel Programming

- The second-order term block-diagram is



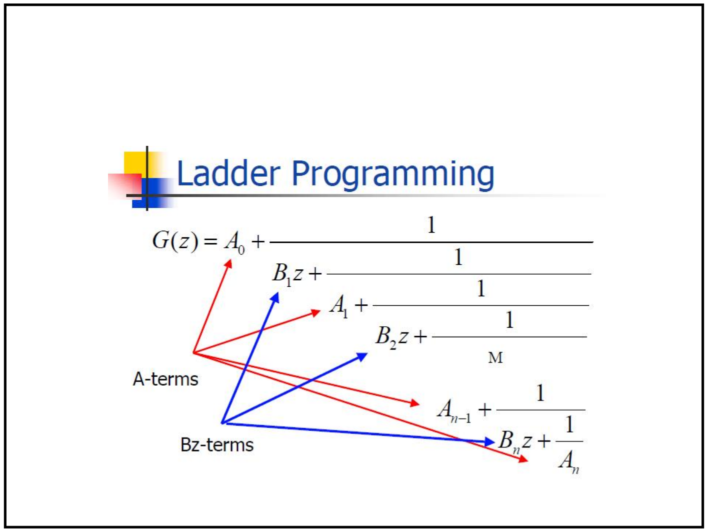
$$\boxed{\frac{Y(z)}{X(z)} = \frac{e_i + f_i z^{-1}}{1 + c_i z^{-1} + d_i z^{-2}}}$$

Accuracy

- The third approach is the Ladder Programming structure.

Ladder Programming

- The Ladder Programming method expands the pulse transfer function in continued-fraction form.



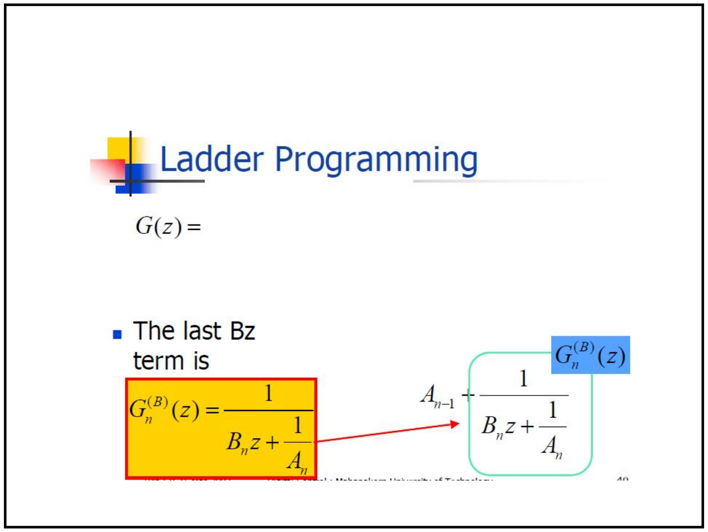
Ladder Programming

$$G(z) = A_0 + \cfrac{1}{B_1 z + \cfrac{1}{A_1 + \cfrac{1}{B_2 z + \cfrac{1}{\vdots}}}}$$

$$A_{n-1} + \cfrac{1}{B_n z + \cfrac{1}{A_n}}$$

A-terms

Bz-terms

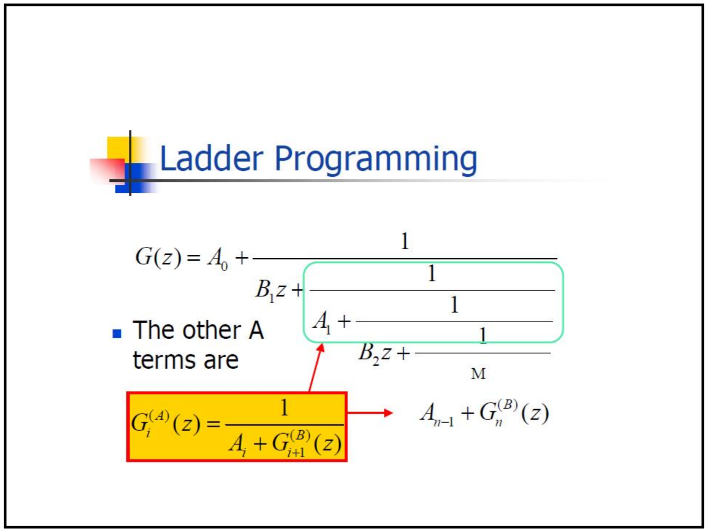


Ladder Programming

$G(z) =$

- The last Bz term is

$$G_n^{(B)}(z) = \cfrac{1}{B_n z + \cfrac{1}{A_n}} \rightarrow A_{n-1} + \underbrace{\cfrac{1}{B_n z + \cfrac{1}{A_n}}}_{G_n^{(B)}(z)}$$

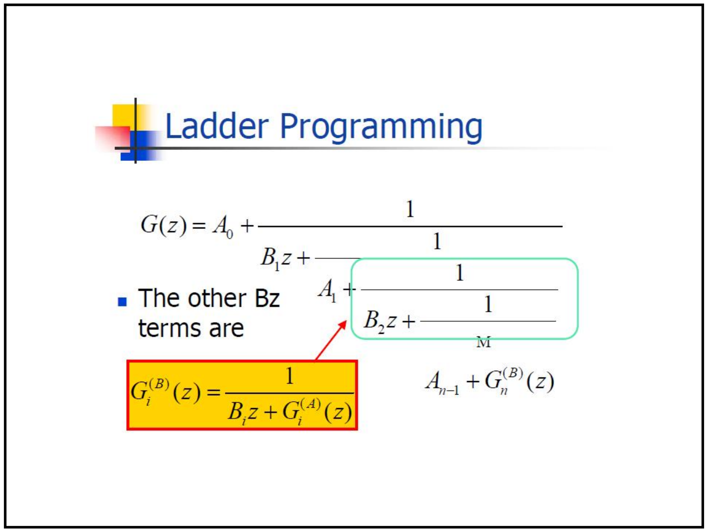


Ladder Programming

$$G(z) = A_0 + \cfrac{1}{B_1 z + \cfrac{1}{A_1 + \cfrac{1}{B_2 z + \cfrac{1}{\vdots}}}}$$

- The other A terms are

$$G_i^{(A)}(z) = \frac{1}{A_i + G_{i+1}^{(B)}(z)} \rightarrow A_{n-1} + G_n^{(B)}(z)$$



Ladder Programming

$$G(z) = A_0 + \cfrac{1}{B_1 z + \cfrac{1}{A_1 + \cfrac{1}{B_2 z + \cfrac{1}{\vdots}}}}$$

- The other Bz terms are

$$G_i^{(B)}(z) = \frac{1}{B_i z + G_i^{(A)}(z)}$$

$$A_{n-1} + G_n^{(B)}(z)$$

Ladder Programming

- Starts from

$$G(z) = A_0 + \cfrac{1}{B_1 z + \cfrac{1}{A_1 + \cfrac{1}{B_2 z + \cfrac{1}{\begin{matrix}\vdots \\ A_{n-1} + \cfrac{1}{B_n z + \cfrac{1}{A_n}}\end{matrix}}}}}$$

$$G(z) = A_0 + G_1^{(B)}(z)$$

Ladder Programming

- Starts from

$$G(z) = A_0 + G_1^{(B)}(z)$$

$$G(z) = A_0 + G_1^{(B)}(z)$$

Ladder Programming

- Continuing

$$G(z) = A_0 + \cfrac{1}{B_1 z + \cfrac{1}{A_1 + \cfrac{1}{B_2 z + \cfrac{1}{\begin{matrix}\vdots \\ A_{n-1} + \cfrac{1}{B_n z + \cfrac{1}{A_n}}\end{matrix}}}}}$$

$$G(z) = A_0 + G_1^{(B)}(z)$$

$$G(z) = A_0 + \frac{1}{B_1 z + G_1^{(A)}(z)}$$

Ladder Programming

- Continuing

$$G(z) = A_0 + \frac{1}{B_1 z + G_1^{(A)}(z)}$$

$$G(z) = A_0 + G_1^{(B)}(z)$$

$$G(z) = A_0 + \frac{1}{B_1 z + G_1^{(A)}(z)}$$

Ladder Programming

- Continuing

$$G(z) = A_0 + G_1^{(B)}(z)$$

$$G(z) = A_0 + \frac{1}{B_1 z + G_1^{(A)}(z)}$$

$$G(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + G_2^{(B)}(z)}}$$

$$G(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + \boxed{\frac{1}{B_2 z + \frac{1}{\begin{matrix}\mathrm{M} \\ A_{n-1} + \frac{1}{B_n z + \frac{1}{A_n}}\end{matrix}}}}}}$$

Ladder Programming

- Continuing

$$G(z) = A_0 + G_1^{(B)}(z)$$

$$G(z) = A_0 + \frac{1}{B_1 z + G_1^{(A)}(z)}$$

$$G(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + G_2^{(B)}(z)}}$$

$$G(z) = A_0 + \frac{1}{B_1 z + \frac{1}{A_1 + G_2^{(B)}(z)}}$$

Ladder Programming

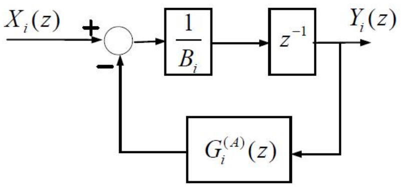
- In summary

$$G_i^{(B)}(z) = \frac{1}{B_i z + G_i^{(A)}(z)}$$

$$G_i^{(A)}(z) = \frac{1}{A_i + G_{i+1}^{(B)}(z)}$$

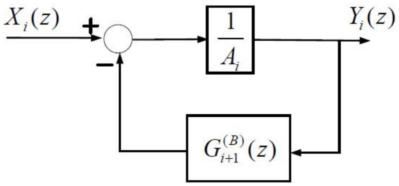
Ladder Programming

$$G_i^{(B)}(z) = \frac{Y_i(z)}{X_i(z)} = \frac{1}{B_i z + G_i^{(A)}(z)}$$



Ladder Programming

$$G_i^{(A)}(z) = \frac{Y_i(z)}{X_i(z)} = \frac{1}{A_i + G_{i+1}^{(B)}(z)}$$



Ladder Programming

- And for the last $G_i^{(A)}(z)$

$$G_n^{(A)}(z) = \frac{1}{A_n}$$

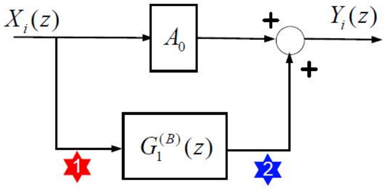
Ladder Programming

- Begin from

$$G(z) = A_0 + G_1^{(B)}(z)$$

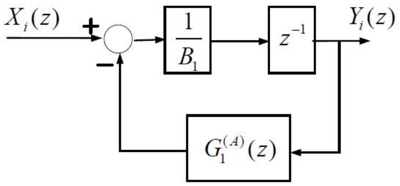
Ladder Programming

$$G(z) = A_0 + G_1^{(B)}(z)$$



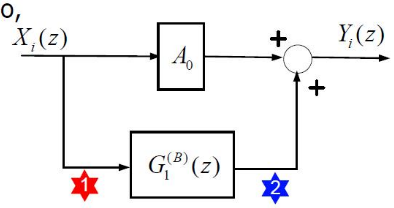
Ladder Programming

- We know that $G_1^{(B)}(z) = \dfrac{1}{B_1 z + G_1^{(A)}(z)}$

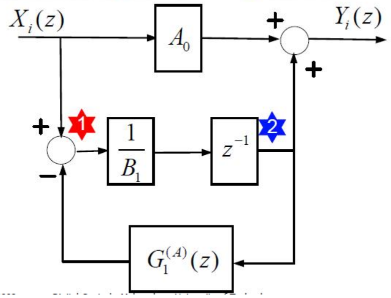


Ladder Programming

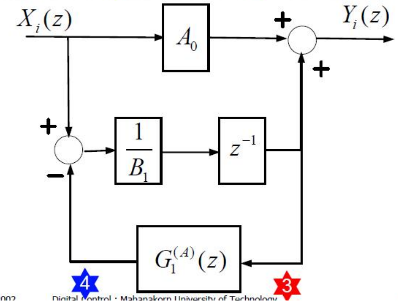
- So,

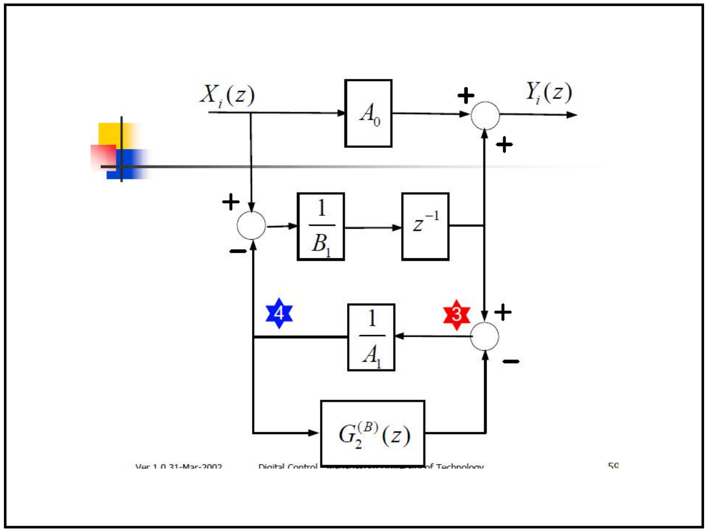


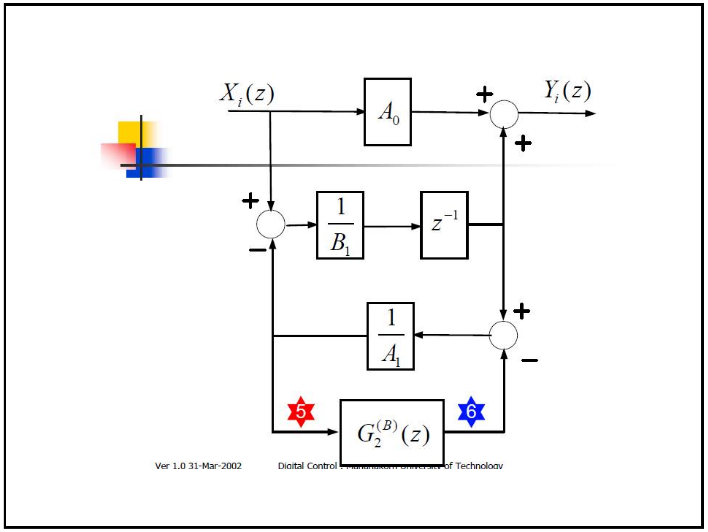
Ladder Programming

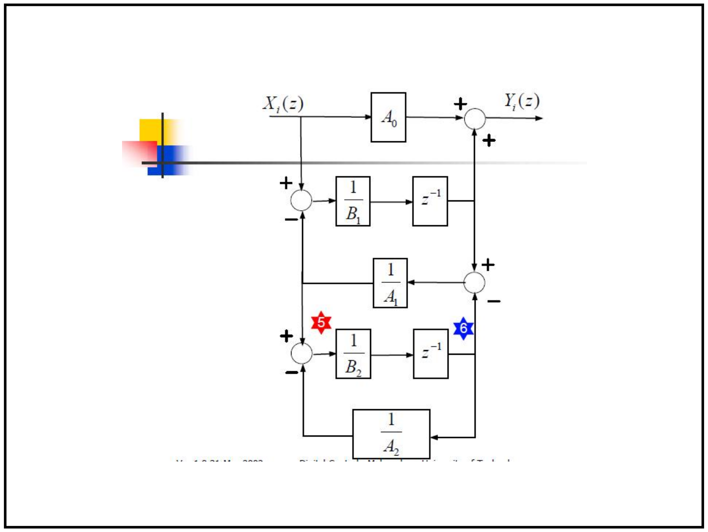


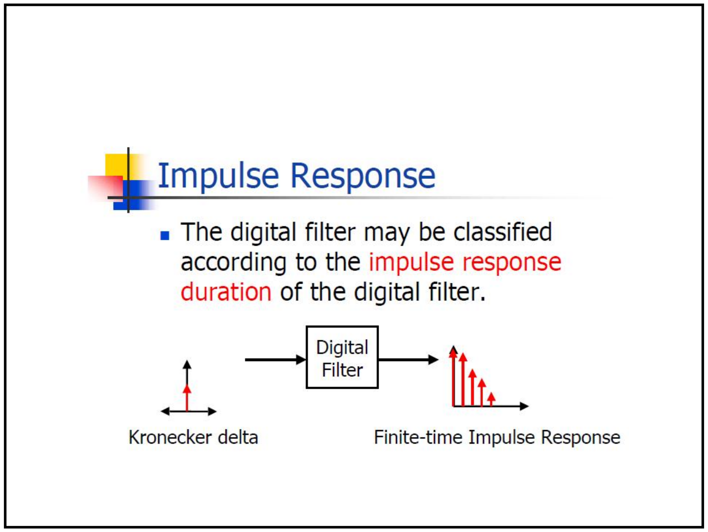
Ladder Programming











Impulse Response

- The digital filter may be classified according to the impulse response duration of the digital filter.

Infinite Impulse Response (IIR)

- If the digital filter model is

$$\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+\mathrm{K}\ +b_mz^{-m}}{1+a_1z^{-1}+a_2z^{-2}\mathrm{K}\ +a_nz^{-n}},\quad n\geq m$$

Infinite Impulse Response (IIR)

- The output sample is

$$\begin{aligned}y(k)&=-a_1y(k-1)-a_2y(k-2)-\mathrm{K}\ -a_ny(k-n)\\&+b_0x(k)+b_1x(k-1)+\mathrm{K}\ +b_mx(k-m)\end{aligned}$$

- We can see that if all a_i's are not zero, the output will recursively generate; *infinite-impulse response filter*.

Finite Impulse Response (FIR)

- If all the a_i's are zero.

$$\frac{Y(z)}{X(z)}=b_0+b_1z^{-1}+\mathrm{K}\ +b_mz^{-m}$$

- The output sample is

$$y(k)=b_0x(k)+b_1x(k-1)+\mathrm{K}\ +b_mx(k-m)$$

Finite Impulse Response (FIR)

- The interval of the impulse response of the filter is limited to a finite number; *finite impulse response*.

Mapping between S plane and z plane

Why are we doing this?

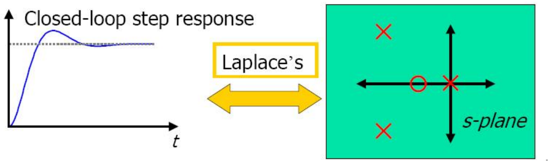
- We will show the mapping between the design in s-plane and the z-plane.

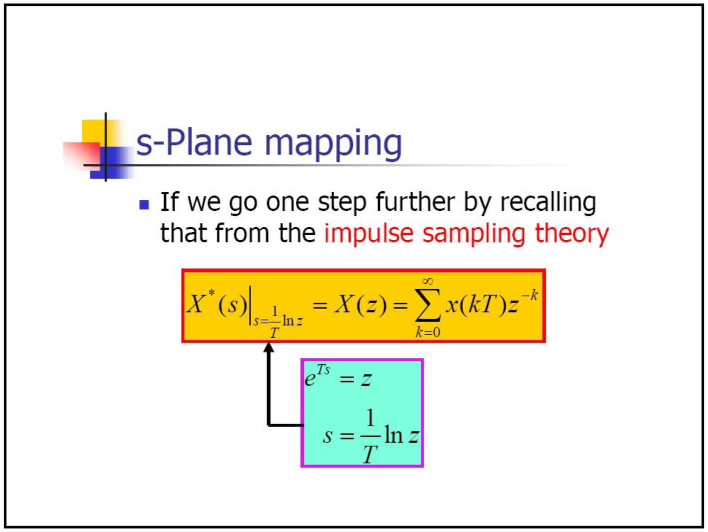
s-Plane design

- The design specifications usually written in time-domain parameters
  - Rise-time
  - Peak-time
  - Settling-time
  - Percentage Overshoot
  - Steady-state gain
  - Etc.

s-Plane design

- We can translate the time-domain specifications into the location of closed-loop poles and zeros in s-plane.





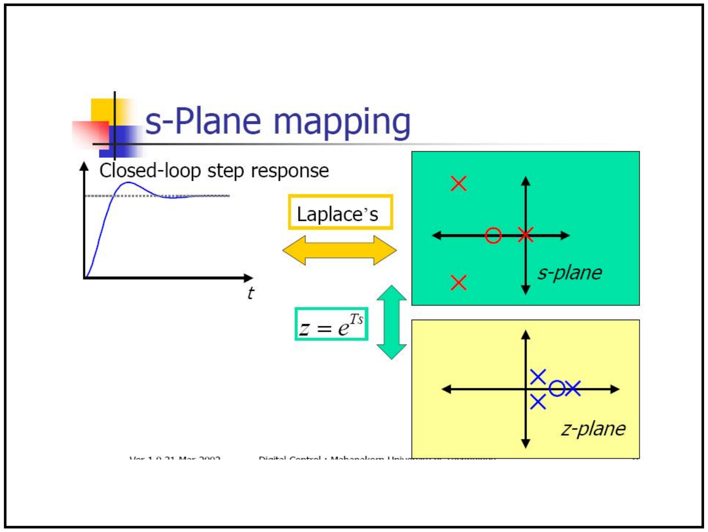
s-Plane mapping

- If we go one step further by recalling that from the impulse sampling theory

$$X^*(s)\big|_{s=\frac{1}{T}\ln z} = X(z) = \sum_{k=0}^{\infty} x(kT)z^{-k}$$

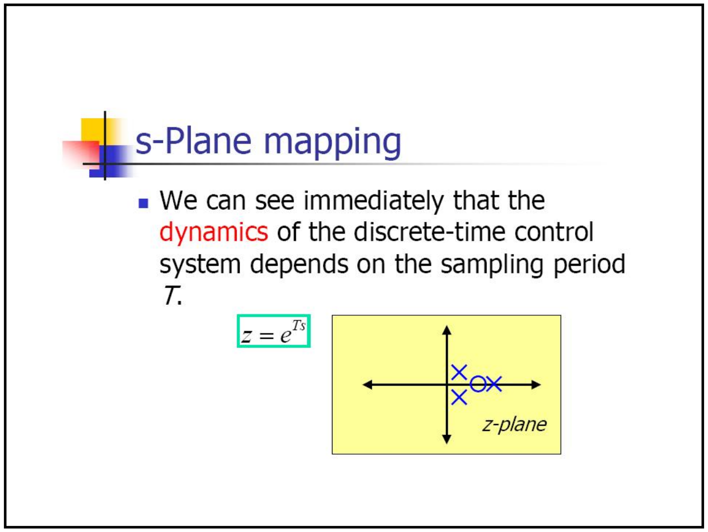
$$e^{Ts} = z$$

$$s = \frac{1}{T}\ln z$$



s-Plane mapping

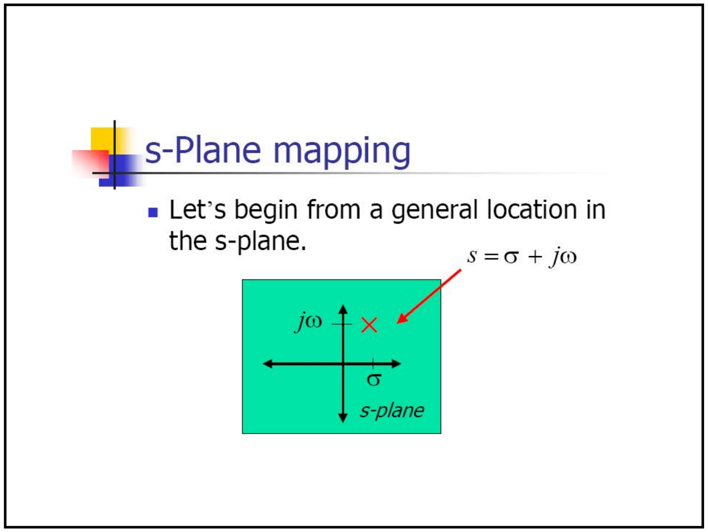
$z = e^{Ts}$



s-Plane mapping

- We can see immediately that the dynamics of the discrete-time control system depends on the sampling period *T*.

$z = e^{Ts}$



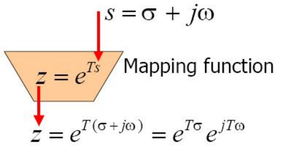
s-Plane mapping

- Let's begin from a general location in the s-plane.

$s = \sigma + j\omega$

s-Plane mapping

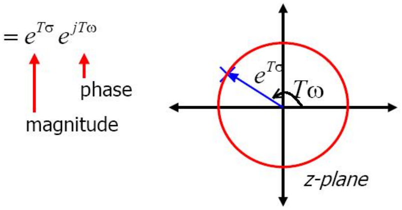
- Input that s-plane location to the mapping function.



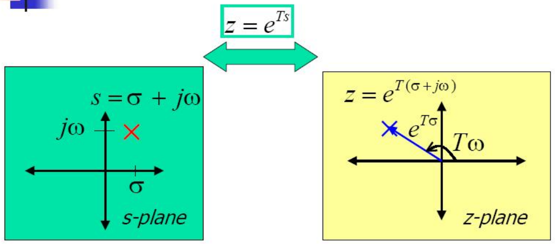
s-Plane mapping

- The result of the mapping can be view as a complex-plane vector $Ae^{j\theta}$

$$z=e^{T(\sigma+j\omega)}=e^{T\sigma}e^{jT\omega}$$



s-Plane mapping



s-Plane mapping

- Note that the angular term is cycle every 2π

$$z=e^{T\sigma}e^{jT\omega}=e^{T\sigma}e^{j(T\omega+2\pi k)}$$

s-Plane mapping

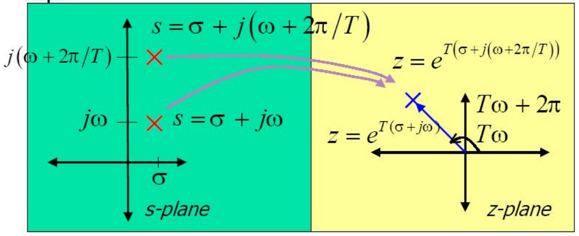
- So, the poles and zeros from location

$$s = \sigma + j\omega$$

- Will fall in the same spot in z-plane as the poles/zeros from

$$s = \sigma + j\left(\omega + 2\pi k/T\right)$$

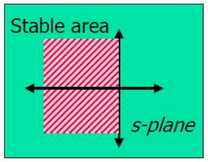
s-Plane mapping



s-Plane mapping

- In designing a system, we must make the closed-loop system stable, i.e., the closed-loop poles are in the left-half of the s-plane.
- The left-half of the s-plane is negative value of(σ for any value of¯ω

s-Plane mapping



$$z = e^{T(\sigma + j\omega)}$$

$$= \underset{\substack{\uparrow \\ \text{negative}}}{e^{T\sigma}}\, e^{jT\omega}$$

- Exponential of negative value is always less than 1.

$$|z| = e^{T\sigma} < 1$$

s-Plane mapping

- The jω axis (σ =0) in s-plane maps into the unit circle in z-plane.

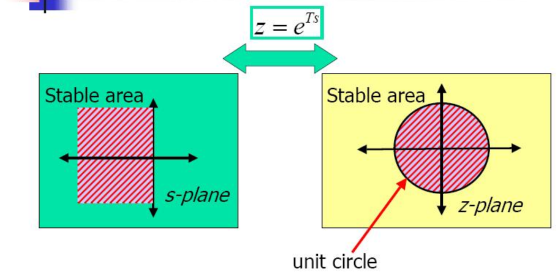
$$z = e^{T(j\omega)}$$

$$= 1 \cdot e^{jT\omega}$$

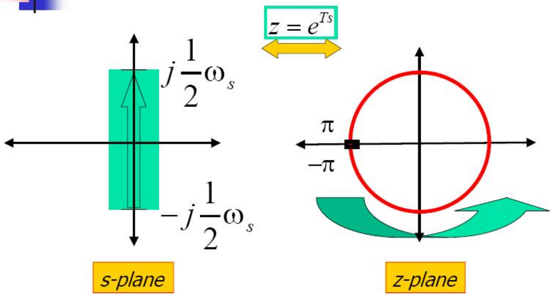
circle

unit amplitude

s-Plane mapping

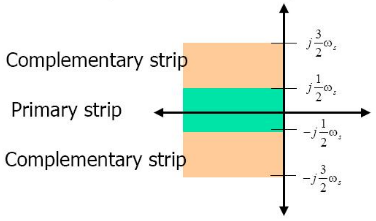


s-Plane mapping



s-Plane mapping

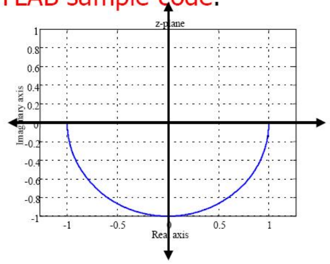
- This defines the primary strip in the s-plane which maps into the z-plane.



s-Plane mapping

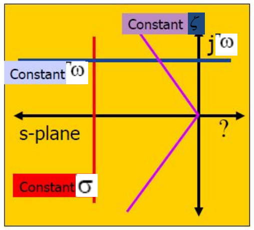
- Here is a MATLAB sample code.

```
>> T=0.20;
>> ws=2*pi*(1/T);
>> w=-1/2*ws:0.01:0;
>> s=j*w;
>> z=exp(T*s);
>> plot(z);
>> axis('equal');
```



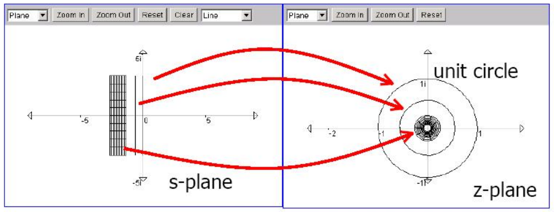
s-Plane mapping

- More complex s-plane mappings are
  - Constant σ (constant settling time)
  - Constant ω
  - Constant damping ratio ζ



s-Plane mapping : Constant σ

- The further to the left in s-plane, the smaller the circle in z-plane.



s-Plane mapping : Constant ω

- The different in ω (imaginary-part location) will only effect the angle of the mapped vector.

$$z = e^{T(\sigma + j\omega)}$$

$$= e^{T\sigma} e^{jT\omega}$$ ← Change angle

s-Plane mapping : Constant ζ

- The changes in damping ratio ζ will effect both |z| and angle of z.

$$s = -\zeta\omega_n + j\omega_n\sqrt{1-\zeta^2} = -\zeta\omega_n + j\omega_d$$

$$\omega_d = \omega_n\sqrt{1-\zeta^2}$$

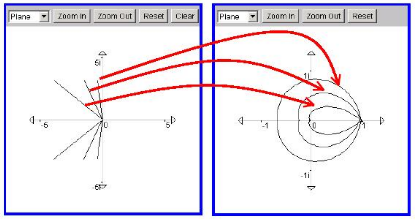
یک خط با نسبت میرایی ثابت در صفحه z :

$$z = e^{Ts} = \exp(-\zeta\omega_n T + j\omega_d T)$$

$$|z| = \exp(-\frac{2\pi\zeta}{\sqrt{1-\zeta^2}}\frac{\omega_d}{\omega_s}) \qquad \angle z = 2\pi\frac{\omega_d}{\omega_s}$$

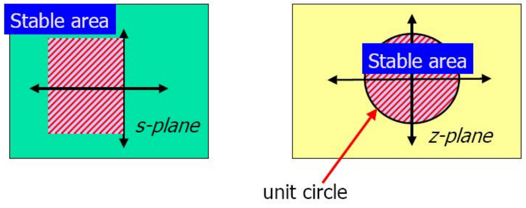
s-Plane mapping : Constant ζ

- The less damped system in s-plane, the wider the spiral curve in z-plane.

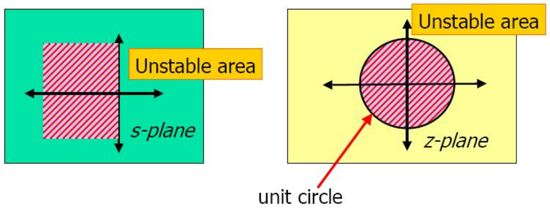


Conclusion on s-plane mapping

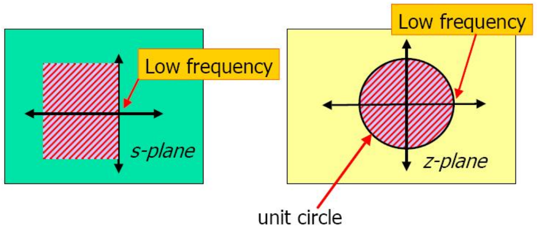
- For the mapping via $z = e^{Ts}$



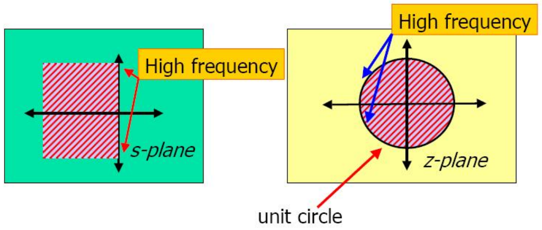
Conclusion on s-plane mapping



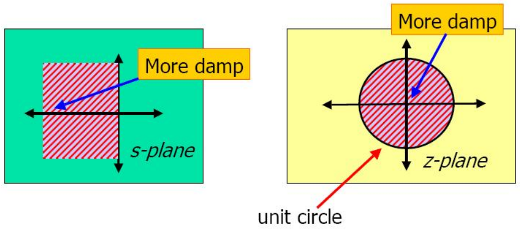
Conclusion on s-plane mapping



Conclusion on s-plane mapping



Conclusion on s-plane mapping



---

s-Plane mapping

- Go back to our Integral Approximation we did for the digital PID design.
- Let's take, for example, a differential equation or transfer function

$$H(s) = \frac{U(s)}{E(s)} = \frac{a}{s+a}$$

s-Plane mapping

- This transfer function can be written in the differential equation form as

$$\frac{U(s)}{E(s)} = \frac{a}{s+a}$$

$$(s+a)U(s) = aE(s)$$

$$\dot{u}(t) + au(t) = ae(t)$$

s-Plane mapping

- The output signal $u(t)$ is

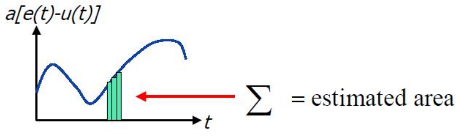
$$\dot{u}(t) + au(t) = ae(t)$$

$$u(t) = \int_0^t a\left[e(\tau) - u(\tau)\right]d\tau$$

- We have different methods in approximating this integral.

s-Plane mapping

$$u(t) = \int_0^t \boxed{a\left[e(\tau) - u(\tau)\right]}d\tau$$



s-Plane mapping

- Forward Rectangular rule

$$u(kT) = u\left((k-1)T\right) + aT\left[e\left((k-1)T\right) - u\left((k-1)T\right)\right]$$

$$= (1-aT)u\left((k-1)T\right) + aTe\left((k-1)T\right)$$

$$U(z) = (1-aT)z^{-1}U(z) + aTz^{-1}E(z)$$

$$\frac{U(z)}{E(z)} = \frac{aTz^{-1}}{1-(1-aT)z^{-1}} = \frac{a}{\left(\frac{z-1}{T}\right)+a}$$

s-Plane mapping

- Backward Rectangular rule

$$u(kT) = u\left((k-1)T\right) + aT\left[e(kT) - u(kT)\right]$$

$$= u\left((k-1)T\right) + aTe(kT) - aTu(kT)$$

$$(1+aT)U(z) = z^{-1}U(z) + aTE(z)$$

$$\frac{U(z)}{E(z)} = \frac{aTz}{z(1+aT)-1} = \frac{a}{\left(\frac{z-1}{Tz}\right)+a}$$

---

s-Plane mapping

- Trapezoid rule

$$\frac{U(z)}{E(z)} = \frac{a}{\left(\frac{2}{T}\right)\left(\frac{z-1}{z+1}\right)+a}$$

- HW : Prove this !

	$H(s)$	$H(z)$
FW rule	$\frac{a}{s+a}$	$\frac{a}{\left(\frac{z-1}{T}\right)+a}$
BW rule	$\frac{a}{s+a}$	$\frac{a}{\left(\frac{z-1}{Tz}\right)+a}$
Trapezoid rule	$\frac{a}{s+a}$	$\frac{a}{\left(\frac{2}{T}\right)\left(\frac{z-1}{z+1}\right)+a}$

---

s-Plane mapping

FW rule	$s \Leftrightarrow \left(\frac{z-1}{T}\right)$
BW rule	$s \Leftrightarrow \left(\frac{z-1}{Tz}\right)$
Trapezoid rule	$s \Leftrightarrow \left(\frac{2}{T}\right)\left(\frac{z-1}{z+1}\right)$

s-Plane mapping

- We can use the same method of mapping the s-plane into the z-plane as previous.
- For example, the forward rectangular rule mapping

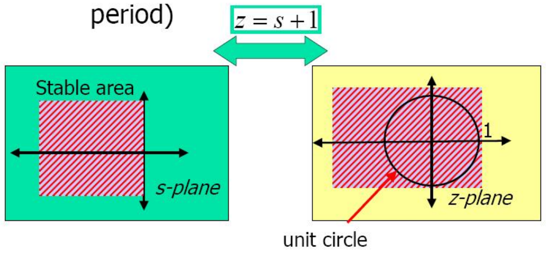
$$s \Leftrightarrow \left(\frac{z-1}{T}\right)$$

$$z \Leftrightarrow Ts+1$$

s-Plane mapping

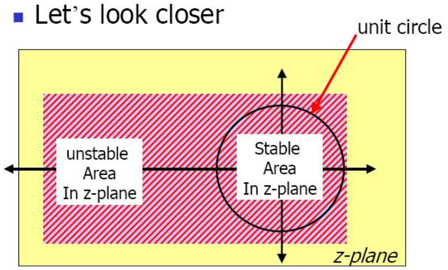
- If we let T=1 (normalized sampling period)

$$\boxed{z = s+1}$$



s-Plane mapping

- Let's look closer



s-Plane mapping

- With this mapping, the stable area in s-plane may falls into the unstable area in the z-plane!

Stability Analysis of Closed-loop Systems in the z-plane

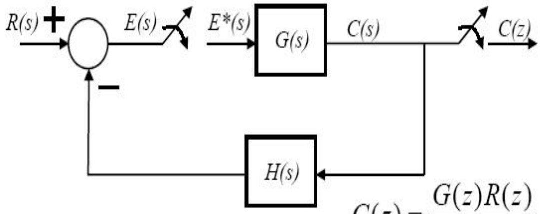
Discrete-Time Control System, Ogata Ch 3

Why are we doing this?

- We will look at the stability analysis in z-plane using the Jury method.

Pole of closed-loop system

- The general closed-loop block diagram is



$$C(z) = \frac{G(z)R(z)}{1+GH(z)}$$

Pole of closed-loop system

- The closed-loop transfer function is

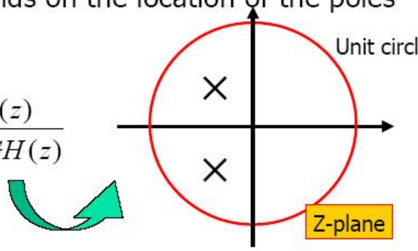
$$C(z) = \frac{G(z)R(z)}{1+GH(z)}$$

$$\frac{C(z)}{R(z)} = \frac{G(z)}{1+GH(z)}$$

Pole of closed-loop system

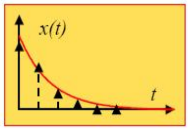
- The stability of the closed-loop system depends on the location of the poles

$$\frac{C(z)}{R(z)}=\frac{G(z)}{1+GH(z)}$$



Pole of closed-loop system

- The time-response (impulse response) of the system can be found by inverse z-transform.
- If the system response ? 0 as time ? ?, the system is stable.



Pole of closed-loop system

- If the system response ? ? as time ? ?, the system is unstable.
- If the system response ? constant (not 0, not ?) as time ? ?, the system is critically stable.

تحلیل پایداری سیستم های حلقه - بسته در حوزه Z

سیستمی با تابع تبدیل پالسی حلقه – بسته زیر را درنظر بگیرید :

$$\frac{C(z)}{R(z)}=\frac{G(z)}{1+GH(z)}$$

معادله مشخصه :

$$P(z)=1+GH(z)=0$$

☐ یک سیستم کنترل حلقه – بسته زمان – گسسته تک ورودی – تک – خروجی تغییر ناپذیر با زمان خطی وقتی **ناپایدار** می شود لااقل یکی از قطبهای حلقه – بسته آن **بیرون دایره واحد قرار گیرد و / یا یک قطب مکرر حلقه – بسته بر روی دایره واحد** صفحه Z قرار گیرد.

$$P(z)=1+GH(z)=0$$

☐ برای اینکه **سیستم پایدار** باشد، **قطبهای حلقه بسته** یا ریشه های معادله مشخصه باید **درون دایره واحد** در صفحه Z قرار گیرند. هر قطب حلقه – بسته بیرون دایره واحد سیستم را ناپایدار می کند.

☐ اگر یک **قطب ساده در z = 1 یا / و z = -1** قرار گیرد، سیستم **پایدار بحرانی** می شود. همچنین اگر **یک جفت قطب مختلط مزدوج بر روی دایره واحد** در صفحه Z قرار گیرد، سیستم **پایدار بحرانی** می شود.

☐هر **قطب مکرر حلقه – بسته بر روی دایره واحد** سیستم را **ناپایدار** می کند.

☐ صفرهای حلقه – بسته تاثیری بر روی پایداری مطلق ندارد و بنابراین می توانند در هر محلی از صفحه Z قرار گیرند.

Pole of closed-loop system

- Example : find z that

$$1+GH(z)=0$$

$$1+\left(z^2+z-3\right)=0$$

$$z^2+z-2=0$$

$$\left(z+2\right)\left(z-1\right)=0$$

$$z=-2,+1$$

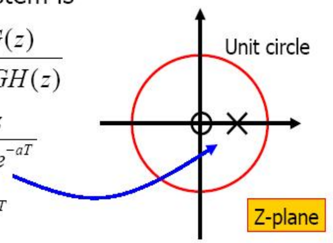
Pole of closed-loop system

- Example : If the closed-loop response of the system is

$$\frac{C(z)}{R(z)}=\frac{G(z)}{1+GH(z)}$$

$$=\frac{z}{z-e^{-aT}}$$

$$c(k)=e^{-akT}$$

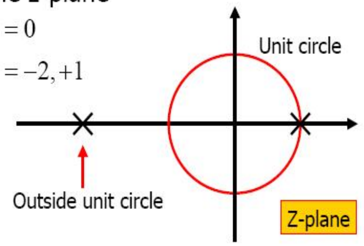


Pole of closed-loop system

- When we plot the closed-loop poles in the z-plane

$1+GH(z)=0$

$z=-2,+1$



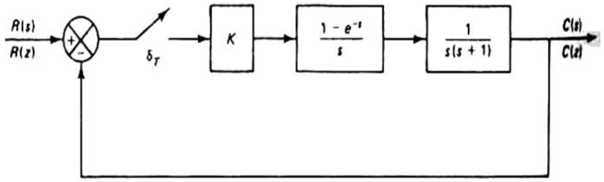
Pole of closed-loop system

- The poles of closed-loop system

$$\frac{C(z)}{R(z)}=\frac{G(z)}{\boxed{1+GH(z)}}$$

- Stable system: ALL poles must have magnitude less than 1, inside the unit circle.

مثال : پایداری سیستم وقتی k = 1 باشد را تعیین کنید.



$$G(s)=\frac{1-e^{-s}}{s}\frac{1}{s(S+1)}$$

حل :

$$G(z)=z[\frac{1-e^{-s}}{s}\frac{1}{s(S+1)}]=\frac{0.3679z+0.2642}{(z-0.3679)(z-1)}$$

تابع تبدیل پالسی حلقه - بسته :

$$\frac{C(z)}{R(z)}=\frac{G(z)}{1+GH(z)}$$

معادله مشخصه چنین است :

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

$$(z-0.3679)(z-1)+0.3679z+0.2642=0$$

$$z^2-z+0.6321=0$$

$$z^2 - z + 0.6321 = 0$$

ریشه های معادله مشخصه :

$z_1 = 0.5 + j0.6181$ $\quad$ $z_2 = 0.5 - j0.6181$

→ سیستم پایدار است

تذکر: سیستم پیوسته پایدار است اما در حضور نمونه بردار یک سیستم مرتبه دوم مانند این می تواند برای مقادیر بزرگ بهره ناپایدار شود. در مثال فوق :

$K > 2.3925$ → سیستم ناپایدار است