Lecture 3

Digital Interfacing, Sampling, Signal Conditioning, and Models of Sampled Data Systems

Prof. Krishna R. Pattipati
Prof. David L. Kleinman

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

ECE 6095/4121
Digital Control of Mechatronic Systems

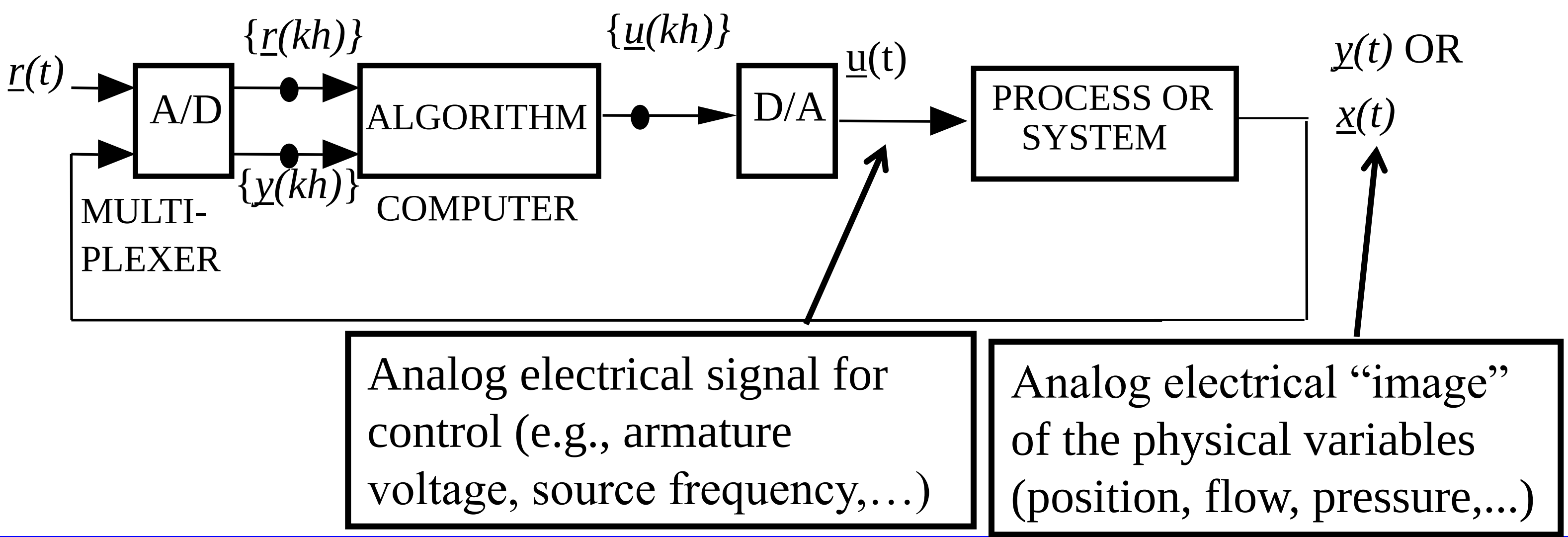
Models of Sampled Data Systems

1. **Digital Interfacing**
   - Signal Conditioning
   - A/D and D/A converters
2. **Signal Sampling and Data Reconstruction**
   - Impulse sampling model; Nyquist theorem; Aliasing and interpretation
   - Signal conditioning circuits
3. **Discrete Equivalents: State-Space Approach**
   - Discretization algorithm
4. **Discrete Equivalents: Transfer Function Approach**
   - Relation to original continuous system
5. **Model Modifications with Delay in Control**

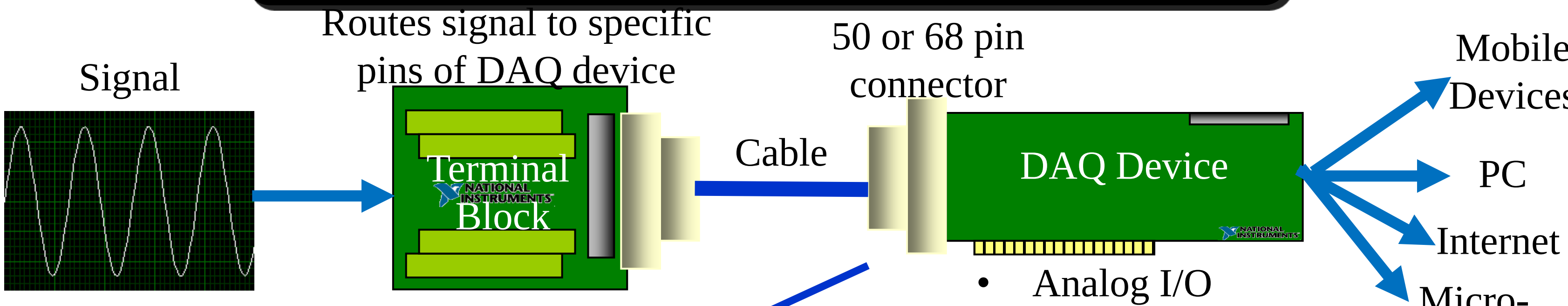
Digital Interfacing

- The system outputs, set points, state variables and control signals are typically "analog" or continuous variables

- For digital control, the sensed and conditioned (i.e., amplified, attenuated, isolated, multiplexed, filtered, compensated) system outputs, state variables and set points are converted from analog to digital form using A/D (or ADC) and the control sequences from the micro-controller (computer) are converted from digital to analog form using D/A (or DAC) prior to applying them to the actuators of the process or system



Data Acquisition (DAQ) Hardware



- Analog I/O
- Digital I/O
- Counters
- Bus connections (PCI, PXI/CompactPCI, USB, ISA/AT, PCMCIA, 1394/Firewire)

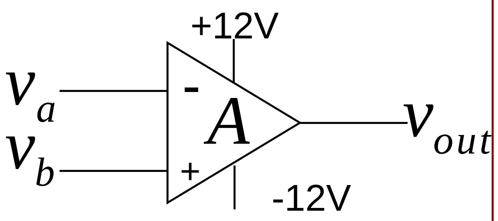
AIGND	1	2	AIGND
ACH0	3	4	ACH8
ACH1	5	6	ACH9
ACH2	7	8	ACH10
ACH3	9	10	ACH11
ACH4	11	12	ACH12
ACH5	13	14	ACH13
ACH6	15	16	ACH14
ACH7	17	18	ACH15
AISENSE	19	20	DAC0OUT[1]
DAC1OUT[1]	21	22	EXTREF[2]
AOGND	23	24	DGND
DIO0	25	26	DIO4
DIO1	27	28	DIO5
DIO2	29	30	DIO6
DIO3	31	32	DIO7
DGND	33	34	+5 V
+5 V	35	36	SCANCLK
EXTSTROBE*	37	38	PFI0/TRIG1
PFI1/TRIG2	39	40	PFI2/CONVERT*
PFI3/GPCTR1_SOURCE	41	42	PFI4/GPCTR1_GATE
GPCTR1_OUT	43	44	PFI5/UPDATE*
PFI6/WFTRIG	45	46	PFI7/STARTSCAN
PFI8/GPCTR0_SOURCE	47	48	PFI9/GPCTR0_GATE
GPCTR0_OUT	49	50	FREQ_OUT

50 pin connector

- DAQ Input Configurations
  - Resolution/Accuracy
  - Range
  - Gain
  - Code Width
  - Mode: typically differential to reject common-mode voltage and common-mode noise. Two channels used for each signal.

Back to Basics: Op-Amps - 1

OP-Amp

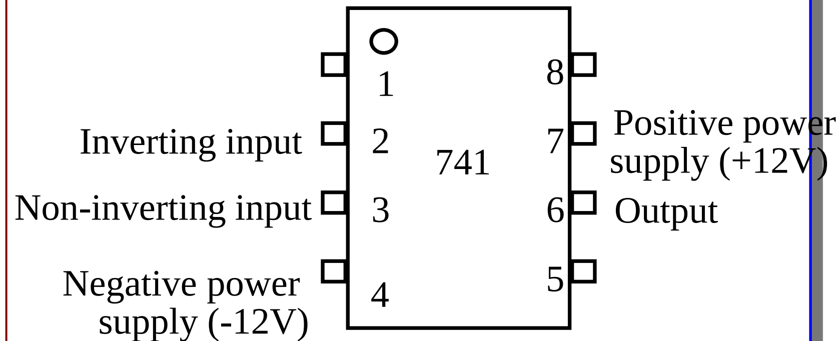


$v_{out} = A(v_b - v_a)$

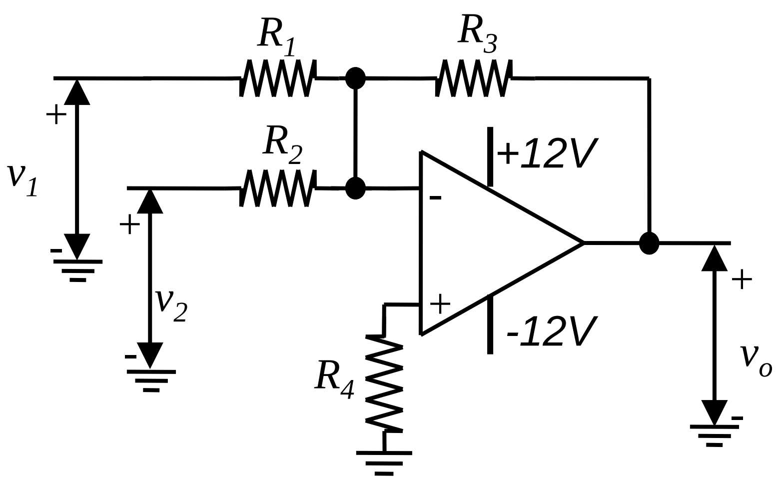
$A = open - loop\ gain$

Ideal Op-Amp:

1. $A = \infty \Rightarrow v_a \approx v_b$
2. Input Resistance $\approx \infty$
3. Output Resistance ≈ 0

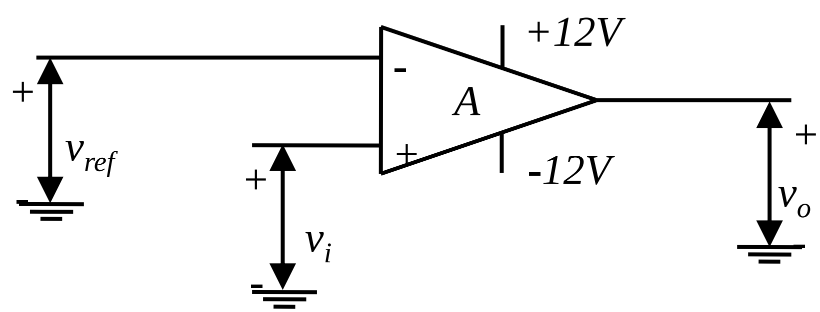


Summing amplifier



$$v_0 = -\left(\frac{R_3}{R_1}v_1 + \frac{R_3}{R_2}v_2\right); \quad \frac{1}{R_4} \approx \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$$

Comparator

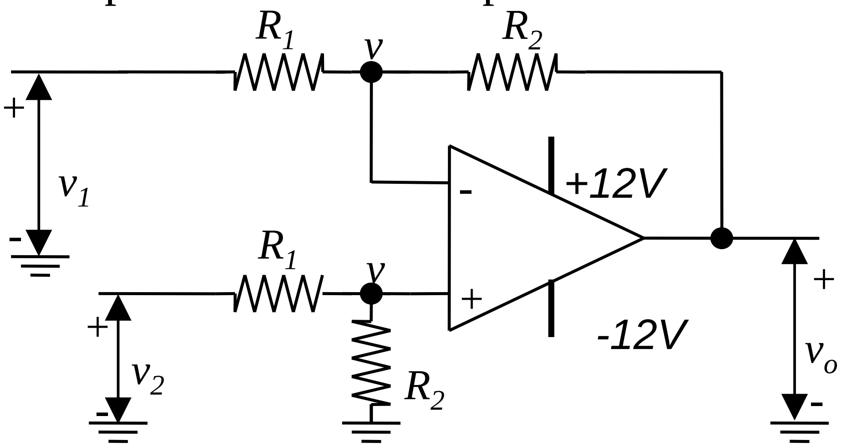


$$v_0 = \begin{cases} \min[A(v_i - v_{ref}), 12V] = 12V \ if \ v_i > v_{ref} \\ \max[A(v_i - v_{ref}), -12V] = -12V \ if \ v_i < v_{ref} \end{cases}$$

Recall A is big! Useful in Temperature Switches.

Back to Basics: Op-Amps - 2

- Dual Input Differential Amplifier

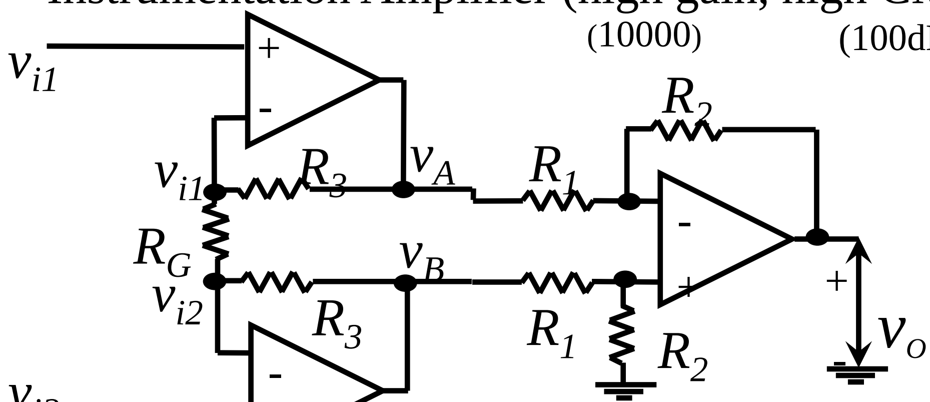


$$\frac{v_1 - v}{R_1} = \frac{v - v_0}{R_2} \Rightarrow \frac{v_0}{R_2} = -\frac{v_1}{R_1} + v\left(\frac{1}{R_2} + \frac{1}{R_1}\right)$$

$$\frac{v_2 - v}{R_1} = \frac{v}{R_2} \Rightarrow \frac{v_2}{R_1} = v\left(\frac{1}{R_2} + \frac{1}{R_1}\right)$$

$$\Rightarrow v_0 = (v_2 - v_1)\frac{R_2}{R_1}$$ ← Gain

- Instrumentation Amplifier (high gain, high CMRR)
(10000) (100dB)



$$So,\ v_0 = \frac{R_2}{R_1}\left(1 + 2\frac{R_3}{R_G}\right)(v_{i2} - v_{i1})$$ ← Gain

Know

$$v_0 = \frac{R_2}{R_1}(v_B - v_A)$$

$$Also,\ \frac{v_{i1} - v_{i2}}{R_G} = \frac{v_A - v_{i1}}{R_3} = \frac{v_{i2} - v_B}{R_3}$$

$$\Rightarrow \frac{v_B}{R_3} = \left(\frac{1}{R_G} + \frac{1}{R_3}\right)v_{i2} - \frac{1}{R_G}v_{i1}$$

$$\frac{v_A}{R_3} = \left(\frac{1}{R_G} + \frac{1}{R_3}\right)v_{i1} - \frac{1}{R_G}v_{i2}$$

$$\Rightarrow v_B - v_A = \left(1 + 2\frac{R_3}{R_G}\right)(v_{i2} - v_{i1})$$

Why Amplify Sensor Signals Prior to Conversion?

- Helps with Code Width of DAQ System
  - Smallest change in the signal that the DAQ system can detect
  - Function of gain, G, A/D resolution (number of bits of A/D, b), range of signal to be digitized, V_{max} - V_{min} (e.g., 0-10V, -10 to +10V)

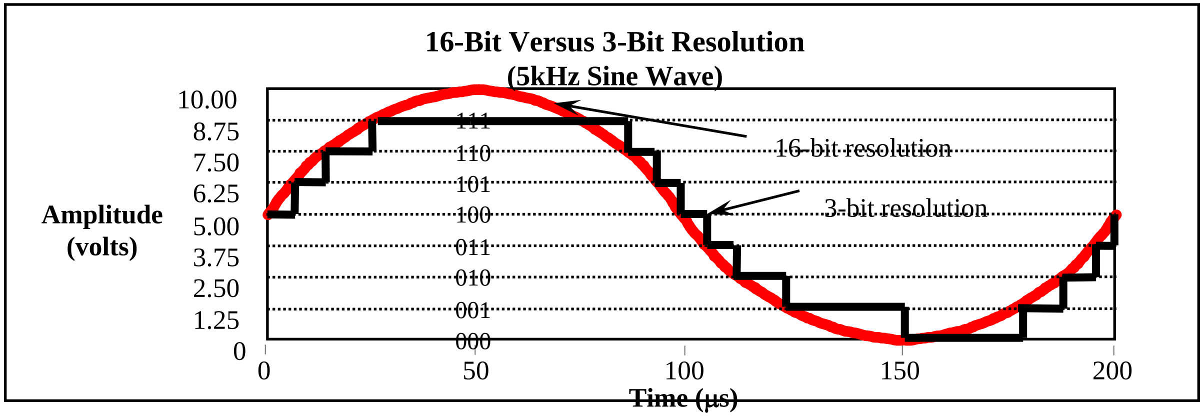
$$Code\ width = \frac{V_{\max} - V_{\min}}{G.(2^b - 1)}$$

  - Uncertainty in your measurement after A/D, U = Code width/2 (recall how you round-off numbers!)
- Thermocouple Example
  - J-type thermocouple (measures 0 to 800^0 C) has sensitivity of 0.052 mv/deg C for 20-30^0 C.
  - Consider a 16-bit A/D with G = 1 and $V_{max} – V_{min}$ = 10V.
  - Code width = 10/65535 = 0.153 mv ⇒ uncertainty in measurement, U = 0.076mV ⇒ No Good
  - A gain of 100 will have a code width of 1.53 μV/deg C and uncertainty, U of 0.765 μV/deg C

You will also be filtering signals prior to conversion. We will see why later.

Some Basic Concepts in Signal Conversion

- Resolution
  - Determines how many different voltage changes can be measured
  - 16 bit-resolution ⇒ 65,536 levels ⇒ 4-5 digit accuracy



- Range
  - DAQ devices have different ranges available (0-10V, -10V to +10V)
  - Smaller range ⇒ more precise representation of your signal (It is like selecting a scale for your plot!)
- Gain
  - Gain setting (typically 0.5, 1, 2, 5, 10, 20, 50, or 100) allows for best fit in A/D range
  - For required measurement uncertainty, U, gain, G is set via

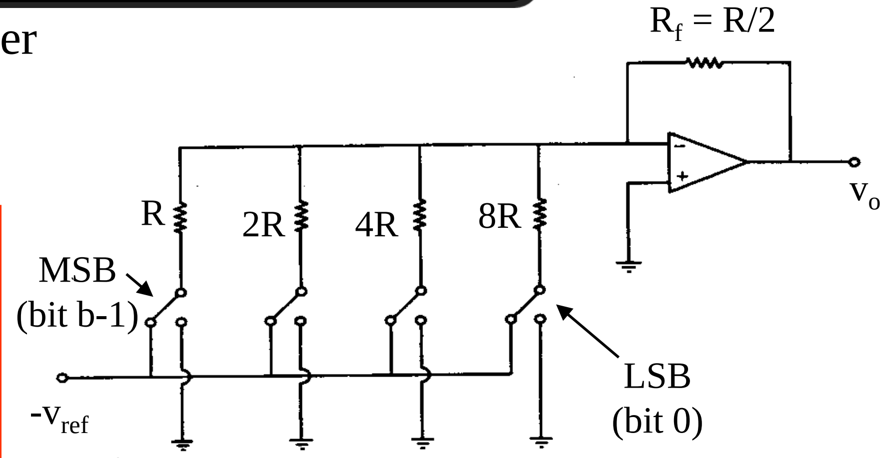
$$G = \frac{2U\,(2^b - 1)}{V_{\max} - V_{\min}}$$

Some D/A Converters

- Simple minded: Use summing amplifier
  - Wide range of precision resistors
  - 16 bits $\Rightarrow 2^{15} = 32{,}768$ range

$$v_o = \left(B_{b-1} + \frac{B_{b-2}}{2} + \frac{B_{b-3}}{2^2} + \ldots + \frac{B_i}{2^{b-i-1}} + \ldots + \frac{B_0}{2^{b-1}} \right) \frac{V_{ref}}{2}; B_i = bit\ i$$

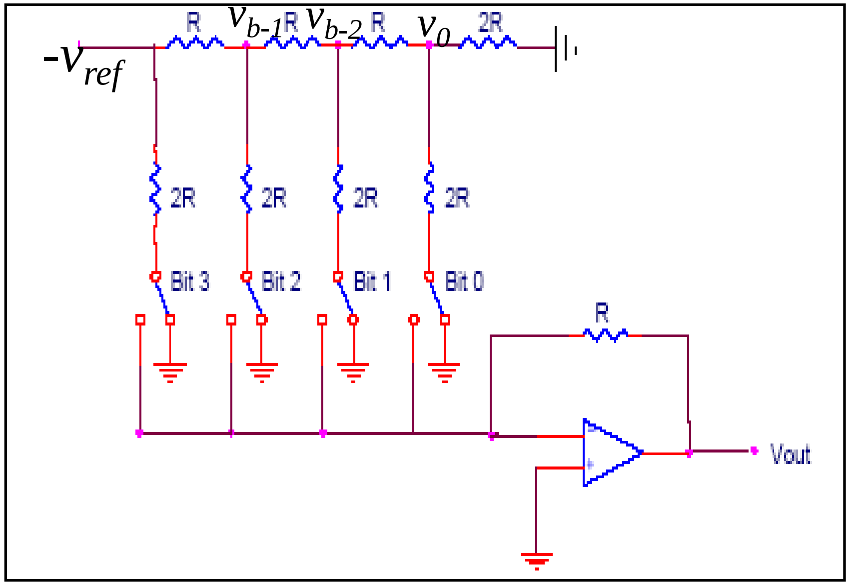
$$Full\ scale\ value\ (FSV) = \left(1 - \left(\frac{1}{2}\right)^b\right) V_{ref}$$



- R-2R Ladder D/A Converter
  - $v_i = v_{i+1}/2;\ i=0,1,2,..,b-1;\ v_b = -v_{ref}$
  - *So,*

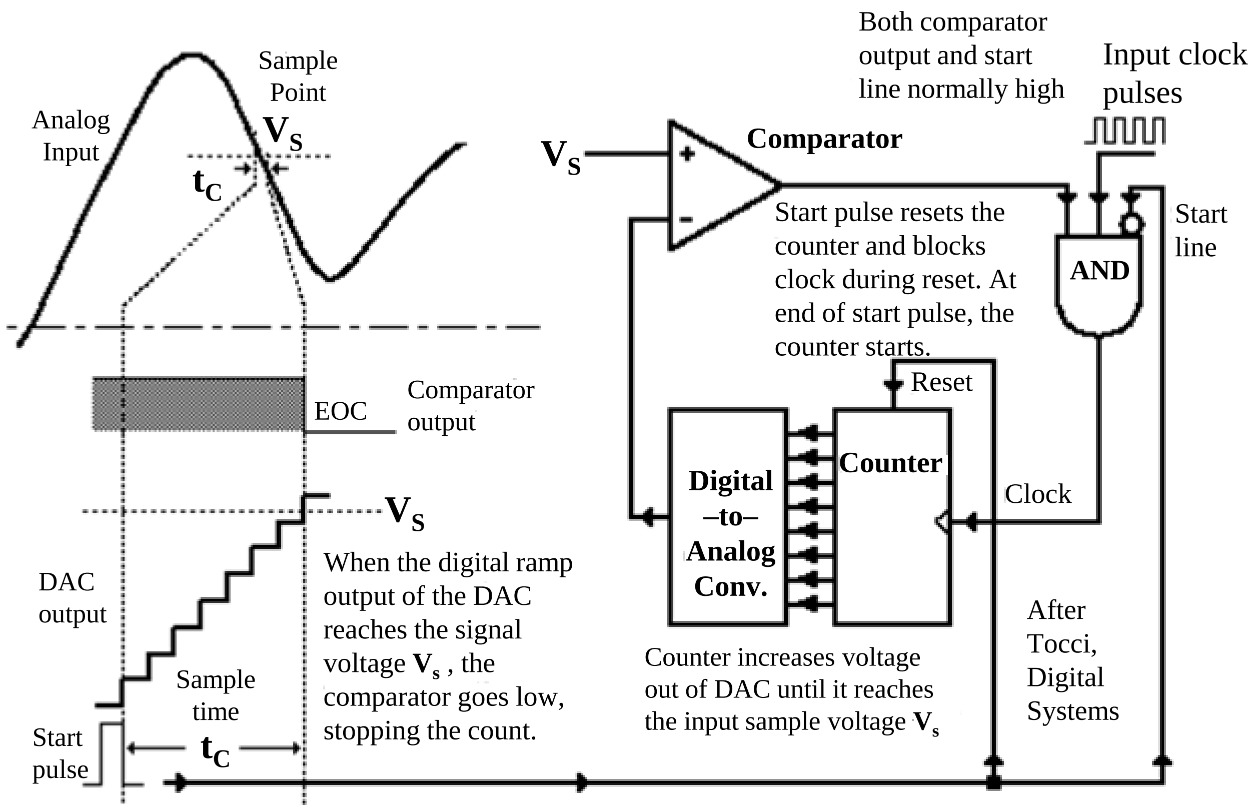
$$v_{out} = \left(\frac{1}{2} + \frac{1}{2^2} + \ldots + \frac{1}{2^b} \right) V_{ref}$$

$$Full\ scale\ value\ (FSV) = \left(1 - \left(\frac{1}{2}\right)^b\right) V_{ref}$$



Ladder Comparison A/D Converter

- Ladder Comparison (Ramp) A/D Converter

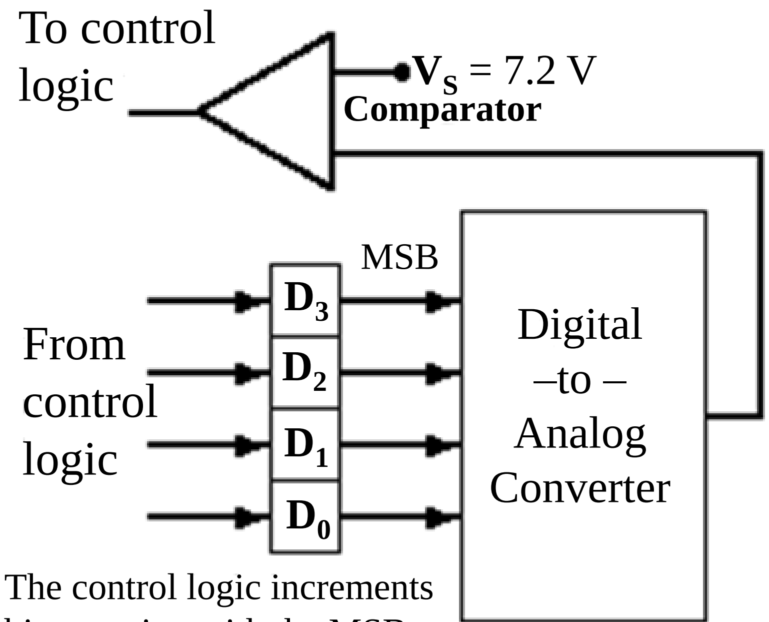


Cheap, but slow

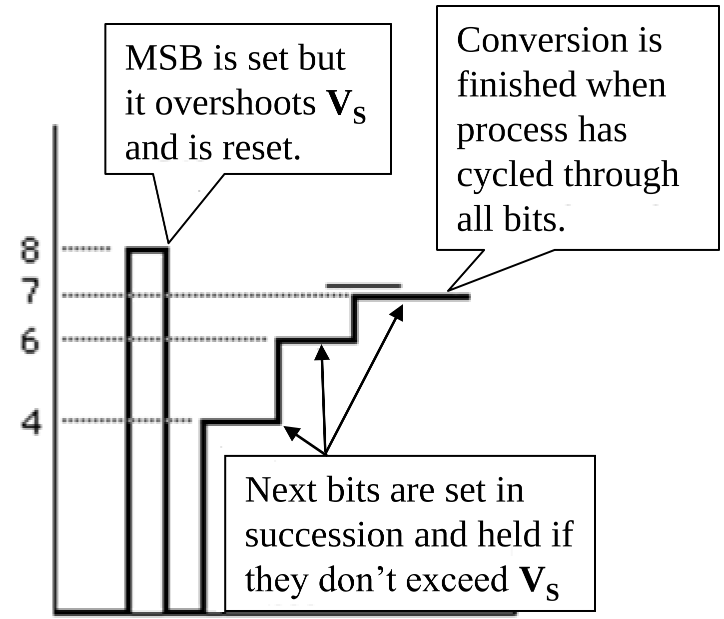
- Apply analog voltage to +ve terminal of a comparator and the output of D/A converter to –ve terminal
- Output of comparator triggers a binary counter which drives the D/A converter
- When the D/A converter voltage exceeds analog voltage, counter stops and outputs the code

Successive Approximation A/D Converter

- Successive Approximation A/D Converter



The control logic increments bits, starting with the MSB

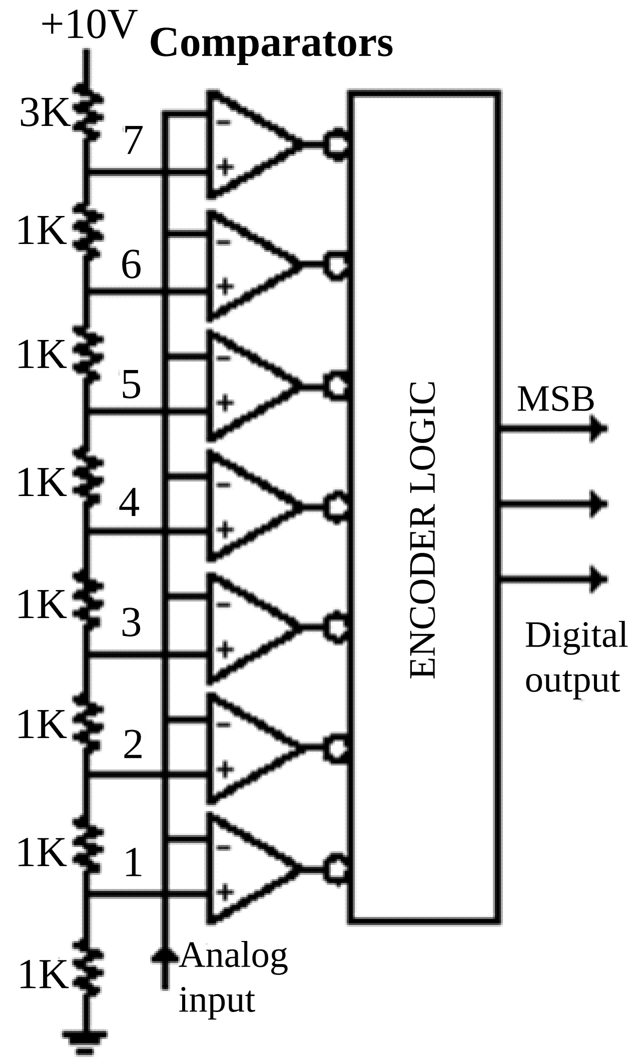


- Check if voltage corresponding to MSB > V_s. If it is, set next bits in succession and see if they don't exceed V_s
- When the D/A converter voltage exceeds analog voltage, counter stops and outputs the code
- Works well in practice

Flash A/D Converter

- Flash A/D Converter

- Basically, a truth table that coverts the ladder of inputs to the binary number output
- Fastest type of A/D converter available
- Very expensive



Mathematics of Signal Sampling

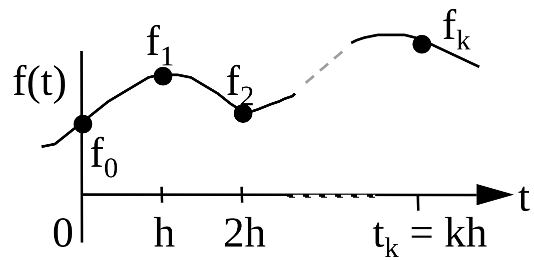
We will examine the sampling process from a mathematical viewpoint.

h = sampling period or time step

f_s = sampling frequency = number of samples/sec = $1/h$

$\omega_s = 2\pi/h$

A/D quantizes (not a major issue if b = 16, 24, 32) and samples



$f_k = f(kh)$ = sampled value of $f(t)$ at $t = kh$

The problem here is that sampling a signal loses information, namely the points in between $(k-1)h$ and kh.

So if we sample too slowly - we lose information

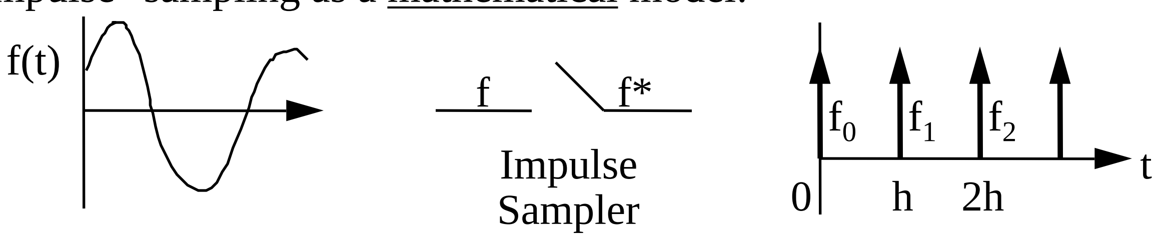
if we sample too fast - we overwork the computer

Major questions are -

(1) how fast to sample so as not to lose information? and

(2) how to reconstruct the signal $f(t)$, or an approximation, from $\{f_k\}$?

"Impulse" sampling as a <u>mathematical</u> model:



Area of impulse k is f_k.

Impulse Sampling

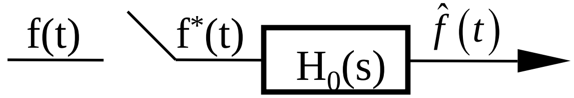
$$f^*(t) = f_0\delta(t) + f_1\delta(t-h) + f_2\delta(t-2h)+\cdots$$

can be written as

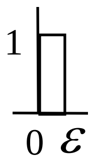
$$f^*(t) = f(t)\cdot m^*(t)$$

← periodic train of unit impulses $-\infty < t < \infty$

The signal f*(t) is not "real" but when an impulse sampler is followed by a suitable transfer function $H_0(s)$, we can model almost any practical sampling situation. We are really going from f(t) to $\hat{f}(t)$ via $f^*(t)$.



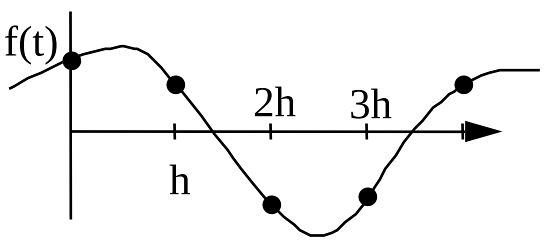
Ex. If impulse response of H_0 is

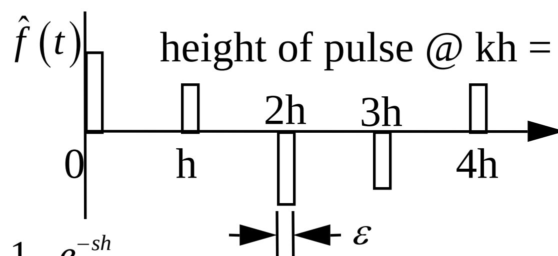


$$\Rightarrow \quad \frac{1}{s} - \frac{e^{-\varepsilon s}}{s} = H_0(s)$$

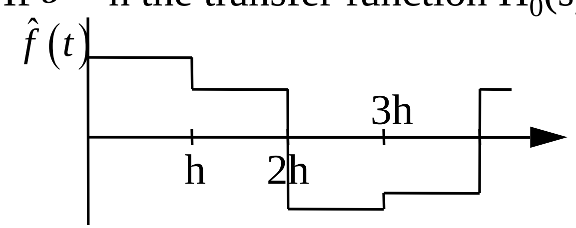
$$\frac{1}{s}\left(1-e^{-\varepsilon s}\right) = H_0(s)$$

and we get as output a pulse train.





If ε = h the transfer function $H_0(s)$ is $\dfrac{1-e^{-sh}}{s}$.



This is a "sample-and-hold" and is the most common form of sampling process plus data reconstruction.

Laplace Transform of a Sampled Signal

Take Laplace transform of $f^*(t) \triangleq F^*(s)$

$$F^*(s) = \int_0^\infty f^*(t) e^{-st} dt = f_0 + f_1 e^{-sh} + f_2 e^{-2sh} + \cdots$$

As an aside, since $z^{-1} = e^{-sh}$

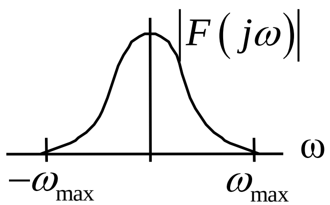
$$F^*(s) = \sum_{k=0}^{\infty} f_k z^{-k} \Big|_{z = e^{sh}} = F(z)\Big|_{z = e^{sh}}$$

where F(z) = z-transform of the sampled sequence $\{f_k\}$. Notationally, $F^*(s) = Z\{f(kh)\}\Big|_{z = e^{sh}}$

We wish to examine the relationship between $F^*(s)$ and F(s) = Laplace transform of f(t), and between

$$S_F(j\omega) = \text{"Spectrum" of } f(t) = |F(j\omega)|^2 \text{ and } S_{F^*}(j\omega) = \text{"Spectrum" of } f^*(t) = |F^*(j\omega)|^2$$

The spectrum indicates where a signal has power. (A sine wave has impulses at $\pm\omega_0$.)



To find $L[f(t) \cdot m^*(t)]$ first use Fourier series to get a different way to write $m^*(t)$. Recall, if a signal x(t) is periodic with period h, $x(t) = \frac{1}{h} \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_s t}, \quad \omega_s = \frac{2\pi}{h}$

where the Fourier coefficients, $c_n = \int_o^h x(t) e^{-jn\omega_s t} dt.$

Nyquist Theorem

Apply Fourier series to $x(t) = \delta(t)$

$$\Rightarrow c_n = \int_0^h \delta(t) e^{-jn\omega_s t} dt = 1 \quad \text{for all } n$$

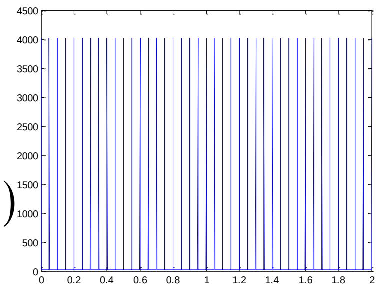
So, an alternate representation of $m^*(t)$ is

$$m^*(t) = \frac{1}{h} \sum_{n=-\infty}^{\infty} e^{jn\omega_s t} = \frac{1}{h}[1 + \sum_{n=1}^{\infty} 2\cos n\omega_s t] \text{ and } f^*(t) = \frac{1}{h} \sum_{n=-\infty}^{\infty} f(t) e^{jn\omega_s t}$$

```
h=0.05;
omegas=2*pi/h;
t=[0:0.001:2];
delta=1/h*ones(size(t));
for i=1:100
    delta=delta+2*cos(omegas*t*i)/h;
    plot(t,delta)
    pause
end
```

Thus,

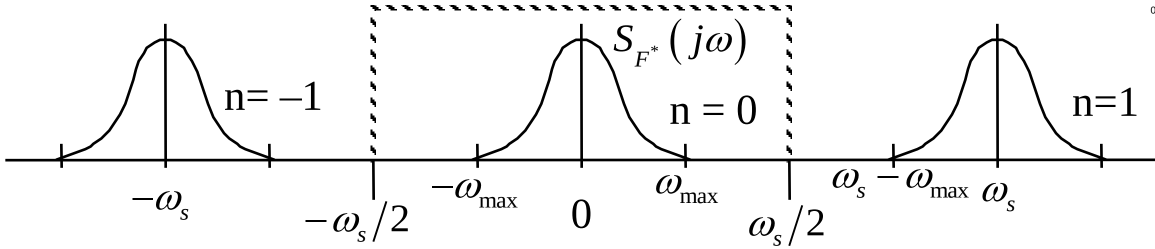
$$F^*(s) = L\left[f^*(t)\right] = \frac{1}{h} \sum_{n=-\infty}^{\infty} L\left[f(t) e^{jn\omega_s t}\right]$$



$$m^*(t) = \frac{1}{h} + \frac{2}{h} \sum_{n=1}^{\infty} \cos(n\omega_s t)$$

Using the relation $L\left[x(t) e^{at}\right] = X(s-a)$,

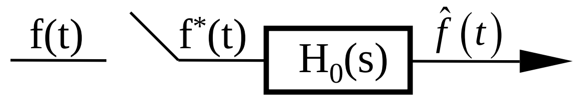
$$F^*(s) = \frac{1}{h} \sum_{n=-\infty}^{\infty} F(s - jn\omega_s) \longrightarrow F^*(j\omega) = \frac{1}{h} \sum_{n=-\infty}^{\infty} F(j\omega - jn\omega_s)$$



Nyquist Result: If original signal f(t) does not have any frequency components $> \omega_s/2$, we can (in theory) reconstruct/recover f(t) from $f^*(t)$ using an ideal low-pass filter. $\omega_N = \omega_s/2 = \pi/h$ is called the Nyquist frequency. Thus, one must sample f(t) at a rate that is at least twice the highest frequency ω_{max} in the signal, $\omega_s > 2\omega_{max}$ ($or \ \omega_N > \omega_{max}$).

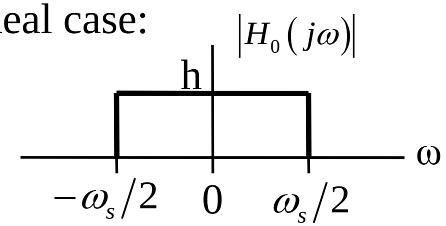
Recovering f(t) from f*(t)

Assume $\omega_s > 2\omega_{max}$



$$\hat{F}(s) = H_0(s) F^*(s)$$

- In ideal case:



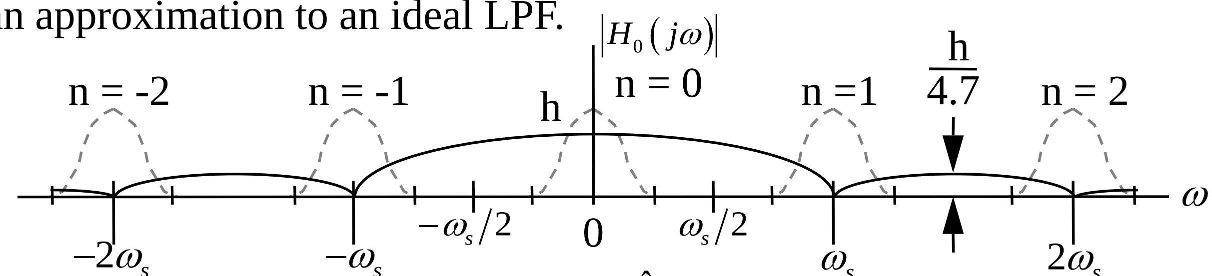
If $|H_0(j\omega)|$ is as shown then $\hat{f}(t) = f(t)$ and the signal is recovered from its samples. However, such an $H_0(s)$ is unrealizable.

- Suppose $H_0(s) = \dfrac{1 - e^{-sh}}{s}$, i.e., $\hat{f}$ is a sample and hold (zero-order hold)

$$H_0(j\omega) = e^{-j\omega h/2}\left[\frac{e^{j\omega h/2} - e^{-j\omega h/2}}{j\omega}\right] = e^{-j\omega h/2} \cdot h \cdot \left(\frac{\sin \omega h/2}{\omega h/2}\right)$$

$$\Rightarrow \quad |H_0(j\omega)| = h\left|\frac{\sin \omega h/2}{\omega h/2}\right| \quad ; \qquad \measuredangle H_0(j\omega) = -\omega h/2 \ \left(\text{delay of } h/2 \text{ sec., for } \omega < 2\pi/h = \omega_s\right)$$

This is an approximation to an ideal LPF.



Still get some high frequency components in $\hat{f}(t)$. Other signal reconstructors $H_0(s)$ are possible (e.g., polynomial interpolators) but usually are not worth the added complexity.

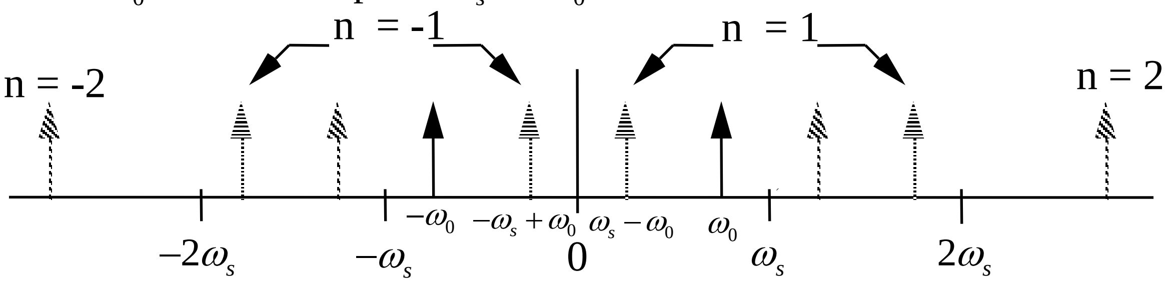
The zero-order hold is the most common form of $H_0(s)$ in digital control.

Aliasing

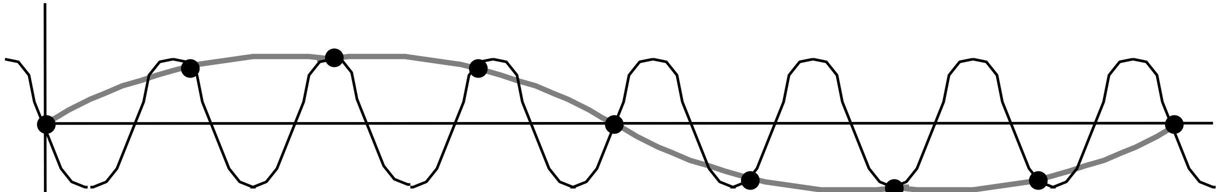
Typically, $\omega_s \sim 10-30\omega_{max}$. An interesting phenomenon happens when $\omega_s/2 < \omega_{max}$. In this case the components of $F(j\omega - jn\omega_s)$ overlap in $S_{F^*}(j\omega)$ and it becomes impossible to recover f(t).

In addition, the sampled signal $f^*(t)$ has power at frequencies not present in the original signal f(t)!

E x. $f(t) = A\sin\omega_0 t$ and we sample at $\omega_s < 2\omega_0$.



$F^*(t)$ has a low frequency component at $(\omega_s - \omega_0)$.



The original signal is "hidden", sampled signal is an "alias". The low frequency signal does not really exist in f(t), but will exist in $\hat{f}(t)$ since $H_0(s)$ is a LP filter.

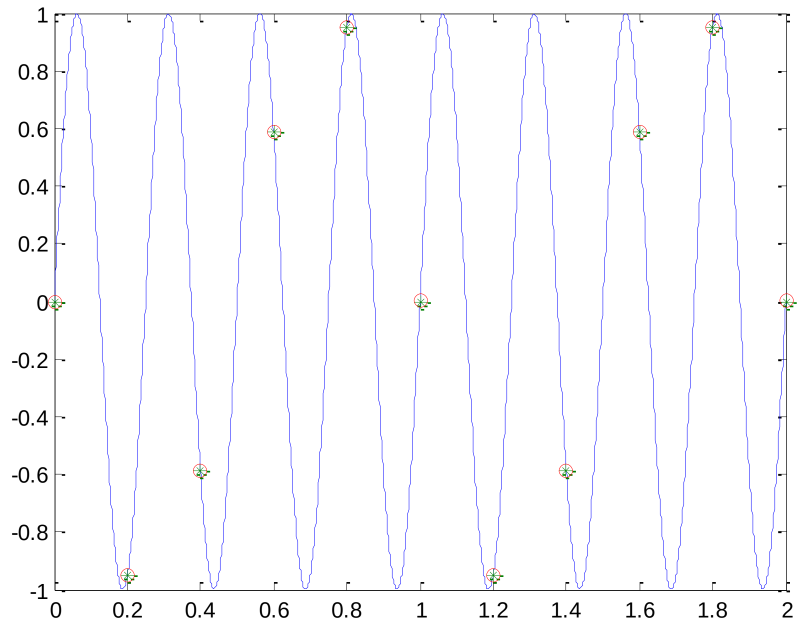
Ex. Sample a signal f(t) that has frequency components at f_1 = 0.1 Hz, f_2 = 0.8 Hz and f_3 = 1.4 Hz using f_s = 2 Hz (note Nyquist says f_s > 2.8 Hz). What are the first 5 positive frequency components of sampled signal?

	n = 0	n = 1		n = 2	
f_1	0.1	1.9	2.1	3.9	4.1
f_2	0.8	1.2	2.8	3.2	4.8
f_3	1.4	0.6	3.4	2.6	5.4

Aliasing Illustrated in Time Domain

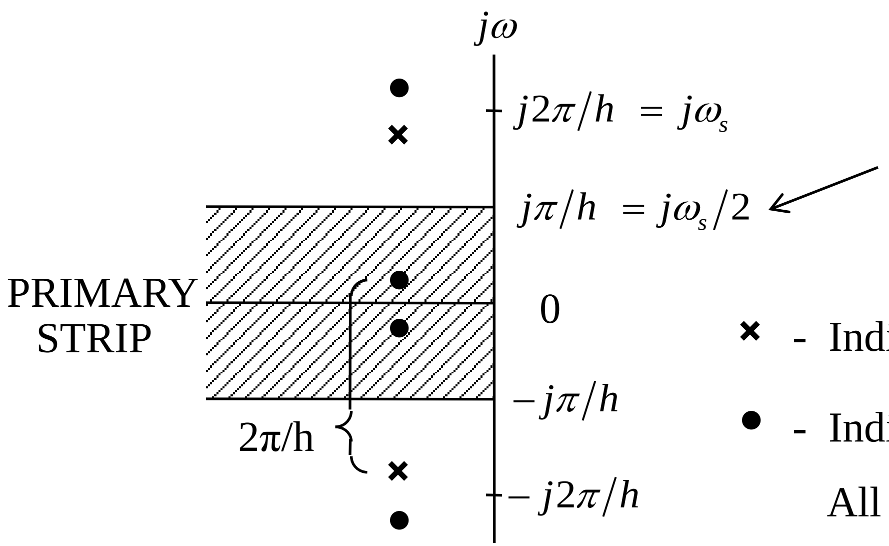
Let us take a simple sinusoid of frequency 4 Hz and sample it at 5Hz. We will show that a signal of 1 Hz is an alias.

```
t=[0:0.001:2]
f=sin(2*pi*4*t);  % continuous signal
t1=[0:0.2:2];
f1=sin(2*pi*4*t1); % sampled signal at 5Hz
f2=sin(2*pi*t1);   % Alias signal 1Hz
plot(t,f,t1,f1,'*',t1,-f2,'o') % note negative sign
```



* sampled signal
o aliased signal

Interpretation of Aliasing in s-Plane



× - Indicates frequency of original signal

● - Indicates alias frequencies

All points $\frac{j2\pi N}{h}$ apart, give same z_i.

After sample and hold (or other type of reconstructor), we pick out predominantly those signals in the primary strip, $-\pi/h < \omega < \pi/h$.

Since the aliased frequencies are not "real", i.e., not in original signal, any controller aimed at reducing the "observed" oscillations will fail.

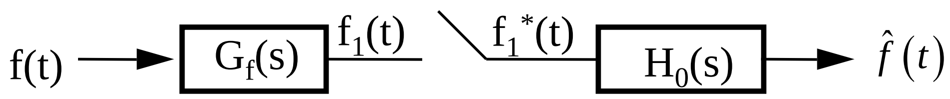
- Aliasing effects will be observed in
  - frequency folding in s-plane
  - time response
  - Fourier spectrum

How to Avoid Aliasing?

- There is no way to fix $f^*(t)$ after you have sampled. So, you must assure that the signal to be sampled has no frequencies higher than $\omega_N = \pi/h$.

But, real signals have power in [–∞, ∞] (with caveat).

$\Rightarrow$ Prefilter the signal $f(t)$ before sampling (anti-aliasing).

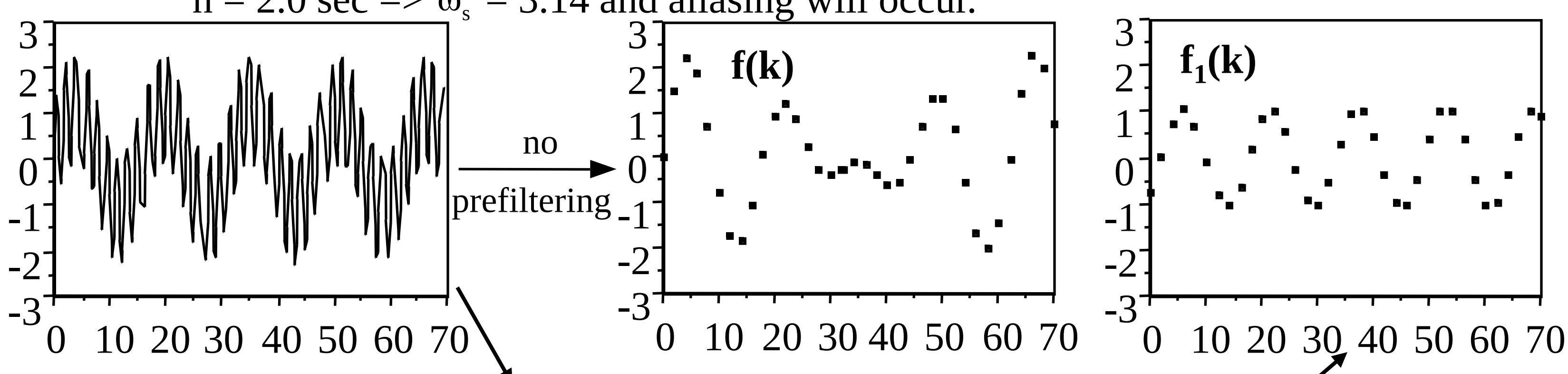


$$\text{Typical } G_f(s) = \frac{\omega_f^2}{s^2 + 2\zeta\omega_f s + \omega_f^2}$$

$$\zeta = \frac{\sqrt{2}}{2} \quad (\text{Butterworth Filter})$$

Usually pick $\omega_f \sim \omega_N/2 = \pi/2h$ to be safe, but beware of using a $G_f(s)$ in a feedback loop due to added negative phase shift that reduces ϕ_m. Some authors suggest $\omega_N/1.28 \approx 0.8\ \omega_N = 0.4\ \omega_s$

Ex: $f(t) = 1.1 \sin 0.4t + 1.2 \sin 3.45t$ ~ signal + high frequency noise. Sample period $h = 2.0$ sec => $\omega_s = 3.14$ and aliasing will occur.

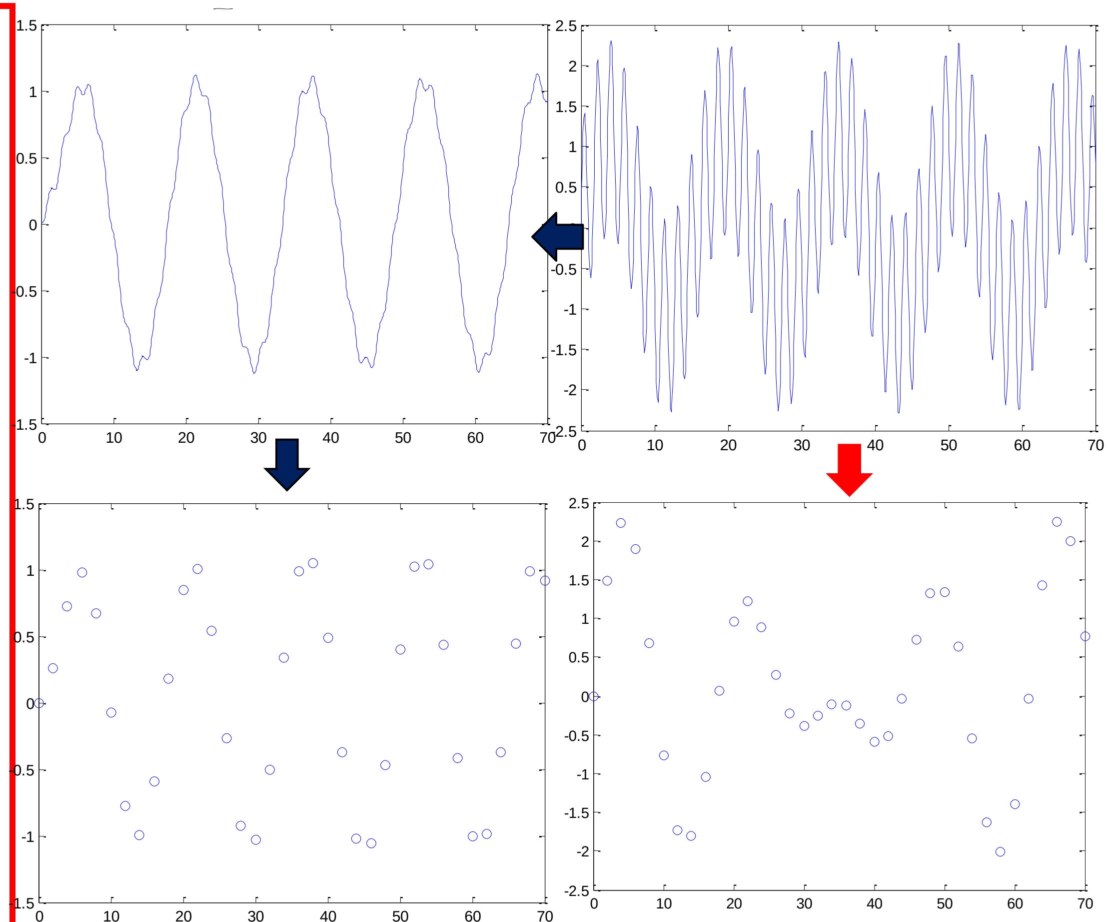


Prefilter f(t) using a 2nd-order Butterworth filter with $\omega_f = 0.785$ and then sample the output, $f_1(t)$.

How to Avoid Aliasing?

<u>Ex</u>: $f(t) = 1.1 \sin 0.4t + 1.2 \sin 3.45t$ signal + high frequency noise. Sample period $h = 2.0$ sec => $\omega_s = 3.14$ and aliasing will occur.

```
% nearly-continuous signal
delt=0.1;
t=[0:delt:70]';
n=length(t);
ft=1.1*sin(0.4*t)+1.2*sin(3.45*t);
plot(t,ft)
pause
% sampled signal
kt=[0:2:70]';
nk=length(kt);
fk=1.1*sin(0.4*kt)+1.2*sin(3.45*kt);
plot(kt,fk,'o')
pause
numgf=[0.785^2];
dengf=[1 2*0.707*0.785 0.785^2];
gfs=tf(numgf,dengf)
[y,t]=lsim(gfs,ft,t);
plot(t,y)
pause
% sampled signal
h=2.;
kt=t([1:h/delt:n])
f1=y([1:h/delt:n])
plot(kt,f1,'O')
```

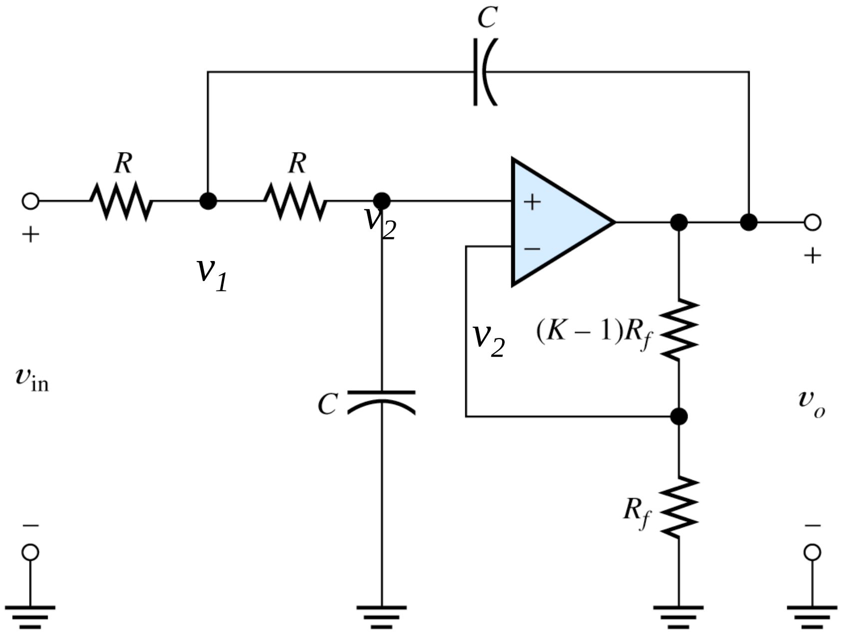


Antialiasing/Aliasing Examples

- Example 1: Consider N = 1024 data points from a signal sampled at 1ms interval (h = 0.001 sec).
  - Sampling frequency , f_s = 1000 Hz = 1kHz $\Rightarrow \omega_s$ = 6280 rad/sec
  - Nyquist frequency, f_N = 500 Hz $\Rightarrow \omega_N$ = 3140 rad/sec
  - Antialiasing filter frequency, f_f = 250-400 Hz $\Rightarrow \omega_N$ = 1570 -2512 rad/sec
  - If you did discrete Fourier transform, you will get 1024 points representing frequencies $(k/N)^* f_s$; $k = 0,1,2,..N-1$. These are also called spectral lines.
  - Spectral line separation = f_s/N = 0.9766 Hz.
  - For an ideal filter with cut-off frequency of 250-400 Hz, keep the first 244-391frequency components (i.e., set the rest to zero) as the useful spectrum and then do an IDFT to recover the noise filtered signal.

- Example 2: Suppose you have a sinusoidal signal of frequency 10 Hz and you sample it at 50Hz. Another sinusoidal signal of the same amplitude, but higher frequency, f was found to yield the same data when sampled at 50Hz. What is the likely frequency, f ?
  - Sampling frequency , f_s = 50 Hz
  - Aliasing frequencies = $n\, f_s \pm$ 10 Hz.
  - So, f = 40Hz, 60 Hz, 90Hz, 110Hz,….

Sallen-Key Low Pass Butterworth Filter



$$G_f(s) = \frac{K/R^2C^2}{s^2 + \frac{(3-K)}{RC}s + \frac{1}{R^2C^2}}$$

$$= \frac{K\omega_n^2}{s^2 + (3-K)\omega_n s + \omega_n^2}$$

$$For\ \xi = 1/\sqrt{2}, K = 3-\sqrt{2} = 1.586$$

$$\frac{v_{in}(s) - v_1(s)}{R} + (v_0(s) - v_1(s))Cs = \frac{v_1(s) - v_2(s)}{R} \quad (1)$$

$$\frac{v_1(s) - v_2(s)}{R} = v_2(s)Cs \Rightarrow v_1(s) = (1+RCs)v_2(s) \ (2)$$

$$v_0(s) = Kv_2(s) \ (3)$$

$$\Rightarrow v_{in}(s) = (2+RCs)v_1(s) - (1+KRCs)v_2(s)$$

$$= \big((2+RCs)(1+RCs) - (1+KRCs)\big)v_2(s)$$

In general, Butterworth low pass filters have flat frequency response . For order *p*

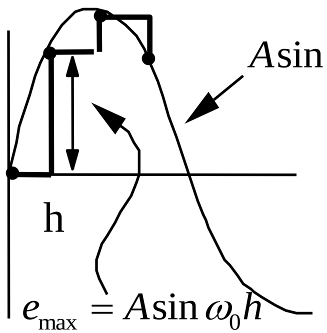
$$|G_f(\omega)| = \frac{G_f(0)}{\sqrt{1+\left(\frac{\omega}{\omega_n}\right)^{2p}}}$$

Sampling for Accuracy

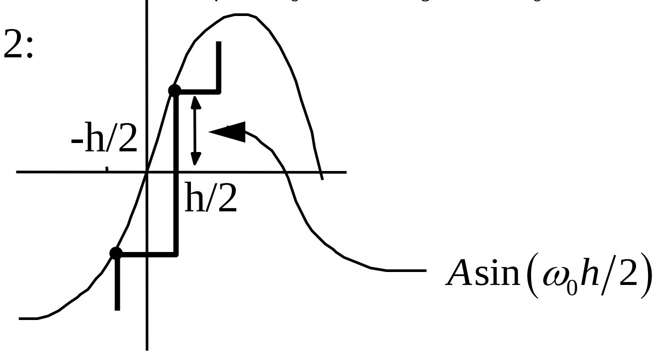
- For a single sine wave, $A\sin\omega_0 t$, Nyquist criterion says use more than two (2) samples/period $(\omega_s > 2\omega_0)$, but reconstruction error using a zero-order hold is terrible ==> we really need to sample at a higher rate.

If we use a sample and hold with N $\geq$ 4 samples/period, then $h = 2\pi / N\omega_0$ *and* $\omega_s = N\omega_0$.

Case 1:



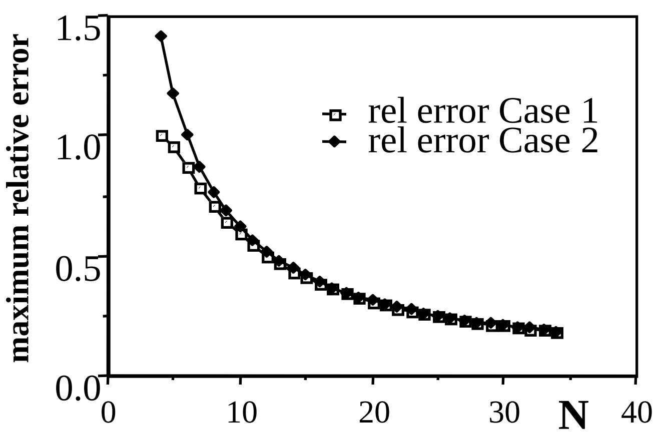
Case 2:



max relative error = $\dfrac{A\sin(2\pi/N)}{A} = \sin(2\pi/N)$

max relative error with $h/2$ shift

$= \dfrac{2A\sin(\pi/N)}{A} = 2\sin(\pi/N)$



Usually we try for $\omega_s = (10 \rightarrow 30)\omega_{\max}$ when using a signal reconstruction criteria

Sampling Period h for Control

- **State space representation**: If $\lambda_1, \lambda_2, \ldots, \lambda_n$ are the eigenvalues of A, then to avoid aliasing we must have λ_i within primary strip in the s-plane, i.e., $| \text{Im}(\lambda_i) | < \pi/h$.

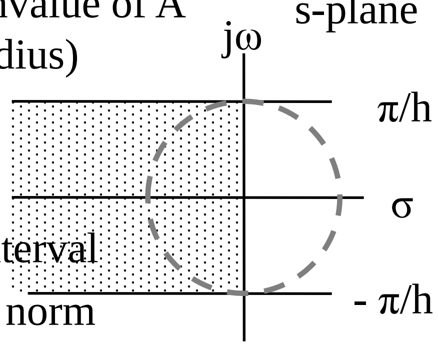
  More manageably, $|\lambda_i| < \pi/h \quad i = 1, 2, \ldots, n$

  i.e., poles within circle of radius π/h. $\Rightarrow h_{max} = \pi / |\lambda_{max}(A)|$ ← largest eigenvalue of A (spectral radius)

  This is too high a limit from a control viewpoint, instead we seek

  $h \leq c / |\lambda_{max}(A)|$ with c = **0.2** to 0.5 (1/6 → 1/15 of Nyquist sampling interval

  An approximation: $|\lambda_{max}(A)| \sim \|A\|$ because $|\lambda_{max}(A)| \leq \|A\|$ for any norm



Primary Strip

- **Relation to Closed-loop bandwidth**: ω_{BW} in rad/sec $\Rightarrow f_{BW} = \omega_{BW}/2\pi$ in Hz

$$\frac{1}{30 f_{BW}} < h < \frac{1}{15 f_{BW}} \Rightarrow \frac{1}{5\omega_{BW}} < h < \frac{2}{5\omega_{BW}}$$

- **Relation to Rise time,** T_r: about 10% of the rise time $\Rightarrow h \approx 0.1\, T_r$

  A rule of thumb: $T_r \approx \dfrac{1}{2 f_{BW}}$

- **Gain cross over frequency,** ω_c

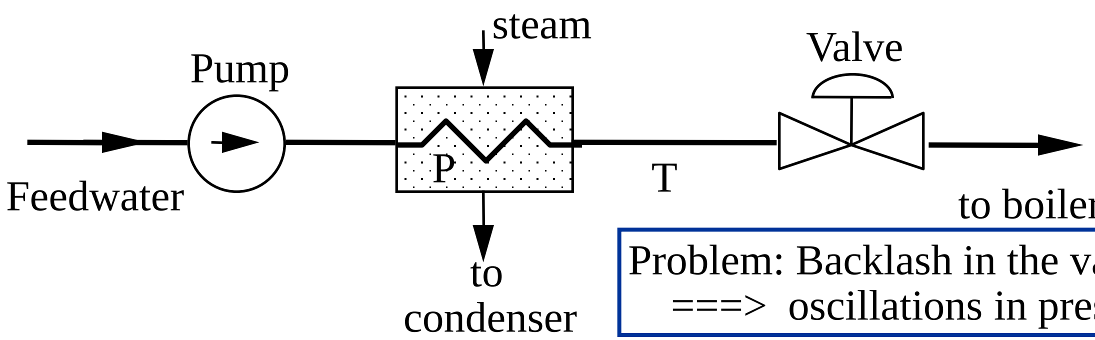
$$0.15 < h\omega_c < 0.5$$

  ω_c is an approx. measure of closed-loop bandwidth $\Rightarrow$ 12 to 40 times $f_c = \omega_c / 2\pi$

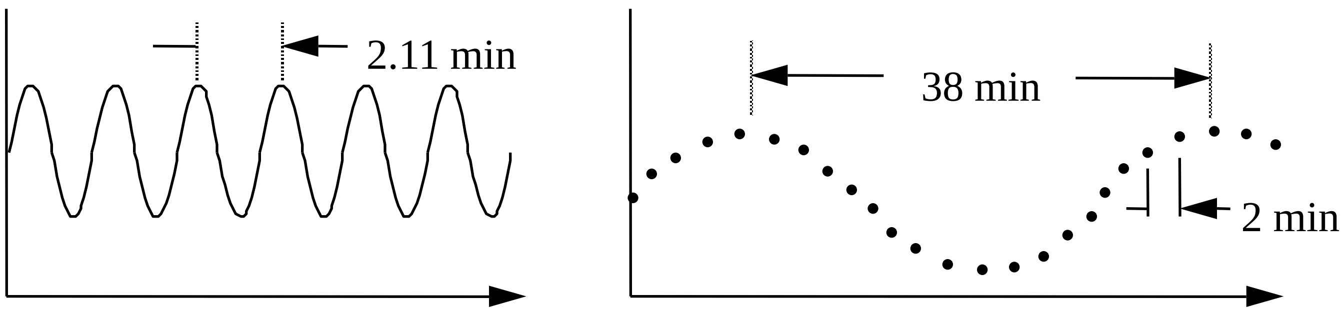
Need to experiment with different values of h during design

Example of Aliasing in a Control Setting

- Feedwater heating in a ship propulsion plant (Astrom & Wittenmark)



Problem: Backlash in the valve positioner
===> oscillations in pressure (P) and temperature (T)



Continuous recording of P t Sampled recording of T t

Pressure and temperature are coupled, and should oscillate at the same frequency! What happened?

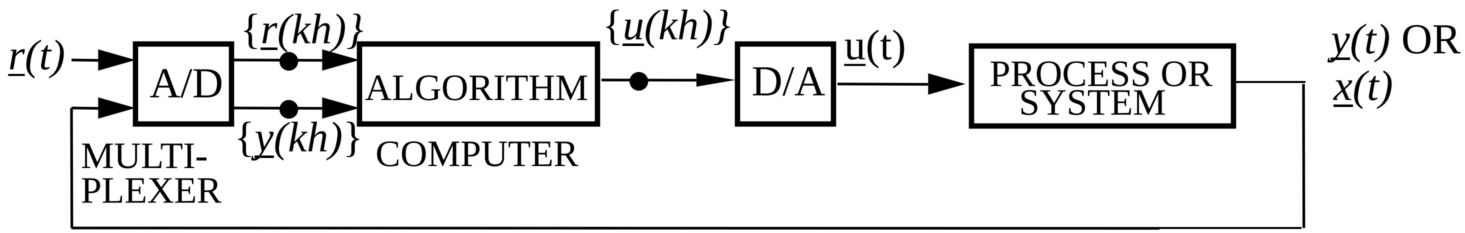
Sampling frequency, $\omega_s = 2\pi/2 = 3.14$ rad / min

Pressure oscillation frequency, $\omega_p = 2\pi/2.11 = 2.98$ rad/min

Lowest aliasing frequency, $\omega_s - \omega_p = 0.16$ rad/min $\Rightarrow$ $T = 38$min

Conclusion: The sampler did not take this course!

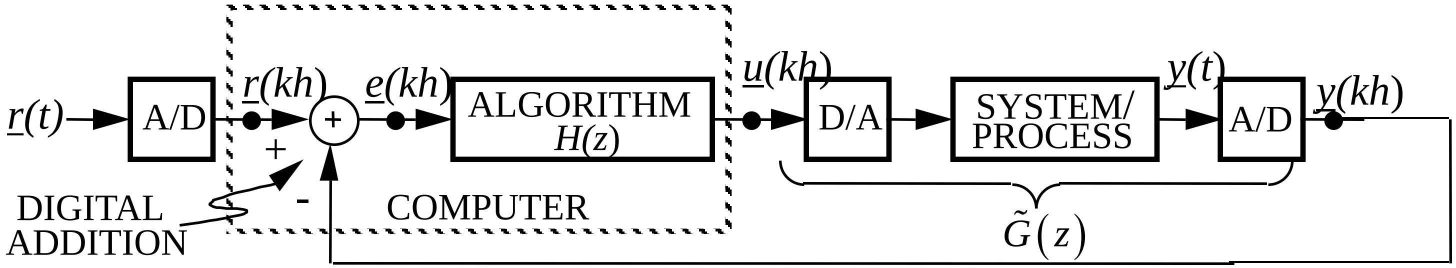
Analysis of the Basic Digital Control Loop



- The computer algorithm generates a sequence of values *u(kh)* from the discrete samples *y(kh)* and *r(kh)*, or from *e(kh) = r(kh) - y(kh)*, e.g., *u(z) = H(z) e(z)*.
- Process Model - continuous inputs and outputs

  transfer function or State-Space Model

  $G(s) \longleftrightarrow \dot{x} = Ax + Bu, \; y = Cx + Du$
- Computer outputs values *u(kh)* and at some time later sees the response *y(mh)*. The computer "puts out" samples and "sees" samples, i.e., it sees a discrete system from *u(kh)* to $y(kh) \Rightarrow \tilde{G}(z)$.
- Redraw loop from computer's view [eg., *u(z) = H(z) e(z)*].



WHY? => 1. to enable analysis as a discrete FB loop

2. to enable design of a discrete *H(z)* vis-a-vis discrete $\tilde{G}(z)$

3. We are "controlling" $\tilde{G}(z)$ not *G(s)*.

Discrete System Time Signals

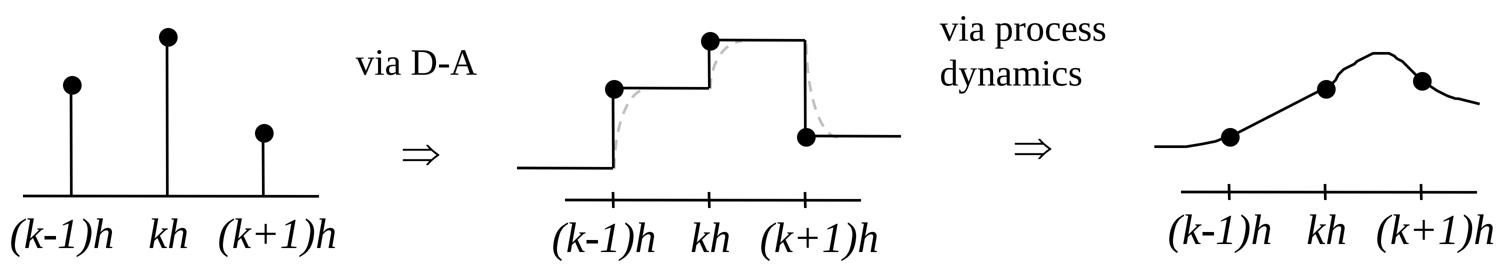
Typically there will be delays in the loop

- computational delays
- measurement delays
- process delays

} lump as some equivalent delay τ

Assume: D/A is a zero-order hold ; All A/Ds are synchronized

Consider signals around the loop



u(kh) from computer algorithms

*u(t)*output of D/A (zero-order hold) = input to system

Output of system, *y(t)*
● = sampled values output of A/D input to algorithm

Definitions

$\underline{y}(k) = \underline{y}(kh)$ = sampled values of $\underline{y}(t)$ <u>at</u> time $t = kh$

$\underline{u}(k) = \underline{u}(kh)$ = values of $\underline{u}(\bullet)$ computed by algorithm using the samples $\underline{y}(kh)$ and $\underline{r}(kh)$; output from computer at time kh^+, if there is no computational delay

==> $\underline{u}(kh)$ = values of system input <u>over</u> $[kh^+, (k+1)h]$

Model for Equivalent Discrete System, $\tilde{G}(z)$

1. System defined by state equations, no delay
2. System defined by transfer function, no delay
3. Modifications to 1 and 2 when $\tau \neq 0$

State-Space Approach

$$\dot{\underline{x}}(t) = A\,\underline{x}(t) + B\,\underline{u}(t)$$
$$\underline{y}(t) = C\,\underline{x}(t) + D\,\underline{u}(t) \quad \Rightarrow \quad G(s) = C(sI - A)^{-1} B + D$$

Compute $\underline{x}\,[(k+1)h] \triangleq \underline{x}(k+1)$ = value of $\underline{x}(t)$ at $t = (k+1)h$ from knowledge of $\underline{x}(kh)$ = value of $\underline{x}(t)$ at $t = kh$ and $\underline{u}(kh)$ = system input over $(kh, (k+1)h]$.

$$h = \frac{0.2}{\|A\|}$$

Use state transition equation,

$$\underline{x}(t_2) = e^{A(t_2-t_1)}\underline{x}(t_1) + \int_{t_1}^{t_2} e^{A(t_2-\xi)}B\underline{u}(\xi)d\xi$$

$t_1 = kh,\ t_2 = (k+1)h$ and $\underline{u}(\xi) = \underline{u}(kh)$ over $(t_1,\ t_2]$

$$\underline{x}[(k+1)h] = e^{Ah}\underline{x}(kh) + \int_{kh}^{(k+1)h} e^{A((k+1)h-\xi)}Bd\xi \cdot \underline{u}(kh)$$

let $\sigma = (k+1)h - \xi$

$$\underline{x}[(k+1)h] = e^{Ah}\underline{x}(kh) + \int_0^h e^{A\sigma}d\sigma B\underline{u}(kh) \;\Rightarrow\; \underline{x}(k+1) = \Phi\underline{x}(k) + \Gamma\underline{u}(k)$$

where $\Phi = e^{Ah}$; $\Psi(h) = \int_0^h e^{A\sigma}d\sigma$; $\Gamma = \Psi(h)B$

value of system input right at time t = kh (subtle point)

Output $\underline{y}(kh) = C\underline{x}(kh) + D\underline{u}[(k-1)h]$

$$\underline{y}(k) = C\underline{x}(k) + D\underline{u}(k-1)$$

Transfer funtion Matrix (TFM): $\tilde{G}(z) = C(zI - \Phi)^{-1}\Gamma + Dz^{-1}$

Computing Φ and Γ (or Ψ)

- Note that Φ and Γ are independent of k. Compute once for a given time step h.

Analytic: $e^{Ah} = L^{-1}\left[(sI - A)^{-1}\right]\Big|_{t=h}$

exact value obtained, but very time-consuming and not practical for n > 3. Then, need to obtain Ψ by integrating $e^{A\sigma}$ over [0, h].

Numerical: If h is small ==> Taylor series approximations are good

$$e^{Ah} = I + Ah + A^2h^2/2! + \ldots$$

To compute $\Psi(h)$ substitute approximation $e^{A\sigma} \sim I + A\sigma + A^2\sigma^2/2! + \ldots$

$$\Psi(h) = \int_0^h e^{A\sigma} d\sigma = \int_0^h \left[1 + A\sigma + A^2\sigma^2/2! + \cdots\right] d\sigma$$

$$\Psi(h) \doteq h\left[I + Ah/2! + A^2h^2/3! + \cdots + A^M h^M/(M+1)!\right]$$

where the number of terms M must be chosen large enough so that the Taylor approximations are valid; i.e., we want,

$(Ah)^M/(M+1)! << I$ ==> $\| A \|^M h^M/(M+1)! < 10^{-6}$. Then $\Phi = e^{Ah} = I + A\Psi(h)$

Algorithm to find M = # terms in series, given h

```
        C1 = || A || h/2
        Do for M = 2, 20
                C1 = C1 * || A || h/(M+1)
                if C1 < 10^-6  stop → return M, if M < 4 set M = 4
        End do
```

(Note: $\| A \|^{19}/20! \sim 10^{-9}$ if $\| Ah \| = \pi$)

Algorithm for Obtaining Ψ(h) and Φ, Γ

Once M is determined, compute Ψ(h) via series. Since the magnitude of the higher-order terms in series decreases as M grows, sum the series using reverse nesting. -

$$\Psi(h) = h\left[I + \cdots \frac{Ah}{M-2}\left(I + \frac{Ah}{M-1}\left(I + \frac{Ah}{M}\left(I + \frac{Ah}{M+1}\right)\right)\right)\cdots\right]$$

This assures that very small numbers are never added to much bigger numbers.

Flow diagram of a Subroutine "Dscrt" (your own c2d function) for general use:

Input n, A, B, h → Select $M \geq 4$ s.t. $\|Ah\|^M/(M+1)! < 10^{-6}$

Initialize $N = M+1$, $\Phi = A$

Do for j = 1, M
  $\Psi = I + \Phi * (h/N)$
  $N = N-1$
  $\Phi = A \cdot \Psi$
End do

At this point we have:
$\Psi = I + Ah/2! + \ldots + (Ah)^M/(M+1)!$
$\Phi = A + A^2h/2! + \ldots + (A)^{M+1} h^M/(M+1)!$

$\Psi \leftarrow h\Psi$
$\Phi \leftarrow I + h\Phi$

→ $\Gamma = \Psi B$ → Return Φ,Ψ,Γ

Discuss how to use MATLAB for this

Then: $\tilde{G}(z) = C(zI-\Phi)^{-1}\Gamma + \{Dz^{-1}\}$

Use SS→TFM code to obtain coefficients.

Modifications to SS → TFM

- Use modified SS→TFM code to obtain coefficients.

Let $\underline{\gamma}_j$ be the j^{th} column of Γ and $\underline{c}_k^T$ be the k^{th} row of C

$$Key\ relation: g_{kj}(z) = \underline{c}_k^T (zI - \Phi)^{-1} \underline{\gamma}_j + d_{kj} z^{-1} = \frac{|zI - \Phi + \underline{\gamma}_j \underline{c}_k^T|}{|zI - \Phi|} - 1 + \frac{d_{kj}}{z}$$

$$= \frac{z|zI - \Phi + \underline{\gamma}_j \underline{c}_k^T| + (d_{kj} - z)|zI - \Phi|}{z|zI - \Phi|}$$

Let $\delta_1, \delta_2, ..., \delta_n$ be the eigen values of $(\Phi - \underline{\gamma}_j \underline{c}_k^T)$ and $\lambda_1, \lambda_2, ..., \lambda_n$ be the eigen values of Φ. Then,

$$g_{kj}(z) = \frac{z\prod_{i=1}^{n}(z - \delta_i) + (d_{kj} - z)\prod_{i=1}^{n}(z - \lambda_i)}{z\prod_{i=1}^{n}(z - \lambda_i)}$$

$$= \frac{z[z^n + \tilde{b}_1 z^{n-1} + \tilde{b}_2 z^{n-2} + ... + \tilde{b}_n] + (d_{kj} - z)[z^n + a_1 z^{n-1} + a_2 z^{n-2} + ... + a_n]}{z(z^n + a_1 z^{n-1} + a_2 z^{n-2} + ... + a_n)}$$

$$= \frac{z^{-1}(b_0 z^n + b_1 z^{n-1} + b_2 z^{n-2} + ... + b_n)}{z^n + a_1 z^{n-1} + a_2 z^{n-2} + ... + a_n}; b_i = \tilde{b}_{i+1} + d_{kj} a_i - a_{i+1}, i = 0,1,2,..,n; a_0 = 1, \tilde{b}_{n+1} = a_{n+1} = 0$$

```
b(n+2,j,k)=0; a(n+2)=0;
for i =1:n+1
        b(i,j,k)=b(i+1,j,k)+D(k,j)*a(i)-a(i+1);
end
```

Mods to SS → TFM code

Example: First Order System

- Example 1:
  Equivalent discrete model for scalar system

$$\dot{x} = -ax + bu,\ \ y = x;\ \ G(s) = b/(s+a)$$

$$\Phi = e^{-ah},\ \Psi = \int_0^h e^{-a\sigma} d\sigma = -\frac{1}{a}\left(e^{-a\sigma}\right)\Big|_0^h = \left(1-e^{-ah}\right)/a;\quad \Gamma = b\Psi = \left(1-e^{-ah}\right)b/a$$

$$x(k+1) = e^{-ah}x(k) + \left[\left(1-e^{-ah}\right)/a\right]bu(k);\qquad y(k) = x(k)$$

$$\tilde{G}(z) = \frac{\Gamma}{z-\Phi}$$

$$\tilde{G}(z) = \frac{b\left(1-e^{-ah}\right)/a}{z-e^{-ah}} = \frac{z^{-1}\left(1-e^{-ah}\right)(b/a)}{1-z^{-1}e^{-ah}}$$

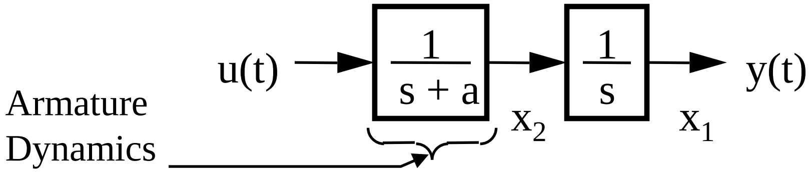
→ Note omnipresent one unit (h) delay in $\tilde{G}$ (z) ($b_0 = 0$).

Example: Second Order System

- Example 2: $\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & -a\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u(t)$ $\qquad \lambda_1 = 0,\ \lambda_2 = -a$

$$y(t) = [1 \quad 0]\underline{x}(k) = x_1(k)$$

This is typical of a model for a motor.



x_2 = shaft RPM (rad/sec)
x_1 = shaft rotation∡ (rad)

Analytic approach for arbitrary *a*:

$$\Phi = e^{Ah} = L^{-1}\left[(sI - A)^{-1}\right]\Big|_{t=h} = L^{-1}\begin{bmatrix}\frac{1}{s} & \frac{1}{s(s+a)}\\ 0 & \frac{1}{s+a}\end{bmatrix}_{t=h} = \begin{bmatrix}1 & \frac{1}{a}\left(1-e^{-ah}\right)\\ 0 & e^{-ah}\end{bmatrix} \quad \text{Eigenvalues } 1,\ e^{-ah}$$

$$\Psi = \int_0^h e^{A\sigma}d\sigma = \begin{bmatrix}h & \frac{1}{a}\left[h + \frac{1}{a}\left(e^{-ah}-1\right)\right]\\ 0 & \frac{1-e^{-ah}}{a}\end{bmatrix}; \quad \Gamma = \Psi B = \begin{bmatrix}a^{-1}\left[h + a^{-1}\left(e^{-ah}-1\right)\right]\\ -a^{-1}\left(e^{-ah}-1\right)\end{bmatrix}$$

$\tilde{G}(z)$ = transfer function of equivalent discrete system, $C(zI - \Phi)^{-1}\Gamma$ (tedious via hand calculation!)

$$= \frac{\left(ah + e^{-ah} - 1\right)\left(z + \dfrac{1 - e^{-ah} - ahe^{-ah}}{ah + e^{-ah} - 1}\right)}{a^2(z-1)\left(z - e^{-ah}\right)}$$

Example 2a: Double Integrator System

Special case of Example 2 when a = 0 => $G(s) = 1/s^2$

We can consider lim as $a \to 0$ using L'Hospital's rule (messy), or redo problem for

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}; \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \qquad C = [\,1 \quad 0\,]$$

$$\Phi = e^{Ah} = \mathcal{L}^{-1}\left[(sI - A)^{-1}\right]\Big|_{t=h} = \mathcal{L}^{-1}\left[\begin{matrix} \frac{1}{s} & \frac{1}{s^2} \\ 0 & \frac{1}{s} \end{matrix}\right]\Bigg|_{t=h} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix}$$

$$\Psi = \int_0^h e^{A\sigma} d\sigma = \begin{bmatrix} h & h^2/2 \\ 0 & h \end{bmatrix}; \quad \Gamma = \Psi B = \begin{bmatrix} h^2/2 \\ h \end{bmatrix}$$

$$\underline{x}(k+1) = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} h^2/2 \\ h \end{bmatrix} u(k)$$

$$\tilde{G}(z) = [1 \quad 0] \begin{bmatrix} \frac{1}{z-1} & \frac{h}{(z-1)^2} \\ 0 & \frac{1}{z-1} \end{bmatrix} \begin{bmatrix} \frac{h^2}{2} \\ h \end{bmatrix}$$

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

Example 3: F-8 Aircraft Model - 1

$a)$ *continuous system model*

$$\underline{\dot{x}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1.5 & -1.5 & 0 & 0.0057 & 1.5 \\ -12 & 12 & -0.6 & -0.0344 & -12 \\ -0.852 & 0.290 & 0 & -0.014 & -0.29 \\ 0 & 0 & 0 & 0 & -0.730 \end{bmatrix} \underline{x} + \begin{bmatrix} 0 & 0 \\ 0.16 & 0.80 \\ -19 & -3 \\ -0.015 & -0.0087 \\ 0 & 0 \end{bmatrix} \underline{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1.1459 \end{bmatrix} d; \underline{y} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \underline{x}$$

$$G(s) = \frac{\begin{bmatrix} -19s^2 - 26.85s - 0.3425 & -3s^2 + 5.058s + 0.06823 \\ 0.16s^3 + 0.09817s^2 - 26.58s - 0.2847 & 0.8s^3 + 0.4912s^2 + 5.107s + 0.06238 \end{bmatrix}}{s^4 + 2.114s^3 + 12.93s^2 + 0.1503s + 0.009442};$$

$$G_d(s) = \frac{\begin{bmatrix} -13.75s^2 - 0.1811s \\ 1.719s^3 + 1.053s^2 + 0.0133s + 0.01082 \end{bmatrix}}{s^5 + 2.844s^4 + 14.47s^3 + 9.588s^2 + 0.1192s + 0.006892}$$

$$b)\ select\ h: h = \frac{0.2}{\|A\|_2} = 0.0095 \Rightarrow h = 0.01\text{sec}$$

$c)$ *Discrete system model*

$$\underline{x}(k+1) = \begin{bmatrix} 0.9994 & 5.958.10^{-4} & 9.968.10^{-3} & -1.705.10^{-6} & -0.0005943 \\ 0.01488 & 0.9851 & 7.447.10^{-5} & 5.656.10^{-5} & 0.01483 \\ -0.1187 & 0.1187 & 0.9934 & -3.3395.10^{-4} & -0.1183 \\ -8.496.10^{-3} & 2.876.10^{-3} & -4.244.10^{-5} & 0.9999 & -2.866.10^{-3} \\ 0 & 0 & 0 & 0 & 0.9927 \end{bmatrix} \underline{x}(k) + \begin{bmatrix} -9.477.10^{-4} & 0 -1.481.10^{-4} \\ 1.583.10^{-3} & 7.94.10^{-3} \\ -0.1893 & -0.02943 \\ -1.10.10^{-4} & -7.503.10^{-5} \\ 0 & 0 \end{bmatrix} \underline{u}(k) + \begin{bmatrix} -2.276.10^{-6} \\ 8.53.10^{-5} \\ -6.81.10^{-4} \\ -1.649.10^{-5} \\ 0.01142 \end{bmatrix} d(k)$$

Example 3: F-8 Aircraft Model - 2

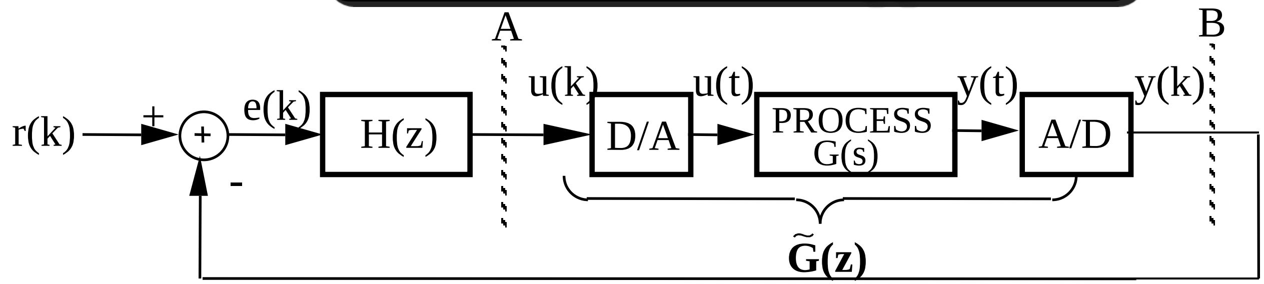
d) Discrete TFM

$$\tilde{G}(z) = \begin{bmatrix} \dfrac{-0.0009477\, z^3 + 0.0009366\, z^2 + 0.0009433\, z - 0.0009322}{z^4 - 3.978\, z^3 + 5.935\, z^2 - 3.936\, z + 0.9791} & \dfrac{-0.0001481\, z^3 + 0.0001525\, z^2 + 0.0001443\, z - 0.0001487}{z^4 - 3.978\, z^3 + 5.935\, z^2 - 3.936\, z + 0.9791} \\ \dfrac{0.001583\, z^3 - 0.004767\, z^2 + 0.004757\, z - 0.001574}{z^4 - 3.978\, z^3 + 5.935\, z^2 - 3.936\, z + 0.9791} & \dfrac{0.00794\, z^3 - 0.02377\, z^2 + 0.02372\, z - 0.007891}{z^4 - 3.978\, z^3 + 5.935\, z^2 - 3.936\, z + 0.9791} \end{bmatrix}$$

$$\tilde{G}_d(z) = \begin{bmatrix} \dfrac{-2.276\text{e-}006\, z^4 - 4.487\text{e-}006\, z^3 + 1.355\text{e-}005\, z^2 - 4.549\text{e-}006\, z - 2.243\text{e-}006}{z^5 - 4.971\, z^4 + 9.884\, z^3 - 9.827\, z^2 + 4.886\, z - 0.972} \\ \dfrac{8.53\text{e-}005\, z^4 - 0.0001707\, z^3 + 1.371\text{e-}006\, z^2 + 0.0001682\, z - 8.415\text{e-}005}{z^5 - 4.971\, z^4 + 9.884\, z^3 - 9.827\, z^2 + 4.886\, z - 0.972} \end{bmatrix}$$

- **MATLAB functions:**
  - **sysc=ss(A,B,C,D)**
  - **gs=tf(sysc)**
  - **sysd=c2d(sysc,h)**
  - **gz=tf(sysd)**
  - **gz=c2d(gs,h)**

Discrete System Equivalents – Xfer Function Approach



If the process to be controlled is described by a transfer function G(s), can we find $\tilde{G}(z)$ directly?

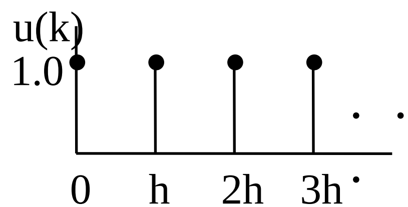
Indirect approach - (1) Write a state-space model for the process e.g., SCF or SOF or Balanced
(2) Find Φ, Γ using state variable approach
(3) Compute $\tilde{G}(z) = C(zI - \Phi)^{-1}\Gamma$

- Direct approach - Find Z-transform of unit pulse response $\equiv \tilde{G}(z)$, between points A and B.

First obtain the <u>step</u> response.

(1) Let u(k) be a unit step input

$u(z) = 1/(1-z^{-1})$.



(2) If the D/A Converter is a zero-order hold, then u(t) will be a pure step,
$u(t) = 1$ for $t > 0$ ==> $u(s) = 1/s$.

(3) Since the process is continuous, $y(s) = G(s)/s$ and $y(t) = \mathcal{L}^{-1}[G(s)/s]$.

(4) Sampling y(t) and taking the z-transform yields y(z)
$y(z) = Z\{\mathcal{L}^{-1}[G(s)/s]\}$ = z-transform of <u>step</u> response usual notation: $Z\{\mathcal{L}^{-1}[F(s)]\} \triangleq Z\{F(s)\}$.

(5) If u(k) = 1, the response is $(1-z^{-1})\, y(z) = (z-1)\, y(z)/z$

$$\tilde{G}(z) = (1-z^{-1})\, Z\{\mathcal{L}^{-1}(G(s)/s)\}$$

Discrete System Equivalents (Cont'd)

The resulting $\tilde{G}(z)$ must be the same as that obtained via state-space.

Example: $G(s) = \frac{a}{s+a} \Rightarrow \frac{G(s)}{s} = \frac{a}{s(s+a)} = \left[\frac{1}{s} - \frac{1}{s+a}\right]$

$$\mathcal{L}^{-1}\left[\frac{G(s)}{s}\right] = 1 - e^{-at} = y(t);\ \text{sampled}\ \ y(kh) = 1 - e^{-ahk}$$

$$\mathcal{Z}\{y(kh)\} = \frac{1}{1-z^{-1}} - \frac{1}{1-e^{-ah}z^{-1}} \Rightarrow (1-z^{-1})\mathcal{Z}\{y(kh)\} = 1 - \frac{1-z^{-1}}{1-e^{-ah}z^{-1}} = \frac{z^{-1}(1-e^{-ah})}{1-e^{-ah}z^{-1}}$$

The direct approach gets quite messy for $n > 2$. Preferred method is via state-space Φ, Γ then $\tilde{G}(z)$.

<u>Remember</u>! (1) The computer is "controlling" a discrete process with transfer function $\tilde{G}(z)$ not a continuous process $G(s)$.
(2) Zero-order D/A holds have been assumed (it is possible to re-do state-space approach with first order holds).

$\Rightarrow$ Of concern is the comparison of $G(s)\big|_{s=j\omega}$ vs. $\tilde{G}(z)\big|_{z=e^{j\omega h}}$.

Relationship Between G(s) and $\tilde{G}(z)$

How close is $\tilde{G}(z)\big|_{z=e^{sh}}$ to original G(s) when s = jω?

Can expect differences in both magnitude and phase

$$\tilde{G}(z) = (1 - z^{-1}) Z\left\{\frac{G(s)}{s}\right\} \Rightarrow \tilde{G}(z)\Big|_{z=e^{sh}} = (1 - e^{-sh})\left[\frac{G(s)}{s}\right]^*$$

$$\left[\text{Recall } F^*(s) \triangleq F(z)\Big|_{z=e^{sh}}, \text{ and relationship between } F^*(s) \text{ and } F(s),\ F^*(s) = \frac{1}{h}\sum_{n=-\infty}^{\infty} F(s - jn\omega_s)\right]$$

$$\Rightarrow \left[\frac{G(s)}{s}\right]^* \sim \frac{1}{h}\left[\frac{G(s)}{s} + \frac{G(s - j\omega_s)}{s - j\omega_s} + \frac{G(s + j\omega_s)}{s + j\omega_s}\right]$$

If $\omega << \omega_s/2 = \pi/h$, and $|G(j\omega \pm j\omega_s)| << 1$ then to a first approximation;

$$\left[\frac{G(s)}{s}\right]^* \sim \frac{1}{h}\left[\frac{G(s)}{s}\right] \text{ and } \tilde{G}(z)\Big|_{z=e^{sh}} \sim \underbrace{\left[\frac{1 - e^{-sh}}{sh}\right]}_{\text{Sample \& Hold} \div h} G(s)$$

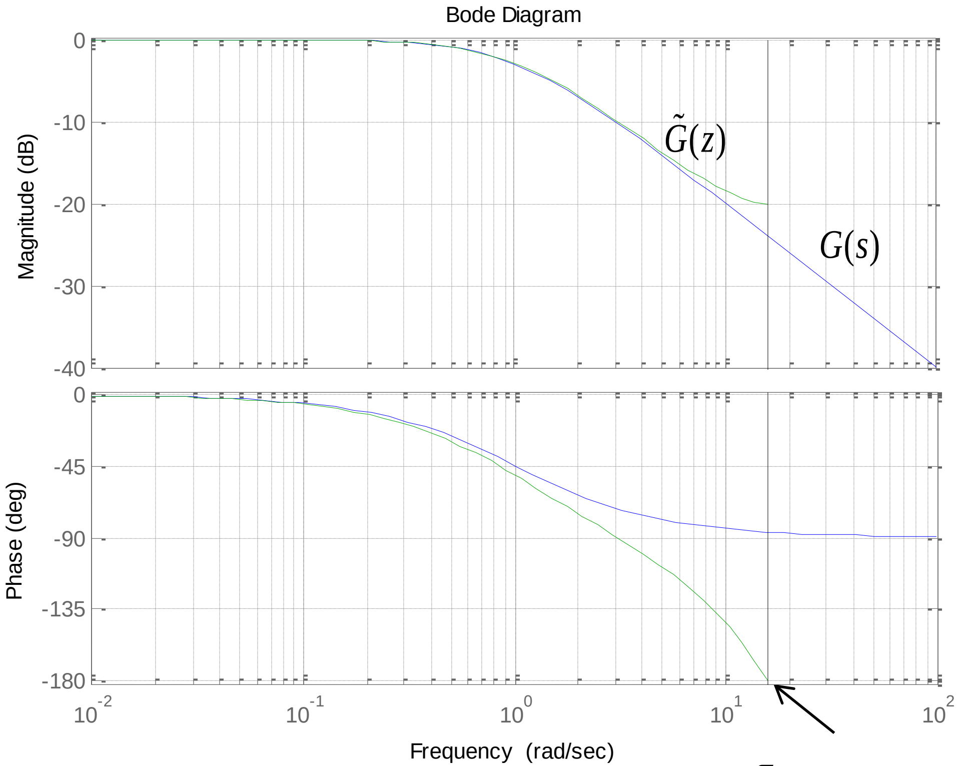
$$\tilde{G}(e^{sh})\Big|_{s=j\omega} = \underbrace{e^{-j\omega h/2}}_{h/2 \text{ sec Delay}} \underbrace{\left(\frac{\sin \omega h/2}{\omega h/2}\right)}_{\text{Magnitude Distortion}} G(j\omega)$$

⇒ To a crude first approximation, equivalent discrete transfer function is ~ original continuous one with some magnitude distortion <u>and an h/2 sec delay</u>, in the region $\omega << \pi/h$.

"Exact" comparison requires Bode plot of $G(j\omega)$ vs. $G(e^{j\omega h})$ – **c2d, bode**

Comparison of Continuous and Discrete Equivalent Bode Plots

$G(s) = 1/(s + 1)$; $h = 0.2 \Longrightarrow \tilde{G}(z) = 0.1813/(z - 0.8187)$



$$\frac{\pi}{h} = 5\pi = 15.71\, rad/\sec$$

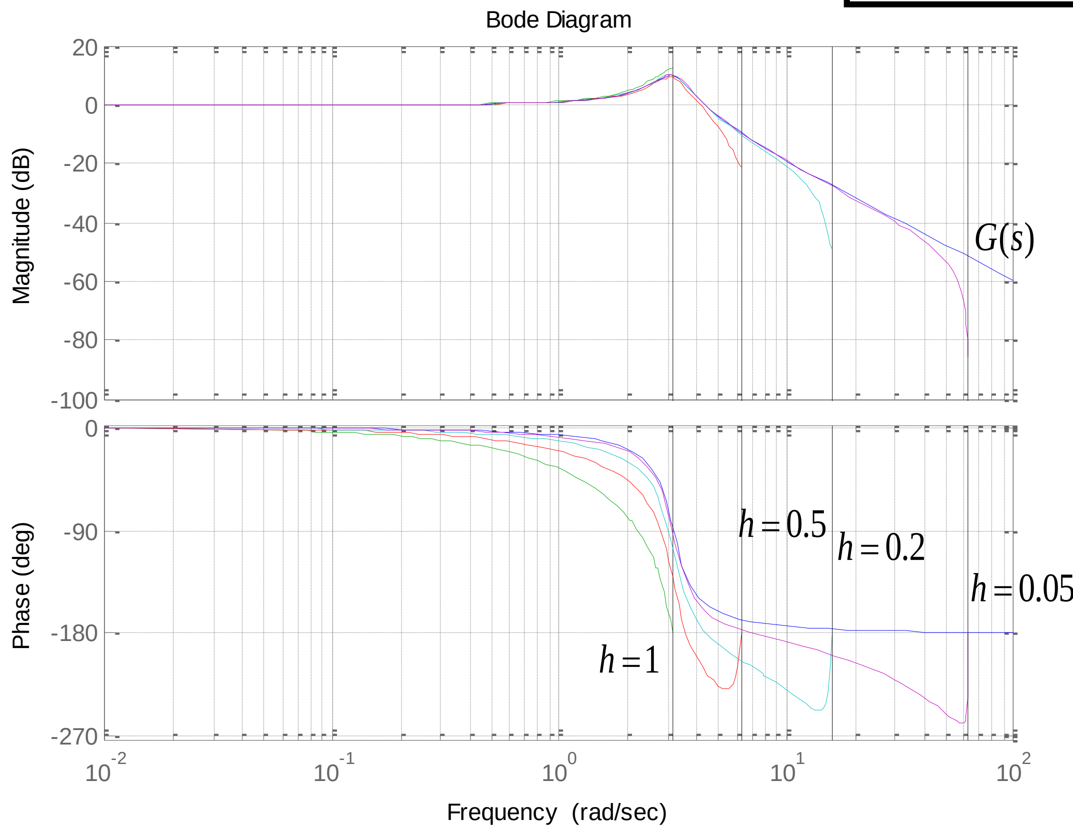
Code:

```
gs = tf([1],[1 1])
h=0.2
gz=c2d(gs,h)
bode(gs),grid
hold
bode(gz)
```

Effects of Time Step h on $\tilde{G}(z)$

$$G(s) = \frac{10}{s^2+s+10};\quad h_{max} = \frac{\pi}{|\lambda_{max}|} = \frac{\pi}{\sqrt{10}} \sim 1.0$$

Suggested $h = 0.05$



Anatomy of a Discrete Transfer Function

- Examine Bode plot structure of $G(e^{j\omega h})$ as a function of ω for $\omega > \pi/h$
  - For any discrete transfer function, G(z), letting $z = e^{j\omega h}$:

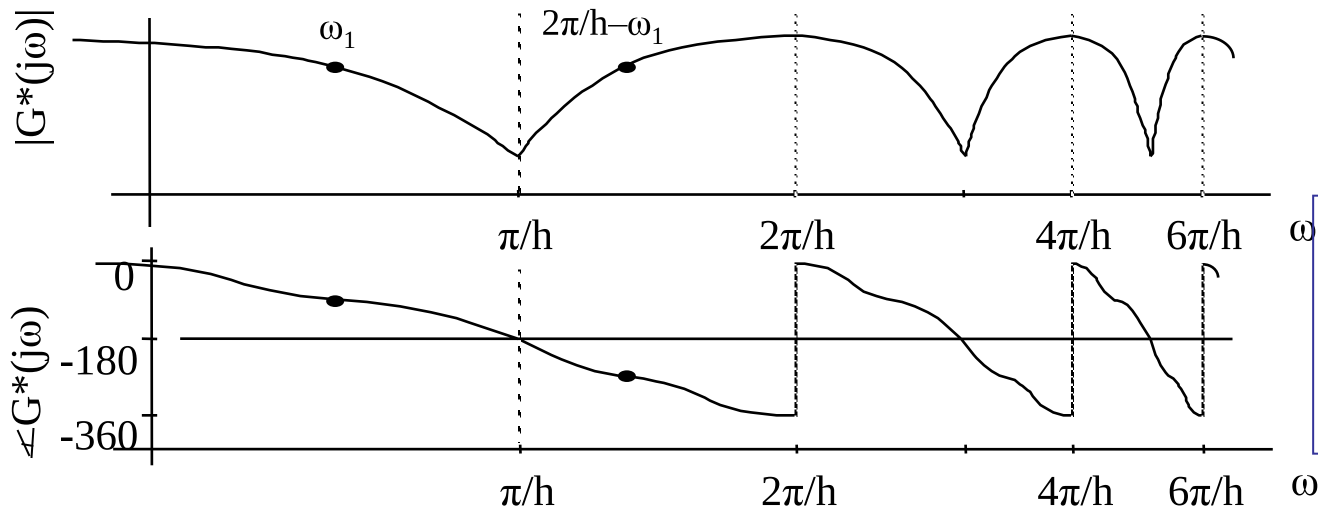
$$G^*(j\omega) \triangleq G(e^{j\omega h}) = G\left[e^{-j(2\pi/h-\omega)h}\right] = \text{conj}\left\{G\left[e^{j(2\pi/h-\omega)h}\right]\right\} \Rightarrow \begin{array}{l} \left|G^*(j\omega)\right| = \left|G^*(2\pi/h - j\omega)\right| \\ \measuredangle G^*(j\omega) = -\measuredangle G^*(2\pi/h - j\omega) \end{array}$$

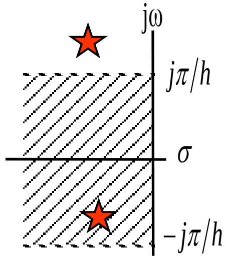
so, over the interval [0, 2π/h]:

$\left|G^*(j\omega)\right|$ has even symmetry about $\omega=\pi/h$

$\measuredangle G^*(j\omega)$ has odd symmetry about $\omega=\pi/h$ $\quad \left\{\measuredangle G^*(j\pi/h) = 0^\circ \text{ or } \pm 180^\circ \text{ since } e^{j\pi} = -1\right\}$

Over $\left[2k\frac{\pi}{h},\ 2(k+1)\frac{\pi}{h}\right]$, k=1, 2, …, $G^*(j\omega)$ is the same as that over $\left[0,\ \frac{2\pi}{h}\right]$





$\frac{\pi}{h} + \omega$ maps to

$-\frac{2\pi}{h} + \frac{\pi}{h} + \omega = -\frac{\pi}{h} + \omega$

$= -(\frac{\pi}{h} - \omega)$

==> If G(s) has a pole at s = 0, then $G^*(j\omega) \to \infty$ for $\omega = 2\pi k/h$, k = 1, 2, ...

Modeling a Process with Delay in Control, $\tau = Mh + \varepsilon$

If

$$\dot{\underline{x}} = A\underline{x} + \sum_{j=1}^{m} \underline{b}_j u_j(t-\tau_j);\ \underline{y} = C\underline{x} + \sum_{j=1}^{m} \underline{d}_j u_j(t-\tau_j);\ \underline{d}_j = \text{ column } j \text{ of } D \quad \text{or} \quad G(s) \rightarrow G(s)\ \mathrm{Diag}\ [\exp(-s\tau_j)]$$

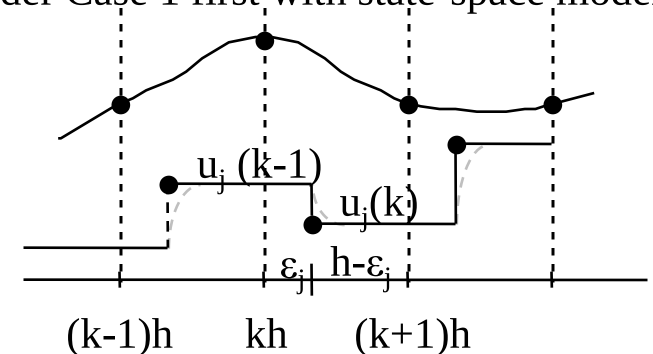
what is the appropriate discrete equivalent model?

Case 1: $M_j = 0$; $\tau_j = \varepsilon_j$ and $0 \le \varepsilon_j < h$
(typical model of computational delay)

Case 2: M_j = integer ≥ 1; $\tau_j = M_j h + \varepsilon_j$ and $0 \le \varepsilon_j < h$
(for cases when there is a large delay)

Delay Sources
- **Computational delays**
- **Transmission delays**
- **Plant delays**

Consider Case 1 first with state-space model.



Obtain $\underline{x}$ [(k+1)h] from $\underline{x}$(kh) and input to system over (kh, (k+1)h].

$$\underline{x}\left[(k+1)h\right] = e^{Ah}\underline{x}(kh) + \sum_{j=1}^{m}\int_{kh}^{(k+1)h} e^{A((k+1)h-\xi)}\underline{b}_j u_j(\xi)d\xi$$

$$= e^{Ah}\underline{x}(kh) + \sum_{j=1}^{m}\Big[\int_{kh}^{kh+\varepsilon_j} e^{A((k+1)h-\xi)}\underline{b}_j d\xi\, u_j(k-1) + \int_{kh+\varepsilon_j}^{(k+1)h} e^{A((k+1)h-\xi)}\underline{b}_j d\xi\, u_j(k)\Big]$$

$\sigma = (k+1)h - \xi$ $\qquad$ $\sigma = (k+1)h - \xi$

$$\underline{x}\left[(k+1)h\right] = e^{Ah}\underline{x}(kh) + \sum_{j=1}^{m}\Big[\int_{h-\varepsilon_j}^{h} e^{A\sigma}d\sigma\,\underline{b}_j u_j(k-1) + \int_{0}^{h-\varepsilon_j} e^{Ah}d\sigma\,\underline{b}_j u_j(k)\Big]$$

State Model for a Process with Fractional Delay

$$\underline{x}(k+1)=\Phi\underline{x}(k)+\sum_{j=1}^{m}\underline{\gamma}_{1j}u_j(k-1)+\sum_{j=1}^{m}\underline{\gamma}_{0j}u_j(k)=\Phi\underline{x}(k)+\Gamma_1\underline{u}(k-1)+\Gamma_0\underline{u}(k)$$

where $\Phi=e^{Ah}$; $\quad \underline{\gamma}_{0j}=\int_0^{h-\varepsilon_j}e^{A\sigma}d\sigma\,\underline{b}_j$; $\quad \underline{\gamma}_{1j}=\int_{h-\varepsilon_j}^{h}e^{A\sigma}d\sigma\underline{b}_j=e^{A(h-\varepsilon_j)}\int_0^{\varepsilon_j}e^{A\sigma}d\sigma\underline{b}_j$

To compute Φ, Γ_1, Γ_0: Do for j = 1,2,..,m

(1) Use c2d with (A, B, ε_j): obtain $e^{A\varepsilon_j}$ and $\Psi(\varepsilon_j)$;

(2) Use c2d with (A, B, h-ε): obtain $e^{A(h-\varepsilon_j)}$ and $\Psi(h-\varepsilon_j)$;

(3) $\underline{\gamma}_{0j}=\Psi(h-\varepsilon_j)\underline{b}_j,\ \underline{\gamma}_{1j}=e^{A(h-\varepsilon_j)}\Psi(\varepsilon_j)\underline{b}_j,\ \Phi=e^{A(h-\varepsilon_j)}e^{A\varepsilon_j}$ (need to do this for any one j).

- Augmented state model, $\underline{\chi}(k)\triangleq\begin{bmatrix}\underline{x}(k)\\ \underline{u}(k-1)\end{bmatrix}$ an $(n+m)-$ vector

  Then $\underline{\chi}(k+1)=\begin{bmatrix}\underline{x}(k+1)\\ \underline{u}(k)\end{bmatrix}=\begin{bmatrix}\Phi & \Gamma_1\\ 0 & 0\end{bmatrix}\underline{\chi}(k)+\begin{bmatrix}\Gamma_0\\ I_m\end{bmatrix}\underline{u}(k)$

  Output equation (as long as ε < h)

  $$\underline{y}(k)=C\underline{x}(k)+\{D\underline{u}(k-1)\}=[C\,|\,D]\underline{\chi}(k)$$

- Transfer function, $\tilde{G}(z)$

  $$\underline{x}(z) = (zI-\Phi)^{-1}[\Gamma_1 z^{-1}+\Gamma_0]\,\underline{u}(z)$$

  $$\underline{y}(z) = z^{-1}[\,C(zI-\Phi)^{-1}(z\Gamma_0+\Gamma_1)+D\,]\,u(z)$$

Invoke the previous SS $\rightarrow$ TFM routine with the augmented system. Alternately, compute $C(zI-\Phi)^{-1}\Gamma_0$ & $C(zI-\Phi)^{-1}\Gamma_1$. Compute numerator and denominator (recall: z $\rightarrow$ shift)

$c_{0kj}=b_{01kj}+d_{kj};\ c_{ikj}=b_{0i+1kj}+b_{1ikj}+d_{kj}a_i;$

$b_{0n+1kj}=0;\ a_{n+1}=0$

$\tilde{g}_{kj}(z)$ will have a form $\quad \tilde{g}_{kj}(z)=\dfrac{c_{0kj}z^n+c_{1kj}z^{n-1}+\cdots+c_{nkj}}{z\left(z^n+a_1z^{n-1}+\cdots+a_n\right)};\ k=1,2,..,p;\ j=1,2,..,m$

State Model for a Process with Large Delay

$\tau_j = M_j\, h + \varepsilon_j;\;\; M_j = \text{integer} \geq 1;\;\; 0 \leq \varepsilon_j < h\;;\; j=1,2,..,m$

Modeling approach same as for Case 1, but with added M_j time-step delay,

$$\underline{x}(k+1) = \Phi \underline{x}(k) + \sum_{j=1}^{m} [\underline{\gamma}_{1j} u_j(k-1-M_j) + \underline{\gamma}_{0j} u_j(k-M_j)]$$

$$\underline{y}(k) = C\underline{x}(k) + \sum_{j=1}^{m} \underline{d}_j u_j(k-1-M_j);\; \underline{d}_j = \text{ column } j \text{ of } D$$

- Transfer function matrix $M = \max_{1\leq j\leq m} M_j$

$$\tilde{G}(z) = \underbrace{\frac{1}{z}\left[C(zI-\Phi)^{-1}(z\Gamma_0+\Gamma_1)+D\right]}_{\text{previous result with M=0}} \text{Diag}[z^{-M_j}] = \frac{1}{z^M}\underbrace{\frac{1}{z}\left[C(zI-\Phi)^{-1}(z\Gamma_0+\Gamma_1)+D\right]}_{\text{previous result with M=0}} \text{Diag}[z^{M-M_j}],$$

$$\underline{\gamma}_{0j} = \int_0^{h-\varepsilon_j} e^{A\sigma} d\sigma \underline{b}_j,\quad \underline{\gamma}_{1j} = e^{A(h-\varepsilon_j)} \int_0^{\varepsilon_j} e^{A\sigma} d\sigma \underline{b}_j;\; j=1,2,..,m$$

- Augmented State Model (m=2): Define

$$\underline{\chi}(k) \triangleq [\underline{x}(k)\quad u_1(k-1-M_1)\quad u_1(k-M_1)\quad .\quad u_1(k-1)\quad u_2(k-1-M_2)\quad u_2(k-M_2)\quad .\quad u_2(k-1)]^T = n + m + \sum_{j=1}^{m} M_j \text{ vector}$$

$$\underline{\chi}(k+1) = \begin{bmatrix} \Phi & \underline{\gamma}_{11} & \underline{\gamma}_{01} & 0 & .. & 0 & \underline{\gamma}_{12} & \underline{\gamma}_{02} & 0 & .. & 0 \\ 0 & 0 & 1 & 0 & .. & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 0 & 1 & .. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .. & 0 & 0 & 0 & 0 & 0 & 0 \\ .. & 0 & 0 & 0 & .. & 1 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 0 & 0 & .. & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & .. & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & .. & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .. & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .. & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .. & 0 \end{bmatrix} \underline{\chi}(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ . & . \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ . & . \\ 0 & 1 \end{bmatrix} \underline{u}(k);\; \underline{y}(k) = [C\quad \underline{d}_1\quad 0\quad 0\quad 0\quad ..\quad 0\quad \underline{d}_2\quad 0\quad ..\quad 0]\,\underline{\chi}(k)$$

SISO State Model for a Process with Large Delay

$\tau = Mh + \varepsilon;\ \ M = \text{integer} \geq 1;\ \ 0 \leq \varepsilon < h$

Modeling approach same as for Case 1, but with added M time-step delay,

$$\underline{x}(k+1) = \Phi\, \underline{x}(k) + \Gamma_1\, u(k-1-M) + \Gamma_0\, u(k-M) \qquad (2.39)$$

$$y(k) = C\, \underline{x}(k) + \{d\, u(k-1-M)\}$$

- Augmented State Model,

$$\text{Define}\quad \underline{\chi}(k) \triangleq \begin{bmatrix} \underline{x}(k) \\ u(k-1-M) \\ u(k-M) \\ \cdot \\ u(k-1) \end{bmatrix} = n+1+M \ \text{ vector}$$

$$\underline{\chi}(k+1) = \left[\begin{array}{c|ccccc} \Phi & \Gamma_1 & \Gamma_0 & 0 & \cdots & 0 \\ \hline 0 & 0 & 1 & & & \\ 0 & & & 1 & & \\ \vdots & & & & & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{array}\right] \underline{\chi}(k) + \left[\begin{array}{c} 0 \\ \hline 0 \\ \vdots \\ 0 \\ 1 \end{array}\right] u(k)$$

$$y(k) = \left[c \,\middle|\, d \quad 0 \quad \cdots \quad 0\right] \underline{\chi}(k)$$

- Transfer function

$$\tilde{G}(z) = \frac{1}{z^M} \cdot \underbrace{\frac{1}{z}\left[C(zI-\Phi)^{-1}(z\Gamma_0 + \Gamma_1) + d\right]}_{\text{previous result with } M=0}, \qquad \Gamma_0 = \int_0^{h-\varepsilon} e^{A\sigma} d\sigma B,\ \ \Gamma_1 = e^{A(h-\varepsilon)} \int_0^{\varepsilon} e^{A\sigma} d\sigma B$$

Transfer Function Approach to Modeling a Process with Delay

Since $g_{kj}(s) \rightarrow g_{kj}(s)e^{-(M_j h+\varepsilon_j)}$, we have $\tilde{g}_{kj}(z) = (1-z^{-1}) Z\left\{\frac{g_{kj}(s)e^{-(M_j h+\varepsilon_j)}}{s}\right\}$

$$But, e^{-M_j hs} = z^{-M_j} \Rightarrow \tilde{g}_{kj}(z) = z^{-M_j}(1-z^{-1}) Z\left\{\frac{g_{kj}(s)e^{-\varepsilon_j s}}{s}\right\}$$

Approach -
(1) Form $\frac{g_{kj}(s)e^{-\varepsilon_j s}}{s}; 0 \le \varepsilon_j \le h$
(2) Take L^{-1} inverse Laplace
(3) Sample resulting time signal
(4) Take z-transforms

} Messy!

Example

$$G(s) = \frac{1}{s+a} e^{-Mhs} e^{-\varepsilon s} \quad \Rightarrow \quad \dot{x} = -ax + u(t-\tau)$$

$$\Phi = e^{-ah}; \quad \Gamma_0 = \int_0^{h-\varepsilon} e^{-a\sigma} d\sigma = \left[1-e^{-a(h-\varepsilon)}\right]/a; \quad \Gamma_1 = e^{-a(h-\varepsilon)} \int_0^{\varepsilon} e^{-a\sigma} d\sigma = e^{-a(h-\varepsilon)}\left(1-e^{-a\varepsilon}\right)/a$$

$$\tilde{G}(z) = \frac{1}{az^{M+1}}\left\{\frac{\left(1-e^{-a(h-\varepsilon)}\right)z + e^{-ah}\left(e^{a\varepsilon}-1\right)}{z-e^{-ah}}\right\}$$

Ex. $a = 1.0, \quad M = 2, \quad \varepsilon = 0.5, \quad h = 1$

$$\Rightarrow \quad \tilde{G}(z) = \frac{1}{z^3}\left\{\frac{\left(1-e^{-0.5}\right)z + e^{-1}\left(e^{0.5}-1\right)}{z-e^{-1}}\right\} = \frac{0.393(z+0.607)}{z^3(z-0.368)}$$

Note: In many applications the time-step is dictated by the on-line computational requirements.
$\Rightarrow \tau$ is often comparable to h.

Summary

1. **Digital Interfacing**
   - Signal Conditioning
   - A/D and D/A converters
2. **Signal Sampling and Data Reconstruction**
   - Impulse sampling model; Nyquist theorem; Aliasing and interpretation
   - Signal conditioning circuits
3. **Discrete Equivalents: State-Space Approach**
   - Discretization algorithm
4. **Discrete Equivalents: Transfer Function Approach**
   - Relation to original continuous system
5. **Model Modifications with Delay in Control**