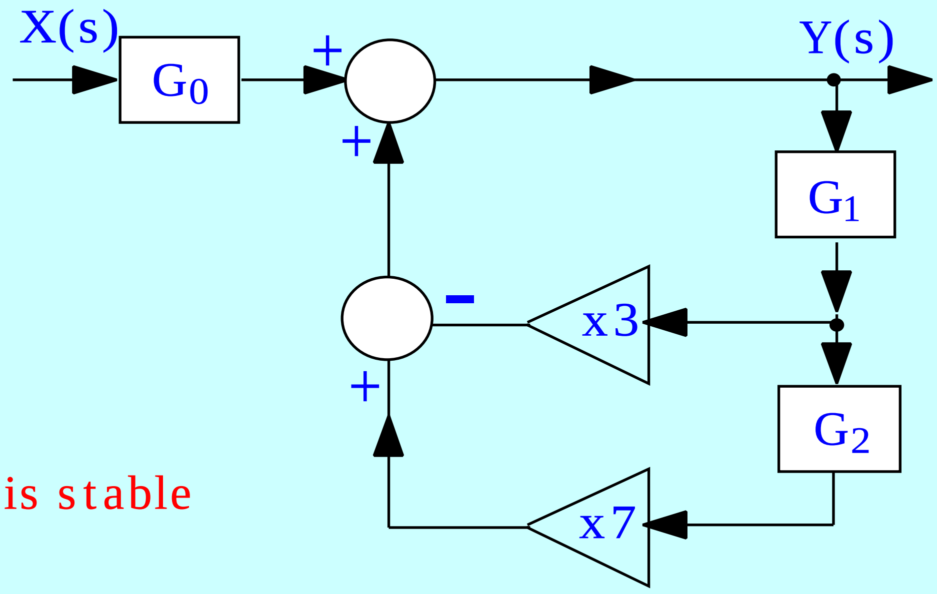
Stability

$$Y(s) = \frac{8X(s)}{(s+2)^2(s-2)}$$



•System has 2 poles: points where Y(s) -> ∞

at s = +2 and s = -2

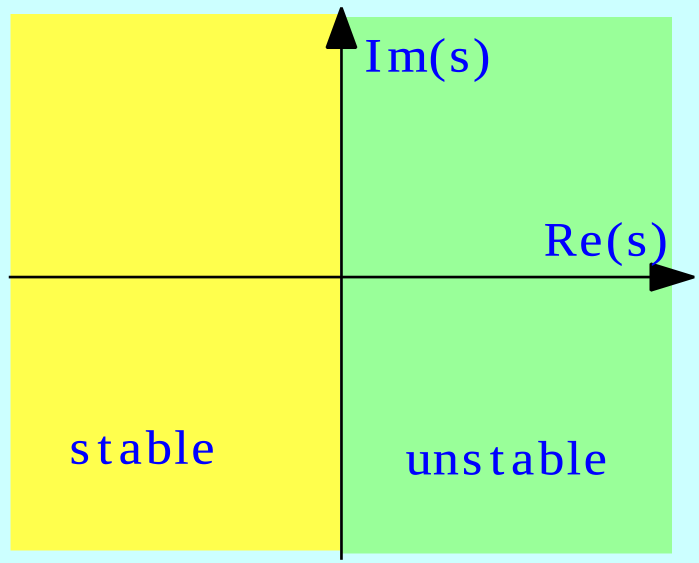
•If all poles are in region where s < 0, system is stable

in Fourier language s = jω

can only have positive frequencies, ie s > 0

so this system is unstable

will see why from solution



•Pole location s could have imaginary part

=> oscillatory solution

Response to step

- $x(t) = u(t) = 1$, for $t > 0$ so $X(s) = 1/s$

$$Y(s) = \frac{8X(s)}{(s+2)^2(s-2)} = \frac{8}{s(s+2)^2(s-2)}$$

$$= \frac{A}{s} + \frac{B}{(s+2)} + \frac{C}{(s+2)^2} + \frac{D}{(s-2)}$$

- Solve by expressing as partial fractions
- Find A, C, D by taking limit s -> a of $(s+a)^N Y(s)$ N is highest power term
- Find A by multiplying by s

RHS $$\underbrace{\text{limit}}_{s->0} \ldots sY(s) = A + \frac{Bs}{(s+2)} + \frac{Cs}{(s+2)^2} + \frac{Ds}{(s-2)} = A$$

LHS $$\underbrace{\text{limit}}_{s->0} \ldots sY(s) = \frac{8}{(s+2)^2(s-2)} = \frac{8}{4(-2)} = -1$$

$$A = -1$$

- Find C by multiplying by $(s+2)^2$

RHS $$\underbrace{\text{limit}}_{s->-2} \ldots (s+2)^2 Y(s) = A(s+2)^2 + B(s+2) + C + \frac{D(s+2)^2}{(s-2)} = C$$

$$C = 1$$

LHS $$\underbrace{\text{limit}}_{s->-2} \ldots (s+2)^2 Y(s) = \frac{8}{s(s-2)} = \frac{8}{(-2)(-4)} = 1$$

similarly $D = 1/4$

Step response... continued

$$Y(s) = \frac{8X(s)}{(s+2)^2(s-2)} = \frac{8}{s(s+2)^2(s-2)} \qquad = \frac{A}{s} + \frac{B}{(s+2)} + \frac{C}{(s+2)^2} + \frac{D}{(s-2)}$$

- Find B by multiplying by $(s+2)^2$, differentiate, then take limit

RHS

$$\frac{d}{ds}(s+2)^2 Y(s) = \frac{d}{ds}\left[\frac{8}{s(s-2)}\right] = 8\left[\frac{-1}{s^2(s-2)} + \frac{-1}{s(s-2)^2}\right]$$

$$\underbrace{\text{limit}}_{s->-2}\left(8\left[\frac{-1}{s^2(s-2)} + \frac{-1}{s(s-2)^2}\right]\right) = 8\left[\frac{-1}{4(-4)} + \frac{-1}{(-2)(-4)^2}\right] = \frac{3}{4}$$

LHS

$$\underbrace{\text{limit}}_{s->-2}\ldots\frac{d}{ds}(s+2)^2 Y(s) = \frac{d}{ds}B(s+2) = B$$

$$B = \frac{3}{4}$$

- now have the solution in s

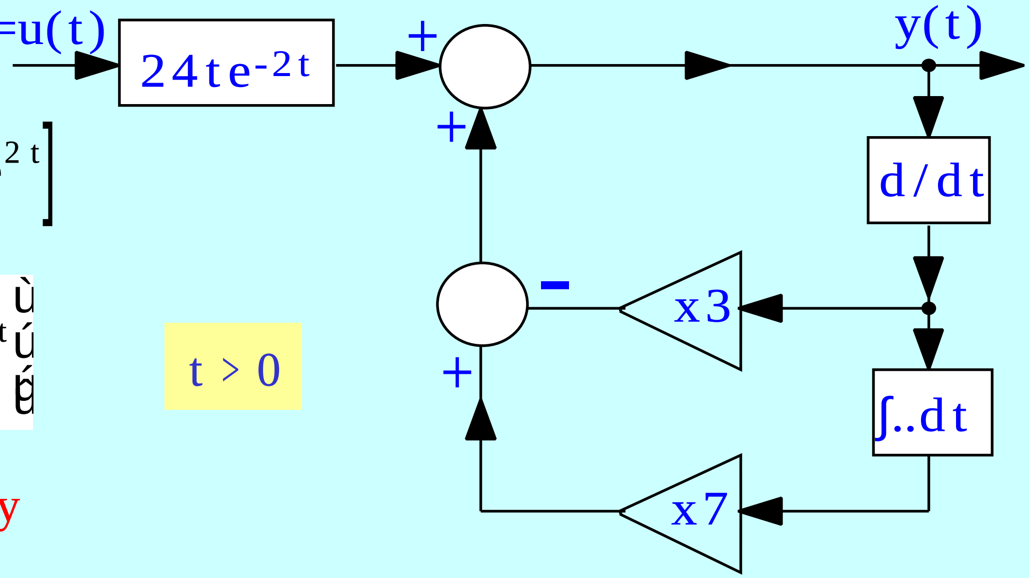
$$Y(s) = \frac{1}{4}\left[\frac{-4}{s} + \frac{3}{(s+2)} + \frac{4}{(s+2)^2} + \frac{1}{(s-2)}\right]$$

Finally... solution

$$Y(s) = \frac{1}{4}\left[\frac{-4}{s} + \frac{3}{(s+2)} + \frac{4}{(s+2)^2} + \frac{1}{(s-2)}\right]$$

•Recall $F(s) = \frac{n!}{(s+a)^{n+1}}$ is LT of $f(t) = t^n e^{-at}$

•and $F(s) = \frac{1}{s}$ is LT of $u(t)$ = unit step



$$y(t) = \frac{1}{4}\left[-4u(t) + 3e^{-2t} + 4te^{-2t} + e^{2t}\right]$$

$$y(t) = -u(t) + \left[\frac{3}{4}e^{-2t} + te^{-2t} + \frac{1}{4}e^{2t}\right]$$

$t > 0$

•Can now see the reason for instability

term with e^{2t}

•By the way: this problem could equally well be solved with Fourier

z transforms

- Laplace transform applies to continuous signals in time domain

  Extend idea to discrete, sampled signals

- from Laplace Transform definition

  $F(s) = \int_0^{\infty} f(t).e^{-st}.dt,$

  sample waveform f(t) at intervals Δt

  sampled signal

  $f(t) = f(0), f(\Delta t), f(2\Delta t), f(3\Delta t), f(4\Delta t),\ldots, f(n\Delta t), \ldots$

  We will assume functions for which f = 0 for t < 0

- transform f(t)

  $F(s) = \sum_{n=0}^{\infty} f(n\Delta t).e^{-sn\Delta t}$

  Define $z = e^{s\Delta t}$

  $F(z) = \sum_{n=0}^{\infty} f(n\Delta t).z^{-n} = \sum_{n=0}^{\infty} f_n.z^{-n}$

  $ZT[f] = F(z)$

  each term in z^{-1} represents a delay of Δt, ie z^{-n} => delay of $n\Delta t$

Examples

•(1) $f_n = \delta_0$ = 1 0 0 0 0 ...

$$F(z) = 1$$

•(2) $f_n = 1$ represents a step function, since f(t) = 0 for all t < 0

$$F(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + \ldots + z^{-n} + \ldots$$

Should recognise geometric series, or binomial expansion of $(1-x)^{-1}$

$$F(z) = \frac{1}{(1 - z^{-1})}$$

•(3) $f_n = e^{-na}$ $a = \Delta t / \tau$ τ = time constant Δt = sampling interval

$$F(z) = 1 + e^{-a}z^{-1} + e^{-2a}z^{-2} + e^{-3a}z^{-3} + e^{-4a}z^{-4}\ldots \ldots + e^{-na}z^{-n} + \ldots$$

$$F(z) = \frac{1}{(1 - e^{-a}z^{-1})}$$

•(4) $f_n = 1 - e^{-na}$

$$F(z) = \frac{1}{(1 - z^{-1})} - \frac{1}{(1 - e^{-a}z^{-1})} = \frac{z^{-1}(1 - e^{-a})}{(1 - z^{-1})(1 - e^{-a}z^{-1})}$$

Digital filters

- What is the output if every previous input sample is summed with weight e^{-na}?

  ie compute $g_m = \sum_n^m e^{-na} f_n$

- Convolution in time, so becomes z-transform multiplication $G(z) = H(z)F(z)$

$$H(z) = ZT[e^{-na}] = \frac{1}{(1 - e^{-a}z^{-1})} \qquad G(z) = \frac{F(z)}{(1 - e^{-a}z^{-1})}$$

$$F(z) = (1 - e^{-a}z^{-1})G(z) = G(z) - G(z)e^{-a}z^{-1}$$

$$f_n = g_n - e^{-a}g_{n-1} \qquad \text{or} \qquad g_n = f_n + e^{-a}g_{n-1}$$

- ie - Latest value of output sampled waveform

  = current input sample + previous output sample x e^{-a}

- Impulse response corresponding to H(z)?

  $h(t) = e^{-n\Delta t/\tau}$ which is impulse response of Low Pass Filter (Problems 2, No 8)

- Conclusion

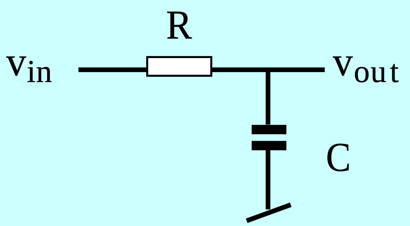
  Low pass digital filter can be made using just two samples $g_n = f_n + e^{-a}g_{n-1}$

  well suited for simple digital processor operation

Step response of previous digital filter

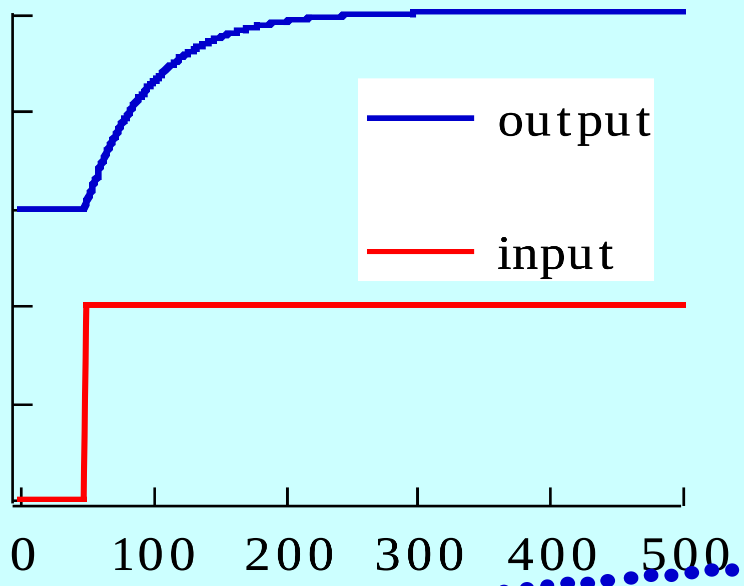
- To be more exact

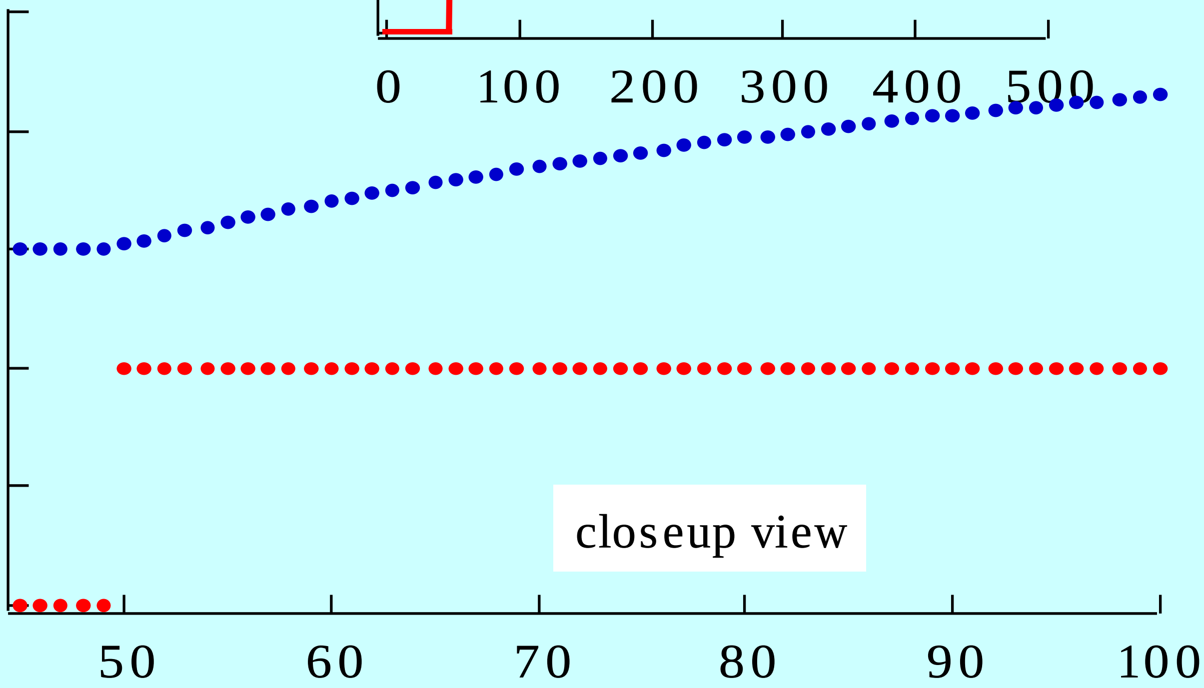
Impulse response of Low Pass filter



$$h(t) = \frac{1}{\tau} e^{-t/\tau}$$

$$g_n = \frac{f_n}{\tau} + e^{-a} g_{n-1}$$





Deconvolution

- Suppose a signal has been filtered by a system with a known response

How to recover the input signal from the samples?

In t: input = f output = g, filter impulse response = h

In z: F(z) G(z) and H(z)

Since g(t) = f(t)*h(t), then $G(z) = F(z)H(z)$

so to recover input $F(z) = H^{-1}(z)G(z)$

- Low pass filter again

$$H(z) = \frac{1}{(1 - e^{-a}z^{-1})}$$

Inverse filter $H^{-1}(z) = (1 - e^{-a}z^{-1})$

$$f_n = g_n - e^{-a}g_{n-1}$$

terms in z^{-1} identify which delayed samples to use

- This time g_n are the measured samples, f_n the result of digital processing

An example of a deconvolution filter

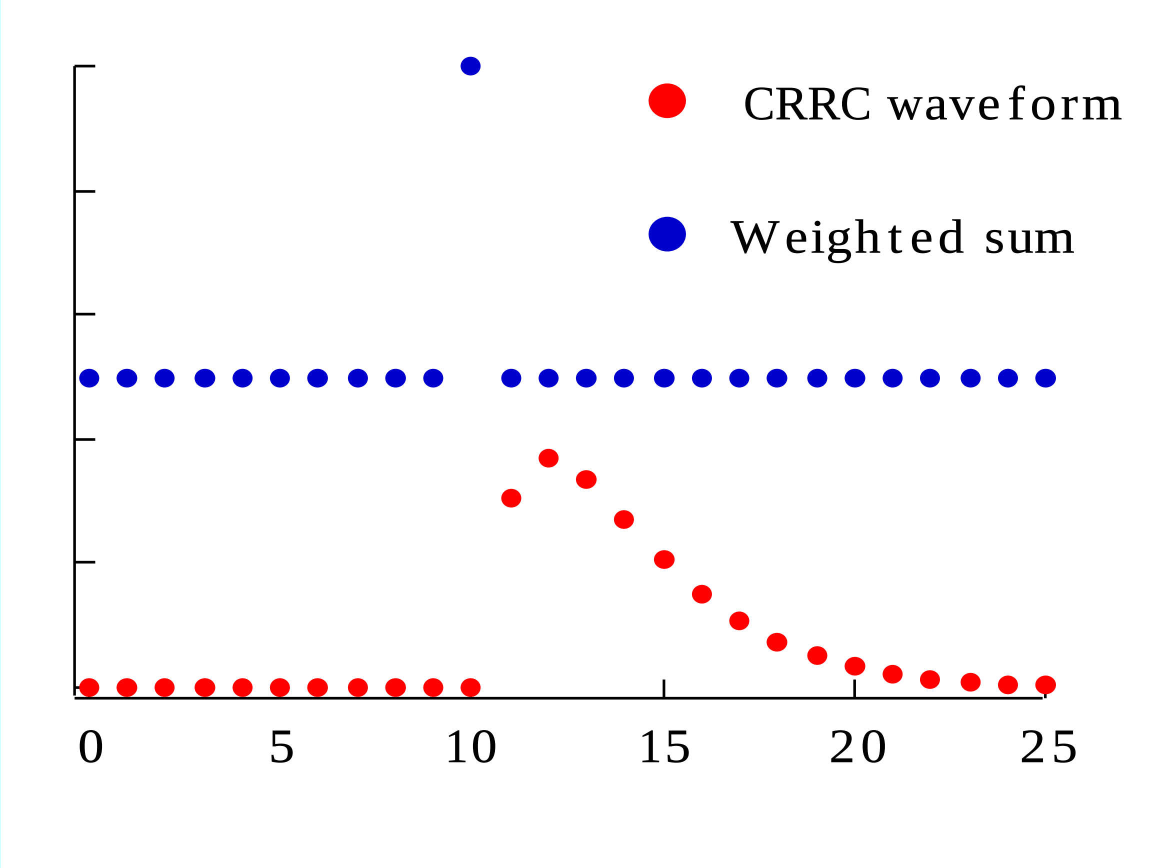
- Integrator + CR-RC bandpass filter waveform

form weighted sum of pulse samples

$g_n = w_1.f_{n+1} + w_2.f_n + w_3.f_{n-1}$

for correct choice of w_i

(Problems 6)



- Note g_n needs f_{n+1}

doesn't violate causality if data are digital, in storage - or could simply delay output

in applications such as image processing, causality does not apply

CMS experiment at Large Hadron Collider

- uses this deconvolution filter

  implemented in CMOS IC

  beam crossings at 40MHz (Δt = 25ns)

  many events per crossing

  small number of weights

  implemented as analogue calculation

  process only data which are to be read out