Lecture 7 Part 1: Implementation of digital controllers

- Computational delay
- Sensor interface
  - Prefiltering
- Actuator interface
  - Saturation, wind-up
- Operator interface
  - Real time programming
- Numerics

Controller templates

State space representation with explicit observer

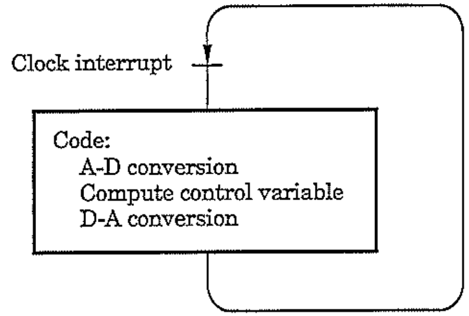
$$
\begin{aligned}
\hat{x}(k|k) &= \hat{x}(k|k-1) + K(y(k) - \hat{y}(k|k-1)) \\
u(k) &= L(x_m(k) - \hat{x}(k|k)) + Du_c(k) \\
\hat{x}(k+1|k) &= \Phi\hat{x}(k|k) + \Gamma u(k) \\
x_m(k+1) &= f(x_m(k), u_c(k)) \\
\hat{y}(k+1|1) &= C\hat{x}(k+1|k)
\end{aligned}
$$

General state-representation

$$
\begin{aligned}
x(k+1) &= Fx(k) + Gy(k) + G_c u_c(k) \\
u(k) &= Cx(k) + Dy(k) + D_c u_c(k)
\end{aligned}
$$

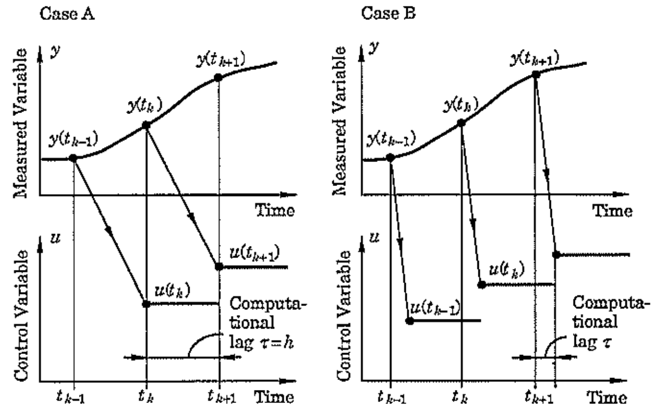
Implementation

```
Procedure Regulate
begin
  Adin y uc
  u := u1 + D*y + Dc*uc
  Daout u
  x := F*x + G*y + Gc*uc
  u1 := C*x
end
```



Send out control signal as soon as possible

Computational delay



Minimize computational delay

Computing time should be included in process model.

Keep it constant!

Prefilters

Eliminate all frequencies above the Nyquist frequency

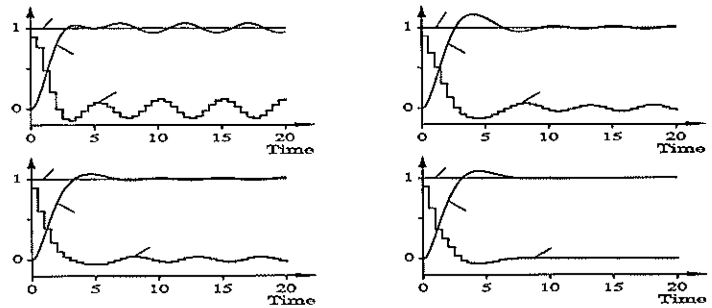
- Analog filters
  - 2 – 6 th order Bessel or Butterworth filters
  - Difficulty to change with h
- Digital filters
  - Sample fast and make a digital filter
  - Useful for long sampling intervals and changing sampling periods

Must usually include the filter in the design

Effect of anti-aliasing filter

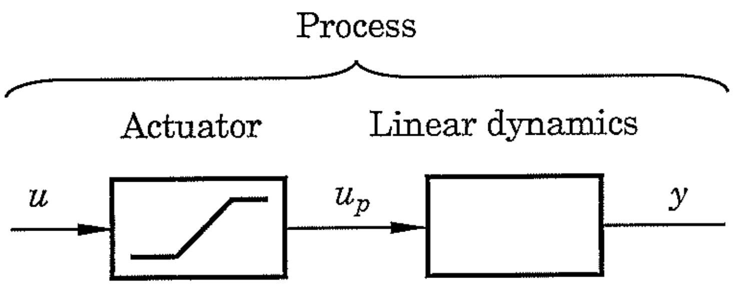
$$G(s) = \frac{1}{s(s+1)} \qquad y_m(t) = y(t) + 0.1\sin(\omega_d t) \qquad \omega_d = 11.3$$

Pole placement ($h = 0.5$) and 4th order Bessel filter cutting at ω_B



a) $\omega_B = 25$, b) $\omega_B = 6.28$, c) $\omega_B = 6.28$, compensation for delay of $0.7h$ d) $\omega_B = 2.51$, compensation for delay of $1.7h$

Nonlinear actuators

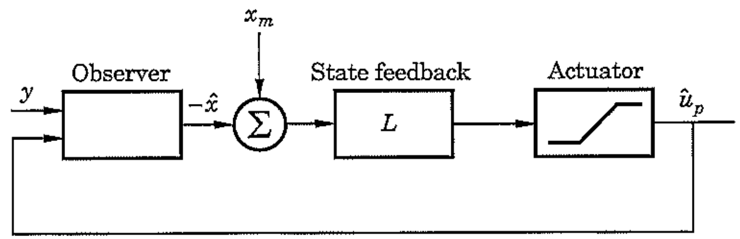


Feedback loop broken if saturation

Is the controller stable?

Controller states may wind-up

Anti-windup

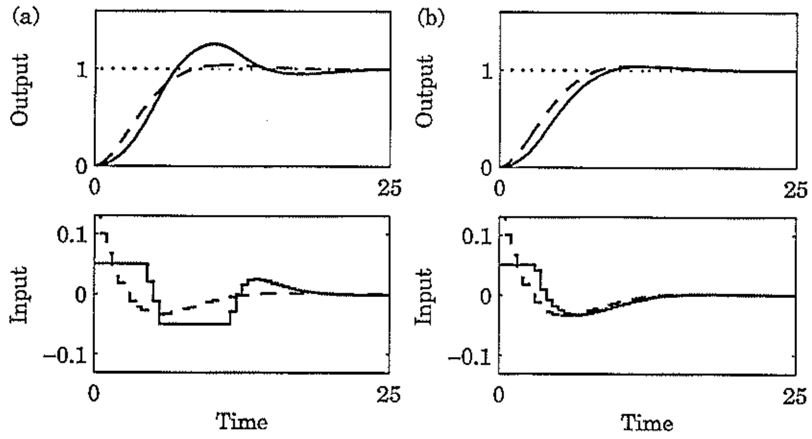


Measure or estimate the actual input u_p

$$\hat{x}(k+1) = (\Phi - KC)\hat{x}(k) + Ky(k) + \Gamma\hat{u}_p(k)$$
$$\hat{u}_p(k) = \text{sat}(v(k))$$

This type of "tracking" can be used for any type of controller

Antiwindup for double integrator



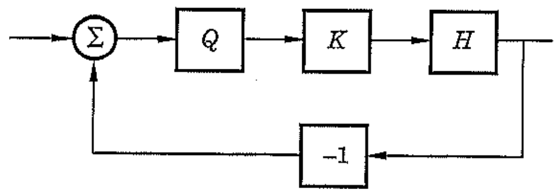
Numerics

- Word-length, Computer, A/D and D/A converters
- Fixed or floating point computations (IEEE standard)
- Special hardware, DSP?
- Influence of noise and quantization
- Choice of realization

Effects of roundoff and quantization

- Nonlinear phenomena
- Limit cycles and/or bias
- Analysis tools
  - Nonlinear analysis
  - Describing function approximation
  - Model quantization as stochastic processes

Effect of roundoff



$$H(z) = \frac{0.25}{(z-1)(z-0.5)}$$

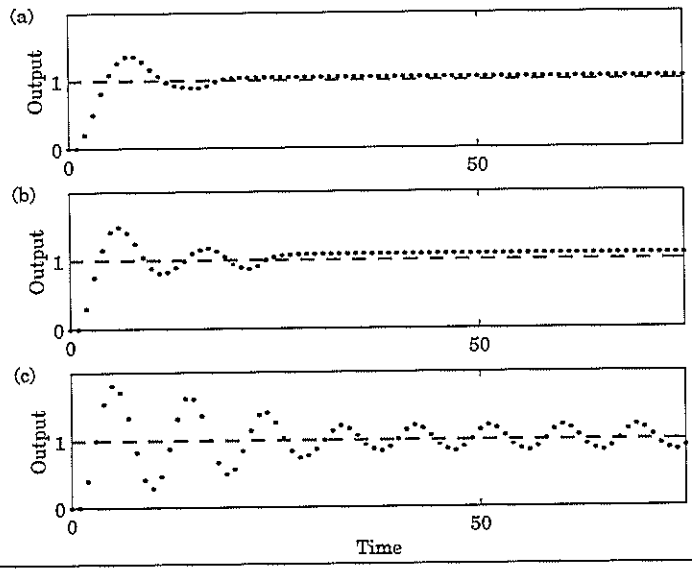
Without quantization: Stable for $K < 2$

With quantization

$K = 0.8$

$K = 1.2$

$K = 1.6$

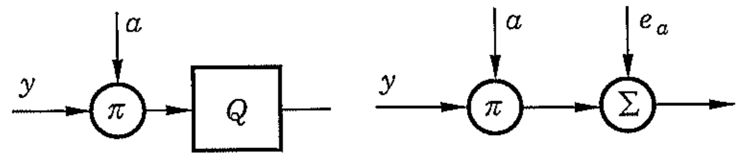


Linear models for quantization and roundoff

Assume

- The input varies sufficiently much
- Sufficiently many bits in the converter (≥ 8)

then the error can be modeled as independent rectangular distributed noise with $\sigma^2 = \delta^2/12$



Realization of controllers

Want to realize (implement) the controller

$$(1 + a_1 q^{-1} + a_2 q^{-2} + \cdots + a_n q^{-n})\, y(k) \quad = (b_0 + b_1 q^{-1} + \cdots + b_m q^{-m})\, u(k)$$

Some different realizations are:

- Companion form
- Direct form
- Series – Jordan
- Parallel form – Diagonal
- Lattice or ladder form
- δ-operator form

The realizations are more or less sensitive to coefficient errors

How do roots vary with oeefficients?

$$A(z) = (z - p_1)\cdots(z - p_n) = z^n + a_1 z^{n-1} + \cdots + a_n$$

A change $a_i \to a_i + \delta a_i$ gives $p_i \to p_i + \delta p_i$

$$0 \quad = \quad A(p_k \ + \ \delta p_k, a_i \ + \ \delta a_i) \quad \approx \quad \underbrace{A(p_k, a_i)}_{=0} + \left.\frac{\delta A}{\delta z}\right|_{p_k} \delta p_k \ + \ \left.\frac{\delta A}{\delta a_i}\right|_{p_k} \delta a_i \ + \ \cdots$$

Because
$$\left.\frac{\delta A}{\delta a_i}\right|_{z=p_k} = p_k^{n-i} \qquad \left.\frac{\delta A}{\delta z}\right|_{z=p_k} = \prod_{j \neq k}(p_k - p_j)$$

For a root p_k with multiplicity m then

$$\delta p_k \approx -\frac{p_k^{n-i}}{\prod_{j \neq k}(p_k - p_j)} \, (\delta a_i)^{1/m}$$

Most sensitive close or multiple roots. a_n is most sensitive

Ill-conditioned realizations

Don't use "direct" polynomial form or companion forms.
For $H(z) = b^4/(z+a)^4$, this would mean

```
begin
  x4:= x3
  x3:= x2
  x2:= x1
  x1:= -4*a*x1 -6*a^2*x2 -4*a^3*x3 -a^4*x4 +u
  y:= b^4*x4
end
```

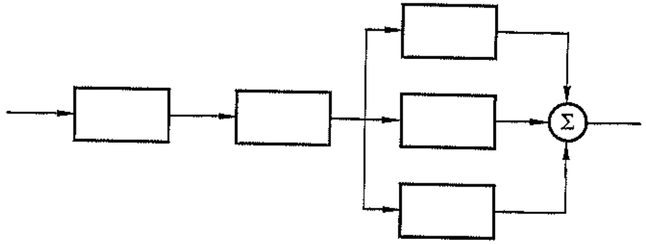
Most sensitive when many equal roots:

$$(z-p)^n - \varepsilon = 0 \quad \Rightarrow \quad z = p + \varepsilon^{1/n}$$

Short $h \Rightarrow$ All poles around $z = 1$

Well-conditioned realizations

Use series and parallel connections of first and second order blocks



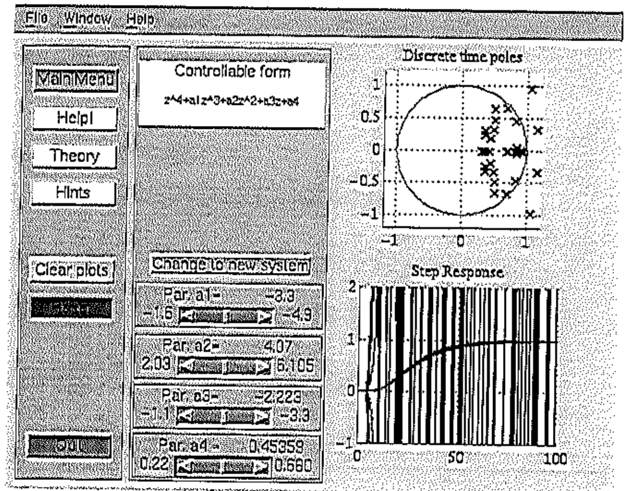
Series or (almost) Jordan form:

```
begin
  x4:= - a*x4 + b*u
  x3:= - a*x3 + b*x4
  x2:= - a*x2 + b*x3
  x1:= - a*x1 + b*x2
  y:= x1
end
```

Example – Controllable form

$$H(q) = \frac{K}{q^4 - 3.3z^3 + 4.07z^2 - 2.223z + 0.45359}$$

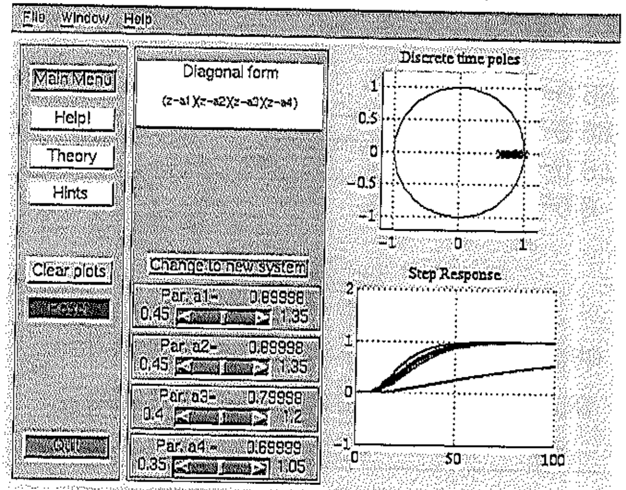
Vary 1% in the coefficients.



Example – Series form

$$H(q) = \frac{K}{(q-0.9)(q-0.9)(q-0.8)(q-0.7)}$$

Vary 1% in the coefficients.



Short sampling interval modification

If h is small, then the matrix Φ in the state update

$$x(kh+h) = \Phi x(kh) + \Gamma y(kh)$$

(shift operator form) is almost equal to identity. Round-off errors in Φ can have drastic effects.

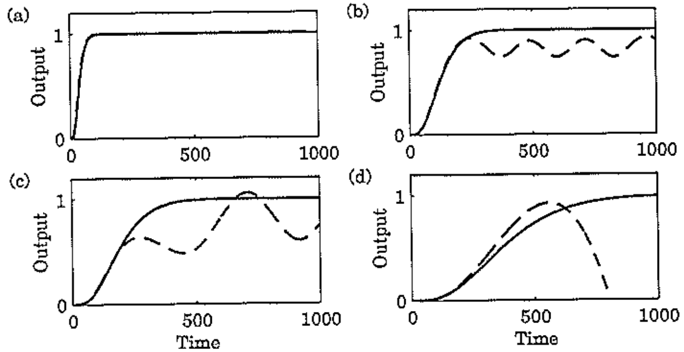
Instead, use the modified equation

$$x(kh+h) = x(kh) + (\Phi - I)x(kh) + \Gamma y(kh)$$

This is called δ-operator form.
Both $\Phi - I$ and Γ are proportional to h.

Example — Three realizations

$H(z) = \dfrac{b^4}{(z+a)^4}$ Chopped to 7 significant figures in Matlab



(a) $a = -0.9$
(b) $a = -0.97$
(c) $a = -0.98$
(d) $a = -0.99$

Shift-operator controllable form (dashed)
Jordan form (dash-dotted)
δ-operator controllable form (full)

Code for modified Jordan form

$(b = 1 + a)$

```
begin
  x4:= x4 + b*(u - x4)
  x3:= x3 + b*(x4 - x3)
  x2:= x2 + b*(x3 - x2)
  x1:= x1 + b*(x2 - x1)
  y:= x1
end
```

Code for δ-operator controller form

```
begin
  x4:= x4+x3
  x3:= x3+x2
  x2:= x2+x1
  x1:= x1- b1 *x1 - b2*x2 - b3*x3 - b4*x4 + u
  y:= b4*x4
end
```

$b_1 = 4b$, $b_2 = 6b^2$, $b_3 = 4b^3$, $b_4 = b^4$

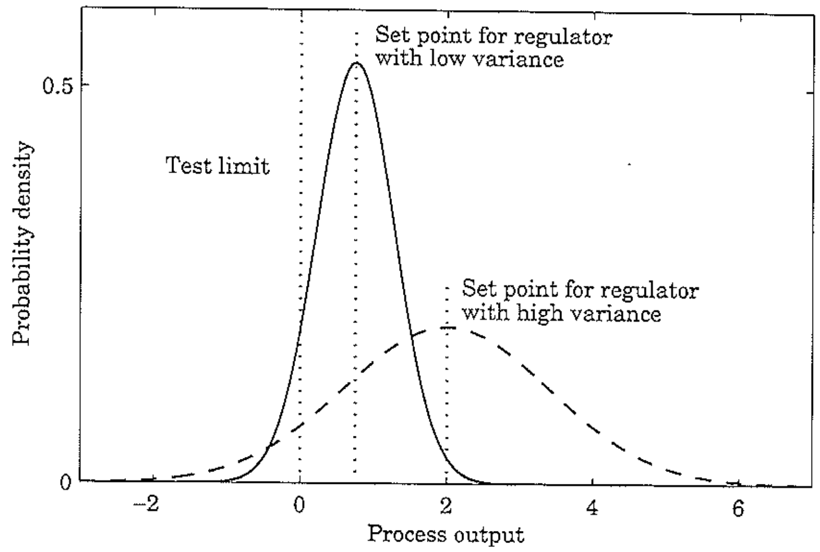
Summary

- Organizing of the code
- Anti-aliasing filters
- Nonlinearities
- Anti reset-windup
- Roundoff and quantization
- Coefficient sensitivity

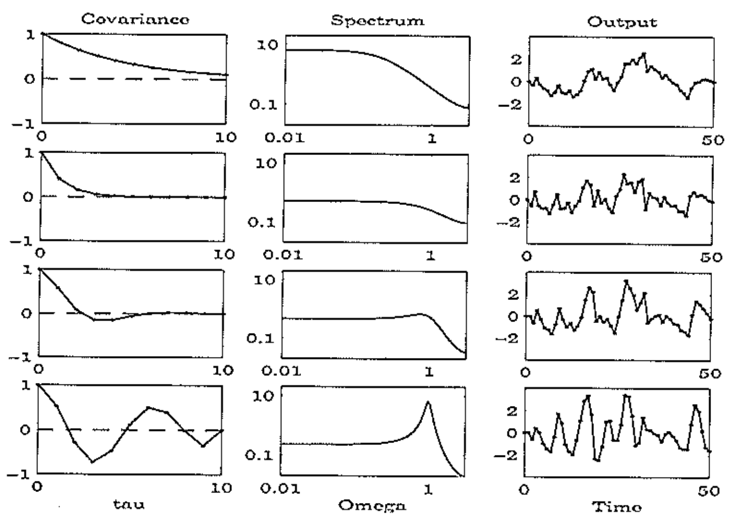
Lecture 7 Part 2: Stochastic disturbances

- State space models
- Input-output models
- Spectral factorization
- Continuous-time stochastic processes

Motivation



Covariance, spectral density, and realization



Error-correction: The spectra should be divided by 2π

Discrete-time white noise

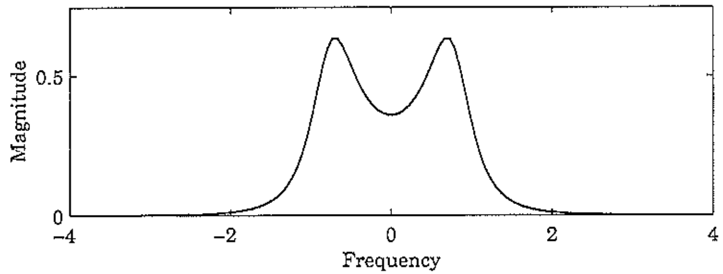
has the covariance function

$$r(\tau) = \begin{cases} \sigma^2 & \tau = 0 \\ 0 & \tau = \pm 1, \pm 2, \ldots \end{cases}$$

and the spectral density

$$\phi(\omega) = \frac{\sigma^2}{2\pi}$$

Today's Challenge Example



How do we filter white noise to get an output with the plotted spectral density:

$$F(z) = \frac{1}{2\pi} \cdot \frac{0.3125 + 0.25\cos\omega}{2.25 - 3\cos\omega + \cos(2\omega)}$$

Are there more than one solutions?

Example – First order filter

The difference equation

$$x(k+1) = ax(k) + v(k) \qquad m_v = 0, \quad \text{cov}\,[v] = r_1$$

with initial condition $\text{E}x(k_0) = m_0$, $\text{cov}\,[x(k_0)] = r_0$ gives:

Mean value

$$m(k+1) = am(k), \qquad m(k_0) = m_0$$
$$m(k) = a^{k-k_0} m_0$$

Covariance function

$$P(k+1) = a^2 P(k) + r_1, \qquad P(k_0) = r_0$$
$$P(k) = a^{2(k-k_0)} r_0 + \frac{1 - a^{2(k-k_0)}}{1 - a^2}\, r_1$$

Linear stochastic difference equations

$$x(k+1) = \Phi x(k) + v(k)$$

where the white noise $v(k)$ is independent of $x(k')$ for $k' \le k$.

Need to specify:

Initial distribution: m_0 and R_0
Noise covariance: R_1

Properties of $x(k+1) = \Phi x(k) + v(k)$

Mean value

$$m(k+1) = \text{E}x(k+1) = \Phi m(k) \qquad m(0) = m_0$$

Covariance function

$$P(k) := \text{cov}\,[x(k), x(k)] = \text{E}\tilde{x}(k)\tilde{x}^T(k)$$

where $\tilde{x} = x - m$ and

$$\begin{aligned}\tilde{x}(k+1)\tilde{x}^T(k+1) &= \{\Phi\tilde{x}(k) + v(k)\}\{\Phi\tilde{x}(k) + v(k)\}^T \\ &= \Phi\tilde{x}(k)\tilde{x}^T(k)\Phi^T + \Phi\tilde{x}(k)v^T(k) + v(k)\tilde{x}^T(k)\Phi^T + v(k)v^T(k)\end{aligned}$$

Taking expectation gives

$$P(k+1) = \Phi P(k)\Phi^T + R_1 \qquad P(0) = R_0$$

Input-output models

White noise through linear filters

$$y(k) = \sum_{l=-\infty}^{k} h(k-l)u(l) = \sum_{n=0}^{\infty} h(n)u(k-n)$$

Taking mean values

$$m_y(k) = \mathrm{E}y(k) = \mathrm{E}\sum_{n=0}^{\infty} h(n)u(k-n)$$
$$= \sum_{n=0}^{\infty} h(n)\mathrm{E}u(k-n) = \sum_{n=0}^{\infty} h(n)m_u(k-n)$$

The mean value behaves as the signal
Use zero mean value in the following

Covariance function

Assume stationarity

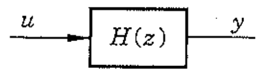
$$r_y(\tau) = \mathrm{E}y(k+\tau)y^T(k) = \mathrm{E}\sum_{n=0}^{\infty} h(n)u(k+\tau-n)\left(\sum_{l=0}^{\infty} h(l)u(k-l)\right)^T$$
$$= \sum_{n=0}^{\infty}\sum_{l=0}^{\infty} h(n)\left(\mathrm{E}u(k+\tau-n)u^T(k-l)\right)h^T(l)$$
$$= \sum_{n=0}^{\infty}\sum_{l=0}^{\infty} h(n)\, r_u(\tau+l-n)\, h^T(l)$$

Some care must be taken with respect to exchange of summations and expectations

Simplification using spectral density

$$\phi_y(\omega) = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} e^{-in\omega} r_y(n)$$
$$= \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} e^{-in\omega}\sum_{k=0}^{\infty}\sum_{l=0}^{\infty} h(k)r_u(n+l-k)h^T(l)$$
$$= \frac{1}{2\pi}\sum_{k=0}^{\infty}\sum_{n=-\infty}^{\infty}\sum_{l=0}^{\infty} e^{-ik\omega}h(k)e^{-i(n+l-k)\omega}\cdot r_u(n+l-k)e^{il\omega}h^T(l)$$
$$= \frac{1}{2\pi}\sum_{k=0}^{\infty} e^{-ik\omega}h(k)\sum_{n'=-\infty}^{\infty} e^{-in'\omega}r_u(n')\sum_{l=0}^{\infty} e^{il\omega}h^T(l)$$
$$= H(e^{i\omega})\phi_u(\omega)H^T(e^{-i\omega})$$

Main result



Input signal $u(k)$ stationary stochastic process, m_u, ϕ_u

If H is stable then $y(k)$ stationary process with mean value

$$m_y = H(1)m_u$$

spectral density

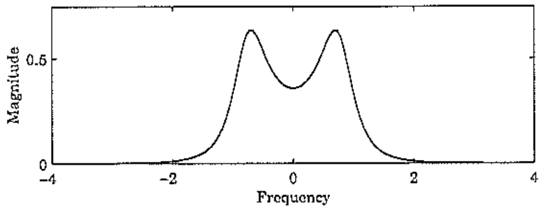
$$\phi_y(\omega) = H(e^{i\omega})\phi_u(\omega)H^T(e^{-i\omega})$$

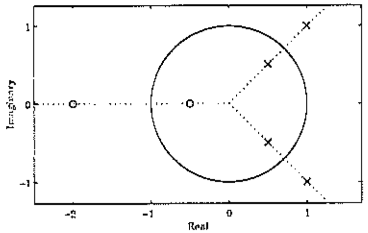
cross-spectral density

$$\phi_{yu}(\omega) = H(e^{i\omega})\phi_u(\omega)$$

"Everything" can be generated by filtering white noise

Solution of Challenge Example





The notation $z = e^{i\omega}$ gives $\cos\omega = (e^{i\omega} + e^{-i\omega})/2 = (z + z^{-1})/2$ and

$$\frac{1}{2\pi}\cdot\frac{0.3125 + 0.25\cos\omega}{2.25 - 3\cos\omega + \cos(2\omega)} = \frac{1}{2\pi}\cdot\frac{0.3125 + 0.125(z + z^{-1})}{2.25 - 1.5(z + z^{-1}) + 0.5(z^2 + z^{-2})}$$
$$= \frac{1}{2\pi}\cdot\underbrace{\frac{(0.5z + 0.25)}{(z^2 - z + 0.5)}}_{H(z)}\cdot\underbrace{\frac{(0.5z^{-1} + 0.25)}{(z^{-2} - z^{-1} + 0.5)}}_{H(z^{-1})}$$

White noise through $H(z)$ gives the desired spectral density

Stationary equivalence

Are there other filters that give the same spectral denisty?

Consider the three processes

$$x(k) = \frac{B(q)}{A(q)}v(k) + e(k) \qquad y(k) = \frac{C(q)}{A(q)}w(k) \qquad z(k) = \frac{D(q)}{A(q)}\varepsilon(k)$$

where v, w, e and ε are uncorrelated white noise processes with zero mean and variance one, then all three processes have the same spectral denisty if and only if

$$\left[\frac{B(e^{i\omega})}{A(e^{i\omega})}\frac{B(e^{-i\omega})}{A(e^{-i\omega})} + 1\right] = \frac{C(e^{i\omega})}{A(e^{i\omega})}\frac{C(e^{-i\omega})}{A(e^{-i\omega})} = \frac{D(e^{i\omega})}{A(e^{i\omega})}\frac{D(e^{-i\omega})}{A(e^{-i\omega})}$$

Spectral factorization

Problem: How generate a stochastic process with given spectral density from white noise u?

$$\phi_y(\omega) = H(e^{i\omega})\phi_u(\omega)H^T(e^{-i\omega}) = F(z)|_{z=e^{i\omega}}$$

Introduce $z = e^{i\omega}$ then the spectral density can be written as

$$F(z) = \frac{1}{2\pi}H(z)H^T(z^{-1})$$

Note: If z_i zero or pole of F then the same is true for z_i^{-1}

$$\text{abs}(1/z) = 1/\text{abs}(z) \qquad \arg(1/z) = -\arg(z)$$

Symmetry with respect to real axis as usual

Spectral factorization cont'd

Poles and zeros of $F(z)$ in pairs such that

$$z_i z_j = 1 \qquad p_i p_j = 1$$

1. Start with the desired spectral density $F(z)$
2. Determine poles and zeros of $F(z)$
3. Choose the poles and zeros that are less than 1 in magnitude, z_i and p_i
4. Form the filter

$$H(z) = K\frac{\prod(z - z_i)}{\prod(z - p_i)} = \frac{B(z)}{A(z)}$$

Stationarity implies that $|p_i| < 1$, but zeros may have unit magnitude. K is chosen to get the right stationary gain.

Innovation representations

$$y(k) = \sum_{n=-\infty}^{k} h(k-n)e(n) = \frac{B(z)}{A(z)}e(k)$$

Assume stable inverse

$$e(k) = \sum_{n=-\infty}^{k} g(k-n)y(n) = \frac{A(z)}{B(z)}y(k)$$

Eliminate old noise parts

$$y(k+1) = \sum_{n=-\infty}^{k+1} h(k+1-n)e(n) = \sum_{n=-\infty}^{k} h(k+1-n)e(n) + h(0)e(k+1)$$
$$= \underbrace{\sum_{n=-\infty}^{k} h(k+1-n) \sum_{l=-\infty}^{n} g(n-l)y(l)}_{\text{Prediction of } y(k+1)} + h(0)\underbrace{e(k+1)}_{\text{Innovation}}$$

Calculation of variances

What is the variance of

$$y(k) = \frac{B(q)}{A(q)}\, e(k)$$

where e is white noise with unit variance?

$$\begin{aligned} \mathrm{E}y^2 &= \int_{-\pi}^{\pi} \phi(\omega)d\omega \\ &= \frac{1}{i}\int_{-\pi}^{\pi} \phi(\omega)e^{-i\omega}d(e^{i\omega}) \\ &= \frac{1}{2\pi i}\oint \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})}\cdot\frac{dz}{z} \end{aligned}$$

Continuous-time white noise

Problems!

$$\phi(\omega) = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-i\omega t} r(t)\, dt$$
$$r(t) = \int_{-\infty}^{\infty} e^{i\omega t}\phi(\omega)\, d\omega$$

White noise (formally)

$$\phi(\omega) = \frac{r_0}{2\pi}$$
$$r(t) = r_0\delta(t)$$

Thus

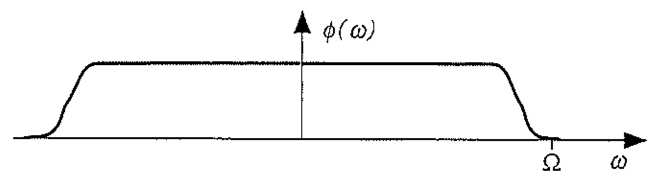
$$r(0) = \int_{-\infty}^{\infty} \phi(\omega)\, d\omega = \infty$$

Infinite variance!

The infinite variance problem

How to circumvent the problem of infinite white noise variance?

- Band-limited white noise ?



- Use Wiener processes $w(t)$ (Model for random walk)

$$dw = w(t+dt) - w(t) \qquad \mathrm{E}(dw)^2 = r_0\, dt \qquad w(t) = \int_0^t e(s)\, ds$$

where e is white noise.

Summary

- Minimize output variance due to stochastic disturbances
- Stochastic processes
- White noise
- Transfer function reshapes spectral density
- Find filters by spectral factorization