Modeling

Outline

1. Sampling
   - Sampling of signals
   - Sampling of systems
   - Choice of sampling period

2. Identification
   - Least-squares method
   - Recursive least-squares method
   - Closed-loop identification

Outline

1. Sampling
   - Sampling of signals
   - Sampling of systems
   - Choice of sampling period
2. Identification
   - Least-squares method
   - Recursive least-squares method
   - Closed-loop identification

Aliasing

Continuous-time signal

$$y(t) = \sin(2\pi ft)$$

sampled at $t = kh$, $k = 1, 2, \ldots$

Discrete-time signal

$$y(kh) = \sin(2\pi fkh)$$

h sampling period

$f_s = \frac{1}{h}$ sampling frequency

Aliasing

Continuous-time signal

$$y(t) = \sin(2\pi f t)$$

sampled at $t = kh$, $k = 1, 2, \ldots$

Discrete-time signal

$$y(kh) = \sin(2\pi f kh)$$

h sampling period

$f_s = \frac{1}{h}$ sampling frequency

Aliasing frequency

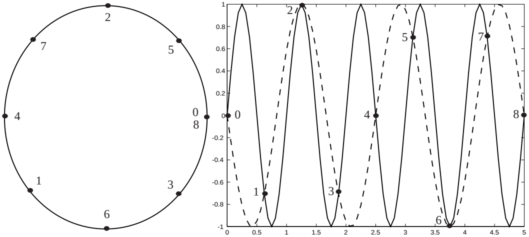
$$f_a = f + n f_s, \quad n \in \{\pm 1, \pm 2, \ldots\}$$

Consider another signal $z(t) = \sin(2\pi f_a t)$

$$z(kh) = \sin(2\pi f_a hk) = \sin(2\pi fhk + 2\pi nk) = \sin(2\pi fhk) = y(kh)$$

Ambiguity: Cannot distinguish y from z at $t = kh$

Example



Actual signal

$$y(t) = \sin(2\pi f t), \quad f = 1$$

Sampling period $h = \frac{5}{8}$
Sampling frequency $f_s = \frac{1}{h} = \frac{8}{5}$
Aliasing frequency
$f_a = f - f_s = -\frac{3}{5}$

Misinterpreted signal
$z(t) = \sin(2\pi f_a t)$

$$z(kh) = y(kh) \quad \text{ambiguity}$$

Anti-aliasing filter

Nyquist frequency

$$f_N = \frac{f_s}{2}, \quad \omega_N = \frac{\omega_s}{2}$$

If $f < f_N$ then aliasing frequencies are above Nyquist frequency

$$|f_a| = |f \pm nf_s| > f_N, \quad n = 1, 2, \ldots$$

Anti-aliasing filter

Nyquist frequency

$$f_N = \frac{f_s}{2}, \quad \omega_N = \frac{\omega_s}{2}$$

If $f < f_N$ then aliasing frequencies are above Nyquist frequency

$$|f_a| = |f \pm nf_s| > f_N, \quad n = 1, 2, \ldots$$

Anti-aliasing filter (analog)

Low pass filter that eliminates frequencies above Nyquist frequency
$\rightarrow$No ambiguity in interpretation of sampled signal

Anti-aliasing filter in control systems

- Closed-loop system $y = Gr$, $G = \frac{BT}{A_c}$
- Normalized frequency $\hat{\omega} = \omega h$
- Bandwidth ω_B: closed-loop gain drops to $|G(e^{-i\hat{\omega}_B})| = 0.7$
- Anti-aliasing filter should cut off frequencies above bandwidth

Anti-aliasing filter in control systems

- Closed-loop system $y = Gr$, $G = \frac{BT}{A_c}$
- Normalized frequency $\hat{\omega} = \omega h$
- Bandwidth ω_B: closed-loop gain drops to $|G(e^{-i\hat{\omega}_B})| = 0.7$
- Anti-aliasing filter should cut off frequencies above bandwidth

Anti-aliasing filter purpose

Cut off noise frequencies before sampling such that noise is not misinterpreted as disturbance-to-be-rejected (below bandwidth) by the controller

Outline

1. Sampling
   - Sampling of signals
   - Sampling of systems
   - Choice of sampling period
2. Identification
   - Least-squares method
   - Recursive least-squares method
   - Closed-loop identification

A first order system

$$\frac{dy(t)}{dt} = py(t) + u(t), \quad G(s) = \frac{1}{s-p}$$

Multiply with integrating factor

$$e^{-pt}(\frac{dy(t)}{dt} - py(t)) = \frac{d}{dt}(e^{-pt}y(t)) = e^{-pt}u(t)$$

Integrate from 0 to t

$$e^{-pt}y(t) - y(0) = \int_0^t e^{-p\tau}u(\tau)d\tau = \int_0^t e^{-p(t-\tau)}u(t-\tau)d\tau$$
$$y(t) - e^{pt}y(0) = \int_0^t e^{p\tau}u(t-\tau)d\tau$$

Sampling of a first order system

$$y(t) - e^{pt}y(0) = \int_0^t e^{p\tau}u(t-\tau)d\tau$$

Let $t = h$ and use zero-order-hold $u(h-t) = u(0), 0 < t < h$

$$y(h) - e^{ph}y(0) = \begin{cases} \frac{1}{p}(e^{ph}-1)u(0) & p \neq 0 \\ hu(0) & p = 0 \end{cases}$$

This corresponds to $k = 0$ while $y(kh)$, for any k, evolves as

$$y(kh+h) = \lambda y(kh) + cu(kh), \begin{cases} \lambda = e^{ph} \\ c = \begin{cases} (\lambda - 1)/p & p \neq 0 \\ h & p = 0 \end{cases} \end{cases}$$

$$G(s) = \frac{1}{s-p} \rightarrow H(\mathrm{q}^{-1}) = \frac{c\mathrm{q}^{-1}}{1-\lambda \mathrm{q}^{-1}}$$

Sampling of poles

Continuous pole ZOH sampling $\longrightarrow$ discrete pole

$$p \longrightarrow \lambda = e^{ph}$$

Stable regions: left-half-plane $\Re s < 0 \longrightarrow |z| < 1$ unit circle

Sampling of poles

Continuous pole ZOH sampling $\longrightarrow$ discrete pole

$$p \longrightarrow \lambda = e^{ph}$$

Stable regions: left-half-plane $\Re s < 0 \longrightarrow |z| < 1$ unit circle

Compare discretizations approximations

- $s = \frac{z-1}{h}$, $\Re s < 0 \longrightarrow \Re z < 1$ (not inside unit circle)
- $s = \frac{1-z^{-1}}{h}$, $\Re s < 0 \longrightarrow |z - 0.5| < 0.5$ (inside unit circle)
- $s = \frac{2}{h}\frac{z-1}{z+1}$. $\Re s < 0 \longrightarrow |z| < 1$ (unit circle)

Sampling of poles

Continuous pole ZOH sampling $\longrightarrow$ discrete pole

$$p \longrightarrow \lambda = e^{ph}$$

Stable regions: left-half-plane $\Re s < 0 \longrightarrow |z| < 1$ unit circle

Compare discretizations approximations

- $s = \frac{z-1}{h}$, $\Re s < 0 \longrightarrow \Re z < 1$ (not inside unit circle)
- $s = \frac{1-z^{-1}}{h}$, $\Re s < 0 \longrightarrow |z - 0.5| < 0.5$ (inside unit circle)
- $s = \frac{2}{h}\frac{z-1}{z+1}$. $\Re s < 0 \longrightarrow |z| < 1$ (unit circle)

Inversely, these approximate the pole transformation

- $z = e^{sh} \approx 1 + sh$ Euler's (Forward) difference
- $z = e^{sh} \approx \frac{1}{1-sh}$ Backward difference
- $z = e^{sh} \approx \frac{1+sh/2}{1-sh/2}$ Tustin (Trapezoidal method)

Sampling of higher order systems

Sampling of first order system

$$G_i(s) = \frac{1}{s - p_i} \rightarrow H_i(q^{-1}) = \frac{c_i q^{-1}}{1 - \lambda_i q^{-1}} \left\{ \begin{array}{l} \lambda_i = e^{p_i h} \\ c_i = \left\{ \begin{array}{ll} (\lambda_i - 1)/p_i & p_i \neq 0 \\ h & p_i = 0 \end{array} \right. \end{array} \right.$$

Sampling of n-th order system with distinct poles $(\lambda_i \neq \lambda_j)$

$$G(s) = \sum_{i=1}^{n} d_i G_i(s) \rightarrow H(q^{-1}) = \sum_{i=1}^{n} d_i H_i(q^{-1})$$

Example: second order system with distinct poles

Continuous system

$$G(s) = \frac{d_1}{s - p_1} + \frac{d_2}{s - p_2}, \quad d_i = \lim_{s \to p_i} (s - p_i) G(s)$$

ZOH sampling

$$H(\mathrm{q}^{-1}) = \frac{c_1 d_1 \mathrm{q}^{-1}}{1 - \lambda_1 \mathrm{q}^{-1}} + \frac{c_2 d_2 \mathrm{q}^{-1}}{1 - \lambda_2 \mathrm{q}^{-1}}$$

In polynomial form

$$H(\mathrm{q}^{-1}) = \frac{b_1 \mathrm{q}^{-1} + b_2 \mathrm{q}^{-1}}{1 + a_1 \mathrm{q}^{-1} + a_2 \mathrm{q}^{-2}} \qquad \begin{cases} a_1 = -(\lambda_1 + \lambda_2) \\ a_2 = \lambda_1 \lambda_2 \\ b_1 = c_1 d_1 + c_2 d_2 \\ b_2 = -(c_1 d_1 \lambda_2 + c_2 d_2 \lambda_1) \end{cases}$$

Coinciding poles

Problem:

If $p_1 = p_2 \Rightarrow d_1, d_2 \to \infty$ unbounded

Trick:

First assume $p_2 - p_1 = \epsilon \neq 0$ such that d_1, d_2 bounded
Find $\sum H_i = H(\mathrm{q}^{-1}) = \frac{B}{A}$ on rational form (a_i, b_i bounded)
Let $\epsilon \to 0$ then d_i unbounded but a_i, b_i remain bounded

Example

$$G(s) = \frac{1}{s^2} \to H(\mathrm{q}^{-1}) = \frac{h^2}{2} \frac{\mathrm{q}^{-1} + \mathrm{q}^{-2}}{(1 - \mathrm{q}^{-1})^2}$$

Other sampling techniques

General and advanced sampling techniques based on

- Laplace and Z-transforms
- Residues
- State-space description and matrix functions

Sampling function in Sysquake (similar in Matlab)

$> help\ c2dm$
$C2DM$
$Continuous - to - discrete - time\ conversion.$

$SYNTAX$
$(numd, dend) = c2dm(numc, denc, Ts)$
$dend = c2dm(numc, denc, Ts)$
$(numd, dend) = c2dm(numc, denc, Ts, method)$
$dend = c2dm(numc, denc, Ts, method)$
$(Ad, Bd, Cd, Dd) = c2dm(Ac, Bc, Cc, Dc, Ts, method)$

$DESCRIPTION$

$\ldots$

Outline

1. Sampling
   - Sampling of signals
   - Sampling of systems
   - Choice of sampling period
2. Identification
   - Least-squares method
   - Recursive least-squares method
   - Closed-loop identification

Rules of thumb

All sampling rules of thumb relate to closed-loop bandwidth ω_B

- $\hat{\omega}_B = \omega_B h \in [0.5, 1]$ $(\omega_s = 2\pi/h)$
- $\omega_s \approx 20\omega_B$
- $\omega_s \approx 40\omega_B$ for discretized analog designs (PID)

Rules of thumb

All sampling rules of thumb relate to closed-loop bandwidth ω_B

- $\hat{\omega}_B = \omega_B h \in [0.5, 1]$ ($\omega_s = 2\pi/h$)
- $\omega_s \approx 20\omega_B$
- $\omega_s \approx 40\omega_B$ for discretized analog designs (PID)

Rule of thumb for selection of sampling frequency

$$\omega_s = \sigma\omega_B, \quad \sigma \in [6, 40]$$

Problems with too fast sampling

Pole clustering at 1 causes numerical problems

Discrete poles cluster $\lambda = e^{ph} \approx 1$ for small h

Perturbation of characteristic polynomial by $\epsilon = 10^{-8}$

$$(\lambda - 0.99)^4 + \epsilon = 0 \rightarrow \lambda = 0.99 + (-\epsilon)^{1/4}$$

Poles spread to circle with radius $r = |\epsilon|^{1/4} = 0.01$, unstable pole!

Problems with too fast sampling

Pole clustering at 1 causes numerical problems

Discrete poles cluster $\lambda = e^{ph} \approx 1$ for small h
Perturbation of characteristic polynomial by $\epsilon = 10^{-8}$

$$(\lambda - 0.99)^4 + \epsilon = 0 \rightarrow \lambda = 0.99 + (-\epsilon)^{1/4}$$

Poles spread to circle with radius $r = |\epsilon|^{1/4} = 0.01$, unstable pole!

Round-off errors due to finite precision

Integral action in PI controller “turned off” for small h

$$i(kh + h) = i(kh) + e(kh) * h/T_i$$

In comparison to $i(kh)$ the term $e(kh)h/T_i$ might fall outside the resolution and is rounded to zero

Discrete design versus discretized analog design

Servo model

$$G(s) = \frac{4}{s(s+2)}$$

Study 3 cases:

- Design 1: Sampling period $h_1 = 0.025s$ and $\lambda_{1k} = 0.9, 0.93, 0.95,\ k = 1, 2, 3.$
- Design 2: Sampling $h_2 = 20h_1 = 0.5s$ and $\lambda_{2k} = e^{p_k h_2}$ where $p_k = \frac{1}{h_1} \ln \lambda_{1k}$
- Design 3: Analog design with pole placement $p_k,\ k = 1, 2, 3$ and discretization using forward-difference approximation of the continuous-time controller

All pole placements correspond to the same continuous poles p_k

Design 1: $h_1 = 0.025s$

ZOH sampling:

$$G_1(\mathrm{q}^{-1}) = \frac{B_1}{A_1} = \frac{1.23 \cdot 10^{-3}\mathrm{q}^{-1}(1 + 0.98\mathrm{q}^{-1})}{(1 - \mathrm{q}^{-1})(1 - 0.95\mathrm{q}^{-1})}$$

Polynomial equation $A_1 R_1 + B_1 S_1 = A_{c1}$ and $T_1 = \frac{A_{c1}}{B_1}(1)$

$$\rightarrow \begin{cases} R_1 = 1 - 0.832\mathrm{q}^{-1} \\ S_1 = 2.931 - 2.788\mathrm{q}^{-1} \\ T_1 = 0.1435 \end{cases}$$

Design 2: $h_2 = 20h_1 = 0.5s$

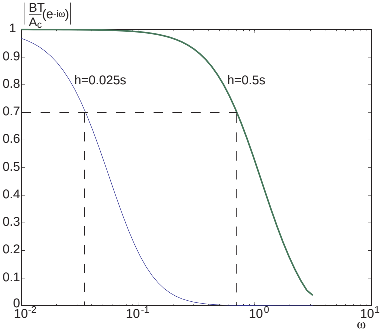
ZOH sampling:

$$G_2(q^{-1}) = \frac{B_2}{A_2} = \frac{0.368q^{-1}(1 + 0.72q^{-1})}{(1 - q^{-1})(1 - 0.37q^{-1})}$$

Polynomial equation $A_2R_2 + B_2S_2 = A_{c2}$ and $T_2 = \frac{A_{c2}}{B_2}(1)$

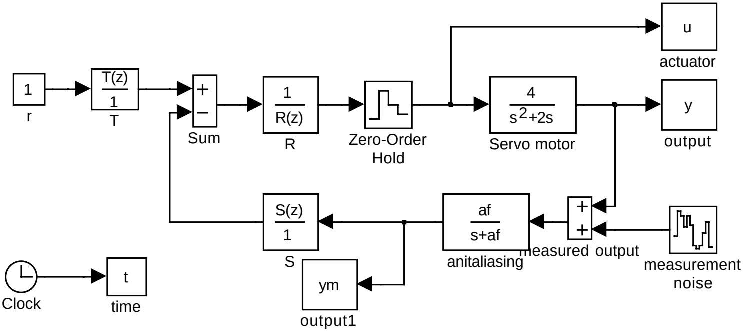
$$\rightarrow \begin{cases} R_2 = 1 + 0.257q^{-1} \\ S_2 = 1.079 - 0.396q^{-1} \\ T_2 = 0.683 \end{cases}$$

Checking rule of thumb

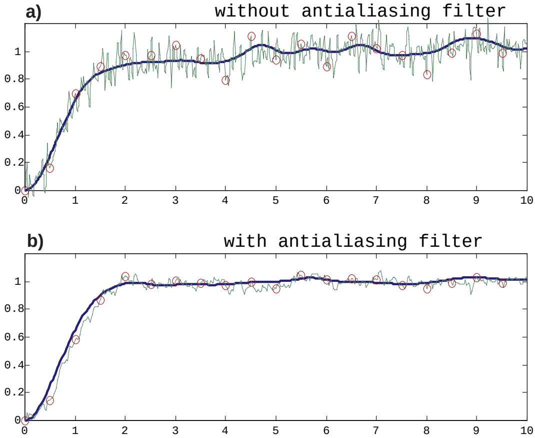


Rule of thumb: $\hat{\omega}_B \in [0.15, 1]$
Design 1: ($h_1 = 0.025s$) unnecessary fast sampling
Design 2: ($h_2 = 0.5$) appropriate sampling

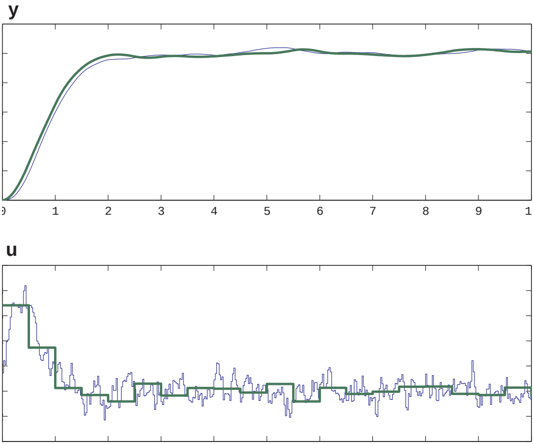
Simulink implementation



Design 2 without and with anti-aliasing filter



Compare design 1 and 2



Design 3: $h_3 = 0.25s$

Continuous system

$$G(s) = \frac{B}{A} = \frac{4}{s(s+2)}$$

Polynomial equation $AR + BS = A_{c3}$

$$\rightarrow \begin{cases} R = s + 7.169 \\ S = 3.1246s + 6.275 \end{cases}$$

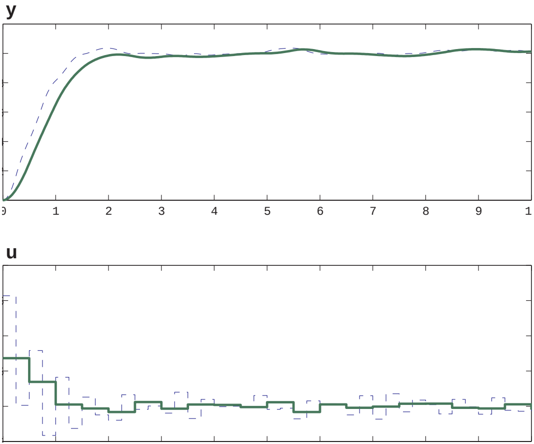
Discrete approximation and $T_3 = \frac{A_{c3}}{B_3}(1)$

$$s \rightarrow \frac{q-1}{h_3} \rightarrow \begin{cases} R_3 = 1 + 0.792\mathrm{q}^{-1} \\ S_3 = 3.125 - 1.556\mathrm{q}^{-1} \\ T_3 = 1.569 \end{cases}$$

Checking actual pole placement on correctly sampled system

$$\lambda_{3k} = 0.63 \pm 0.19i, -0.78 \quad \text{oscillatory pole}$$

Design 3



Outline

The identification problem

Difference equation in polynomial form with *equation error* ε_θ

$$A(\theta)y(k) = B(\theta)u(k) + \varepsilon_\theta(k)$$
$$\begin{cases} A(\theta) = 1 + a_1 \mathrm{q}^{-1} + \ldots + a_{\deg A} q^{-\deg A} \\ B(\theta) = b_\tau q^{-\tau} + \ldots + b_{\deg B} q^{-\deg B} \end{cases}$$

Linear regression form $y(k) = \varphi(k)^T \theta + \varepsilon_\theta(k)$

$$\theta = \begin{pmatrix} a_1 & \ldots & a_{\deg A} & b_\tau & \ldots & b_{\deg B} \end{pmatrix}^T$$
$$\varphi(k) = \begin{pmatrix} -y(k-1) & -y(k-2) & \ldots & u(k-\tau) & \ldots \end{pmatrix}^T$$

Find estimate $\hat{\theta}$ such that ε_θ "small"

The least-square problem

Collect many data during excitation $N >> \dim(\theta)$

$$\begin{aligned} y(1) &= \varphi(1)^T\theta + \varepsilon_\theta(1) \\ y(2) &= \varphi(2)^T\theta + \varepsilon_\theta(2) \\ &\vdots \\ y(N) &= \varphi(N)^T\theta + \varepsilon_\theta(N) \end{aligned}$$

In matrix form

$$Y = \Phi\theta + \varepsilon_\theta, \quad \begin{cases} Y = \begin{pmatrix} y(1) & \dots & y(N) \end{pmatrix}^T \\ \Phi = \begin{pmatrix} \varphi(1) & \dots & \varphi(N) \end{pmatrix}^T \\ \varepsilon_\theta = \begin{pmatrix} \varepsilon_\theta(1) & \dots & \varepsilon_\theta(N) \end{pmatrix}^T \end{cases}$$

Criterion is sum of squares of equation errors

$$V(\theta) = \sum_{k=1}^{N} \varepsilon_\theta(k)^2 = \varepsilon_\theta^T \varepsilon_\theta \to \hat{\theta} = \arg\min V(\theta) = (\Phi^T\Phi)^{-1}\Phi^T Y$$

Example

Assume true system is

$$y(k) - 0.9y(k-1) = 0.1u(k-1) + e(k), \quad \theta = \begin{pmatrix} -0.9 \\ 0.1 \end{pmatrix}$$

where u and e independent white noise ($E[u]) = E[e] = 0$, $E[u^2] = E[e^2] = 1$)

Construct equation system

$$\Phi = \begin{pmatrix} -y(1) & u(1) \\ \vdots & \vdots \\ -y(999) & u(999) \end{pmatrix}, \quad Y = \begin{pmatrix} y(2) \\ \vdots \\ y(1000) \end{pmatrix}$$

Least-squares estimate

$$\hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y = \begin{pmatrix} -0.897 \\ 0.090 \end{pmatrix}$$

Example in Sysquake

$$> u = randn(1, 1000);$$
$$> e = 0.1 * randn(1, 1000);$$
$$> A = [1, -0.9];$$
$$> B = [0, 0.1];$$
$$> y = filter(B, A, u) + filter(1, A, e);$$
$$> Y = y(2 : end)';$$
$$> Phi = [-y(1 : end - 1)', u(1 : end - 1)'];$$
$$> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)$$
$$theta\_hat =$$
$$-0.9099$$
$$0.1009$$

Unbiased least-squares estimate

Suppose the true system is

$$\begin{aligned} y(k) &= \varphi(k)^T\theta_0 + e(k), \quad k = 1, \dots N \\ Y &= \Phi\theta_0 + e \end{aligned}$$

Least-squares estimate

$$\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T(\Phi\theta_0 + e) = \theta_0 + (\Phi^T\Phi)^{-1}\Phi^T e$$

If e zero mean and uncorrelated with Φ

$$\hat{\theta} = \theta_0 \quad (N \to \infty)$$

Nonzero mean data destroys identification result!

$$\left.\begin{array}{l} Mean[y] \neq 0 \\ Mean[\varepsilon_\theta] = Mean[Ay - Bu] = 0 \end{array}\right\} \rightarrow A(1) = 0$$

$> e = e + 1;$ *%Nonzero mean disturbance*

$> y = filter(B, A, u) + filter(1, A, e);$

$> mean(y)$

$ans =$

0.9766

$> Phi = [-y(1 : end - 1)', u(1 : end - 1)'];$

$> Y = y(2 : end)';$

$> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)$

$theta\_hat =$

-0.9918

0.1018

Eliminate mean from data before estimation

$> yy = y - mean(y); uu = u - mean(u);$ %*Eliminate mean*
$> Y = yy(2:end)';$
$> Phi = [-yy(1:end-1)', uu(1:end-1)'];$
$> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)$
$theta\_hat =$
-0.9086
0.1009

Colored equation error

Natural to assume measurement error e is “white”

$$y = \frac{B}{A}u + e \rightarrow Ay = Bu + Ae$$

Colored equation error $\varepsilon = Ae$ correlated to Φ

$$
\begin{aligned}
&> e = randn(1, 1000);\\
&> y = filter(B, A, u) + e;\\
&> Y = y(2 : end)';\\
&> Phi = [-y(1 : end - 1)', u(1 : end - 1)'];\\
&> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)\\
&theta\_hat =\\
&\quad -5.6094e - 2\\
&\quad\;\; 8.3903e - 2
\end{aligned}
$$

Improve signal-to-noise ratio

$> u = 10 * randn(1, 1000);$ $\%improve\ signal - to - noise\ ratio$

$> y = filter(B, A, u) + e;$

$> Y = y(2 : end)';$

$> Phi = [-y(1 : end - 1)', u(1 : end - 1)'];$

$> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)$

$theta\_hat =$

-0.7604

$9.7216e - 2$

Data filter

System

$$Ay = Bu + Ae$$

Filter data by estimate of A;

$$y_f = \frac{1}{D} y$$
$$u_f = \frac{1}{D} u$$

Filtered data satisfy

$$Ay_f = Bu_f + \frac{A}{D}e, \quad \varepsilon = \frac{A}{D}e \approx e \text{ ``whiter''}$$

Data filter: example

$$> D = [1 - 0.72]; \%datafilter$$
$$> yf = filter(1, D, y);$$
$$> uf = filter(1, D, u);$$
$$> Y = yf(2 : end)';$$
$$> Phi = [-yf(1 : end - 1)', uf(1 : end - 1)'];$$
$$> theta\_hat = (Phi' * Phi)\backslash(Phi' * Y)$$
$$theta\_hat =$$
$$-0.8879$$
$$0.1025$$

Outline

Recursive formulation

$$\hat{\theta}(k) = [\sum_{t=1}^{k} \varphi(t)\varphi(t)^T]^{-1} \sum_{t=1}^{k} \varphi(t)y(t)$$

Introduce the notation

$$P(k) = [\sum_{t=1}^{k} \varphi(t)\varphi(t)^T]^{-1}$$

Then since

$$P^{-1}(k) = P^{-1}(k-1) + \varphi(k)\varphi(k)^T$$

it follows that

$$\begin{aligned}\hat{\theta}(k) &= P(k)[\textstyle\sum_{t=1}^{k-1} \varphi(t)y(t) + \varphi(k)y(k)] \\ &= P(k)[P^{-1}(k-1)\hat{\theta}(k-1) + \varphi(k)y(k)] \\ &= \hat{\theta}(k-1) + P(k)\varphi(k)[y(k) - \varphi(k)^T\hat{\theta}(k-1)]\end{aligned}$$

Simplifications

Avoid matrix inversion (matrix inversion lemma gives)

$$P(k) = P(k-1) - \frac{P(k-1)\varphi(k)\varphi(k)^T P(k-1)}{1 + \varphi(k)^T P(k-1)\varphi(k)}$$

Since

$$P(k)\varphi(k) = P(k-1)\varphi(k)/[1 + \varphi(k)^T P(k-1)\varphi(k)]$$

Introduce auxiliary variables

$$\begin{cases} n(k) = P(k-1)\varphi(k) \\ d(k) = 1 + \varphi(k)^T n(k) \\ K(k) = n(k)/d(k) \\ \varepsilon(k) = y(k) - \varphi(k)^T \hat{\theta}(k-1) \end{cases}$$

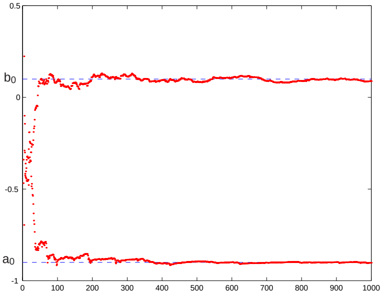
Recursive equations

$$\begin{cases} \hat{\theta}(k) = \hat{\theta}(k-1) + K(k)\varepsilon(k) \\ P(k) = P(k-1) - n(k)n(k)^T/d(k) \end{cases}$$

Example: recursive least-squares algorithm

Previous example with initial conditions

$$\hat{\theta}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad P(0) = 10^4 \cdot I_{2\times 2}$$



Outline

Closed-loop identification

The open-loop equation error

$$\varepsilon = Ay - Bu$$

Closed-loop response

$$y = \underbrace{\frac{BT}{A_c}}_{y_m} r + \frac{R}{A_c}\varepsilon$$

Unmodeled response

$$e_u = y - y_m = \frac{R}{A_c}\varepsilon = \frac{R}{A_c}(Ay - Bu) = A\underbrace{(\frac{R}{A_c}y)}_{y_F} - B\underbrace{(\frac{R}{A_c}u)}_{u_F}$$

Appropriate data filter $\frac{R}{A_c}$

Example

Unstable servo model sampled with period $h = 0.5$

$$G(s) = \frac{4}{s(s+2)} \rightarrow H(\mathrm{q}^{-1}) = \frac{0.3679\mathrm{q}^{-1} + 0.2642\mathrm{q}^{-2}}{1 - 1.3679\mathrm{q}^{-1} + 0.3679\mathrm{q}^{-2}}$$

Controller

$$\begin{cases} R = 1 + 0.2567\mathrm{q}^{-1} \\ S = 1.0787 - 0.3961\mathrm{q}^{-1} \\ T = 0.68266 \end{cases}$$

Effect of data filter $F = \frac{R}{A_c}$

	θ_0	$\hat{\theta}$	$\hat{\theta}_F$
a_1	-1.3679	-1.2099	-1.3552
a_2	0.3679	0.2178	0.3543
b_1	0.3679	0.3735	0.3744
b_2	0.2642	0.2881	0.2707