Today's Outline

LECTURE:
- Prediction-Error (Parametric) Estimation Techniques
- Classical Model Validation Techniques
- "Hairdryer" data example

LAB:
- Illustration of System ID Toolbox Graphical Interface
- Simulated Fifth-order System Laboratory #2
- Phenol Plant Data Lab #2
• In the prediction problem, current and previous measurements from the plant are used to obtain estimates k+1 (or beyond) time steps in the future.

**Prediction-Error Model Structures**

Some general info

- Use of regression techniques to get a model estimate
- Regression may be linear or nonlinear, depending on the model structure
- 32 general formulations for prediction-error models, of which only five are commonly used

**System Identification Problem**

- Input Signal (Random or Deterministic)
- Disturbance Signal (random, autocorrelated)
- True Plant: \( y(t) = p(z)u(t) + H(z)\nu(t) \)
- Plant Model: \( y(t) = \hat{p}(z)u(t) + \hat{\nu}_e(z)\epsilon(t) \)

\[
\begin{align*}
A(z) & = 1 + a_1z^{-1} + \ldots + a_nz^{-n} \\
B(z) & = b_1 + b_2z^{-1} + \ldots + b_nz^{-n+1} \\
C(z) & = 1 + c_1z^{-1} + \ldots + c_nz^{-n} \\
D(z) & = 1 + d_1z^{-1} + \ldots + d_nz^{-n+1} \\
F(z) & = 1 + f_1z^{-1} + \ldots + f_nz^{-n+1}
\end{align*}
\]

In transfer function form:

\[
\begin{align*}
y(t) & = \hat{p}(z)u(t) + \hat{\nu}_e(z)\epsilon(t) \\
\hat{p}(z) & = \frac{B(z)}{A(z)F(z)}z^{-nk} \\
\hat{\nu}_e(z) & = \frac{C(z)}{A(z)D(z)}
\end{align*}
\]
Popular PEM Structures

<table>
<thead>
<tr>
<th>Method</th>
<th>( \hat{p}(z) )</th>
<th>( \hat{p}_r(z) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARX</td>
<td>( \frac{B(z)}{A(z)} )</td>
<td>( -\frac{B(z)}{A(z)} )</td>
</tr>
<tr>
<td>ARMAX</td>
<td>( \frac{C(z)}{A(z)} )</td>
<td>( \frac{C(z)}{A(z)} )</td>
</tr>
<tr>
<td>FIR</td>
<td>( \frac{B(z)}{A(z)} )</td>
<td>( 1 )</td>
</tr>
<tr>
<td>Box-Jenkins</td>
<td>( \frac{B(z)}{A(z)} )</td>
<td>( \frac{C(z)}{A(z)} )</td>
</tr>
<tr>
<td>Output Error</td>
<td>( \frac{B(z)}{A(z)} )</td>
<td>( 1 )</td>
</tr>
</tbody>
</table>

\[
A(z)y(t) = \frac{B(z)}{F(z)}u(t - nk) + \frac{C(z)}{D(z)}e(t)
\]
\[
y(t) = \hat{p}(z)u(t) + \hat{p}_r(z)e(t)
\]

The One-Step Ahead Prediction Error

- \( e(t) \) is the one-step ahead prediction error
- \( e(t) = y(t) - \hat{y}(t| t-1) = \hat{p}_r^{-1}(z)(y(t) - \hat{p}(z)u(t)) \)
- \( \hat{y}(t| t-1) \) is the one-step ahead prediction of \( y \):
  \[
  \hat{y}(t| t-1) = \hat{p}_r^{-1}(z)\hat{p}(z)u(t) + (1 - \hat{p}_r^{-1}(z))y(t)
  \]

The notation for \( \hat{y}(t| t-1) \) indicates that prediction at time \( t \) must be based on plant data from the time interval \([0, t-1]\). The variable \( e(t) \) (sometimes referred to as the equation error) represents part of the output \( y(t) \) that cannot be predicted from past data. In the absence of a noise model (\( \hat{p}_r = 1 \)), the prediction error reduces into the residual or output error, computed as

\[
e_{\text{resid}}(t) = y(t) - \hat{p}(z)u(t) = \hat{e}(t)
\]

PEM Estimation is a Regression Problem

The one-step ahead prediction error for a general PEM model

\[
C(z)F(z)\hat{y}(t|\theta) = D(z)B(z)u(t - nk) + F(z)[C(z) - D(z)A(z)]y(t)
\]

can be written in “pseudolinear” regression form as follows:

\[
\hat{y}(t|\theta) = \varphi^T(t|\theta)\theta
\]

\[
\varphi(t|\theta) = \left[ -y(t-1), \ldots, -y(t-n_0), u(t-nk), \ldots, u(t-nk+n_0), 1, -w(t-1), \ldots, -w(t-n_0/\theta), \nu(t-1), \ldots, \nu(t-n_0/\theta) \right]^T
\]

\[
\theta = \begin{bmatrix} a_1 & \ldots & a_{n_0} & b_{n_1} & \ldots & b_{n_0} & f_1 & \ldots & f_{n_0} & c_1 & \ldots & c_{n_0} & d_1 & \ldots & d_{n_0} \end{bmatrix}^T
\]

\( w, \nu, \) and \( \nu(t|\theta) \) are auxiliary variables that depend on both the model parameters and the data

\[
w(t|\theta) = \frac{B(z)}{F(z)}u(t)
\]
\[
\nu(t|\theta) = \frac{A(z)y(t) - w(t|\theta)}{\frac{D(z)}{C(z)}e(t|\theta)}
\]

Least-Squares Solution

\[
\arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} e_i^2(\theta) = \arg \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \left[ y - \varphi^T(i|\theta)\theta \right]^2
\]

If using FIR or ARX estimation: The solution is the standard linear-least squares formula

\[
\hat{\theta} = \left[ \frac{1}{N} \sum_{i=1}^{N} \varphi(i|\theta)^T \right]^{-1} \frac{1}{N} \sum_{i=1}^{N} \varphi(i|\theta)y(t)
\]

If using ARMAX, Output Error, or Box-Jenkins estimation: An iterative, numerical solution relying on explicit search techniques (e.g., Newton-Raphson, Levenberg-Marquardt, Gauss-Newton) is required. The basis of all these methods is the need to find a rule for iterating on the parameter vector \( \theta \):

\[
\hat{\theta}^{(i+1)} = \hat{\theta}^{(i)} + \alpha f^{(i)}
\]

\( f^{(i)} \) is the search direction determined on the basis of the objective function values for previous iterations, its gradients (first derivatives), and Hessians (second derivatives). \( \alpha \) is a positive constant determined so an appropriate decrease in the value of the objective function is obtained.

© Copyright 1997, 1998 by D.E. Rivera, All Rights Reserved
AutoRegressive with eXternal input structure (ARX)

\[ A(z)y(t) = B(z)u(t - nk) + e(t) \]

\[ A(z) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n} \]
\[ B(z) = b_1 + b_2 z^{-1} + \ldots + b_n z^{-n+1} \]

- Estimation problem involves a linear regression problem.

- High-order ARX estimation (na and nb large) yields consistent estimates but may result in variance problems in the presence of significant noise.

- Low-order ARX estimation is problematic 1) in the presence of significant noise and 2) when an incorrect model structure is chosen.

AutoRegressive Moving Average with eXternal input structure (ARMAX)

\[ A(z)y(t) = B(z)u(t - nk) + C(z)e(t) \]
\[ A(z) = 1 + a_1 z^{-1} + \ldots + a_n z^{-n} \]
\[ B(z) = b_1 + b_2 z^{-1} + \ldots + b_n z^{-n+1} \]
\[ C(z) = 1 + c_1 z^{-1} + \ldots + c_n z^{-n} \]

- Estimation problem is a nonlinear regression problem

- Model orders (na, nb, nc) usually chosen to be low.

- Presence of autoregressive polynomial can yield bias problems in the presence of significant noise and/or model structure mismatch; moving average polynomial will sometimes counteract negative effects, however.

ARX Parameter Estimation

The one-step ahead predictor for \( y \)

\[ \hat{y}(t+1) = -a_1 y(t) - \ldots - a_n y(t-n_a) + b_1 u(t-n_k) + \ldots + b_n u(t-n_k-n_b+1) \]

can be expressed as a linear regression problem via

\[ \varphi = [ -y(t-1) \ldots -y(t-n_a) \ u(t-n_k) \ldots u(t-nk-n_b+1) ]^T \]

and \( \theta \), the vector of parameter estimates:

\[ \theta = [ a_1 \ldots a_n \ b_1 \ldots b_n ]^T \]

Rewriting the objective (“loss”) function as

\[ \min_\theta V = \min_\theta \frac{1}{N} \sum_{i=1}^{N} [y(t) - \varphi^T(t)\theta]^2 \]

leads to the well-established linear least-squares solution

\[ \hat{\theta} = \left[ \frac{1}{N} \sum_{i=1}^{N} \varphi(t)\varphi^T(t) \right]^{-1} \frac{1}{N} \sum_{i=1}^{N} \varphi(t)y(t) \]

Finite Impulse Response (FIR)

\[ y(t) = B(z)u(t - nk) + e(t) \]
\[ B(z) = b_1 + b_2 z^{-1} + \ldots + b_n z^{-n_b+1} \]

- "Structure-free" model representation equivalent to what we saw in correlation analysis

- Estimation problem is linear regression problem

- Because of fast sampling, model order (nb) is usually high (20 coefficients or more)

- No autocorrelated noise model is estimated.
ChE 494/598, EEE 598  Intro to System Identification:  Parametric Model Estimation and Validation, March 26, 1998

**Box-Jenkins (B-J) Model Structure**

\[
y(t) = \frac{B(z)}{F(z)} u(t - nk) + \frac{C(z)}{D(z)} e(t)
\]

\[
B(z) = b_1 + b_2 z^{-1} + \ldots + b_n z^{-n_k+1}
\]

\[
C(z) = 1 + c_1 z^{-1} + \ldots + c_n z^{-n_c}
\]

\[
D(z) = 1 + d_1 z^{-1} + \ldots + d_n z^{-n_d}
\]

\[
F(z) = 1 + f_1 z^{-1} + \ldots + f_n z^{-n_f}
\]

- Estimation problem is a nonlinear regression problem
- Model orders (nb, nc, nd, nf) usually chosen to be low.
- Independently parametrizes transfer function and noise models; lots of decisions and possibly many iterations to be made by the user, however.

**Output Error (OE) Model Structure**

\[
y(t) = \frac{B(z)}{F(z)} u(t - nk) + e(t)
\]

\[
B(z) = b_1 + b_2 z^{-1} + \ldots + b_n z^{-n_k+1}
\]

\[
F(z) = 1 + f_1 z^{-1} + \ldots + f_n z^{-n_f}
\]

- Estimation problem a nonlinear regression problem
- Model orders (nb, nf) usually chosen to be low.
- Independently parametrizes the input and noise, although an autocorrelated noise model is not obtained.
- Works great in conjunction with control-relevant prefiltering

**Selecting a "Suitable" Model Structure**

Represent the ZOH first-order delay model using the five PEM structures (assume a noise-free situation):

ARX: \( n_a = 1, \ nb = 1, \ nk = N + 1 \)

ARMAX: \( n_a = 1, \ nb = 1, \ nc = \text{Not Applicable}, \ nk = N + 1 \)

FIR: \( nb > 3\tau/T, \ nk = N + 1, \) (model must capture at least 95% settling time)

Box-Jenkins: \( nb = 1, \ nf = 1, \ nc = NA, \ nd = NA, nk = N+1 \)

Output Error: \( nb = 1, \ nf = 1, \ nk = N+1 \)

**Principal Sources of Error in System Identification**

\[ \text{ERROR} = \text{BIAS} + \text{VARIANCE} \]

- **BIAS.** Systematic errors caused by
  - input signal characteristics (i.e., excitation)
  - choice of model structure
  - mode of operation (i.e., closed-loop instead of open-loop)

- **VARIANCE.** Random errors introduced by the presence of noise in the data, which do not allow the model to exactly reproduce the plant output. It is affected by the following factors:
  - number of model parameters
  - duration of the identification test
  - signal-to-noise ratio

© Copyright 1997, 1998 by D.E. Rivera, All Rights Reserved
Sources of Error (More Precisely)

The Mean-Square Error (MSE) is a sum of the bias and variance contributions to the model error:

\[
E[p_n(e^{i\omega}) - p(e^{i\omega})]^2 = \frac{\text{Bias}^2}{E[p_n(e^{i\omega})] - E[p_n(e^{i\omega})]}^2 + \frac{\text{Variance}}{E[p_n(e^{i\omega})] - E[p_n(e^{i\omega})]}^2
\]

\[E \equiv \text{Expectation operator}
\]
\[p \equiv \text{true plant}
\]
\[\hat{p}_N \equiv \text{model estimate (based on N data points)}
\]
\[E\hat{p}_N \equiv \text{Expectation (mean) of model estimate}
\]

Sources of Bias

\[
\Phi_{e_p} = \frac{|L|^2}{|p|^2} \left( |p - \hat{p}|^2 \Phi_p + 2Re((p - \hat{p})^H (e^{i\omega}) \Phi_u) + |H(e^{i\omega})|^2 \sigma_u^2 \right)
\]

Input signal power \(\Phi_p\). The input signal must have sufficient power over the frequency range of importance to the control problem.

Choice of prefilter \(L(z)\). The prefilter acts as a frequency-dependent weight on the estimation problem that can be used to influence the goodness of fit in selected portions of the model’s response.

Structure of \(\hat{p}\). Expanding the model set (e.g. by increasing model order) decreases bias.

Structure of \(\hat{p}_0\). Acts as a weight similar to prefiltering. Autoregressive terms (\(A(z)\) or \(D(z)\)) emphasize the fit at high frequencies.

Noise spectrum \(\Phi_u = |H(e^{i\omega})|^2 \sigma_u^2\). If noise dynamics differ substantially from plant dynamics, a trade-off between fitting to \(\hat{p}\) and fitting to \(\hat{p}_0\) will result whenever \(A(z) \neq 1\).

Crossspectrum \(\Phi_u\). Correlated inputs (as a result of closed-loop operation or operator intervention) may result in bias.

Open-Loop Prediction Error Estimation

True Plant: \(y(t) = p(z)u(t) + H(z)a(t)\)

Plant Model: \(\hat{y}(t) = \hat{p}(z)u(t) + \hat{p}_0(z)\epsilon(t)\)

Consider prefiltered input/output data

\[y_F(t) = L(z)y(t) \quad u_F(t) = L(z)u(t)\]

Asymptotically (as the number of observations \(N \rightarrow \infty\)), the least-squares estimation problem can be written as:

\[
\min_{\hat{p}_N, \hat{p}_0} \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^{N} \epsilon_F(t)^2 = \min_{\hat{p}_N, \hat{p}_0} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{\epsilon_F}(\omega) d\omega
\]

where \(\Phi_{\epsilon_F}\), the prefiltered prediction-error spectrum is

\[
\Phi_{\epsilon_F} = \frac{|L|^2}{|p|^2} (|p - \hat{p}|^2 \Phi_p + 2Re((p - \hat{p})^H (e^{i\omega}) \Phi_u) + |H(e^{i\omega})|^2 \sigma_u^2)
\]

Why low-order ARX gives trouble in cases of undermodelling or significant noise

\[
\Phi_{\epsilon_F} = \left| p(e^{i\omega}) - B(e^{i\omega})^2 \Phi_u(\omega) + \Phi_u(\omega) \right| A(e^{i\omega})^2 |L(e^{i\omega})|^2 d\omega.
\]

The autoregressive term places a high-frequency emphasis on the goodness-of-fit, which becomes exacerbated when an incorrect model structure or significant high-frequency noise is present.
Requirements for Consistent (Bias-Free) Estimation

Consistent least-squares estimation, i.e.,
\[
\min_{\hat{\theta}, \hat{\theta}_c} \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} e_i^2(t) = \min_{\hat{\theta}, \hat{\theta}_c} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{eF}(\omega) d\omega = \sigma_e^2
\]
As \( N \to \infty \), \( \hat{\theta}(z) \to \theta(z) \) and \( \hat{\theta}_c(z) \to H(z) \) is obtained when the following are true:

1. The model structure for \( \hat{\theta} \) and \( \hat{\theta}_c \) describes the true plant.
2. \( u \) shows persistent excitation. The input signal \( u \) must have nonzero power (\( \Phi_u(\omega) \neq 0 \)) over “enough” frequencies.

Note: The theory does not require \( u \) and \( a \) to be independent, uncorrelated sequences (i.e., \( \Phi_{ua}(\omega) = 0 \) for all \( \omega \)); however, achieving independence between the inputs is a desirable practical requirement.

Persistence of Excitation

Definition (from Ljung, 1987). A quasi-stationary signal \( u(t) \) with spectrum \( \Phi_u(\omega) \) is said to be persistently exciting of order \( n \) if, for all filters of the form
\[
M_n(z) = m_1z^{-1} + \ldots + m_nz^{-n}
\]
the relation
\[
|\Phi_u(e^{i\omega})|^2 \Phi_u(\omega) = 0
\]
implies that \( M_n(e^{i\omega}) \equiv 0 \)

Interpretation: The input \( u(t) \) is persistently exciting of order \( n \) if \( \Phi_u(\omega) \) is different from zero in at least \( n \) points on the interval \(-\pi < \omega \leq \pi \).

Comment 1: If the numerator and denominator of a model have the same degree \( n \), then the input should be persistently exciting of order \( 2n + 1 \).

Comment 2: A single sinusoid \( u(t) = \sin(\omega_0 k) \) has degree two of persistent excitation; hence a sum of \( n + 1 \) sinusoids will identify an \( n \)-th order system.

How Input Signal Design Influences Bias

Consider prefiltered input/output data
\[
y_F(t) = L(z)y(t) \quad u_F(t) = L(z)u(t)
\]
Asymptotically (as the number of observations \( N \to \infty \)), the least-squares estimation problem can be written as:
\[
\min_{\hat{\theta}, \hat{\theta}_c} \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} e_i^2(t) = \min_{\hat{\theta}, \hat{\theta}_c} \frac{1}{2\pi} \int_{-\pi}^{\pi} \Phi_{eF}(\omega) d\omega
\]
where \( \Phi_{eF} \), the prefiltered prediction-error spectrum is
\[
\Phi_{eF}(\omega) = |L(z)|^2 \phi_{\hat{\theta}_c}^2(\omega) + 2\text{Re}(\phi_{\hat{\theta}_c}\phi_{\hat{\theta}_c}^\ast \phi_{\hat{\theta}}\phi_{\hat{\theta}_c}^\ast) + |H(e^{i\omega})|^2 \sigma_a^2
\]
Input signal power \( \Phi_u \). The input signal must have sufficient power over the frequency range of importance to the control problem.

Crosspectrum \( \Phi_{ua} \). Correlated inputs (as a result of closed-loop operation or operator intervention) may result in bias.
Benefits of Open-Loop Operation
(assuming that input and disturbance are uncrosscorrelated, independent sequences)
• Requirements on the structure of the model can be relaxed
• Consistent estimation of $p(z)$ can still be obtained with a "wrong" structure for the noise model
• One is guaranteed that a good fit to the data implies a good model
• A "perfect" fit to closed-loop or correlated data may result from an erroneous model
• Data quality (and corresponding quality of the model estimates) is less of an issue than in closed-loop operation.

Sources of Variance
For large $n$ (model order) and large $N$ (number of data), the asymptotic covariance for the unbiased model estimate is:

$$
\text{Cov}\left[ \hat{p}(e^{j\omega})\right] \sim \frac{n}{N}\Phi_u(\omega)\begin{bmatrix}
\Phi_u(\omega) & \Phi_v(\omega) \\
\Phi_v(\omega) & \sigma_a^2
\end{bmatrix}^{-1}
$$

$$
\Phi_u \equiv \text{Input Power Spectrum}
\Phi_v \equiv \text{Disturbance Power Spectrum} = |H(e^{j\omega})|^2\sigma_a^2
\Phi_v \equiv \text{Covariance between u(t) and a(t)}
$$

In the case of independent open-loop operation ($\Phi_v = 0$), then

$$
\text{Cov}\hat{p}(e^{j\omega}) \sim \frac{n}{N}\Phi_u(\omega)
\text{Cov}\hat{p}_a(e^{j\omega}) \sim \frac{n}{N}\sigma_a^2 = \frac{n}{N}|H(e^{j\omega})|^2
$$

Minimizing Variance Effects

$$
\text{Cov}\hat{p}(e^{j\omega}) \sim \frac{n}{N}\Phi_u(\omega)
\text{Cov}\hat{p}_a(e^{j\omega}) \sim \frac{n}{N}\Phi_v(\omega) = \frac{n}{N}|H(e^{j\omega})|^2
$$

ARMA Models (For the Record)

$$
A(z)y(t) = C(z)e(t)
$$

$$
A(z) = 1 + a_1z^{-1} + \ldots + a_nz^{-n} \\
C(z) = 1 + c_1z^{-1} + \ldots + c_nz^{-n}
$$

• ARMA (in the Astrom/Ljung context) is used to describe models for univariate time series analysis.
• AR model estimation is a linear regression problem but full ARMA modeling involves nonlinear estimation.
• High-order AR modeling is used in correlation analysis to obtain the prewhitening filter.
### Multi-Input PEM Structure

\[ A(z)y(t) = \frac{B_1(z)}{F_1(z)}u_1(t-nk_1) + \ldots + \frac{B_m(z)}{F_m(z)}u_m(t-nk_m) + \frac{C(z)}{D(z)}e(t) \]

- Supported by the \texttt{pem} command in MATLAB.
- Autoregressive polynomial affects all transfer functions.
- Identification data can be collected either by manipulating all inputs simultaneously or one-at-a-time.
- Practical benefits from using independent (uncorrelated) inputs in all channels.

### Interesting PEM modifications

- Consider minimizing the sum of squares of the j-step-ahead prediction errors as the identification objective (Ljung, Shook, Mohtadi and Shah)

\[ \epsilon(t+j) = y(t+j) - \hat{y}(t+j|t-1) \]

- Use a Laguerre expansion in lieu of the FIR model (Walhberg)

\[ \tilde{p}(z) = \sum_{k=1}^{\infty} \bar{g}_k K z^{-k} \left( \frac{1 - az}{z - a} \right)^{k-1} \]

\[ K = \frac{\sqrt{1 - a^2}}{T} \]

- Partial Least Squares-based system identification (Wise and Ricker).

### Assessing Model Quality

(as noted by Ljung and Glad, 1994)

- Model quality is related to the intended purpose of the model (e.g., simulation, control).
- Typically, model quality is related to the ability of the model to reproduce the behavior of the system.
- Model quality is also related to the model's stability, i.e., how well the model can be reproduced from different measured data sets.

### Classical Model Validation Techniques

- **Simulation** (plot the measured output time series versus the predicted output from the model).
- **Crossvalidation** (Simulate on a data set different than the one used for parameter estimation; for a number of different model structures, plot the loss function and select the minimum).
- **Impulse, step, and frequency responses** (compare with results from nonparametric techniques - correlation and spectral analysis).
- **Correlation analysis** on the prediction errors (make sure they resemble white noise).
- **Pole/Zero Plots**. Can be used to simplify the model structure.
Information Criteria

In the absence of crossvalidation data, these criteria can be used to balance between model fit and the number of parameters used.

Akaike’s information criterion (AIC):

$$\min_{d, \theta} \left( 1 + \frac{2d}{N} \sum_{i=1}^{N} e^2(t, \theta) \right)$$

Final Prediction Error (FPE):

$$\min_{d, \theta} \left( 1 + \frac{d}{N} \sum_{i=1}^{N} e^2(t, \theta) \right)$$

Rissanen’s minimal description length (MDL):

$$\min_{d, \theta} \left( 1 + \frac{2d}{N} \log N \sum_{i=1}^{N} e^2(t, \theta) \right)$$

$N \equiv$ Data length
$\theta \equiv$ Vector of parameter estimates
$d \equiv \text{dim } \theta$ (no. of estimated parameters)
$e(t, \theta) \equiv$ One-step-ahead prediction error for a given $\theta$

Feedback’s Process Trainer PT326

“Hairdryer” process; input (MV) is the voltage over the heating device; output (CV) is outlet temperature

Prediction-Error Correlation Analysis

$$x(t) = \hat{p}_i(z) ((p(z) - \hat{p}(z))u(t) + H(z)a(t))$$

Autocorrelation in $e$:

$$\rho_e(k) = \frac{\gamma_e(k)}{\sigma^2_e}$$

Cross-correlation between $e$ and $u$:

$$\rho_{eu}(k) = \frac{\gamma_{eu}(k)}{\sqrt{\sigma^2_e \sigma^2_u}}$$

• autocorrelation in the residuals means that the noise model structure is incorrect.
• crosscorrelation between the residuals and the input signifies undermodelling, the input/output model structure is incorrect.

ARX Estimation - MATLAB Commands

% ARX estimation
% th = arx([y u],[na nb nk])
% th = arx([2 1 1]);
% th = sett(th,0.08); % Set the correct sampling interval.

pause, present(th) % Press any key to see model.

This matrix was created by the command ARX on 6/23 1994 at 21:22
Loss fcn: 0.03657  Akaike’s FPE: 0.03706 Sampling interval 0.08
The polynomial coefficients and their standard deviations are

$$B =$$

| 0 | 0.0453 |
| 0 | 0.0074 |

$$A =$$

| 1.0000 | -0.9598 |
| 0 | 0.0133 |
Prediction Error Analysis - ARX [1 1 1]

% Check with correlation analysis on the prediction error
resid(z2,th); pause  % Press any key to continue

Both noise and plant models appear inadequate based on this validation
criteria!

Prediction-Error Analysis - ARX [2 2 3]

% Correlation analysis seems to tell us that this model is o.k.
% Another way to find out is to simulate it
% and compare the model output with measured output. We then
% select a portion of the original data that was not used to build
% the model, viz from sample 800 to 900:
u = dtrend(u2(800:900)); y = dtrend(y2(800:900));
yh = idsim(u,th);

% Now compute the transfer function of the model:
gth = th2ff(th);% We may compare this transfer function with what
% is obtained from spectral analysis
pause, bodeplot([gs gth]), pause

% We can compute the step response of the
% estimated model as follows.
step = ones(20,1);
msrpr = idsim(step,th);

% This step response can be compared with that from
% correlation
pause, plot(slen,msrpr,'-',0.08*[0:19],mstepr,'--'), pause
% Press any key for plot.

Simulation on Crossvalidation Data-ARX [2 2 3]

% Correlation analysis seems to tell us that this model is o.k.
% Another way to find out is to simulate it
% and compare the model output with measured output. We then
% select a portion of the original data that was not used to build
% the model, viz from sample 800 to 900:
u = dtrend(u2(800:900)); y = dtrend(y2(800:900));
yh = idsim(u,th);

% Now compute the transfer function of the model:
gth = th2ff(th);% We may compare this transfer function with what
% is obtained from spectral analysis
pause, bodeplot([gs gth]), pause

% We can compute the step response of the
% estimated model as follows.
step = ones(20,1);
msrpr = idsim(step,th);

% This step response can be compared with that from
% correlation
pause, plot(slen,msrpr,'-',0.08*[0:19],mstepr,'--'), pause
% Press any key for plot.

Step and Frequency Response Comparison
% MODEL STRUCTURE SELECTION FOR ARX
% Cross-Validation method
% V = arxstruc(z2,[y u],NN);
% nn = selstruc(V);
% NN contains the different ARX model structures
NN = [1 1 1; 1 2 1; 1 1 2; 1 2 2; 2 2 1; 2 2 2; 2 2 3; 2 3 1; 2 3 2; 2 3 3; 3 1 4; 3 2 1; 3 2 4; 3 3 1; 3 3 2; 3 3 4; 4 1 5; 4 2 1; 4 2 6; 4 2 7; 4 2 8; 4 2 9; 5 5 1; 6 6 1; 7 7 1; 8 8 1; 9 9 1; 10 10 1];
% 4 Parameters to be estimated
% is the Best Answer
V = arxstruc(z2,[y u],NN);
nn = selstruc(V)

arx([y u],[na nb nc nk],maxiter,tol,lim,maxsize,T)

MAXITER: The maximum number of iterations to be performed when searching for the minimum. Default is MAXITER=10. With MAXITER=0 only a non-iterative initial value estimation procedure is carried out.

TOL: The iterations are continued until the candidate update vector has a norm less than TOL. Default is TOL=0.01. The iterations are also terminated when MAXITER is reached, or when the search procedure fails to find a lower value of the criterion along the candidate direction.

LIM: The criterion is robustified, so that a residual that is larger than LIM*(estimated standard deviation) carries a linear, rather than quadratic weight. Default is LIM=1.6. LIM=0 means that a non-robustified (truly quadratic) criterion is used.

MAXSIZE: No matrix with more than MAXSIZE elements is formed by the algorithms. The default value is set by the .m-file idmsize. (On a PC the default value is MAXSIZE = 4096). If you run into memory problems try lower values of MAXSIZE. See also HELP MEMORY.

T: The sampling interval. Default is T=1. T is essential to obtain physical frequency scales, and when transforming to continuous time.

Omitting trailing arguments or entering them as [] gives default
Intermediate Results from Nonlinear PEM Estimation

ITERATION # 3
Current loss: 0.001283  Previous loss: 0.001283
Current th prev. th  gn-dir

theta =

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>-1.3002, -1.3006, 0.0003</td>
<td></td>
</tr>
<tr>
<td>a2</td>
<td>0.4177, 0.4180, -0.0003</td>
<td></td>
</tr>
<tr>
<td>b1</td>
<td>0.0670, 0.0669, 0.0000</td>
<td></td>
</tr>
<tr>
<td>b2</td>
<td>0.0418, 0.0417, 0.0001</td>
<td></td>
</tr>
<tr>
<td>c1</td>
<td>-0.2901, -0.2915, 0.0014</td>
<td></td>
</tr>
</tbody>
</table>

Norm of gn-vector: 0.001482 <- Norm of gradient vector at zero means that local optimum has been obtained

This information appears on OE, BJ, ARMAX, and PEM commands.

Prediction-Error Analysis

ARMAX [2 2 3]

% Try ARMAX model instead
% th1 = armax(z2,[na nb nc nk]);
th1 = armax(z2,[2 2 1 3]);
% Set the correct sampling interval.

Box-Jenkins [2 1 2 2 3]

% Go for Output Error
% th2 = oe(z2,[nb nf nk]);
th2 = oe(z2,[2 2 3]);
% Set the correct sampling interval.

%Box-Jenkins (BJ);
% th3 = bj(z2,[nb nc nd nf nk]);
th3 = bj(z2,[2 1 2 2 3]);
% Set the correct sampling interval.

All autocorrelation in the residuals has been removed thanks to the C(z) term in the ARMAX model.

Autocorrelation in the residuals is unavoidable because of the “lack” of a noise model.
Finite Impulse Response (FIR)

%FIR
% th4 = arx(z2,[0 nb nk]);
th4 = arx(z2,[0 17 3]);  % Set the correct sampling interval.
pause, present(th4) % Press any key to see model.

A Structured Approach to Parameter Estimation
(per L. Ljung, “From Data To Model: An Overview of System Identification”)

- Start with an ARX [4 4 1] model
- If the model is not adequate, try the following:
  - increase the model order (e.g., ARX [8 8 1])
  - increase the number of inputs to the problem (i.e., try multi-input PEM estimation)
- If still not adequate, resort to semiphysical modeling
- If still unsuccessful, give up! (his words, not mine...)

Lab Session #3

- Prediction-error methods and classical validation on the simulated fifth-order system; comparison with Lab 2a results
- Shell phenol plant data file. Prediction-error methods and validation; comparison with Lab 2b results.