System Identification

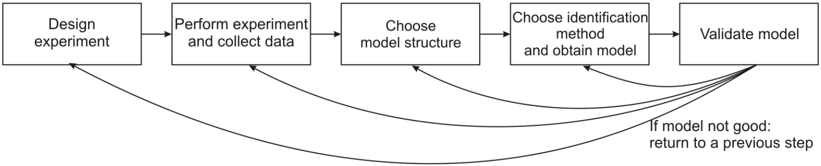
Control Engineering EN, 3rd year B.Sc.
Technical University of Cluj-Napoca
Romania

Lecturer: Lucian Buşoniu

Part IX

Model validation and structure selection

Recall: Importance of validation



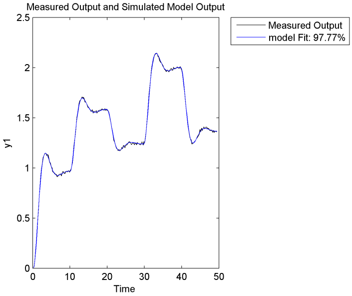
Model validation is a crucial step: the model must be good enough (for its intended usage).

If validation is unsuccessful, previous steps in the workflow must be redone, for instance:

- Rerun the identification algorithm with different parameters (e.g. δ in recursive methods).
- Change the model structure: e.g. orders of polynomials *na*, *nb* in ARX, or even the model type entirely, e.g. IV instead of ARX
- Design and run a new experiment (e.g. more data, different input signal)

Motivation

So far, we validated and selected models mostly informally, by examining plots or comparing errors – using *common sense*.



Next, some mathematically well-founded tests will be given.

However, common sense remains indispensable – mathematical tests work under assumptions that may not always be satisfied.

Focus: Prediction error methods

This lecture focuses on *single-output* models obtained by *prediction error methods*.

Some of the tests can be extended to other settings.

Table of contents

Whiteness: Intuition

Recall general PEM model structure:

$$y(k) = G(q^{-1})u(k) + H(q^{-1})e(k)$$

where $e(k)$ is *assumed* to be zero-mean white noise.

PEM are derived so that the prediction error $\varepsilon(k) = y(k) - \widehat{y}(k) = e(k)$. If the system satisfies the model structure (so the assumption holds), and moreover if the model is accurate, then $\varepsilon(k)$ is also zero-mean white noise.

Whiteness hypothesis

(W) The prediction errors $\varepsilon(k)$ are zero-mean white noise.

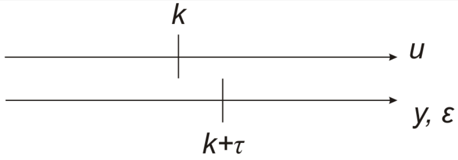
Independence of past inputs: Intuition

$$y(k) = G(q^{-1})u(k) + v(k)$$

If the model G is accurate, it entirely explains the influence of inputs $u(k)$ on current and future outputs $y(k+\tau)$. Therefore, the errors $\varepsilon(k+\tau) = y(k+\tau) - \widehat{y}(k+\tau)$ are only influenced by the disturbances v, and are *independent* of inputs $u(k)$.

Independence hypothesis 1

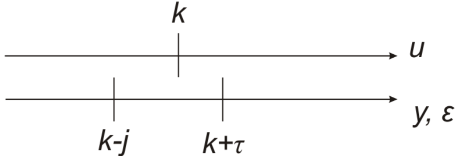
(I1) The prediction errors $\varepsilon(k+\tau)$ are independent of inputs $u(k)$ for $\tau \geq 0$ (i.e., current and future errors are independent of current inputs).



Independence of all inputs: Intuition

$$y(k) = G(q^{-1})u(k) + v(k)$$

If the experiment is closed-loop, $u(k)$ depends on past outputs and this will lead to a correlation of past errors $\varepsilon(k+\tau), \tau < 0$ with $u(k)$ (note the independence for $\tau \geq 0$ is not affected). If open-loop, then $\varepsilon(k+\tau), \tau < 0$ is also independent from $u(k)$.



Independence hypothesis 2

(I2) The prediction errors $\varepsilon(k+\tau)$ are independent of $u(k)$ for *any* τ (i.e., all the errors are independent of all the inputs).

All hypotheses

- (W) The prediction errors $\varepsilon(k)$ are zero-mean white noise.
- (I1) The prediction errors $\varepsilon(k+\tau)$ are independent of $u(k)$ for $\tau \geq 0$ (current and future errors are independent of current inputs).
- (I2) The prediction errors $\varepsilon(k+\tau)$ are independent of $u(k)$ for *any* τ (all the errors are independent of all the inputs).

A good model should satisfy (W) and (I1), and if there is no feedback, also (I2).

We will develop tests that allow to either accept or reject these hypotheses for a given model, and therefore validate or reject the model.

Whiteness: Correlations

Recall the correlation function (equal to the covariance in zero-mean case):

$$r_\varepsilon(\tau) = \mathrm{E}\left\{\varepsilon(k+\tau)\varepsilon(k)\right\}$$

If $\varepsilon(k)$ is zero-mean white noise:

- The correlation function is zero, $r_\varepsilon(\tau) = 0$ for any nonzero τ.
- At zero, $r_\varepsilon(0)$ is the variance σ^2 of the white noise.

Whiteness: Correlations from data

Estimating correlations from data:

$$\widehat{r}_\varepsilon(\tau) = \frac{1}{N} \sum_{k=1}^{N-\tau} \varepsilon(k+\tau)\varepsilon(k)$$

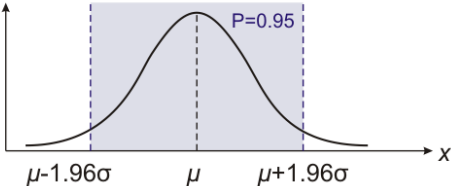
Normalization by the (estimated) variance:

$$x(\tau) = \frac{\widehat{r}_\varepsilon(\tau)}{\widehat{r}_\varepsilon(0)}$$

Then, $x(\tau)$ should be small for nonzero τ (it will usually not be exactly zero for finite data).

Whiteness test

For any Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$, the probability of drawing a sample in the interval $\mu - 1.96\sigma, \mu + 1.96\sigma$ is found (from the Gaussian formula) to be 0.95.



It can be shown that for large N, $x(\tau)$ is distributed according to Gaussian $\mathcal{N}(0, \frac{1}{N})$, and therefore that:

$$P(|x(\tau)| \leq \frac{1.96}{\sqrt{N}}) = 0.95$$

Whiteness test (continued)

Whiteness test

If $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}$ for all $\tau \neq 0$ supported by the data, then the whiteness hypothesis (W) is accepted. Otherwise, (W) is rejected.

Intuition: If the test succeeds, then the hypothesis is likely to be true.

More detailed intuition

Define the following logical propositions:

P1: The model is correct.

P2: The prediction errors are white noise.

P3: The correlation $x(\tau)$ is Gaussian with $\mathcal{N}(0, \frac{1}{N})$.

P4: $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}, \quad \forall \tau \neq 0.$

We have $P1 \Rightarrow P2 \Rightarrow P3$, so when $P3$ is false, $\neg P3 \Rightarrow \neg P2 \Rightarrow \neg P1$ and the model is rejected.

The implication from $P3$ to $P4$ is probabilistic so things become more complicated, but informally speaking we decide that $\neg P3$ when $\neg P4$, i.e., when $|x(\tau)| > \frac{1.96}{\sqrt{N}}$ for at least one $\tau \neq 0$. If $P3$ is true, there is still a small, but acceptable probability of $\neg P4$ occurring, and therefore of rejecting the hypothesis when it was in fact true.

Independence: Correlations

To verify independence of ε from u, use *cross-correlation* function:

$$r_{\varepsilon u}(\tau) = \mathrm{E}\left\{\varepsilon(k+\tau)u(k)\right\}$$

1. If (I1) is true, then $r_{\varepsilon u}(\tau) = 0$ for $\tau \geq 0$.
2. If (I2) is true, then $r_{\varepsilon u}(\tau) = 0$ for any τ.

Estimation from data and normalization:

$$\widehat{r}_{\varepsilon u}(\tau) = \begin{cases} \frac{1}{N}\sum_{k=1}^{N-\tau} \varepsilon(k+\tau)u(k) & \text{if } \tau \geq 0 \\ \frac{1}{N}\sum_{k=1-\tau}^{N} \varepsilon(k+\tau)u(k) & \text{if } \tau < 0 \end{cases}$$

$$x(\tau) = \frac{\widehat{r}_{\varepsilon u}(\tau)}{\sqrt{\widehat{r}_{\varepsilon}(0)\widehat{r}_{u}(0)}}$$

Independence test at τ

Like before, for large N, $x(\tau)$ is distributed according to Gaussian $\mathcal{N}(0, \frac{1}{N})$, and therefore:

$$P(|x(\tau)| \leq \frac{1.96}{\sqrt{N}}) = 0.95$$

Independence tests

If $|x(\tau)| \leq \frac{1.96}{\sqrt{N}}$, $\forall \tau \geq 0$ supported by the data, then the independence hypothesis (I1) is accepted.
If the condition holds $\forall \tau$ supported by the data (including negative τ), then (I2) is also accepted.

In practice, if the model is accurate (I1 holds), then checking the condition at $\tau < 0$ (I2) verifies the presence of feedback.

Table of contents

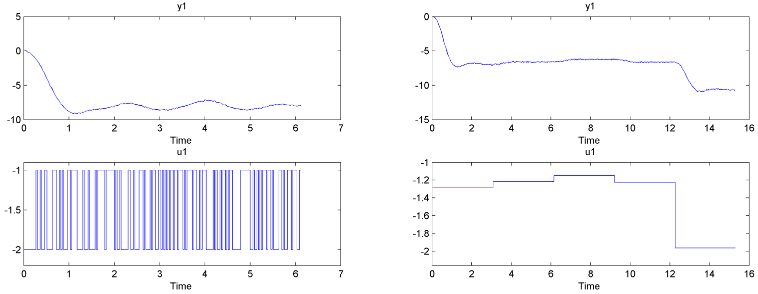
Matlab example: Experimental data

The real system is in output-error form:

$$y(k) = \frac{B(q^{-1})}{F(q^{-1})}u(k) + e(k)$$

and has order $n = 3$.

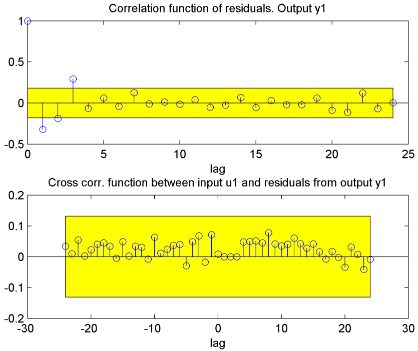
`plot(id);` and `plot(val);`



Matlab: ARX model

First, we try an ARX model:

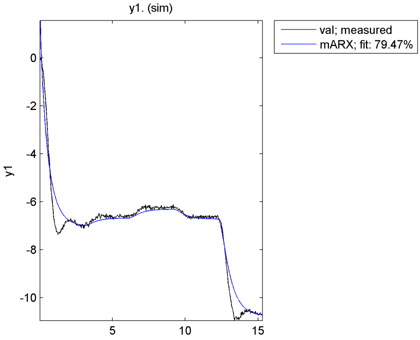
```
mARX = arx(id, [3, 3, 1]); resid(mARX, id);
```



The model fails the whiteness test (W). This is because the system is not within the model class, leading to an inaccurate model. The model is rejected.

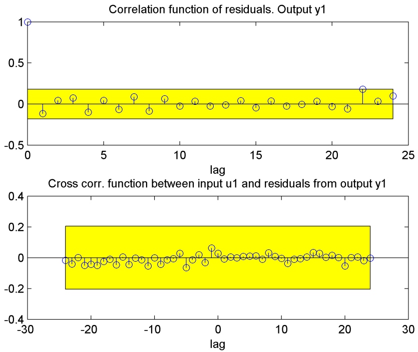
Matlab: ARX model (continued)

Simulating on the validation data confirms the fact that the model is poor.



Matlab: OE model

```
mOE = oe(id, [3, 3, 1]); resid(mOE, id);
```



The OE model passes all the tests – as expected because the OE model class contains the real system. The model is therefore validated.

Important note: The Matlab functions use 0.99 probability instead of 0.95, so they are even less likely of rejecting a correct model.

Matlab: OE model (continued)

Simulating the model on the validation data confirms the model has good quality.

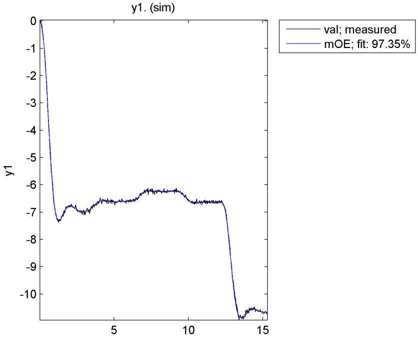
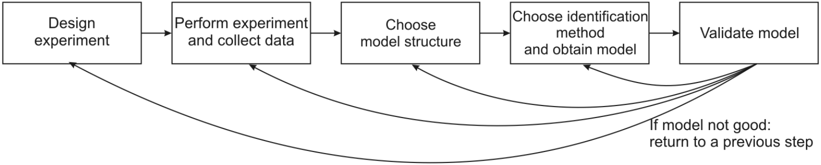


Table of contents

Structure selection in workflow



While we nearly always tuned the model structure (e.g. type, orders, length), the criteria for doing so were often informal.

Next, we discuss structure selection in a formal way.

Consider we are given several model structures $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_\ell$.
Example: ARX structures of increasing order.

How to choose among them?

First idea: choose $\mathcal{M}_i$ leading to the smallest mean squared error:

$$V(\widehat{\theta}) = \frac{1}{N} \sum_{k=1}^{N} \varepsilon(k)^2$$

This does not take into account the complexity of the model, so it ignores the computational effort needed to simulate it, and also has a risk of overfitting. We explore other options that do consider model complexity (without going into their mathematical derivation).

Akaike's information criterion (AIC)

$$W_{\text{AIC}} = N \log V(\widehat{\theta}) + 2p, \text{ or equivalently: } \log V(\widehat{\theta}) + \frac{2p}{N}$$

where N is the number of data points and p the number of parameters (e.g., $na + nb$ in ARX).

Choice: Model with smallest W_{AIC}.

Intuition:

- The term $2p$ penalizes the complexity of the model (number of parameters).
- Division by the number N of data points in $2p/N$ takes into account that more data allows more parameters to be identified.
- Taking the logarithm of the MSE allows to better differentiate between small values of the MSE.

Final prediction error (FPE)

$$W_{\text{FPE}} = V(\widehat{\theta})\frac{1 + p/N}{1 - p/N}$$

Choice: Model with smallest W_{FPE}.

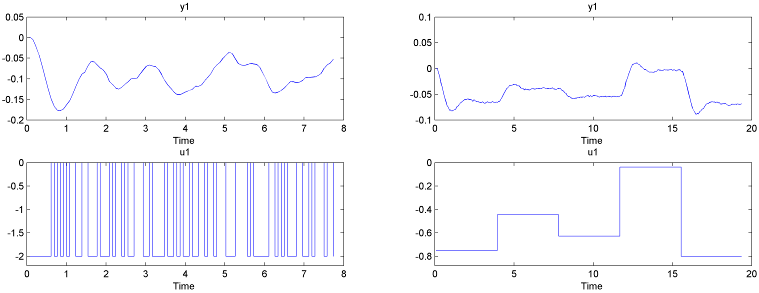
Intuition: When N is large:

$$V(\widehat{\theta})\frac{1 + p/N}{1 - p/N} = V(\widehat{\theta})(1 + \frac{2p/N}{1 - p/N}) \approx V(\widehat{\theta})(1 + \frac{2p}{N})$$

The term $\frac{2p}{N}$ works like before, but now it *multiplies* the MSE rather than getting added.

Matlab example

An OE system with $n = 2$.



Matlab: `selstruc` with AIC

Recall `arxstruc`:

```
Na = 1:15; Nb = 1:15; Nk = 1:5;
NN = struc(Na, Nb, Nk); V = arxstruc(id, val, NN);
```

- `struc` generates all combinations of orders in `Na, Nb, Nk`.
- `arxstruc` identifies for each combination an ARX model on the data `id`, simulates it on the data `val`, and returns information about the MSEs, model orders etc. in `V`.

Matlab: `selstruc` with AIC (continued)

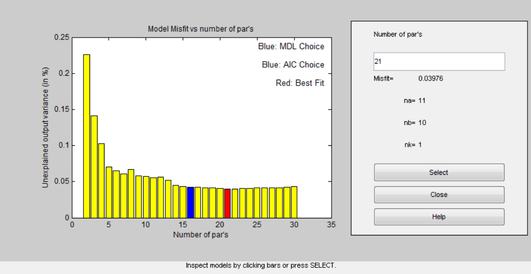
To choose the structure with the Akaike's information criterion:

```
N = selstruc(V, 'aic');
```

For our data, `N`$= [8, 8, 1]$.

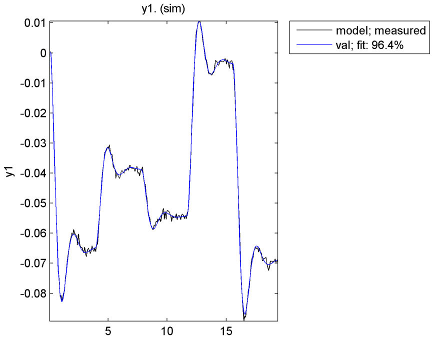
Alternatively, graphical selection also allows using AIC:

```
N = selstruc(V, 'plot');
```



Note that the best-AIC model is not (always) the same as the best-fit model!

Matlab: Results



Remarks

AIC, FPE also work if the system is not in the model class.

Matlab offers functions `aic, fpe` that compute these criteria for a list of models.

Table of contents

Motivation

Consider *the real system* obeys the ARMAX structure:

$$A_0(q^{-1})y(k) = B_0(q^{-1})u(k) + C_0(q^{-1})e(k)$$

where subscript 0 indicates quantities related to the real system.

This is equivalent to any model:

$$W(q^{-1})A_0(q^{-1})y(k) = W(q^{-1})B_0(q^{-1})u(k) + W(q^{-1})C_0(q^{-1})e(k)$$

with $W(q^{-1})$ some polynomial of order nw.

So, using ARMAX identification with $na = na_0 + nw$, $nb = nb_0 + nw$, $nc = nc_0 + nw$ can produce an accurate model. This model is however too complicated (overparametrized), and will have some nearly common factors $W(q^{-1})$ in all polynomials (only "nearly" because of the approximate nature of the identification).

Pole-zero cancellations

This type of situation can be identified by checking if some poles and zeros of the model (approximately) cancel each other out.

We exemplify using Matlab function `pzmap`, which shows the poles and zeros of G in the generic model:

$$y(k) = G(q^{-1})u(k) + v(k)$$

For the ARMAX example, $G(q^{-1}) = \frac{W(q^{-1})B_0(q^{-1})}{W(q^{-1})A_0(q^{-1})}$, so the roots of W are both poles and zeros and (approximately) cancel each other out.

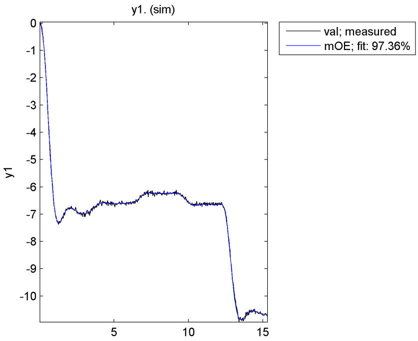
This idea extends to other model types besides ARMAX.

Matlab: overparameterized OE model

On the same data as for correlation tests (recall system has order $n = 3$):

```
mOE = oe(id, [5, 5, 1]);
```

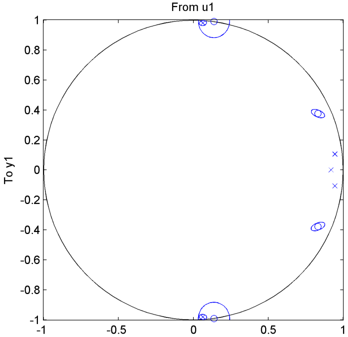
Looking at the validation data, the model is accurate:



Matlab: testing for pole-zero cancellations

```
pzmap(mOE, 'sd', nsd);
```

Arguments `'sd', nsd` specify the confidence region as a number of standard deviations. Here we take `nsd`=1.96 to get 0.95 confidence.



Two pairs of poles and zeros have overlapping confidence regions ⇒ likely they are canceling each other. This indicates the order should be 3 (the true system order).