

# System Identification

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

Lecturer: Lucian Buşoniu



# Part IV

## Correlation analysis

# Table of contents

- 1 Correlation analysis method
  - Analytical development
  - A practical algorithm. FIR model
- 2 Accuracy guarantee (simplified)
- 3 Matlab example

# Context

Transient analysis of step and impulse responses gives a rough, heuristic model of the system.

The upcoming methods provide:

- Fully implementable algorithms, which derive a solution programatically.
- Solution accuracy guarantees (under appropriate conditions).

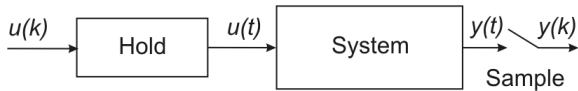
# Classification

Recall **Types of models** from Part I:

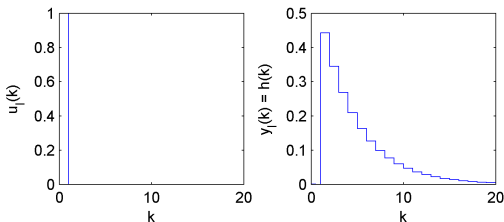
- 1 Mental or verbal models
- 2 **Graphs and tables (nonparametric)**
- 3 Mathematical models, with two subtypes:
  - First-principles, analytical models
  - Models from system identification

Correlation analysis is still a nonparametric method; it produces an *impulse response model*.

# Recall: discrete-time model



# Discrete-time impulse response



Discrete-time, unit impulse signal:

$$u_I(k) = \begin{cases} 1 & k = 0 \\ 0 & k > 0 \end{cases}$$

(does not have area 1, so it's different from the discrete-time realization of the continuous-time impulse!)

Discrete-time impulse response:

$$y_I(k) = h(k), \quad k \geq 0$$

$h(k), k \geq 0$  is also called the **weighting function** of the system.

# Convolution

The (disturbance-free) response to an arbitrary signal  $u(k)$  is the *convolution* of the input and the impulse response:

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

**Intuition:** Consider a signal  $\tilde{u}_j(k)$  equal to  $u(j)$  at  $k = j$ , and 0 elsewhere; just a shifted and scaled unit impulse:

$$\tilde{u}_j(k) = u(j)u_1(k-j)$$

So, the response to  $\tilde{u}_j(k)$  is a shifted and scaled impulse response:

$$\tilde{y}_j(k) = u(j)h(k-j)$$

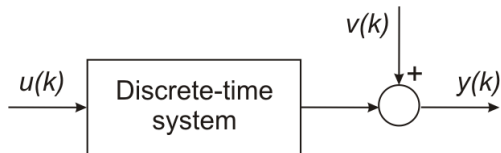
Now,  $u(k)$  is the superposition of all signals  $\tilde{u}_j$ , and due to linearity:

$$y(k) = \sum_{j=0}^k \tilde{y}_j(k) = \sum_{j=0}^k u(j)h(k-j) = \sum_{j=0}^k h(j)u(k-j) = \sum_{j=0}^{\infty} h(j)u(k-j)$$

where zero initial conditions were assumed, i.e.  $u(j) = 0 \forall j < 0$ .



# Impulse-response model



$$y(k) = \sum_{j=0}^{\infty} h(j)u(k-j) + v(k)$$

Includes, in addition to the ideal model, a disturbance term  $v(k)$ .

# Assumptions

## Assumptions

- 1 The input  $u(k)$  is a stationary stochastic process.
- 2 The input  $u(k)$  and the disturbance  $v(k)$  are independent.

## Recall:

- Independence of random variables.
- Stationary stochastic process: constant mean at every time step, covariance only depends on difference between time steps and not on absolute time.

# Covariance functions

The **covariance functions** are defined as follows:

$$r_{yu}(\tau) = E \{y(k + \tau)u(k)\}$$

$$r_u(\tau) = E \{u(k + \tau)u(k)\}$$

**Note:** These quantities are true covariances only when the input and output are zero-mean, so if they are not, then the means must be subtracted prior to applying the algorithm. .

# Relationship of covariances and impulse response

If there were no disturbance, then:

$$\begin{aligned} r_{yu}(\tau) &= E \{y(k + \tau)u(k)\} \\ &= E \left\{ \left[ \sum_{j=0}^{\infty} h(j)u(k + \tau - j) \right] u(k) \right\} \\ &= \sum_{j=0}^{\infty} h(j)E \{u(k + \tau - j)u(k)\} = \sum_{j=0}^{\infty} h(j)r_u(\tau - j) \end{aligned}$$

The errors coming from the disturbance are dealt with later, using linear regression.

# Impulse response identification

Writing the covariance relationship for all  $\tau$ :

$$r_{yu}(0) = \sum_{j=0}^{\infty} h(j)r_u(-j)$$

$$r_{yu}(1) = \sum_{j=0}^{\infty} h(j)r_u(1-j)$$

...

we obtain (in principle) an infinite system of linear equations:

- Coefficients  $r_u(\tau)$ ,  $r_{yu}(\tau)$ .
- Unknowns  $h(0)$ ,  $h(1)$ ,  $\dots$ , solution of the system.

Next, a practical algorithm working with finite data is given.

# Table of contents

- 1 Correlation analysis method
  - Analytical development
  - A practical algorithm. FIR model
- 2 Accuracy guarantee (simplified)
- 3 Matlab example

# Covariances from data

Consider we are given signals  $u(k), y(k)$  with  $k = 1, \dots, N$ .  
Denote by  $\{a(k)\}_{k=b,c,\dots}$  the signal  $a(k)$  at time steps  $b, c, \dots$

We have, for positive  $\tau$ :

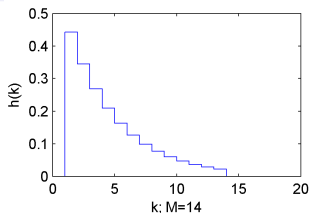
$$\begin{aligned} r_u(\tau) &= \text{E} \{u(k + \tau)u(k)\} \\ &\approx \frac{1}{N - \tau} \sum_{k=1}^{N-\tau} u(k + \tau)u(k) \\ &=: \hat{r}_u(\tau), \quad \forall \tau \geq 0 \end{aligned}$$

and  $\hat{r}_u(-\tau) = \hat{r}_u(\tau)$  for negative values, as  $u$  is a stationary process.

For positive and negative  $\tau$ :

$$\begin{aligned} r_{yu}(\tau) &= \text{E} \{y(k + \tau)u(k)\} \\ &\approx \frac{1}{N - |\tau|} \sum_{k=1 - \min\{\tau, 0\}}^{N - \max\{\tau, 0\}} y(k + \tau)u(k) \\ &=: \hat{r}_{yu}(\tau), \quad \forall \tau \geq 0 \end{aligned}$$

# Finite impulse response model



Impose the condition  $h(k) = 0$  for  $k \geq M$ . We obtain the **finite impulse response (FIR)** model:

$$y(k) = \sum_{j=0}^{M-1} h(j)u(k-j) + v(k)$$

The covariance relationship is similarly truncated:

$$r_{yu}(\tau) = \sum_{j=0}^{M-1} h(j)r_u(\tau-j)$$

**Note:**  $M$  must be taken so that  $MT_s \gg$  dominant time constants (or equivalently, the system is close to steady-state)



# Linear system

Using  $\hat{r}_{yu}$ ,  $\hat{r}_u$  estimated from data, write the truncated equations for  $\tau = 0, \dots, T - 1$ :

$$\hat{r}_{yu}(0) = \sum_{j=0}^{M-1} h(j)\hat{r}_u(-j)$$

$$\hat{r}_{yu}(1) = \sum_{j=0}^{M-1} h(j)\hat{r}_u(1 - j)$$

...

$$\hat{r}_{yu}(T - 1) = \sum_{j=0}^{M-1} h(j)\hat{r}_u(T - 1 - j)$$

This is a linear system of  $T$  equations in  $M$  unknowns  $h(0), \dots, h(M - 1)$ .

## Linear system (continued)

In matrix form:

$$\begin{bmatrix} \hat{r}_{yu}(0) \\ \hat{r}_{yu}(1) \\ \vdots \\ \hat{r}_{yu}(T-1) \end{bmatrix} = \begin{bmatrix} \hat{r}_u(0) & \hat{r}_u(1) & \dots & \hat{r}_u(M-1) \\ \hat{r}_u(1) & \hat{r}_u(0) & \dots & \hat{r}_u(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_u(T-1) & \hat{r}_u(T-2) & \dots & \hat{r}_u(T-M) \end{bmatrix} \cdot \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{bmatrix}$$

(keep in mind that  $\hat{r}_u(-\tau) = \hat{r}_u(\tau)$ ).

Taking  $T = M$  would give *in principle* an exact solution. Due to disturbances and errors, it is a good idea to take  $T > M$ .

Then we can apply the machinery of linear regression (see Part 3) to solve this problem.

# Table of contents

- 1 Correlation analysis method
- 2 Accuracy guarantee (simplified)**
- 3 Matlab example

# Special case: White noise input

## Assumptions

- 1 The input  $u(k)$  is zero-mean white noise.

Then  $r_u(\tau) = 0$  whenever  $\tau \neq 0$  (since white noise is uncorrelated), and  $r_{yu}(\tau) = \sum_{j=0}^{\infty} h(j)r_u(\tau - j)$  simplifies to:

$$r_{yu}(\tau) = h(\tau)r_u(0)$$

leading to the easy algorithm:

$$h(\tau) = \frac{\hat{r}_{yu}(\tau)}{\hat{r}_u(0)}$$

# Simplified guarantee

## Theorem

In the white-noise case, as the number of data points  $N$  grows to infinity, the estimates  $\hat{h}(\tau)$  converge to the true values  $h(\tau)$ .

**Remark:** This type of property, where the true solution is obtained in the limit of infinite data, is called *consistency*.

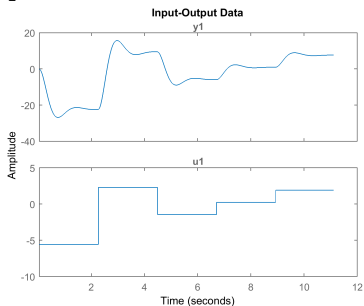
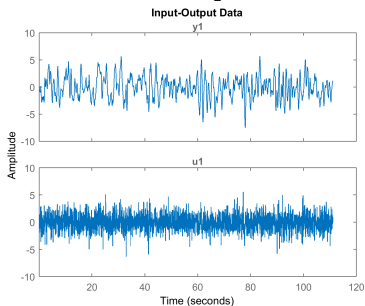
# Table of contents

- 1 Correlation analysis method
- 2 Accuracy guarantee (simplified)
- 3 Matlab example**

# Experimental data

Consider we are given the following, separate, identification and validation data sets.

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



Note the identification input is white noise, while the validation input is not. There are 1100 samples in the identification data.

# Applying correlation analysis

```
fir = cra(id, M, 0); or fir = cra(id, M, 0, plotlevel);
```

Arguments:

- 1 Identification data.
- 2 FIR length  $M$ , here it is set to 45.
- 3 Third argument 0 means no *input whitening* is performed.

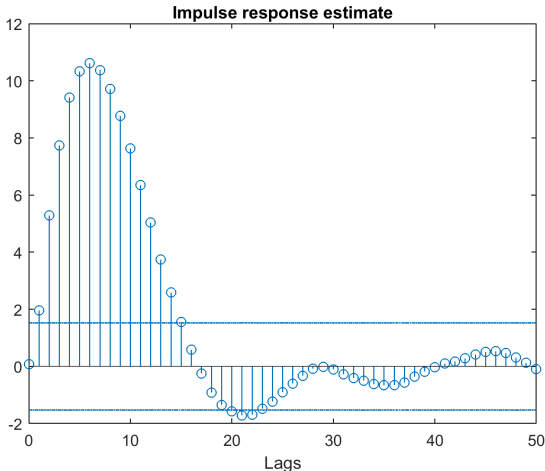
Dealing with non-ideal inputs:

- If input is not zero-mean, pass the data through `detrend` to remove the means.
- If input is not white noise, the third argument should be left to default (by not specifying it or setting it to an empty matrix), which means input whitening is performed.



# Applying correlation analysis (continued)

By default (or with `plotlevel=1`) the FIR parameters are shown with a 99% confidence interval.

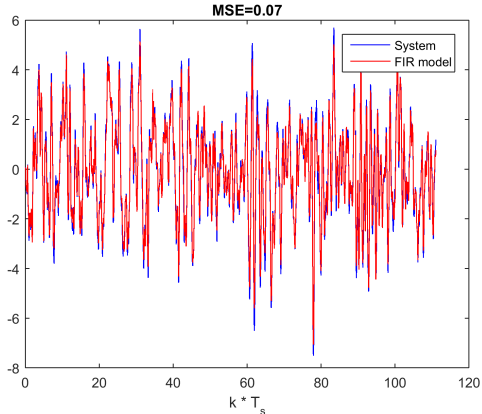


`plotlevel=2` also produces the covariance functions.

# Results on the identification data

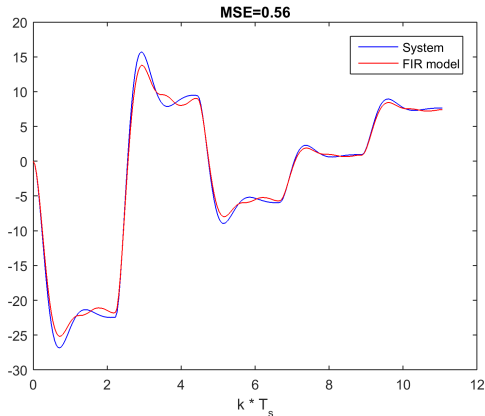
```
yhat = conv(fir, id.u); yhat = yhat(1:length(id.u));
```

To simulate the FIR model, a *convolution* between the FIR parameters and the input is performed. The simulated output is longer than needed so we cut it off at the right length.



# Validation of the FIR model

```
yhat = conv(fir, val.u); yhat = yhat(1:length(val.u));
```



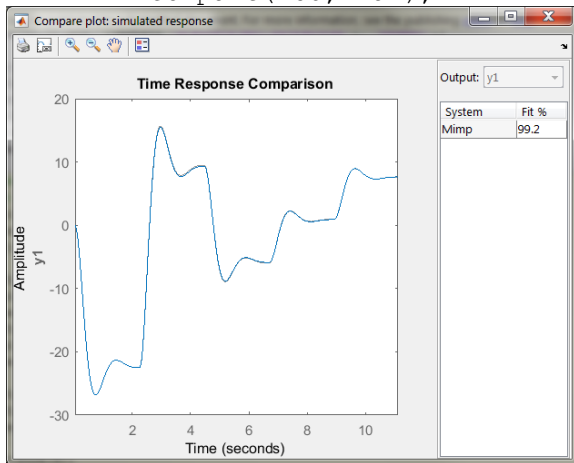
Results OK, not great.

## Alternative: impulseest function

```
model = impulseest(id, M); or model = impulseest(id);
```

Uses a more involved algorithm than the one studied in the lectures.

```
compare(mod, val);
```



The fit is very good.