System Identification: Black Box

Black box - obsah

1. Prerequisites:
   - Least Squares – basic method for parameter estimation
   - Discrete dynamic systems, discretization
2. How to estimate parameters of a linear DDS using the Least Squares method?
   - Example: Mechanical Oscillator
3. Problem: Noise

Discretization

- Linear DDS looks like:

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \ldots + a_n x_{k-n} \quad + b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_m u_{k-m}$$

where *k* is the number of the „simulation step" with sampling time Ts, and $a_1, a_2, \ldots a_n$, b_1, b_2, b_m are parameters.
- LDDS is equivalent to a continuous model of dynamic system.
- Discretization = a way to (roughly) approximate a continuous model by a discrete one.

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \qquad \xrightarrow{Ts} \qquad \mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}\mathbf{u}_k$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \qquad\qquad \mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1}$$

- Approximate forms of the matrices:

$$\mathbf{\Phi} = \mathbf{I} + \mathbf{A}T_s + \frac{\mathbf{A}^2 T_s^2}{2!}$$

$$\mathbf{\Gamma} = (\mathbf{I}T_s + \frac{\mathbf{A}T_s^2}{2!})\mathbf{B}$$

Discretization represents solving an ODE – numerical integration!

Estimating DDS parameters using LS

- Generally, DDS is written as:

$$x_k = a_1 x_{k-1} + a_2 x_{k-2} + \ldots + a_n x_{k-n} \quad + b_1 u_{k-1} + b_2 u_{k-2} + \ldots + b_m u_{k-m}$$

- A very simple example : first order system without input.

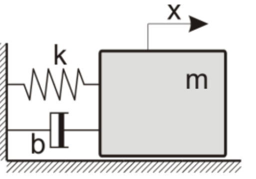
$$\tau \dot{x} + x = 0$$

- Discretization (Euler): $x_{k+1} = \left(1 - \frac{T_s}{\tau}\right) x_k$
- Simplified as: $x_{k+1} = a\, x_k$
- We can see, that a task to find the parameter **a** can be understood as solving many (n-1, k=1...n) equations of one variable – this can be easily formulated as a LS task.

See the exercise about estimating the parameter **a** of a sinusoid.

Example: Parameter estimation of a harmonic oscillator 1/5

- **Task:** Consider a linear harmonic oscillator from the figure. A force **F** is applied to the oscillator. The parameters k,b,m are unknown and we want to estimate them from the measured position **x**.



- For pedagogic reasons, we explain the whole solution including the discretization – it is important **to have a clear idea** about how can the DDS look like, its parameters, where they come from etc..
- A continuous dynamic system is described by these equations, which can be written in a matrix form.

$$m\ddot{x} + b\dot{x} + kx = F = u$$

$$\ddot{x} + \frac{b}{m}\dot{x} + \frac{k}{m}x = \frac{1}{m}u$$

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

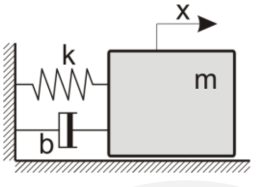
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u_k$$

Example: Parameter estimation of a harmonic oscillator 2/5

- **Discretization**: in general:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}$$
$$\mathbf{y} = \mathbf{C}\mathbf{x}$$

$$\mathbf{x}_{k+1} = \mathbf{\Phi}\mathbf{x}_k + \mathbf{\Gamma}\mathbf{u}_k$$
$$\mathbf{y}_{k+1} = \mathbf{C}\mathbf{x}_{k+1}$$
$$\mathbf{\Phi} = \mathbf{I} + \mathbf{A}T_s + \frac{\mathbf{A}^2 T_s^2}{2!}$$
$$\mathbf{\Gamma} = (\mathbf{I}T_s + \frac{\mathbf{A}T_s^2}{2!})\mathbf{B}$$



specifically:

k = Ts steps counter

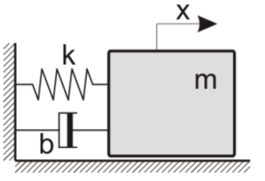
$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{b}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u_k$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T_s \\ -\frac{k}{m}T_s & 1 - \frac{b}{m}T_s \end{bmatrix}$$

$$\mathbf{\Gamma} = \begin{bmatrix} 0 \\ \frac{1}{m}T_s \end{bmatrix}$$

$$x_{k+2} = a_1 x_{k+1} + a_2 x_k + c u_k$$
$$a_1 = 2 - \frac{bT_s}{m}$$
$$a_2 = -\frac{kT_s^2}{m} + \frac{bT_s}{m} - 1$$
$$c = \frac{T_s^2}{m}$$

Example: Parameter estimation of a harmonic oscillator 3/5



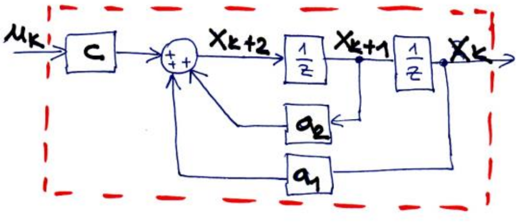
Discretization summary:

- A continuous model is transferred into:

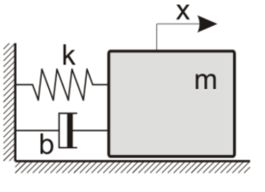
$$x_{k+2} = a_1 x_{k+1} + a_2 x_k + c u_k$$

where the parameters does not have a physical meaning (they depend on the sampling Ts)

$$a_1 = 2 - \frac{bT_s}{m}$$
$$a_2 = -\frac{kT_s^2}{m} + \frac{bT_s}{m} - 1$$
$$c = \frac{T_s^2}{m}$$



Example: Parameter estimation of a harmonic oscillator 4/5

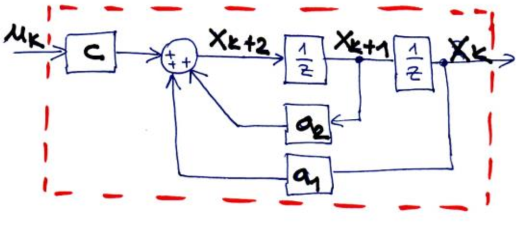


- The model can be written in a clearer form as:

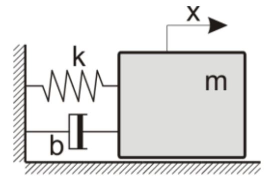
$$x_{k+2} = a_1 x_{k+1} + a_2 x_k + c u_k$$

$$x(T + 2Ts) = a_1 x(T + Ts) + a_2 x(T) + cu(T)$$

where T is the actual time of the discrete simulation.



Example: Parameter estimation of a harmonic oscillator 5/5



3. **Parameter estimation** using the least squares method (OLS):

- model: $x_{k+2} = a_1 x_{k+1} + a_2 x_k + c u_k$
- OLS:

$$y_i = \sum_{j=1}^{n} \theta_j \varphi_j, \; i = 1 \ldots m$$

  - i –equation number= the number of a „measurement“ (i = 1...n)
  - j – number of the parameter (j = 1...3)

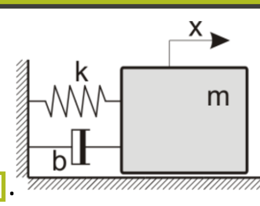
```
%% Simulation
% Ini conditions
x(1,1) = 0;
x(2,1) = 0;
% simulation
for i = 3:length(t)
    x(i,1) = a1*x(i-1) + a2*x(i-2) + c*u(i-2);
    y(i,1) = x(i);
end

%% OLS
Y = y(3:end);
X(:,1) = y(2:end-1);
X(:,2) = y(1:end-2);
X(:,3) = u(1:end-2);

disp('Estimated parameters:')
Theta = inv(X'*X)*X'*Y
```

[ID01]

Exercise: **Parmeter estimation of a harmonic oscillator using OLS** 1/1



- Try to reproduce the task from the previous slides.
- If you have any trouble, try to find inspiration in the examples [ID01] and [ID02].
- Does it work?

- Add noise to the simulation. Consider the noise applied to the output of the position **x** sensor and it has the Normal distribution (Gauss).

```
% simulation
for i = 3:length(t)
    x(i,1) = a1*x(i-1) + a2*x(i-2) + c*u(i-2);
    y(i,1) = x(i) + 0.01*randn;
end
```

Problem = noise: case study of a simple example 1/5

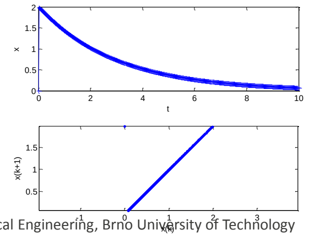
- Consider a first order system $\tau\dot{x} + x = u$
- Discretization:

$$\dot{x} = -\frac{1}{\tau}x + \frac{1}{\tau}u \quad \xrightarrow{Ts} \quad x_{k+1} = (1 - \frac{T_s}{\tau})x_k + \frac{T_s}{\tau}u_k$$

- For simplicity, consider the system has no input (u=0).

$$x_{k+1} = ax_k$$

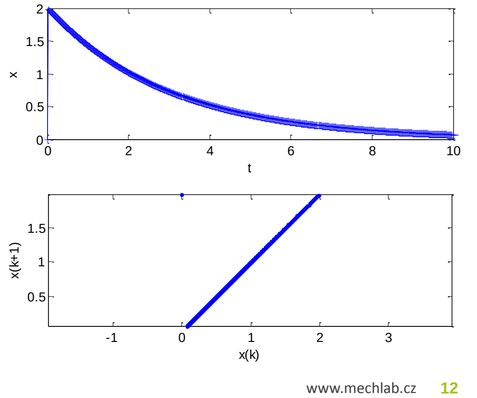
- We estimate a single parameter **a**.
- Simulation:



Problem = noise: case study of a simple example 2/5

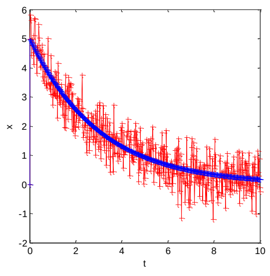
- The initial value is x_0, simulate the equation: $x_{k+1} = ax_k$
- The lower figure shows the relationship between x(k) and x(k+1). According to the equation it is clear that it has to be a line.
- If we have data without noise, **the estimation of the parameter** a **will be precise.**

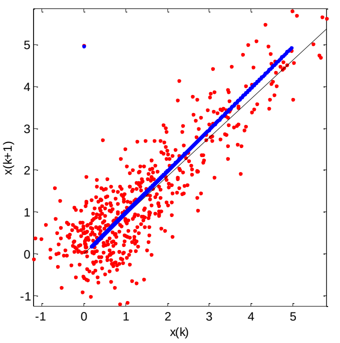
[ID03]



Problem = noise: case study of a simple example 3/5

- Consider a noise applied to the output (for example noisy sensor). The equation would look like: $x_{k+1} = ax_k$
  (*w* is a random number with the Normal distribution) $y_k = x_k + w$
- A **deviation of the estimation** occurs.





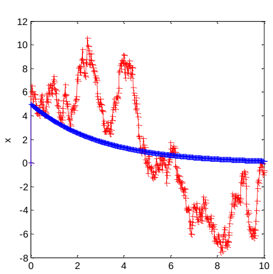

```
% simulation
 x(1) = x0;
    for i = 2:length(t)
        x(i,1) = a*x(i-1);
        y(i,1) = x(i) + randn*0.2;
    end
```

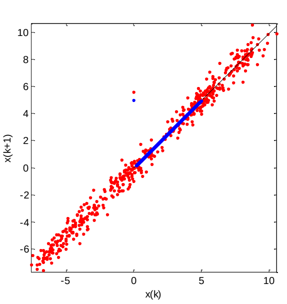
[ID04]

www.mechlab.cz 13
ISMMB, Faculty of Mechanical Engineering, Brno University of Technology

Problem = noise: case study of a simple example 4/5

- Further consider noise „inside“ the systém, which modifies the state of x(k). $x_{k+1} = ax_k + w$
- This noise can be considered as $y_k = x_k$
  a random disturbance (input) influencing the system.
- Surprisingly, the **deviation of the parameter** a **estimation** does **not occur**.






```
% simulation
 for i = 2:length(t)
        x(i,1) = a*x(i-1) + randn*0.2;
        y(i,1) = x(i) ;
    end
```

[ID04]

www.mechlab.cz 14
ISMMB, Faculty of Mechanical Engineering, Brno University of Technology

Problem = noise: case study of a simple example 5/5

Summary:

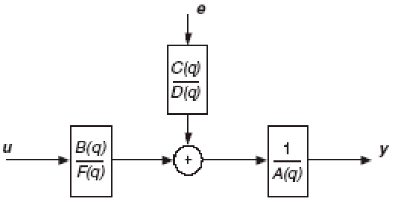
- The least squares method works well for parameter estimation of a linear discrete system without noise.
- The noise added to the system output deviates the estimation.
  Tle Least squares method cannot be used for parameter estimation of systems with Output Error (OE).
- The noise added to the system input does not deviate the estimation.
  The least squares method can be used for an ARX type model.

mechatronics laboratory

www.mechlab.cz 15
ISMMB, Faculty of Mechanical Engineering, Brno University of Technology

Polynomial models for system identification

- Linear system
  - u – input
  - y – output
  - e – disturbance, consider white noise.



- Meaning of the operator **q**:
  - q = forward shift operator
  - example 1: $q^{-1}\,x(k) = x(k-1)$
  - example 2:

$$y(t)+a_1y(t-T)+a_2y(t-2T)=$$
$$b_1u(t-T)+b_2u(t-2T)$$

$$A(q)y(t)=B(q)u(t)$$
$$y(t)+a_1q^{-1}y(t)+a_2q^{-2}y(t)=$$
$$b_1q^{-1}u(t)+b_2q^{-2}u(t)$$

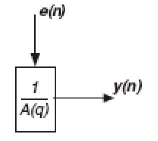
$$A(q)=1+a_1q^{-1}+a_2q^{-2}\ \text{and}\ B(q)=b_1q^{-1}+b_2q^{-2}\,.$$

mechatronics laboratory

Polynomial models for system identification

Simple examples:

$$y(k) = \frac{1}{D(q)} v(k)$$



- AR (autoregressive) model

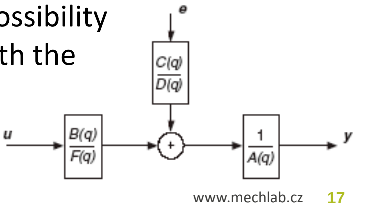
$$y_k + a_1 y_{k-1} + a_2 y_{k-2} + ... = e_k$$

- MA (moving average) model

$$y(k) = \frac{C(q)}{1} v(k)$$

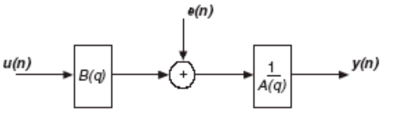
$$y_k = c_k e_k + c_{k-1} e_{k-1} + c_{k-2} e_{k-2} + ...$$

- These two model explain why the is the general diagram so ,,complicated‘‘ and contains the polynomials A,B,C,D,F. The reason is that we want to have the possibility to model the AR and MA dynamics of both the input **u** and the noise **e** side, and their combination.



Polynomial models for system identification

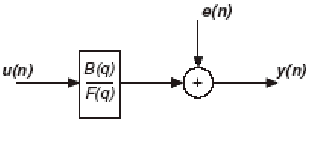
Most important models:



- **ARX**
  - advantage: simple, solved directly by LS.
  - disadvantage: often models the system dynamics UNREALISTICALLY.
  - When it can be used: in case of a good signal-to-noise ratio.

$$y(k) = \frac{B(q)}{A(q)} u(k) + \frac{1}{A(q)} v(k)$$

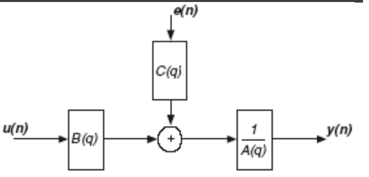
- **OE (output error)**
  - Noise only on the output
  - = the disturbance has no dynamics
  - example: deterministic system, noisy sensor (potentiometer)



Polynomial models for system identification

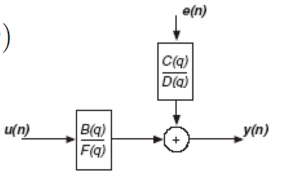
Other models:

- **ARMAX** $$y(k) = \frac{B(q)}{A(q)}u(k) + \frac{C(q)}{A(q)}v(k$$



  - Unlike ARX includes the disturbance dynamics
  - When to use: if the disturbance is applied to the system ,,sooner‘‘.
- **BJ (Box-Jenkins)**
  - The system dynamics and the disturbance dynamics are modeled separately.

$$y(k) = \frac{B(q)}{F(q)}u(k) + \frac{C(q)}{D(q)}v(k)$$

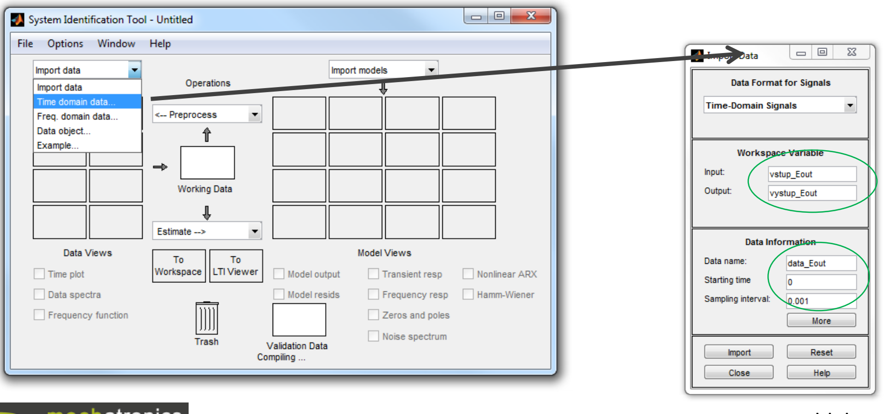


Black-box identification
- System Identification tool/GUI

System Identification Toolbox - GUI

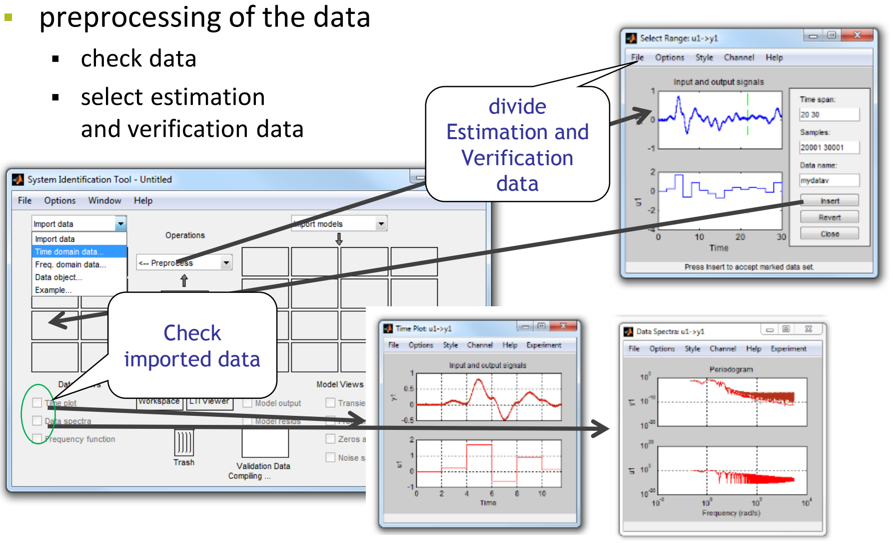
- System Identification Toolbox
  - very (very) complex tool, author L. Ljung
  - you can work using
    - command line
    - GUI
- how to run it?: `ident`

System Identification Toolbox – GUI: Import data



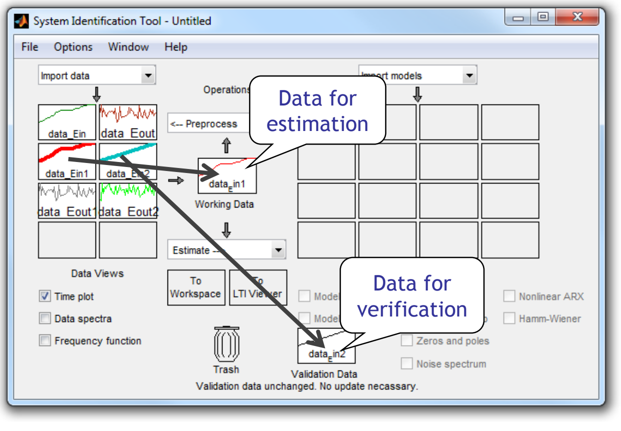
System Identification Toolbox – GUI: Preprocessing

- preprocessing of the data
  - check data
  - select estimation and verification data



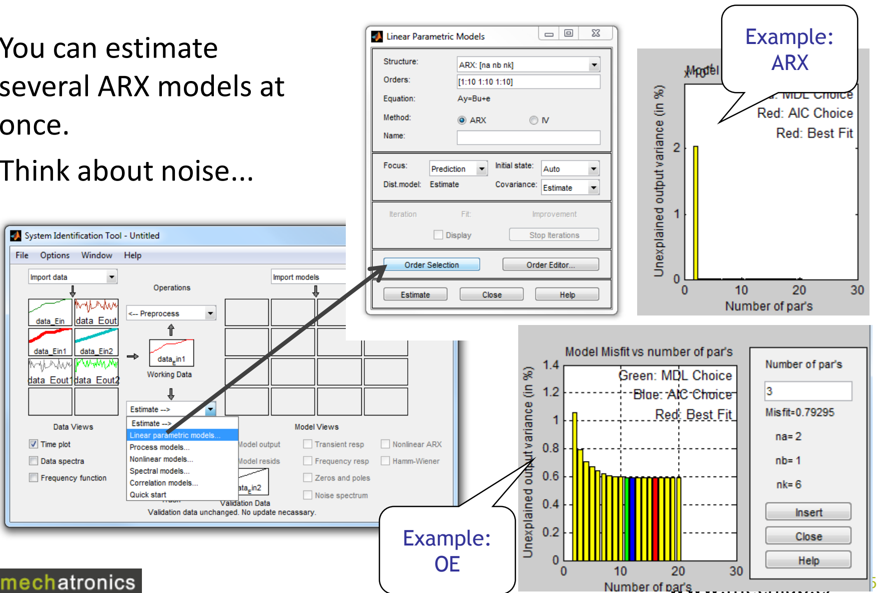
System Identification Toolbox – GUI: Select data

- Select data for Estimation and for Verification

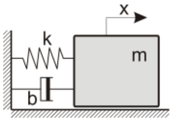


System Identification Toolbox – GUI: Estimate order of the system

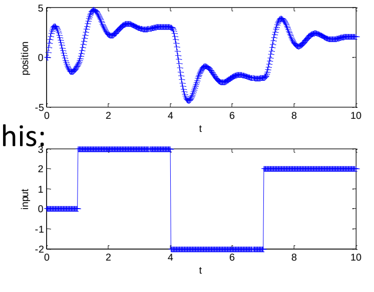
- You can estimate several ARX models at once.
- Think about noise...



ID05: Black-box identification using SIT/GUI 1/8



- The purpose of this exercise is to try working with the System Identification toolbox GUI.
- The modeled systém is an oscillator with input (force).
- The data can be generated by the script **ID05**.
- A simulation WITHOUT noise looks like this:

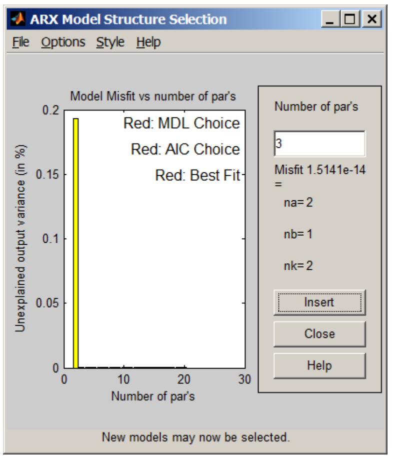


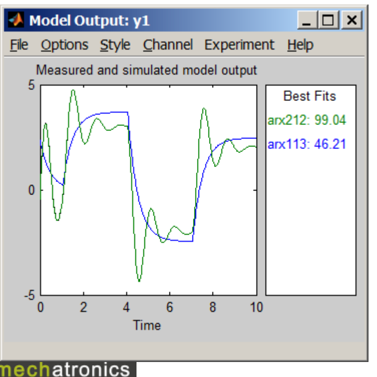

```
%% Simulation
x(1) = 0.5;
x(2) = 0.5;
for i = 3:length(t)
    x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2);
    y(i,1) = x(i);
end
```

- Start the GUI/SIT with the `ident` command, load the data and try to estimate the system order (using ARX).

ID05: Black-box identification using SIT/GUI 2/8

- Order selection:
- Explain!

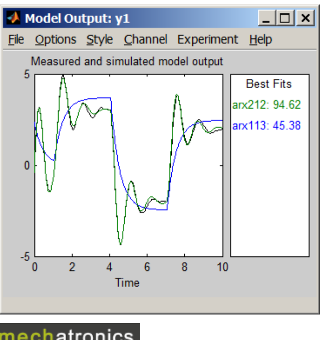


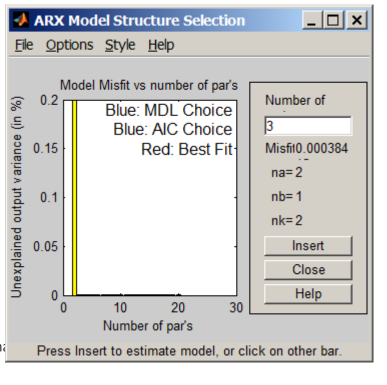


ID05: Black-box identification using SIT/GUI 3/8

- Model the system using an ARX noise:

```
%% Simulation
x(1) = 0.5;
x(2) = 0.5;
for i = 3:length(t)
     x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2)   + 0.005*randn;
    y(i,1) = x(i);
end
```



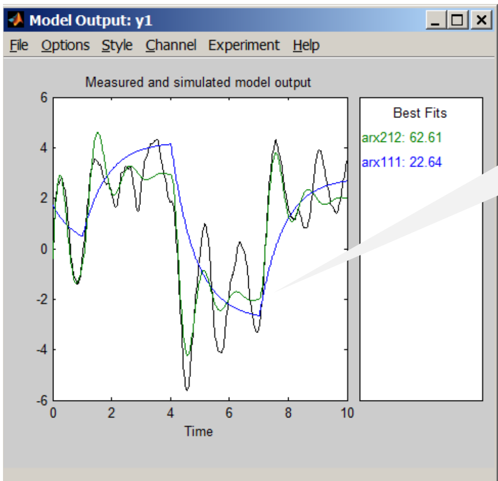


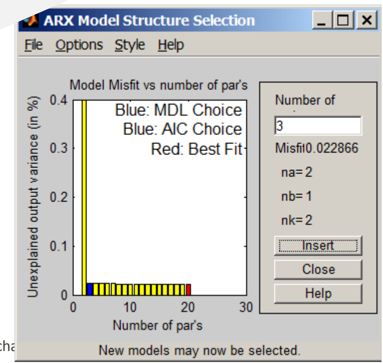
ID05: Black-box identification using SIT/GUI 4/8

- Model the system using 4x more ARX noise:

```
x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2)  + 0.02*randn;
```

We can see that the simulation cannot be exactly fitted to the measured output.





ID05: Black-box identification using SIT/GUI 5/8

- Thanks to the noise (ARX), we cannot precisely fit the simulation to the measured data.
- But how precise is the parameter estimation?
- SIT:
  - Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
    A(z) = 1 - **1.946** z^-1 + **0.9588** z^-2
    B(z) = **0.01229** z^-2
- Simulation:

```
x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2)  + 0.02*randn;
```

where: a1 = **1.9333**, a2 = -**0,94666**, b = **0,1333**

ID05: Black-box identification using SIT/GUI 6/8

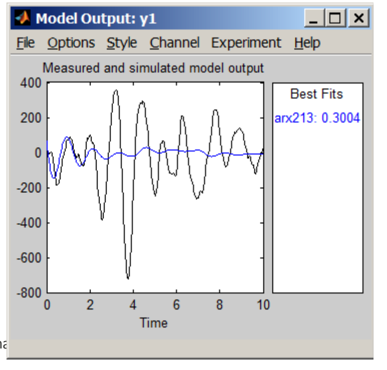
- Model the system with 1000x more ARX noise:

```
x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2)  + 5*randn;
```

- SIT:
  - Discrete-time ARX model: A(z)y(t) = B(z)u(t) + e(t)
    A(z) = 1 - **1.956** z^-1 + **0.9669** z^-2
  - B(z) = -**0.05404** z^-3
- Simulation:

where: a1 = **1.9333**, a2 = **-0,94666**,
b = **0,1333**

Explain!



mechatronics laboratory

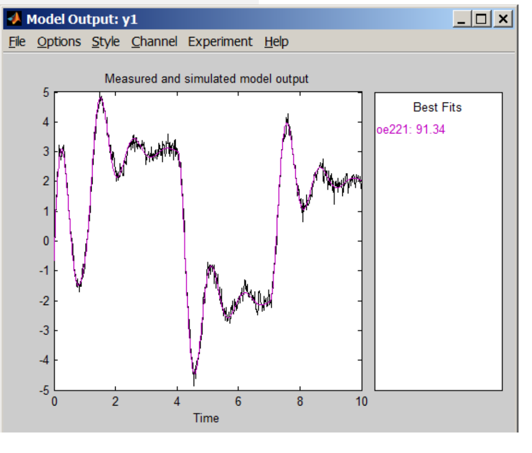
ISMMB, Faculty of Mecha

ID05: Black-box identification using SIT/GUI 7/8

- Model the system with OE noise:

```
x(1) = 0.5;
x(2) = 0.5;
for i = 3:length(t)
    x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2);
    y(i,1) = x(i) + 0.2*randn ;
end
```

- Result:



mechatronics laboratory

IS

ID05: Black-box identification using SIT/GUI 8/8

- Model the system with a combination of ARX and OE noise:

```
x(1) = 0.5;
x(2) = 0.5;
for i = 3:length(t)
    x(i,1) = a1*x(i-1) + a2*x(i-2) + b*u(i-2) + 0.02*randn;
    y(i,1) = x(i) + 0.2*randn ;
end
```

- Result?

Appendix

- Offline vs. online parameter estimation
- IV (Instrumental variable) method
- Nonlinear least squares

Offline vs. online Parameter Estimation

- batch estimation
  - measurement in time (0,T)
  - data processing
  - parameter estimation using all data
- online estimation
  - parameter estimation during system operation
  - recursive algorithm
  - advantages:
    - » memory requirements (often) reduced – OLS vs. RLS
    - » the slow changes of parameters can be captured

Recursive Least Squares (RLS) (Plackett's algorithm)

- online estimation
- gives identical result as OLS
- assume model $y_k = \sum_{r=1}^{p} a_r y_{k-r} + \sum_{r=1}^{q} b_r u_{k-r}$
- in matrix form: $y_k = \mathbf{x}_{k-1}^T \boldsymbol{\theta}$

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \end{bmatrix}$$

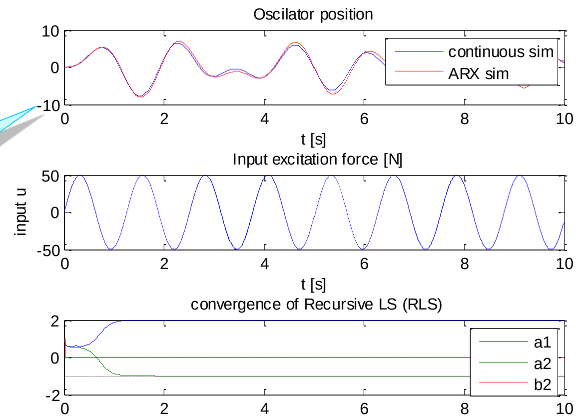
$$\boldsymbol{\theta}_k = \boldsymbol{\theta}_{k-1} - \frac{\mathbf{P}_{k-1}\mathbf{x}_{k-1}(\mathbf{x}_{k-1}^T\boldsymbol{\theta}_{k-1} - y_k)}{1 + \mathbf{x}_{k-1}^T \mathbf{P}_{k-1}\mathbf{x}_{k-1}}$$

$$\mathbf{P}_k = \mathbf{P}_{k-1} - \frac{\mathbf{P}_{k-1}\mathbf{x}_{k-1}\mathbf{x}_{k-1}^T \mathbf{P}_{k-1}}{1 + \mathbf{x}_{k-1}^T \mathbf{P}_{k-1}\mathbf{x}_{k-1}}$$

OLS & RLS example:
Estimation of mass+spring+damper system

- Solve LS/RLS problem
  - both LS and RLS provide the same results
  - convergence of RLS/precision of LS depends on excitation signal
- {E003}

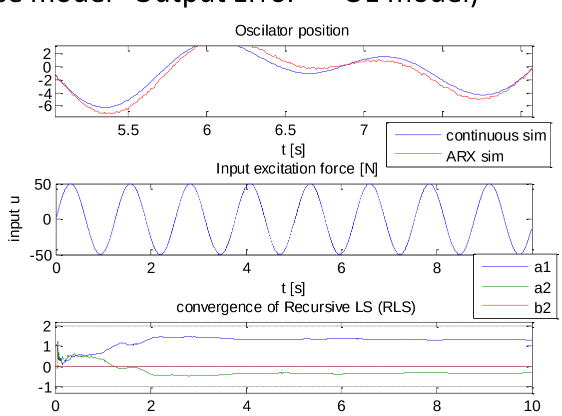
No noise considered in our simulation!



mechatronics laboratory

OLS & RLS example:
Estimation of mass+spring+damper system **with noise**

- Assume noise added to measurement (e.g. if potentiometer used)
- noise_stdvar = 0.1;
- (later, we name this noise model “Output Error” = OE model)
- {E003}



mechatronics laboratory

Estimation of OE: Instrumental Variable (IV) Method

- solution for consistency problem
- 1. Estimate ARX $\boldsymbol{\theta}_{ARX} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$
- 2. Simulate model with estimated parameters $y_u = \frac{\hat{B}(q)}{\hat{A}(q)}u(k)$
- 3. Create instrumental matrix **Z**

$$\mathbf{Z} = \begin{bmatrix} -y_u(m) & \dots & -y_u(1) & u(m) & \dots & u(1) \\ -y_u(m+1) & \dots & -y_u(2) & u(m+1) & \dots & u(2) \\ \vdots & \vdots & \vdots & \vdots & & \\ -y_u(n-1) & \dots & -y_u(n-m) & u(n-1) & \dots & u(n-m) \end{bmatrix}$$

- 4. Estimate $\boldsymbol{\theta}_{IV} = (\mathbf{Z}^T\mathbf{X})^{-1}\mathbf{Z}^T\mathbf{y}$
- repeat steps 2-4 (usually upto 3-4 steps) [E009]

Nonlinear Least Squares 1/2

- motivation: need to find parameters for functions which are **nonlinear in parameters**
- assume vector field $\mathbf{f} : \mathbb{R}^n \mapsto \mathbb{R}^m \text{ kde } m \geq n$
- goal: minimize criterion $F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m}(f_i(\boldsymbol{\theta}))^2$
- Taylor series

$$F(\boldsymbol{\theta}+\mathbf{h}) = F(\boldsymbol{\theta}) + \mathbf{h}^T\mathbf{g} + \frac{1}{2}\mathbf{h}^T\mathbf{H}\mathbf{h}$$

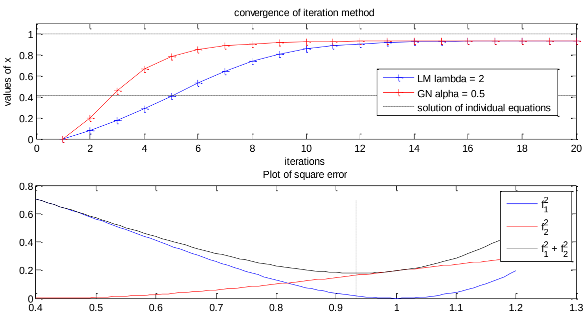
  – gradient: $\mathbf{g} = \begin{bmatrix} \frac{\partial F}{\partial \theta_1}(\boldsymbol{\theta}) \\ \dots \\ \frac{\partial F}{\partial \theta_n}(\boldsymbol{\theta}) \end{bmatrix} = \mathbf{J}(\boldsymbol{\theta})^T\mathbf{f}(\boldsymbol{\theta})$

Nonlinear Least Squares 2/2

- all NLS methods are iterative $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \alpha \mathbf{h}$
- methods for computation of **h**:
  - Steepest descent $\mathbf{h}_d = -\mathbf{g}(\boldsymbol{\theta})$
  - Newton's method $\mathbf{H}\mathbf{h}_n = -\mathbf{g}(\boldsymbol{\theta})$
  - Gauss-Newton $(\mathbf{J}^T\mathbf{J})\mathbf{h}_{gn} = -\mathbf{J}^T\mathbf{f}$
  - **Levenberg-Marquardt** $(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\mathbf{h}_{lm} = -\mathbf{J}^T\mathbf{f}$

Example NLS: Solution of overdetermined system

- Assume vector field $\mathbf{f} = \begin{bmatrix} x^2 - 1 \\ \sin(x) - 0.4 \end{bmatrix} = \mathbf{0}$
- clearly, the system is overdetermined
- x = 1 / 0.4115
- Task: Find solution which has minimal mean square error.
- [E002]



References

- [Iserman 2011] Isermann, R., Muenchhof, M.: Identification of Dynamics Systems, Springer 2011
- [Nelles, 2001] O. Nelles: Nonlinear system identification: from classical approaches to neural networks and fuzzy models, Springer-Verlag, 2001
  google books
- [Noskievič, 1999] P. Noskievič: Modelování a identifikace systémů, Montanex, 1999
- [Ljung 2008] L. Ljung: Perspectives on System Identification, Seoul 2008, http://www.control.isy.liu.se/~ljung/
- [Denery 2005] T. Denery: Modeling Mechanical, Electric and Hydraulic Systems in Simulink, Matlab webinars, 2005
- [Blaha] doc. Ing. Petr Blaha, PhD., FEKT, CAK
  http://sites.google.com/site/modelovaniaidentifikace/
- [Gander, 1980] Gander, W.: Algorithms for QR-Decomposition, research report, 1980
- http://www.emeraldinsight.com/journals.htm?articleid=1943785&show=html
- http://courses.ae.utexas.edu/ase463q/design_pages/fall02/AWE_web/secondorder.htm
- http://www.ni.com/white-paper/4028/en/

Contact: www.mechlab.cz
Institute of Solide Mechanics, Mechatronics and Biomechanics
Faculty of Mechanical Engineering
Brno Uneversity of Technology
doc. Ing. Robert Grepl, Ph.D.
grepl@fme.vutbr.cz