ESS101
Modelling and Simulation, 2025

Lecturer and examiner: Yasemin Bekiroğlu
Course assistant: Ahmet Tekden

Systems & Control Division
Department of Electrical Engineering
Chalmers University of Technology

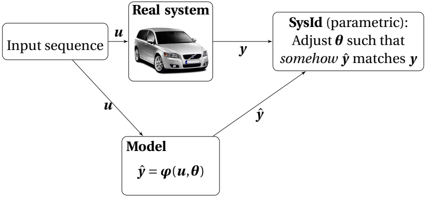
September, 2025

Lecture 6 – System Identification

- Recap on linear regression and least-squares
- Least-squares for dynamic systems
- The prediction error method

The system identification problem

SysId: Adjust the model (with adjustable parameters) to data.



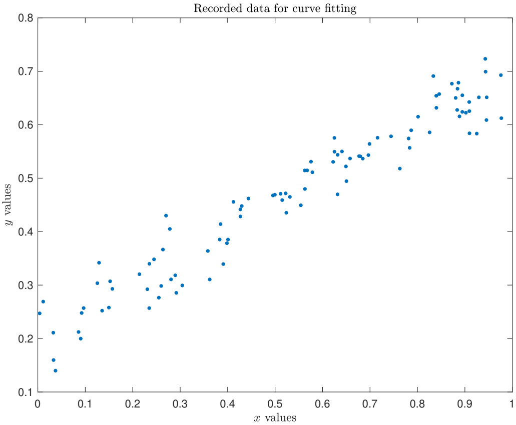
Some of the key issues:

- Experiment design: *selection of inputs and outputs* to be used and construction of the input sequence $\mathbf{u}$ to be applied to the system.
- Selection of *model structure*: the model $\hat{\mathbf{y}}(\mathbf{u}, \boldsymbol{\theta})$ can take various forms, allowing e.g. both linear and nonlinear dynamics, different parametrizations etc.
- Algorithm design: define *what is a good fit of the model to data*, and how to find the best model parameter vector $\boldsymbol{\theta}$.
- Model validation: *assess the resulting model* and whether it fills its purpose? (simulation, statistical tests)

Example: Curve fitting using linear regression

Data: $x(i), y(i), \quad i = 1, \ldots, N$

Model: $y(i) = a + b \cdot x(i) = \theta^\top \varphi(i), \quad \theta = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \varphi(i) = \begin{bmatrix} 1 \\ x(i) \end{bmatrix}$



Linear regression and least-squares

Consider the *linear-in-the-parameters* model

$$y(i) = \theta^\top \varphi(i), \qquad \theta = [\theta_1 \cdots \theta_d]^\top$$

where the *regression vector* $\varphi(i)$ contains known, deterministic signals.
Example: Polynomial trend.

The *least-squares (LS)* criterion is defined as

$$V_N(\theta) = \frac{1}{N} \sum_{i=1}^{N} \varepsilon^2(i, \theta),$$

where the *residual* ε expresses the discrepancy between data and model:

$$\varepsilon(i, \theta) = y(i) - \hat{y}(i|\theta) = y(i) - \theta^\top \varphi(i).$$

The *least-squares estimate* minimizes the criterion, i.e.

$$\hat{\theta}_N = \arg\min V_N(\theta)$$

Solution to the LS problem

The LS criterion can be written as:

$$\mathbf{y} = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}, \qquad \Phi = \begin{bmatrix} \varphi^\top(1) \\ \vdots \\ \varphi^\top(N) \end{bmatrix}, \tag{1}$$

$$V_N(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{y} - \Phi\boldsymbol{\theta}\|^2 = \frac{1}{2}(\mathbf{y} - \Phi\boldsymbol{\theta})^\top(\mathbf{y} - \Phi\boldsymbol{\theta}) \tag{2}$$

The LS solution is found by:

$$\frac{dV_N(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \boldsymbol{\theta}^\top\Phi^\top\Phi - \mathbf{y}^\top\Phi = 0, \tag{3}$$

giving

$$\hat{\boldsymbol{\theta}}_N = (\Phi^\top\Phi)^{-1}\Phi^\top\mathbf{y}, \tag{4}$$

$$\hat{\boldsymbol{\theta}}_N = R_N^{-1}f_N = \Big(\frac{1}{N}\sum_{i=1}^{N}\varphi(i)\varphi^\top(i)\Big)^{-1}\frac{1}{N}\sum_{i=1}^{N}\varphi(i)y(i) \tag{5}$$

Solution to the LS problem

The *LS estimate* is

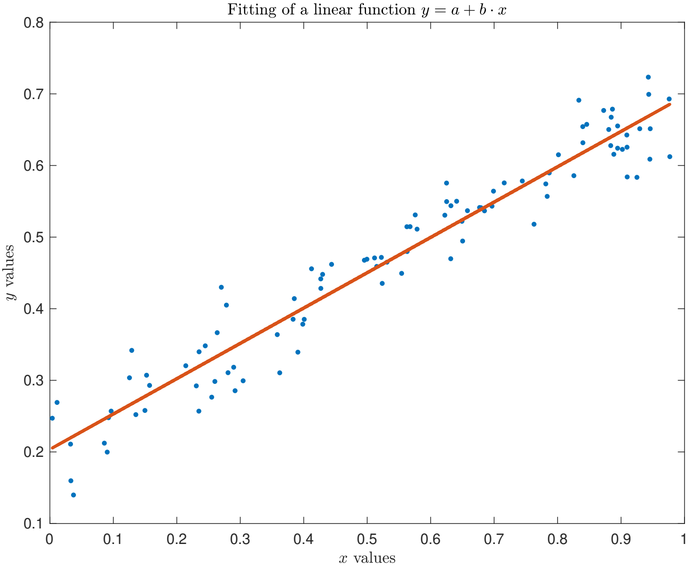
$$\hat{\theta}_N = R_N^{-1} f_N$$

$$\hat{\boldsymbol{\theta}}_N = \begin{bmatrix} \hat{a}_N \\ \hat{b}_N \end{bmatrix} = \Big(\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\varphi}(i)\boldsymbol{\varphi}^\top(i)\Big)^{-1} \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\varphi}(i)y(i)$$

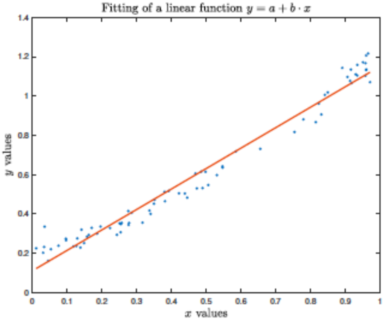
$$\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\varphi}(i)\boldsymbol{\varphi}^\top(i) = \frac{1}{N}\begin{bmatrix} N & \sum_{i=1}^{N} x(i) \\ \sum_{i=1}^{N} x(i) & \sum_{i=1}^{N} x^2(i) \end{bmatrix}$$

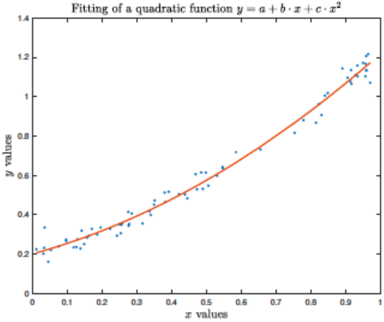
$$\frac{1}{N}\sum_{i=1}^{N} \boldsymbol{\varphi}(i)y(i) = \frac{1}{N}\begin{bmatrix} \sum_{i=1}^{N} y(i) \\ \sum_{i=1}^{N} x(i)y(i) \end{bmatrix}$$

Curve fitting, cont'd

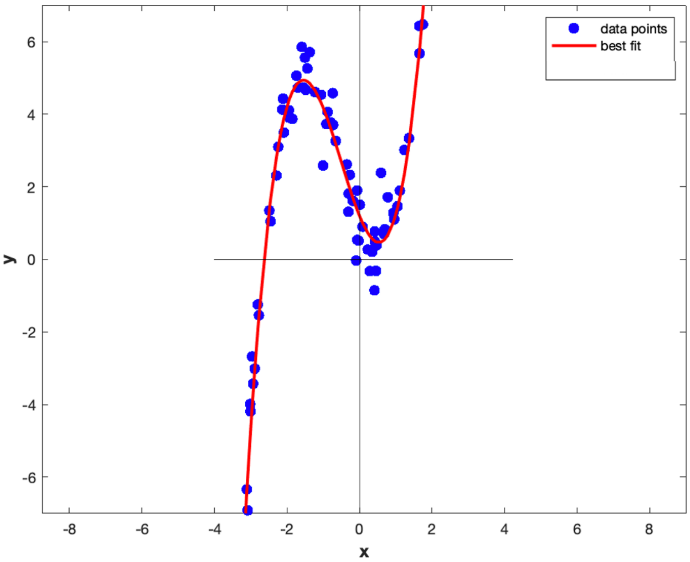


Curve fitting examples





Curve fitting examples



Random variables

Definition (Random variable, CDF)

A real *random variable* (r.v.) X is defined by its (cumulative) *distribution function (CDF)*, describing the probability that X takes a value less than or equal x:

$$F_X(x) = \mathbb{P}[X \leq x]$$

Definition (Probability density function)

The *probability density function (PDF)* $f_X(x)$ of a continuous r.v. is defined by

$$F_X(x) = \int_{-\infty}^{x} f_X(y)dy$$

Definition (Expected value)

The *expected value* of a function $g(X)$ of a r.v. X with PDF $f_X(x)$ is given by

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x)f_X(x)dx$$

Normal distribution

Definition (Normal distribution)

A scalar random variable X with *normal (Gaussian) distribution* has the PDF

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

where μ is the *mean*, σ is the *standard deviation*, and σ^2 is the *variance*.
Notation: $X \sim \mathcal{N}(\mu, \sigma^2)$.

Definition (Multivariate normal distribution)

A vector random variable $X = (X_1, \ldots, X_n)$ with *(multivariate) normal (Gaussian) distribution* has the PDF

$$f_X(x) = \frac{1}{(2\pi)^{n/2}} \frac{1}{(\det \Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)},$$

where μ is the *mean* and Σ is the *covariance matrix*. Notation: $X \sim \mathcal{N}(\mu, \Sigma)$.

Variance, Covariance

EXPECTED VALUE. The expected value or *expectation* of a function $g(X)$ of a r.v. X with PDF $f_X(x)$ is given by

$$\mathbb{E}[g(X)] = \int_{-\infty}^{\infty} g(x) f_X(x) dx.$$

The expected value can be thought of as "the average taken over many experiments", i.e. if one were to draw the random variable X many times and average the result of $g(X)$, one would get something close to the expected value.
The *mean* μ and the *variance* λ of a r.v. X are particular expected values:

$$\mu = \mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx, \qquad \lambda = \mathrm{Var}[X] = \mathbb{E}[(X-\mu)^2] = \int_{-\infty}^{\infty} (x-\mu)^2 f_X(x) dx. \tag{1.68}$$

The *covariance* of two jointly distributed random variables X and Y is defined as

$$\mathrm{Cov}(X, Y) = \mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])] \tag{1.69}$$

In the multivariate case, we will need the concept of covariance with itself (*auto-covariance*), and we will refer to the *covariance matrix* defined as

$$\mathrm{Cov}\, \boldsymbol{X} = \mathbb{E}[(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])(\boldsymbol{X} - \mathbb{E}[\boldsymbol{X}])^\top]. \tag{1.70}$$

Properties of the LS estimate

Assume that the data is generated by the true system

$$y(i) = \theta_0^T \varphi(i) + e(i), \quad e(\cdot) \text{ i.i.d. with variance } \sigma^2$$

Then the following holds for the LS estimate $\hat{\theta}_N$:

1. If biased, the model is unable to capture the true system dynamics. The estimate is *unbiased*:

$$\mathbb{E}[\hat{\theta}_N] = \mathbb{E}[(\frac{1}{N}\sum_{i=1}^{N} \varphi(i)\varphi^T(i))^{-1} \frac{1}{N}\sum_{i=1}^{N} \varphi(i)y(i)]$$

$$= \theta_0 + (\frac{1}{N}\sum_{i=1}^{N} \varphi(i)\varphi(i)^T)^{-1} \cdot \mathbb{E}[\frac{1}{N}\sum_{i=1}^{N} \varphi(i)e(i)] = \theta_0$$

2. Variance - fluctuations in the estimated model due to random disturbances (typically reduced by using larger data sets). The *covariance* of the parameter estimate is:

$$\mathbb{E}[(\hat{\theta}_N - \theta_0)(\hat{\theta}_N - \theta_0)^T] = R_N^{-1}\,\mathbb{E}[(\frac{1}{N}\sum \varphi e)(\frac{1}{N}\sum \varphi e)^T]R_N^{-1} = \frac{\sigma^2}{N} R_N^{-1}$$

Parametric identification

Parametric identification aims at determining models that are parametrized (e.g. state model, transfer function).

- Tailor-made (*white box*) models from physics , e.g.

$$\dot{x}(t) = f(x(t), u(t), \theta) \qquad \theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_d \end{bmatrix}$$
$$y(t) = h(x(t), \theta)$$

- General-purpose (*black-box*) models, e.g.

$$y(t) = G(q, \theta)u(t) + w(t) = G(q, \theta)u(t) + H(q, \theta)e(t)$$

where

$$G(q, \theta) = \frac{B(q, \theta)}{F(q, \theta)} = \frac{b_1 q^{-1} + \ldots + b_{n_b} q^{-n_b}}{1 + f_1 q^{-1} + \ldots + f_{n_f} q^{-n_f}}$$
$$H(q, \theta) = \frac{C(q, \theta)}{D(q, \theta)} = \frac{c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}}{1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d}}$$

Least squares for dynamic systems

The *ARX (Auto-Regressive with eXogenous input)* or equation error model

$$A(q)y(t) = B(q)u(t) + e(t)$$

can be written as a linear regression:

$$y(t) = \theta^T \varphi(t) + e(t)$$

with

$$\begin{aligned} \varphi^T(t) &= [\, -y(t-1) \cdots - y(t-n_a) \;\; u(t-1) \cdots u(t-n_b) \,] \\ \theta^T &= [\, a_1 \cdots a_{n_a} \;\; b_1 \cdots b_{n_b} \,] \end{aligned}$$

The LS estimate can be computed as before:

$$\hat{\theta}_N = \arg\min_{\theta} \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \theta) = (\frac{1}{N} \sum \varphi \varphi^T)^{-1} (\frac{1}{N} \sum \varphi y)$$

- The residual $\varepsilon(t, \theta) = y(t) - \theta^T \varphi(t)$ can be interpreted as a *prediction error*.
- The LS estimate is strongly consistent under mild conditions if the noise is white.

Prediction error methods

The least-squares method can be generalized in the following way:

1. Compute the model prediction error

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|t-1;\theta), \quad t = 1,\ldots,N$$

2. Compute the model fit (the *cost*)

$$V_N(\theta) = \frac{1}{N}\sum l(t,\theta,\varepsilon(t,\theta)),$$

   where l is a scalar, positive function.

3. Pick the best model

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta)$$

This is the so called *prediction error method (PEM)*, which can be applied to both black-box and white-box models, be they linear or non-linear.

A common choice of cost function is $l(t,\theta,\varepsilon(t,\theta)) = \varepsilon^2(t,\theta)$.

How to calculate predictions?

Consider the model

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t)$$

where

$$G(q,\theta) = \sum_{k=1}^{\infty} g(k,\theta)q^{-k}, \quad H(q,\theta) = 1 + \sum_{k=1}^{\infty} h(k,\theta)q^{-k}$$

Then (omitting the argument θ)

$$\begin{aligned} y(t) &= G(q)u(t) + (H(q)-1)e(t) + e(t) \\ &= G(q)u(t) + \big(H(q)-1\big)H(q)^{-1}\big(y(t) - G(q)u(t)\big) + e(t) \\ &= H(q)^{-1}G(q)u(t) + (1 - H(q)^{-1})y(t) + e(t) \end{aligned}$$

Since $e(\cdot)$ is assumed to be white noise, the optimal mean-square predictor is

$$\hat{y}(t|t-1,\theta) = H^{-1}(q,\theta)G(q,\theta)u(t) + (1 - H^{-1}(q,\theta))y(t)$$

and the optimal prediction error is $\varepsilon(t,\theta) = e(t)$.

How to calculate predictions – summary

Model:

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) = \frac{B(q,\theta)}{F(q,\theta)}u(t) + \frac{C(q,\theta)}{D(q,\theta)}e(t)$$

Predictor:

$$\begin{aligned}\hat{y}(t|t-1,\theta) &= H^{-1}(q,\theta)G(q,\theta)u(t) + (1 - H^{-1}(q,\theta))y(t) \\ &= \frac{D(q)}{C(q)} \cdot \frac{B(q)}{F(q)}u(t) + \frac{C(q)-D(q)}{C(q)}y(t)\end{aligned}$$

Special model structures

$$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) = \frac{B(q,\theta)}{F(q,\theta)}u(t) + \frac{C(q,\theta)}{D(q,\theta)}e(t)$$

$$\hat{y}(t|t-1,\theta) = \frac{D(q)}{C(q)} \cdot \frac{B(q)}{F(q)}u(t) + \frac{C(q)-D(q)}{C(q)}y(t)$$

FIR (Finite Impulse Response):

$$y(t) = B(q,\theta)u(t) + e(t) \qquad \hat{y}(t|t-1,\theta) = B(q,\theta)u(t)$$

ARX (Auto-Regressive with eXogenous input):

$$y(t) = \frac{B(q,\theta)}{A(q,\theta)}u(t) + \frac{1}{A(q,\theta)}e(t) \quad \text{or} \quad A(q,\theta)y(t) = B(q,\theta)u(t) + e(t)$$

$$\hat{y}(t|t-1,\theta) = B(q,\theta)u(t) + (1 - A(q,\theta))y(t)$$

Special model structures, cont'd

ARMAX (Auto-Regressive, Moving Average with eXogenous input):

$$y(t) = \frac{B(q,\theta)}{A(q,\theta)}u(t) + \frac{C(q,\theta)}{A(q,\theta)}e(t)$$

$$\hat{y}(t|t-1,\theta) = \frac{B(q,\theta)}{C(q,\theta)}u(t) + \frac{C(q,\theta) - A(q,\theta)}{C(q,\theta)}y(t)$$

OE (Output Error):

$$y(t) = \frac{B(q,\theta)}{F(q,\theta)}u(t) + e(t) \qquad \hat{y}(t|t-1,\theta) = \frac{B(q,\theta)}{F(q,\theta)}u(t)$$

Special model structures

OE (Output Error):

$$\hat{y}(t|t-1,\boldsymbol{\theta}) = (1 - F(q,\boldsymbol{\theta}))\hat{y}(t|t-1,\boldsymbol{\theta}) + B(q,\boldsymbol{\theta})u(t) \tag{6}$$

The first term in this expression contains delayed values of the *prediction*. We can still gather delayed signals, u and $\hat{y}$, in a vector φ to get an expression for the prediction like

$$\hat{y}(t|t-1,\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \varphi(t,\boldsymbol{\theta}), \tag{7}$$

φ *now depends on* $\boldsymbol{\theta}$ via $\hat{y}$. The conclusion is that the prediction is *nonlinear in* $\boldsymbol{\theta}$.

Special model structures

ARMAX:

$$\hat{y}(t|t-1,\theta) = B(q,\theta)u(t) + (1-A(q,\theta))y(t) + (C(q,\theta)-1)(y(t)-\hat{y}(t|t-1,\theta)), \tag{8}$$

The last expression contains previous values of the *prediction error* $\varepsilon(t,\theta)$.

Linear black-box model structures

Model: $$y(t) = G(q,\theta)u(t) + H(q,\theta)e(t) = \frac{B(q,\theta)}{F(q,\theta)}u(t) + \frac{C(q,\theta)}{D(q,\theta)}e(t)$$

Predictor: $$\hat{y}(t|t-1,\theta) = \frac{D(q)}{C(q)} \cdot \frac{B(q)}{F(q)}u(t) + \frac{C(q)-D(q)}{C(q)}y(t)$$

FIR (Finite Impulse Response):

$$y(t) = B(q,\theta)u(t) + e(t)$$

ARX (Auto-Regressive with eXogenous input):

$$A(q,\theta)y(t) = B(q,\theta)u(t) + e(t)$$

ARMAX (Auto-Regressive, Moving Average with eXogenous input):

$$A(q,\theta)y(t) = B(q,\theta)u(t) + C(q,\theta)e(t)$$

OE (Output Error):

$$y(t) = \frac{B(q,\theta)}{F(q,\theta)}u(t) + e(t)$$

Computing the estimate

Assume that we apply a PEM with quadratic cost function

$$V_N(\theta) = \frac{1}{N}\sum_{t=1}^{N} \varepsilon^2(t,\theta) = \frac{1}{N}\sum_{t=1}^{N} \big(y(t) - \hat{y}(t|t-1,\theta)\big)^2$$

Then the following important observations can be made:

- For the ARX (and the special case FIR) model, the predictor $\hat{y}(t|t-1,\theta) = \theta^T\varphi(t)$ is linear in θ. The implication is that the estimate – the minimizer of $V_N(\theta)$ – can be computed as the solution of a *linear* system of equations.
- For the other model structures, the predictor is *nonlinear* in θ, so that the minimizer of $V_N(\theta)$ has to be found by an iterative search.

Recap on prediction error methods (PEM)

The PEM "recipe":

1. Compute the model prediction error

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t|t-1;\theta), \quad t = 1,\dots,N$$

2. Compute the model fit (the *cost*)

$$V_N(\theta) = \frac{1}{N}\sum l(t,\theta,\varepsilon(t,\theta)),$$

   where l is a scalar, positive function.

3. Pick the best model

$$\hat{\theta}_N = \arg\min_{\theta} V_N(\theta)$$

- The PEM can be applied to both black-box and white-box models, be they linear or non-linear.
- A common choice of cost function is $l(t,\theta,\varepsilon(t,\theta)) = \varepsilon^2(t,\theta)$ (least-squares, ML with Gaussian noise).

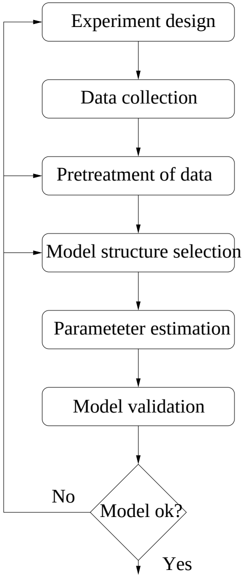
System identification in practice

- System identification workflow
- Experiment design
- Pretreatment of data
- Model structure selection
- Parameter estimation
- Model validation

Learning objectives:

- Use methods and tools to develop mathematical models of dynamical systems from measurement data.

System identification workflow



Design of experimental conditions

Several factors influence the quality of the data obtained, e.g.:

- Choice of operating point (nonlinearities?)
- Choice of sampling interval
  - focus on frequency range of interest
  - avoid modeling irrelevant disturbances
  - be aware of aliasing — prefilter!
  - fast sampling may give practical problems
  - a rule-of-thumb: 6-10 samples per settling time of a step response
- Choice of input signal
  - spectral properties: think of intended use of model
  - amplitude: accuracy vs nonlinearities
  - if the input is generated by feedback, special care needs to be taken

Pretreatment of data

Data often need to be prepared for system identification:

- Looking at data is always a good advice!
- Remove non-zero means and trends in data by e.g.:
  - fitting a polynomial to data and then subtract it, or
  - using differentiated data
- Remove high-frequency disturbances by low-pass filtering
- Filter data to focus on particular frequency regions (bias distribution in the frequency domain)
- Remove *outliers*

Model structure selection

The selection of model structures include e.g.:

- Choice between white box or black box
- Choice of parametrization: ARX, FIR, OE, ARMAX, ...
- Model order selection
- Determination of time delays

A good rule: try simple things first! And be prepared to revise initial choices!

Choice of identification method

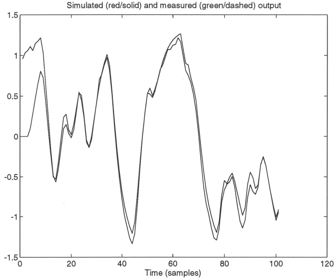
The choice of identification method is influenced by e.g.:

- Experimental conditions, e.g. on-line vs off-line
- Available input signals
- Intended use of the model
- Accuracy requirements
- Robustness requirements

Model validation: testing model quality

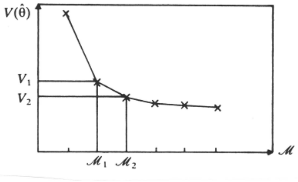
There are many alternatives to test model quality, e.g.:

- evaluate the loss function (part of PEM!)
- simulate the system, i.e. compare the real output $y(t)$ with the (noise-free) model output $y_m(t) = G(q, \theta)u(t)$
- investigate frequency response, poles, zeros, ...
- analyze prediction errors (residuals)
- try the model on fresh data (cross validation)

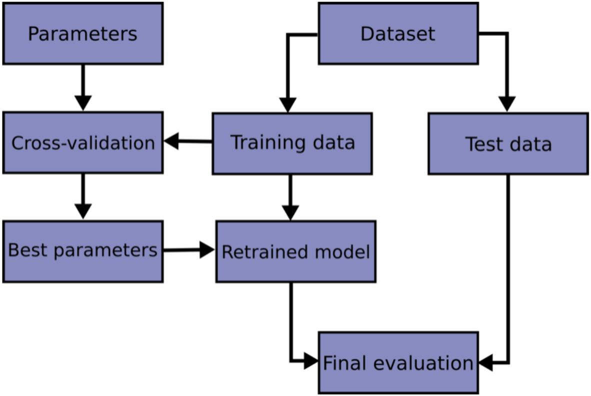


Model order selection

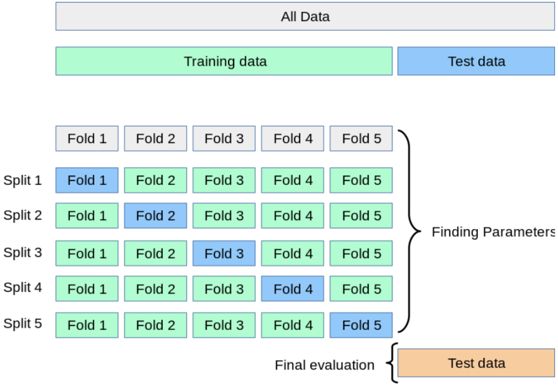
Increased model flexibility will always improve the fit, but the model will not be "stable" w.r.t. new data ("overfit")!



Validation


Parameters
Dataset
Cross-validation
Training data
Test data
Best parameters
Retrained model
Final evaluation

Cross-validation



System identification in Matlab – a tutorial

The following steps summarize a tutorial available in the System identification toolbox:

- Load experimental data: `load dyer2`
- Open app: `systemIdentification`
- Import time domain data
- Time plot
- Preprocess - remove means (detrend)
- Select detrended data as working data
- Select estimation data (1-500) -> working data
- Select validation data (501-1000) -> validation data
- Trash data not used
- Estimate a range of ARX models, select two models
- Estimate ARMAX models [2222] and [3322]
- Inspect and compare models: model outputs and residuals, parameter uncertainties