EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION

Katalin Hangos

Anna Pózna

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Contents

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Katalin Hangos:
  - Parameter estimation – 1: Basic notions, Elements of random variables and mathematical statistics
  - Parameter estimation – 2: The properties of the estimates, Linear regression
  - Parameter estimation – 3: Stochastic processes, Discrete time stochastic dynamic models
  - Parameter estimation – 4: Least Squares (LS) estimation and its properties in the dynamic case
  - Parameter estimation – 5: Practical implementation of parameter estimation, Evaluation of the results
  - Parameter estimation – 6: Special methods for dynamic LS estimation: Instrumental variable method, Nonlinear dynamic case
- Anna Pózna:
  - Parameter estimation: Computer laboratory 1
  - Parameter estimation: Computer laboratory 2
  - Parameter estimation: Computer laboratory 3
  - Parameter estimation: Computer laboratory 4
  - Matlab codes

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 1

Basic notions
Elements of random variables and mathematical statistics

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Lecture overview

- What does ”parameter estimation” mean?
  - Model, variables and parameters
  - Why estimation?
  - Course content and requirements
- Random variables
  - Scalar-valued random variables
  - Vector-valued random variables
- Elements of mathematical statistics
  - Sample and statistics
  - Constructing an estimate from a measured data set
  - Estimation of the mean value and the covariances
  - Estimating the probability density function - histogram
- Tutorial

Model, variables and parameters

Relationships between data are described by a so called model

$$y = \mathcal{M}(x, p)$$

where

- the vector x is the *measurable independent variable* that we can manipulate/set in an error-free way
- the vector y is the *measurable dependent variable* (*subject to measurement errors*)
- the vector p of *constant parameters*

Model, variables and parameters

Relationships between data are described by a so called model

$$y = \mathcal{M}(x, p)$$

where

- the vector x is the *measurable independent variable* that we can manipulate/set in an error-free way
- the vector y is the *measurable dependent variable* (*subject to measurement errors*)
- the vector p of *constant parameters*

Important

The aim of parameter estimation is to estimate the unknown parameters p from measured sets of dependent and independent variable values $(y_i, x_i),\ i = 1, ..., n$ and given model form $\mathcal{M}$.

Model types

$$y = \mathcal{M}(x, p)$$

- **linear in parameters**

$$\mathcal{M}(x, p) = p^T \mathcal{F}(x)$$

  where $\mathcal{F}(x)$ is a possibly nonlinear function of the independent variable vector x

- **dynamic**
  discrete time index $k = 0, 1, ..., K, ...$ such that

$$y(k) = \mathcal{M}(x(k), x(k-1), ..., x(k-K); p) \ , \ \ k = K, K+1, ..., n$$

Errors and estimates

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The measurable dependent variable vector y is subject to *measurement errors*, and the model is also often not precise (*modelling errors* are also present):

$$y = \mathcal{M}(x, p) + \varepsilon \quad \left(y^{(M)} = \mathcal{M}(x, p)\right)$$

where ε, and thus also y are (vector valued) *random variables*.

The measurable dependent variable vector y is subject to *measurement errors*, and the model is also often not precise (*modelling errors* are also present):

$$y = \mathcal{M}(x, p) + \varepsilon \quad \left(y^{(M)} = \mathcal{M}(x, p)\right)$$

where ε, and thus also y are (vector valued) *random variables*.

Important

The result of a parameter estimation procedure can only be an "estimate" of the true model parameter vector p (denoted by $\hat{p}$), such that $\hat{p}$ is a vector valued random variable in itself.

Course structure

The course is given weakly in the form of a

- Lecture and tutorial, or a
- Laboratory with the use of MATLAB

Course structure

The course is given weakly in the form of a

- Lecture and tutorial, or a
- Laboratory with the use of MATLAB

Important (Course web page)

http://virt.uni-pannon.hu/index.php/hu/oktatas/tantargyak/160-pa

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Evaluation

The pre-requisite of the course signature is

- to be present at least 75% of the lectures-tutorials-laboratories,
- to submit the project results and documentation to the given deadline, and
- to achieve at least 50% on the closed-book exam (the results of the homework are added to the points of the exam).

The pre-requisite of the course signature is

- to be present at least 75% of the lectures-tutorials-laboratories,
- to submit the project results and documentation to the given deadline, and
- to achieve at least 50% on the closed-book exam (the results of the homework are added to the points of the exam).

Important

The evaluation is based upon a mid-semester closed-book exam and on a parameter estimation project work to be implemented in MATLAB.

Overview

- What does "parameter estimation" mean?
- Random variables
  - Scalar-valued random variables
  - Vector-valued random variables
- Elements of mathematical statistics
- Tutorial

Scalar-valued random variables

A scalar-valued random variable ξ is characterized by its *probability density function (p.d.f.)* $f_\xi : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$.

Properties
The *mean value* and *variance* of the random variable ξ with its p.d.f. f_ξ are

$$E\{\xi\} = \int x f_\xi(x) dx \ , \ \ \sigma^2\{\xi\} = \int (x - E\{\xi\})^2 f_\xi(x) dx$$

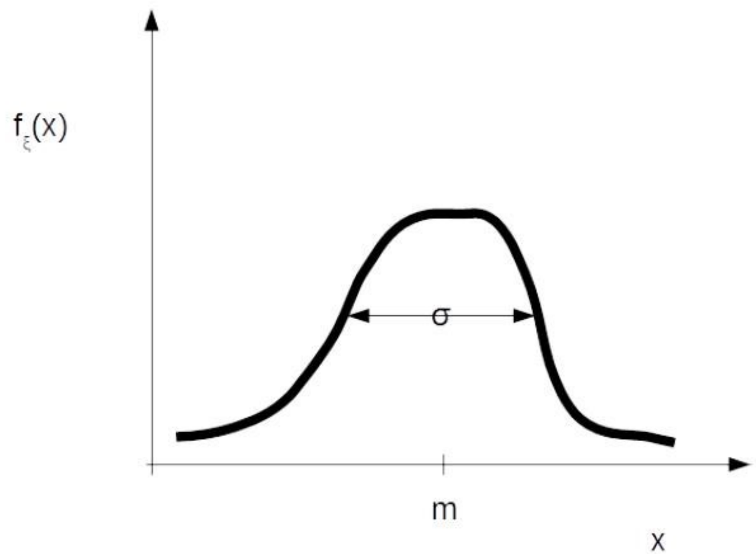
Normally distributed scalar-valued random variables

The random variable ξ has a *normal or Gaussian distribution*, in notation

$$\xi \sim \mathbb{N}(m, \sigma^2) \tag{1}$$

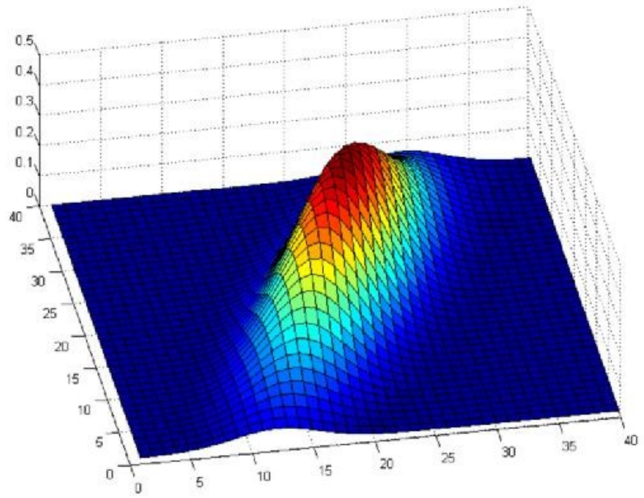
where m is its *mean value* and σ^2 is its *variance*, when

$$f_\xi(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{(x-m)^2}{\sigma^2}\right)}$$



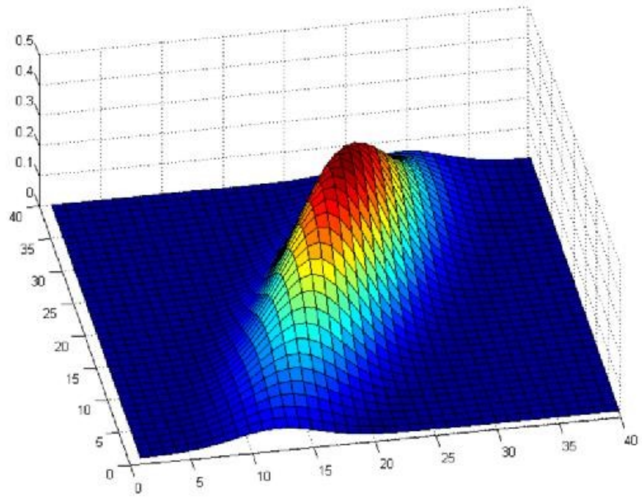
Independence of two scalar-valued random variables

The joint distribution of two scalar-valued random variables ξ and θ is characterized by their *joint p.d.f* $f_{\xi,\theta}(x, y) : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$.



Independence of two scalar-valued random variables

The joint distribution of two scalar-valued random variables ξ and θ is characterized by their *joint p.d.f* $f_{\xi,\theta}(x, y) : \mathbb{R} \times \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$.



Important

Two scalar-valued random variables ξ and θ are called independent, if $f_{\xi,\theta}(x, y) = f_\xi(x) \cdot f_\theta(y)$.

Covariance and correlation of random variables

The *covariance* of two scalar-valued random variables ξ and θ is

$$COV\{\xi,\theta\} = E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}$$

where $\overline{\xi} = (\xi - E\{\xi\})$ is a *centered random variable*.

Correlation (normed covariance): $\rho\{\xi,\theta\} = \frac{E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}}{\sigma\{\xi\}\sigma\{\theta\}}$

Covariance and correlation of random variables

The *covariance* of two scalar-valued random variables ξ and θ is

$$COV\{\xi, \theta\} = E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}$$

where $\overline{\xi} = (\xi - E\{\xi\})$ is a *centered random variable*.

Correlation (normed covariance): $\rho\{\xi, \theta\} = \frac{E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}}{\sigma\{\xi\}\sigma\{\theta\}}$

Important

Independence implies $\rho\{\xi, \theta\} = 0$, but $\rho\{\xi, \theta\} = 0$ implies independence only in case of Gaussian joint distribution.

Covariance and correlation of random variables

The *covariance* of two scalar-valued random variables ξ and θ is

$$COV\{\xi,\theta\} = E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}$$

where $\overline{\xi} = (\xi - E\{\xi\})$ is a *centered random variable*.

Correlation (normed covariance): $\rho\{\xi,\theta\} = \frac{E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}}{\sigma\{\xi\}\sigma\{\theta\}}$

Important

Independence implies $\rho\{\xi,\theta\} = 0$, but $\rho\{\xi,\theta\} = 0$ implies independence only in case of Gaussian joint distribution.

Important

The covariance of a scalar-valued random variables ξ with itself is its variance , i.e. $COV\{\xi,\xi\} = \sigma^2\{\xi\}$

Vector valued random variables – 1

Given a vector valued random variable ξ

$$\xi \ : \ \ \xi(\omega), \quad \omega \in \Omega, \quad \xi(\omega) \in \mathbb{R}^{\mu}$$

Scalar valued entries of vector valued random variables

$$\xi = \begin{bmatrix} \xi_1 \\ ... \\ \xi_{\mu} \end{bmatrix}$$

where each entry ξ_i is a scalar valued random variable

Vector valued random variables – 2

Given a vector valued random variable $\xi \in \mathbb{R}^\mu$

- Its ***mean value*** *$m \in R^\mu$ is a real vector*.

Vector valued random variables – 2

Given a vector valued random variable $\xi \in \mathbb{R}^{\mu}$

- Its ***mean value*** $m \in R^{\mu}$ *is a real vector*.
- Its ***variance*** $COV\{\xi\}$ is a square real matrix, the *covariance matrix*:

$$COV\{\xi\} = E\{(\xi - E\{\xi\})(\xi - E\{\xi\})^T\}$$

Vector valued random variables – 2

Given a vector valued random variable $\xi \in \mathbb{R}^{\mu}$

- Its ***mean value*** $m \in R^{\mu}$ *is a real vector*.
- Its ***variance*** $COV\{\xi\}$ is a square real matrix, the *covariance matrix*:

$$COV\{\xi\} = E\{(\xi - E\{\xi\})(\xi - E\{\xi\})^T\}$$

Important

Covariance matrices are positive definite symmetric matrices:

$$z^T COV\{\xi\} z \geq 0 \quad , \quad \forall z \in \mathbb{R}^{\mu}$$

Covariance matrix and covariances

Consider a two dimensional vector valued random variable

$$\xi = \begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix}$$

with $E\{\xi_i\} = m_i$, and its centered version $\overline{\xi}_i = \xi_i - m_i$

Covariance matrix: $COV\{\xi\} = E\{(\xi - E\{\xi\})(\xi - E\{\xi\})^T\}$

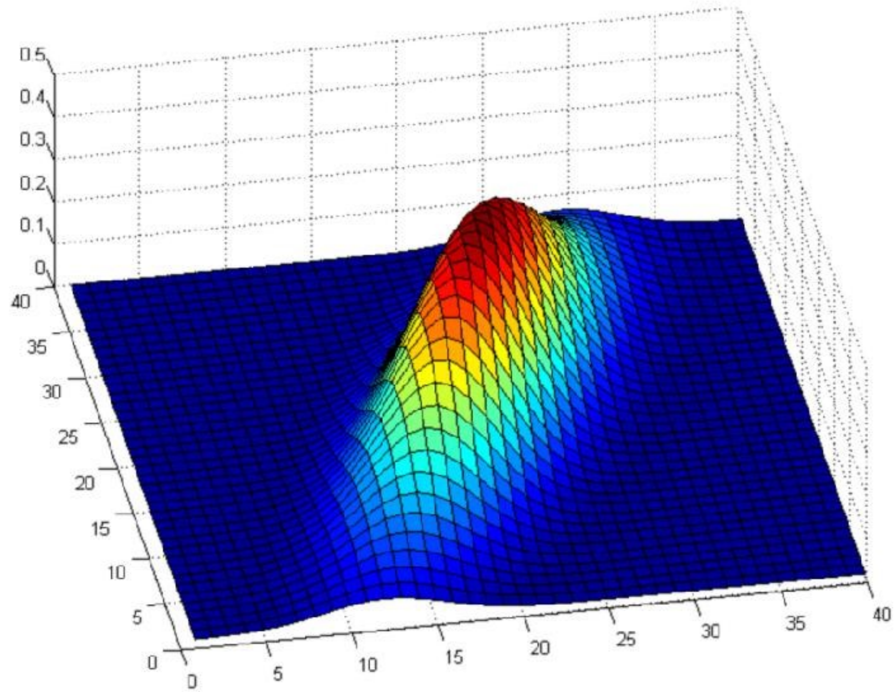
$$E\{\overline{\xi\xi}^T\} = E\left\{\begin{bmatrix} \overline{\xi}_1^2 & \overline{\xi}_1\overline{\xi}_2 \\ \overline{\xi}_1\overline{\xi}_2 & \overline{\xi}_2^2 \end{bmatrix}\right\} = \begin{bmatrix} \sigma^2\{\xi_1\} & COV\{\xi_1, \xi_2\} \\ COV\{\xi_1, \xi_2\} & \sigma^2\{\xi_1\} \end{bmatrix}$$

diagonal: variances, off-diagonal: covariances

Two dimensional Gaussian distribution

Probability density function:

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-r^2}} e^{-\frac{1}{2(1-r^2)}\left(\frac{(x_1-m_1)^2}{\sigma_1^2} - 2r\frac{(x_1-m_1)(x_2-m_2)}{\sigma_1\sigma_2} + \frac{(x_2-m_2)^2}{\sigma_2^2}\right)}$$



Two dimensional Gaussian distribution - 1

Probability density function:

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-r^2}} e^{-\frac{1}{2(1-r^2)}\left(\frac{(x_1-m_1)^2}{\sigma_1^2} - 2r\frac{(x_1-m_1)(x_2-m_2)}{\sigma_1\sigma_2} + \frac{(x_2-m_2)^2}{\sigma_2^2}\right)}$$

Assume non-correlated elements ξ_1 and ξ_2 with $r = 0$. Then

$$f(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2}\left(\frac{(x_1-m_1)^2}{\sigma_1^2} + \frac{(x_2-m_2)^2}{\sigma_2^2}\right)} = f_1(x_1) \cdot f_2(x_2)$$

with $f_i(x_i) = \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{(x_i-m_i)^2}{\sigma_i^2}\right)}$

Therefore ξ_1 and ξ_2 are independent.

Linearly transformed random variables

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us transform the vector-valued random variable $\xi(\omega) \in R^n$ using the non-singular square transformation matrix $T \in R^{n \times n}$:

$$\eta = T\xi$$

The properties of the vector-valued random variable η:

$$E\{\eta\} = TE\{\xi\} \quad , \quad COV\{\eta\} = TCOV\{\xi\}T^T$$

Linearly transformed random variables

Let us transform the vector-valued random variable $\xi(\omega) \in R^n$ using the non-singular square transformation matrix $T \in R^{n \times n}$:

$$\eta = T\xi$$

The properties of the vector-valued random variable η:

$$E\{\eta\} = TE\{\xi\} \quad , \quad COV\{\eta\} = TCOV\{\xi\}T^T$$

Important (Gaussian case)

If the random variable ξ has a Gaussian distribution $N(m_\xi, \Delta_\xi)$ with mean value m_ξ and covariance matrix Δ_ξ, then the transformed random variable η will also be Gaussian $N(m_\eta, \Delta_\eta)$, where

$$m_\eta = Tm_\xi \quad , \quad \Delta_\eta = T\Delta_\xi T^T$$

Overview

- What does "parameter estimation" mean?
- Random variables
- Elements of mathematical statistics
  - Sample and statistics
  - Constructing an estimate from a measured data set
  - Estimation of the mean value and the covariances
  - Estimating the probability density function - histogram
- Tutorial

Consider a (scalar valued) random variable ξ *with probability density function* $f_\xi(x)$.

- Sample

  is a collection (set) of *n* independent random variables

$$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$$

  where every ξ_i has the same distribution as ξ.

  — the sample corresponds to a set of *measurements* about ξ

Sample, statistics

Consider a (scalar valued) random variable ξ with probability density function $f_\xi(x)$.

- Sample
  is a collection (set) of n independent random variables

  $$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$$

  where every ξ_i has the same distribution as ξ.
  — the sample corresponds to a set of *measurements* about ξ
- Statistics
  is a (deterministic) function of the sample elements (a random variable itself)

  $$s(S) = F(\xi_1, \xi_2, ..., \xi_n)$$

  — a statistics is used to construct an *estimate*

Consider a (scalar valued) random variable ξ *with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Measured data set

is a collection (set) of n measurements of the sample elements $\{\xi_1, \xi_2, ..., \xi_n\}$

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

D is a realization of S.

- the measured data set contains an *actual set of measurements* about ξ that are **not** random variables but deterministic values (a realization).

Estimates

Consider a (scalar valued) random variable ξ *with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$, and with a *measured data set*

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

Estimate
is a realization of a statistics $s(S) = F(\xi_1, \xi_2, ..., \xi_n)$

$$\hat{s}(D) = F(x_1, x_2, ..., x_n)$$

An estimate is computed from the *actual measurement values in the data set D*

Estimates

Consider a (scalar valued) random variable ξ *with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$, and with a *measured data set*

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

Estimate
is a realization of a statistics $s(S) = F(\xi_1, \xi_2, ..., \xi_n)$

$$\hat{s}(D) = F(x_1, x_2, ..., x_n)$$

An estimate is computed from the *actual measurement values in the data set D*

Important

Unbiased estimate
if the mean value of the statistics is the real value of the parameter to be estimated

Estimation of the mean value – 1

Assume that the underlying scalar-valued random variable ξ *has a mean value m and the variance* σ^2

- Statistics for the mean value: *sample mean*

$$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

Property: $E[\mu] = m \quad ===> \quad$ unbiased

Estimation of the mean value – 1

Assume that the underlying scalar-valued random variable ξ *has a mean value m and the variance* σ^2

- Statistics for the mean value: *sample mean*

$$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

Property: $E[\mu] = m \quad ===> \quad$ unbiased

- Estimate of the mean value

$$\hat{m}(D) = \frac{1}{n}(x_1 + x_2 + ... + x_n)$$

Estimation of the variance

Assume that the underlying scalar valued random variable ξ *has a mean value m and the variance* σ^2

- Statistics for the variance: *corrected empirical variance*

$$\theta(S) = \frac{1}{n-1}\left((\xi_1 - \mu)^2 + (\xi_2 - \mu)^2 + ... + (\xi_n - \mu)^2\right)$$

with $\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$
Property: $E[\theta] = \sigma^2 \quad ===> \quad$ unbiased

Estimation of the variance

Assume that the underlying scalar valued random variable ξ *has a mean value m and the variance* σ^2

- Statistics for the variance: *corrected empirical variance*

$$\theta(S) = \frac{1}{n-1}\left((\xi_1 - \mu)^2 + (\xi_2 - \mu)^2 + ... + (\xi_n - \mu)^2\right)$$

  with $\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$
  Property: $E[\theta] = \sigma^2 \quad ===> \quad$ unbiased

- Estimate of the variance

$$\hat{\sigma}^2(D) = \frac{1}{n-1}\left((x_1 - \hat{m}(D))^2 + ... + (x_n - \hat{m}(D))^2\right)$$

  with $\hat{m}(D) = \frac{1}{n}(x_1 + x_2 + ... + x_n)$

Estimation of the mean value – 2

Assume that the underlying μ-dimensional vector valued random variable ξ has a mean value $m \in \mathbb{R}^\mu$. The sample is a collection of independent vector valued random variables

$$S(\xi) = \{\xi_1, ..., \xi_n\}$$

where $\xi_i = [\xi_{i,1},\ ..., \xi_{i,\nu}]^T$ and the independence is considered entry-wise.

- Statistics for the mean value: *sample mean*

  $$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

  Property: $E[\mu] = m \quad ===> \quad$ unbiased

Estimation of the mean value – 2

Assume that the underlying μ-dimensional vector valued random variable ξ has a mean value $m \in \mathbb{R}^{\mu}$. The sample is a collection of independent vector valued random variables

$$S(\xi) = \{\xi_1, ..., \xi_n\}$$

where $\xi_i = [\xi_{i,1}, \ ..., \xi_{i,\nu}]^T$ and the independence is considered entry-wise.

- Statistics for the mean value: *sample mean*

$$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

  Property: $E[\mu] = m \quad ===> \quad$ unbiased

- Estimate of the mean value

$$\hat{m}(D) = \frac{1}{n}(x_1 + x_2 + ... + x_n)$$

  is computed entry-wise.

Estimation of the covariances

Assume that the underlying μ-dimensional vector valued random variable ξ has a mean value $m \in \mathbb{R}^{\mu}$. The sample is $S(\xi) = \{\xi_1, ..., \xi_n\}$ and $\overline{\xi}_i = \xi_i - m_i$ is the centered version of the sample element ξ_i.

- Statistics for the covariances of the entries (i, j)

$$\rho_{ij}(S) = \frac{1}{n-1} \sum_{k=1}^{n} (\overline{\xi}_{k,i} \cdot \overline{\xi}_{k,j})$$

Estimation of the covariances

Assume that the underlying μ-dimensional vector valued random variable ξ has a mean value $m \in \mathbb{R}^{\mu}$. The sample is $S(\xi) = \{\xi_1, ..., \xi_n\}$ and $\overline{\xi}_i = \xi_i - m_i$ is the centered version of the sample element ξ_i.

- Statistics for the covariances of the entries (i, j)

$$\rho_{ij}(S) = \frac{1}{n-1} \sum_{k=1}^{n} (\overline{\xi}_{k,i} \cdot \overline{\xi}_{k,j})$$

- Estimate of the covariance $r_{ij} = COV\{\xi_{\cdot,i}, \xi_{\cdot,j}\}$

$$\hat{r}_{ij}(D) = \frac{1}{n-1} \sum_{k=1}^{n} (x_{k,i} - \hat{m}_i) \cdot (x_{k,j} - \hat{m}_j)$$

with $\hat{m}_i = \frac{1}{n-1} \sum_{k=1}^{n} x_{k,i}$

Estimation of the auto-covariances – 1

*Consider (scalar valued) random variables ξ_i from the same distribution but **not independent**.*
They form a "generalized" sample: $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Estimation of the mean value m using the sample mean as statistics

- Estimate

$$\hat{m}(D) = \frac{1}{n}(x_1 + ... + x_n)$$

- This is an unbiased estimate

Estimation of the auto-covariances – 2

Consider (scalar valued) random variables ξ_i *from the same distribution and with a* ***pairwise constant covariance*** $r = COV\{\xi_i, \xi_{i+1}\}$ and with a "generalized" sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

The estimate of the mean value m is

$$\hat{m}(D) = \frac{1}{n}(x_1 + ... + x_n)$$

The estimate of the autocovariance r is

$$\hat{r}(D) = \frac{1}{n-1}\sum_{i=1}^{n-1}((x_i - \hat{m})(x_{i+1} - \hat{m}))$$

It may be a biased estimate

Histogram construction

Consider a (scalar valued) random variable ξ with the probability density function $f_\xi(z)$ and a sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

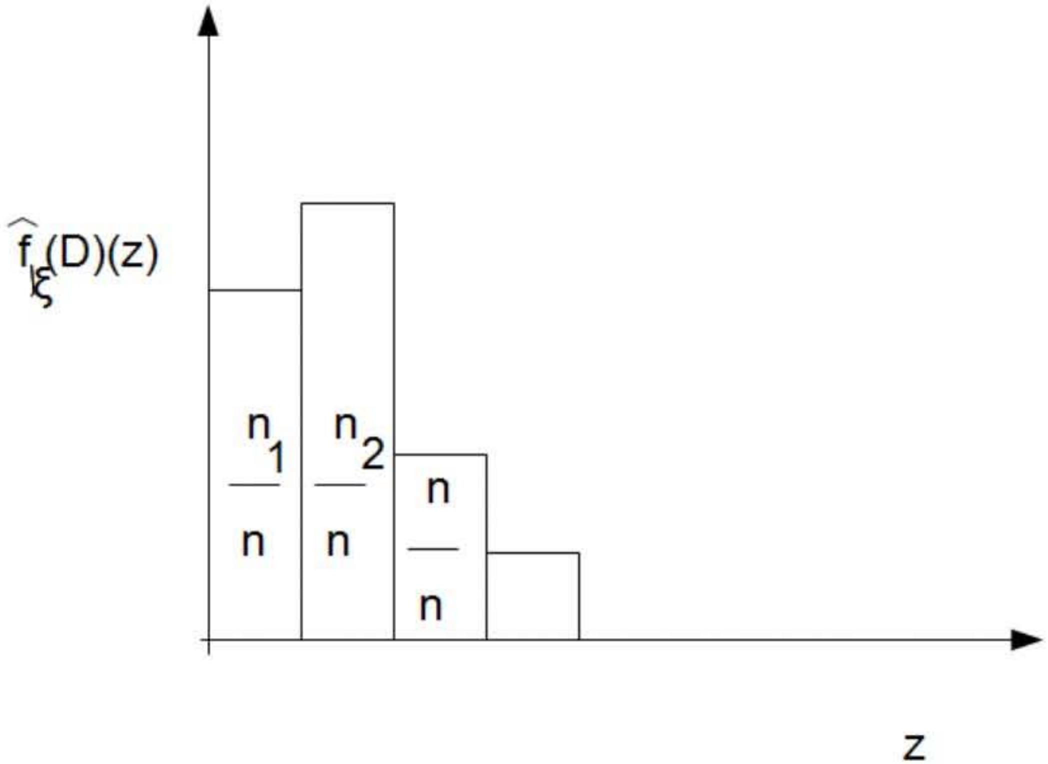
Histogram construction

- Let x_M the maximal and x_m the minimal element of the data set D with n elements.
- Divide the interval $[x_m, x_M]$ into ℓ sub-intervals ($\delta = \frac{x_M - x_m}{\ell}$) such that $z_i = x_m + (i-1)\delta$
- Denote by n_i the number of data set elements in the interval $[z_i, z_{i+1}]$

Estimate of $f_\xi(z)$:
the piece-wise constant function $\hat{f}_\xi(D)(z)$ such that

$$\hat{f}_\xi(D)(z) = \frac{n_i}{n} \text{ for } z \in [z_i, z_{i+1}] \ , \ i = 1, ..., \ell$$

A simple histogram



Tutorial problems

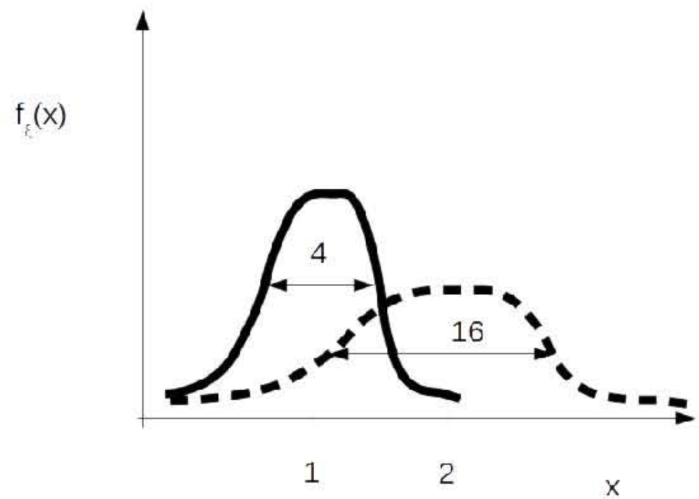
- A. Vector valued random variables
- B. Mean value and covariance estimation

Tutorial problems – A

Example (Vector valued random variables – 1)

Given two scalar-valued Gaussian random variables $\eta_1 \sim \mathbb{N}(1,4)$, *and* $\eta_2 \sim \mathbb{N}(2,16)$.

- *Plot the probability density functions* f_{η_1} *and* f_{η_2} *of random variables* η_1 *and* η_2 *in the same coordinate system!*



Example (Vector valued random variables – 2)

Given two scalar-valued Gaussian random variables $\eta_1 \sim \mathbb{N}(1,4)$, and $\eta_2 \sim \mathbb{N}(2,16)$.
Assume that the random variables η_1 and η_2 are independent and form a vector valued random variable $\eta = [\eta_1, \eta_2]^T$ from them.

- *Which type of distribution does the vector valued random variable η have?*
  vector valued Gaussian (2 dimensional)

Tutorial problems – A

Example (Vector valued random variables – 2)

Given two scalar-valued Gaussian random variables $\eta_1 \sim \mathbb{N}(1,4)$, *and* $\eta_2 \sim \mathbb{N}(2,16)$.
Assume that the random variables η_1 *and* η_2 *are independent and form a vector valued random variable* $\eta = [\eta_1, \eta_2]^T$ *from them.*

- *Which type of distribution does the vector valued random variable* η *have?*
  vector valued Gaussian (2 dimensional)

- *Compute the mean value and the variance (covariance matrix) of the vector valued random variable* η.

$$m_\eta = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \;,\; COV\{\eta\} = \begin{bmatrix} 4 & 0 \\ 0 & 16 \end{bmatrix}$$

Example (Vector valued random variables – 3)

Given two scalar-valued Gaussian random variables $\eta_1 \sim \mathbb{N}(1, 4)$, *and* $\eta_2 \sim \mathbb{N}(2, 16)$.
Assume that the random variables η_1 *and* η_2 *have a covariance* $COV(\eta_1, \eta_2) = 2.3$ *and form a vector valued random variable* $\eta = [\eta_1, \eta_2]^T$ *from them.*

- *Which type of distribution does the vector valued random variable* η *have?*
  vector valued Gaussian (2 dimensional)

Example (Vector valued random variables – 3)

Given two scalar-valued Gaussian random variables $\eta_1 \sim \mathbb{N}(1, 4)$, *and* $\eta_2 \sim \mathbb{N}(2, 16)$.
Assume that the random variables η_1 *and* η_2 *have a covariance* $COV(\eta_1, \eta_2) = 2.3$ *and form a vector valued random variable* $\eta = [\eta_1, \eta_2]^T$ *from them.*

- *Which type of distribution does the vector valued random variable* η *have?*
  vector valued Gaussian (2 dimensional)

- *Compute the mean value and the variance of* η.

$$m_\eta = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \ , \ COV\{\eta\} = \begin{bmatrix} 4 & 2.3 \\ 2.3 & 16 \end{bmatrix}$$

Example (Mean value and covariance estimation – 1)

Consider a scalar valued random variable ξ with a measured data set

$$D(5) = \{0.5,\ -0.6,\ 0.3,\ -0.2,\ 0.0\}$$

- *Compute an estimate of the mean value of ξ.*
  statistics: sample mean
  $\hat{m} = \frac{0.5-0.6+0.3-0.2+0.0}{5} = 0$

Tutorial problems – B

Example (Mean value and covariance estimation – 1)

Consider a scalar valued random variable ξ with a measured data set

$$D(5) = \{0.5,\ -0.6,\ 0.3,\ -0.2,\ 0.0\}$$

- *Compute an estimate of the mean value of ξ.*
  statistics: sample mean
  $\hat{m} = \frac{0.5-0.6+0.3-0.2+0.0}{5} = 0$
- *Compute an estimate of the variance of ξ.*
  statistics: corrected empirical variance
  $\hat{\sigma}^2 = \frac{0.5^2-0.6^2+0.3^2-0.2^2+0.0^2}{4}$

Tutorial problems – B

Example (Mean value and covariance estimation – 1)

Consider a scalar valued random variable ξ with a measured data set

$$D(5) = \{0.5,\ -0.6,\ 0.3,\ -0.2,\ 0.0\}$$

- *Compute an estimate of the mean value of ξ.*
  statistics: sample mean
  $\hat{m} = \frac{0.5-0.6+0.3-0.2+0.0}{5} = 0$
- *Compute an estimate of the variance of ξ.*
  statistics: corrected empirical variance
  $\hat{\sigma}^2 = \frac{0.5^2-0.6^2+0.3^2-0.2^2+0.0^2}{4}$
- *Could the measured data be independent? Compute an estimate of $r = COV\{\xi_i, \xi_{i+1}\}$.*
  $\hat{r} = \frac{-0.5\cdot0.6-0.6\cdot0.3-0.2\cdot0.3-o.2\cdot0.0}{4} << 0 \quad ==> $ *NOT independent*

HOMEWORK

Consider a scalar valued random variable ξ and with a measured data set

$$D(5) = \{0.1,\ 0.2,\ 0.3,\ 0.4,\ 0.5\}$$

- Compute an estimate of the mean value of ξ.
- Compute an estimate of the variance of ξ.
- Could the measured data be independent? Compute an estimate of $r = COV\{\xi_i, \xi_{i+1}\}$.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION

COMPUTER LABORATORY 1

Created by: Anna Pózna

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

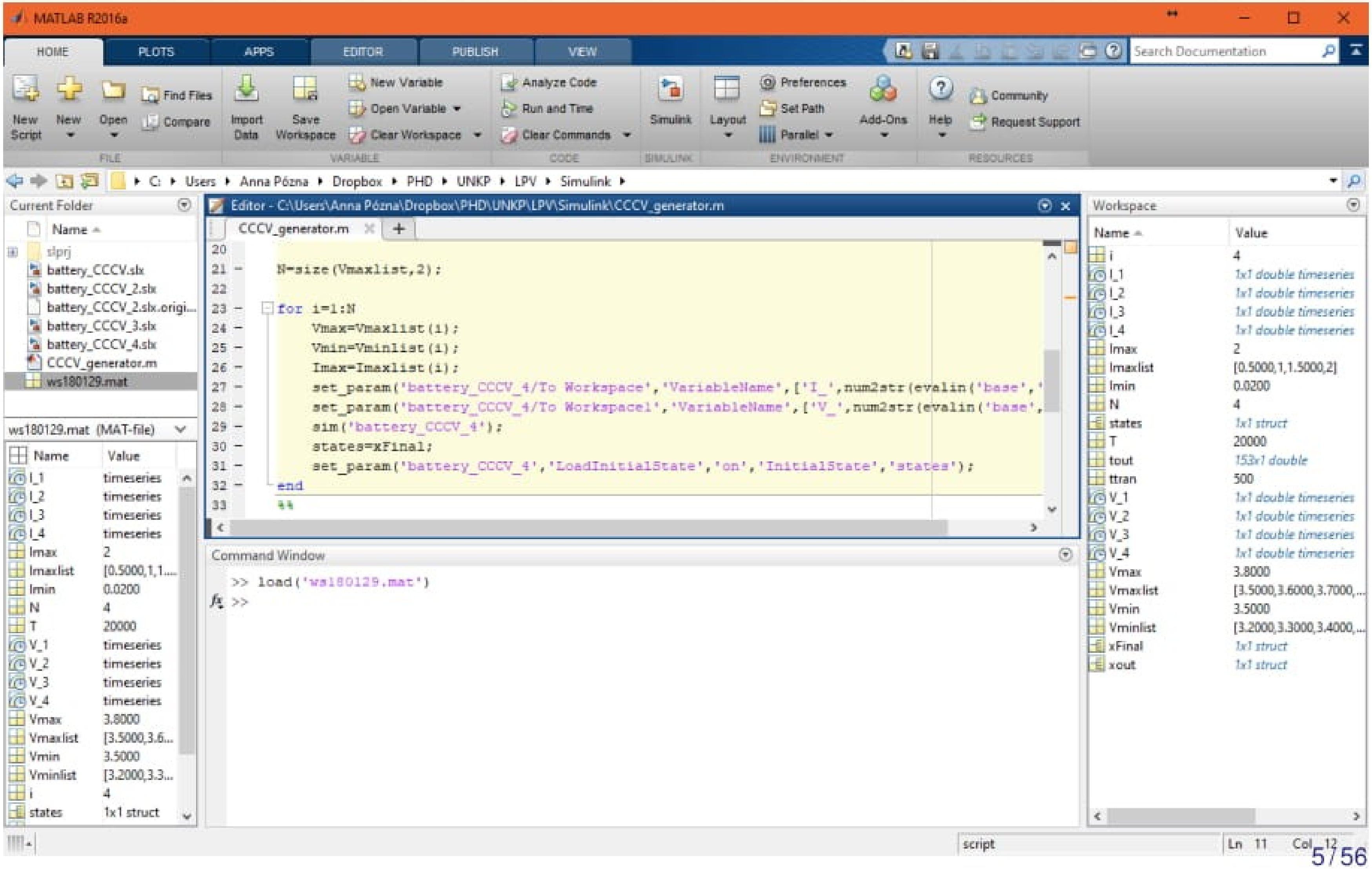
Overview

- Introduction to Matlab
  - User interface
  - Basic commands
  - Vectors
  - Matrices
- Random numbers
  - Uniformly distributed random numbers
  - Normally distributed random numbers
  - Random numbers from any distribution
  - Control random number generation
  - White noise process
- Basic statistics

- Introduction to Matlab
  - User interface
  - Basic commands
  - Vectors
  - Matrices
- Random numbers
- Basic statistics

- MATLAB = MATrix LABoratory
- Numerical computing environment
- Programming language
  - scripting language
  - weakly typed
  - variables are arrays/matrices
  - matrix manipulations

Main window


MATLAB R2016a
HOME
PLOTS
APPS
EDITOR
PUBLISH
VIEW
Search Documentation
New Script
New
Open
Find Files
Compare
Import Data
Save Workspace
New Variable
Open Variable
Clear Workspace
Analyze Code
Run and Time
Clear Commands
Simulink
Layout
Preferences
Set Path
Parallel
Add-Ons
Help
Community
Request Support
C: ▸ Users ▸ Anna Pózna ▸ Dropbox ▸ PHD ▸ UNKP ▸ LPV ▸ Simulink ▸
Current Folder
Editor - C:\Users\Anna Pózna\Dropbox\PHD\UNKP\LPV\Simulink\CCCV_generator.m
CCCV_generator.m
N=size(Vmaxlist,2);
for i=1:N
Vmax=Vmaxlist(i);
Vmin=Vminlist(i);
Imax=Imaxlist(i);
set_param('battery_CCCV_4/To Workspace','VariableName',['I_',num2str(evalin('base','
set_param('battery_CCCV_4/To Workspace1','VariableName',['V_',num2str(evalin('base',
sim('battery_CCCV_4');
states=xFinal;
set_param('battery_CCCV_4','LoadInitialState','on','InitialState','states');
end
%%
Command Window
>> load('ws180129.mat')
Workspace
slprj
battery_CCCV.slx
battery_CCCV_2.slx
battery_CCCV_2.slx.origi...
battery_CCCV_3.slx
battery_CCCV_4.slx
CCCV_generator.m
ws180129.mat
ws180129.mat (MAT-file)
Name
Value
i 4
I_1 1x1 double timeseries
I_2 1x1 double timeseries
I_3 1x1 double timeseries
I_4 1x1 double timeseries
Imax 2
Imaxlist [0.5000,1,1.5000,2]
Imin 0.0200
N 4
states 1x1 struct
T 20000
tout 153x1 double
ttran 500
V_1 1x1 double timeseries
V_2 1x1 double timeseries
V_3 1x1 double timeseries
V_4 1x1 double timeseries
Vmax 3.8000
Vmaxlist [3.5000,3.6000,3.7000,...
Vmin 3.5000
Vminlist [3.2000,3.3000,3.4000,...
xFinal 1x1 struct
xout 1x1 struct
script
Ln 11
Col 12

Command window

- Enter variables or statements
- Execute statements, run commands
- Commands ended without ; → result is displayed in the next line
- Commands ended with ; → result is not displayed
- Recall previous lines with the ↑ ↓ arrow keys
- Clear command window:
  `>> clc`
- Save command history:
  `>> diary`
- Help on commands, functions etc.:
  `>> help function/command name`

Workspace window

- Contains all of the created variables
- Variable name, value, max and min
- Clear all variables from the workspace:

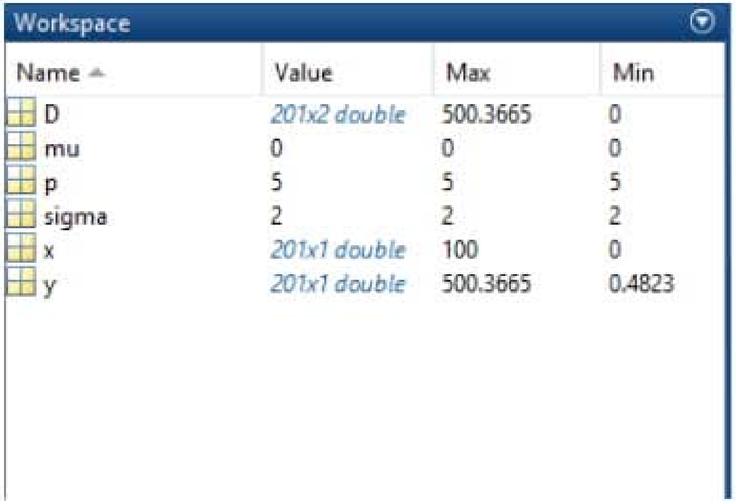
  `>> clear`

- Save workspace variables into file:

  `>> save myworkspace.mat`

- Load/restore a saved workspace:

  `>> load myworkspace.mat`



Editor window

- Writing/editing functions and scripts
- Execute selected commands: F9
- Run the whole script: F5

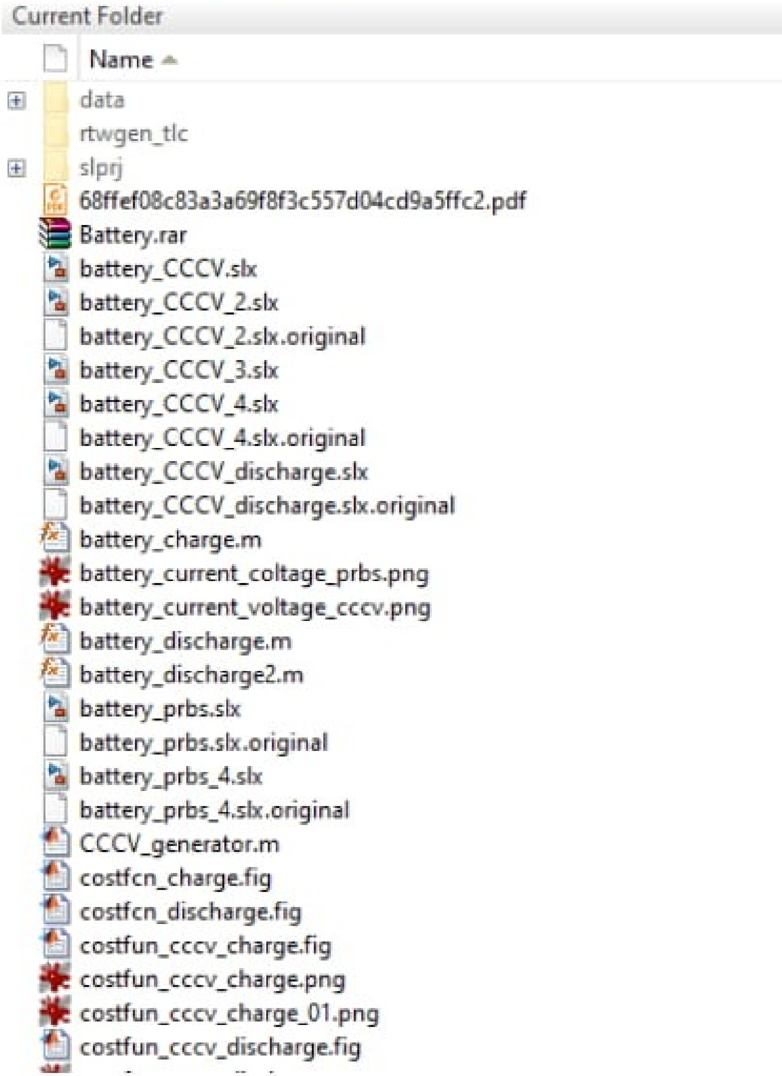
Editor - C:\Users\Anna Pózna\Dropbox\PHD\EFOP343\Lab-2\leastsquares.m

leastsquares.m

```
function [P,COV,r,sigma_r]=leastsquares(D)
    X=D(:,2:end);
    y=D(:,1);
    if rank(X)<min(size(X))
        error('Error: matrix is rank deficient!')
    end
    P=(inv(X'*X)*X')*y;
    y_M=X*P;
    r=y-y_M;
    sigma_r=var(r);
    m_r=mean(r);
    COV=inv(X'*X)*sigma_r;
    figure
    subplot(2,1,1)
    title('Estimation')
    hold on;
    plot(y,'b');
    plot(y_M,'r');
    legend('measurement','estimation')
```

Current folder

- If you do not specify the full path, MATLAB looks for the files in the current folder
- Files not included in the search path are greyed out
- You can add folders outside of the current folder to the search path:
  - Right click on the folder
  - Add to Path
  - Selected Folders/Selected Folders and Subfolders
- If two functions have the same name, the one in the current folder takes precedence



Create a variable *a* with value 1!

- Type the name and the value in the command prompt and press Enter:
- $>>$ a=1
  a=1
- The variable *a* appeared in the workspace
- The result is displayed in the command window

Perform some basic arithmetic operations!

- Add 2 to *a*:
- $>>$ a+2
  ans=3
- If you do not specify an output variable, the result is stored in *ans*
- Create a new variable *b* which is 5 times *a*!
- $>>$ b=a $*$ 5 ;
- If you put a semicolon (;) after the statement, the result is not displayed (but still computed!)

- Create a vector **x** with value [1 2 3 4] !
- `>> x=[1 2 3 4]`
  `x=1 2 3 4`
- Create a column vector **y** with value $[5\ 6\ 7\ 8]^T$ (end of row = ;)
- `>> y=[5; 6; 7; 8]`

```
y=      5
        6
        7
        8
```

Vectors

- Create a vector **z** using the : operator!
- Syntax: first element : increment : last element
- ```
  >> z = 1:0.5:5
  z = 1.0   1.5   2.0   2.5   3.0   3.5   4.0   4.5
  5.0
  ```
- Transpose a vector: '
- ```
  >> y'
  ans = 5 6 7 8
  ```

Vectors

- Access element of a vector:
  - The indexing in MATLAB starts with 1
  - >> x(2)
    ans = 2
  - Last index: end
  - >> x(**end**)
    ans = 4
- Access range of elements: first index : last index
- >> x(2:4)
  ans = 2 3 4

- Create a matrix
- >> A=[1 2 3; 4 5 6]

```
A =      1 2 3
         4 5 6
```

- Transpose the matrix: '
- >> A'

```
ans =    1 4
         2 5
         3 6
```

- Access element: (row index, column index)
- 2nd row, 1st column:

```
>> A(2,1)
ans = 4
```

- Access range of elements using the : operator
- 2nd row, all columns:

```
>>A(2,:)
ans = 4 5 6
```

Special matrices

- Matrix full of ones: ones(dim1,dim2,...,dimn)

```
>> O = ones(2)
O =       1  1
          1  1
```

- Matrix full of zeroes: zeroes(dim1,dim2,...,dimn)

```
>> Z = zeroes(2,3)
Z =       0  0  0
          0  0  0
```

- Identity matrix: eye(dim1,dim2,...,dimn)

```
>> E = eye(2)
E =       1  0
          0  1
```

- If only one dimension is specified, then it is a square matrix!

Matrix operations

- By default all operations are matrix operations in Matlab
- Addition: +, subtraction: -
  - Matrices must have equal dimensions!

```
>> B= [1 3 5; 2 4 6];
>> A+B
ans =    2    5    8
         6    9    12
```

Matrix operations

- Multiplication: *
  - Inner matrix dimensions must be the same!

```
>> A*B
Error using *
Inner matrix dimensions must agree
>> A*B'
ans =     22        28
          49        64
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Elementwise operators

- Elementwise multiplication `.*` , power `.^` , and division `./`
- The operations are performed by elements
- Matrices must have equal dimensions!

- ```
  >> A.*B
  ans= 1   6   15
       8  20   36
  >> A.^2
  ans =  1  4  9
        16 25 36
  ```
  ```

- Determinant

```
>> C = [1 2; 3 4];
>> det(C)
ans = -2
```

- Eigenvalues and eigenvectors

```
>> [V,D] = eig(C)
V =       -0.8246   -0.4160
           0.5658   -0.9094
D =       -0.3723  0
           0         5.3723
```

Matrix functions

- Inverse

```
>> inv(C)
ans =    -2.0000    1.0000
          1.5000   -0.5000
```

- Introduction to Matlab
- Random numbers
  - Uniformly distributed random numbers
  - Normally distributed random numbers
  - Random numbers from any distribution
  - Control random number generation
  - White noise process
- Basic statistics

Random number generators

- True random numbers
  - Based on some physical phenomena
  - e.g. atmospheric noise, thermal noise, dice
- Pseudorandom numbers
  - Generated by computer algorithms
  - Long sequence of apparently random numbers
  - Determined by the initial value (seed)
  - Can be reproduced

Uniformly distributed random numbers

- The **rand** function generates *uniformly distributed* random numbers in the interval (0,1)
- rand returns one random number

```
>> x = rand
x = 0.0923
```

- rand(n) generates an $n \times n$ matrix of random numbers

```
>> x = rand(2)
x =        0.1863   0.3968
           0.3456   0.5388
```

- rand(dim1,dim2,...,dimn) generates $dim_1 \times dim_2 \times ... \times dim_n$ matrix of random numbers

```
>> x = rand(1,4)
x = 0.4192        0.6852   0.2045   0.8781
```

Uniformly distributed random numbers

- Generate uniformly distributed random numbers on a specified interval, for example (-1,2)!

```
>> a = -1;
>> b = 2;
```

- Multiply the function with the length of the interval (stretch)

```
>> r = (b-a)*rand(1,4);
```

- Shift to the start of the interval

```
>> r = a + (b-a)*rand(1,4)
 r = 1.4442  1.7174  -0.6190  1.7401
```

Uniformly distributed random integers

- The **randi** function generates *uniformly distributed random integers* between 1 and i_{max}
- randi(imax) returns one random integer

```
>> randi(5)
ans = 3
```

- randi([imin,imax]) returns one random integer between i_{min} and i_{max}

```
>> randi([-5,5])
ans = -4
```

- randi(imax,n) generates an $n \times n$ matrix of random integers

```
>> randi(5,2)
ans =    3    4
         5    5
```

Uniformly distributed random integers

- randi(imax,dim1,dim2,...,dimn) generates $dim_1 \times dim_2 \times \ldots \times dim_n$ matrix of random integers

```
>> randi(4,1,3)
ans = 3 1 4
```

Normally distributed random numbers

- The **randn** function generates *normally distributed* random numbers with 0 mean and 1 variance (standard normal distribution).
- randn returns one random number
- randn(n) generates an $n \times n$ matrix of random numbers
- randn(dim1,dim2,...,dimn) generates $dim_1 \times dim_2 \times ... \times dim_n$ matrix of random numbers

Normally distributed random numbers

- Generate normally distributed random numbers with given mean and variance! For example $\mu = -1$, $\sigma^2 = 4 \rightarrow \sigma = 2$
- Solution 1. : Transform the standard normal distribution:
  - Squeeze and shift the standard normal distribution:

```
>> mu = -1; sigma = 2;
>> r = mu + sigma*randn(1,10);
```

- Solution 2. : Use the **normrnd** function:
  - normrnd(mu,sigma,dim1,dim2,...,dimn)
  - Important: the arguments of the **normrnd** function are the **mean** and the **standard deviation**!

Multivariate normal distribution

- The **mvnrnd** function generates multivariate normally distributed random numbers with a given **mean vector** and **covariance matrix**
- The mean vector is an $1 \times n$ vector
- The covariance matrix is an $n \times n$ **symmetric** matrix
- The result is one or more random vectors

- mean vector

```
>> mu = [3, 5];
```

- covariance matrix

```
>> sigma = [1, 0.5; 0.5, 2];
```

- one random vector

```
>> mvnrnd(mu, sigma)
ans = 3.5377        7.6948
```

- three random vectors

```
>> mvnrnd(mu, sigma, 3)
ans =     0.7412        2.1407
          3.8622        4.8575
          3.3188        5.6126
```

Random numbers from any distribution

- The random function can be used to generate random numbers from any distribution
- The distribution can be specified as a *probability distribution object*
  - makedist('distname') creates a probability distribution object specified by 'distname' using the default values
  - makedist('distname','name',value) creates a probability distribution object with parameters given in the name-value pairs

```
>> pd=makedist('poisson','lambda',5)
Pd = PoissonDistribution
         Poisson distribution
         lambda = 5
```

Random numbers from any distribution

- Or with its name and parameter values

```
>> random(pd)
ans = 8
>> random('binomial',10,0.3,1,100);
```

Control random number generation

- The numbers generated by **rand**, **randi** and **randn** depend on the initial state of the random number generator
- The generator resets its state when MATLAB is restarted $\rightarrow$ you get the same numbers at every start of MATLAB

Control random number generation

- The **rng** function can be used to control the random number generation
  - Specify the seed and the type of the random number generator
  - rng(seed) specifies only the seed, e.g. rng(2), rng('shuffle') (current time)
  - rng('generator') specifies only the generator e.g. rng('twister')
  - rng(s) restores the settings of the random number generator to the values captured previously with s = rng

Control random number generation


```
>> rng(2);          % seed = 2
>> x = rand(1,4)
x =      0.4360     0.0259     0.5497     0.4353
>> y=rand(1,4)
y =      0.4204     0.3303     0.2046     0.6193
>> s=rng            % save state
s =        Type: 'twister'
           Seed: 2
           State: [625x1 uint32]
>> w=rand(1,4)
w =      0.4204     0.3303     0.2046     0.6193
>> rng(2)           % seed = 2
>> z=rand(1,4)
z =      0.4360     0.0259     0.5497     0.4353
```

White noise process

- Sequence of independent identically distributed random variables
- Independent: no correlation between the samples
- Identically distributed: random variables have the same distribution
- *The random numbers generated by* ***rand****,* ***randi****,* ***randn****,... compose a white noise process*
- Noise in the nature is usually Gaussian white noise i.e. the random variables have normal (Gaussian) distribution

- Introduction to Matlab
- Random numbers
- Basic statistics

Mean

- $\hat{m} = \frac{1}{n}(x_1 + x_2 + \ldots x_n)$
- The **mean** function can be used in MATLAB to compute the mean value
- mean(x) returns the mean value of the vector x

```
>> x = [2 5 8];
>> mean(x)
ans = 5
```

- mean(A) returns the mean value of the matrix A, along the columns

```
>> A = [5 4 7;1 3 -2];
>> mean(A)
ans = 3 3.5 2.5
```

Mean

- mean(A,dim) returns the mean value of the matrix A along the specified dimension (1 - columns, 2 - rows)

```
>> mean(A,2)
ans =     5.3333
          0.6667
```

- $\sigma = \sqrt{\frac{1}{n}((x_1 - \mu)^2 + (x_2 - \mu)^2 + \cdots + (x_n - \mu)^2)}$
- The **std** function can be used in MATLAB
- std(x) returns the standard deviation of the vector x

```
>> std(x)
ans = 2.4495
```

- std(A) returns the standard deviation of the matrix A, along the columns

```
>> std(A)
ans = 2  0.5        4.5
```

Standard deviation

- std(A,w) returns the weighted standard deviation of each column of A, using the w weight vector
  - w = 0 means that the std is normalized by N-1 (default value)
  - w = 1 means that the std is normalized by N
  - w = [w1, w2,...wk] is a weight vector
- std(A,w,dim) returns the standard deviation of the matrix A along the specified dimension. (1 – columns, 2 – rows)

```
>> std(A,0,2)
ans =     1.5275
          2.5166
```

Variance

- $\sigma^2 = \frac{1}{n}((x_1 - \mu)^2 + (x_2 - \mu)^2 + \cdots + (x_n - \mu)^2)$
- The **var** function can be used in MATLAB
- var(x) returns the variance of the vector x

```
>> var(x)
ans = 6
```

- std(A) returns the variance of the matrix A, along the columns

```
>> var(A)
ans = 8.0000    0.5000    40.5000
```

Variance

- var(A,w) returns the weighted variance of each column of A, using the w weight vector
  - w = 0 means that the variance is normalized by N-1 (default value)
  - w = 1 means that the variance is normalized by N
  - w = [w1, w2,...wk] is a weight vector
- var(A,w,dim) returns the variance of the matrix A along the specified dimension. (1 – columns, 2 – rows)

```
>> var(A,0,2)
ans =     2.3333
          6.3333
```

- Covariance matrix:

$$COV = \begin{bmatrix} \sigma_X^2 & cov(X,Y) & cov(X,Z) \\ cov(Y,X) & \sigma_Y^2 & cov(Y,Z) \\ cov(Z,X) & cov(Z,Y) & \sigma_Z^2 \end{bmatrix}$$

  — symmetric
  — variances in the diagonal
- The **cov** function can be used in MATLAB
- cov(A): If A is a vector of observations, cov(A) returns the variance of A
- if A is a matrix whose columns are observations of random variables, then cov(A) returns the covariance matrix

Covariance

```
>> A2=[1  2; -1  -1; 3  0];
>> cov(A2)
ans =     4.0000     1.0000
          1.0000     2.3333
```

Covariance

- cov(A,B) returns the covariance between the two random variables A and B. If A and B are vectors, then cov(A,B) is a 2×2 covariance matrix. If A and B are matrices of the same size then the covariance is computed so that the matrices are treated as vectors (create one big column vector from the columns of the matrix)

```
>> X = [0.5, 0.25, 2 1];
>> Y = [-1, 5, 2.3, -1.8];
>> cov(X,Y)
ans =     0.5990   -0.2229
         -0.2229    9.8225
```

- correlation coefficient:

$$r_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

- Correlation matrix:

$$R = \begin{bmatrix} r_{X,X} & r_{X,Y} & r_{X,Z} \\ r_{Y,X} & r_{Y,Y} & r_{Y,Z} \\ r_{Z,X} & r_{Z,Y} & r_{Z,Z} \end{bmatrix}$$

  - symmetric
  - diagonal elements are ones
- corrcoef(x,y) returns the matrix of the correlation coefficients between two random variables x and y

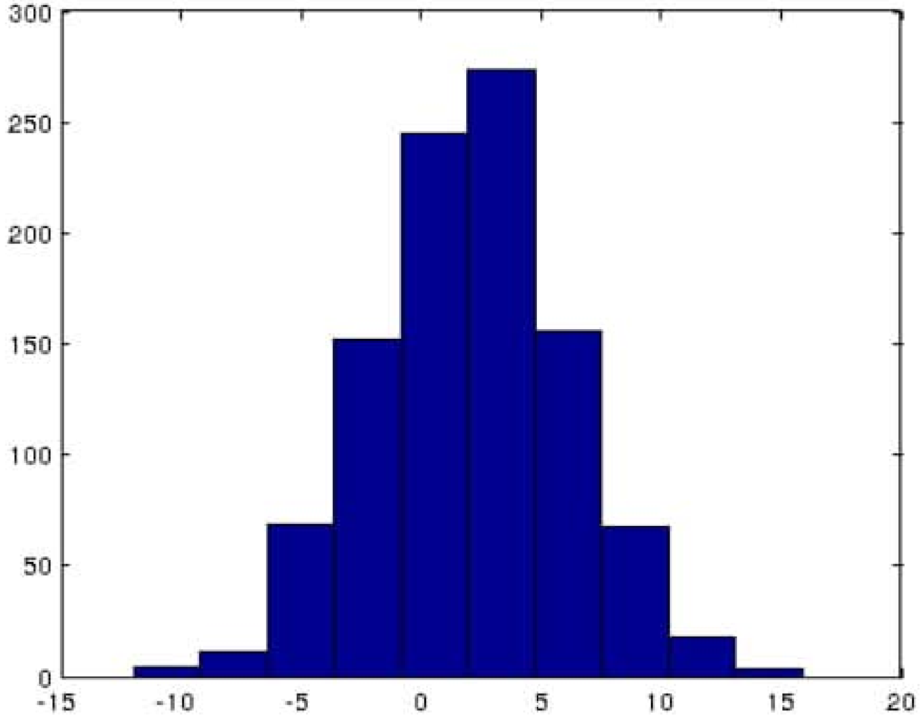
```
>> corrcoef(X,Y)
ans =      1.0000    -0.0919
          -0.0919    1.0000
```

Histogram

- Represents of the frequencies of the data
- Divide the range of values into intervals (bins)
- Count how many of the data falls into each bin
- hist(X) creates a histogram plot of X using 10 bins
- hist(X,nbins) uses a number of bins specified by the scalar, nbins

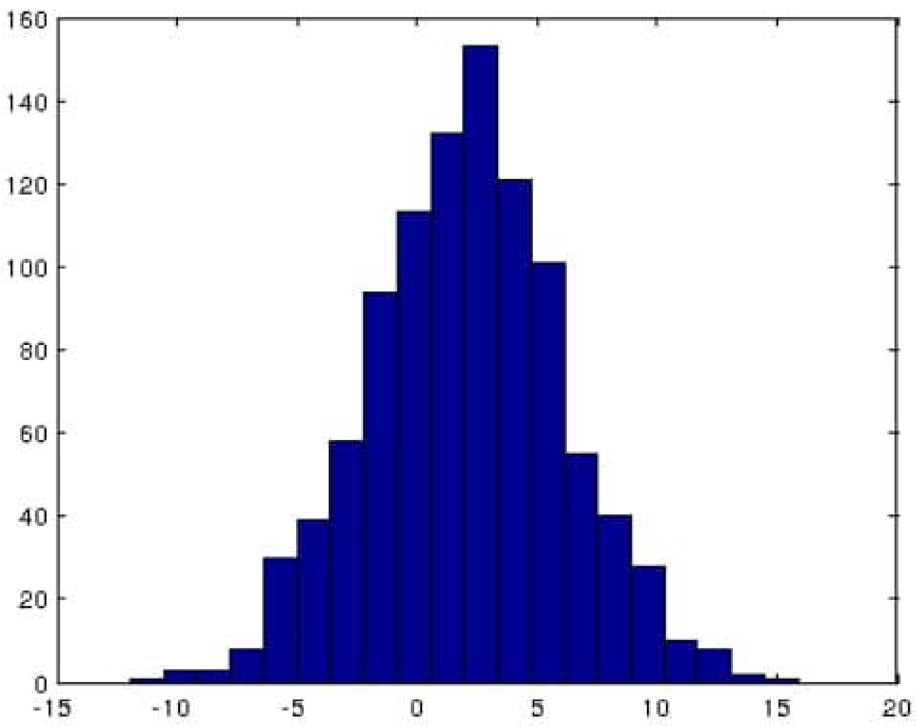
Histogram

```
>> Z = normrnd(2,4,1000,1);
>> hist(Z)
```



Histogram

```
>> hist(Z,20)
```



Autocovariance

$$\hat{r}(D) = \frac{1}{n-s}((x_1 - \hat{m})(x_{s+1} - \hat{m}) + \cdots + (x_{n-s} - \hat{m})(x_n - \hat{m}))$$

- There is no function in MATLAB which computes autocovariance
- **Task**: write a function which gets the vector of data and s as input and computes the autocovariance!

Function declaration in MATLAB:

- Open the Editor window!
- The syntax of the function declaration:

```
function [output_args]=function_name(input_args)
```

- After that you can write the computation steps.
- Save the function in your Current Folder!
- The name of the .m file should be the same as the name of the function!

Solution

```
function r = autocov(data,s)
N = length(data);
r = 0;
m = mean(data);
for i=1:N-s
r=r+(data(i)-m)*(data(i+s)-m);
end
r=1/(N-s)*r;
end
```

Solution

- Test the function!
- Create a data vector, for example $x = [0.1, 0.2, 0.3, 0.4, 0.5]$

```
>> x = [0.1,0.2,0.3,0.4,0.5];
```

- Call the function with s=1!

```
>> r = autocov(x,1)
ans = 0.01
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 2

The properties of the estimates
Linear regression

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Lecture overview

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Analysis of the properties of the estimates
  - Unbiased estimates
  - Confidence intervals, statistical hypothesises
- Linear static models for parameter estimation
  - Simple linear scalar case
  - Linear models with vector valued parameters
- Linear regression
  - The principle of LS estimation
  - The LS estimate
- Properties of the LS estimate
  - Unbiasedness
  - Evaluation of the residuals
- Tutorial

Overview

- Analysis of the properties of the estimates
  - Unbiased estimates
  - Confidence intervals, statistical hypothesises
- Linear static models for parameter estimation
- Linear regression
- Properties of the LS estimate
- Tutorial

Recall
Sample, statistics

Consider a (scalar valued) random variable ξ with probability density function $f_\xi(x)$.

- Sample
  is a collection (set) of n independent random variables

$$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$$

  where every ξ_i has the same distribution as ξ.
  — the sample corresponds to a set of *measurements* about ξ

Recall
Sample, statistics

Consider a (scalar valued) random variable ξ with probability density function $f_\xi(x)$.

- Sample
  is a collection (set) of n independent random variables

  $$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$$

  where every ξ_i has the same distribution as ξ.
  — the sample corresponds to a set of *measurements* about ξ
- Statistics
  is a (deterministic) function of the sample elements (a random variable itself)

  $$s(S) = F(\xi_1, \xi_2, ..., \xi_n)$$

  — a statistics is used to construct an *estimate*

Statistical properties of the sample mean – 1

*Consider a **scalar valued** random variable* ξ *with probability density function* $f_\xi(z)$ and a sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Sample mean: a statistics for estimating the mean value

$$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

Statistical properties of the sample mean – 1

Consider a ***scalar valued*** *random variable* ξ *with probability density function* $f_\xi(z)$ and a sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Sample mean: a statistics for estimating the mean value

$$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n)$$

Important

If the random variable ξ *has a normal or Gaussian distribution* $(\xi \sim \mathbb{N}(m, \sigma^2))$
then μ *has also a normal or Gaussian distribution.*
(For large n the distribution of μ *is approximately Gaussian).*

$$\mu \sim \mathbb{N}(m, \frac{\sigma^2}{n})$$

Recall
Measured data set

*Consider a **scalar valued** random variable ξ with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Measured data set
is a collection (set) of n measurements of the sample elements $\{\xi_1, \xi_2, ..., \xi_n\}$

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

D is a realization of S.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

*Consider a **scalar valued** random variable ξ with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Measured data set

is a collection (set) of n measurements of the sample elements $\{\xi_1, \xi_2, ..., \xi_n\}$

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

D is a realization of S.

Important

*The measured data set contains an actual set of measurements about ξ that are **not** random variables but deterministic values (a realization).*

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

*Consider a **scalar valued** random variable ξ with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$, and with a *measured data set*

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

Estimate
is a realization of a statistics $s(S) = F(\xi_1, \xi_2, ..., \xi_n)$

$$\hat{s}(D) = F(x_1, x_2, ..., x_n)$$

Consider a ***scalar valued*** *random variable* ξ *with a sample* $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$, and with a *measured data set*

$$D(\xi, n) = \{x_1, x_2, ..., x_n\}$$

Estimate

is a realization of a statistics $s(S) = F(\xi_1, \xi_2, ..., \xi_n)$

$$\hat{s}(D) = F(x_1, x_2, ..., x_n)$$

Important

an estimate is computed from the actual measurement values in the data set D

Important (Unbiased estimate)

*An estimate $\hat{s}(D)$ realizing a statistics $s(S)$ of a parameter p is **unbiased**, if the mean value of its statistics is equal to the parameter, i.e. $E\{s(S)\} = p$.*

Unbiased estimates

Important (Unbiased estimate)

*An estimate $\hat{s}(D)$ realizing a statistics $s(S)$ of a parameter p is **unbiased**, if the mean value of its statistics is equal to the parameter, i.e. $E\{s(S)\} = p$.*

Important (Unbiasedness of the sample mean)

The sample mean

$$\hat{\mu}(D) = \frac{x_1 + ... + x_n}{n}$$

is an unbiased estimate of the mean value of the random variable ξ underlying the sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.

Consider a ***scalar valued*** *statistics (i.e. a random variable) $s(S)$ of a parameter p with probability density function $f_s(z)$ and a confidence (significance) level* $(1-\pi)$ $(0 < \pi << 1)$.

Confidence intervals

*Consider a **scalar valued** statistics (i.e. a random variable) $s(S)$ of a parameter p with probability density function $f_s(z)$ and a confidence (significance) level* $(1-\pi)$ $(0 < \pi << 1)$.

Important (Confidence interval)

*The **confidence interval***

$$[p_m(1-\pi), p_M(1-\pi)]$$

is an interval estimation of p on the significance level $(1-\pi)$ *if*

$$\int_{p_m(1-\pi)}^{p_M(1-\pi)} f_s(z)dz = (1-\pi)$$

i.e. p is in the interval $[p_m(1-\pi), p_M(1-\pi)]$ *with probability* $(1-\pi)$

Statistical hypothesis

Consider a ***scalar valued*** *statistics (i.e. a random variable)* $s(S)$ *and an estimate* $\hat{s}(D)$ *of a parameter* p *and a confidence (significance) level* $(1 - \pi)$ $(0 < \pi << 1)$

Statistical hypothesis

*Consider a **scalar valued** statistics (i.e. a random variable) $s(S)$ and an estimate $\hat{s}(D)$ of a parameter p and a confidence (significance) level* $(1-\pi)$ $(0 < \pi << 1)$

Important (Statistical hypothesis)

*A **simple statistical hypothesis** is a relation*

$$H_0 \; : \; p = p^*$$

for the parameter p with a given constant value p^* (we suggest that the value of p is p^*).

Consider a (scalar valued) statistics (i.e. a random variable) $s(S)$ *and an estimate* $\hat{s}(D)$ *of a parameter* p *and a confidence (significance) level* $(1 - \pi)$ $(0 < \pi << 1)$ *with a simple statistical hypothesis* $H_0 \; : \; p = p^*$.

Consider a (scalar valued) statistics (i.e. a random variable) $s(S)$ and an estimate $\hat{s}(D)$ of a parameter p and a confidence (significance) level $(1-\pi)$ $(0 < \pi << 1)$ with a simple statistical hypothesis $H_0 \ : \ p = p^*$.

Important (Statistical hypothesis testing)

Hypothesis testing is to make a decision if we accept the hypothesis H_0 on the confidence (significance) level $(1-\pi)$.

Consider a (scalar valued) statistics (i.e. a random variable) $s(S)$ and an estimate $\hat{s}(D)$ of a parameter p and a confidence (significance) level $(1-\pi)$ $(0 < \pi << 1)$ with a simple statistical hypothesis $H_0 \ : \ p = p^*$.

Important (Statistical hypothesis testing)

Hypothesis testing is to make a decision if we accept the hypothesis H_0 on the confidence (significance) level $(1-\pi)$.

***Hint**: if the estimate $\hat{s}(D)$ is within the confidence interval $[p_m^*(1-\pi), p_M^*(1-\pi)]$ for the parameter p then we accept the hypothesis H_0 on the confidence (significance) level $(1-\pi)$.*

Overview

- Analysis of the properties of the estimates
- Linear static models for parameter estimation
  - Simple linear scalar case
  - Linear models with vector valued parameters
- Linear regression
- Properties of the LS estimate
- Tutorial

Model types

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

$$y = \mathcal{M}(x, p)$$

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

$$y = \mathcal{M}(x, p)$$

- **linear in parameters**

$$\mathcal{M}(x, p) = p^T \mathcal{F}(x)$$

where $\mathcal{F}(x)$ is a possibly nonlinear function of the independent variable vector x

Recall
Model types

$$y = \mathcal{M}(x, p)$$

- **linear in parameters**

$$\mathcal{M}(x, p) = p^T \mathcal{F}(x)$$

  where $\mathcal{F}(x)$ is a possibly nonlinear function of the independent variable vector x

- *dynamic*
  discrete time index $k = 0, 1, ..., K, ...$ such that

$$y(k) = \mathcal{M}(x(k), x(k-1), ..., x(k-K); p) \ , \ \ k = K, K+1, ..., n$$

Simple linear scalar case: model form

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Consider a (scalar valued) dependent variable y with a scalar independent variable x and a scalar parameter p.

Linear model

$$y^{(M)} = p \cdot x$$

Simple linear scalar case: model form

Consider a (scalar valued) dependent variable y with a scalar independent variable x and a scalar parameter p.

Linear model

$$y^{(M)} = p \cdot x$$

Measurements (independent!):

$$Y = \{y_1, y_2, ..., y_m\} \text{ for fixed } X = \{x_1, x_2, ..., x_m\}$$

such that $y_j = p \cdot x_j + \varepsilon_j$, *where* $\varepsilon_j,\ j = 1, ..., m$ *are independent identically distributed random variables with p.d.f.* $f_\varepsilon(z)$.

Simple linear scalar case: model form

Consider a (scalar valued) dependent variable y with a scalar independent variable x and a scalar parameter p.

Linear model

$$y^{(M)} = p \cdot x$$

Measurements (independent!):

$$Y = \{y_1, y_2, ..., y_m\} \text{ for fixed } X = \{x_1, x_2, ..., x_m\}$$

such that $y_j = p \cdot x_j + \varepsilon_j$, where $\varepsilon_j,\ j = 1, ..., m$ are independent identically distributed random variables with p.d.f. $f_\varepsilon(z)$.

Important

Sample for the measurement error

$$S(\varepsilon) = \{(y_1 - px_1),\ ..., (y_m - px_m)\}$$

Simple linear scalar case: residuals

Consider a ***scalar valued dependent variable*** y *with a* ***scalar independent variable*** x *and a* ***scalar parameter*** p

$$y^{(M)} = p \cdot x$$

and with independent measurements such that $y_j = p \cdot x_j + \varepsilon_j$, $\varepsilon_j,\ j = 1, ..., m$ are *independent identically distributed random variables*

Simple linear scalar case: residuals

Consider a ***scalar valued dependent variable*** y *with a* ***scalar independent variable*** x *and a* ***scalar parameter*** p

$$y^{(M)} = p \cdot x$$

and with independent measurements such that $y_j = p \cdot x_j + \varepsilon_j$, $\varepsilon_j,\ j = 1, ..., m$ are *independent identically distributed random variables*

Measured data set: $D_m = \{(y_j; x_j) \mid j = 1, ..., m\}$

Residuals:

$$r_j = y_j - y_j^{(M)} = y_j - p \cdot x_j$$

Sample *for the estimation of the residual properties:*
$S(\varepsilon) = \{r_1, r_2, ..., r_m\}$ *where every* r_i *has the same distribution as* ε.

Static linear models: vector valued parameter – 1

Static linear model
that is linear in parameters $p \in \mathbb{R}^n$ and also in independent variables $x \in \mathbb{R}^n$ but has a single dependent variable y

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

Static linear models: vector valued parameter – 1

Static linear model
that is linear in parameters $p \in \mathbb{R}^n$ and also in independent variables $x \in \mathbb{R}^n$ but has a single dependent variable y

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

Measured data: *m independent measurements*

$$y_j = \sum_{i=1}^{n} x_{ji} p_i + \varepsilon_j \quad , \quad D_m = \{(y_j; x_{j1}, ..., x_{jn}) \mid j = 1, ..., m\}$$

with fixed independent variable values x_{ji}, $j = 1, ..., m$; $i = 1, ..., n$ *and independent identically distributed measurement errors* ε_j.

Static linear models: vector valued parameter – 2

Static linear model

that is linear in parameters $p \in \mathbb{R}^n$ and also in independent variables $p \in \mathbb{R}^n$ but has a single dependent variable y

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

$$y_j = \sum_{i=1}^{m} x_{ji} p_i + \varepsilon_j \quad , \quad D_m = \{(y_j; x_{j1}, ..., x_{jn}) \mid j = 1, ..., m\}$$

Static linear models: vector valued parameter – 2

Static linear model
that is linear in parameters $p \in \mathbb{R}^n$ and also in independent variables $p \in \mathbb{R}^n$ but has a single dependent variable y

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

$$y_j = \sum_{i=1}^{m} x_{ji} p_i + \varepsilon_j \quad , \quad D_m = \{(y_j; x_{j1}, ..., x_{jn}) \mid j = 1, ..., m\}$$

Residuals

$$r_j = y_j - y_j^{(M)} = y_j - x^{(j)T} p$$

where $x^{(j)}$ is the jth fixed independent variable set.

Overview

- Analysis of the properties of the estimates
- Linear static models for parameter estimation
- Linear regression
  - The principle of LS estimation
  - The LS estimate
- Properties of the LS estimate
- Tutorial

Simple linear scalar case: LS estimation – 1

Linear model

$$y^{(M)} = p \cdot x$$

Residuals

$$r_j = y_j - y_j^{(M)} = y_j - p \cdot x_j \ \ , \ \ j = 1, ..., m$$

Simple linear scalar case: LS estimation – 1

Linear model

$$y^{(M)} = p \cdot x$$

Residuals

$$r_j = y_j - y_j^{(M)} = y_j - p \cdot x_j \quad , \quad j = 1, ..., m$$

Loss function: *squared deviation from the model*

$$V(p; X) = \sum_{j=1}^{m} r_j^2 = \sum_{j=1}^{m} (y_j - p \cdot x_j)^2$$

with fixed X.

Simple linear scalar case: LS estimation – 1

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Linear model

$$y^{(M)} = p \cdot x$$

Residuals

$$r_j = y_j - y_j^{(M)} = y_j - p \cdot x_j \;\;,\;\; j = 1, ..., m$$

Loss function: *squared deviation from the model*

$$V(p; X) = \sum_{j=1}^{m} r_j^2 = \sum_{j=1}^{m} (y_j - p \cdot x_j)^2$$

with fixed X.

Important (LS principle)

The least squares (LS) estimation principle: choose the parameter estimate $\hat{p}$ such that the quadratic function $V(p)$ is minimal.

Simple linear scalar case: LS estimation – 2

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Loss function: squared deviation from the model

$$V(p; X) = \sum_{j=1}^{m} r_j^2 = \sum_{j=1}^{m} (y_j - p \cdot x_j)^2$$

with *fixed* X.

Choose the parameter estimate $\hat{p}$ such that the quadratic function $V(p)$ is minimal.

Simple linear scalar case: LS estimation – 2

Loss function: squared deviation from the model

$$V(p;X) = \sum_{j=1}^{m} r_j^2 = \sum_{j=1}^{m} (y_j - p \cdot x_j)^2$$

with *fixed* X.

Choose the parameter estimate $\hat{p}$ such that the quadratic function $V(p)$ is minimal.

Important

Solution: *using optimization*

$$\frac{dV(p)}{dp} = -2 \cdot \sum_{j=1}^{m} x_j (y_j - p \cdot x_j) = 0 \quad \Rightarrow \quad \hat{p} = \frac{1}{\sum_{j=1}^{m} x_j x_j} \cdot \sum_{j=1}^{m} x_j y_j$$

The estimate is a linear function of the measured y_j-s.

Parameter estimation of linear static models – 1

Problem statement

Given:

- A model that is linear in parameters $p \in \mathbb{R}^n$

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

where $x \in \mathbb{R}^n$ are deterministic independent variables (measured and set) and $y^{(M)} \in \mathbb{R}$ is the model output, measured value y is a random variable with ***measurement error***.

Parameter estimation of linear static models – 1

Problem statement

Given:

- A model that is linear in parameters $p \in \mathbb{R}^n$

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

  where $x \in \mathbb{R}^n$ are deterministic independent variables (measured and set) and $y^{(M)} \in \mathbb{R}$ is the model output, measured value y is a random variable with ***measurement error***.
- From m ($m \geq n$) measurements we form

$$y = \begin{bmatrix} y_1 \\ y_2 \\ .. \\ y_m \end{bmatrix} , \quad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$$

Parameter estimation of linear static models – 2

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Consider a weighted ***quadratic loss function*** V

$$
\begin{aligned}
V(p; X) &= r^T W r = \sum_{i=1}^{m} \sum_{j=1}^{m} r_i W_{ij} r_j \\
r_j &= y_j - y_j^{(M)} = y_j - x^{(j)T} p \ , \ j = 1, ..., m
\end{aligned}
$$

  where r is the *residual vector* and W is a weighting matrix (often $W = I$)

Parameter estimation of linear static models – 2

- Consider a weighted ***quadratic loss function*** V

$$V(p;X) = r^T W r = \sum_{i=1}^{m} \sum_{j=1}^{m} r_i W_{ij} r_j$$

$$r_j = y_j - y_j^{(M)} = y_j - x^{(j)T} p \ , \quad j = 1, \ldots, m$$

where r is the *residual vector* and W is a weighting matrix (often $W = I$)

Important (Least squares (LS) estimate)

The LS estimate $\hat{p}$ of the parameters p minimizes V.

The minimum of V is at $\frac{\partial V}{\partial p} = 0$ with $W = I$

$$\hat{p} = (X^T X)^{-1} X^T y$$

The estimate is a linear function of the measured y_j-s.

Overview

- Analysis of the properties of the estimates
- Linear static models for parameter estimation
- Linear regression
- Properties of the LS estimate
  - Unbiasedness
  - Evaluation of the residuals
- Tutorial

Recall
Vector-valued random variables

Given a vector valued random variable ξ

$$\xi \ : \ \xi(\omega), \quad \omega \in \Omega, \quad \xi(\omega) \in \mathbb{R}^{\mu}$$

Its ***mean value*** $m \in R^{\mu}$ *is a real vector.*

Its ***variance*** $COV\{\xi\}$ is a square real matrix, the *covariance matrix*:

$$COV\{\xi\} = E\{(\xi - E\{\xi\})(\xi - E\{\xi\})^{T}\}$$

Covariance matrices are positive definite symmetric matrices:

$$z^{T} COV\{\xi\} z \geq 0 \quad , \quad \forall z \in \mathbb{R}^{\mu}$$

Recall
Linearly transformed random variables

Let us transform the vector-valued random variable $\xi(\omega) \in R^n$ using the non-singular square transformation matrix $T \in R^{n\times n}$:

$$\eta = T\xi$$

The properties of the vector-valued random variable η:

$$E\{\eta\} = TE\{\xi\} \quad , \quad COV\{\eta\} = TCOV\{\xi\}T^T$$

If the random variable ξ has a Gaussian distribution $N(m_\xi, \Delta_\xi)$ with mean value m_ξ and covariance matrix Δ_ξ, then the transformed random variable η will also be Gaussian $N(m_\eta, \Delta_\eta)$, where

$$m_\eta = Tm_\xi \quad , \quad \Delta_\eta = T\Delta_\xi T^T$$

The distribution of the LS estimate

The LS estimate

$$\hat{p} = (X^T X)^{-1} X^T y$$

with X being a fixed independent variable value matrix, and the measured dependent variable vector y is

$$y = X \cdot p + \varepsilon$$

where the measurement errors $\varepsilon_j,\ j = 1, ..., m$ are independent identically distributed random variables with p.d.f. $f_\varepsilon(z)$ and zero mean $E\{\varepsilon\} = 0$.

$$\hat{p} = (X^T X)^{-1} X^T (X \cdot p + \varepsilon) = p + (X^T X)^{-1} X^T \varepsilon$$

The LS estimate

$$\hat{p} = (X^T X)^{-1} X^T y$$

with X being a fixed independent variable value matrix, and the measured dependent variable vector y is

$$y = X \cdot p + \varepsilon$$

where the measurement errors ε_j, $j = 1, ..., m$ are independent identically distributed random variables with p.d.f. $f_\varepsilon(z)$ and zero mean $E\{\varepsilon\} = 0$.

$$\hat{p} = (X^T X)^{-1} X^T (X \cdot p + \varepsilon) = p + (X^T X)^{-1} X^T \varepsilon$$

Important (Unbiasedness of the LS estimate)

The LS estimate is unbiased, because $E\{\hat{p}\} = p$.

The covariance matrix of the LS estimate

The LS estimate

$$\hat{p} = (X^T X)^{-1} X^T y = p + (X^T X)^{-1} X^T \varepsilon$$

with X being a fixed independent variable value matrix resulting in the transformation matrix $T = (X^T X)^{-1} X^T$ (from ε to $\hat{p}$).

The covariance matrix of the LS estimate

The LS estimate

$$\hat{p} = (X^T X)^{-1} X^T y = p + (X^T X)^{-1} X^T \varepsilon$$

with X being a fixed independent variable value matrix resulting in the transformation matrix $T = (X^T X)^{-1} X^T$ (from ε to $\hat{p}$).

The covariance matrix of the estimate is

$$COV\{\hat{p}\} = (X^T X)^{-1} \sigma_\varepsilon^2$$

where σ_ε^2 is the variance of the measurement errors.

The covariance matrix of the LS estimate

The LS estimate

$$\hat{p} = (X^T X)^{-1} X^T y = p + (X^T X)^{-1} X^T \varepsilon$$

with X being a fixed independent variable value matrix resulting in the transformation matrix $T = (X^T X)^{-1} X^T$ (from ε to $\hat{p}$).

The covariance matrix of the estimate is

$$COV\{\hat{p}\} = (X^T X)^{-1} \sigma_\varepsilon^2$$

where σ_ε^2 is the variance of the measurement errors.

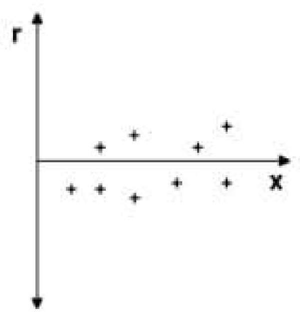
Important (Experiment design)

We can influence the covariance matrix of the estimate by choosing the fixed values of the independent variables properly.

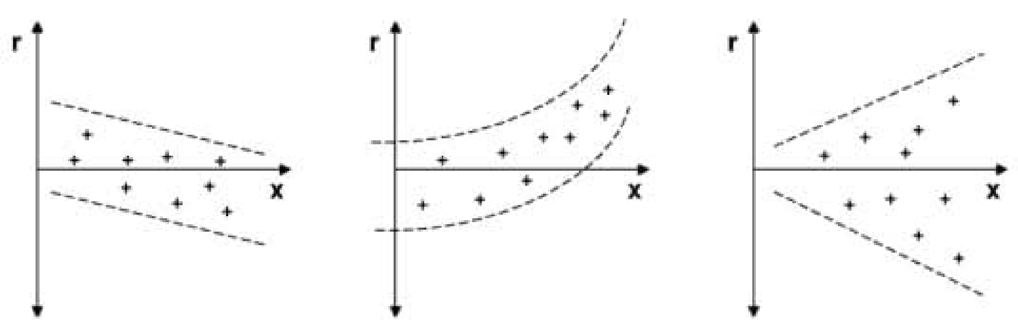
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

For unbiased estimates the residuals should be realizations of independent identically distributed random variables with zero mean.



random



non random

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- A. Scalar valued parameter
- B. Vector valued parameter

Example (Linear regression for scalar parameter – 1)

Consider the following model that is linear in parameters:

$$y^{(M)} = px \tag{1}$$

- *How many parameters does this model have?*
  1 (scalar parameter)

Tutorial problems – A

Example (Linear regression for scalar parameter – 1)

Consider the following model that is linear in parameters:

$$y^{(M)} = px \tag{1}$$

- *How many parameters does this model have?*
  1 (scalar parameter)

- *Consider a measured data set consisting of $(y_j; x_j)$ pairs*

  $$D_5 = \{(0.5; 1.0),\ (0.2; 1.0),\ (0.0; 1.0),\ (-0.5; 1.0),\ (-0.2; 1.0)\}$$

  Compute an estimate of p if possible with its mean value and variance.
  $\hat{p} = 0.0 \quad , \quad \hat{\sigma}_p^2 = 0.0825$

Example (Linear regression for scalar parameter – 2)

Consider the following model that is linear in parameters:

$$y^{(M)} = px \tag{2}$$

- *Consider a measured data set consisting of* $(y_j; x_j)$ *pairs*

  $$D_5 = \{(0.5; 1.0),\ (0.2; 1.0),\ (0.0; 1.0),\ (-0.2; 1.0),\ (-0.5; 1.0)\}$$

  Compute an estimate of p if possible with its mean value and variance.
  $\hat{p} = 0.0\ ,\ \ \hat{\sigma}_p^2 = 0.0825$

Example (Linear regression for scalar parameter – 2)

Consider the following model that is linear in parameters:

$$y^{(M)} = px \tag{2}$$

- *Consider a measured data set consisting of $(y_j; x_j)$ pairs*

  $$D_5 = \{(0.5; 1.0),\ (0.2; 1.0),\ (0.0; 1.0),\ (-0.2; 1.0),\ (-0.5; 1.0)\}$$

  Compute an estimate of p if possible with its mean value and variance.
  $\hat{p} = 0.0 \ ,\ \hat{\sigma}_p^2 = 0.0825$

- *Evaluate the properties of the residuals (mean value, variance)*
  may not be independent – slow drift

Tutorial problems – B

Example (Linear regression for vector valued parameter – 1.1)

Consider the modified model that is linear in parameters:

$$y^{(M)} = ax + b$$

where a and b are unknown scalar parameters.

How many parameters does this model have? Construct the parameter vector p.

$2;\ p = [\, a\ ,\ b\,]^T$

Tutorial problems – B

Example (Linear regression for vector valued parameter – 1.2)

Consider the modified model that is linear in parameters:

$$y^{(M)} = ax + b$$

where a and b are unknown scalar parameters.

Consider a measured data set consisting of $(y_j; x_j)$ pairs

$$D_5 = \{(0.5; 1.0),\ (0.6; 1.0),\ (0.3; 1.0),\ (-0.2; 1.0),\ (0.5; 1.0)\}$$

Construct the matrix X and the vector y needed for the estimation. Comment on the solvability of the estimation problem.

$$y = \begin{bmatrix} 0.5 \\ 0.6 \\ 0.3 \\ -0.2 \\ 0.5 \end{bmatrix}, \quad X = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \\ 1.0 & 1.0 \\ 1.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$$

Estimation is NOT possible, matrix X is singular.

Example (Linear regression for vector valued parameter – 2)

Consider the model that is linear in parameters:

$$y^{(M)} = ax + b$$

where a and b are unknown scalar parameters.

Consider a ***modified measured data set*** *consisting of* $(y_j; x_j)$ *pairs*

$$D_4 = \{(0.5; 1.0),\ (0.6; 1.0),\ (0.3; 0.5),\ (0.2; 0.5)\}$$

Construct the matrix X and the vector y needed for the estimation. Comment on the solvability of the estimation problem.

$$y = \begin{bmatrix} 0.5 \\ 0.6 \\ 0.3 \\ 0.2 \end{bmatrix} , \quad X = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 1.0 \\ 0.5 & 1.0 \\ 0.5 & 1.0 \end{bmatrix}$$

Estimation is POSSIBLE, matrix X is of full rank.

Consider the following model that is linear in parameters:

$$y^{(M)} = \sum_{i+1}^{2} p_i x_i + b$$

where the unknown model parameters are p_1, p_2 and b.

- Consider a measured data set consisting of $(y_j; x_{j1}, x_{j2})$ values

  $$D_4 = \{(0.5; 1.0, 1.0), (0.6; 1.0, 0.9), (0.3; 1.0, 0.5), (0.2; 0.5, 1.0)\}$$

  Compute an estimate of p if possible with its mean value and covariance matrix.

- Evaluate the properties of the residuals (mean value, variance).

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION

COMPUTER LABORATORY 2

Created by: Anna Pózna

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Overview

- Recall
  - Model types
  - Linear regression
- LS estimation of linear scalar models
- LS estimation of static linear models

Overview

- Recall
  - Model types
  - Linear regression
- LS estimation of linear scalar models
- LS estimation of static linear models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Model:

$$y = \mathcal{M}(x, p)$$

  — y: vector of dependent variables, output
  — x: vector of independent variables, input
  — p: parameter vector

- Model linear in parameters:

$$y = p^T \mathcal{F}(x)$$

  — $\mathcal{F}$: function of the independent variables
  — The output is the **product** of the parameter vector and the function of the independent variables!

Recall
Model types

- Linear **scalar** model:

$$y = p \cdot x$$

  - one dependent variable: y
  - one independent variable: x
  - one parameter: p

- **Static** linear model:

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

  - one dependent variable: y
  - vector of independent variables: $x \in \mathbb{R}^n$
  - parameter vector: $p \in \mathbb{R}^n$

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- m number of measurements
- measured output: y_j
- model output: $y_j^{(M)}$
- residuals: difference between the measured output and model output

$$r_j = y_j - y_j^{(M)}$$

- Loss function:

$$V(p; X) = \sum_{j=1}^{m} r_j^2 = \sum_{j=1}^{m} (y_j - y_j^{(M)})^2$$

- **Least squares estimation**: find the parameter p which **minimize the loss function**!

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Solution of the LS estimation:

- LS estimation of linear scalar models:

$$\hat{p} = \frac{1}{\sum_{j=1}^{m} x_j x_j} \sum_{j=1}^{m} x_j y_j$$

- LS estimation of static linear models:

$$\hat{p} = (X^T X)^{-1} X^T y$$

- LS estimation of linear scalar models:
  - variance of the estimation:

$$\sigma_{\hat{p}}^2 = \frac{1}{x^T x} \sigma_\epsilon^2$$

  - σ_ϵ^2 is the variance of the measurement errors (= the variance of the residuals)
- LS estimation of static linear models:
  - covariance of the estimation:

$$COV_{\hat{p}} = (X^T X)^{-1} \sigma_\epsilon^2$$

LS estimation of linear scalar models

- Create a function in Matlab which computes the least squares estimate of linear scalar models!
- The input of the function should be the data matrix which contains the measured values of y_j and x_j in the form of:

$$D = \begin{bmatrix} y_1 & x_1 \\ y_2 & x_2 \\ \vdots & \vdots \\ y_m & x_m \end{bmatrix}$$

- The output of the function should be
  - the estimated parameter $\hat{p}$
  - the variance of the estimation σ
  - the vector of residuals r

Step 1

- Open the Editor and create a new script!
- Declare the function:

```
function [p, sigma, r]=LS_scalar(D)
end
```

Step 2

In the function body:

- Separate the vector of measured values (y) and the vector of the variables (x) from the measured data set!
  - y is the first column of the D matrix
  - x is in the second column of the D matrix

```
y=D(:,1);
x=D(:,2);
```

Step 3

In the function body:

- Compute the estimated value of the parameter using the formula:

$$\hat{p} = \frac{1}{\sum_{j=1}^{m} x_j x_j} \sum_{j=1}^{m} x_j y_j$$

  - $\sum_{j=1}^{m} x_j x_j = x_1 x_1 + x_2 x_2 + \cdots + x_m x_m = x^T x$ is the scalar product of x an x vectors
  - $\sum_{j=1}^{m} x_j y_j = x_1 y_1 + x_2 y_2 + \cdots + x_m y_m = x^T y$ is the scalar product of x and y vectors

```
p=1/(x'*x)*(x'*y);
```

Step 4

In the function body:

- Compute the model output ($y^{(M)} = px$) substituting the estimated value of p:

  ```
  y_M=p*x;
  ```

Step 4

In the function body:

- Compute the model output ($y^{(M)} = px$) substituting the estimated value of p:

  ```
  y_M=p*x;
  ```

- Compute the residuals as the difference between the measured output and the model computed output:

  ```
  r=y-y_M;
  ```

Step 5

In the function body:

- Compute the mean and variance of the residuals:

```
m_r=mean(r);
sigma_r=var(r);
```

Step 5

In the function body:

- Compute the mean and variance of the residuals:

```
m_r=mean(r);
sigma_r=var(r);
```

- Compute the variance of the estimation:

```
sigma=1/(x'*x)*sigma_r;
```

Step 5

In the function body:

- Compute the mean and variance of the residuals:

```
m_r=mean(r);
sigma_r=var(r);
```

- Compute the variance of the estimation:

```
sigma=1/(x'*x)*sigma_r;
```

Save the function as LS_scalar.m in your Current Folder!

Example 1

- Test the function with the data found in dataset_0.txt and dataset_1.txt!
- The structure of the files:
  - Two columns separated by tab
  - First column: y - measured output
  - Second column: x - measured input

- Import data from file:

Import data from file

- Import data from file:
  - On the Home tab, click on the **Import Data** 

 button and choose the file

Import data from file

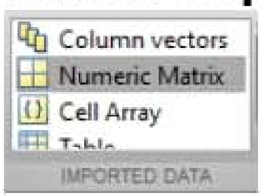
- Import data from file:
  - On the Home tab, click on the **Import Data** 

    button and choose the file
  - Or right click on the file in your Current Folder and choose "Import Data"

Import data from file

- Import data from file:
  - On the Home tab, click on the **Import Data** 

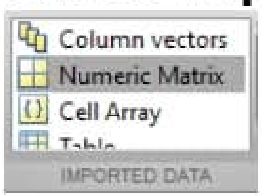
 button and choose the file
  - Or right click on the file in your Current Folder and choose "Import Data"
  - On the **Imported Data** tab choose **Numeric Matrix**



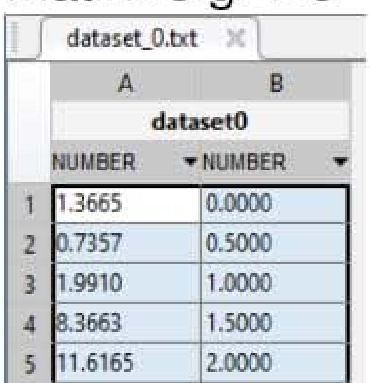
Import data from file

- Import data from file:
  - On the Home tab, click on the **Import Data** 

button and choose the file
  - Or right click on the file in your Current Folder and choose "Import Data"
  - On the **Imported Data** tab choose **Numeric Matrix**



  - Make sure that the whole data is selected and rename the matrix e.g. D0



Import data from file

- Import data from file:
  - Click on the Import Selection button 

Import data from file

- Import data from file:
  - Click on the Import Selection button 

  - The data is successfully imported into the variable D0

Test the function

- Call the function:
  - If you do not specify an output variable, the function returns its first output, in this case the estimated parameter.

```
>> LS_scalar(D0)
ans = 5.0053
```

Test the function

- Call the function:
  - If you do not specify an output variable, the function returns its first output, in this case the estimated parameter.

```
>> LS_scalar(D0)
ans = 5.0053
```

  - If you specify the output variables, the function returns the first , second, ... outputs.
  - In this case, if you want to know the variance of the estimation you should call the LS_scalar function with two output variables.

```
>> [p,s]=LS_scalar(D0)
p = 5.0053
s = 5.8554e-06
```

Test the function

- Call the function:
  - If you do not specify an output variable, the function returns its first output, in this case the estimated parameter.

```
>> LS_scalar(D0)
ans = 5.0053
```

  - If you specify the output variables, the function returns the first , second, ... outputs.
  - In this case, if you want to know the variance of the estimation you should call the LS_scalar function with two output variables.

```
>> [p,s]=LS_scalar(D0)
p = 5.0053
s = 5.8554e-06
```

  - If you want to know the residuals too, you should call the function with three output variables.

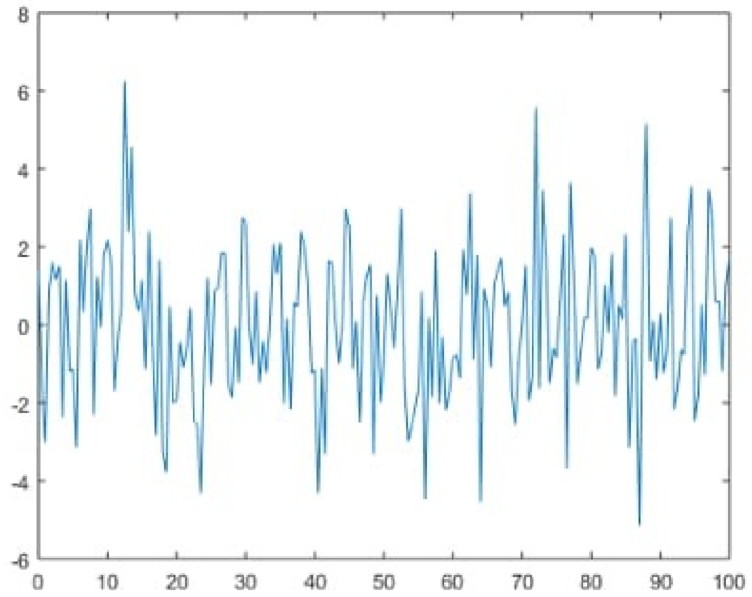
```
>>[p,s,r]=LS_scalar(D0);
```

Evaluation of the results

- The estimated value of the parameter is $p = 5.0053$
- The variance of the estimate is $\sigma^2 = 5.8554 \cdot 10^{-6}$
  - It is very small with respect to the value of the parameter therefore the estimate is good
- Display the residuals!
  - You can use the **plot** function to display data in a 2D coordinate system
  - syntax: plot(xdata, ydata, line_specification,...)

Evaluation of the results

```
>> xdata=D0(:,2);
>> plot(xdata,r);
```



The residuals show random pattern → unbiased estimate!

Example 2

- Estimate the parameter of the linear scalar model based on the measured data found in dataset_1.txt !
- What is the estimated value of the parameter?
- What is the variance of the estimation?
- What can you say about the residuals?

Example 2 - Solution

- Estimate the parameter of the linear scalar model based on the measured data found in dataset_1.txt !
  - Import the data from the file!
  - Call the function:

```
>>[p2,s2,r2]=LS_scalar(D1);
```

- What is the estimated value of the parameter?

Example 2 - Solution

- Estimate the parameter of the linear scalar model based on the measured data found in dataset_1.txt !
  - Import the data from the file!
  - Call the function:

```
>>[p2,s2,r2]=LS_scalar(D1);
```

- What is the estimated value of the parameter?

```
>> p2
p2 = 5.4504
```

- What is the variance of the estimation?

Example 2 - Solution

- Estimate the parameter of the linear scalar model based on the measured data found in dataset_1.txt !
  - Import the data from the file!
  - Call the function:

```
>>[p2,s2,r2]=LS_scalar(D1);
```

- What is the estimated value of the parameter?

```
>> p2
p2 = 5.4504
```

- What is the variance of the estimation?

```
>>s2
s2 = 2.5526e-04
```

  - It is much bigger than in the previous case!

Example 2 - Solution

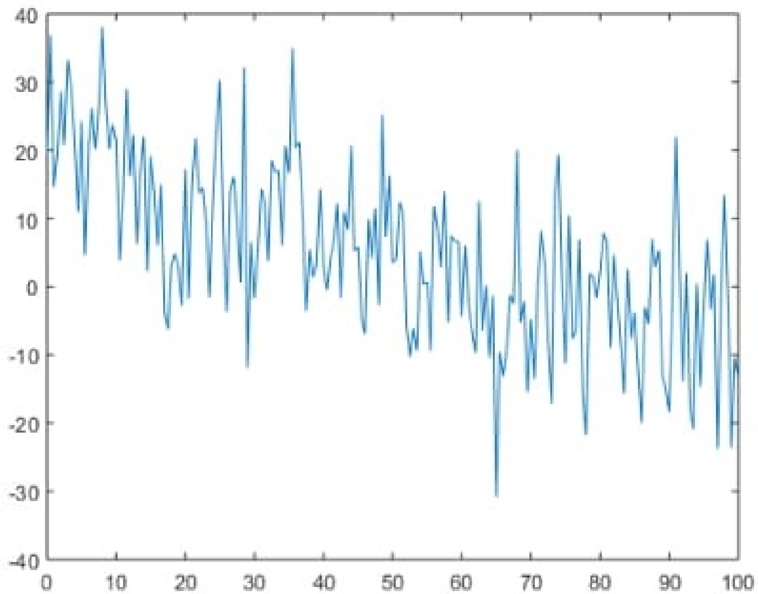
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- What can you say about the residuals?

Example 2 - Solution

- What can you say about the residuals?

```
>> xdata2=D1(:,2);
>> plot(xdata2,r2);
```



The residuals have a trend $\rightarrow$ non random! $\rightarrow$ estimation is biased

LS estimation of static linear models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Model form:

$$y^{(M)} = x^T p = \sum_{i=1}^{n} x_i p_i$$

- x: vector of independent variables
- p: vector of parameters
- Solution of the least squares estimation:

$$\hat{p} = (X^T X)^{-1} X^T y$$

- X: matrix of measured values of the independent variables $(x_1, \ldots, x_n)$

$$X = \begin{bmatrix} x_1^1 & \ldots & x_n^1 \\ \vdots & \vdots & \vdots \\ x_1^m & \ldots & x_n^m \end{bmatrix}$$

- Data matrix: measured values of the dependent and independent variables

$$D = \begin{bmatrix} y_1 & x_1^1 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ y_m & x_m^1 & \dots & x_m^n \end{bmatrix}$$

- Important: The solution exists only if the matrix $X^T X$ is invertible!
- Matrix is invertible $\Leftrightarrow$ it has full rank $\Leftrightarrow$ its determinant is not 0

Task

- Create a function in Matlab which computes the least squares estimate of static linear models!
- The input of the function should be the data matrix
- The output of the function should be
  - the estimated parameter $\hat{p}$
  - the covariance matrix of the estimation
  - the vector of residuals r

Solution

- You can use the `LS_scalar` function with some modifications:

Solution

- You can use the `LS_scalar` function with some modifications:
  - Rename the function to `LS_static`

Solution

- You can use the `LS_scalar` function with some modifications:
  - Rename the function to `LS_static`
  - Rename the output variables to `[p,COV,r]` (just for consistency)

Solution

- You can use the `LS_scalar` function with some modifications:
  - Rename the function to `LS_static`
  - Rename the output variables to `[p,COV,r]` (just for consistency)
  - Separate the matrix of measured independent variables (change the second row of the function):

```
X=D(:,2:end);
```

Solution

- Change the estimation formula:

```
p=inv(X'*X)*X'*y;
```

Solution

- Change the estimation formula:

```
p=inv(X'*X)*X'*y;
```

- Change the computation of the model output:

```
y_M=X*p;
```

Solution

- Change the estimation formula:

```
p=inv(X'*X)*X'*y;
```

- Change the computation of the model output:

```
y_M=X*p;
```

- Compute the covariance instead of the variance:

```
COV=inv(X'*X)*sigma_r;
```

Example 3

- Estimate the parameters of the model using the measured data found in dataset_2.txt!
- The model is in the form of $y = p_1 x_1 + p_2 x_2$
- What is the estimated value of p_1 and p_2?
- Analyze the covariance matrix and the residuals!

Example 3 - Solution

- Estimate the parameters of the model using the measured data found in dataset_2.txt!
  - Import the data from the file!
  - Call the function:

```
>> [P2,COV2,R2]=LS_static(D2);
```

- What is the estimated value of p_1 and p_2?

Example 3 - Solution

- Estimate the parameters of the model using the measured data found in dataset_2.txt!
  - Import the data from the file!
  - Call the function:

```
>> [P2,COV2,R2]=LS_static(D2);
```

- What is the estimated value of p_1 and p_2?

```
>> P2
P2 =  3.0019
      4.8523
```

Example 3 - Solution

- Covariance matrix

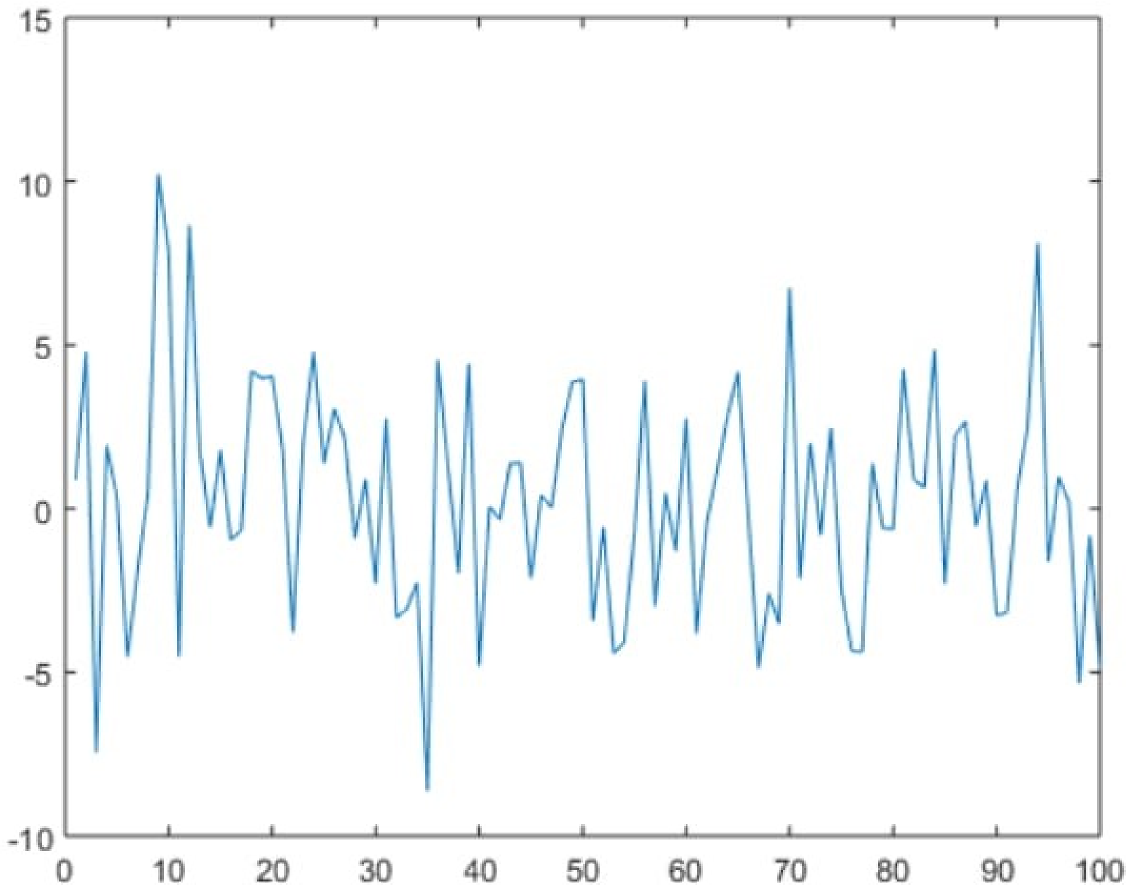
```
>> COV2
COV2 =
        0.0000    -0.0002
       -0.0002     0.0152
```

covariances are small compared to the parameter value

Example 3 - Solution

- Residuals

```
>> plot(R2);
```



Example 4

- Estimate the parameters of the model using the measured data found in dataset_3.txt!
- The model is in the form of $y = p_1 x_1 + p_2 x_2 + p_3 x_3$
- What is the estimated value of p_1, p_2 and p_3?
- Analyze the covariance matrix and the residuals!

Example 4 - Solution

- Estimate the parameters of the model using the measured data found in dataset_4.txt!
  - Import the data from the file!
  - Call the function:

```
>> [P3,COV3,R3]=LS_static(D3);
```

- What is the estimated value of p_1, p_2 and p_3?

Example 4 - Solution

- Estimate the parameters of the model using the measured data found in dataset_4.txt!
  - Import the data from the file!
  - Call the function:

```
>> [P3,COV3,R3]=LS_static(D3);
```

- What is the estimated value of p_1, p_2 and p_3?

```
>> P3
P3 =
   -2.5265
    0.9775
    6.1143
```

Example 4 - Solution

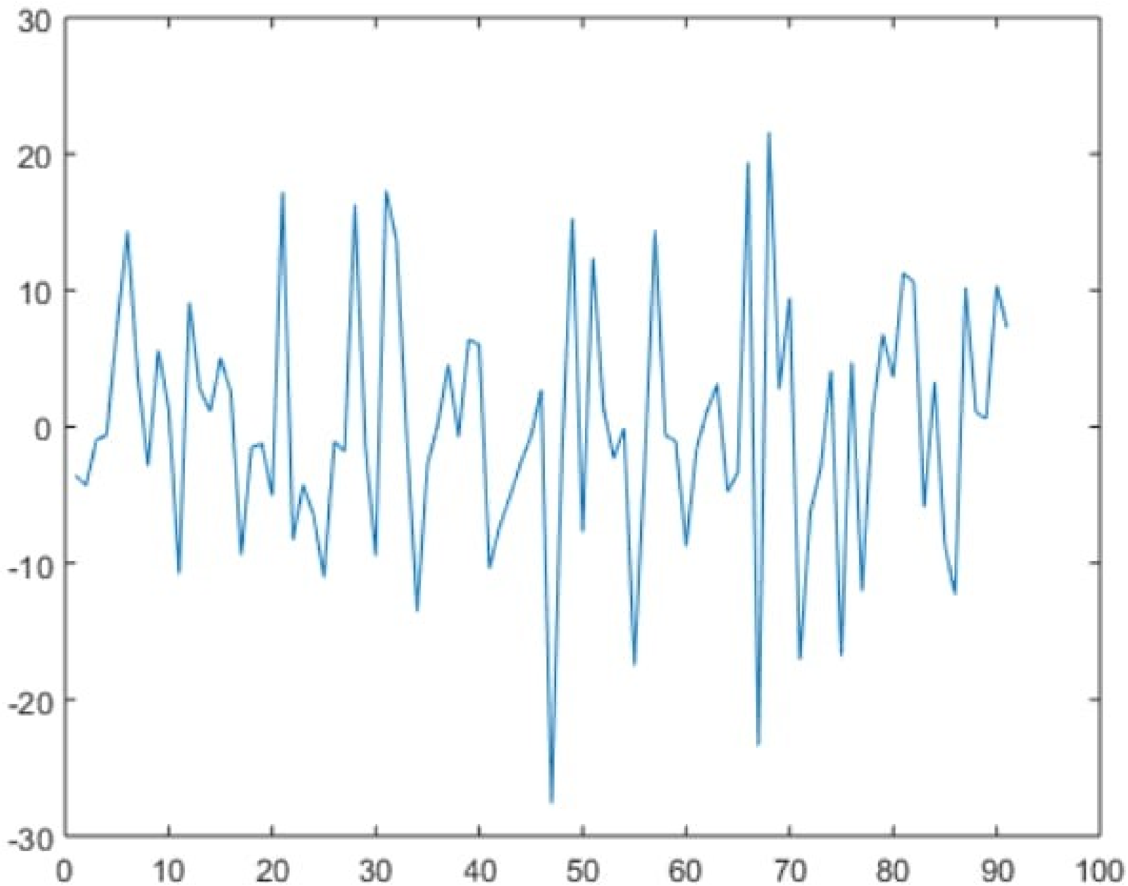
- Covariance matrix

```
>> COV3
COV3 =
     1.5144    0.7795   -1.2201
     0.7795    0.5537   -0.5830
    -1.2201   -0.5830    1.0205
```

Example 4 - Solution

- Residuals

```
>> plot(R3);
```



Special cases

- Model with constant:

$$y = x^T p + c$$

- Introduce a new auxiliary variable x_{aux}, with constant value 1
- The new independent variable vector and parameter vector:

$$x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \\ x_{aux} \end{bmatrix} \qquad p = \begin{bmatrix} p_1 \\ \vdots \\ p_n \\ c \end{bmatrix}$$

- The model is in the standard form now:

$$y = p_1 x_1 + \ldots + p_n x_n + c x_{aux}$$

Example 5

- Estimate the parameters of the model using the measured data found in dataset_4.txt!
- The model is in the form of $y = p_1 x_1 + c$
- What is the estimated value of p_1 and c?
- Analyze the covariance matrix and the residuals!

- Estimate the parameters of the model using the measured data found in dataset_4.txt!
  - Import the data from the file! The data already contain the constant variable vector.
  - Call the function:

```
>> [P4,COV4,R4]=LS_static(D4);
```

- What is the estimated value of p_1 and c?

Example 5 - Solution

- Estimate the parameters of the model using the measured data found in dataset_4.txt!
  - Import the data from the file! The data already contain the constant variable vector.
  - Call the function:

```
>> [P4,COV4,R4]=LS_static(D4);
```

- What is the estimated value of p_1 and c?

```
>> P4
P4 =
   -1.9992
    1.4091
```

Example 5 - Solution

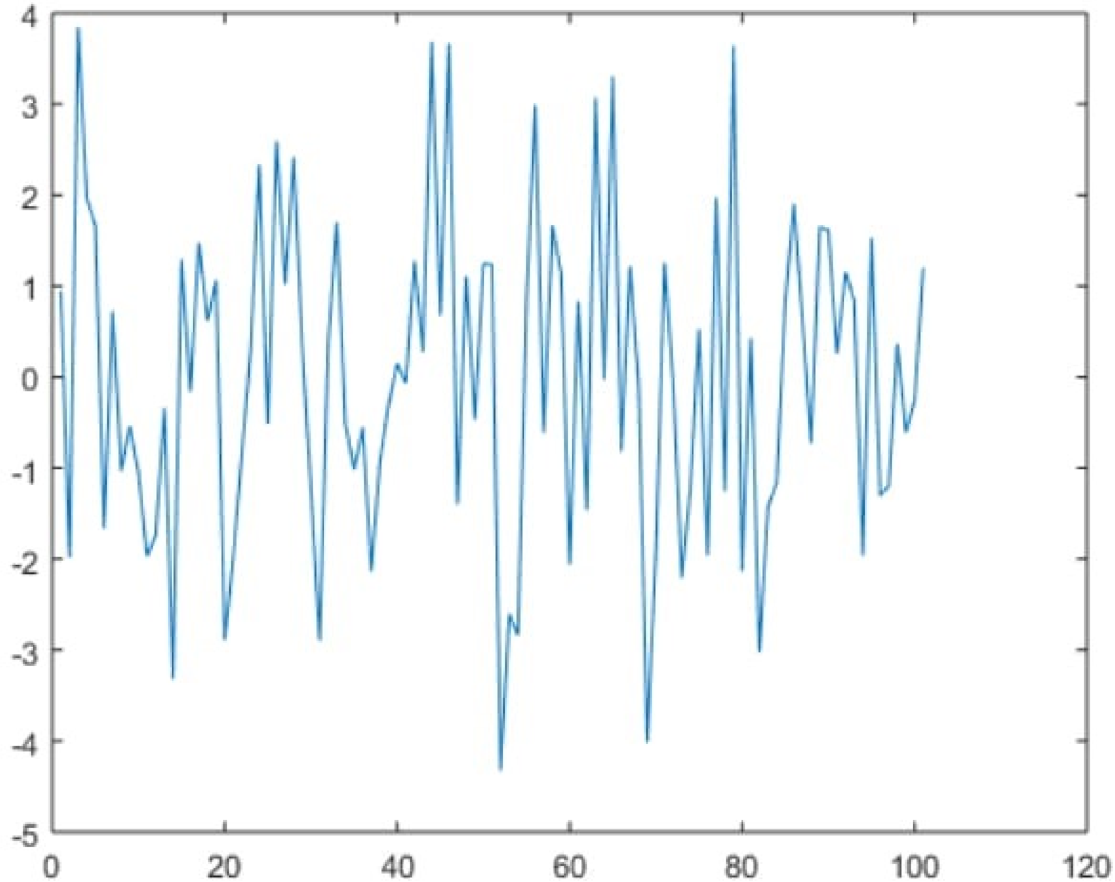
- Covariance matrix

```
>> COV4
COV4 =
      0.0000     -0.0018
     -0.0018      0.1220
```

Example 5 - Solution

- Residuals

```
>> plot(R4);
```



Special cases

- Degenerate data: the columns of the measured data matrix are not independent
- The parameters cannot be estimated

Special cases

- Estimate the parameters of the model using the measured data found in dataset_5.txt!
  - Import the data from the file!
  - Call the function:

```
>> [P5,COV5,R5]=LS_static(D5);
Error using LS_static (line 5)
Error:matrix is rank deficient.
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 3

Stochastic processes
Discrete time stochastic dynamic models

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Lecture overview

- Discrete time stochastic processes
  - Stochastic processes
  - Mean value and covariance
  - White noise processes
- Dynamic models of discrete time systems
  - DT-LTI SISO I/O system models
  - DT-LTI stochastic SISO I/O model
- The principle of parameter estimation – dynamic case
  - Predictive ARX models
- Tutorial

Overview

- Discrete time stochastic processes
  - Stochastic processes
  - Mean value and covariance
  - White noise processes
- Dynamic models of discrete time systems
- The principle of parameter estimation – dynamic case
- Tutorial

Stochastic processes – 1

Stochastic processes are used for describing random disturbances in systems and control theory.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Stochastic processes are used for describing random disturbances in systems and control theory.

Important

Stochastic process
family (indexed sequence) of random variables $x(.,.)$ *where*

$$x : T \times \Omega \to \mathbb{R}^p$$

The set T *is called time.*

Stochastic processes – 1

Stochastic processes are used for describing random disturbances in systems and control theory.

Important

Stochastic process
family (indexed sequence) of random variables $x(.,.)$ *where*

$$x : T \times \Omega \to \mathbb{R}^p$$

The set T *is called time.*

- continuous time *process:* $T \subseteq \mathbb{R}$
- discrete time process*:* $T \subseteq \mathbb{N}$
  discrete time variable $k \sim t_k$

Given a discrete time stochastic process

$$x : T \times \Omega \to \mathbb{R}^p$$

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Given a discrete time stochastic process

$$x : T \times \Omega \to \mathbb{R}^p$$

- Realization
  the (deterministic) function $x(., \omega_0)$ with ω_0 being fixed

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Given a discrete time stochastic process

$$x : T \times \Omega \to \mathbb{R}^p$$

- Realization
  the (deterministic) function $x(., \omega_0)$ with ω_0 being fixed
- Fixed-time value
  $x(k_0, .)$ with k_0 is being fixed is a random variable

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Given a discrete time stochastic process

$$x : T \times \Omega \to \mathbb{R}^p$$

- Realization
  the (deterministic) function $x(., \omega_0)$ with ω_0 being fixed
- Fixed-time value
  $x(k_0, .)$ with k_0 is being fixed is a random variable
- Notation
  $x(k, .) = x(k)$ for the random variable generated from the stochastic process x by fixing the time at k

Distribution functions of a stochastic process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

A stochastic process can be specified by describing all of its finite dimensional distribution functions

Distribution functions of a stochastic process

A stochastic process can be specified by describing all of its finite dimensional distribution functions

Definition

A finite dimensional distribution function of a stochastic process is defined by the formulae

$$F(\zeta_1, ..., \zeta_n; k_1, ..., k_n) = P\{x(k_1) \leq \zeta_1, ..., x(k_n) \leq \zeta_n\}$$

Distribution functions of a stochastic process

A stochastic process can be specified by describing all of its finite dimensional distribution functions

Definition

A finite dimensional distribution function of a stochastic process is defined by the formulae

$$F(\zeta_1, ..., \zeta_n; k_1, ..., k_n) = P\{x(k_1) \le \zeta_1, ..., x(k_n) \le \zeta_n\}$$

Gaussian or normal process: all finite dimensional distribution functions of the process are Gaussian.

Recall
Mean value, covariance

The mean value and variance of the random variable ξ *with its p.d.f.* f_ξ *are*

$$E\{\xi\} = \int x f_\xi(x) dx \ , \ \sigma^2\{\xi\} = \int (x - E\{\xi\})^2 f_\xi(x) dx$$

Mean value, covariance

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The mean value and variance of the random variable ξ *with its p.d.f.* f_ξ *are*

$$E\{\xi\} = \int x f_\xi(x) dx \ , \ \sigma^2\{\xi\} = \int (x - E\{\xi\})^2 f_\xi(x) dx$$

The covariance of two scalar-valued random variables ξ *and* θ *is*

$$COV\{\xi, \theta\} = E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}$$

Recall
Mean value, covariance

The mean value and variance of the random variable ξ *with its p.d.f.* f_ξ *are*

$$E\{\xi\} = \int x f_\xi(x) dx \ , \ \sigma^2\{\xi\} = \int (x - E\{\xi\})^2 f_\xi(x) dx$$

The covariance of two scalar-valued random variables ξ *and* θ *is*

$$COV\{\xi, \theta\} = E\{(\xi - E\{\xi\})(\theta - E\{\theta\})\}$$

Important

The covariance of a scalar-valued random variables ξ *with itself is its variance, i.e.* $COV\{\xi, \xi\} = \sigma^2\{\xi\}$

Mean value function, (auto)covariance function

Definition (mean value function)

The mean-value function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is as follows

$$m_x(k) = Ex(k) = \int_{-\infty}^{\infty} \zeta dF(\zeta, k) \quad , \; k = 0, ..., K, ...$$

Mean value function, (auto)covariance function

Definition (mean value function)

The mean-value function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is as follows

$$m_x(k) = Ex(k) = \int_{-\infty}^{\infty} \zeta dF(\zeta, k) \quad , \; k = 0, ..., K, ...$$

Important

Note that $m_x(k)$ is an ordinary (deterministic) function of time k.

Mean value function, (auto)covariance function

Definition (mean value function)

The mean-value function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is as follows

$$m_x(k) = Ex(k) = \int_{-\infty}^{\infty} \zeta dF(\zeta, k) \quad , \ k = 0, ..., K, ...$$

Important

Note that $m_x(k)$ is an ordinary (deterministic) function of time k.

Definition ((auto)covariance function)

The (auto)covariance function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is defined as

$$r_{xx}(\ell, k) = cov\,[x(\ell), x(k)] = E\{\ [x(\ell) - m(\ell)][x(k) - m(k)]^T\ \}$$

Mean value function, (auto)covariance function

Definition (mean value function)

The mean-value function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is as follows

$$m_x(k) = Ex(k) = \int_{-\infty}^{\infty} \zeta dF(\zeta, k) \quad , \; k = 0, ..., K, ...$$

Important

Note that $m_x(k)$ is an ordinary (deterministic) function of time k.

Definition ((auto)covariance function)

The (auto)covariance function of the stochastic process $\{x(k)\}_{k=0}^{\infty}$ is defined as

$$r_{xx}(\ell, k) = cov\,[x(\ell), x(k)] = E\{\;[x(\ell) - m(\ell)][x(k) - m(k)]^T\;\}$$

Cross-covariance function

Cross-covariance characterizes the inter-dependence of two discrete time stochastic processes.

Cross-covariance function

Cross-covariance characterizes the inter-dependence of two discrete time stochastic processes.

Definition (cross-covariance function)

The cross-covariance function of the stochastic processes $\{x(k)\}_{k=0}^{\infty}$ $\{x(k)\}_{k=0}^{\infty}$ and $\{y(k)\}_{k=0}^{\infty}$ is defined as

$$r_{xy}(\ell, k) = cov\,[x(\ell), y(k)] = E\{\,[x(\ell) - m_x(\ell)][y(k) - m_y(k)]^T\,\}$$

Cross-covariance function

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Cross-covariance characterizes the inter-dependence of two discrete time stochastic processes.

Definition (cross-covariance function)

The cross-covariance function of the stochastic processes $\{x(k)\}_{k=0}^{\infty}$ $\{x(k)\}_{k=0}^{\infty}$ and $\{y(k)\}_{k=0}^{\infty}$ is defined as

$$r_{xy}(\ell, k) = cov\,[x(\ell), y(k)] = E\{\ [x(\ell) - m_x(\ell)][y(k) - m_y(k)]^T\ \}$$

The cross-covariance function is a deterministic two-variate function.

White noise processes

Definition (discrete time white noise, e)

A stochastic process $e = \{e(k)\}_{k=-\infty}^{\infty}$ is a discrete time white noise process if it is a sequence of identically distributed, independent random variables.

Important

Properties

White noise processes

Definition (discrete time white noise, e)

A stochastic process $e = \{e(k)\}_{k=-\infty}^{\infty}$ is a discrete time white noise process if it is a sequence of identically distributed, independent random variables.

Important

Properties

- *stationary process (usually $m(k) = 0$ is assumed)*

White noise processes

Definition (discrete time white noise, e)

A stochastic process $e = \{e(k)\}_{k=-\infty}^{\infty}$ is a discrete time white noise process if it is a sequence of identically distributed, independent random variables.

Important

Properties

- *stationary process (usually $m(k) = 0$ is assumed)*
- *the covariance function in real-valued case is*

$$r_{ee}(\ell) = cov\left[e(k), e(k-\ell)\right] = \begin{cases} \sigma^2 & \ell = 0 \\ 0 & \ell = \pm 1, \pm 2, \ldots \end{cases}$$

White noise processes

Definition (discrete time white noise, e)

A stochastic process $e = \{e(k)\}_{k=-\infty}^{\infty}$ is a discrete time white noise process if it is a sequence of identically distributed, independent random variables.

Important

Properties

- *stationary process (usually $m(k) = 0$ is assumed)*
- *the covariance function in real-valued case is*

$$r_{ee}(\ell) = cov\left[e(k), e(k-\ell)\right] = \begin{cases} \sigma^2 & \ell = 0 \\ 0 & \ell = \pm 1, \pm 2, \ldots \end{cases}$$

- *A white noise process is **not** necessarily a Gaussian process.*

MA processes

Important (unit time delay operator)

Given a signal (time-dependent sequence) $\{x(k),\ k = ..., -1, 0, 1, ...\}$. *The time delay operator* q^{-1} *acts as*

$$q^{-1}x(k) = x(k-1)$$

MA processes

Important (unit time delay operator)

Given a signal (time-dependent sequence) $\{x(k),\ k = ..., -1, 0, 1, ...\}$. *The time delay operator* q^{-1} *acts as*

$$q^{-1}x(k) = x(k-1)$$

Definition (moving average process (MA process))

Let $e = \{\ e(k)\ ,\ k = ..., -1, 0, 1, 2, ...\}$ be a white noise process with variance σ^2. Then the related process $y = \{y(t)\}_{k=-\infty}^{\infty}$ which fulfils

$$y(k) = e(k) + b_1 e(k-1) + ... + b_n e(k-n) = B^*(q^{-1})e(k)$$

is termed a MA process.

Important (unit time delay operator)

Given a signal (time-dependent sequence) $\{x(k),\ k = ..., -1, 0, 1, ...\}$. *The time delay operator* q^{-1} *acts as*
$$q^{-1}x(k) = x(k-1)$$

Definition (moving average process (MA process))

Let $e = \{\ e(k)\ ,\ k = ..., -1, 0, 1, 2, ...\}$ be a white noise process with variance σ^2. Then the related process $y = \{y(t)\}_{k=-\infty}^{\infty}$ which fulfils

$$y(k) = e(k) + b_1 e(k-1) + ... + b_n e(k-n) = B^*(q^{-1})e(k)$$

is termed a MA process.

Mean value and auto-covariance function of a MA process

$$m_y(k) = 0,\ r_{yy}(0) = \sigma^2(1 + b_1^2 + ... + b_n^2),$$
$$r_{yy}(1) = \sigma^2(b_1 + b_1 b_2 + ... + b_{n-1} b_n)$$

Definition (autoregressive process (AR process))

With the white noise process $e = \{e(t)\}_{k=-\infty}^{\infty}$ an AR process is defined as follows

$$y(k) + a_1 y(k-1) + ... + a_n y(k-n) = A^*(q^{-1})y(k) = e(k)$$

Definition (autoregressive process (AR process))

With the white noise process $e = \{e(t)\}_{k=-\infty}^{\infty}$ an AR process is defined as follows

$$y(k) + a_1 y(k-1) + ... + a_n y(k-n) = A^*(q^{-1})y(k) = e(k)$$

Definition (ARMAX process)

An autoregressive-moving average process with an exogeneous signal (ARMAX process) is a linear combination an AR and MA process extended with an exogeneous signal $u = \{u(t)\}_{k=-\infty}^{\infty}$:

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + C^*(q^{-1})e(k)$$

with $A^*(q^{-1}) = 1 + a_1 q^{-1} + a_n q^{-n}$, $B^*(q^{-1}) = b_0 + b_1 q^{-1} + b_m q^{-m}$, $C^*(q^{-1}) = 1 + c_1 q^{-1} + c_n q^{-n}$ and $m < n$.

Overview

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Discrete time stochastic processes
- Dynamic models of discrete time systems
  - DT-LTI SISO I/O system models
  - DT-LTI stochastic SISO I/O model
- The principle of parameter estimation – dynamic case
- Tutorial

Systems

System (**S**): acts on signals

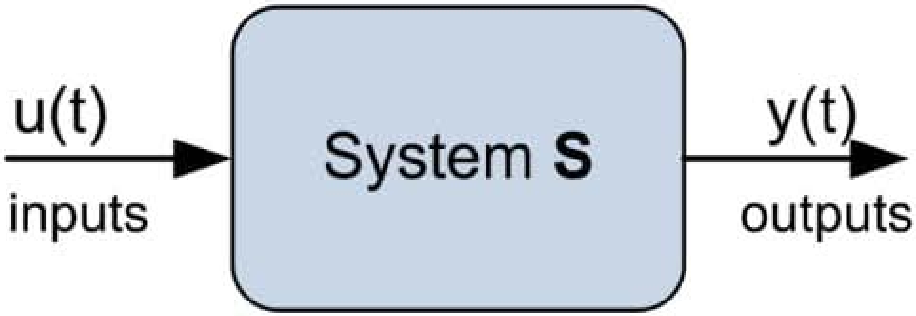
$$y = \mathbf{S}[u]$$

Systems

System (**S**): acts on signals

$$y = \mathbf{S}[u]$$

- inputs (u) and outputs (y)



Basic system properties

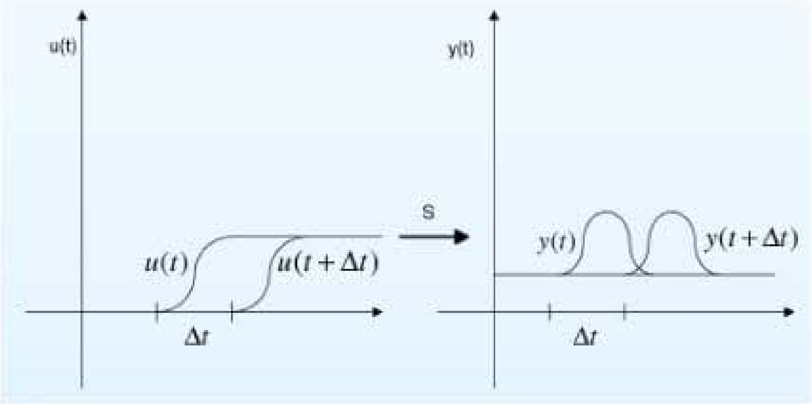
- Linearity

$$\mathbf{S}[c_1 u_1 + c_2 u_2] = c_1 y_1 + c_2 y_2$$

with $c_1, c_2 \in \mathbb{R}, \quad u_1, u_2 \in \mathcal{U}, \quad y_1, y_2 \in \mathcal{Y}$ and
$\mathbf{S}[u_1] = y_1, \quad \mathbf{S}[u_2] = y_2$
Linearity check: use the definition

Basic system properties

- Linearity
  $$\mathbf{S}[c_1 u_1 + c_2 u_2] = c_1 y_1 + c_2 y_2$$
  with $c_1, c_2 \in \mathbb{R}, \quad u_1, u_2 \in \mathcal{U}, \quad y_1, y_2 \in \mathcal{Y}$ and
  $\mathbf{S}[u_1] = y_1, \quad \mathbf{S}[u_2] = y_2$
  Linearity check: use the definition
- Time-invariance
  $$\mathbf{T}_\tau \circ \mathbf{S} = \mathbf{S} \circ \mathbf{T}_\tau$$
  where $\mathbf{T}_\tau$ is the time-shift operator: $\mathbf{T}_\tau(u(t)) = u(t+\tau), \quad \forall t$
  Time invariance check: **constant parameters**



Discrete time LTI SISO I/O system models

Discrete difference equation models: for SISO (single-input single-output) systems

- Backward difference form

$$y(k)+a_1 y(k-1)+...+a_n y(k-n) = b_d u(k-d)+...+b_m u(k-m)$$

where $d = n - m > 0$ is the *pole excess (time delay)*.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Discrete difference equation models: for SISO (single-input single-output) systems

- Backward difference form

$$y(k)+a_1 y(k-1)+...+a_n y(k-n) = b_d u(k-d)+...+b_m u(k-m)$$

  where $d = n - m > 0$ is the *pole excess (time delay)*.

- *Compact form*

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k-d)$$

  where $A^*(q^{-1}) = 1 + a_1 q^{-1} + ... + a_n q^{-n}$ and $B^*(q^{-1}) = b_0 + b_1 q^{-1} + ... + b_m q^{-m}$ are polynomials of the time delay operator q^{-1}.

Discrete time LTI stochastic SISO I/O model

Important (discrete time stochastic LTI input-output model)

The general form of the input-output model of discrete time stochastic LTI SISO systems is the following canonical ARMAX process:

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + C^*(q^{-1})e(k) \tag{1}$$

with the polynomials

$$A^*(q^{-1}) = 1 + a_1 q^{-1} + ... + a_n q^{-n}, C^*(q^{-1}) = c_0 + c_1 q^{-1} + ... + c_n q^{-n}$$
$$B^*(q^{-1}) = b_0 + b_1 q^{-1} + ... + b_m q^{-m}$$

where $C^*(q^{-1})$ is assumed to be a stable polynomial.

Overview

- Discrete time stochastic processes
- Dynamic models of discrete time systems
- The principle of parameter estimation – dynamic case
  - Predictive ARX models
- Tutorial

ARX models

Important (simplest discrete time stochastic LTI input-output model)

Assuming only independent measurement noise, *the model is an ARX model in the form*

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + e(k) \tag{2}$$

where $\{e(k)\}_{k=-\infty}^{\infty}$ *is a white noise process.*

ARX models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Important (simplest discrete time stochastic LTI input-output model)

Assuming only independent measurement noise, *the model is an ARX model in the form*

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + e(k) \tag{2}$$

where $\{e(k)\}_{k=-\infty}^{\infty}$ *is a white noise process.*

Important (predictive form of ARX models)

The predictive form of the ARX model is

$$y(k) = -a_1 y(k-1) - .. - a_n y(k-n) + b_0 u(k) + ... +$$
$$+ b_m u(k-m) + e(k) = p^T \varphi(k) + e(k)$$

This model is ***linear in parameters*** $p = [-a_1 \; ... \; -a_n \mid b_0 \; ... b_m]^T$ *if one measures the data*

$$\varphi(k) = [y(k-1) \; ... \; y(k-n) \mid u(k) \; ... \; u(k-m)]^T$$

Tutorial problems
Stochastic processes

- A. Moving average processes
- B. Two stochastic processes

Tutorial problems – A

Example (Simple MA process – 1)

Given a scalar-valued white noise stochastic process $\{e(k)\}_{-\infty}^{\infty}$ with variance σ^2. Let us construct from it a stochastic process by the equation

$$y(k) = e(k) + 0.5e(k-1) + 0.6e(k-2)$$

- *What kind of process is the stochastic process $\{y(k)\}_{-\infty}^{\infty}$?*
  A moving average (MA) process

Example (Simple MA process – 1)

Given a scalar-valued white noise stochastic process $\{e(k)\}_{-\infty}^{\infty}$ with variance σ^2. Let us construct from it a stochastic process by the equation

$$y(k) = e(k) + 0.5e(k-1) + 0.6e(k-2)$$

- *What kind of process is the stochastic process $\{y(k)\}_{-\infty}^{\infty}$?*
  A moving average (MA) process

- *Compute the mean value function $m_y(k)$ and the (auto)covariance function $r_{yy}(k)$ of the stochastic process $\{y(k)\}_{-\infty}^{\infty}$.*
  $m_y(k) \equiv 0$ *for* $k + 0, 1, \ldots$
  $r_{yy}(0) = \sigma^2(1 + 0.5^2 + 0.6^2)$, $r_{yy}(\pm 1) = \sigma^2(0.5 + 0.5 \cdot 0.6)$
  $r_{yy}(\pm 2) = \sigma^2 \cdot 0.6$, $r_{yy}(\pm \ell) = 0$, $\ell > 2$

Example (Simple MA process – 2)

Consider the following stochastic process:

$$w(k) = z(k) + 0.1z(k-1) + 0.8z(k-3)$$

where z is a sequence of independent scalar valued random variables with the same distribution, $E(z(k)) = 0$, and $D(z(k)) = \sigma$, for every k.

- *What kind of process is the stochastic process $\{z(k)\}_{-\infty}^{\infty}$?*
  A white noise process

Example (Simple MA process – 2)

Consider the following stochastic process:

$$w(k) = z(k) + 0.1z(k-1) + 0.8z(k-3)$$

where z is a sequence of independent scalar valued random variables with the same distribution, $E(z(k)) = 0$, and $D(z(k)) = \sigma$, for every k.

- *What kind of process is the stochastic process $\{z(k)\}_{-\infty}^{\infty}$?*
  A white noise process
- *What kind of process is the stochastic process $\{w(k)\}_{-\infty}^{\infty}$?*
  A moving average (MA) process

Tutorial problems – A

Example (Simple MA process – 2)

Consider the following stochastic process:

$$w(k) = z(k) + 0.1z(k-1) + 0.8z(k-3)$$

where z is a sequence of independent scalar valued random variables with the same distribution, $E(z(k)) = 0$, and $D(z(k)) = \sigma$, for every k.

- *What kind of process is the stochastic process $\{z(k)\}_{-\infty}^{\infty}$?*
  A white noise process
- *What kind of process is the stochastic process $\{w(k)\}_{-\infty}^{\infty}$?*
  A moving average (MA) process
- *Compute the (auto)covariance function $r_{ww}(k)$ for $k = 1, 3, -2$.*
  $m_w(k) \equiv 0$ *for* $k + 0, 1, \ldots$
  $r_{ww}(1) = \sigma^2 \cdot 0.1$, $r_{ww}(3) = \sigma^2 \cdot 0.8$
  $r_{ww}(-2) = \sigma^2 \cdot 0.1 \cdot 0.8$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$
\begin{aligned}
z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\
y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2)
\end{aligned}
$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

Tutorial problems – B

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$\begin{aligned} z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\ y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2) \end{aligned}$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

$m_z(k) \equiv 0$, $m_z(k) \equiv 0$, $r_{zy}(k) \neq r_{zy}(-k)$ *!!!*

Tutorial problems – B

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$\begin{aligned} z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\ y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2) \end{aligned}$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

$m_z(k) \equiv 0$, $m_z(k) \equiv 0$, $r_{zy}(k) \neq r_{zy}(-k)$ *!!!*

- $r_{zy}(0) = \sigma^2(1 + 0.6 \cdot 0.3 + 0.1 \cdot 0.8)$

Tutorial problems – B

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$\begin{aligned} z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\ y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2) \end{aligned}$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

$m_z(k) \equiv 0$, $m_z(k) \equiv 0$, $r_{zy}(k) \neq r_{zy}(-k)$!!!

- $r_{zy}(0) = \sigma^2(1 + 0.6 \cdot 0.3 + 0.1 \cdot 0.8)$
- $r_{zy}(1) = \sigma^2(1 \cdot 0.6 + 0.1 \cdot 0.3)$, $r_{zy}(-1) = \sigma^2(1 \cdot 0.3 + 0.6 \cdot 0.8)$

Tutorial problems – B

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$\begin{aligned} z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\ y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2) \end{aligned}$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

$m_z(k) \equiv 0$, $m_z(k) \equiv 0$, $r_{zy}(k) \neq r_{zy}(-k)$ *!!!*

- $r_{zy}(0) = \sigma^2(1 + 0.6 \cdot 0.3 + 0.1 \cdot 0.8)$
- $r_{zy}(1) = \sigma^2(1 \cdot 0.6 + 0.1 \cdot 0.3)$, $r_{zy}(-1) = \sigma^2(1 \cdot 0.3 + 0.6 \cdot 0.8)$
- $r_{zy}(2) = \sigma^2 \cdot 1 \cdot 0.1$, $r_{zy}(-2) = \sigma^2 \cdot 1 \cdot 0.8$

Example (Cross-covariance)

Consider the following two moving-average (MA) processes:

$$\begin{aligned} z(k) &= e(k) + 0.6e(k-1) + 0.1e(k-2) \\ y(k) &= e(k) + 0.3e(k-1) + 0.8e(k-2) \end{aligned}$$

where $\{e(k)\}_{-\infty}^{\infty}$ *is a discrete time white noise process with variance* $D^2(e(k)) = \sigma^2$

Compute the cross-covariance function $r_{zy}(k)\ \forall k$

$m_z(k) \equiv 0$, $m_z(k) \equiv 0$, $r_{zy}(k) \neq r_{zy}(-k)$ *!!!*

- $r_{zy}(0) = \sigma^2(1 + 0.6 \cdot 0.3 + 0.1 \cdot 0.8)$
- $r_{zy}(1) = \sigma^2(1 \cdot 0.6 + 0.1 \cdot 0.3)$, $r_{zy}(-1) = \sigma^2(1 \cdot 0.3 + 0.6 \cdot 0.8)$
- $r_{zy}(2) = \sigma^2 \cdot 1 \cdot 0.1$, $r_{zy}(-2) = \sigma^2 \cdot 1 \cdot 0.8$
- $r_{zy}(k) = r_{zy}(-k) = 0$, $|k| > 2$

HOMEWORK

Given a scalar-valued white noise stochastic process $\{e(k)\}_{-\infty}^{\infty}$ with variance σ^2. Let us construct from it a stochastic process by the equation

$$y(k) = e(k) - 0.2e(k-1)$$

- What kind of process is the stochastic process $\{y(k)\}_{-\infty}^{\infty}$?
- Compute the mean value function $m_y(k)$ and the (auto)covariance function $r_{yy}(k)$ of the stochastic process $\{y(k)\}_{-\infty}^{\infty}$ for the values $k = 0, \pm 1, \pm 2, \pm 3, ...$!
- Compute the cross-covariance function $r_{ye}(k)$ for the values $k = 0, \pm 1, \pm 2, \pm 3, ...$!

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 4

Least squares (LS) estimation and its properties in the dynamic case

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Lecture overview

- Discrete time LTI stochastic input-output models
  - DT-LTI SISO I/O system models
- Minimizing the prediction error
  - Predictive input-output models
  - Minimizing the prediction errors
- The least squares estimate
  - Predictive models linear in parameters
  - LS estimation of ARX model parameters
- Properties of the dynamic least squares estimate
  - Asymptotic behavior of the LS estimate

Overview

- Discrete time LTI stochastic input-output models
  - DT-LTI SISO I/O system models
- Minimizing the prediction error
- The least squares estimate
- Properties of the dynamic least squares estimate

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

System (**S**): acts on signals

$$y = \mathbf{S}[u]$$

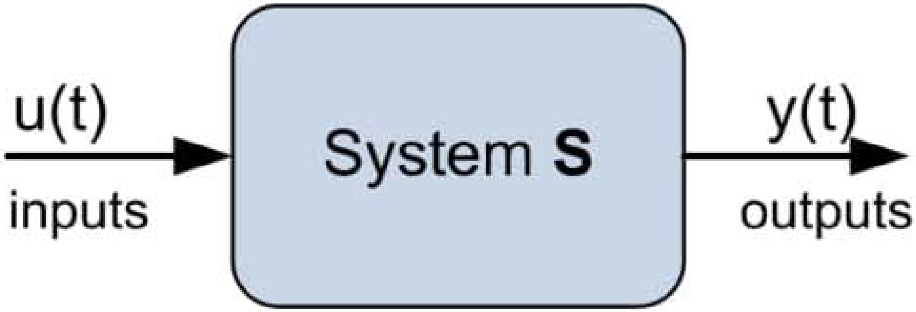
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

System (**S**): acts on signals

$$y = \mathbf{S}[u]$$

- inputs (u) and outputs (y)



Recall
System properties

- Linearity

$$\mathbf{S}[c_1 u_1 + c_2 u_2] = c_1 y_1 + c_2 y_2$$

with $c_1, c_2 \in \mathbb{R}, \quad u_1, u_2 \in \mathcal{U}, \quad y_1, y_2 \in \mathcal{Y}$ and
$\mathbf{S}[u_1] = y_1, \quad \mathbf{S}[u_2] = y_2$
Linearity check: use the definition

- Linearity

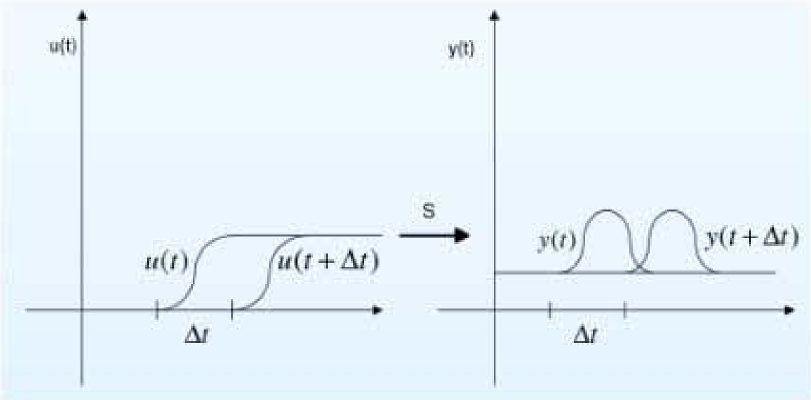
$$\mathbf{S}[c_1 u_1 + c_2 u_2] = c_1 y_1 + c_2 y_2$$

with $c_1, c_2 \in \mathbb{R}, \quad u_1, u_2 \in \mathcal{U}, \quad y_1, y_2 \in \mathcal{Y}$ and
$\mathbf{S}[u_1] = y_1, \quad \mathbf{S}[u_2] = y_2$
Linearity check: use the definition

- Time-invariance

$$\mathbf{T}_\tau \circ \mathbf{S} = \mathbf{S} \circ \mathbf{T}_\tau$$

where $\mathbf{T}_\tau$ is the time-shift operator: $\mathbf{T}_\tau(u(t)) = u(t+\tau), \quad \forall t$
Time invariance check: **constant parameters**



Recall: Discrete time LTI SISO I/O system models

Discrete difference equation models: for SISO (single-input single-output) systems

- Backward difference form

$$y(k)+a_1 y(k-1)+...+a_n y(k-n) = b_0 u(k-d)+...+b_m u(k-d-m)$$

where $d = n - m > 0$ is the *pole excess (time delay)*.

Recall: Discrete time LTI SISO I/O system models

Discrete difference equation models: for SISO (single-input single-output) systems

- Backward difference form

$$y(k)+a_1y(k-1)+...+a_ny(k-n)=b_0u(k-d)+...+b_mu(k-d-m)$$

where $d=n-m>0$ is the *pole excess (time delay)*.

- *Compact form*

$$A^*(q^{-1})y(k)=B^*(q^{-1})u(k-d)$$

where $A^*(q^{-1})=1+a_1q^{-1}+...+a_nq^{-n}$ and $B^*(q^{-1})=b_0+b_1q^{-1}+...+b_mq^{-m}$ are polynomials of the time delay operator q^{-1}.

Recall: Discrete time LTI stochastic SISO I/O model

Important (discrete time stochastic LTI input-output model)

The general form of the input-output model of discrete time stochastic LTI SISO systems is the following canonical ARMAX process:

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + C^*(q^{-1})e(k)$$

with the polynomials

$$A^*(q^{-1}) = 1 + a_1 q^{-1} + ... + a_n q^{-n}, \; C^*(q^{-1}) = c_0 + c_1 q^{-1} + ... + c_n q^{-n}$$
$$B^*(q^{-1}) = b_0 + b_1 q^{-1} + ... + b_m q^{-m}$$

where $C^*(q^{-1})$ is assumed to be a stable polynomial.

Important (simplest discrete time stochastic LTI input-output model)

Assume only independent measurement noise, the model is an ARX model in the form

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + e(k) \tag{1}$$

where $\{e(k)\}_{k=-\infty}^{\infty}$ is a white noise process.

Recall
ARX models

Important (simplest discrete time stochastic LTI input-output model)

Assume only independent measurement noise, the model is an ARX model in the form

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + e(k) \tag{1}$$

where $\{e(k)\}_{k=-\infty}^{\infty}$ *is a white noise process.*

Important

The predictive form of the ARX model (with $d = n - m > 0$*) is*

$$y(k) = -a_1 y(k-1) - .. - a_n y(k-n) + b_0 u(k) + .. + b_m u(k-m) + e(k$$
$$= p^T \varphi(k-1) + e(k)$$

This model is ***linear in parameters*** $p = [-a_1 \; ... \; -a_n \; b_0 \; ...b_m]^T$ *if one measures the data* $\varphi(k-1) = [y(k-1) \; ... \; y(k-n) \; u(k) \; ... \; u(k-m)]^T$.

Overview

- Discrete time LTI stochastic input-output models
- Minimizing the prediction error
  - Predictive input-output models
  - Minimizing the prediction errors
- The least squares estimate
- Properties of the dynamic least squares estimate

Predictive input-output models, SISO case

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

SISO LTI stochastic input-output models - general form

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

where F, G and Δ are linear functions of the time shift operator q^{-1} and $\{e(k)\}_0^\infty$ is a white noise process.

Predictive input-output models, SISO case

SISO LTI stochastic input-output models - general form

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

where F, G and Δ are linear functions of the time shift operator q^{-1} and $\{e(k)\}_0^\infty$ is a white noise process.

*The predictive form is **without the stochastic term***

$$\hat{y}(k|p) = W_y(q^{-1}, p) \cdot y(k) + W_u(q^{-1}, p) \cdot u(k)$$

*The coefficients $W_y(q^{-1}, p)$ and $W_u(q^{-1}, p)$ are so-called linear filters , where p is the **vector of constant, unknown parameters to be estimated***

Predictive form of ARMAX models

General I/O model of discrete time linear time invariant stochastic SISO systems

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

Predictive form of ARMAX models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

General I/O model of discrete time linear time invariant stochastic SISO systems

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

Important

Predictive form:

$$\hat{y}(k|p) = y(k) - C^*(q^{-1}) \cdot e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$$

where

$$p = [a_1 \; ... \; a_n \;\; b_0 \; ...b_m \;\; c_1 \; ... \; c_n]^T$$

Predictive form of ARMAX models

General I/O model of discrete time linear time invariant stochastic SISO systems

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

Important

Predictive form:

$$\hat{y}(k|p) = y(k) - C^*(q^{-1}) \cdot e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$$

where

$$p = [a_1 \ \dots \ a_n \ \ b_0 \ \dots b_m \ \ c_1 \ \dots \ c_n]^T$$

It contains only the past measured data (!!) without the noise term.

Predictive form of ARX models

Consider the simplest case:

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

when the output noise is white . In this case $C^*(q^{-1}) = 1$.

Predictive form of ARX models

Consider the simplest case:

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

when the output noise is white. In this case $C^*(q^{-1}) = 1$.

Important

Predictive form

$$\hat{y}(k|p) = y(k) - e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$$

Predictive form of ARX models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Consider the simplest case:

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + C^*(q^{-1}) \cdot e(k)$$

when the output noise is white. In this case $C^*(q^{-1}) = 1$.

Important

Predictive form

$$\hat{y}(k|p) = y(k) - e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$$

The elements of the estimator:

$$p = [-a_1 \; - a_2 \; \ldots \; - a_n \;\; b_0 \; b_1 \; \ldots \; b_m]^\top \qquad N > n + m$$

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots +$$
$$+ \ldots + b_m \cdot u(k-m)$$

Nonlinear time-invariant single output systems

The general predictive form:

$$\hat{y}(k|p) = g(k, D[1, k-1]; p)$$

with ***time series of measured data:***

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$$

Nonlinear time-invariant single output systems

The general predictive form:

$$\hat{y}(k|p) = g(k, D[1, k-1]; p)$$

with ***time series of measured data:***

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$$

Important (Linear-in-parameter case)

Systems that are linear-in-parameters :

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D[1, k-1])$$

Example: ARX model

ARX model is a model that is **linear in parameters**.
Model elements:

- predictive model form

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + \\ + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Example: ARX model

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARX model is a model that is **linear in parameters**.

Model elements:

- predictive model form

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + \\ + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

- parameters

$$p = [-a_1 \ \ldots \ -a_n \ \ b_0 \ \ldots \ b_m]^\top$$

Example: ARX model

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARX model is a model that is **linear in parameters**.
Model elements:

- predictive model form

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + \\ + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

- parameters

$$p = [-a_1 \ \ldots \ -a_n \ \ b_0 \ \ldots \ b_m]^\top$$

- **regressor** ($\varphi(k)$)

$$g^*(k, D[1, k-1]) = \varphi(k)$$
$$\varphi(k) = [y(k-1) \ \ \ldots \ -y(k-n) \ \ u(k) \ \ldots \ u(k-m)]^\top$$

The prediction error

The **prediction error series** can be computed from the measured variables and the model output:

$$\varepsilon(k,p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$$

The prediction error

The **prediction error series** can be computed from the measured variables and the model output:

$$\varepsilon(k,p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$$

Important

Principle of parameter estimation: *A parameter estimation method generates an estimated parameter from the measured data :*

$$D^N \to \hat{p}_N$$

The model is "good", i.e. the estimated parameters are "good" if the prediction errors are "small".

The prediction error

The **prediction error series** can be computed from the measured variables and the model output:

$$\varepsilon(k,p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$$

Important

Principle of parameter estimation: *A parameter estimation method generates an estimated parameter from the measured data :*

$$D^N \to \hat{p}_N$$

The model is "good", i.e. the estimated parameters are "good" if the prediction errors are "small".

Magnitude of the prediction error
The "size" of the prediction error series $\varepsilon(k,p)$ is measured using an appropriate signal norm .

Parameter estimation method is a mapping: $D^N \to \hat{p}_N$

Important (The general parameter estimation problem)

Given:

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Parameter estimation method is a mapping: $D^N \to \hat{p}_N$

Important (The general parameter estimation problem)

Given:

- ***measured data***: $D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$

Parameter estimation method is a mapping: $D^N \to \hat{p}_N$

Important (The general parameter estimation problem)

Given:

- ***measured data***: $D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$
- ***predictive parametrized model*** $\hat{y}(k|p) = g(k, D[1, k-1]; p)$ *generating the* ***prediction error series*** *(discrete time signal):* $\varepsilon(k, p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$

Minimizing the prediction error

Parameter estimation method is a mapping: $D^N \rightarrow \hat{p}_N$

Important (The general parameter estimation problem)

Given:

- ***measured data***: $D[1,N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$
- ***predictive parametrized model*** $\hat{y}(k|p) = g(k, D[1, k-1]; p)$ *generating the **prediction error series** (discrete time signal):* $\varepsilon(k,p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$
- *norm of the prediction error (**objective/loss function**):* $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \ell(\varepsilon(k,p))$ *where* $\ell(\cdot)$ *is a positive scalar-valued function; most frequently:* $\ell(\varepsilon) = \frac{1}{2}\varepsilon^2$

Minimizing the prediction error

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Parameter estimation method is a mapping: $D^N \to \hat{p}_N$

Important (The general parameter estimation problem)

Given:

- ***measured data***: $D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$
- ***predictive parametrized model*** $\hat{y}(k|p) = g(k, D[1, k-1]; p)$ *generating the **prediction error series** (discrete time signal):* $\varepsilon(k, p) = y(k) - \hat{y}(k|p) \qquad k = 1, \dots, N$
- *norm of the prediction error (**objective/loss function**):* $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \ell(\varepsilon(k, p))$ *where $\ell(\cdot)$ is a positive scalar-valued function; most frequently:* $\ell(\varepsilon) = \frac{1}{2}\varepsilon^2$

From the known D^N measurements and the p parameter vector we can compute the value of the $V_N(p, D^N)$ objective/loss function, that is minimized by the estimated $\hat{p}_N$ parameter vector.

Example: SISO ARX models

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Example: SISO ARX models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Example: SISO ARX models

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Parameter vector:

$$p = [-a_1 \ \ -a_2 \ \ \ldots \ \ -a_n \ \ b_0 \ b_1 \ \ \ldots \ \ b_m]^\top$$

Example: SISO ARX models

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Parameter vector:

$$p = [-a_1 \ \ -a_2 \ \ \ldots \ \ -a_n \ \ b_0 \ b_1 \ \ \ldots \ \ b_m]^\top$$

*Prediction error (**white noise!**):*

$$\varepsilon(k) = \hat{y}(k|p) - y(k) = e(k)$$

Overview

- Discrete time LTI stochastic input-output models
- Minimizing the prediction error
- The least squares estimate
  - Predictive models linear in parameters
  - LS estimation of ARX model parameters
- Properties of the dynamic least squares estimate

Parameter estimation of predictive models by linear regression

In the case of models linear-in-parameters :

$$\hat{y}(k|p) = p^\top \varphi(k) = \varphi(k)^\top p$$

where $\varphi(\cdot)$ is the so-called regressor , containing the measured data; p is the vector of model parameters to be estimated.

Parameter estimation of predictive models by linear regression

In the case of models linear-in-parameters :

$$\hat{y}(k|p) = p^\top \varphi(k) = \varphi(k)^\top p$$

where $\varphi(\cdot)$ is the so-called regressor , containing the measured data; p is the vector of model parameters to be estimated.

***Prediction error**:*

$$\varepsilon(k,p) = y(k) - p^\top \varphi(k)$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Parameter estimation of predictive models by linear regression

In the case of models linear-in-parameters :

$$\hat{y}(k|p) = p^\top \varphi(k) = \varphi(k)^\top p$$

where $\varphi(\cdot)$ is the so-called regressor , containing the measured data; p is the vector of model parameters to be estimated.

***Prediction error**:*

$$\varepsilon(k,p) = y(k) - p^\top \varphi(k)$$

***Objective/loss function** to be minimized: sum of squares (**L**east **S**quares)*

$$V_N(p, D^N) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}\left[y(k) - p^\top \varphi(k)\right]^2$$

LS estimate for models linear-in-parameters

Taking the partial derivatives w.r.t. the elements of the parameter vector:

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\left[y(k)-\varphi^{\top}(k)\cdot p\right]=0$$

LS estimate for models linear-in-parameters

Taking the partial derivatives w.r.t. the elements of the parameter vector:

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\left[y(k)-\varphi^{\top}(k)\cdot p\right]=0$$

We solve the above equation for p

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)=\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\cdot p$$

LS estimate for models linear-in-parameters

Taking the partial derivatives w.r.t. the elements of the parameter vector:

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\left[y(k)-\varphi^{\top}(k)\cdot p\right]=0$$

We solve the above equation for p

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)=\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\cdot p$$

Important (LS estimate)

$$\hat{p}_{LS}=\left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

Example: LS estimate of ARX model parameters

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Example: LS estimate of ARX model parameters

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Example: LS estimate of ARX model parameters

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Parameter vector:

$$p = [-a_1 \ \ -a_2 \ \ldots \ -a_n \ \ b_0 \ b_1 \ \ldots \ b_m]^\top$$

Example: LS estimate of ARX model parameters

ARX model is the basic case: *the output noise is white*

$$A^*(q^{-1}) \cdot y(k) = B^*(q^{-1}) \cdot u(k) + e(k)$$

Predictive form of the model:

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

Parameter vector:

$$p = [-a_1 \;\; -a_2 \;\; \ldots \;\; -a_n \;\; b_0 \; b_1 \;\; \ldots \;\; b_m]^\top$$

The regressor:

$$\varphi(k) = [y(k-1) \; y(k-2) \;\; \ldots \;\; y(k-n) \;\; u(k) \; u(k-1) \;\; \ldots \;\; u(k-m)]^\top$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Discrete time LTI stochastic input-output models
- Minimizing the prediction error
- The least squares estimate
- Properties of the dynamic least squares estimate
  - Asymptotic behavior of the LS estimate

Dynamic LS estimate Asymptotic properties – 1

Difference from standard linear regression: the measured outputs appear in the regression vector $\varphi(k)$ $\Longrightarrow$ the measured values $y(k)$ may contain not only independent white noise errors compared to the deterministic case even for ARX models.

Dynamic LS estimate
Asymptotic properties – 1

Difference from standard linear regression: the measured outputs appear in the regression vector $\varphi(k)$ $\Longrightarrow$ the measured values $y(k)$ may contain not only independent white noise errors compared to the deterministic case even for ARX models.

Important (Asymptotic properties)

Asymptotic properties of the estimate hold in the limit when the time k goes to infinity.

Dynamic LS estimate Asymptotic properties – 1

Difference from standard linear regression: the measured outputs appear in the regression vector $\varphi(k)$ $\Longrightarrow$ the measured values $y(k)$ may contain not only independent white noise errors compared to the deterministic case even for ARX models.

Important (Asymptotic properties)

Asymptotic properties of the estimate hold in the limit when the time k goes to infinity.

Model for analysing the asymptotic behaviour of the estimate
The system can be described as

$$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$$

*with $\{\nu_0(k)\}$ error series, p_0 is the so-called **nominal value** or "true" value of the parameter.*

Dynamic LS estimate Asymptotic properties – 2

Important (LS estimate and notation)

$$\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\cdot\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)\ ,\ R(N) = \frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)$$

Dynamic LS estimate Asymptotic properties – 2

Important (LS estimate and notation)

$$\hat{p}_{LS} = \left[\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k) \right]^{-1} \cdot \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k) \, , \; R(N) = \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k)$$

Important (Estimation error)

$$\hat{p}_{LS}(N) = [R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \left[\varphi(k)^\top \cdot p_0 + \nu_0(k) \right]$$

$$\hat{p}_{LS}(N) = p_0 + [R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

The estimation error is the second term in the above equation.

Dynamic LS estimate
Asymptotic unbiasedness – 1

Estimation error

$$[R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\nu_0(k)$$

We would like:

- to have this term as “small” as possible, since in that case the estimated parameter will be close to p_0,

.

Dynamic LS estimate
Asymptotic unbiasedness – 1

Estimation error

$$[R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\nu_0(k)$$

We would like:

- to have this term as “small” as possible, since in that case the estimated parameter will be close to p_0,
- that this term converges to 0 as the sample size is growing, i.e. $N\to\infty$

.

Dynamic LS estimate Asymptotic unbiasedness – 1

Estimation error

$$[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

We would like:

- to have this term as “small” as possible, since in that case the estimated parameter will be close to p_0,
- that this term converges to 0 as the sample size is growing, i.e. $N \to \infty$

Important (Asymptotic unbiasedness)

The behaviour of an estimate when the sample size is growing is called the asymptotic behaviour of the estimate. We are talking e.g. about asymptotic unbiasedness in this sense.

.

Stochastic properties of predictive models

$$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$$

When the $\nu_0(k)$ error is small compared to the regressor $\varphi(k)$ containing measured values, then the estimation error

$$[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

will also be small.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

$$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$$

When the $\nu_0(k)$ error is small compared to the regressor $\varphi(k)$ containing measured values, then the estimation error

$$[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

will also be small.

Important

If both the input $(u(k) \quad k = 1, 2, \ldots)$ and the error $(\nu_0(k) \quad k = 1, 2, \ldots)$ are stationary stochastic processes in an AR(MA)X model, then the output $(y(k) \quad k = 1, 2, \ldots)$ will also be a stationary process.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION

COMPUTER LABORATORY 3

Created by: Anna Pózna

Overview

- Dynamic models
  - Recall
  - Input-output models in MATLAB
  - ARX models in MATLAB
- LS estimation of ARX models
  - Recall
  - LS estimation of ARX models in MATLAB
  - SISO models
  - MIMO models

- Dynamic models
  - Recall
  - Input-output models in MATLAB
  - ARX models in MATLAB
- LS estimation of ARX models
  - Recall
  - LS estimation of ARX models in MATLAB
  - SISO models
  - MIMO models

Recall - Dynamic models of DT systems

- DT-LTI stochastic SISO I/O model
  - Backward difference form:

$$y(k)+a_1 y(k-1)+\ldots+a_n y(k-n) = b_0 u(k-d)+\ldots+b_m u(k-d-m)$$

  - y: output
  - u: input
  - k: discrete time instance
  - $a_1,\ldots,a_n,b_0,\ldots,b_m$: parameters
  - $d = n - m$: time delay
- Compact form

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k-d)$$

Recall - Dynamic models of DT systems

- General form - ARMAX process

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + C^*(q^{-1})e(k)$$

  - $A^*(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$
  - $B^*(q^{-1}) = b_0 q^{-d} + b_1 q^{-d-1} + \ldots + b_m q^{-d-m}$
  - $C^*(q^{-1}) = c_0 + c_1 q^{-1} + \ldots + c_n q^{-n}$
  - q^{-1}: time shift operator
  - $e(k)$: white noise process

Recall - Dynamic models of DT systems

- ARX (AutoRegressive with eXogenous input) process

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k) + e(k)$$

  - $A^*(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$
  - $B^*(q^{-1}) = b_0 q^{-d} + b_1 q^{-d-1} + \ldots + b_m q^{-d-m}$
  - q^{-1}: time shift operator
  - $e(k)$: white noise process

I/O models in MATLAB

- I/O models can be represented by an `idpoly` object in MATLAB
- part of the System Identification Toolbox
- The general form of the underlying model

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

- $A(q^{-1}) = 1 + a_1 q^{-1} + \ldots + a_n q^{-n}$
- $B(q^{-1}) = b_0 + b_1 q^{-1} + \ldots + b_m q^{-m}$
- $C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_l q^{-l}$

- syntax: `idpoly(A,B,C,D,F,NoiseVariance,Ts)`
- `NoiseVariance` is the variance or covariance matrix of $e(k)$
- `Ts` is the sampling time
- You can define the A, B, C, D, F polynomials with their coefficients
- Default values
  - $B = []$
  - $C = 1$
  - $D = 1$
  - $F = []$
  - `NoiseVariance=1`, or `eye(n)`
  - `Ts=1`

Example 1

- Consider the following I/O model:

$$y(k) - 2y(k-1) + 1.5y(k-2) = u(k-1) + 3u(k-2) + \\ + e(k) - 0.2e(k-1)$$

- What kind of process is $y(k)$?
- What are the coefficients of the polynomials $A(q^{-1}), B(q^{-1})$ and $C(q^{-1})$?
- Create the I/O model in MATLAB using `idpoly`!

Example 1
Solution

$$y(k) - 2y(k-1) + 1.5y(k-2) = u(k-1) + 3u(k-2) + \\ + e(k) - 0.2e(k-1)$$

- What kind of process is $y(k)$?
  - ARMAX process

$$y(k) - 2y(k-1) + 1.5y(k-2) = u(k-1) + 3u(k-2) + \\ + e(k) - 0.2e(k-1)$$

- What are the coefficients of the polynomials $A(q^{-1}), B(q^{-1})$ and $C(q^{-1})$?
  - $A(q^{-1}) = 1 - 2q^{-1} + 1.5q^{-2} \Rightarrow a_0 = 1,\ a_1 = -2,\ a_2 = 1.5$
  - $B(q^{-1}) = 0 + q^{-1} + 3q^{-2} \Rightarrow b_0 = 0,\ b_1 = 1,\ b_2 = 3$
  - $C(q^{-1}) = 1 - 0.2q^{-1} \Rightarrow c_0 = 1,\ c_1 = -0.2$

Example 1 Solution

$$y(k) - 2y(k-1) + 1.5y(k-2) = u(k-1) + 3u(k-2) + \\ + e(k) - 0.2e(k-1)$$

- Create the I/O model in MATLAB using `idpoly`!

```
>> A=[1 -2 1.5];
>> B=[0 1 3];
>> C=[1 -0.2];
>> sys1=idpoly(A,B,C);
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

```
>> sys1
sys1 =
Discrete-time ARMAX model:
A(z)y(t) = B(z)u(t) + C(z)e(t)
  A(z) = 1 - 2 z^-1 + 1.5 z^-2
  B(z) = z^-1 + 3 z^-2
  C(z) = 1 - 0.2 z^-1
Sample time: unspecified
Parameterization:
   Polynomial orders:   na=2   nb=2   nc=1   nk=1
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for
   parameters and their uncertainties.
Status:
Created by direct construction or transformation.
Not estimated.
```

Example 1
Solution

- `z^-1` is equivalent to `q^-1`
- You can change the variable name:
  `idpoly(A,B,C,'Variable','q^-1')`
- Polynomial orders:
  - na: order of $A(q^{-1})$
  - nb: order of $B(q^{-1}) + 1$
  - nc: order of $C(q^{-1})$
  - nk: delay between the input and the output = the difference between the greatest exponent of $A(q^{-1})$ and $B(q^{-1})$ = number of zeros at the beginning of B

- ARX model:

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

- Only the $A(q^{-1})$ and $B(q^{-1})$ polynomials need to be defined

Example 2

- Create the following ARX model in MATLAB:

$$y(k)-1.5y(k-1)+0.7y(k-2) = u(k-1)+0.5u(k-2)+e(k)$$

- What are the coefficients of the polynomials $A(q^{-1}), B(q^{-1})$?
- What are the orders of the polynomials (na, nb)?

Example 2
Solution

- $y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + e(k)$
- What are the coefficients of the polynomials $A(q^{-1}), B(q^{-1})$?
  $A(q^{-1}) = 1 - 1.5q^{-1} + 0.7q^{-2} \Rightarrow a_0 = 1,\ a_1 = -1.5,\ a_2 = 0.7$
  $B(q^{-1}) = 0 + q^{-1} + 0.5q^{-2} \Rightarrow b_0 = 0,\ b_1 = 1,\ b_2 = 0.5$

Example 2
Solution

- $y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + e(k)$
- What are the orders of the polynomials?
  na=2, nb=2

- $y(k) - 1.5y(k-1) + 0.7y(k-2) = u(k-1) + 0.5u(k-2) + e(k)$
- MATLAB code:

```
>> A = [1 -1.5 0.7];
>> B = [0 1 0.5];
>> sys2=idpoly(A,B);
```

- Dynamic models
  - Recall
  - Input-output models in MATLAB
  - ARX models in MATLAB
- LS estimation of ARX models
  - Recall
  - LS estimation of ARX models in MATLAB
  - SISO models
  - MIMO models

Recall - LS estimation

- Predictive form of ARX models

$$\begin{aligned}\hat{y}(k|p) = & - a_1 y(k-1) - \ldots - a_n y(k-n) + b_0 u(k-d) + \ldots \\ & + b_m u(k-d-m) + e(k) = \\ & = p^T \varphi(k-1) + e(k)\end{aligned}$$

- parameter vector: $p = [-a_1 \ldots - a_n \, b_0 \ldots b_m]^T$
- regressor: $\varphi(k-1) = [y(k-1) \; \ldots \; y(k-n) \; u(k-d) \; \ldots \; u(k-d-m)]^T$
- Solution:

$$\hat{p} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\varphi^T(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)y(k)$$

LS estimation of ARX models in MATLAB

- The `arx` function in MATLAB performs the LS estimation of an ARX or AR models
- syntax:

```
sys=arx(data,[na nb nk]);
```

- **data** is the measured data in `iddata` format
- **na** is the order of the polynomial $A(q^{-1})$
- **nb** is the order of the polynomial $B(q^{-1}) + 1$
- **nk** is the delay between the input and the output

The iddata object

- Syntax: `data = iddata(y,u,Ts)`
- Time domain signal of input output data
- **y**: output signal
- **u**: input signal
- **Ts**: sampling time (default = 1)
- Get input data: `data.u`
- Get output data: `data.y`

Example 3
SISO models

- SISO = Single Input Single Output
- Estimate the parameters of the following SISO ARX model:

$$y(k)+a_1y(k-1)-a_2y(k-2)=b_1u(k-1)+b_2y(k-2)+e(k)$$

- The measurement data can be found in D1 in the `arx_data.mat` workspace

Example 3
Solution

- $y(k)+a_1y(k-1)-a_2y(k-2)=b_1u(k-1)+b_2y(k-2)+e(k)$
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
  $A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2}$
  $B(q^{-1}) = 0 + b_1q^{-1} + b_2q^{-2}$

Example 3
Solution

- $y(k)+a_1y(k-1)-a_2y(k-2)=b_1u(k-1)+b_2y(k-2)+e(k)$
- What is the value of na, nb and nk?
  na=2, nb=2, nk=1

- Matlab code:

```
>> m1=arx(D1,[2 2 1])
m1 =
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 1.513 z^-1 + 0.7083 z^-2
  B(z) = 1.214 z^-1 + 0.3834 z^-2
Sample time: 1 seconds
Parameterization:
    Polynomial orders:   na=2   nb=2   nk=1
    Number of free coefficients: 4
    Use "polydata", "getpvec", "getcov" for parameters
Status:
Estimated using ARX on time domain data "D1".
Fit to estimation data: 89.97% (prediction focus)
FPE: 0.9973, MSE: 0.8844
```

Example 3
Solution

- Get the parameter vector: `getpvec(model)`

```
>> getpvec(m1)
ans =
   -1.5131
    0.7083
    1.2143
    0.3834
```

$p = [a_1\ a_2\ b_1\ b_2]^T$

Example 3 Solution

- Get the coefficients of the A and B polynomials

```
>> m1.A
ans =   1.0000     -1.5131      0.7083
>> m1.B
ans = 0       1.2143       0.3834
```

Example 3
Solution

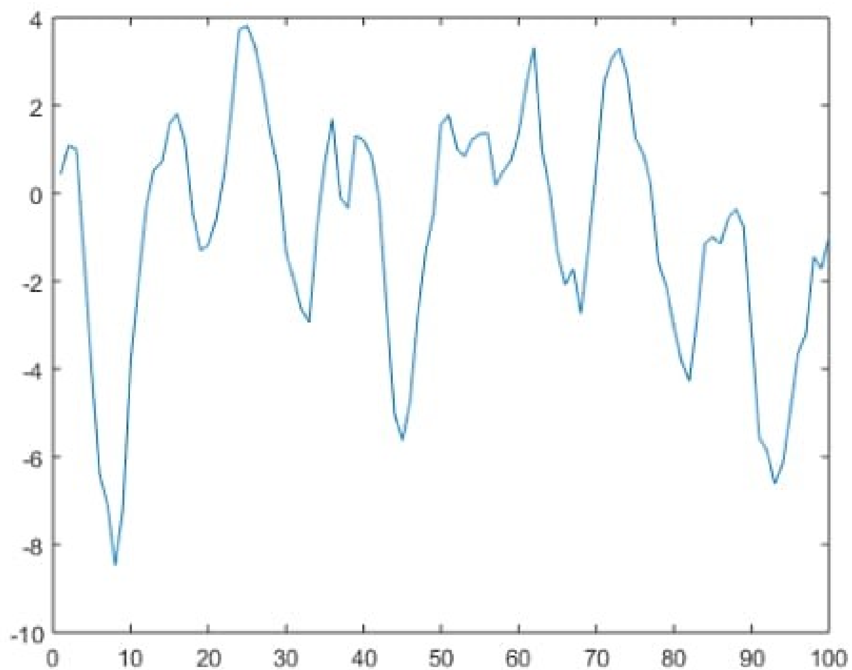
- Get the covariance matrix: `getcov()`

```
>> getcov(m1)
ans =
    0.0018   -0.0015    0.0004    0.0046
   -0.0015    0.0013   -0.0007   -0.0027
    0.0004   -0.0007    0.0300   -0.0252
    0.0046   -0.0027   -0.0252    0.0470
```

Example 3 Solution

- Prediction error:
  - Simulate the estimated model (m1) with the input data

```
>> yest=sim(m1,D1);
>> r=yest.y-D1.y;
>> plot(r)
```



Example 4

- Perform the estimation of the same model using the D2 data! D2 contains more samples than D1.
- Compare the results of the two estimations!

Example 4
Solution

```
>> m2=arx(D2,[2 2 1])
m2 =
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 1.525 z^-1 + 0.7251 z^-2
  B(z) = 0.9985 z^-1 + 0.4544 z^-2
Sample time: 1 seconds
Parameterization:
   Polynomial orders:   na=2   nb=2   nk=1
   Number of free coefficients: 4
   Use "polydata", "getpvec", "getcov"
   for parameters and their uncertainties.
Status:
Estimated using ARX on time domain data "D2".
Fit to estimation data: 88.64% (prediction focus)
FPE: 1.01, MSE: 0.9976
```

Example 4
Solution

- Compare the covariance matrices!

```
>> getcov(m2)
ans =
    0.0002   -0.0001   -0.0000    0.0004
   -0.0001    0.0001   -0.0000   -0.0003
   -0.0000   -0.0000    0.0029   -0.0025
    0.0004   -0.0003   -0.0025    0.0042
>> getcov(m1)
ans =
    0.0018   -0.0015    0.0004    0.0046
   -0.0015    0.0013   -0.0007   -0.0027
    0.0004   -0.0007    0.0300   -0.0252
    0.0046   -0.0027   -0.0252    0.0470
```

- MIMO = Multiple Input Multiple Output
- General model with m input and n output:

$$A_{11}(q^{-1})y_1(k)+\ldots+A_{1n}(q^{-1})y_n(k) = B_{11}(q^{-1})u_1(k)+\ldots+B_{1m}(q^{-1})u_m(k)$$

$$\vdots$$

$$A_{n1}(q^{-1})y_1(k)+\ldots+A_{nn}(q^{-1})y_n(k) = B_{m1}(q^{-1})u_1(k)+\ldots+B_{mm}(q^{-1})u_m(k)$$

- A_{ij} and B_{kl} polynomials form a matrix A and B:

$$A(q^{-1}) = \begin{bmatrix} A_{11}(q^{-1}) & \ldots & A_{1n}(q^{-1}) \\ \vdots & \ddots & \vdots \\ A_{n1}(q^{-1}) & \ldots & A_{nn}(q^{-1}) \end{bmatrix}$$

$$B(q^{-1}) = \begin{bmatrix} B_{11}(q^{-1}) & \ldots & B_{1m}(q^{-1}) \\ \vdots & \ddots & \vdots \\ B_{m1}(q^{-1}) & \ldots & B_{mm}(q^{-1}) \end{bmatrix}$$

MIMO models

- The $A(q^{-1})$ and $B(q^{-1})$ matrices can be represented as *cell arrays* in MATLAB
- A cell array may contain different type of data in its entries e.g. text, numbers, vectors...
- syntax: similar to matrices, but the entries are between $\{$ and $\}$

```
>> C={1 [1 2 3]; 'abc' [4;5]}
C =
      [   1]      [1x3 double]
      'abc'       [2x1 double]
```

Example 5

- Consider the following model:
  $y(k) + 0.5y(k-2) =$
  $u_1(k) + 2u_1(k-1) + 3u_1(k-2) - u_2(k-2) + e(k)$
- How many inputs and outputs does the model have?
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
- Construct the model in MATLAB!

Example 5
Solution

- $y(k) + 0.5y(k-2) =$
  $u_1(k) + 2u_1(k-1) + 3u_1(k-2) - u_2(k-2) + e(k)$
- How many inputs and outputs does the model have?
  2 input and 1 output

Example 5
Solution

- $y(k) + 0.5y(k-2) = u_1(k) + 2u_1(k-1) + 3u_1(k-2) - u_2(k-2) + e(k)$
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
  $A(q^{-1}) = 1 + 0q^{-1} + 0.5q^{-2}$
  $B_{11}(q^{-1}) = 1 + 2q^{-1} + 3q^{-2}$
  $B_{12}(q^{-1}) = 0 + 0u(q^{-1}) - (q^{-2})$

Example 5
Solution

- Construct the model in MATLAB!

```
>> A = [1 0 0.5];
>> B = {[1 2 3] [0 0 -1]};
>> sys3=idpoly(A,B)
sys3 =
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 + 0.5 z^-2
  B1(z) = 1 + 2 z^-1 + 3 z^-2
  B2(z) = -z^-2
Sample time: unspecified
Parameterization:
   Polynomial orders:   na=2   nb=[3 1]   nk=[0 2]
   Number of free coefficients: 6
   Use "polydata", "getpvec", "getcov"
   for parameters and their uncertainties.
Status:
Created by direct construction or transformation. No
```

Example 6

- Estimate the parameters of the following MIMO model:
  $y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) =$
  $b_0^{11} u_1(k) + b_1^{11} u_1(k-1) + b_2^{11} u_1(k-2) - b_2^{12} u_2(k-2) + e(k)$
- The arx function can be used but the na, nb and nk values are vectors or matrices in the MIMO case
- How many inputs and outputs does the model have?
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
- Determine the values of na, nb and nk.
- Estimate the model using the data in D3.

Example 6
Solution

- $y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) = b_0^{11} u_1(k) + b_1^{11} u_1(k-1) + b_2^{11} u_1(k-2) - b_2^{12} u_2(k-2) + e(k)$
- How many inputs and outputs does the model have?
  2 input and 1 output

Example 6
Solution

- $y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) = b_0^{11} u_1(k) + b_1^{11} u_1(k-1) + b_2^{11} u_1(k-2) - b_2^{12} u_2(k-2) + e(k)$
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
  $A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}$
  $B_{11} = b_0^{11} + b_1^{11} q^{-1} + b_2^{11} q^{-2}$
  $B_{12} = 0 + 0q^{-1} + b_2^{12} q^{-2}$

Example 6
Solution

- $y(k) + a_1 y(k-1) + a_2 y(k-2) + a_3 y(k-3) = b_0^{11} u_1(k) + b_1^{11} u_1(k-1) + b_2^{11} u_1(k-2) - b_2^{12} u_2(k-2) + e(k)$
- Determine the values of na, nb and nk.
  $na = 3$, $nb = [3\ 1]$, $nk = [0\ 2]$
- Estimate the model using the data in D3.

```
>> na=3, nb=[3 1], nk=[0 2];
>> m3=arx(D3,[na,nb,nk]);
```

Example 6
Solution


```
>> m3
m3 =
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 1.969 z^-1 + 0.4234 z^-2 + 0.9541 z^-3
  B1(z) = 1.034 + 2.271 z^-1 + 3.001 z^-2
  B2(z) = -1.066 z^-2
Sample time: 1 seconds
Parameterization:
   Polynomial orders:   na=3   nb=[3 1]   nk=[0 2]
   Number of free coefficients: 7
   Use "polydata", "getpvec", "getcov"
   for parameters and their uncertainties.
Status:
Estimated using ARX on time domain data "D3".
Fit to estimation data: 100% (prediction focus)
FPE: 1.217, MSE: 0.9959
```

covariance matrix

```
>> getcov(m3)
ans =
   0.0052   -0.0131    0.0092   -0.0028    0.0088    0.0142   -0.0037
  -0.0131    0.0328   -0.0232    0.0071   -0.0220   -0.0356    0.0092
   0.0092   -0.0232    0.0164   -0.0050    0.0155    0.0252   -0.0065
  -0.0028    0.0071   -0.0050    0.0366   -0.0354   -0.0050    0.0032
   0.0088   -0.0220    0.0155   -0.0354    0.0763   -0.0069   -0.0060
   0.0142   -0.0356    0.0252   -0.0050   -0.0069    0.0742   -0.0125
  -0.0037    0.0092   -0.0065    0.0032   -0.0060   -0.0125    0.0151
```

Example 7

- Estimate the parameters of the following MIMO model:
  $y_1(k) + a_1^{11} y_1(k-1) + a_2^{11} y_1(k-2) + a_1^{12} y_2(k-1) = -b_2^{11} u_1(k-2) + b_1^{12} u_2(k-1) + b_2^{12} u_2(k-2) + e_1(k)$

  $a_1^{21} y_1(k-1) + a_2^{21} y_1(k-2) + y_2(k) + a_1^{22} y_2(k-1) = -b_1^{21} u_1(k-1) + b_1^{22} u_2(k-1) + b_2^{22} u_2(k-2) + e_2(k)$
- How many inputs and outputs does the model have?
- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!
- Determine the values of na, nb and nk.
- Estimate the model using the data in D4.

Example 7
Solution

- $y_1(k) + a_1^{11} y_1(k-1) + a_2^{11} y_1(k-2) + a_1^{12} y_2(k-1) = b_2^{11} u_1(k-2) + b_1^{12} u_2(k-1) + b_2^{12} u_2(k-2) + e_1(k)$

  $a_1^{21} y_1(k-1) + a_2^{21} y_1(k-2) + y_2(k) + a_1^{22} y_2(k-1) = b_1^{21} u_1(k-1) + b_1^{22} u_2(k-1) + b_2^{22} u_2(k-2) + e_2(k)$
- How many inputs and outputs does the model have?
  2 inputs and 2 outputs

- $y_1(k) + a_1^{11} y_1(k-1) + a_2^{11} y_1(k-2) + a_1^{12} y_2(k-1) = b_2^{11} u_1(k-2) + b_1^{12} u_2(k-1) + b_2^{12} u_2(k-2) + e_1(k)$

  $a_1^{21} y_1(k-1) + a_2^{21} y_1(k-2) + y_2(k) + a_1^{22} y_2(k-1) = b_1^{21} u_1(k-1) + b_1^{22} u_2(k-1) + b_2^{22} u_2(k-2) + e_2(k)$

- Give the polynomials $A(q^{-1})$ and $B(q^{-1})$!

  $A_{11}(q^{-1}) = 1 + a_1^{11} q^{-1} + a_2^{11} q^{-2}$
  $A_{12}(q^{-1}) = 0 + a_1^{12} q^{-1}$
  $A_{21}(q^{-1}) = 0 + a_1^{21} q^{-1} + a_2^{21} q^{-2}$
  $A_{22}(q^{-1}) = 1 + a_1^{22} q^{-1}$
  $B_{11}(q^{-1}) = 0 + 0q^{-1} + b_2^{11} q^{-2}$
  $B_{12}(q^{-1}) = 0 + b_1^{12} q^{-1} + b_2^{12} q^{-2}$
  $B_{21}(q^{-1}) = 0 + b_1^{21} q^{-1}$
  $B_{22}(q^{-1}) = 0 + b_1^{22} q^{-1} + b_2^{22} q^{-2}$

Example 7
Solution

- $y_1(k) + a_1^{11} y_1(k-1) + a_2^{11} y_1(k-2) + a_1^{12} y_2(k-1) = b_2^{11} u_1(k-2) + b_1^{12} u_2(k-1) + b_2^{12} u_2(k-2) + e_1(k)$

  $a_1^{21} y_1(k-1) + a_2^{21} y_1(k-2) + y_2(k) + a_1^{22} y_2(k-1) = b_1^{21} u_1(k-1) + b_1^{22} u_2(k-1) + b_2^{22} u_2(k-2) + e_2(k)$
- Determine the values of na, nb and nk.
  na=[2 1; 2 1], nb=[1 2; 1 2], nk= [2 1;1 1]

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Estimate the model using the data in D4.

```
>> na=[2 1;2 1], nb=[1 2;1 2], nk=[2 1; 1 1];
>> m4=arx(D4,[na,nb,nk])
m4 =
Discrete-time ARX model:
  Model for output "y1": A(z)y_1(t) = - A_i(z)y_i(t) + B(z)u(t) + e_1(t)
    A(z) = 1 + 0.9944 z^-1 + 1.185 z^-2
    A_2(z) = 0.5238 z^-1
    B1(z) = -0.8961 z^-2
    B2(z) = 0.4543 z^-1 + 0.4458 z^-2
  Model for output "y2": A(z)y_2(t) = - A_i(z)y_i(t) + B(z)u(t) + e_2(t)
    A(z) = 1 - 0.7224 z^-1
    A_1(z) = -0.1947 z^-1 + 0.8141 z^-2
    B1(z) = -1.188 z^-1
    B2(z) = -0.175 z^-1 + 0.2398 z^-2
Sample time: 1 seconds
Parameterization:
   Polynomial orders:   na=[2 1;2 1]   nb=[1 2;1 2]   nk=[2 1;1 1]
   Number of free coefficients: 12
   Use "polydata", "getpvec", "getcov" for parameters and their uncertainties.
Status:
Estimated using ARX on time domain data "D4".
Fit to estimation data: [100;100]% (prediction focus)
FPE: 1.456, MSE: 2.06
```

Example 7 Solution

- Get the polynomial coefficients of $A_{11}(q^{-1})$ and $A_{12}(q^{-1})$

```
>> m4.A{1,2}
ans =
     1.0000      0.9944      1.1850
>> m4.A{1,2}
ans =
     0      0.5238
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 5

Practical implementation of parameter estimation
Evaluation of the results

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Lecture overview

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Statistical properties of the dynamic LS estimate
  - Conditions for asymptotic unbiasedness
- Preparing and checking measurement data
- Experiment design
  - Sufficient exitation
  - PRBS test signal
- Evaluating the quality of the estimate
  - Analyzing the residuals/prediction errors
  - Analysing the covariances of the estimates
  - Nonlinear case - an example

Overview

- Statistical properties of the dynamic LS estimate
  - Conditions for asymptotic unbiasedness
- Preparing and checking measurement data
- Experiment design
- Evaluating the quality of the estimate

Recall
Dynamic LS estimate

Dynamic predictive model: $y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$

Recall
Dynamic LS estimate

Dynamic predictive model: $y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$

Important (LS estimate)

$$\hat{p}_{LS} = \left[\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k) \right]^{-1} \cdot \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k)$$

Recall
Dynamic LS estimate

Dynamic predictive model: $y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$

Important (LS estimate)

$$\hat{p}_{LS} = \left[\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k) \right]^{-1} \cdot \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k)$$

Important (Estimation error)

$$\hat{p}_{LS}(N) = p_0 + [R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

The estimation error is the second term in the above equation.

Recall
Dynamic LS estimate

Dynamic predictive model: $y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$

Important (LS estimate)

$$\hat{p}_{LS} = \left[\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k) \right]^{-1} \cdot \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k)$$

Important (Estimation error)

$$\hat{p}_{LS}(N) = p_0 + [R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

The estimation error is the second term in the above equation.

(Asymptotic unbiasedness)

is the property of the estimate when the sample size is growing.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

$$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$$

When the $\nu_0(k)$ error is small compared to the regressor $\varphi(k)$ containing measured values, then the estimation error

$$[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

will also be small.

Recall
Stochastic properties

$$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$$

When the $\nu_0(k)$ error is small compared to the regressor $\varphi(k)$ containing measured values, then the estimation error

$$[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$$

will also be small.

Important

If both the input $(u(k) \quad k = 1, 2, \ldots)$ and the error $(\nu_0(k) \quad k = 1, 2, \ldots)$ are stationary stochastic processes in an AR(MA)X model, then the output $(y(k) \quad k = 1, 2, \ldots)$ will also be a stationary process.

Conditions for the analysis of asymptotic behavior

(Assumptions for the analysis)

For the analysis of the asymptotic behavior of the estimation error, let us assume that:

Conditions for the analysis of asymptotic behavior

(Assumptions for the analysis)

For the analysis of the asymptotic behavior of the estimation error, let us assume that:

- *the error $\{\nu_0(k)\}_{k=1}^{N}$ is the realization of a stationary discrete time stochastic process*

Conditions for the analysis of asymptotic behavior

(Assumptions for the analysis)

For the analysis of the asymptotic behavior of the estimation error, let us assume that:

- *the error $\{\nu_0(k)\}_{k=1}^{N}$ is the realization of a stationary discrete time stochastic process*
- *the system itself can be described by an ARX model ,*

Conditions for the analysis of asymptotic behavior

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

(Assumptions for the analysis)

For the analysis of the asymptotic behavior of the estimation error, let us assume that:

- *the error $\{\nu_0(k)\}_{k=1}^{N}$ is the realization of a stationary discrete time stochastic process*
- *the system itself can be described by an ARX model ,*
- *the input $\{u(k)\}_{k=1}^{N}$ is implemented as a stationary discrete time stochastic process .*

Entries of the $R(N)$ matrix

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

$$R(N) = \frac{1}{N}\sum_{k=1}^{N} \varphi(k)\varphi^{\top}(k)$$

where

$$\varphi(k) = [y(k-1)\ y(k-2)\ \dots\ y(k-n)\ \ u(k)\ u(k-1)\ \dots\ u(k-m)]^{\top}$$

(The elements of the $R(N)$ matrix)

$$R(N) = \frac{1}{N}\sum_{k=1}^{N} \left[\begin{array}{ccc|ccc} y(k-1)y(k-1) & \dots & y(k-1)y(k-n) & y(k-1)u(k) & \dots & y(k-1)u(k-m) \\ y(k-2)y(k-1) & \dots & y(k-2)y(k-n) & y(k-2)u(k) & \dots & y(k-2)u(k-m) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y(k-n)y(k-1) & \dots & y(k-n)y(k-n) & y(k-n)u(k) & \dots & y(k-n)u(k-m) \\ --- & - & --- & --- & - & --- \\ u(k)y(k-1) & \dots & u(k)y(k-n) & u(k)u(k) & \dots & u(k)u(k-m) \\ u(k-1)y(k-1) & \dots & u(k-1)y(k-n) & u(k-1)u(k) & \dots & u(k-1)u(k-m) \\ \dots & \dots & \dots & \dots & \dots & \dots \\ u(k-m)y(k-1) & \dots & u(k-m)y(k-n) & u(k-m)u(k) & \dots & u(k-m)u(k-m) \end{array}\right]$$

Asymptotic behavior of the entries of $R(N) - 1$

The regressor $\varphi(\cdot)$ contains only earlier discrete time inputs and outputs in the AR(MA)X case, therefore the entries of $[R(N)]_{ij}$ can be divided into three classes

Asymptotic behavior of the entries of $R(N) - 1$

The regressor $\varphi(\cdot)$ contains only earlier discrete time inputs and outputs in the AR(MA)X case, therefore the entries of $[R(N)]_{ij}$ can be divided into three classes

- *input autocovariance*: autocovariance function $r_{uu}(\tau)$ of the $\{u(k)\}_{k=1}^{N}$ stochastic process

$$\hat{R}_u^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} u(k) \cdot u(k-\tau) \quad \rightarrow \quad R_u(\tau) = r_{uu}(\tau)$$

Asymptotic behavior of the entries of $R(N)^{-1}$

The regressor $\varphi(\cdot)$ contains only earlier discrete time inputs and outputs in the AR(MA)X case, therefore the entries of $[R(N)]_{ij}$ can be divided into three classes

- *input autocovariance*: autocovariance function $r_{uu}(\tau)$ of the $\{u(k)\}_{k=1}^{N}$ stochastic process

$$\hat{R}_u^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} u(k)\cdot u(k-\tau) \quad \rightarrow \quad R_u(\tau) = r_{uu}(\tau)$$

- *output autocovariance*: autocovariance function of the $r_{yy}(\tau)$ of the $\{y(k)\}_{k=1}^{N}$ stochastic process

$$\hat{R}_y^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} y(k)\cdot y(k-\tau) \quad \rightarrow \quad R_y(\tau) = r_{yy}(\tau)$$

Asymptotic behavior of the entries of $R(N)^{-1}$

The regressor $\varphi(\cdot)$ contains only earlier discrete time inputs and outputs in the AR(MA)X case, therefore the entries of $[R(N)]_{ij}$ can be divided into three classes

- *input autocovariance*: autocovariance function $r_{uu}(\tau)$ of the $\{u(k)\}_{k=1}^{N}$ stochastic process

$$\hat{R}_u^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} u(k)\cdot u(k-\tau) \quad \rightarrow \quad R_u(\tau) = r_{uu}(\tau)$$

- *output autocovariance*: autocovariance function of the $r_{yy}(\tau)$ of the $\{y(k)\}_{k=1}^{N}$ stochastic process

$$\hat{R}_y^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} y(k)\cdot y(k-\tau) \quad \rightarrow \quad R_y(\tau) = r_{yy}(\tau)$$

- *input-output covariance*: $r_{yu}(\tau)$ cross-covariance function of the two previous stochastic processes

$$\hat{R}_{yu}^N(\tau) = \frac{1}{N}\sum_{k=1}^{N} y(k)\cdot u(k-\tau) \quad \rightarrow \quad R_{yu}(\tau) = r_{yu}(\tau)$$

Asymptotic behavior of the entries of $R(N)$ – 2

Important

Consequently, the matrix $R(N)$ converges to a constant matrix R^* in the case of large sample size ($N \to \infty$).

(The error term)

The error process $\{\nu_0(k)\}_{k=1}^N$ is stationary, therefore

$$\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\nu_0(k) \quad \to \quad h^*$$

where h^* is a constant vector containing the elements of the cross-covariance functions of $\{u(k)\}_{k=1}^N$ and $\{\nu_0(k)\}_{k=1}^N$, or $\{y(k)\}_{k=1}^N$ and $\{\nu_0(k)\}_{k=1}^N$

Conditions for asymptotic unbiasedness

Assume that the conditions for studying the asymptotic behavior are fulfilled. Then, the estimate is asymptotically unbiased, if:

Conditions for asymptotic unbiasedness

Assume that the conditions for studying the asymptotic behavior are fulfilled. Then, the estimate is asymptotically unbiased, if:

- *matrix R^* is non-singular (**sufficient excitation**)*
  It is fulfilled if the processes $\{u(k)\}_{k=1}^{N}$ and $\{\nu_0(k)\}_{k=1}^{N}$ are independent and the R_{ij} composed of the $R_u(i-j)$ auto-correlations is non-singular (sufficiently exciting inputs).

Conditions for asymptotic unbiasedness

Assume that the conditions for studying the asymptotic behavior are fulfilled. Then, the estimate is asymptotically unbiased, if:

- *matrix R^* is non-singular (**sufficient excitation**)*
  It is fulfilled if the processes $\{u(k)\}_{k=1}^N$ and $\{\nu_0(k)\}_{k=1}^N$ are independent and the R_{ij} composed of the $R_u(i-j)$ auto-correlations is non-singular (sufficiently exciting inputs).
- *$h^* = 0$ It is true if one of the below conditions are fulfilled:*
  - *The $\{\nu_0(k)\}_{k=1}^N$ **error process is a white noise process with zero mean**: there is no modelling error, and the measurement error is white. Then the $\nu_0(k)$ error is independent of the past. Therefore, all the terms in $E\{\varphi(k) \cdot \nu_0(k)\}$ are zero.*
  - *The input $\{u(k)\}_{k=1}^N$ is a white noise process, and the order of the system $n = 0$: the actual output does not depend on past outputs. The regressor $\varphi(k)$ contains only the values of the past inputs, thus $E\{\varphi(k) \cdot \nu_0(k)\} = 0$.*

Properties of LS estimates of predictive models

Important (Asymptotic distribution of the estimate)

If the Conditions for asymptotic unbiasedness are fulfilled, then the distribution of the random variable

$$\sqrt{N} \cdot (\hat{p}_{LS}(N) - p_0)$$

will be a ***multi-dimensional Gaussian distribution*** *with* 0 *mean.*

(Covariance matrix of the estimate)

The covariance matrix in the SISO case is $\lambda_0[R^*]^{-1}$*, where* λ_0 *is the variance of the* $\{\nu_0(k)\}_{k=1}^{N}$ *error.*

SUMMARY: LS estimate of ARX model parameters

Important (Steps of the estimation)

- *1. Collect the data* $D^N = \{(y(k), u(k)), k = 1, .., N\}$ *using a white noise input sequence and form the regressor vectors for* $k = 1, ..., N$

$$\varphi(k) = [y(k-1)\ y(k-2)\ \ldots\ y(k-n)\ \ u(k)\ u(k-1)\ \ldots\ u(k-m)]^\top$$

SUMMARY: LS estimate of ARX model parameters

Important (Steps of the estimation)

- *1. Collect the data* $D^N = \{(y(k), u(k)), k = 1, .., N\}$ *using a white noise input sequence and form the regressor vectors for* $k = 1, ..., N$

$$\varphi(k) = [y(k-1) \; y(k-2) \; \ldots \; y(k-n) \;\; u(k) \; u(k-1) \; \ldots \; u(k-m)]^\top$$

- *2. LS estimate of the parameters* $p = [-a_1 \; -a_2 \; \ldots \; -a_n \;\; b_0 \; b_1 \; \ldots \; b_m]^\top$:

$$\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^\top(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

SUMMARY: LS estimate of ARX model parameters

Important (Steps of the estimation)

- *1. Collect the data* $D^N = \{(y(k), u(k)), k = 1, .., N\}$ *using a white noise input sequence and form the regressor vectors for* $k = 1, ..., N$

$$\varphi(k) = [y(k-1)\ y(k-2)\ \dots\ y(k-n)\ \ u(k)\ u(k-1)\ \dots\ u(k-m)]^\top$$

- *2. LS estimate of the parameters* $p = [-a_1\ \ -a_2\ \dots\ -a_n\ \ b_0\ b_1\ \dots\ b_m]^\top$:

$$\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^\top(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

- *3. From the prediction error sequence* $\varepsilon(k) = y(k) - \hat{y}(k|p) = y(k) - \hat{p}_{LS}^\top\varphi(k)$ *calculate its estimated variance* λ_0

SUMMARY: LS estimate of ARX model parameters

Important (Steps of the estimation)

- *1. Collect the data $D^N = \{(y(k), u(k)), k = 1, .., N\}$ using a white noise input sequence and form the regressor vectors for $k = 1, ..., N$*

$$\varphi(k) = [y(k-1)\ y(k-2)\ \ldots\ y(k-n)\ \ u(k)\ u(k-1)\ \ldots\ u(k-m)]^\top$$

- *2. LS estimate of the parameters $p = [-a_1\ \ -a_2\ \ldots\ -a_n\ \ b_0\ b_1\ \ldots\ b_m]^\top$:*

$$\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^\top(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

- *3. From the prediction error sequence $\varepsilon(k) = y(k) - \hat{y}(k|p) = y(k) - \hat{p}_{LS}^\top\varphi(k)$ calculate its estimated variance λ_0*
- *4. Covariance matrix of the estimated parameters $\hat{p}_{LS}$*

$$\hat{COV}\{\hat{p}_{LS}\} = \lambda_0 \cdot \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^\top(k)\right]^{-1}$$

Overview

- Statistical properties of the dynamic LS estimate
- Preparing and checking measurement data
- Experiment design
- Evaluating the quality of the estimate

Overview of data

With a careful overview of data, we can recognize the following phenomena:

- trends
- outliers
- apparent errors

Overview of data

With a careful overview of data, we can recognize the following phenomena:

- trends
- outliers
- apparent errors

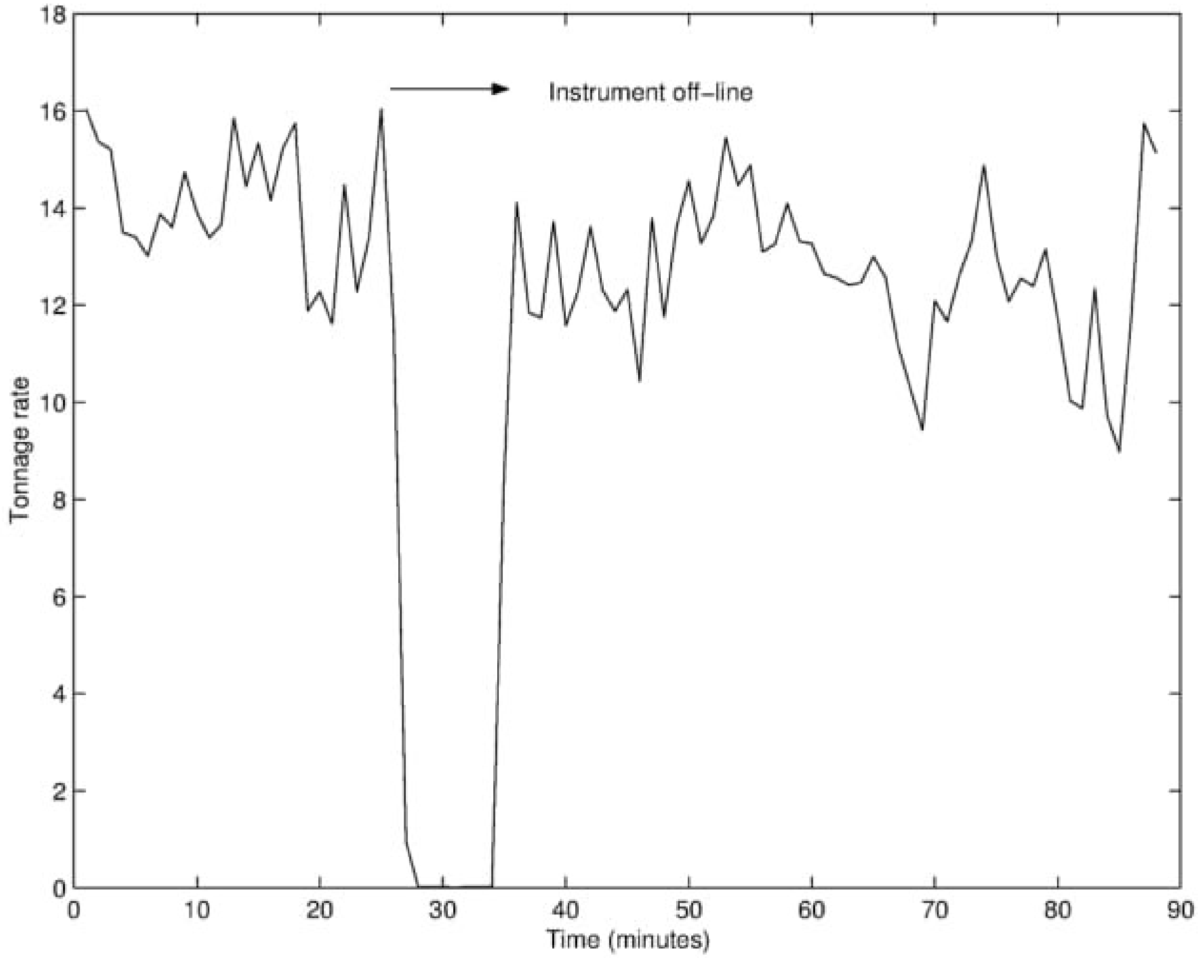
(Visual overview)

For the visual overview, we should plot the data:

- *as a function of time (as data sequences)*
- *as a function of each other*

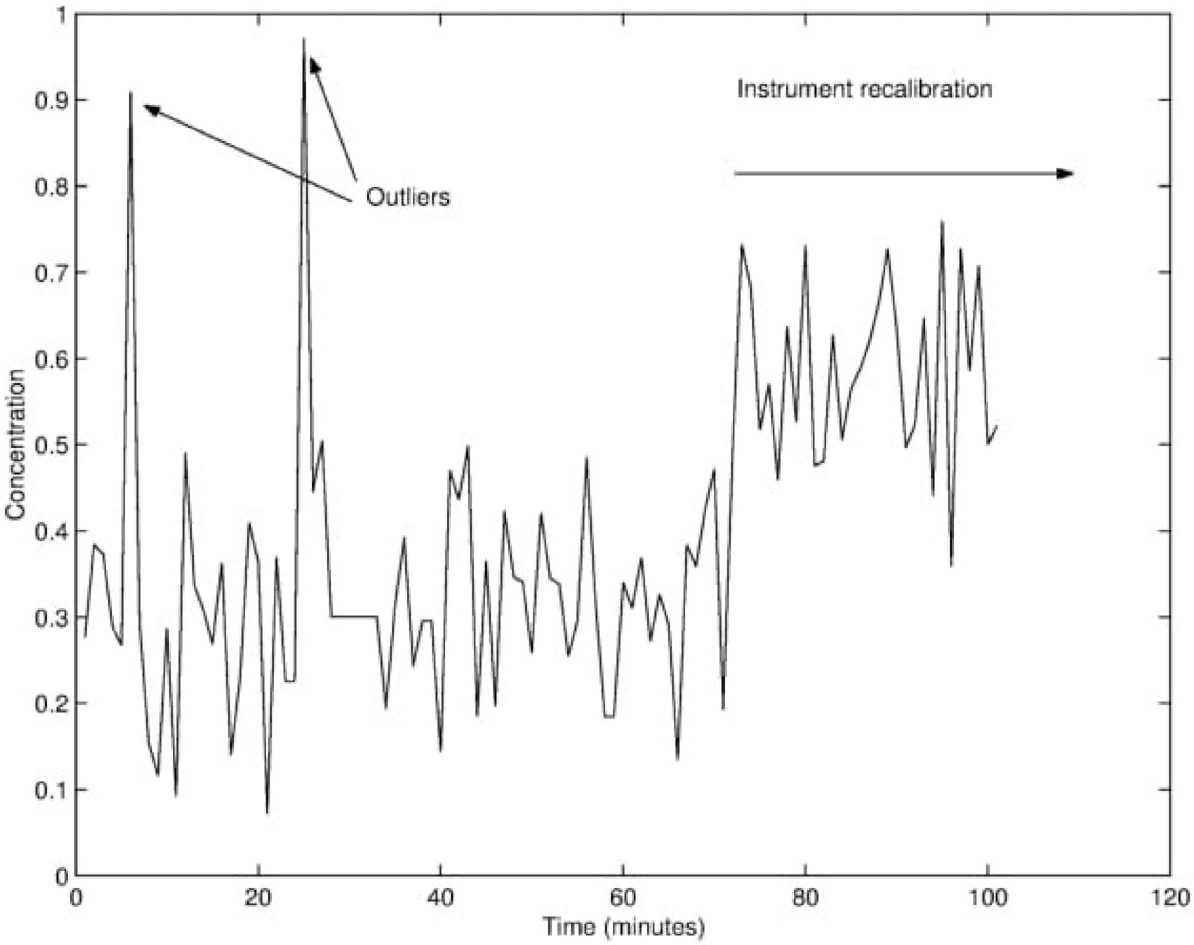
Visual overview
Serious error

Measurement with serious error:



Visual overview
Outliers and gross error

Measurement with outliers and error:



Possible causes of trends

Important (Monitoring of trends and steady states)

Most frequent causes of trends in measured data:

Possible causes of trends

Important (Monitoring of trends and steady states)

Most frequent causes of trends in measured data:

- *fault of sensors that might be indicated by a slow drift-like change*

Possible causes of trends

Important (Monitoring of trends and steady states)

Most frequent causes of trends in measured data:

- *fault of sensors that might be indicated by a slow drift-like change*
- *slow, unmodeled process (ageing, equipment deterioration, equipment becomes dirty) indicated by a drift-like change, too*

Possible causes of trends

Important (Monitoring of trends and steady states)

Most frequent causes of trends in measured data:

- *fault of sensors that might be indicated by a slow drift-like change*
- *slow, unmodeled process (ageing, equipment deterioration, equipment becomes dirty) indicated by a drift-like change, too*
- *slow, usually periodic disturbance: seasonal, weekly or daily variation (e.g. temperature), effects of change of shifts, weekend, different operation of night-shifts etc.*

Outliers

Important

Data generated by Gaussian random variables may (theoretically) contain arbitrarily large or small values with nonzero (but very small) probability.

Outliers

Important

Data generated by Gaussian random variables may (theoretically) contain arbitrarily large or small values with nonzero (but very small) probability.

(Notion of outliers)

In practice, a measured data point can be considered an outlier if its relative magnitude, i.e. $||d(i) - \overline{d}||$ is significantly larger than the deviation of the measurement errors, where $\overline{d}$ is the mean and $||.||$ is a suitable vector norm.

Outliers

Important

Data generated by Gaussian random variables may (theoretically) contain arbitrarily large or small values with nonzero (but very small) probability.

(Notion of outliers)

In practice, a measured data point can be considered an outlier if its relative magnitude, i.e. $||d(i) - \overline{d}||$ is significantly larger than the deviation of the measurement errors, where $\overline{d}$ is the mean and $||.||$ is a suitable vector norm.

How to ***determine****:*
– by simple visual overview of the data
– check normality of data using e.g. χ^2 test

Overview

- Statistical properties of the dynamic LS estimate
- Preparing and checking measurement data
- Experiment design
  - Sufficient exitation
  - PRBS test signal
- Evaluating the quality of the estimate

Experiment design

Aim: to determine the optimal input for parameter estimation

- asymptotic unbiasedness
- minimal variance, uncorrelated elements

Experiment design

Aim: to determine the optimal input for parameter estimation

- asymptotic unbiasedness
- minimal variance, uncorrelated elements

Important (Experiment parameters to be chosen)

- *sampling time*
- *number of samples*
- *test signals for sufficient excitation*

Choosing the sampling time

(Aims)

We should aim at

- *provide sufficiently high frequency sampling for sufficiently long time,*
- *the sample should contain enough information for each important and modelled time constant (pole) of the system*

- *sampling time should be about 1/4 of (or smaller than) the fastest (smallest) time constant*
- *measurement time should be at least 4 times larger than the slowest time constant*

Choosing the the number of samples

The number of measurements needed for parameter estimation depends:

- *on the number of measurements in one record*
- *on the number of repetitions (records)*

Choosing the the number of samples

The number of measurements needed for parameter estimation depends:

- *on the number of measurements in one record*
- *on the number of repetitions (records)*

Important

The overall number of samples should be significantly larger than the order of the system and also much larger than the number of estimated parameters.

Test signals for sufficient excitation

Main considerations:

Test signals for sufficient excitation

Main considerations:

- *Appropriate signal to noise ratio*
  For this, a suitably chosen test-signal is often added to the normal input of the system to ensure sufficient excitation

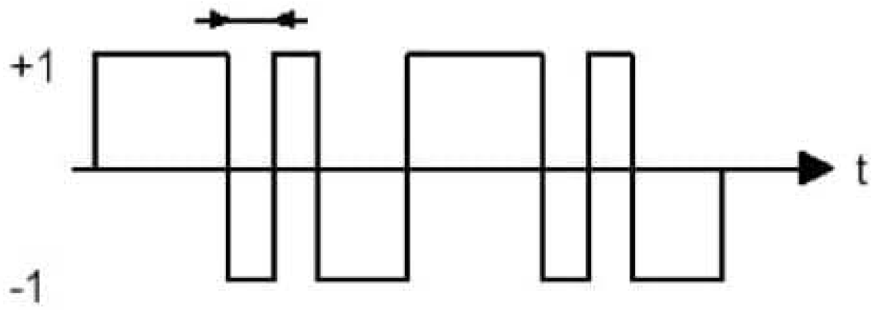
Test signals for sufficient excitation

Main considerations:

- *Appropriate signal to noise ratio*
  For this, a suitably chosen test-signal is often added to the normal input of the system to ensure sufficient excitation
- *Asymptotic unbiasedness*
  The inputs should be independent from the other noises and disturbances. Moreover, it is advantageous if the input is (approximately) white noise.

The PRBS test signal

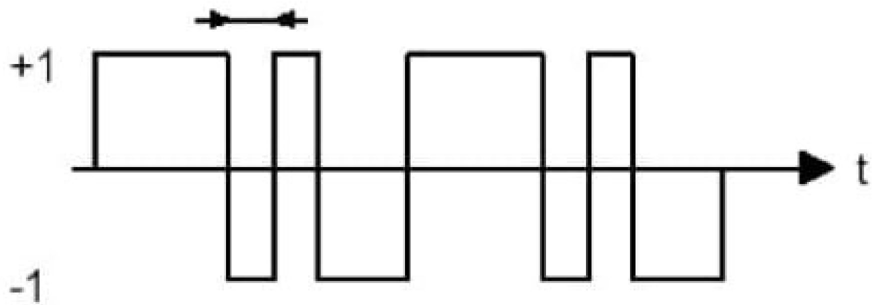
It has only two values, and jumps randomly between them (takes the value +1 with the probability p)



white noise process with binomial distribution

The PRBS test signal

It has only two values, and jumps randomly between them
(takes the value +1 with the probability p)



white noise process with binomial distribution

Important (Sampling time and number of samples)

sampling time should be 1/4 - 1/5 of the smallest time constant
number of samples should be 4-5 times the largest time constant

Overview

- Statistical properties of the dynamic LS estimate
- Preparing and checking measurement data
- Experiment design
- Evaluating the quality of the estimate
  - Analyzing the residuals/prediction errors
  - Analysing the covariances of the estimates
  - Nonlinear case - an example

Quality of the estimate – 1

Analysis of the residuals

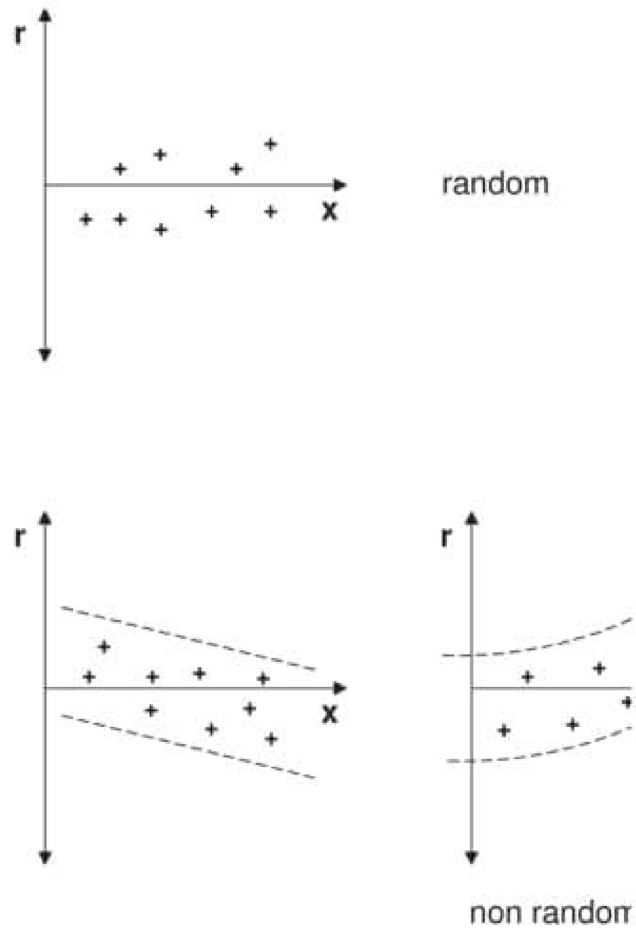
Residual: realization of the prediction error series

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

Important

For an unbiased estimation, the residuals are uncorrelated and have 0 mean.

Residuals



Possible residuals

Detecting trends:

- **fitting a linear function to the data**

Testing the zero mean property

Detecting trends:

- **fitting a linear function to the data**
- standard statistical analysis for data distribution (in the case of independent, identically distributed measurement errors)

Testing the zero mean property

Detecting trends:

- **fitting a linear function to the data**
- standard statistical analysis for data distribution (in the case of independent, identically distributed measurement errors)
- cumulative sum (CUSUM) method (recursive mean):

$$s[k] = \frac{1}{k}\sum_{i=1}^{k} d(i) = \frac{1}{k}\left((k-1)s[k-1] + d(k)\right)$$

The computed $s[k]$ is plotted as a function of time (k), and the trend is monitored. The variance of $s[k]$ decreases with the increase of measurement data.

Recall
Estimating the mean value and the variance

Assume that the underlying random variable ξ has a mean value m and the variance σ^2

- Mean value
  statistics is the *sample mean*
  $$\mu(S) = \frac{1}{n}(\xi_1 + \xi_2 + ... + \xi_n) \ , \ \ \hat{m}(D) = \frac{1}{n}(x_1 + x_2 + ... + x_n)$$
  Property: $E[\mu] = m$
- Variance
  statistics is the *corrected empirical variance*
  $$\theta(S) = \frac{1}{n-1}\left((\xi_1 - \mu)^2 + (\xi_2 - \mu)^2 + ... + (\xi_n - \mu)^2\right)$$
  Property: $E[\theta] = \sigma^2$
- Unbiased estimate
  if the *mean value of the statistics is the real value of the parameter to be estimated*

Recall
Estimation of the covariances

*Consider (scalar valued) random variables ξ_i from the same distribution but **not independent**. They form a "generalized" sample $S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\}$.*

Estimation of the mean value m

- Estimate

$$\hat{m}(D) = \frac{1}{n}(x_1 + ... + x_n)$$

- It may be a biased estimate

Estimation of the auto-covariances $r_{\xi\xi}(s),\ s = 0, 1, ...$

- Estimate for $s << n$

$$\hat{r}(D) = \frac{1}{n-s}((x_1 - \hat{m})(x_{s+1} - \hat{m}) + ... + (x_{n-s} - \hat{m})(x_n - \hat{m}))$$

- It may be a biased estimate

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Important

Variance and correlation: *only for data samples without trends!*

Testing the independence

Important

***Variance and correlation**: only for data samples without trends!*

Consider scalar valued measured data $D[1,k] = D^k$ of k measurements: $d_1, ..., d_k$. They form a "generalized" sample

$$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\} \quad \sim \quad S(D^k) = \{d_1, d_2, ..., d_n\}$$

Testing the independence

Important

Variance and correlation: *only for data samples without trends!*

Consider scalar valued measured data $D[1,k] = D^k$ *of k measurements:* $d_1, ..., d_k$*. They form a "generalized" sample*

$$S(\xi) = \{\xi_1, \xi_2, ..., \xi_n\} \quad \sim \quad S(D^k) = \{d_1, d_2, ..., d_n\}$$

Important

Variance and correlation computation: *from the generalized sample* $S(D^k)$*.*

Quality of the estimate in the parameter space

Analysis of the covariance matrix: *estimate*

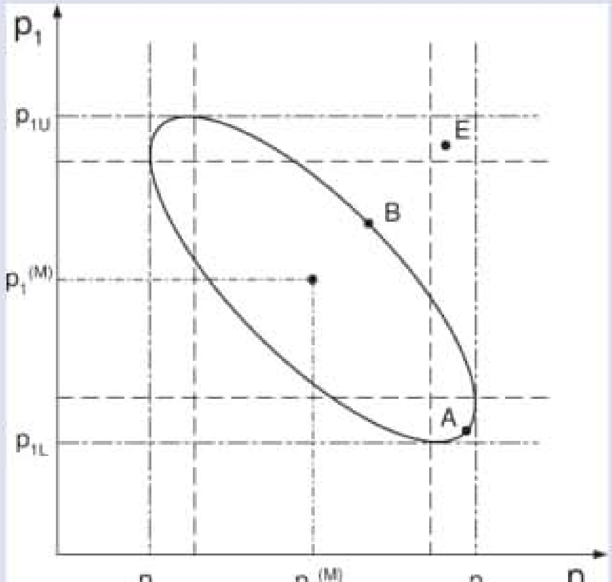
$$\widehat{COV}(\hat{p}_{LS}) = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\varphi^T(k)\right]^{-1}\Delta_\varepsilon$$

Quality of the estimate in the parameter space

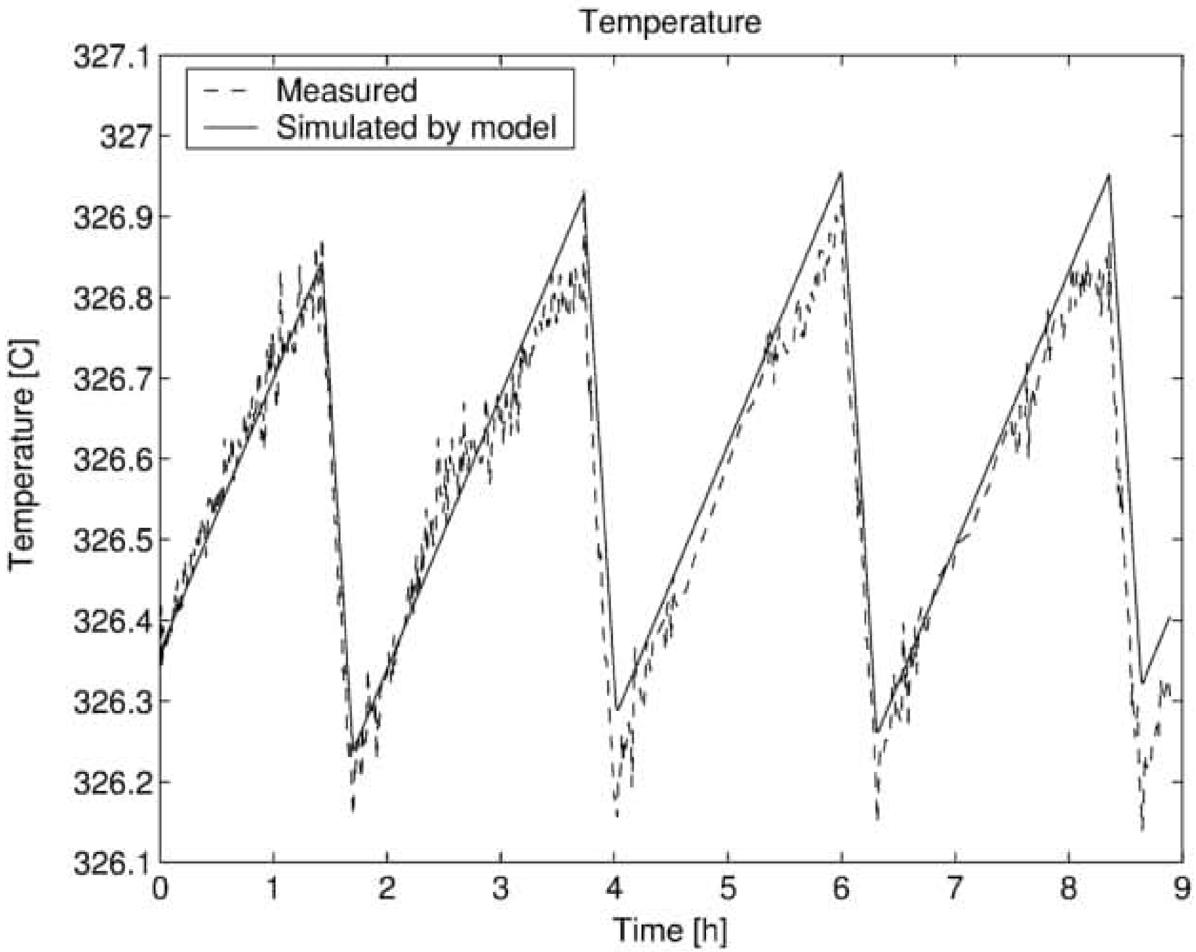
***Analysis of the covariance matrix**: estimate*

$$\widehat{COV}(\hat{p}_{LS}) = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\varphi^T(k)\right]^{-1}\Delta_\varepsilon$$

*For a 'good' estimate, the parameter values are **uncorrelated***

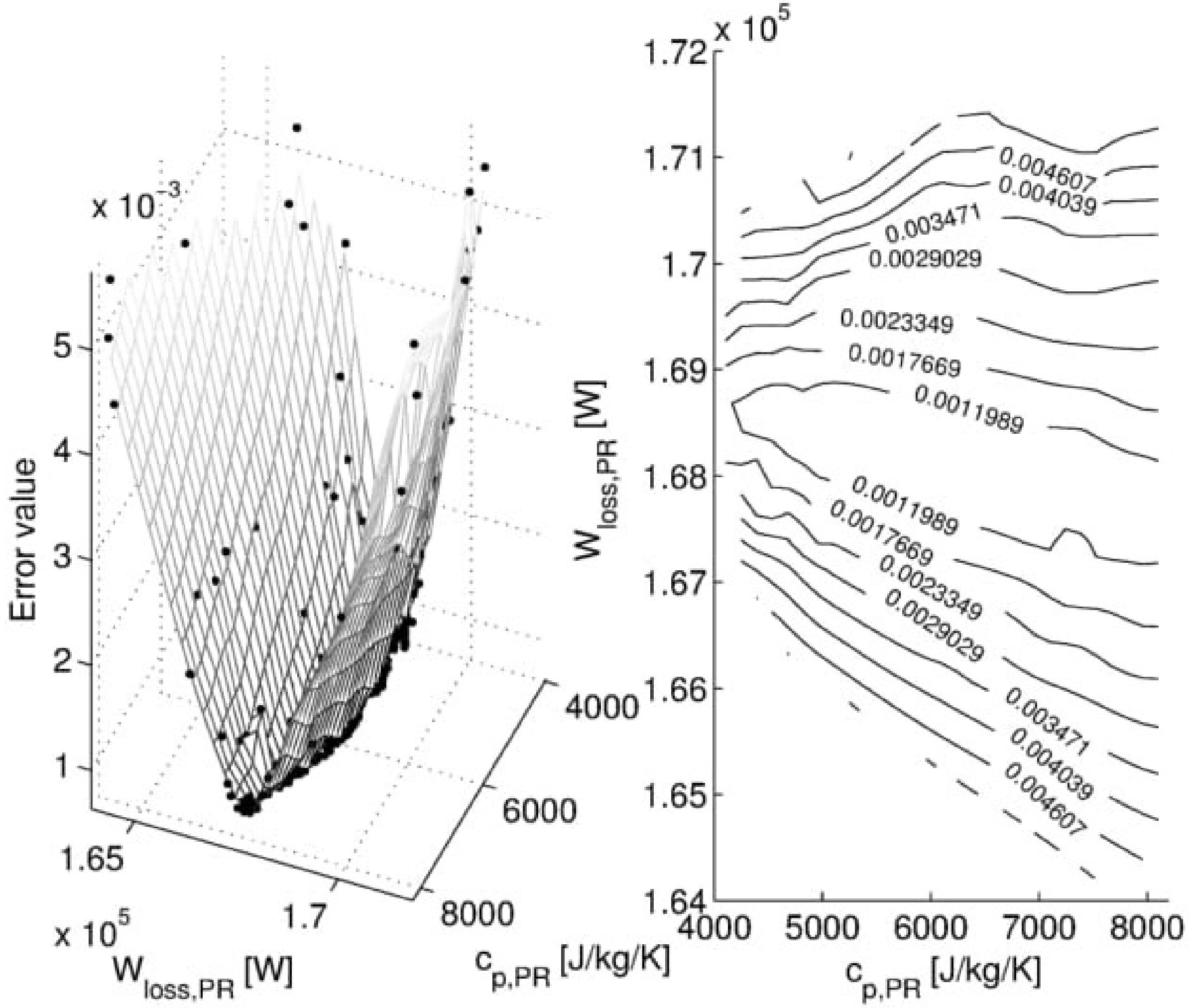


Example: quality of the estimate, prediction error



Measured and model computed (predicted) data

Example: estimated confidence intervals



Level sets of the loss function

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION – 5

Special methods for dynamic LS estimation: Instrumental variable method Nonlinear dynamic case

SZÉCHENYI 2020

Magyarország Kormánya

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Created by: Katalin Hangos

Contents
Lectures and tutorials

- Basic notions, Elements of random variables and mathematical statistics
- The properties of the estimates, Linear regression
- Stochastic processes, Discrete time stochastic dynamic models
- Least squares (LS) estimation by minimizing the prediction error, The properties of the LS estimation
- Special methods for LS estimation of dynamic model parameters: Instrumental variable (IV) method, Parameter estimation of dynamic nonlinear models
- Practical implementation of parameter estimation: Data checking and preparation, Evaluation of the results of parameter estimation

Lecture overview

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Revision of the LS method for ARX models
- The instrumental variable method
  - IV method: basic idea
  - The IV method
  - The IV4 algorithm in Matlab
- Parameter estimation of dynamic nonlinear models
  - Nonlinear models that are linear in parameters
  - Revision of the general parameter estimation task
  - Parameter estimation as nonlinear optimization
  - The gradient method

ARX model: Elements for the LS estimation – 1

***Predictive form** of ARX models:*

$$y(k+1) = -a_1 y(k) - \ldots - a_n y(k-n) + b_0 u(k) + \ldots + \\ + b_m u(k-m) + e(k)$$

ARX model: Elements for the LS estimation – 1

Predictive form *of ARX models:*

$$y(k+1) = -a_1 y(k) - \ldots - a_n y(k-n) + b_0 u(k) + \ldots + \\ + b_m u(k-m) + e(k)$$

Unknown ***parameter vector*** *to be determined:*

$$p := \begin{bmatrix} -a_1 & \ldots & -a_n & b_0 & \ldots & b_m \end{bmatrix}^T$$

ARX model: Elements for the LS estimation – 1

Predictive form *of ARX models:*

$$y(k+1) = -a_1 y(k) - \ldots - a_n y(k-n) + b_0 u(k) + \ldots + \\ + b_m u(k-m) + e(k)$$

Unknown ***parameter vector*** *to be determined:*

$$p := \begin{bmatrix} -a_1 & \ldots & -a_n & b_0 & \ldots & b_m \end{bmatrix}^T$$

based on the ***measured values****:*

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots, N\}$$

ARX model: Elements for the LS estimation – 2

Regressor:

$$\varphi(k) := [\ y(k-1) \quad \dots \quad y(k-n) \quad u(k) \quad \dots \quad u(k-m)\ ]^T$$

Predictive model linear in parameters:

$$\hat{y}(k,p) = \varphi^T(k)p$$

ARX model: Elements for the LS estimation – 2

Regressor:

$$\varphi(k) := [\ y(k-1) \quad \dots \quad y(k-n) \quad u(k) \quad \dots \quad u(k-m)\ ]^T$$

Predictive model linear in parameters:

$$\hat{y}(k,p) = \varphi^T(k)p$$

Prediction error:

$$\varepsilon(k,p) = y(k) - \hat{y}^T(k); \;\; \varepsilon(k,p) = y(k) - \varphi^T(k)p$$

The 2-norm of the error should be ***minimized***:

$$V_N(\theta, D^N) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}(y(k) - \varphi^T(k)p)^2$$

ARX model
The estimation

The optimal parameter value is obtained from $\frac{\partial V_N(p, D^N)}{\partial p} = 0$

$$\hat{p}_N^{LS} = \left[\frac{1}{N} \sum_{i=1}^{N} \varphi(k)\varphi(k)^T \right]^{-1} \frac{1}{N} \sum_{i=1}^{N} \varphi(k)y(k)$$

The optimal parameter value is obtained from $\frac{\partial V_N(p, D^N)}{\partial p} = 0$

$$\hat{p}_N^{LS} = \left[\frac{1}{N} \sum_{i=1}^{N} \varphi(k)\varphi(k)^T \right]^{-1} \frac{1}{N} \sum_{i=1}^{N} \varphi(k)y(k)$$

Important

If the observed data were generated by the real system with p_0*:*

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

then the estimate has the form:

$$\hat{p}_N^{LS} = p_0 + \left[\frac{1}{N} \sum_{i=1}^{N} \varphi(k)\varphi(k)^T \right]^{-1} \frac{1}{N} \sum_{i=1}^{N} \varphi(k)\nu_0(k)$$

ARX model: properties of the LS estimate

The LS estimate should converge to the real p_0 when the data size grows, i.e. when $N \to \infty$. Equivalently:

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \varphi(k)\nu_0(k) = 0$$

i.e. **the observations $\varphi(k)$ and $\nu_0(k)$ noise should be uncorrelated.**

ARX model: properties of the LS estimate

The LS estimate should converge to the real p_0 when the data size grows, i.e. when $N \to \infty$. Equivalently:

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \varphi(k)\nu_0(k) = 0$$

i.e. **the observations $\varphi(k)$ and $\nu_0(k)$ noise should be uncorrelated.**

Important

The LS estimate can be written as (using $\nu_0(k) = y(k) - \varphi^T(k)p_0$):

$$\hat{p}_N^{LS} = sol\left\{ \frac{1}{N} \sum_{i=1}^{N} \varphi(k) \left[y(k) - \varphi^T(k)p_0 \right] = 0 \right\}$$

Overview

- Revision of the LS method for ARX models
- The instrumental variable method
  - IV method: basic idea
  - The IV method
  - The IV4 algorithm in Matlab
- Parameter estimation of dynamic nonlinear models

Instrumental variable (IV) method

Given a dynamic model linear in parameters p_0

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

Use the IV method in the following cases

- Identification of weakly damped or unstable systems
- Correlated measurements and noise
- LS method is not optimal in this case

Instrumental variable (IV) method

Given a dynamic model linear in parameters p_0

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

Use the IV method in the following cases

- Identification of weakly damped or unstable systems
- Correlated measurements and noise
- LS method is not optimal in this case

Important (ARMAX model)

In case of an ARMAX model

$$y(k) = \quad -a_1 y(k\text{-}1) - ... - a_n y(k-n) + b_0 u(k) + ... + b_m u(k-m) + \\ +c_0 e(k) + c_1 e(k-1) + ... + c_n e(k-n)$$

$$y(k) = \varphi^T(k)p_0 + c_0 e(k) + ... + c_n e(k-n)$$

IV method
Basic idea

Given

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

with correlated $\varphi(k)$ and $\nu_0(k)$.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Given

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

with correlated $\varphi(k)$ and $\nu_0(k)$.

Important (Basic idea)

Idea: change $\varphi(k)$ to a suitably chosen $\xi(k)$ (the instrumental variable), that is uncorrelated with $\nu_0(k)$. For this, the $y(\cdot)$ part in $\varphi(k)$ should be changed.

$$\varphi(k) := [\ y(k-1)\ \ \dots\ \ y(k-n)\ \ u(k)\ \ \dots\ \ u(k-m)\ ]^T$$

IV method
The estimate

We are searching for an instrumental variable $\xi(t)$ for which:

$$\hat{p}_N^{IV} = sol\left\{\frac{1}{N}\sum_{i=1}^{N}\xi(k)\left[y(k) - \varphi^T(k)\theta_0\right] = 0\right\}$$

IV method
The estimate

We are searching for an instrumental variable $\xi(t)$ for which:

$$\hat{p}_N^{IV} = sol\left\{\frac{1}{N}\sum_{i=1}^{N}\xi(k)\left[y(k) - \varphi^T(k)\theta_0\right] = 0\right\}$$

Important (IV estimate)

$$\hat{p}_N^{IV} = \left[\frac{1}{N}\sum_{k=1}^{N}\xi(k)\varphi(k)^T\right]^{-1}\frac{1}{N}\sum_{i=1}^{N}\xi(k)y(k)$$

IV method
The estimate

We are searching for an instrumental variable $\xi(t)$ for which:

$$\hat{p}_N^{IV} = sol\left\{\frac{1}{N}\sum_{i=1}^{N}\xi(k)\left[y(k) - \varphi^T(k)\theta_0\right] = 0\right\}$$

Important (IV estimate)

$$\hat{p}_N^{IV} = \left[\frac{1}{N}\sum_{k=1}^{N}\xi(k)\varphi(k)^T\right]^{-1}\frac{1}{N}\sum_{i=1}^{N}\xi(k)y(k)$$

For large N, the conditions of convergence of $\hat{p}_N^{IV}$ to the real p_0 are

$$\mathcal{E}\left\{\xi(k)\varphi^T(k)\right\} \text{ is non-singular}$$

$$\mathcal{E}\left\{\xi(k)\nu_0(k)\right\} = 0$$

The selection of the instrumental variable

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The prediction of the output $\hat{y}(k)$ of an ARX model:

$$A(q^{-1})\hat{y}(k) = B(q^{-1})u(k)$$

Important (Idea of selection)

The instrumental variables are computed using the ARX model:

$$\xi(k) = [\ -z(k-1) \quad \dots \quad -z(k-n) \quad u(k) \quad \dots \quad u(k-m)\ ]^T$$

where $z(k)$ is generated as the output of a linear system with input u:

$$N(q^{-1})z(k) = M(q^{-1})u(k)$$

where $N(q^{-1})$ and $M(q^{-1})$ define a stable filter , too.

The selection of the instrumental variable

The prediction of the output $\hat{y}(k)$ of an ARX model:

$$A(q^{-1})\hat{y}(k) = B(q^{-1})u(k)$$

Important (Idea of selection)

The instrumental variables are computed using the ARX model:

$$\xi(k) = [\ -z(k-1) \quad \dots \quad -z(k-n) \quad u(k) \quad \dots \quad u(k-m)\ ]^T$$

where $z(k)$ is generated as the output of a linear system with input u:

$$N(q^{-1})z(k) = M(q^{-1})u(k)$$

where $N(q^{-1})$ and $M(q^{-1})$ define a stable filter, too.

HOW to choose the filter parameters $N(q^{-1})$ and $M(q^{-1})$?

The principle of the IV method

The simplest selection of $N(q^{-1})$ *and* $M(q^{-1})$ *are given by an ordinary LS estimation (pre-estimation step), and*
$K_u(q^{-1}, p_{pre}) = \frac{\hat{A}(q^{-1})}{\hat{B}(q^{-1})}$.

The principle of the IV method

The simplest selection of $N(q^{-1})$ and $M(q^{-1})$ are given by an ordinary LS estimation (pre-estimation step), and $K_u(q^{-1}, p_{pre}) = \frac{\hat{A}(q^{-1})}{\hat{B}(q^{-1})}$.

Important (principle of the IV method)

- *Computing the instrumental variables*

$$\begin{aligned} z(k) &= K_u(q^{-1}, p_{pre})u(k) \\ \xi(k) &= [z(k-1) \ \dots \ z(k-n) \ \ u(k) \ \dots \ u(k-m)]^T \\ \varepsilon_F(k,p) &= y(k) - \varphi^T(k,p)p \end{aligned}$$

The principle of the IV method

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The simplest selection of $N(q^{-1})$ and $M(q^{-1})$ are given by an ordinary LS estimation (pre-estimation step), and $K_u(q^{-1}, p_{pre}) = \frac{\hat{A}(q^{-1})}{\hat{B}(q^{-1})}$.

Important (principle of the IV method)

- *Computing the instrumental variables*

$$
\begin{aligned}
z(k) &= K_u(q^{-1}, p_{pre})u(k) \\
\xi(k) &= [z(k-1) \; ... \; z(k-n) \;\; u(k) \; ... \; u(k-m)]^T \\
\varepsilon_F(k,p) &= y(k) - \varphi^T(k,p)p
\end{aligned}
$$

- *Computing the IV-estimate*

$$
f_N(p, D^N) = \frac{1}{N}\sum_{i=1}^{N} \xi(k)\varepsilon_F(k,p) \; , \; \hat{p}_N^{IV} = sol\left[f_N(p, D^N) = 0\right]
$$

IV 4 algorithm – 1

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

IV 4 algorithm – 1

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

Important (Step 1: Pre-estimation)

The model structure is written in linear regression form. Then the LS estimate of p and the corresponding DT-LTI model are computed:

$$\hat{p}_N^{(1)} = \hat{p}_N^{LS}, \;\; \hat{G}_N^{(1)}(q^{-1}) = \frac{\hat{B}_N^{(1)}(q^{-1})}{\hat{A}_N^{(1)}(q^{-1})}$$

IV 4 algorithm – 2

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

Important (Step 2: IV estimation I.)

The instrumental variables with the pre-estimated $\hat{G}_N^{(1)}(q^{-1})$

$$z^{(1)}(k) = \hat{G}_N^{(1)}(q^{-1})u(k)$$

$$\xi^{(1)}(k) = \begin{bmatrix} z^{(1)}(k-1) & \dots & z^{(1)}(k-n) & u(t) & \dots & u(k-m) \end{bmatrix}^T$$

then compute the corresponding IV estimate and the DT-LTI model:

$$\hat{p}_N^{(2)} = \hat{p}_N^{IV}, \quad \hat{G}_N^{(2)}(q^{-1}) = \frac{\hat{B}_N^{(2)}(q^{-1})}{\hat{A}_N^{(2)}(q^{-1})}$$

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

Important (Step 3: Estimation of the prediction error model)

Compute the prediction error of Step 2:

$$\hat{w}_N^{(2)}(k) := \hat{A}_N^{(2)}(q^{-1})y(k) - \hat{B}_N^{(2)}(q^{-1})u(k)$$

and prescribe an AR model with order $n_a + n_b$:

$$L(q^{-1})\hat{w}_N^{(2)}(k) = e(k)$$

compute an LS estimate for $L(q^{-1})$: $\hat{L}_N(q^{-1})$-t.

IV 4 algorithm – 4.1

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

IV 4 algorithm – 4.1

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

MATLAB implementation in four steps:

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 3.

Important (Step 4: Refined IV estimation using the results of Step 2. and 3.)

The new instrumental variables using $\hat{G}_N^{(2)}(q^{-1})$ from Step 2

$$z^{(2)}(k) = \hat{G}_N^{(2)}(q^{-1})u(k)$$

$$\xi^{(2)}(t) = \hat{L}_N(q^{-1})\left[z^{(2)}(k-1)\ \dots\ z^{(2)}(k-n)\ \ u(k)\ \dots\ u(k-m)\right]^T$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

MATLAB implementation in four steps:

- Step 4: Refined IV estimation using the results of Step 2. and 3.

MATLAB implementation in four steps:

- Step 4: Refined IV estimation using the results of Step 2. and 3.

Important (Step 4: Refined IV estimation using the results of Step 2. and 3.)

The last refining IV estimate is

$$z^{(2)}(k) = \hat{G}_N^{(2)}(q^{-1})u(k)$$
$$\xi^{(2)}(t) = \hat{L}_N(q^{-1})\left[z^{(2)}(k-1)\ \dots\ z^{(2)}(k-n)\ \ u(k)\ \dots\ u(k-m)\right]^T$$

$$\varphi_F(k) = \hat{L}_N(q^{-1})\varphi(k)$$
$$y_F(k) = \hat{L}_N(q^{-1})y(k)$$

$$\hat{p}_N^{(IV)} = \left[\frac{1}{N}\sum_{k=1}^{N}\xi^{(2)}(k)\varphi_F(k)^T\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\xi^{(2)}(k)y_F(k)$$

Overview

- Revision of the LS method for ARX models
- The instrumental variable method
- Parameter estimation of dynamic nonlinear models
  - Nonlinear models that are linear in parameters
  - Revision of the general parameter estimation task
  - Parameter estimation as nonlinear optimization
  - The gradient method

Nonlinear time-invariant single output systems

Time series of measured data:

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$$

Nonlinear time-invariant single output systems

Time series of measured data:

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$$

The general predictive form:

$$\hat{y}(k|p) = g(k, D[1, k-1]; p)$$

Systems that are linear-in-parameters:

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D[1, k-1])$$

Nonlinear time-invariant single output systems

Time series of measured data:

$$D[1,N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$$

The general predictive form:

$$\hat{y}(k|p) = g(k, D[1, k-1]; p)$$

Systems that are linear-in-parameters:

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D[1, k-1])$$

Example: ARX model

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \dots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \dots + b_m \cdot u(k-m)$$

$$p = [-a_1 \ \dots \ -a_n \ \ b_0 \ \dots \ b_m]^\top$$

$$g^*(k, D[1, k-1]) = \varphi(k) = [y(k-1) \ \ \dots \ -y(k-n) \ \ u(k) \ \dots \ u(k-m)]^\top$$

Example: Nonlinear model linear-in-parameters

Nonlinear ARX model that depends linearly on the parameters

$$y(k) = a_1 \cdot y^2(k-1) + b_0 \cdot u^4(k) + e(k)$$

$$p = [a_1 \; b_0]^\top \qquad \hat{y}(k|p) = y(k) - e(k)$$

Example: Nonlinear model linear-in-parameters

Nonlinear ARX model that depends linearly on the parameters

$$y(k) = a_1 \cdot y^2(k-1) + b_0 \cdot u^4(k) + e(k)$$

$$p = [a_1 \ b_0]^\top \qquad \hat{y}(k|p) = y(k) - e(k)$$

Using auxiliary variables:

$$y^2(k-1) = z(k-1) \qquad u^4(k) = w(k)$$

the model can be written as a simple ARX model :

$$\hat{y}(k|p) = a_1 \cdot z(k-1) + b_0 \cdot w(k)$$

$$p = [a_1 \ b_0]^\top \ , \ \varphi(k) = [z(k-1) \ w(k)]^\top$$

Standard LS estimation can be applied ***without guarantee of asymptotic unbiasedness***

Minimizing the prediction error

Method of parameter estimation: $D^N \rightarrow \hat{p}_N$

Minimizing the prediction error

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Method of parameter estimation: $D^N \to \hat{p}_N$

Important (The general task of parameter estimation)

Given

- *measured values:* $D[1,N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots, N\}$
- *parametrized predictive model:* $\hat{y}(k|p) = g(k, D[1, k-1]; p)$ *and the sequence of prediction errors (discrete-time signal):* $\varepsilon(k,p) = y(k) - \hat{y}(k|p) \ , \ k = 1, \ldots, N$
- *norm defined on the prediction error* $V_N(p, D^N) = \frac{1}{N}\sum_{k=1}^{N} \ell(\varepsilon(k,p))$ *where* $\ell(\cdot)$ *is a scalar valued positive function, most often:* $\ell(\varepsilon) = \frac{1}{2}\varepsilon^2$

To be computed: estimated parameter $\hat{p}_N$ *that minimizes* $V_N(p, D^N)$

$$\hat{p}_N = \hat{p}_N(D^N) = \arg\min_p V_N(p, D^N)$$

Method of parameter estimation: $D^N \to \hat{p}_N$

The estimate $\hat{p}_N$ *minimizes* $V_N(p, D^N)$ *:*

$$\hat{p}_N = \hat{p}_N(D^N) = \arg\min_{p} V_N(p, D^N)$$

Method of parameter estimation: $D^N \to \hat{p}_N$

The estimate $\hat{p}_N$ *minimizes* $V_N(p, D^N)$ *:*

$$\hat{p}_N = \hat{p}_N(D^N) = \arg\min_p V_N(p, D^N)$$

- The estimation of nonlinear and nonlinear-in-parameters systems requires the solving of a general parameter estimation problem, that is an **optimization problem**.

Method of parameter estimation: $D^N \to \hat{p}_N$

The estimate $\hat{p}_N$ *minimizes* $V_N(p, D^N)$ *:*

$$\hat{p}_N = \hat{p}_N(D^N) = \arg\min_{p} V_N(p, D^N)$$

- The estimation of nonlinear and nonlinear-in-parameters systems requires the solving of a general parameter estimation problem, that is an **optimization problem**.
- Parameter estimation methods are introduced on the example of single input single output systems here.

Parameter estimation as nonlinear optimization – 1

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

What is known:

- **A sequence of measured values** $D^N = \{(y(k), u(k)) | k = 1, \ldots, N\}$

Parameter estimation as nonlinear optimization – 1

What is known:

- **A sequence of measured values** $D^N = \{(y(k), u(k)) | k = 1, \ldots, N\}$
- **Predictive model**: SISO case

$$\hat{y}(k|p) = g(k, D^{k-1}; p)$$

(Special case: linear-in-parameters

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D^{k-1}; p)$$

with a given vector-valued nonlinear in measured values function $g^*(\cdot)$)

Parameter estimation as nonlinear optimization – 1

What is known:

- **A sequence of measured values** $D^N = \{(y(k), u(k)) | k = 1, \ldots, N\}$
- **Predictive model**: SISO case

  $$\hat{y}(k|p) = g(k, D^{k-1}; p)$$

  (Special case: linear-in-parameters

  $$\hat{y}(k|p) = p^\top \cdot g^*(k, D^{k-1}; p)$$

  with a given vector-valued nonlinear in measured values function $g^*(\cdot)$)
- Using this and the measured values the **prediction error sequence** can be determined

  $$\varepsilon(k, p) = y(k) - \hat{y}(k|p)$$

Parameter estimation as nonlinear optimization – 1

What is known:

- **A sequence of measured values** $D^N = \{(y(k), u(k)) | k = 1, \ldots, N\}$
- **Predictive model**: SISO case

$$\hat{y}(k|p) = g(k, D^{k-1}; p)$$

(Special case: linear-in-parameters

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D^{k-1}; p)$$

with a given vector-valued nonlinear in measured values function $g^*(\cdot)$)

- Using this and the measured values the **prediction error sequence** can be determined

$$\varepsilon(k, p) = y(k) - \hat{y}(k|p)$$

- A **loss function** with a suitable norm $\ell(\cdot) = \frac{1}{2}(\cdot)^2$

$$V_N(p, D^N) = \frac{1}{N}\sum_{k=1}^{N} \ell(\varepsilon(k, p)) = \frac{1}{N}\sum_{k=1}^{N} \frac{1}{2}(y(k) - \hat{y}(k|p))^2$$

Parameter estimation as nonlinear optimization – 2

Goal:

a parameter estimation method that computes an estimated value $\hat{p}_m(D^N)$ from the measured values so that

$$\hat{p}_m(D^N) = \arg\min_p V_N(p, D^N)$$

Parameter estimation as nonlinear optimization – 2

Goal:

a parameter estimation method that computes an estimated value $\hat{p}_m(D^N)$ from the measured values so that

$$\hat{p}_m(D^N) = \arg\min_p V_N(p, D^N)$$

Special case: *the parameters of the nonlinear-in-parameters SISO system with constant parameters can be estimated by applying* ***least squares estimation***.

Loss function:

$$V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} [y(k) - g(k, D^{k-1}; p)]^2$$

Optimization task:

$$\hat{p}_{LKN}(D^N) = \arg\min_p \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} [y(k) - g(k, D^{k-1}; p)]^2$$

Possible solutions of the optimization problem – 1

The objective function of the **optimization problem** in case of given measurements D^N is seemingly quadratic, but in general it is **analytically not solvable** because of the nonlinear function $g(k, D^{k-1}; p) \quad \rightarrow \quad$ there might be **several local minima**

Due to the lack of analytical solutions the optimization problem can be solved by using the following two theoretically different numerical approximation methods.

Possible solutions of the optimization problem – 1

The objective function of the **optimization problem** in case of given measurements D^N is seemingly quadratic, but in general it is **analytically not solvable** because of the nonlinear function $g(k, D^{k-1}; p)$ $\rightarrow$ there might be **several local minima**

Due to the lack of analytical solutions the optimization problem can be solved by using the following two theoretically different numerical approximation methods.

- ***A. Method leading to the solution of a system of nonlinear equations***
  In this case the nonlinear equations

$$\frac{1}{N}\sum_{k=1}^{N}[y(k) - g(k, D^{k-1}; p)]\frac{\partial g}{\partial p_j} = 0 \qquad j = 1, \ldots, N$$

  that define a necessary condition for the optimum needs to be solved using numerical methods.

  *Problem: **the solution is not unique in general**, furthermore the speed of **convergence** and the limit value **highly depends on the initial estimates**.*

Possible solutions of the optimization problem – 2

Due to the lack of analytical solutions the optimization problem can be solved by using the following two theoretically different numerical approximation methods.

- ***B. Direct solution – minimizing the loss function*** $\mathbf{V_N(p, D^N)}$ ***as the function of*** $\mathbf{p}$

  This problem can be ***solved*** *by using numerical optimization procedures, such as the so-called gradient method.*

  In this case as well it is a problem that a ***global optimization method*** *is (would be) required because of the several possible local minimums.*

Important

*The gradient method is a general method for determining extremum (minimum or maximum) values of functions. It is suitable for determining a **local extremum**.*

The gradient method

Important

The gradient method is a general method for determining extremum (minimum or maximum) values of functions. It is suitable for determining a ***local extremum****.*

The method in short*:*
Let $V : \mathbb{R}^m \to \mathbb{R}$ *be the objective function to be minimized.*

- *Gradient:* $\frac{\partial V}{\partial x}$ *(row vector)*
  Let $V_x(x) = \frac{\partial V}{\partial x}^T$ *(gradient transpose, column vector)*

The gradient method

Important

The gradient method is a general method for determining extremum (minimum or maximum) values of functions. It is suitable for determining a ***local extremum****.*

The method in short*:*
Let $V : \mathbb{R}^m \to \mathbb{R}$ *be the objective function to be minimized.*

- *Gradient:* $\frac{\partial V}{\partial x}$ *(row vector)*
  Let $V_x(x) = \frac{\partial V}{\partial x}^T$ *(gradient transpose, column vector)*
- *Step 1: Start from an initial guess* $x_0 \in \mathbb{R}^n$*.*

Important

The gradient method is a general method for determining extremum (minimum or maximum) values of functions. It is suitable for determining a ***local extremum****.*

The method in short*:*
Let $V : \mathbb{R}^m \to \mathbb{R}$ *be the objective function to be minimized.*

- *Gradient:* $\frac{\partial V}{\partial x}$ *(row vector)*
  Let $V_x(x) = \frac{\partial V}{\partial x}^T$ *(gradient transpose, column vector)*
- *Step 1: Start from an initial guess* $x_0 \in \mathbb{R}^n$.
- *Step i: Refine* $x_{i+1} = x_i - \delta_i \cdot V_x(x_i)$ *until convergence, where* $\delta_i \in \mathbb{R}$ *is the step size*

The gradient vector

Let us given a scalar-valued multiple variable nonlinear function

$$V : \mathbb{R}^m \to \mathbb{R}$$

Let us given a scalar-valued multiple variable nonlinear function

$$V : \mathbb{R}^m \to \mathbb{R}$$

*The **gradient** (gradient vector) of V at **some point** $\mathbf{x} \in \mathbb{R}^{\mathbf{m}}$ is the m-dimensional vector*

$$\vec{grad}\ V(x) = [\ \frac{\partial V}{\partial x_1}(x)\ \dots\ \frac{\partial V}{\partial x_m}(x)] \tag{1}$$

where x_i is the ith coordinate of the vector x.
gradient transpose: $V_x(x) = \vec{grad}\ V^T(x)$

Let us given a scalar-valued multiple variable nonlinear function

$$V \; : \; \mathbb{R}^m \to \mathbb{R}$$

*The **gradient** (gradient vector) of V at **some point $\mathbf{x} \in \mathbb{R}^\mathbf{m}$** is the m-dimensional vector*

$$\vec{grad}\; V(x) = [\; \frac{\partial V}{\partial x_1}(x) \;\ldots\; \frac{\partial V}{\partial x_m}(x)] \tag{1}$$

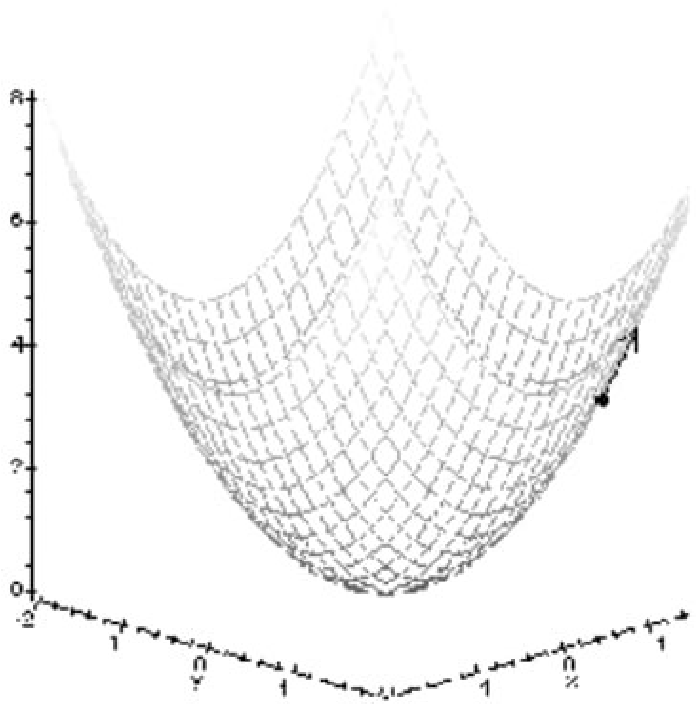
where x_i is the ith coordinate of the vector x.
gradient transpose: $V_x(x) = \vec{grad}\; V^T(x)$

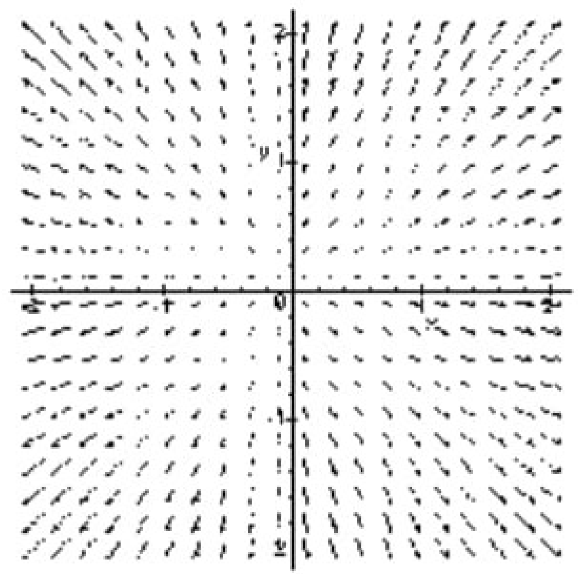
Important

The gradient $\vec{grad}\; V(x)$ in point x shows the direction of the biggest (local) change in the value of the function.

Principle of the gradient method –1

Example: a function with two variables and its gradient vector field





Principle of the gradient method –2

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The curvature of the function V, i.e. its convex or concave property in any point $x^* \in \mathbb{R}^m$ is shown by the so-called ***Hessian matrix*** *$H_V = V_{xx}$, where the entry in row i and column j is*

$$[V_{xx}]_{ij} = \frac{\partial^2 V}{\partial x_i \partial x_j}$$

Principle of the gradient method –2

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The curvature of the function V, i.e. its convex or concave property in any point $x^* \in \mathbb{R}^m$ is shown by the so-called ***Hessian matrix*** *$H_V = V_{xx}$, where the entry in row i and column j is*

$$[V_{xx}]_{ij} = \frac{\partial^2 V}{\partial x_i \partial x_j}$$

Important

The function V ***has a minimum*** *(possibly a local minimum)* ***at the point*** $\mathbf{x}^*$ ***if***

$$\vec{grad}\ V(x^*) = \vec{0} \quad \text{and} \quad V_{xx}(x^*) > 0$$

i.e. the Hessian matrix $V_{xx}(x^*)$ is positive definite.

Example of the gradient and the Hessian matrix

Consider a simple two-variate function
$V(x) = 3(x_1 - 4)^2 + 4(x_2 - 1)^2$

- Compute its gradient vector $\vec{grad}\ V(x)$ and Hessian matrix $H_V = V_{xx}(x)$.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example of the gradient and the Hessian matrix

Consider a simple two-variate function
$V(x) = 3(x_1 - 4)^2 + 4(x_2 - 1)^2$

- Compute its gradient vector $\vec{grad}\ V(x)$ and Hessian matrix $H_V = V_{xx}(x)$.
- Does this V have a minimum? If yes, where?

Example of the gradient and the Hessian matrix

Consider a simple two-variate function
$V(x) = 3(x_1 - 4)^2 + 4(x_2 - 1)^2$

- Compute its gradient vector $\vec{grad}\ V(x)$ and Hessian matrix $H_V = V_{xx}(x)$.
- Does this V have a minimum? If yes, where?

Important

HOMEWORK
Consider another $V'(x) = x_1^2 - 2x_1x_2 + x_2^2$.

- *Compute its gradient vector $\vec{grad}\ V'(x)$ and Hessian matrix $H_V' = V_{xx}'(x)$.*
- *Does this V' have a minimum? If yes, where?*

The steps of the gradient method

The gradient method is an iterative approximation method for determining an extremum of a function with multiple variables. *For the application it is necessary to have*

- *a suitable* ***initial value*** $\mathbf{x_0}$,
- *an* ***accuracy limit*** ε,
- *and a* ***step size*** δ .

The steps of the gradient method

The gradient method is an iterative approximation method for determining an extremum of a function with multiple variables. *For the application it is necessary to have*

- *a suitable* ***initial value*** $\mathbf{x_0}$,
- *an* ***accuracy limit*** ε,
- *and a* ***step size*** δ .

In case of seeking the minimum the main steps of the algorithm *are the following:*

- *Let* $i := 0$ *where* i *is the number of iteration steps, and let* $x_i := x_0$
- *Let us compute the gradient vector* $V_x(x_i)$ *of the loss function in the point* x_i.
- *If the gradient vector is "small enough", i.e. if* $||V_x(x_i)|| < \varepsilon$, *then we have found the minimum and* $x_{min} = x_i$.
- *Otherwise we step once to the direction of the negative gradient, i.e.*

$$x_{i+1} = x_i - \delta V_x(x_i)$$

increase the counter: $i := i + 1$, *then continue from step 2.*

Minimizing the loss function using the gradient method

The algorithm of the gradient method can be applied for minimizing the loss function $V_N(p, D^N)$ according to p as well, using the following assignments:

$$\begin{aligned} \mathbf{V_N(p, D^N)} &\sim V(x) \\ \mathbf{p} &\sim x \end{aligned}$$

Minimizing the loss function using the gradient method

The algorithm of the gradient method can be applied for minimizing the loss function $V_N(p, D^N)$ according to p as well, using the following assignments:

$$\begin{aligned} \mathbf{V_N}(\mathbf{p}, \mathbf{D^N}) \quad &\sim \quad V(x) \\ \mathbf{p} \quad &\sim \quad x \end{aligned}$$

*The required **initial (a priori) data**:*

- *a suitable **initial parameter vector** p_0,*
- *an **accuracy limit** ε ,*
- *a **step size** δ **in the space of parameters**.*

Minimizing the loss function using the gradient method

The algorithm of the gradient method can be applied for minimizing the loss function $V_N(p, D^N)$ according to p as well, using the following assignments:

$$\begin{aligned} \mathbf{V_N(p, D^N)} &\sim V(x) \\ \mathbf{p} &\sim x \end{aligned}$$

*The required **initial (a priori) data**:*

- *a suitable **initial parameter vector** p_0,*
- *an **accuracy limit** ε ,*
- *a **step size** δ **in the space of parameters**.*

*It is important to note that the **gradient method***

- ***is a method for determining local extremum, i.e. the estimated value of the parameter can depend on the compliance of the choice of the initial value,** that is its proximity to the real value,*
- *an iteration step has **polynomial time-complexity** and the **step size** δ **can be suitably modified** by the application of practical algorithms, i.e. it is **decreased near the minimum.***

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

PARAMETER ESTIMATION

COMPUTER LABORATORY 4

Created by: Anna Pózna

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Overview

- Instrumental Variable method
  - Recall
  - The IV 4 algorithm
  - Implementation
  - Example
- Parameter estimation of nonlinear models
  - Recall
  - The gradient method
  - Implementation
  - Examples

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Instrumental Variable method
  - Recall
  - The IV 4 algorithm
  - Implementation
  - Example
- Parameter estimation of nonlinear models
  - Recall
  - The gradient method
  - Implementation
  - Examples

Recall - IV method

- Dynamic model linear in parameters:

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

- Previous assumption:
  - LS estimation
  - uncorrelated measurement and noise
- Problem:
  - **correlated measurement and noise**
  - **LS method is not optimal in this case**
- Solution: **Instrumental Variable (IV)** method

Recall - IV method

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Basic idea
  - Given: $y(k) = \varphi^T(k)p_0 + \nu_0(k)$, $\varphi(k)$ and $\nu_0(k)$ are correlated
    $\varphi(k) = [-y(k-1) \; \ldots \; -y(k-n_a) \; u(k) \; \ldots \; u(k-n_b+1)]^T$
  - Change $\varphi(k)$ to a suitable $\xi(k)$ signal that is uncorrelated with $\nu_0(k)$
  - $\xi(k) = [-z(k-1) \; \ldots \; -z(k-n_a) \; u(k) \; \ldots \; u(k-n_b+1)]^T$
  - $z(k)$ is generated as the output of a linear system with input u:$N(q^{-1})z(k) = M(q^{-1})u(k)$, where $N(q^{-1})$ and $M(q^{-1})$ define a stable filter
  - choosing $N(q^{-1})$ and $M(q^{-1})$ with a LS pre-estimation step

Recall - IV method

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- The IV estimate

$$\hat{p}_N^{IV} = \left[\frac{1}{N} \sum_{k=1}^{N} \xi(k)\varphi(k)^T \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \xi(k)y(k)$$

The IV 4 algorithm in MATLAB

- Step 1: Pre-estimation (to choose $N(q^{-1})$ and $M(q^{-1})$)
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 2 and 3

The IV 4 algorithm in MATLAB

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Step 1: Pre-estimation
  - The model structure is written in linear regression form. LS estimate of p:

$$\hat{p}_N^{(1)} = \hat{p}_n^{LS},\ \hat{G}_N^{(1)}(q^{-1}) = \frac{\hat{B}_N^{(1)}(q^{-1})}{\hat{A}_N^{(1)}(q^{-1})}$$

- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 2 and 3

The IV 4 algorithm in MATLAB

- Step 1: Pre-estimation
- Step 2: IV estimation I.
  - instrumental variables:

$$z^{(1)}(k) = \hat{G}_N^{(1)}(q^{-1})u(k)$$

$$\xi^{(1)}(k) = \left[-z^{(1)}(k-1) \ \dots \ -z^{(1)}(k-n_a)u(k) \ \dots \ u(k-n_b+1)\right]^T$$

  - corresponding IV estimate:

$$\hat{p}_N^{(2)} = \hat{p}_N^{IV}, \ \hat{G}_N^{(2)}(q^{-1}) = \frac{\hat{B}_N^{(2)}(q^{-1})}{\hat{A}_N^{(2)}(q^{-1})}$$

- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 2 and 3

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
  - compute the prediction error of Step 2:

$$\hat{w}_N^{(2)}(k) := \hat{A}_N^{(2)}(q^{-1})y(k) - \hat{B}_N^{(2)}(q^{-1})u(k)$$

  - prescribe an AR model with order $n_a + n_b$

$$L(q^{-1})\hat{w}_N^{(2)}(k) = e(k)$$

  - compute an LS estimate for $L(q^{-1}) : \hat{L}(q^{-1})$
- Step 4: Refined IV estimation using the results of Step 2 and 3

The IV 4 algorithm in MATLAB

- Step 1: Pre-estimation
- Step 2: IV estimation I.
- Step 3: Estimation of the prediction error model
- Step 4: Refined IV estimation using the results of Step 2 and 3
  - new instrumental variables from Step 2:

$$z^{(2)}(k) = \hat{G}_N^{(2)}(q^{-1})u(k)$$

$$\xi^{(2)}(k) = \hat{L}_N(q^{-1})[-z^{(2)}(k-1)\ \dots\ -z^{(2)}(k-n_a)\ u(k)\ \dots$$
$$\dots\ u(k-n_b+1)]^T$$

  - last refining IV estimate using Step 3:

$$\varphi_F(k) = \hat{L}_N(q^{-1})\varphi(k)$$

$$y_F(k) = \hat{L}_N(q^{-1})y(k)$$

$$\hat{p}_N^{IV} = \left[\frac{1}{N}\sum^{N}\xi^{(2)}(k)\varphi_F(k)^T\right]^{-1}\frac{1}{N}\sum^{N}\xi^{(2)}(k)y_F(k)$$

Implementation

- `iv4` function in Matlab
- ARX model estimation using the 4 stage IV method
- syntax: `sys=iv4(data,[na,nb,nk])`
- `sys` is a discrete time `idpoly` object representing the estimated ARX model
- `na` is the order of the polynomial $A(q^{-1})$
- `nb` is the order of the polynomial $B(q^{-1})$+1
- `nk` is the input-output delay

Example 1

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Estimate the parameters p_1 and p_2 using the LS method!
- Estimate the parameters p_1 and p_2 using the IV method!
- The data is in the `data1.txt`, first column contains $y(k)$, second column contains $u(k)$

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Estimate the parameters p_1 and p_2 using the LS method!
  - Import the data from `data1.txt` into the matrix `D1`!
  - Determine the values of `na`, `nb` and `nk`
  - Use the `arx(data,[na,nb,nk])` function

```
>> na=1;
>> nb=2;
>> nk=1;
>> sys1=arx(D1,[na nb nk]);
```

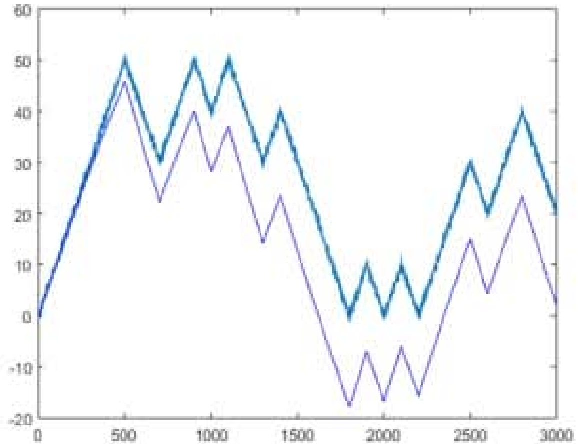
Example 1 - Solution 1

- Compare the output of the estimated model with the measurement data!
- Compute and plot the residuals! Is the output noise white?

Example 1 - Solution 1

- Compare the output of the estimated model with the measurement data!
  - simulate the estimated model with the original input

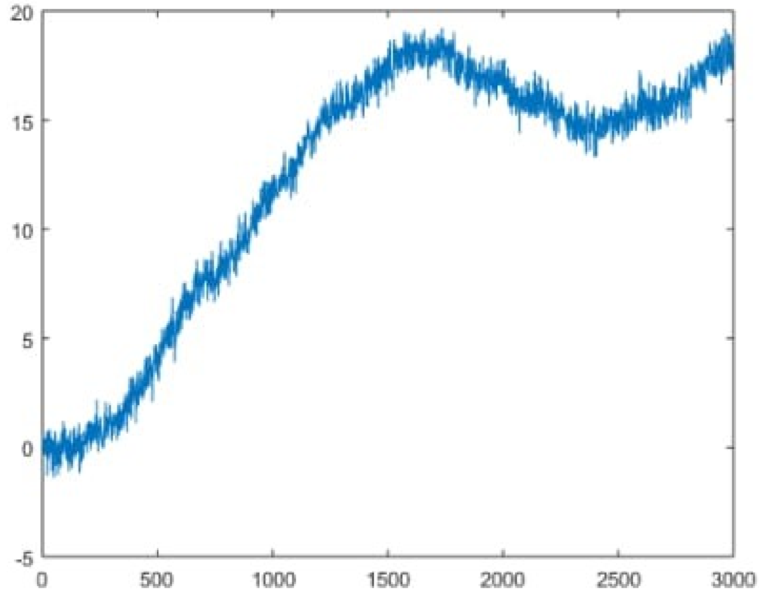
```
>> y_M=sim(sys1,D1(:,2));
>> y=D1(:,1);
>> plot(y);
>> hold on
>> plot(y_M);
```



Example 1 - Solution 1

- Compute and plot the residuals! Is the output noise white?

```
>> res1=y-y_M;
>> plot(res1);
```



Output noise is not white!

Example 1 - Solution 2

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Correlated measurement and the noise
- Estimate the parameters p_1 and p_2 using the IV method!

Example 1 - Solution 2

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Estimate the parameters p_1 and p_2 using the IV method!

```
>> sys_iv=iv4(D,[na,nb,nk]);
```

  - Compare the output of the estimated model with the measurement data!
  - Compute and plot the residuals!

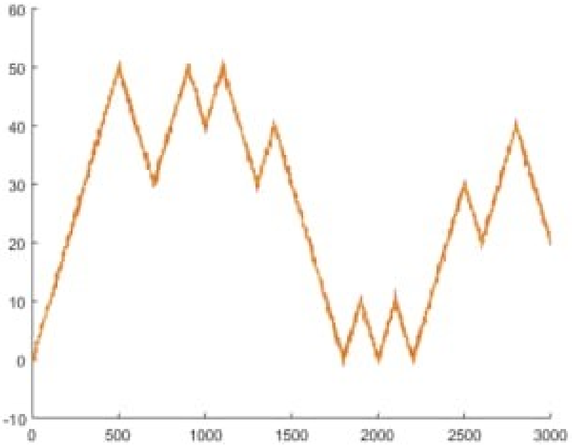
Example 1 - Solution 2

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Estimate the parameters p_1 and p_2 using the IV method!
  - Compare the output of the estimated model with the measurement data!

```
>> y_iv=sim(sys_iv,D(:,2))
>> plot(y);
>> hold on
>> plot(y_iv);
```



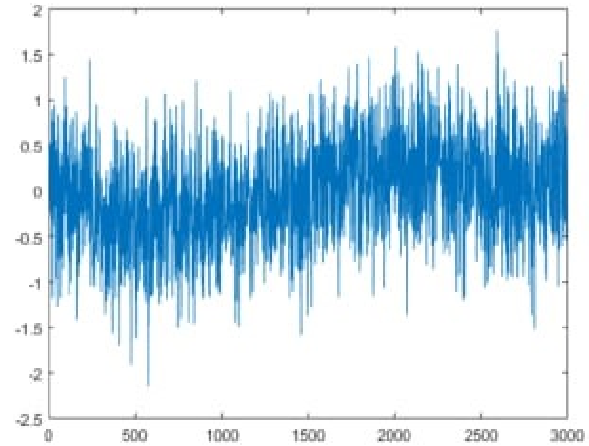
Example 1 - Solution 2

- Consider the following ARX model

$$y(k) = y(k-1) + p_1 u(k-1) + p_2 u(k-2) + \nu(k)$$

- Estimate the parameters p_1 and p_2 using the IV method!
  - Compare the output of the estimated model with the measurement data!
  - Compute and plot the residuals!

```
>> res_iv=y-y_iv
>> plot(res_iv);
```



- Instrumental Variable method
  - Recall
  - The IV 4 algorithm
  - Implementation
  - Example
- Parameter estimation of nonlinear models
  - Recall
  - The gradient method
  - Implementation
  - Examples

Recall - Nonlinear models

- Nonlinear models

$$\hat{y}(k|p) = g(k, D[1, k-1]; p)$$

- Nonlinear models that are linear in parameters

$$\hat{y}(k|p) = p^T \cdot g^*(k, D[1, k-1])$$

  - can be written as a linear model using auxiliary variables
  - e.g. $y(k) = a_1 y^2(k-1) + b_0 u^4(k) + e(k)$
    $p = [a_1\ b_0]^T \qquad y^2(k-1) = z(k-1) \qquad u^4(k) = w(k)$
    $\hat{y}(k|p) = a_1 z(k-1) + b_0 w(k)$

Recall - Parameter estimation of NL models

- Measured values:
  $D[1, N] = D^N = \{(y(k), u(k)) | k = 1, \ldots, N\}$
- Prediction error:
  $\varepsilon(k, p) = y(k) - \hat{y}(k|p)$
- Basic idea: minimizing the prediction error
  - norm of the prediction error: $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \ell(\varepsilon(k, p))$
  - find
    $\hat{p}_N = \hat{p}_D^N = arg\min_p V_N(p, D^N)$
- Nonlinear optimization problem

Recall - Parameter estimation of NL models

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Special case: nonlinear SISO system with constant parameters
- Parameters can be estimated by LS estimation
- Loss function:
  $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2}[y(k) - g(k, D^{k-1}; p)]^2$
- Optimization task:
  $\hat{p}_{LKN}(D^N) = arg\min_p \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2}[y(k) - g(k, D^{k-1}; p)]^2$

Recall - Parameter estimation of NL models

- Possible solutions:
  - solving a system of nonlinear equations
  - minimizing the loss function (e.g. gradient method)

The gradient method

- Iterative approximation method to find an extremum of a function with multiple variables
- Inputs:
  - initial value x_0
  - accuracy limit ε
  - step size δ
- Algorithm to find minimum of a function $V(x)$
  1. Initialization: $i := 0$, $x_i = x_0$
  2. Compute the gradient vector $V_x(x_i) = \vec{grad} V^T(x_i)$ in the point x_i
  3. If $||V_x(x_i)|| < \varepsilon$ then $x_{min} = x_i$
  4. Else $x_{i+1} = x_i - \delta V_x(x_i)$, $i = i + 1$
  5. Continue with step 2.

Implementation of the gradient method

- Create a function which uses the simple gradient method to find the minimum of a function!
- Inputs of the function:
  - function to minimize
  - initial value
  - accuracy limit (tolerance)
  - step size
  - max number of iterations (to avoid too long runtime)
- Outputs of the function
  - minimum of the function
  - function value at the minimum point
  - optional: list of points where the function was evaluated + the function and gradient values
  - optional: iteration number

Steps of implementation

- Define the function
- Initialize the variables
- Create a `while` loop to perform iteration steps
- Update the output variables
- Save the function

Steps of implementation 1

- Define the function

```
function [p,V,p_hist,fun_hist,grad_hist,iter]=
gradient(fun,p0,tol,step,max_iter)
```

  - `p`: estimated parameter
  - `V`: value of the loss function in p
  - `p_hist`: list of the parameter values where the loss function was evaluated
  - `fun_hist`: list of the loss function values during the algorithm
  - `grad_hist`: list of the gradient of the loss function values
  - `iter`: number of iterations at the end of the algorithm
  - `fun`: loss function to evaluate (to be created later)
  - `p0`: initial value
  - `tol`: tolerance, accuracy limit
  - `step`: initial step size
  - `max_iter`: maximum number of iterations

Steps of implementation 2

- Initialize the variables

```
iter=0;
p=p0;
[V,dV]=feval(fun,p);
p_hist=p;
fun_hist=[];
grad_hist=[];
```

  — `[V,dV]=feval(fun,p);` is the evaluation of the function `fun` at the point `t`. The result is the function value `V` and the gradient `dV` of the loss function at the point `p`.

Steps of implementation 2

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Create a `while` loop to perform iteration steps
  - condition: $||\vec{grad}V|| > \varepsilon$ and `iter<max_iter`
  - $||\vec{grad}V||$ can be defined in several ways, e.g. $max|\vec{grad}V|$

```
while (max(abs(dV))>tol)&(iter<max_iter)
        ...
end
```

Steps of implementation 2

- Create a `while` loop to perform iteration steps
  - compute the new value of `p`
    (take a step in the direction of the negative gradient)

```
while(max(abs(dV))>tol)&(iter<max_iter)
        [V,dV]=feval(fun,p);
    p=p-step*dV;
end
```

Steps of implementation 2

- Create a `while` loop to perform iteration steps
  - change the step size to its half or double
    - compute the loss function at the new point with half or double step size
    - choose the new step size where the value of the loss function is smaller

```
while (max(abs(dV))>tol)&(iter<max_iter)
  [V,dV]=feval(fun,p);
  p=p-step*dV;
  if  feval(fun,p-0.5*step*dV)<feval(fun,p-2*step*dV)
      step=0.5*step;
  else
      step=2*step;
  end
end
```

Steps of implementation 2

- Create a `while` loop to perform iteration steps
  - update the vectors `p_hist`, `fun_hist`, `grad_hist` by appending the last computed value to the end of them
  - increase the iteration number

```
while (max(abs(dV))>tol)&(iter<max_iter)
  [V,dV]=feval(fun,p);
  p=p-step*dV;
  if feval(fun,p-0.5*step*dV)<feval(fun,p-2*step*dV)
      step=0.5*step;
  else
      step=2*step;
  end
  p_hist=[p_hist;p];
  fun_hist=[fun_hist;V];
  grad_hist=[grad_hist;dV];
  iter=iter+1;
end
```

Steps of implementation 3

- Update the output variables

```
[V,dV]=feval(fun,p);
fun_hist=[fun_hist;V];
grad_hist=[grad_hist;dV]
```

- Save the function

Example 2

- Estimate the parameter a of the nonlinear model

$$y(k) = e^{-a}y(k-1) - 5u(k-1) + e(k-1)$$

- The data can be found in `data2.txt`. First column: y, second column u.
- The loss function:

$$V(a) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}(y(k) - e^{-a}y(k-1) + 5u(k-1))^2$$

- Derivative of the loss function:

$$\frac{dV}{da} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}2(y(k) - e^{-a}y(k-1) + 5u(k-1))e^{-a}y(k-1)$$

Example 2 - Solution

- Create a global variable for the data

```
global y;
global u;
```

  - it is needed to access the variables in the workspace from the function
- Import the data as column vectors (y,u).
- Create a function which returns the value of the loss function and its derivative!

Example 2 - Solution

- Create a function which returns the value of the loss function and its derivative!

```
function [f,df]=V(a)
global y u;
    N=length(y);
    tmp=0;
tmp2=0;
    for k=2:N;
        tmp=tmp+(y(k)-exp(-a)*y(k-1)+5*u(k-1))^2;
        tmp2=tmp2+2*(y(k)-exp(-a)*y(k-1)+
        +5*u(k-1))*exp(-a)*y(k-1);
    end
    f=1/(2*(N-1))*tmp;   %function
    df=1/(2*(N-1))*tmp2;          %derivative
end
```

- Use the created `gradient` function to find the minimum of the loss function
- e.g. with p0=1, tol=0.05, step=0.1, max_iter=500

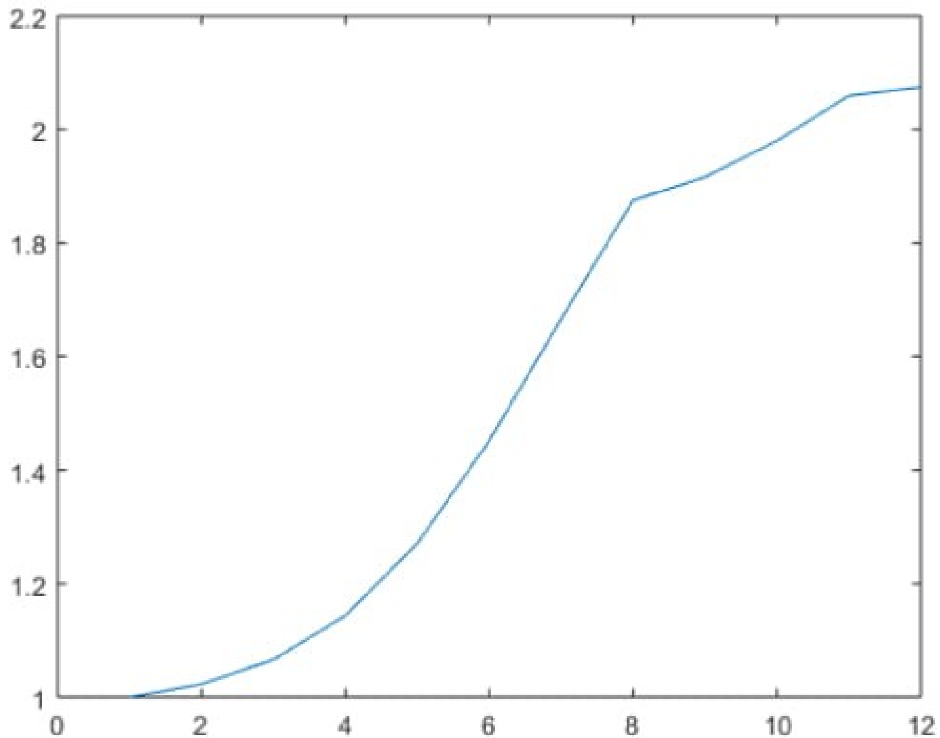
```
[p,Vp,p_hist,V_hist,grad_hist,iter]=
=gradient('V',1,0.05,0.01,500);
>> p
p =
    2.4041
>> Vp
Vp =
    0.4772
```

- estimated parameter: $\hat{p} = 2.4041$
- loss function value: $V(\hat{p}) = 0.4772$

Example 2 - Solution

- The solution progress:
- Plot the parameter values during the iterations

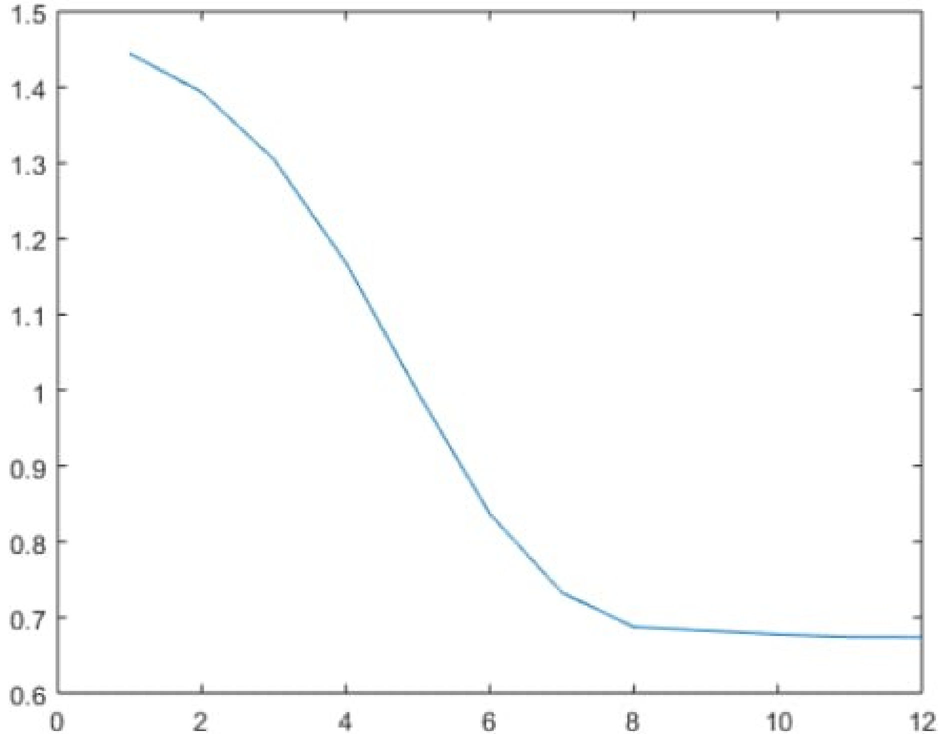
```
>> plot(p_hist)
```



Example 2 - Solution

- The solution progress:
- Plot the loss function values during the iterations

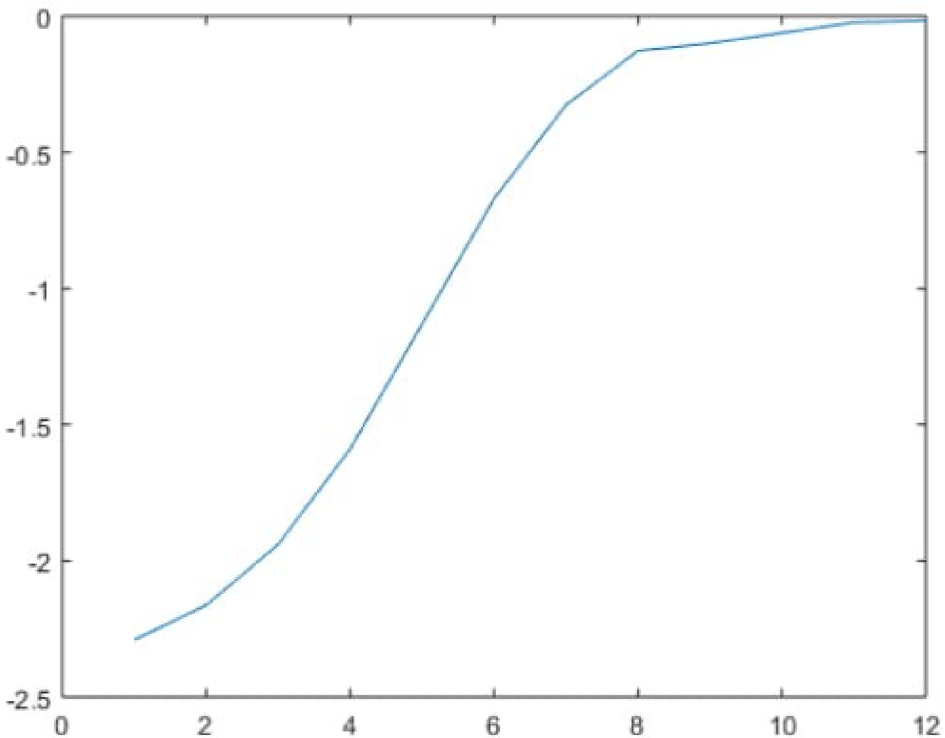
```
>> plot(V_hist)
```



Example 2 - Solution

- The solution progress:
- Plot the function derivative values during the iterations

```
>> plot(grad_hist)
```



Example 3

- Estimate the parameters p_1 and p_2 of the nonlinear model

$$y(k) = p_1^2 0.1 y(k-1) + p_2^2 0.2 y(k-2) + u(k-1) + e(k-1)$$

- The data can be found in `data3.txt`. First column: y, second column u.
- The loss function:

$$V_2(p_1, p_2) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} (y(k) - p_1^2 0.1 y(k-1) - p_2^2 0.2 y(k-2) - u(k-1))^2$$

- Gradient of the loss function:

$$\frac{dV_2}{dp_1} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}2(y(k) - p_1^2 0.1y(k-1) - p_2^2 0.2y(k-2) - $$
$$- u(k-1))(-2p_1 0.1y(k-1))$$

$$\frac{dV_2}{dp_2} = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}2(y(k) - p_1^2 0.1y(k-1) - p_2^2 0.2y(k-2) - $$
$$- u(k-1))(-2p_2 0.2y(k-2))$$

Example 3 - Solution

- Create a global variable for the data

```
global y2;
global u2;
```

  - it is needed to access the variables in the workspace from the function
- Import the data as column vectors (y2,u2).
- Create a function which returns the value of the loss function and its derivative!

Example 3 - Solution

- Create a function which returns the value of the loss function and its gradient!

Example 3 - Solution

```
function [f,df]=V2(P)
global y2 u2;
p1=P(1);
p2=P(2);
N=length(y2);
tmp=0;
tmp2=[0 0];
   for k=3:N;
      tmp=tmp+(y2(k)-p1^2*0.1*y2(k-1)-p2^2*0.2*y2(k-2)-
      -u2(k-1))^2;
      dVp1=2*(y2(k)-p1^2*0.1*y2(k-1)-p2^2*0.2*y2(k-2)-
      -u2(k-1))*(-2*p1*0.1*y2(k-1));
      dVp2=2*(y2(k)-p1^2*0.1*y2(k-1)-p2^2*0.2*y2(k-2)-
      -u2(k-1))*(-2*p2*0.2*y2(k-2));
      tmp2=tmp2+[dVp1 dVp2];
   end
f=1/(2*(N-2))*tmp;
df=1/(2*(N-2))*tmp2;
end
```

Example 3 - Solution

- Use the created `gradient` function to find the minimum of the loss function
- e.g. with p0=[1,1], tol=0.1, step=0.025, max_iter=500

```
[p2,Vp2,p2_hist,V2_hist,grad2_hist,iter2]=
=gradient('V2',[1,1],0.1,0.025,500);
>> p2
p2 =
        1.4786    2.0302
>> Vp2
Vp2 =
        0.4106
```

- estimated parameter values: $\hat{p}_1 = 1.4786$, $\hat{p}_2 = 2.0302$
- loss function value: $V_2(\hat{p}_1, \hat{p}_2) = 0.4106$

Example 3 - Solution

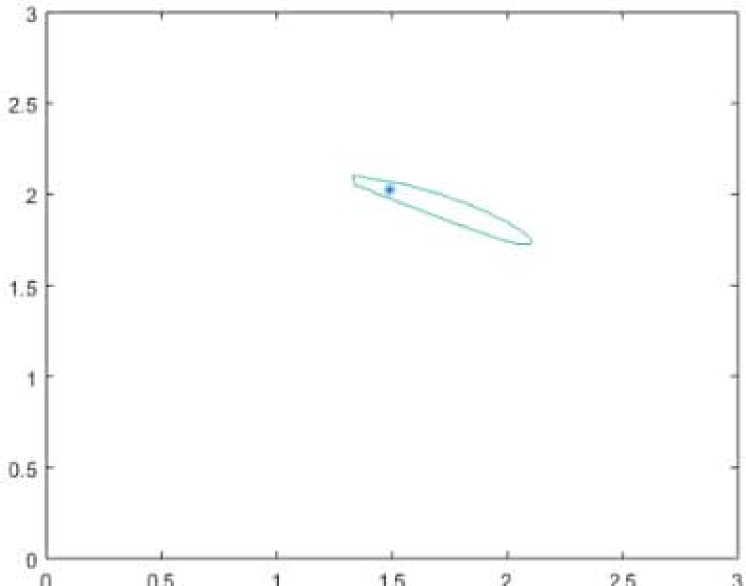
- Plot the confidence region of the estimate
  - first you have to compute the loss function at some point of the parameter space

```
x1=0:0.05:3;
x2=0:0.05:3;
for i=1:length(x1)
for j=1:length(x2)
        x=[x1(i) x2(j)];
        z(j,i)=V2(x);
end
end
```

Example 3 - Solution

- Plot the confidence region of the estimate
  - you can display the contour lines by the `contour` function
  - the confidence region belongs to the 105% of $V_2(\hat{p}_1, \hat{p}_2)$:

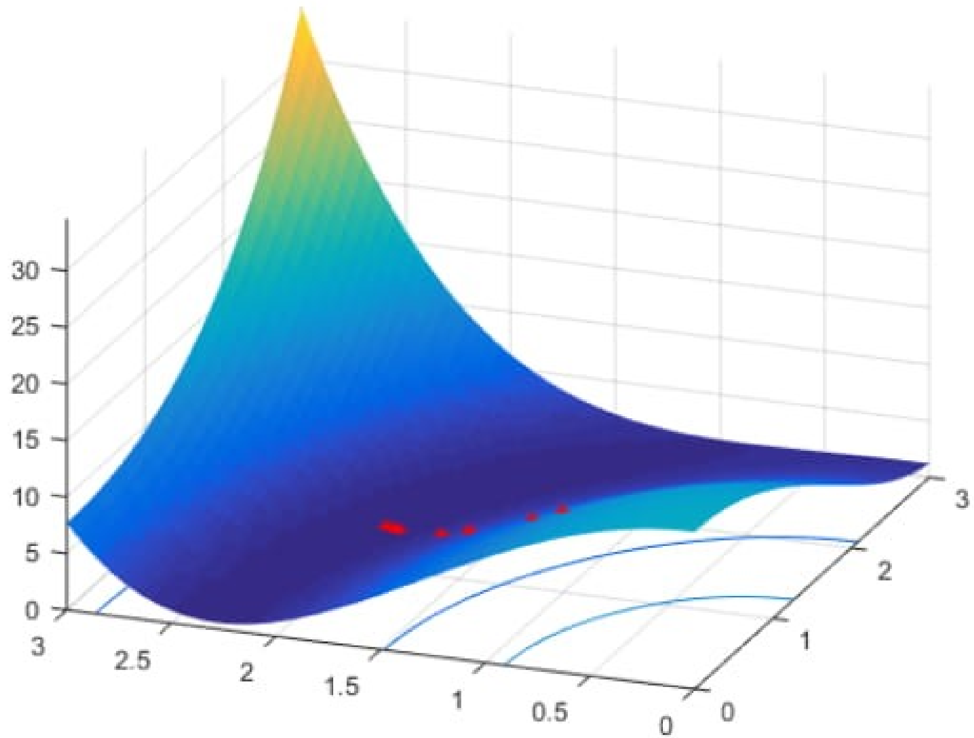
```
>> conf=[1.05*Vp2,1.05*Vp2];
>> contour(x1,x2,z,conf);
>> hold on
>> plot(p2(1),p2(2),'*');
```



Example 3 - Solution

- Plot the surface of the loss function
  - use the `surf` function

```
>> surf(x1,x2,z);
```



EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

MATLAB CODES

Készítette: Anna Pózna

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

autocov.m


```
function r=autocov(data,s)
N=length(data);
r=0;
m=mean(data);
for i=1:N-s
    r=r+(data(i)-m)*(data(i+s)-m);
end
r=1/(N-s)*r;
end
```

LS_scalar.m


```
function [p,sigma,r]=LS_scalar(D)
    y=D(:,1);
    x=D(:,2);
    p=1/(x'*x)*(x'*y);
    y_M=p*x;
    r=y-y_M;
    m_r=mean(r);
    sigma_r=var(r);
    sigma=1/(x'*x)*sigma_r;
end
```

LS_static.m


```
function [p,COV,r]=LS_static(D)
y=D(:,1);
X=D(:,2:end);
if rank(X)<min(size(X))
    error('Error: matrix is rank deficient!')
end
p=inv(X'*X)*X'*y;
y_M=X*p;
r=y-y_M;
m_r=mean(r);
sigma_r=var(r);
COV=inv(X'*X)*sigma_r;
end
```

gradient.m


```
function
    [p,Vp,p_hist,fun_hist,grad_hist,iter]=gradient(fun,p0,tol,step,
    max_iter)
   iter=0;
   p=p0;
   [Vp,dV]=feval(fun,p);
%     Vp=feval(grad,p);
   p_hist=p;
   grad_hist=[];
   fun_hist=[];
```

gradient.m

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

```
...
  while(max(abs(dV))>tol)&(iter<max_iter)
    [Vp,dV]=feval(fun,p);
    p=p-step*dV;
    if feval(fun,p-0.5*step*dV)<feval(fun,p-2*step*dV)
      step=0.5*step;
    else
      step=2*step;
    end
    p_hist=[p_hist;p];
    fun_hist=[fun_hist;Vp];
    grad_hist=[grad_hist;dV];
    iter=iter+1;
  end
...
```

gradient.m


```
...
    [Vp,dV]=feval(fun,p);
    fun_hist=[fun_hist;Vp];
    grad_hist=[grad_hist;dV];
end
```

V.m


```
function [f,df]=V(a)
%y(k+1)=exp(-a)*y(k)-5*u(k)+e(k)
global y u;
   N=length(y);
   tmp=0;
     tmp2=0;
   for k=2:N;
      tmp=tmp+(y(k)-exp(-a)*y(k-1)+5*u(k-1))^2;
            tmp2=tmp2+2*(y(k)-exp(-a)*y(k-1)+5*u(k-1))*exp(-a)*y(k-1);
   end
   f=1/(2*(N-1))*tmp;
   df=1/(2*(N-1))*tmp2;
end
```

V2.m


```
function [f,df]=V2(P)
%y(k+1)=p1^2*0.1*y(k)+p2^2*0.2*y(k-1)+u(k)+e(k)
global y2 u2;
p1=P(1);
p2=P(2);
    N=length(y2);
    tmp=0;
    tmp2=[0 0];
...
```

V2.m

```
...
 for k=1:N-2;
     tmp=tmp+(y2(k+2)-p1^2*0.1*y2(k+1)-p2^2*0.2*y2(k)-u2(k+1))^2;
     %gradients
     dVp1=2*(y2(k+2)-p1^2*0.1*y2(k+1)-p2^2*0.2*y2(k)-u2(k+1))*(-
    2*p1*0.1*y2(k+1));
     dVp2=2*(y2(k+2)-p1^2*0.1*y2(k+1)-p2^2*0.2*y2(k)-u2(k+1))*(-
    2*p2*0.2*y2(k));
     tmp2=tmp2+[dVp1 dVp2];
  end
  f=1/(2*(N-2))*tmp;
  df=1/(2*(N-2))*tmp2;
end
```

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE