PARAMETER ESTIMATION – 4

Least squares (LS) estimation and its properties in the dynamic case

Anna Ibolya Pózna

University of Pannonia
Faculty of Information Technology
Department of Electrical Engineering and Information Systems

pozna.anna@virt.uni-pannon.hu

October 21, 2020

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

1. Discrete time LTI stochastic input-output models
   - DT-LTI SISO I/O system models
2. Minimizing the prediction error
   - Predictive input-output models
   - Minimizing the prediction errors
3. The least squares estimate
   - Predictive models linear in parameters
   - LS estimation of ARX model parameters
4. Properties of the dynamic least squares estimate
   - Asymptotic behaviour of the LS estimate
5. Tutorial
   - Homework

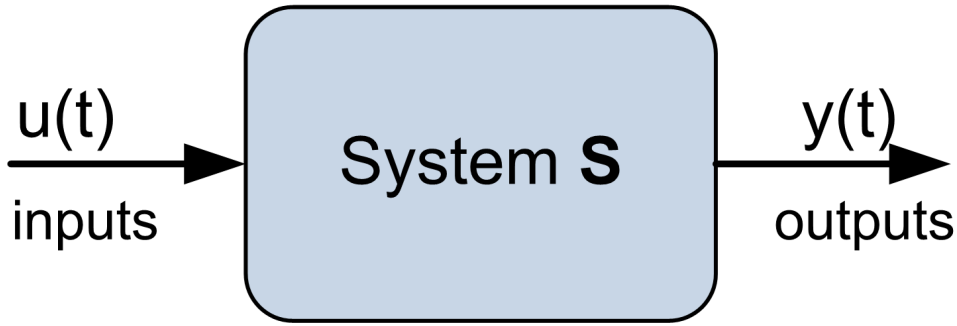
Overview

Recall – Systems

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

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

- inputs (u) and outputs (y)



System: an object in which variables interact and produce observable signals. The system acts on signals. The system is affected by external stimuli.
Outputs = observable signals that are interest of us
Inputs = external signals that can be manipulated by the observer

Recall – System properties

- Linearity

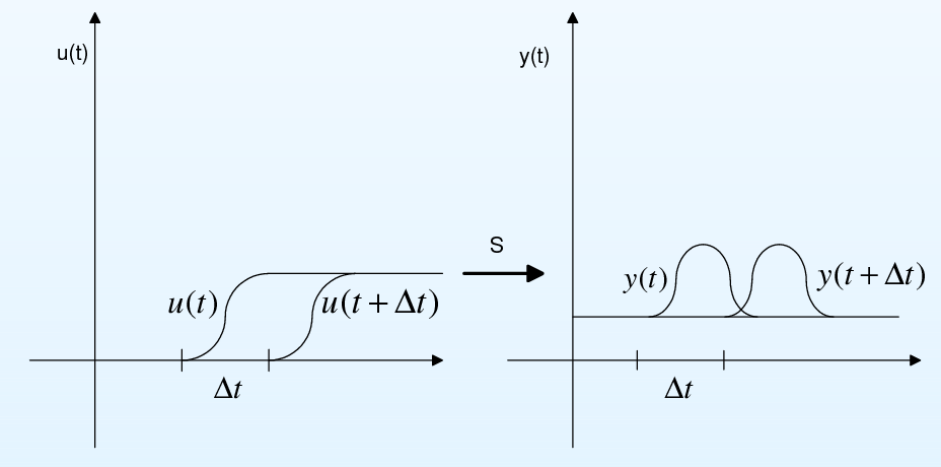
$$\mathsf{S}[c_1 u_1 + c_2 u_2] = c_1 y_1 + c_2 y_2$$

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

- Time-invariance

$$\mathsf{T}_\tau \circ \mathsf{S} = \mathsf{S} \circ \mathsf{T}_\tau$$

where $\mathbf{T}_\tau$ is the time-shift operator: $\mathsf{T}_\tau(u(t)) = u(t+\tau), \quad \forall t$
Time invariance check: **constant parameters**



The most important system properties in this course are:

- Linearity: S is a linear operator. The system is said to be linear if its output response to a linear combination of inputs is the same linear combination of the output responses of the individual inputs.
- Time invariance: the output does not depend on the particular time the input applied. If the input is applied now or τ seconds later, then the output will be identical, except for the time delay. The parameters are independent of time, constant parameters.
- Causality: the system is causal if the output at a certain time depends on the input up to that time only

Recall – Discrete time LTI SISO I/O system models

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

- Backward difference form

$$y(k)+a_1y(k-1)+\ldots+a_ny(k-n)=b_0u(k-d)+\ldots+b_mu(k-d-m)$$

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 $A^*(q^{-1})=1+a_1q^{-1}+\ldots+a_nq^{-n}$ and $B^*(q^{-1})=b_0+b_1q^{-1}+\ldots+b_mq^{-m}$ are polynomials of the time delay operator q^{-1}.

The systems can be represented in different forms.
The Discrete Time Linear Time Invariant Single Input Single Output (DT LTI SISO) systems can be written in a discrete difference equation form.
discrete time: observations of inputs and outputs at discrete time instances (e.g. $0, T, ..., kT$, where k is the sampling instant, and T is the sampling interval)
In this course the **backward difference** form is used. Practical causes: in the backward difference form, we look at the system from the current time instance(k), from where we can look back to the past values (k-1, k-2,...). We know the past input/output values, because they have been measured already. (causality)
The time delay of the output is n.
The time delay d between the output and the input (i.e. the pole excess time delay) has to be greater than 0, because of the causality. It means that output can be affected by the present and past values of the inputs, but not the future inputs.
In the forward difference form, we need to consider future values, which is not applicable in practice, and violates causality.
The time delay operator q^{-1} is often used to make a compact representation of the input-output model.

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.

- Deterministic model: the outputs are uniquely determined by a mathematical expression.
- **Stochastic** model: the output values cannot be uniquely determined, because uncontrollable input effects (e.g. noise, disturbance, inputs that we cannot control). Such signals are modelled as random processes.
- The general form of the input-output model of discrete time stochastic LTI SISO systems is the canonical ARMAX process.
  ARMAX= AutoRegressive Moving Average with eXogenous inputs
  Autoregressive (AR) process: $A^*(q^{-1})y(k) = e(k)$
  Moving Average (MA) process: $y(k) = C^*(q^{-1})e(k)$
  exogenous inputs: $B^*(q^{-1})u(k)$
- $y(k)$ output, $u(k)$ input, $e(k)$ disturbance/noise
- Stable polynomial (discrete time): all of its roots are inside the unit circle.

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*

$$\begin{aligned} y(k) &= -a_1 y(k-1) - .. - a_n y(k-n) + b_0 u(k) + .. + b_m u(k-m) + \\ &+ e(k) \\ &= p^T \varphi(k-1) + e(k) \end{aligned}$$

This model is linear in parameters $p = [-a_1 \ ... \ -a_n \ b_0 \ ... b_m]^T$ *if one measures the data* $\varphi(k-1) = [y(k-1) \ ... \ y(k-n) \ u(k) \ ... \ u(k-m)]^T$.

- If we assume only **independent** measurement noise, we get an ARX process (AutoRegressive with eXogenous inputs). Independent measurement noise $\rightarrow C^*(q-1) = 1$
- This is the simplest discrete time stochastic LTI input-output model.
- The ARX model can be written in a **predictive** form: we move all past output values to the right side of the equation. Now it can be written as the product of the parameter vector p and the predictor φ.
- The parameter vector includes all coefficients of $A^*(q^{-1})$ and $B^*(q^{-1})$.
- The predictor contains the past values of the outputs and the current and past input values.
- The current value of the output can be predicted based on the past measured data.

Overview

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

The general form of a SISO LTI stochastic input-output model is $F(q^{-1})y(k) = G(q^{-1})u(k) + \Delta(q^{-1})e(k)$.

- F, G and Δ are linear functions of the time shift operator q^{-1} (in case of ARMAX models, they are polynomials).
- $\{e(k)\}_0^\infty$ is a white noise process (sequence of identically distributed, independent random variables).
- To be able to perform the parameter estimation, the model should be written in a predictive form. In the predictive form, the estimated output $\hat{y}(k|p)$ at the current time k assuming the parameter vector p can be expressed as the function of the past inputs and outputs, WITHOUT the noise.
- $W_y(q^{-1}, p)$ is a function (a linear filter) of the time shift operator that depends on the parameter vector. It gives the relationship between the past outputs and the current output.
- $W_u(q^{-1}, p)$ is a function (a linear filter) of the time shift operator that depends on the parameter vector. It gives the relationship between the past inputs and the current output.
- p is the vector of unknown parameters to be estimated

Predictive form of ARMAX models

General I/O model of discrete time linear time invariant stochastic SISO systems

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

Important

Predictive form:

$$\hat{y}(k|p) = y(k) - e(k) = (1 - H^{-1}(q^{-1}, p)) \cdot y(k) + H^{-1}(q^{-1}, p) G(q^{-1}, p) \cdot u(k)$$

where

$$H^{-1}(q^{-1}, p) = \frac{A^*(q^{-1})}{C^*(q^{-1})}, \; G(q^{-1}, p) = \frac{B^*(q^{-1})}{A^*(q^{-1})}$$

It contains only the past measured data (!!) without the noise term.

In case of ARMAX models, the predictive from is $\hat{y}(k|p) = y(k) - e(k) = (1 - H^{-1}(q^{-1}, p)) \cdot y(k) + H^{-1}(q^{-1}, p)G(q^{-1}, p) \cdot u(k)$

- dividing both sides of the ARMAX equation by $C^*(q^{-1})$, we can express $e(k)$:
  $e(k) = \frac{A^*(q^{-1})}{C^*(q^{-1})} \cdot y(k) - \frac{B^*(q^{-1})}{C^*(q^{-1})} \cdot u(k)$
- substituting this into the equation of $\hat{y}(k|p)$, we get
  $y(k) - e(k) = y(k) - \frac{A^*(q^{-1})}{C^*(q^{-1})} \cdot y(k) + \frac{B^*(q^{-1})}{C^*(q^{-1})} \cdot u(k) =$
  $= (1 - \frac{A^*(q^{-1})}{C^*(q^{-1})}) \cdot y(k) + \frac{B^*(q^{-1})}{C^*(q^{-1})} \cdot u(k)$
- $H^{-1}(q^{-1}, p) = \frac{A^*(q^{-1})}{C^*(q^{-1})}$
- $G(q^{-1}, p) = \frac{B^*(q^{-1})}{A^*(q^{-1})}$

It contains only the measured data ($u(k)$ and $y(k)$), without the measurement noise. This form can be used to estimate the parameters of the ARMAX model.

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

$$\hat{y}(k|p) = y(k) - e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$$

The elements of the estimator:

$$p = [-a_1 \ -a_2 \ \ldots \ -a_n \ \ b_0 \ b_1 \ \ldots \ b_m]^\top \qquad N > n + m$$

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + b_0 \cdot u(k) + \ldots + \\ + \ldots + b_m \cdot u(k-m)$$

The simplest stochastic model is the ARX model where the output noise is white, i.e. $C^*(q^{-1}) = 1$

- $H^{-1}(q^{-1}, p)$ becomes $A^*(q^{-1})$, $H^{-1}(q^{-1}, p) = \frac{A^*(q^{-1})}{C^*(q^{-1})} = A^*(q^{-1})$
- $H^{-1}(q^{-1}, p)G(q^{-1}, p)$ becomes $B^*(q^{-1})$
  $H^{-1}(q^{-1}, p)G(q^{-1}, p) = \frac{A^*(q^{-1})}{C^*(q^{-1})} \cdot \frac{B^*(q^{-1})}{A^*(q^{-1})} = B^*(q^{-1})$

which result in the predictive form of ARX models: $\hat{y}(k|p) = y(k) - e(k) = (1 - A^*(q^{-1})) \cdot y(k) + B^*(q^{-1}) \cdot u(k)$

- the parameter vector contains the coefficients of $A^*(q^{-1})$ and $B^*(q^{-1})$
- the regressor is $\varphi = [y(k-1) \dots y(k-n), u(k) \dots u(k-m)]^T$
- $\hat{y}(k|p) = p^T \cdot \varphi$

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:

$$D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots N\}$$

Important (Linear-in-parameter case)

Systems that are linear-in-parameters :

$$\hat{y}(k|p) = p^\top \cdot g^*(k, D[1, k-1])$$

In case of nonlinear time invariant SISO systems the general predictive form is a nonlinear function of the measured data and the parameter vector

- $g(k, D[1, k-1]; p)$ is a nonlinear function, that depends on the past k-1 measured data and the parameter vector
- $D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$ is the time series of the measured input (u) and output (y) data.

In many cases the nonlinear model is linear in parameters, which means it can be written as the product of the parameter vector and the measured variables: $\hat{y}(k|p) = p^\top \cdot g^*(k, D[1, k-1])$

These kinds of models can be estimated by linear methods, using auxiliary variables.

Example: ARX model

ARX model is a model that is **linear in parameters**.
Model elements:

- predictive model form

$$\hat{y}(k|p) = -a_1 \cdot y(k-1) - \ldots - a_n \cdot y(k-n) + \\ + b_0 \cdot u(k) + \ldots + b_m \cdot u(k-m)$$

- 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:

$$\varepsilon(k,p) = y(k) - \hat{y}(k|p) \qquad k = 1,\ldots,N$$

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 .

The prediction error is the difference between the real value $y(k)$ and the model predicted value $\hat{y}(k|p)$. The principle of the parameter estimation is to generate the value of the estimated parameter from the measured data such that the prediction error is as small as possible. The model and the estimated parameters are good if the prediction error is small. If the estimation is good, then the estimated parameters are close to the real parameters. For a good estimation we need a good assumption about the model structure, too (number of inputs and outputs, linear or nonlinear, etc).

What does it mean that the prediction error is small? It can be measured using an appropriate signal norm. The signal norm is a function from a vector space to the nonnegative real numbers that satisfies certain properties (scalability, triangle inequality, zero condition), e.g. absolute value.

Minimizing the prediction error

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.

The parameter estimation is a mapping from the sequence of measured data to the parameter vector.
The general parameter estimation problem is the following:
The known input of the problem are

- the sequence of measured input-output data
  $D[1, N] = D^N = \{(y(k), u(k)) \mid k = 1, \ldots N\}$
- the predictive parametrized system model
  $\hat{y}(k|p) = g(k, D[1, k-1]; p)$
  predictive: it contains only the input and output variables of the model
  parametrized: it depends on the parameters to be estimated. The parameters are not known. The model structure defines the set of possible models which differs in the parameters only.
- From the measured i-o data and the predictive parametrized model we can generate the prediction error series:
  $\varepsilon(k, p) = y(k) - \hat{y}(k|p) \qquad k = 1, \ldots, N$
  It is the difference between the real measured output and the model predicted output.

Minimizing the prediction error

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.

- a suitable norm of the prediction error
  $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \ell(\varepsilon(k, p))$, where $\ell(\cdot)$ is a positive scalar-valued function; most frequently: $\ell(\varepsilon) = \frac{1}{2}\varepsilon^2$.
  The function $V_N(p, D^N)$ for a given D^N is a well defined **scalar valued** function of the model parameter p. The choice of the quadratic norm for ℓ is a standard choice which is convenient for computation and analysis.

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. Then we need to find that parameter $\hat{p}_N$ that minimize this loss function. The estimate is the parameter, that minimizes the loss function.

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:

$$p = [-a_1 \;\; -a_2 \;\; \ldots \;\; -a_n \;\; b_0 \;\; b_1 \;\; \ldots \;\; b_m]^\top$$

*Prediction error (*white noise!*):*

$$\varepsilon(k) = \hat{y}(k|p) - y(k) = e(k)$$

Overview

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 (Least Squares)*

$$V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} \left[y(k) - p^\top \varphi(k) \right]^2$$

Now we know the predictive form of the models, let's see how can we estimate the parameters. We will consider the special case, when the model is linear in parameters. It is one of the simplest model structures. The linear in parameter model can be written in $\hat{y}(k|p) = p^\top \varphi(k) = \varphi(k)^\top p$, which is a so called linear regression structure. $\hat{y}(k|p)$ is the **predictor** (= it predicts the next output value based on the parameters and the regressor). $\varphi(k)$ is the so-called **regressor** which contains the measured data, are used to predict the target, outcome, dependent variable, $p^\top$ is the parameter vector. In the linear regression model structure the predictor is the product of the parameter vector and the regressor.

The **prediction error** in this case is $\varepsilon(k, p) = y(k) - \hat{y}(k|p) = y(k) - p^\top \varphi(k)$.

The loss function to be minimized is a quadratic function, which contains the sum of squares of the prediction error sequence: $V_N(p, D^N) = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} \left[y(k) - p^\top \varphi(k) \right]^2$. Note that it is similar to the loss function used at the least squares estimation of static models.

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)

$$\hat{p}_{LS}=\left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

The task is to minimize the quadratic loss function.

- The loss function has a minima where its derivative is 0. Because the loss function is a quadratic function in the parameter vector p, it can be minimized analytically. $(f \circ g)' = (f' \circ g) \times g'$, $f = ()^2$, $g = \left[y(k) - \varphi^\top(k) \cdot p\right]$
- Taking the partial derivatives w.r.t. the elements of the parameter vector we get:
  $\frac{d}{dp} \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} \left[y(k) - p^\top \varphi(k)\right]^2 =$
  $2 \cdot \frac{1}{N} \sum_{k=1}^{N} \frac{1}{2} \left[y(k) - p^\top \varphi(k)\right] \cdot (-\varphi(k)) = 0$
  which is the same as $\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \left[y(k) - \varphi^\top(k) \cdot p\right] = 0$ (dividing by -1)
- We need to solve the above equation for p. Resolving the braces, we get $\frac{1}{N} \sum_{k=1}^{N} \varphi(k)y(k) - \varphi(k)\varphi^\top(k) \cdot p = 0$
  Moving the second term to the right side:
  $\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k) = \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k) \cdot p$
- Then multiply both sides by the inverse of $\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k)$
  We get the LS estimate of p:
  $\hat{p}_{LS} = \left[\frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \varphi^\top(k)\right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot y(k)$

Example: LS estimate of ARX model parameters

ARX model is the basic case: *the output noise is white*

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

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*:*

$$\varphi(k) = [y(k-1) \ y(k-2) \ \ldots \ y(k-n) \ \ u(k) \ u(k-1) \ \ldots \ u(k-m)]^\top$$

Overview

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.*

The LS estimate of dynamic models is a bit different from the standard linear regression, because the measured outputs appear in the regressor. It means that the measured values of $y(k)$ may contain **not only independent** white noise errors. Therefore the estimate can be biased.

The behaviour of the estimate when the number of samples goes to infinity is called asymptotic behaviour. For example we can talk about asymptotic unbiasedness, which means that the estimated parameter will be closer to its true value, if we increase the number of samples.

We can analyse the asymptotic properties of the least squares estimate using the following model:

$y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$ where $\nu_0(k)$ is the error series and p_0 is the "true" value of the parameter.

Dynamic LS estimate: Asymptotic properties – 2

Important (LS estimate and notation)

$$\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\cdot\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$$

$$R(N) = \frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)$$

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.

- The LS estimate of p can be written as before:
  $\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\cdot\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$
- We can denote the term $\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)$ by $R(N)$.
- Substituting $R(N)$ and $y(k)$ to the equation of the LS estimate, it can be be written as
  $\hat{p}_{LS}(N) = [R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\left[\varphi(k)^{\top}\cdot p_0 + \nu_0(k)\right]$
- Resolving the braces, the first term on the right side is
  $[R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi(k)^{\top}\cdot p_0 = [R(N)]^{-1}R(N)\cdot p_0 = p_0$
- The second term is $[R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\nu_0(k)$
- Therefore the LS estimate can be written in the following form $\hat{p}_{LS}(N) = p_0 + [R(N)]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\nu_0(k)$. It can be seen that it is composed of the true value of p (p_0) and the estimation error .

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.

.

We want that the estimation error be as small as possible, and to converge to 0 as the sample size is growing. If these conditions are fulfilled then the estimate will be asymptotically unbiased. If the estimation error is small then the estimated parameter is close to the real parameter. If the error converges to 0 as the sample size is growing then we can improve the 'accuracy' of the estimate by increasing the sample size.

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.

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.

We want that the estimated model be close to the real model. Looking at equation of the estimated model $y(k) = p_0^\top \cdot \varphi(k) + \nu_0(k)$ we can see that it is close to the real model if the error $\nu_0(k)$ is as small as possible. Similarly the estimated parameter $\hat{p}_{LS}$ is close to the real parameter p_0 if the estimation error is small. The estimation error can be expressed as $[R(N)]^{-1} \frac{1}{N} \sum_{k=1}^{N} \varphi(k) \cdot \nu_0(k)$. We want to make this expression as small as possible. The estimation error $[R(N)]^{-1}$ contains the product of the regressor $\varphi(k)$ and the error $\nu_0(k)$. When the $\nu_0(k)$ error is small compared to the regressor $\varphi(k)$ containing measured values, then the estimation error will also be small.

stationary process: the statistical characteristics (mean, covariance,...) of the process do not change in the course of time t. The statistical characteristics at time t are the same as at time $t+\tau$ (similar to the time invariance of dynamic systems). It is important to not that if $u(k)$ and $\nu_0(k)$ are stationary processes then the output $y(k)$ of an AR(MA)X process will be also a stationary process.

Overview

Tutorial problem: Dynamic LS estimation

Consider the following ARX model:

$$y(k) = -a_1 y(k-1) + b_1 u(k-1)$$

Consider the measured input and output data:
$u(0) = 0, u(1) = 1, u(2) = 1, u(3) = 0, u(4) = 0,$
$y(0) = 0, y(1) = 0, y(2) = -1, y(3) = 0.5, y(4) = -0.75$

- Construct the parameter vector p.
- Construct the regressor $\varphi(k)$ for $k = 1, 2, 3, 4$
- Compute the LS estimate of p

Tutorial problem: Dynamic LS estimation

Consider the following ARX model:

$$y(k) = -a_1 y(k-1) + b_1 u(k-1)$$

Consider the measured input and output data:
$u(0) = 0, u(1) = 1, u(2) = 1, u(3) = 0,$
$y(0) = 0, y(1) = 0, y(2) = -1, y(3) = 0.5$

- Construct the parameter vector p.
  $p = [-a_1,\ b_1]^T$
- Construct the regressor $\varphi(k)$ for $k = 1, 2, 3$
  $\varphi(k) = [y(k-1), u(k-1)]^T$
  $\varphi(1) = [y(0), u(0)]^T = [0, 0]^T$
  $\varphi(2) = [y(1), u(1)]^T = [0, 1]^T$
  $\varphi(3) = [y(2), u(2)]^T = [-1, 1]^T$
- Compute the LS estimate of p

Tutorial problem: Dynamic LS estimation

- Compute the LS estimate of p

  $\hat{p}_{LS} = \left[\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot\varphi^{\top}(k)\right]^{-1}\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\cdot y(k)$

- $\varphi(1)\varphi^{T}(1) = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}$, $\varphi(2)\varphi^{T}(2) = \begin{bmatrix}0 & 0\\ 0 & 1\end{bmatrix}$, $\varphi(3)\varphi^{T}(3) = \begin{bmatrix}1 & -1\\ -1 & 1\end{bmatrix}$
- Computing the sum $\frac{1}{N}\sum_{k=1}^{N}\varphi(k)\varphi^{T}(k)$:

$$\frac{1}{3}\sum_{k=1}^{3}\varphi(k)\varphi^{T}(k) = \frac{1}{3}\begin{bmatrix}1 & -1\\ -1 & 2\end{bmatrix} = \begin{bmatrix}\frac{1}{3} & -\frac{1}{3}\\ -\frac{1}{3} & \frac{2}{3}\end{bmatrix}$$

- Computing the inverse:

$$\begin{bmatrix}\frac{1}{3} & -\frac{1}{3}\\ -\frac{1}{3} & \frac{2}{3}\end{bmatrix}^{-1} = \frac{1}{det(A)}adj(A) = 9\begin{bmatrix}\frac{2}{3} & \frac{1}{3}\\ \frac{1}{3} & \frac{1}{3}\end{bmatrix} = \begin{bmatrix}6 & 3\\ 3 & 3\end{bmatrix}$$

Tutorial problem: Dynamic LS estimation

- $\varphi(1)y(1) = [0, 0]^T$, $\varphi(2)y(2) = [0, -1]^T$, $\varphi(3)y(3) = [-0.5, 0.5]^T$
- Computing the sum $\frac{1}{N}\sum_{k=1}^{N}\varphi(k)y(k)$:

$$\frac{1}{3}\sum_{k=1}^{3}\varphi(k)y(k) = \frac{1}{3}\begin{bmatrix}-0.5\\-0.5\end{bmatrix} = \begin{bmatrix}-\frac{1}{6}\\-\frac{1}{6}\end{bmatrix}$$

- Computing the estimate:

$$\hat{p}_{LS} = \begin{bmatrix}6 & 3\\3 & 3\end{bmatrix}\cdot\begin{bmatrix}-\frac{1}{6}\\-\frac{1}{6}\end{bmatrix} = \begin{bmatrix}-1.5\\-1\end{bmatrix}$$

- $a_1 = 1.5,\ b_1 = -1$

HOMEWORK Deadline: 28 October 2020. 10:00

Consider the following ARX model:

$$y(k) = -a_1 y(k-1) + b_0 u(k)$$

Consider the measured input and output data:
$u(1) = 1, u(2) = 0, u(3) = 1,$
$y(0) = 0, y(1) = 2, y(2) = -2, y(3) = 4$

- Construct the parameter vector p.
- Construct the regressor $\varphi(k)$ for $k = 1, 2, 3$
- Compute the LS estimate of p

Send the solution to **pozna.anna@virt.uni-pannon.hu**!