PARAMETER ESTIMATION – 5

Special methods for dynamic LS estimation:
Instrumental variable method
Estimation of nonlinear dynamic models

Katalin Hangos
Department of Electrical Engineering and Information Systems
University of Pannonia

Nov 2020

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. Revision of the LS method for ARX models
2. The instrumental variable method
   - IV method: basic idea
   - The IV method
   - The IV4 algorithm in Matlab
3. 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:*

$$y(k+1) = -a_1 y(k) - \ldots - a_n y(k-n) + b_0 u(k) + \ldots + \\ + b_m u(k-m) + e(k)$$

Unknown **parameter vector** *to be determined:*

$$p := \left[\begin{array}{cccccc} -a_1 & \ldots & -a_n & b_0 & \ldots & b_m \end{array} \right]^T$$

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:

$$\varphi(k) := \left[\begin{array}{cccccc} y(k-1) & \ldots & y(k-n) & u(k) & \ldots & u(k-m) \end{array}\right]^T$$

Predictive model linear in parameters:

$$\hat{y}(k,p) = \varphi^T(k)p$$

Prediction error:

$$\varepsilon(k,p) = y(k) - \hat{y}^T(k); \quad \varepsilon(k,p) = y(k) - \varphi^T(k)p$$

The 2-norm of the error should be **minimized**:

$$V_N(p, D^N) = \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}(y(k) - \varphi^T(k)p)^2$$

ARX model: The estimation

The optimal parameter value is obtained from $\frac{\partial V_N(p,D^N)}{\partial p} = 0$

$$\hat{p}_N^{LS} = \left[\frac{1}{N}\sum_{i=1}^{N}\varphi(k)\varphi(k)^T\right]^{-1}\frac{1}{N}\sum_{i=1}^{N}\varphi(k)y(k)$$

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.**

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

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

The LS method is not optimal in this case

Important (AR**MA**X model)

In case of an ARMAX model: past outputs are correlated with the noise

$$\begin{aligned} y(k) = \quad & -a_1 y(k\text{-}1) - \ldots - a_n y(k-n) + b_0 u(k) + \ldots + b_m u(k-m) + \\ & + c_0 e(k) + c_1 e(k-1) + \ldots + c_n e(k-n) \end{aligned}$$

$$y(k) = \varphi^T(k)p_0 + c_0 e(k) + \ldots + c_n e(k-n)$$

IV method: Basic idea

Given

$$y(k) = \varphi^T(k)p_0 + \nu_0(k)$$

with correlated $\varphi(k)$ and $\nu_0(k)$.

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.

$$\varphi(k) := \left[\; y(k\text{-}1) \;\; \ldots \;\; y(k-n) \;\; u(k) \;\; \ldots \;\; u(k-m) \;\right]^T$$

IV method: The estimate

We are searching for an instrumental variable $\xi(t)$ for which:

$$\hat{p}_N^{IV} = sol \left\{ \frac{1}{N} \sum_{i=1}^{N} \xi(k) \left[y(k) - \varphi^T(k)\theta_0 \right] = 0 \right\}$$

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

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:

$$\xi(k) = \left[\begin{array}{cccccc} -z(k-1) & \ldots & -z(k-n) & u(k) & \ldots & u(k-m) \end{array} \right]^T$$

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 $K_u(q^{-1}, p_{pre}) = \frac{\hat{A}(q^{-1})}{\hat{B}(q^{-1})}$.

Important (principle of the IV method)

- *Computing the instrumental variables*

$$\begin{aligned} z(k) &= K_u(q^{-1}, p_{pre}) u(k) \\ \xi(k) &= [z(k-1) \ \dots \ z(k-n) \ \ u(k) \ \dots \ u(k-m)]^T \\ \varepsilon_F(k,p) &= y(k) - \varphi^T(k,p) p \end{aligned}$$

- *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

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:

$$\hat{p}_N^{(1)} = \hat{p}_N^{LS}, \quad \hat{G}_N^{(1)}(q^{-1}) = \frac{\hat{B}_N^{(1)}(q^{-1})}{\hat{A}_N^{(1)}(q^{-1})}$$

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.

Important (Step 2: IV estimation I.)

The instrumental variables with the pre-estimated $\hat{G}_N^{(1)}(q^{-1})$

$$z^{(1)}(k) = \hat{G}_N^{(1)}(q^{-1})u(k)$$

$$\xi^{(1)}(k) = \left[\ z^{(1)}(k-1) \quad \dots \quad z^{(1)}(k-n) \quad u(t) \quad \dots \quad u(k-m)\ \right]^T$$

then compute the corresponding IV estimate and the DT-LTI model:

$$\hat{p}_N^{(2)} = \hat{p}_N^{IV}, \quad \hat{G}_N^{(2)}(q^{-1}) = \frac{\hat{B}_N^{(2)}(q^{-1})}{\hat{A}_N^{(2)}(q^{-1})}$$

IV 4 algorithm – 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.

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*

$$z^{(2)}(k) = \hat{G}_N^{(2)}(q^{-1})u(k)$$
$$\xi^{(2)}(t) = \hat{L}_N(q^{-1})\left[z^{(2)}(k-1)\ \dots\ z^{(2)}(k-n)\ \ u(k)\ \dots\ u(k-m)\right]^T$$

IV 4 algorithm – 4.2

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

$$z^{(2)}(k) = \hat{G}_N^{(2)}(q^{-1})u(k)$$
$$\xi^{(2)}(t) = \hat{L}_N(q^{-1}) \left[z^{(2)}(k-1) \; ... \; z^{(2)}(k-n) \;\; u(k) \; ... \; u(k-m) \right]^T$$

$$\varphi_F(k) = \hat{L}_N(q^{-1})\varphi(k)$$
$$y_F(k) = \hat{L}_N(q^{-1})y(k)$$

$$\hat{p}_N^{(IV)} = \left[\frac{1}{N} \sum_{k=1}^{N} \xi^{(2)}(k)\varphi_F(k)^T \right]^{-1} \frac{1}{N} \sum_{k=1}^{N} \xi^{(2)}(k)y_F(k)$$

Overview

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: *linear in both the model parameters and the signals*

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

$$p = [-a_1 \ \dots \ -a_n \ \ b_0 \ \dots \ b_m]^\top$$

$$g^*(k, D[1, k-1]) = \varphi(k) = [y(k-1) \ \ \dots \ -y(k-n) \ \ u(k) \ \dots \ u(k-m)]^\top$$

Example: Nonlinear model linear-in-parameters

Nonlinear ARX model that depends linearly on the parameters

$$y(k) = a_1 \cdot y^2(k-1) + b_0 \cdot u^4(k) + e(k)$$

$$p = [a_1 \ b_0]^\top \qquad \hat{y}(k|p) = y(k) - e(k)$$

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**

Recall: Minimizing the prediction error

Method of parameter estimation: $D^N \to \hat{p}_N$

Important (The general task of parameter estimation)

Given

- *measured values:* $D[1,N] = D^N = \{(y(k), u(k)) \mid k = 1, \dots, N\}$
- *parametrized predictive model:* $\hat{y}(k|p) = g(k, D[1, k-1]; p)$ *and the sequence of prediction errors (discrete-time signal):* $\varepsilon(k,p) = y(k) - \hat{y}(k|p) \quad , \quad k = 1, \dots, N$
- *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* $V_N(p, D^N)$

$$\hat{p}_N = \hat{p}_N(D^N) = \arg\min_p V_N(p, D^N)$$

Parameter estimation of nonlinear systems

Method of parameter estimation: $D^N \rightarrow \hat{p}_N$

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

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

$$\hat{p}_m(D^N) = \arg\min_p V_N(p, D^N)$$

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:

$$\hat{p}_{LS}(D^N) = \arg\min_p \frac{1}{N}\sum_{k=1}^{N}\frac{1}{2}[y(k) - g(k, D^{k-1}; p)]^2$$

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 $g(k, D^{k-1}; p)$ $\rightarrow$ 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

$$\frac{1}{N}\sum_{k=1}^{N}[y(k) - g(k, D^{k-1}; p)]\frac{\partial g}{\partial p_j} = 0 \qquad j = 1, \dots, N$$

  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 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.*

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* $x_0 \in \mathbb{R}^n$.
- *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

$$V \ : \ \mathbb{R}^m \rightarrow \mathbb{R}$$

The **gradient** *(gradient vector) of V at* **some point $\mathbf{x} \in \mathbb{R}^\mathbf{m}$** *is the m-dimensional vector*

$$\vec{grad}\ V(x) = [\ \frac{\partial V}{\partial x_1}(x)\ \dots\ \frac{\partial V}{\partial x_m}(x)] \tag{1}$$

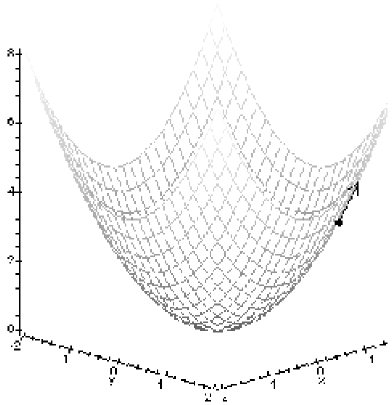
where x_i is the ith coordinate of the vector x.
gradient transpose: $V_x(x) = \vec{grad}\ V^T(x)$

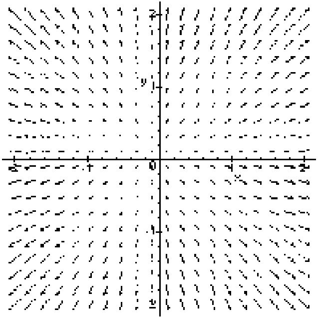
Important

The gradient $\vec{grad}\ V(x)$ in point x shows the direction of the biggest (local) change in the value of the function in the space of the independent variables.

Principle of the gradient method –1

Example: a function with two variables and its gradient vector field





Principle of the gradient method –2

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**

$$\vec{grad}\ V(x^*) = \vec{0} \quad and \quad V_{xx}(x^*) > 0$$

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 $H_V = V_{xx}(x)$.
- Does this V have a minimum? If yes, where?

Solution:

- *gradient vector and Hessian matrix*

$$\vec{grad}\ V(x) = \left[\ 6(x_1 - 4)\ ,\ 8(x_2 - 1)\ \right]\ ,\quad H_V = V_{xx}(x) = \begin{bmatrix} 6 & 0 \\ 0 & 8 \end{bmatrix}$$

- *yes, at the point* $x^* = [4, 1]^T$, *where* $\vec{grad}\ V(x) = [0, 0]$.
  The Hessian matrix is constant and positive definite everywhere.

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:

$$\mathbf{V_N(p, D^N)} \sim V(x)$$
$$\mathbf{p} \sim x$$

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**.

HOMEWORK

Consider the loss function

$$V(x) = x_1^2 - 2x_1x_2 + x_2^2$$

- 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?