System Identification

Anders Hansson and Martin Andersen

Linköping University and Technical Unversity of Denmark

June 17, 2024

Dynamical System Models

A discrete-time dynamical system is described by *state equation*

$$x_{k+1} = F_k(x_k, u_k) \tag{1}$$

where $k \in \mathbf{Z}_+$ is *time*, $x_k \in \mathbf{R}^n$ is the *state*, $u_k \in \mathbf{R}^m$ is the *input signal* and $F_k : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^n$. The *initial value* x_0 is given.

System called *time-varying* if F depends explicitly on k. Otherwise *time-invariant*.

The *output signal* is

$$y_k = G_k(x_k, u_k) \tag{2}$$

where $y_k \in \mathbf{R}^p$ and $G_k : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}^p$.

Linear Time-Invariant System

Obtained by taking F_k and G_k as linear functions independent of k, i.e.

$$x_{k+1} = Ax_k + Bu_k \tag{3}$$

$$y_k = Cx_k + Du_k \tag{4}$$

where $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times m}$, $C \in \mathbf{R}^{p\times n}$ and $D \in \mathbf{R}^{p\times m}$ are called the *system matrices*.

System Identification Problem

Given (y_k, u_k), $0 \leq k \leq N$ and would like to compute $\theta = (A, B, C, D, x_0)$ such that (3–4) holds

Normally the measured sequences do not satisfy (3–4) exactly for any θ due to measurement errors and/or other limitations in the model.

Hence reasonable to define *predictor* of the output as

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ \hat{y}_k &= Cx_k + Du_k \end{aligned}$$

and define the *prediction error*

$$e_k = y_k - \hat{y}_k$$

Equivalent Formulation and Generalization

Equivalently

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned}$$

If errors also in state equation we consider

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + v_k \\ y_k &= Cx_k + Du_k + e_k \end{aligned}$$

where $v_k \in \mathbf{R}^n$ models errors in the state equation.

Maximum Likelihood Formulation

Assume (v_k, e_k) are realizations of a sequence of Gaussian zero mean random variables (V_k, E_k) with covariance

$$R = \begin{bmatrix} R_1 & R_{12} \\ R_{12}^T & R_2 \end{bmatrix} \in \mathbf{S}_{++}^{n+p}$$

such that (V_j, E_j) and (V_k, E_k) are independent for $j \neq k$.

The ML problem of estimating $\theta = (A, B, C, D, R, x_0)$ is

$$\begin{aligned} \text{minimize} \ & \frac{1}{2}\sum_{k=0}^{N-1} \begin{bmatrix} v_k \\ e_k \end{bmatrix}^T R^{-1} \begin{bmatrix} v_k \\ e_k \end{bmatrix} + \frac{1}{2} e_N^T R_2^{-1} e_N - \frac{N}{2} \ln \det R \\ & - \frac{1}{2} \ln \det R_2 \\ \text{subject to} \ & x_{k+1} = A x_k + B u_k + v_k, \quad 0 \leq k \leq N-1 \\ & y_k = C x_k + D u_k + e_k, \quad 0 \leq k \leq N \end{aligned}$$

with variables (θ, x, v, e), where $x = (x_1, \ldots, x_N)$, $e = (e_0, \ldots, e_N)$ and $v = (v_0, \ldots, v_{N-1})$

Innovation Form

Parameters to estimate can be reduced by considering innovation form:

$$x_{k+1} = Ax_k + Bu_k + Ke_k \tag{5}$$
$$y_k = Cx_k + Du_k + e_k \tag{6}$$

where $K \in \mathbf{R}^{n \times p}$.

Now $\theta = (A, B, C, D, K, R_2, x_0)$, and the ML problem is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} \sum_{k=0}^{N} e_k^T R_2^{-1} e_k - \frac{N+1}{2} \ln \det R_2 \\ \text{subject to} & x_{k+1} = Ax_k + Bu_k + Ke_k, \quad 0 \leq k \leq N-1 \\ & y_k = Cx_k + Du_k + e_k, \quad 0 \leq k \leq N \end{array} \tag{7}$$

with variables (θ, x, e).

Remarks

- When $p = 1$ we can solve a constrained LS problem with no weighting with R_2, and then compute R_2 as $e^T e/(N+1)$, where e is the optimal solution.
- The optimization problem is not convex due to bilinearity of the variables in the constraints.
- Even worse is that there are uncountably many solutions to the problem because of the fact that the input-output relations are unaffected by state transformations.

Single-Input-Single-Output Models ($m = p = 1$)

Define the Z-transform of a signal $x = (x_0, x_1, \ldots)$, where $x_k \in \mathbf{R}^n$, as the function $X : \mathbf{C} \to \mathbf{C}^n$ defined by

$$X(z) = \sum_{k=0}^{\infty} x_k z^{-k}$$

Applied to (5–6) this results in

$$\begin{aligned} zX(z) - zx_0 &= AX(z)BU(z) + KE(z) \\ Y(z) &= CX(z) + DU(z) + E(z) \end{aligned}$$

where Y, U and E are the Z-transforms of y, u, and e, respectively.

Transfer Functions

Eliminate X using first equation and substitute into second equation:

$$Y(z) = C(zI - A)^{-1} z x_0 + \mathcal{G}(z)U(z) + \mathcal{H}(z)E(z)$$

where $\mathcal{G}(z) = C(zI - A)^{-1}B + D$ and $\mathcal{H}(z) = C(zI - A)^{-1}K + I$ are rational and proper functions of z with degree n in denominator.

It holds that

$$\mathcal{G}(z) = \frac{\mathcal{B}(z)}{\mathcal{A}(z)}; \qquad \mathcal{H}(z) = \frac{\mathcal{C}(z)}{\mathcal{A}(z)}$$

where we have defined the polynomials

$$\mathcal{A}(z) = z^n + a_1 z^{n-1} + \cdots + a_{n-1} z + a_n$$
$$\mathcal{B}(z) = b_0 z^n + b_1 z^{n-1} + \cdots + b_{n-1} z + b_n$$
$$\mathcal{C}(z) = z^n + c_1 z^{n-1} + \cdots + c_{n-1} z + c_n$$

Dynamics as Linear Constraint

From $\det(zI - A) = \mathcal{A}(z)$ and $\mathcal{A}(z)(zI - A)^{-1} = \mathrm{adj}(zI - A)$:

$$\mathcal{A}(z)Y(z) = C\,\mathrm{adj}(zI - A)zx_0 + \mathcal{B}(z)U(z) + \mathcal{C}(z)E(z)$$

Multiply this equation with $z^{-k}/(2\pi i)$ for $k = -n, -n+1, \ldots N-n$ and integrate along the unit circle C in the complex plane. Since

$$\oint_C z^k dz = \begin{cases} 2\pi i, & k = -1 \\ 0, & k \neq -1 \end{cases}$$

it follows that

$$T_a y = \xi + T_b u + T_c e \tag{8}$$

where $y = (y_0, \ldots, y_N)$, $u = (u_0, \ldots, u_N)$, and $e = (e_0, \ldots, e_N)$

Toeplitz Matrices

Above

$$T_a = I + a_1 S + \cdots a_{n-1} S^{n-1} + a_n S^n$$
$$T_b = b_0 I + b_1 S + \cdots b_{n-1} S^{n-1} + b_n S^n$$
$$T_c = I + c_1 S + \cdots c_{n-1} S^{n-1} + c_n S^n$$

are Topelitz matrices with the shift matrix $S \in \mathbf{R}^{(N+1)\times(N+1)}$ being a matrix of all zeros except for ones on the first sub-diagonal.

The vector $\xi \in \mathbf{R}^{N+1}$ is a vector of all zeros except for the first n elements, which are functions of the initial value x_0.

Hence $\xi = (\xi_0, 0)$, where $\xi_0 \in \mathbf{R}^n$.

Shift Operator

Each and every row in (8) is related to a specific k above. Except for the first n rows in the above equation, one may equivalently write

$$\mathcal{A}(q)y_k = \mathcal{B}(q)u_k + \mathcal{C}(q)e_k$$

where q shifts the time index of a signal , i.e. $qy_k = y_{k+1}$.

The model is called an Auto-Regressive-Moving-Average model with exogenous terms (ARMAX).

When $C(q) = 1$ we have an Auto-Regressive model with exogenous terms (ARX)

The case when $\mathcal{A}(q) = \mathcal{C}(q)$ is called an Output Error (EO) model.

Constrained Nonlinear LS Problem

The ML problem can be stated as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|e\|_2^2 \\ \text{subject to} & T_a y = \xi + T_b u + T_c e \end{array} \tag{9}$$

with variables (θ, e), where $\theta = (a, b, c, \xi_0)$ with $a = (a_1, \ldots, a_n)$, $b = (b_0, \ldots, a_n)$, and $c = (c_1, \ldots, c_n)$.

Non-uniqueness related to the fact that there are infinitely many realizations of an input-output model have been removed.

Still possible to have non-unique solution related to that the data (y_k, u_k) may not contain enough information to uniquely determine the input-output model.

Criteria that ensure uniqueness are in system identification called *persistence of excitation*

Un-Constrained Nonlinear LS Problem

Since T_c is invertible, we have

$$e = T_c^{-1}\left(T_a y - \xi - T_b u\right)$$

Hence the following un-constrained problem is equivalent:

$$\text{minimize}_{\theta} \frac{1}{2}\left\| T_c^{-1}\left(T_a y - \xi - T_b u\right)\right\|_2^2$$

This is a nonlinear LS problem.

It is a separable LS problem, since if we fix c, then the problem is a linear LS in the remaining variables.

Gradients

$$\frac{\partial e}{\partial a_k} = T_c^{-1} S^k y, \quad 1 \leq k \leq n$$
$$\frac{\partial e}{\partial b_k} = -T_c^{-1} S^k u, \quad 0 \leq k \leq n$$
$$\frac{\partial e}{\partial c_k} = -T_c^{-1} S^k e, \quad 1 \leq k \leq n$$
$$\frac{\partial e}{\partial \xi_0^T} = -T_c^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix}$$

with the convention that $S^0 = I$.

All the gradients obtained by solving a linear system of equations

$$T_c z = r$$

for some right hand side r that is trivial to compute.

Filtration

Since T_c is lower triangular, this system of equations can be solved recursively, and it can also be interpreted as a filtration

$$z_k = \frac{1}{\mathcal{C}(q)} r_k$$

When there are zeros of the polynomial $\mathcal{C}(z)$ such that $|z| > 1$, where $z \in \mathbf{C}$, then this filtration is not numerically well-behaved, and it is also the case that T_c is an ill-conditioned matrix.

State-Space Model from ARMAX Model

$$x_{k+1} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & 0 & \cdots & 0 & -a_{n-1} \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_1 \end{bmatrix} x_k + \begin{bmatrix} b_n - a_n b_0 \\ b_{n-1} - a_{n-1} b_0 \\ \vdots \\ b_1 - a_1 b_0 \end{bmatrix} u_k$$

$$+ \begin{bmatrix} c_n - a_n \\ c_{n-1} - a_{n-1} \\ \vdots \\ c_1 - a_1 \end{bmatrix} e_k$$

$$y_k = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \end{bmatrix} x_k + b_0 u_k + e_k$$

Initial Value

Initial value x_0 satisfies

$$Cx_0 = y_0 - Du_0 - e_0$$

where the right hand side is known. Moreover

$$CAx_0 = Cx_1 - CBu_1 - CKe_1 = y_1 - Du_1 - e_1 - CBu_1 - CKe_1$$

with known right hand side. Continuing like this we obtain

$$\mathcal{O}x_0 = r$$

where

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}$$

is the *observability matrix*, and where r can be computed recursively. Since we have used the observer canonical form the observability matrix is invertible, and hence we can solve for x_0.

Missing Data

Consider

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|e\|_2^2 \\ \text{subject to} & T_a y = \xi + T_b u + T_c e \end{array} \tag{10}$$

with variables (θ, e, y_m). which is the same as (9) except that we also have missing outputs y_m as optimization variables. Here $y_m = T_m y$, where $T = \begin{bmatrix} T_m & T_o \end{bmatrix} \in \mathbf{R}^{(N+1)\times(N+1)}$ is a permutation matrix.

Equivalent unconstrained problem

$$\text{minimize} \frac{1}{2} \left\| T_c^{-1} \left(T_a y - \xi - T_b u \right) \right\|_2^2 \tag{11}$$

with variables (θ, y_m). This is also a separable nonlinear LS problem, since the residual e is linear in the remaining variables when c is fixed.

Block-Coordinate Minimization

A common way of solving this problem is to consider a block-coordinate method, where in every other step (y_m, ξ_0) and (a, b, c) are fixed, respectively.

When (y_m, ξ_0) is fixed we have a standard system identification problem with no missing data and known initial value.

When $(a, b, c,)$ is fixed we have a linear LS problem for (y_m, ξ_0).

Consistency

Unfortunately the solution obtained will not be consistent unless $T_c^{-1} T_a = I$, which is true for Output Error (OE) models, for which $T_a = T_c$, and for Finite Impulse Response (FIR) models, for which $T_a = T_c = I$.

The reason for this is that the above formulation is not equivalent to the ML problem formulation when data is missing

ML Problem for Missing Data

Assume that e is a realization from a normal distribution with zero mean, and covariance equal to $\sigma^2 I$. Let $T_y = T_c^{-1} T_a$, $T_u = T_c^{-1} T_b$ and $T_i = T_c^{-1} \begin{bmatrix} I \\ 0 \end{bmatrix}$. Then we can write

$$e = T_y y - T_i \xi_0 - T_u u$$

Also we define $\mathcal{T}_m = T_y T_m^T$ and $\mathcal{T}_o = T_y T_o^T$, Then it can be shown that the ML problem is equivalent to

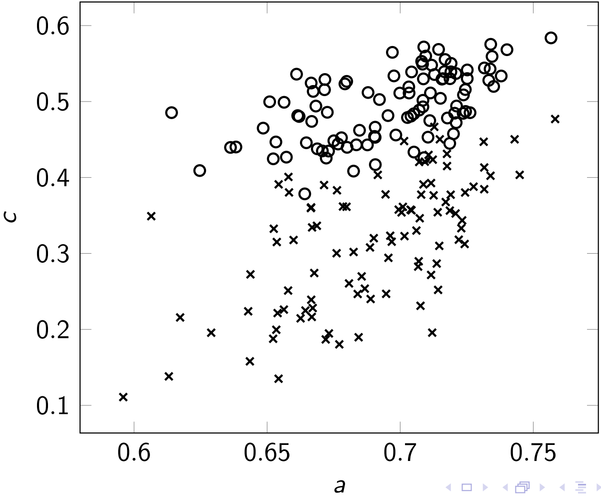
$$\text{minimize} \frac{1}{2} \left\| \det\left(\mathcal{T}_m^T \mathcal{T}_m\right)^{\frac{1}{2n_o}} \left(T_y y - T_i \xi_0 - T_u u\right) \right\|_2^2 \qquad (12)$$

with variables (θ, y_m), where n_o is the number of observed outputs.

This is a separable nonlinear LS problem. It is linear in (y_m, ξ_0) when (a, b, c) is fixed.

Example

Consider an ARMAX model with $a = 0.7$, $b = (0.7, 0)$ and $c = 0.5$. In total 40% of the data is missing. In the figure the results of 100 experiments is presented for the estimates of a and c when using both the criterion in (11) (x) and the one in (12) (o).



Gaussian Processes for Identification

1. ARMAX systems can be arbitrarily well approximated with ARX models, if the model order n in the model is taken large enough.
2. If both the number of data N and the model order n goes to infinity, and N faster than n, then the estimate is consistent.
3. Since computing ARX models is equivalent of solving a linear LS problem, the solution can be obtained both efficiently and accurately.
4. A draw-back with the approach of using high model order is that for a modest number of data, the variance of the estimate of the model parameters will be large.
5. A remedy to this is to model the dynamical system as a Gaussian process, which is as an MAP estimate.

FIR Model

For simplicity we only consider the FIR model

$$y = \xi + T_b u + e$$

which is a special case of the ARMAX model.

Define

$$U = u_0 I + u_1 S + \cdots + u_N S^N$$

and let U_{n+1} be the first $n+1$ columns of U. Then with $\theta = (b, \xi_0)$:

$$y = \Phi\theta + e$$

where

$$\Phi = \begin{bmatrix} U_{n+1} & J \end{bmatrix}$$

with $J = \begin{bmatrix} I & 0 \end{bmatrix}^T \in \mathbf{R}^{(N+1)\times n}$. This is clearly the same model as used for Gaussian processes, if we take $X = \Phi$ and $a = \theta$.

Estimate

We now assume that e is the outcome of a zero mean normally distributed random vector with covariance $\sigma^2 I$ and that θ is a the outcome of a zero mean normally distributed random vector with covariance Σ, and independent of e.

Then the estimate of θ is given by the conditional mean of θ given observations of y as

$$\hat{\theta} = \Sigma\Phi^T \left(\sigma^2 I + \Phi\Sigma\Phi^T\right)^{-1} y$$

This is a consistent estimate if $\Sigma \in \mathbf{S}_{++}^N$, $\Phi\Phi^T/N$ converges to an invertible matrix and if $\Phi e/N$ converges to zero as N goes to infinity.

Emperical Bayes or Type II Maximum Likelihood

Assume that $\Sigma : \mathbf{R}^q \to \mathbf{S}_+^{2n+1}$, where $\Sigma(\eta)$ is describing a suitable subset of $\mathbf{S}_+^{2n+1}$ as the *hyper parameter* η varies.

We choose η to maximize the likelihood function for the observations y, which is the outcome of a zero mean normally distributed random vector with covariance $Z : \mathbf{R}^q \times \mathbf{R}_+ \to \mathbf{S}_{++}^{N+1}$ defined by

$$Z(\eta, \sigma^2) = \Phi\Sigma(\eta)\Phi^T + \sigma^2 I$$

Therefore the ML problem is equivalent to

$$\text{maximize} y^T Z(\eta, \sigma^2)^{-1} y + \log\det Z(\eta, \sigma^2)$$

with variables (η, σ^2), where we also consider σ^2 as a hyper parameter.

Covariance Matrices

We consider $\Sigma_b : \mathbf{R}^3 \to \mathbf{S}_+^{n+1}$ defined by

$$\{\Sigma_b(\eta)\}_{j,k} = \lambda \alpha^{(k+j)/2} \rho^{|j-k|}$$

for $(j,k) \in \mathbf{N}_{n+1} \times \mathbf{N}_{n+1}$, where $\eta = (\lambda, \alpha, \rho)$ with $\lambda \in \mathbf{R}_+$, $0 \leq \alpha < 1$, and $|\rho| \leq 1$. This is called a diagonal/correlated (DC) kernel (covariance function).

We let $\Sigma = \mathbf{bdiag}(\Sigma_b, \Sigma_\xi)$, where $\Sigma_\xi \in \mathbf{S}_+^n$.

The advantage with this parameterization is that the ML criterion then is the difference between two convex functions $g : \mathbf{R}^q \to \mathbf{R}$ and $h : \mathbf{R}^q \to \mathbf{R}$ defined by $g(\eta) = y^T Z(\eta, \sigma^2)^{-1} y$ and $h(\eta) = -\log\det Z(\eta, \sigma^2)$.

Hence the sequential convex optimization technique based on the majorization minimization principle applies.

Example

We consider an FIR system given by

$$y_k = \sum_{j=1}^{75} j e^{-0.2j} u(k-j) + e_k, \quad k \in \mathbf{N}_{200}$$

where e_k is a realizations of i.i.d. Gaussian random variables with zero mean and unit variance.

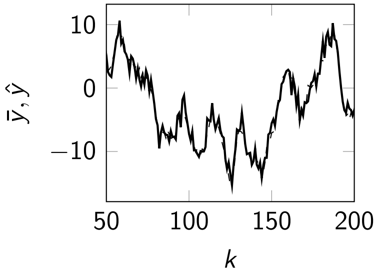
The input signal u_k is generated in a similar way. However, it is then low-pass filtered with a filter with cut-off frequency of 0.9.

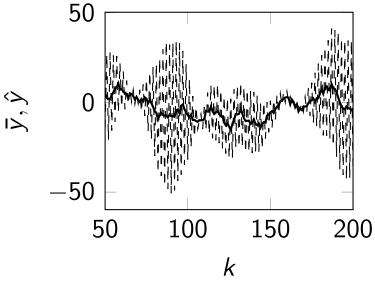
We estimate an FIR model $\hat{b}$ of order $n = 50$ using Empirical Bayes with the DC kernel and another FIR model without using Empirical Bayes.

Validation

We generate validation data $(\bar{u}, \bar{y})$ in the same way as we generated the data (u, y).

From $\bar{u}$ we compute $\hat{y}_k = \sum_{j=1}^{n} \hat{b}_j \bar{u}(k-j)$. This results in one $\hat{y}$ for the Empirical Bayes estimate and one for the estimate without using Empirical Bayes. We compare these $\hat{y}$ with one another and with $\bar{y}$ in the figure





Recurrent Neural Network (RNN)

Model the predictor (1–2) as

$$\begin{aligned} x_{k+1} &= F(x_k, u_k, \theta), \quad 0 \le k \le N-1 \\ \hat{y}_k &= G(x_k, u_k, \theta), \quad 0 \le k \le N \end{aligned}$$

where $F : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^q \to \mathbf{R}^n$ and $G : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^q \to \mathbf{R}^p$ are functions given by ANNs.

The vector θ contains all the parameters that specify the affine propagation functions of the ANN.

The word recurrent comes from the fact that the same ANN is used for each time step.

Often one-layer ANNs used to model F and G. Then it is possible to interpret the evolution of the state equation as a traditional ANN, for which the same weights are used in each layer. This interpretation is called *unrolling* of the RNN.

Challenges

The resulting optimization problems might become ill-conditioned.

In the learning community this is called *vanishing gradient* or *exploding gradient*.

A remedy to this is the so-called *Long Short-Term Memory* (LSTM) network.

Nonlinear ARX-Model

$$\hat{y}_{k+1} = f(y_k, \ldots, y_{k-n+1}, u_k, \ldots, u_{k-n+1}, \theta), \quad n-1 \leq k \leq N-1$$

where $f : \mathbf{R}^{2n} \times \mathbf{R}^q \to \mathbf{R}$ is a nonlinear function.

The vector θ contains all the parameters that define the predictor.

The idea in *Temporal Convolutional Networks* (TCN)s is to build up the function f in a tree structure, where each node in the tree is defined by a nonlinear ARX-model as well.

TCN

Let $x_k = (y_k, u_k)$ and let

$$\begin{aligned} \hat{y}_{k+1} &= f^{(L)}(Z_k^{(L-1)}) \\ z_k^{(l)} &= f^{(l)}(Z_k^{(l-1)}), \qquad l \in \mathbf{N}_{L-1} \\ z_k^{(0)} &= x_k \end{aligned}$$

where

$$Z_k^{(l-1)} = \left(z_k^{(l-1)}, z_{k-d_l}^{(l-1)}, \ldots, z_{k-(\bar{n}-1)d_l}^{(l-1)} \right)$$

and where d_l is the so-called *dilation factor*. Typically d_l increases exponentially with l, e.g. $d_l = 2^{l-1}$. We notice that $n - 1 = (\bar{n} - 1)d_L$ is the effective memory of the overall predictor.

Each function $f^{(l)}$ can be defined in many different ways. A very simple example would be to take it as a standard one-layer ANN.

Experiment Design

FIR model:

$$y(t) = b_0 u(t) + b_1 u(t-1) + \ldots + b_n u(t-n) + e(t), \quad 1 \leq t \leq N$$

Let

$$y = \begin{bmatrix} y(1) \\ \vdots \\ y(N) \end{bmatrix}; \quad \varphi(t) = \begin{bmatrix} u(t) \\ \vdots \\ u(t-n) \end{bmatrix}; \quad X = \begin{bmatrix} \varphi(1)^T \\ \vdots \\ \varphi(N)^T \end{bmatrix}; \quad e = \begin{bmatrix} e(1) \\ \vdots \\ e(N) \end{bmatrix}$$

and $\theta = \begin{bmatrix} b_0 & \cdots & b_n \end{bmatrix}^T$. Then

$$y = X\theta + e$$

where e realization of zero mean random variable with covariance $\sigma^2 I$.

Covariance function

Covariance $\bar{P} \in \mathbf{S}_{+}^{n+1}$ for the LS estimate:

$$\bar{P} = \sigma^2 \left(X^T X \right)^{-1}$$

Let $R_u : \mathbf{Z} \to \mathbf{R}$ be the covariance function for input signal $u(k)$.

From definition of X:

$$X^T X \approx N \begin{bmatrix} R_u(0) & \cdots & R_u(n) \\ \vdots & \ddots & \vdots \\ R_u(n) & \cdots & R_u(0) \end{bmatrix}$$

Our idea is to **find a covariance function** R_u for the input, which will give us a good $\bar{P}$.

Some definitions

Let $r = \begin{bmatrix} R_u(0) & \cdots R_u(n) \end{bmatrix}^T$ be a variable that we are going to find a good value for. Define

$$R(r) = N \begin{bmatrix} r_1 & \cdots & r_{n+1} \\ \vdots & \ddots & \vdots \\ r_{n+1} & \cdots & r_1 \end{bmatrix}$$

which is linear in r, and

$$P(r) = \sigma^2 R(r)^{-1}$$

Now $\bar{P} \approx P$ assuming that we are able to generate input with the covariance function defined by r.

Optimality criteria

The following criteria should be small:

A-optimality	$\mathbf{tr}\, P(r)$
D-optimality	$\log\det P(r)$
E-optimality	$\lambda_{\max} P(r)$
L-optimality	$\mathbf{tr}\, WP(r)$, where $W \in \mathbf{S}_{++}^{n+1}$

Then the confidence ellipsoid

$$\left\{\hat{\theta} \in \mathbf{R}^{n+1} \mid (\hat{\theta} - \theta_0)^T P(r)^{-1} (\hat{\theta} - \theta_0) \leq \alpha\right\}$$

will be small.

We limit the power in the experiment by $r_1 \leq L$. Then SNR is L/σ^2.

Not all r are covariance functions

Consider covariance functions from output of MA filter:

$$u(k) = c_0 v(k) + c_1 v(k-1) + \cdots + c_n v(k-n)$$

where $v(k)$ is white noise with unit variance.

Covariance function for u:

$$R_u(k) = Eu(l)u(l-k) = E\left(\sum_{i=0}^{n} c_i v(l-i) \sum_{j=0}^{n} c_j v(l-k-j)\right)$$

Notice $R_u(k) = 0$ for $k > n$. The z-transform of R_u:

$$\Phi_u(z) = \sum_{k=-n}^{n} R_u(k) z^{-k}$$

Require $\Phi_u(e^{i\omega}) \geq 0$ for all $\omega \in \mathbf{R}$ for r to be valid.

Postive real lemma

Let

$$A = \begin{bmatrix} 0 & 0 \\ I_{n-1} & 0 \end{bmatrix};\ B = \begin{bmatrix} 1 \\ 0 \end{bmatrix};\ C = \begin{bmatrix} r_2 & \cdots r_{n+1} \end{bmatrix}$$

and

$$K(Q, r) = \begin{bmatrix} Q - A^T QA & C^T - A^T QB \\ C - B^T QA & r_1 - B^T QB \end{bmatrix}$$

where $Q \in \mathbf{S}^n$. Then $\Phi_u(e^{i\omega}) \geq 0$ for all $\omega \in \mathbf{R}$ is equivalent to $K(Q, r) \succeq 0$, i.e being positive semidefinite.

The set of Q and r that satisfies $K(Q, r) \succeq 0$ is a convex set!

D-optimality

The function $\log\det P(r) = \log\det R(r)^{-1} + 2(n+1)\log\sigma$ is a convex function of r.

Optimal experiment design using D-optimality:

$$\begin{aligned} \text{minimize} \quad & -\log\det R(r) \\ \text{subject to} \quad & r_1 \leq L \\ & K(Q, r) \succeq 0 \end{aligned}$$

with variable r. Conic optimization problem that can be solved efficiently.

Convenient Matlab-interface: YALMIP.

All other optimality criteria also result in conic optimization problems.

Spectral factorization

The z-transform of R_u given by

$$\Phi_u(z) = \sum_{k=-n}^{n} R_u(k) z^{-k}$$

Since R_u is even function, we have $\Phi_u(z) = \Phi_u(1/z) \Rightarrow z_i$ is zero or pole of Φ_u so is $1/z_i$. Hence the following **spectral factorization** holds:

$$\Phi_u(z) = \kappa H(z) H(1/z)$$

where κ some constant, and where

$$H(z) = \frac{\prod_{i=1}^{n}(z - z_i)}{\prod_{i=1}^{n}(z - p_i)}$$

with $|z_i| \leq 1$ and $|p_i| \leq 1$.

Polynomial spectral factorization

Notice that $z^n \Phi_u(z)$ is a polynomial, and hence

$$z^n \Phi_u(z) = \kappa H(z) z^n H(1/z)$$

where right hand side is also polynomial. Take $p_i = 0$. With

$$C(z) = z^n + c_1 z^{n-1} + \cdots + c_n = \Pi_{i=1}^n (z - z_i)$$

we hence have the **polynomial spectral factorization**

$$z^n \Phi_u(z) = \kappa C(z) \tilde{C}(z)$$

where $\tilde{C}(z) = z^n C(1/z)$ with all its zeros outside the unit circle. Moreover, $H(z) = C(z)/z^n$.

Hence the MA-process

$$u(k) = v(k) + c_1 v(k-1) + \cdots + c_n v(k-n)$$

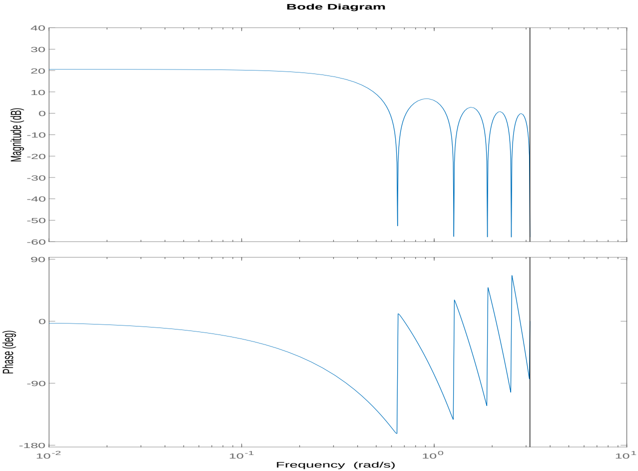
with $v(k)$ white noise with variance κ will have covariance function R_u.

Matlab function

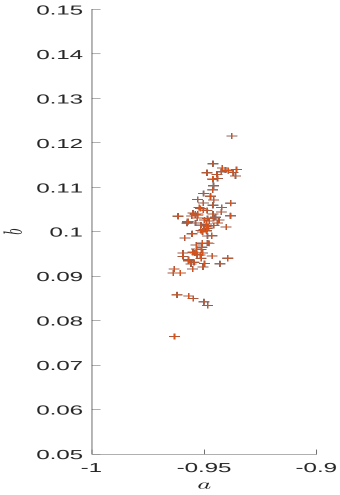
```
function [c,kappa] = specfac(r)
% covariance function to MA-conversion
% r contains the values of the covariance function:
% r = [R(0) R(1) ...  R(n)]
% c are cofficients of MA-filter
% kappa is variance of white noise input to MA-filter
r = [r(end:-1:2) r];
zero = roots(r);
ind = find(abs(zero)<1);
c = poly(zero(ind));
tildec = c(end:-1:1);
kappa = polyval(r,1)/polyval(c,1)/polyval(tildec,1);
```

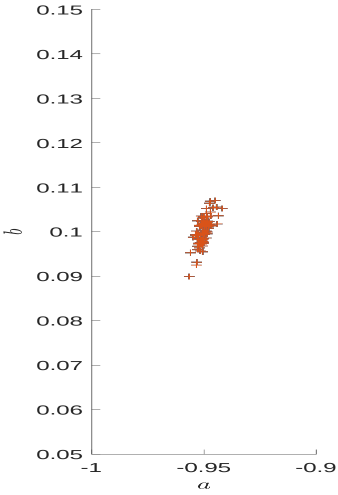
OE model example: D-optimality

$$y(t) + ay(t-1) = bu(t) + e(t) + ae(t-1)$$



Scatter plot for 100 estimates





Left: white noise input

Right: D-optimal input