Gdańsk University of Technology

System Identification

by

Maciej Niedźwiecki

Gdańsk University of Technology

Departament of Automatic Control

Gdańsk, Poland

Motto (George E.P. Box):

"Essentially all models are wrong,
but some are useful"

Basic notions
Podstawowe pojęcia

system/process / obiekt/proces

$$y(t) = f\left[y(i), u(i), n(i), i < t\right], \quad f[\cdot] = ?$$

where:

t	-	time / czas
$y(t)$	-	measurable output signal / mierzalny sygnał wyjściowy
$u(t)$	-	measurable input signal / mierzalny sygnał wejściowy
$n(t)$	-	unmeasurable input noise / niemierzalny szum wejściowy

mathematical system model /
matematyczny model obiektu

mathematical model is a mathematical description of how the system works, often taking the form

$$\mathcal{M} = \{s(\cdot), \boldsymbol{\theta}\}$$

where $s(\cdot)$ denotes the system equation and $\boldsymbol{\theta}$ denotes the vector of system parameters that appear in this equation.

"Physical" approach to modeling
"Fizyczne" podejście do modelowania

This is an analytic approach. Basic rules of physics (such as Newton's laws, conservation laws and balance equations) are used to describe the dynamic behavior of a phenomenon or a process. Physical models are usually continuous time, nonlinear and fairly complex

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, u, n, \boldsymbol{\theta})$$
$$y = \mathbf{g}(\mathbf{x}, u, n, \boldsymbol{\theta})$$

Physical model can be very useful as a system "substitute" in realistic simulations. Many industrial companies have developed very accurate simulation models of their plants, e.g. planes, ships,missiles, engines, mills, furnances. So accurate are some of these models that the new designs can be tested by means of computer simulations only. The Boeing 777 was the first civil aircraft belonging to this new generation of computer-verified constructions; the entire design was concieved "behind the desk" without conducting costly experiments in aerodynamic tunnels (the same models are used in flight simulators).

But ... physical models, because of their large complexity, are not very helpful when it comes to solving practical problems.

"Instrumental" approach to modeling
"Instrumentalne" podejście do modelowania

This is an entirely different approach to model building, based on process identification: the form of the model is to some extent arbitrary and its coefficients are determined experimentally using statistical procedures similar to curve fitting. Instrumental models are not phenomenologically justified and therefore their coefficients have no physical significance. However, they have some obvious practical advantages: they are easy to build and update without any physical insights and, more importantly, due to their relative simplicity they allow mathematically tractable formulations and then solutions for many practically important problems such as adaptive prediction, predictive coding of signals, adaptive filtering, spectrum estimation, change detection and adaptive control. These solutions are usually known as model-based solutions.

The model-based approach is the pragmatic one. In this framework, building a mathematical model of a dynamic process is not a goal in itself; models are only the means of solving practical problems in the sense that the corresponding prediction, filtering and control formulas depend explicitly on model coefficients.

Example of an instrumental model
Przykład modelu instrumentalnego

After linearization of a nonlinear system description around a certain set point, followed by time discretization of the obtained set of differential equations, one may arrive at the following approximate linear discrete-time system description

$$y(t) = \sum_{i=1}^{r} a_i y(t-i) + \sum_{i=1}^{p} b_i u(t-i) + e(t)$$

where:

- $t = \ldots, -1, 0, 1, \ldots$ denotes normalized (dimensionless) discrete time
- $a_1, \ldots, a_r$ and $b_1, \ldots, b_p$ denote the real-valued coefficients which make up the vector of model parameters $\boldsymbol{\theta}$
- r and p denote the integer-valued model orders which determine the model structure
- $e(t)$ denotes the modeling error due to the applied approximations and/or external disturbances

System identification
Identyfikacja obiektu

System identification is an experimental approach. Some experiments are performed on the system. Then a model is fitted to the recorded data by assigning suitable numerical values to its parameters.

The process of building a mathematical model based on experimental data and (possibly) on some prior information about the modeled system, is called *system identification*.

data / dane

$$t = 1, \ldots, N$$

$$\mathcal{Y}(N) = \{y(1), \ldots, y(N)\}$$
$$\mathcal{U}(N) = \{u(1), \ldots, u(N)\}$$
$$\Omega(N) = \{\mathcal{Y}(N), \mathcal{U}(N)\}$$

prior information / informacja wstępna (*a priori*)

$$r = 10, \quad p > 4$$
$$a_1 = 0.5, \quad a_2 = 0.01v^2$$
$$b_1 \in [1.0, 1.5]$$
$$\pi(b_2) = \mathcal{N}(0.2, 0.05)$$

Selected applications of process identification

Wybrane zastosowania identyfikacji procesów

- prediction, forecasting (predykcja, prognozowanie)
- predictive coding of signals (predykcyjne kodowanie sygnałów)
- spectrum estimation (estymacja widma)
- equalization of communication channels (korekcja kanałów telekomunikacyjnych)
- elimination of impulsive disturbances (usuwanie zakłóceń impulsowych)
- active noise and vibration suppression (aktywne tłumienie hałasu i drgań)
- adaptive control (sterowanie adaptacyjne)

Autoregressive (AR) modeling of signals
Autoregresyjne modelowanie sygnałów

$$\mathrm{AR}(r): \quad y(t) = \sum_{i=1}^{r} a_i y(t-i) + n(t)$$

where r is the order of autoregression (rząd autoregresji), $a_1, \ldots, a_r$ denote autoregressive coefficients (współczynniki autoregresji) and $\{n(t)\}$ denotes zero-mean white driving noise (szum tworzący) - the sequence of independent random variables with zero mean $\mathrm{E}[n(t)] = 0$ and identical variance $\mathrm{var}[n(t)] = \sigma_n^2$:

$$\mathrm{E}[\, n(t)n(s)\,] = \begin{cases} \sigma_n^2 & \text{if} \quad t = s \\ 0 & \text{if} \quad t \neq s \end{cases}$$

Let q^{-1} denote the backward shift operator: $q^{-1}y(t) = y(t-1)$. Then the AR equation can be rewritten in the form

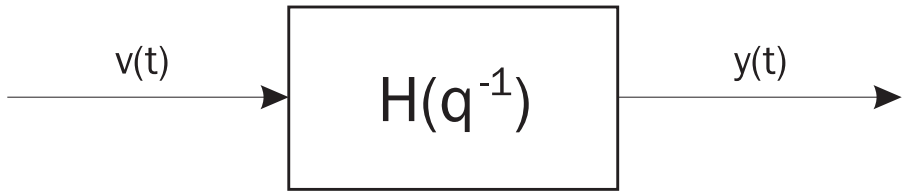
$$A(q^{-1})y(t) = n(t), \quad A(q^{-1}) = 1 - \sum_{i=1}^{r} a_i q^{-i}$$

i.e., $\{y(t)\}$ is the process observed at the output of a linear all-pole filter

$$H(q^{-1}) = \frac{1}{A(q^{-1})} = \frac{1}{1 - \sum_{i=1}^{r} a_i q^{-i}}$$

excited by white noise $\{n(t)\}$.

Forming (shaping) filter
Filtr formujący



The second-order characteristics of an autoregressive process depend solely on the values of autoregressive coefficients $a_1, \ldots, a_r$ and the driving noise variance σ_n^2. Denote by $q_1, \ldots, q_r$ the poles of the forming filter $H(q^{-1})$, i.e. the roots of the *characteristic polynomial* $A(q^{-1})$:

$$A(q_k^{-1}) = 1 - \sum_{i=1}^{r} a_i q_k^{-i} = 0, \qquad k = 1, \ldots, r.$$

The necessary and sufficient condition for the asymptotic weak stationarity of an AR(r) process takes the following form:

stability/stationary condition

All zeros of $A(q^{-1})$, i.e., all poles of $H(q^{-1})$, lie inside the unit circle in the complex plane:

$$|q_k| < 1, \qquad k = 1, \ldots, r.$$

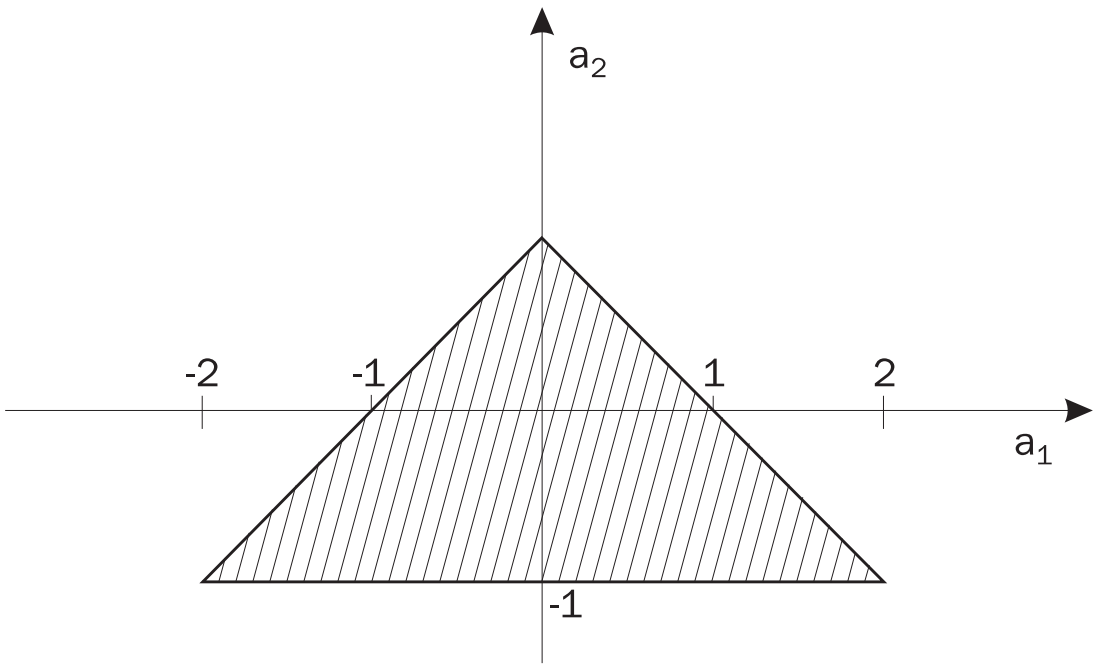
Example
Przykład

Consider a second-order autoregressive process governed by

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + n(t).$$

Here the asymptotic stationarity condition takes the following form:

$$a_1 + a_2 \leq 1,$$
$$a_2 - a_1 \leq 1,$$
$$|a_2| \leq 1.$$



Stability region (shaded) for the second-order AR process in terms of autoregressive coefficients.

Autocorrelation function of an AR process
Funkcja autokorelacji procesu AR

When the stability condition is fulfilled and the forming filter $H(q^{-1})$ is excited in the 'infinite past', autocorrelation function of an AR(r) process is, irrespective of the initial conditions, translation invariant:

$$\mathrm{E}[y(t)y(t-\tau)] = R_y(\tau).$$

The recursive expression for computation of $R_y(\tau)$ can be easily obtained by multiplying both sides of the AR equation by $y(t-\tau)$ and taking expectations:

$$\mathrm{E}[\,y(t)y(t-\tau)\,] = \sum_{i=1}^{r} a_i \mathrm{E}[\,y(t-i)y(t-\tau)\,] + \mathrm{E}[\,n(t)y(t-\tau)\,]$$

$$\underline{\tau > 0:}$$

Since the random variable $n(t)$ is uncorrelated with past signal values $y(t-i), i > 0$, one arrives at

$$R_y(\tau) = \sum_{i=1}^{r} a_i R_y(\tau - i),$$

or alternatively

$$A(q^{-1})R_y(\tau) = 0.$$

Autocorrelation function of an AR process
Funkcja autokorelacji procesu AR

$\underline{\tau = 0:}$

It holds that $\mathrm{E}[\,n(t)y(t)\,] = \sigma_n^2$ and therefore, since $R_y(\tau) = R_y(-\tau)$,

$$R_y(0) = \sum_{i=1}^{r} a_i R_y(-i) + \sigma_n^2 = \sum_{i=1}^{r} a_i R_y(i) + \sigma_n^2$$

$\underline{\tau < 0 \quad ?}$

The initial conditions $R_y(0), \ldots, R_y(r)$ can be determined as functions of autoregressive coefficients $a_1, \ldots, a_r$ and the driving noise variance σ_n^2 by solving the set of autocorrelation equations corresponding to $\tau = 0, 1, \ldots, r$:

$$\begin{aligned}
R_y(0) &= a_1 R_y(1) + a_2 R_y(2) + \ldots + a_r R_y(r) + \sigma_n^2 \\
R_y(1) &= a_1 R_y(0) + a_2 R_y(1) + \ldots + a_r R_y(r-1) \\
&\vdots \\
R_y(r) &= a_1 R_y(r-1) + a_2 R_y(r-2) + \ldots + a_r R_y(0)
\end{aligned}$$

For $\tau > r$ the autocorrelation coefficients can be computed recursively.

Evaluation of an autocorrelation function
Wyznaczanie funkcji autokorelacji

$$\{a_1, \ldots, a_r, \sigma_n^2\} \longrightarrow \{R_y(0), \ldots, R_y(r)\}$$
$$\longrightarrow \{R(\tau), \tau > r\}$$

$$\begin{bmatrix} 1 & -a_1 & \ldots & -a_r \\ -a_1 & 1-a_2 & \ldots & 0 \\ -a_2 & -a_1 & \ldots & 0 \\ \vdots & & & \vdots \\ -a_r & -a_{r-1} & \ldots & 1 \end{bmatrix} \begin{bmatrix} R_y(0) \\ R_y(1) \\ R_y(2) \\ \vdots \\ R_y(r) \end{bmatrix} = \begin{bmatrix} \sigma_n^2 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Since the obtained equations are linear with respect to initial conditions, the solution is straightforward.

$$R_y(\tau) = \sum_{i=1}^{r} a_i R_y(\tau - i), \quad \tau > r$$

example / przykład

$$\mathrm{AR}(1): \quad y(t) = ay(t-1) + n(t)$$

$$R_y(0) = aR_y(1) + \sigma_n^2, \quad R_y(1) = aR_y(0)$$
$$R_y(0) = \frac{\sigma_n^2}{1-a^2}, \quad R_y(\tau) = R_y(0)a^\tau, \ \tau > 0$$

Yule-Walker equations
Równania Yule-Walkera

The same set of linear equations, known in a slightly rearranged form as Yule–Walker equations, can also be used to solve an inverse problem: determining the parameters of an autoregressive process from the known autocorrelation coefficients.

$$\{R_y(0), \ldots, R_y(r)\} \longrightarrow \{a_1, \ldots, a_r, \sigma_n^2\}$$

$$\begin{bmatrix} R_y(0) & R_y(1) & \ldots & R_y(r-1) \\ R_y(1) & R_y(0) & \ldots & R_y(r-2) \\ R_y(2) & R_y(1) & \ldots & R_y(r-3) \\ \vdots & & & \vdots \\ R_y(r-1) & R_y(r-2) & \ldots & R_y(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_r \end{bmatrix}$$

$$= \begin{bmatrix} R_y(1) \\ R_y(2) \\ R_y(3) \\ \vdots \\ R_y(r) \end{bmatrix}$$

$$\sigma_n^2 = R_y(0) - \sum_{i=1}^{r} a_i R_y(i)$$

Autocorrelation function of an AR process
Funkcja autokorelacji procesu AR

When all poles of the forming filter are single, which is a typical situation in practice, the autororrelation function of an AR(r) process can be explicitly written down in the form

$$R_y(\tau) = \sum_{i=1}^{r} A_i q_i^{\tau}, \qquad \tau \geq 0,$$

where $A_1, \ldots, A_r$ is the set of parameter-dependent coefficients.

The contribution of real system poles takes the form of damped exponentials, i.e. terms that decay exponentially to zero.

All complex poles appear in conjugate pairs – if $\mathrm{Im}(q_i) \neq 0$ both q_i and $q_i^{\star}$ must be the roots of $A(q^{-1})$. Such poles yield terms of a damped sinusoidal nature:

$$A_i q_i^{\tau} + A_i^{\star} (q_i^{\star})^{\tau} = \alpha_i \rho_i^{\tau} \cos(\omega_i \tau + \beta_i),$$

where $\alpha_i = 2|A_i|$, $\beta_i = \arg(A_i)$, $\rho_i = |q_i|$ and $\omega_i = \arg(q_i)$.

Power spectrum of an AR process
Widmo mocy procesu AR

Power spectrum (power spectral density function / funkcja widmowej gęstości mocy) of a wide sense stationary process is defined as a Fourier transform of its autocorrelation function

$$S_y(\omega) = \mathcal{F}[R_y(\tau)] = \sum_{\tau=-\infty}^{\infty} R_y(\tau) e^{-j\omega\tau}, \quad \omega \in [-\pi, \pi]$$

It shows distribution of the signal power (mean energy), equal to σ_n^2, over different frequency bands.

Notation:

$$f = \frac{f_c}{F_s} \in \left[-\frac{1}{2}, \frac{1}{2}\right], \quad \omega = 2\pi f = \frac{\omega_c}{F_s} \in [-\pi, \pi]$$

f_c	-	frequency of a continuous-time signal expressed in cycles per second [Hz]
ω_c	-	angular frequency of a continuous-time signal expressed in radians per second [rad/sec]
F_s	-	sampling frequency / częstotliwość próbkowania
f	-	normalized frequency / unormowana częstotliwość expressed in cycles per sample [c/sa]
ω	-	normalized angular frequency / unormowana pulsacja, expressed in radians per sample [rad/sa]

Power spectrum of an AR process
Widmo mocy procesu AR

Theorem

Consider a discrete-time, linear, time-invariant and stable system characterized by the transfer function $H(q^{-1})$ excited by a wide-sense stationary signal $\{x(t)\}$ with power spectrum $S_x(\omega)$

$$y(t) = \sum_{i=0}^{\infty} h(i)x(t-i), \quad H(e^{-j\omega}) = \sum_{s=0}^{\infty} e^{-j\omega s}h(s)$$

Then the output signal $\{y(t)\}$ is also asymptotically wide-sense stationary and its power spectrum can be expressed in the form

$$S_y(\omega) = |H(e^{-j\omega})|^2 S_x(\omega).$$

Proof

Since it holds that

$$S_x(\omega) = \sum_{\tau=-\infty}^{\infty} R_x(\tau)e^{-j\omega\tau}$$

and

$$R_y(\tau) = \sum_{s_1=0}^{\infty}\sum_{s_2=0}^{\infty} h(s_1)h(s_2)R_x(\tau - s_2 + s_1)$$

one obtains

$$
\begin{aligned}
S_y(\omega) &= \sum_{\tau=-\infty}^{\infty} R_y(\tau) e^{-j\omega\tau} \\
&= \sum_{\tau=-\infty}^{\infty} e^{-j\omega\tau} \sum_{s_1=0}^{\infty} \sum_{s_2=0}^{\infty} h(s_1) h(s_2) R_x(\tau - s_2 + s_1) \\
&= \sum_{s_1=0}^{\infty} e^{j\omega s_1} h(s_1) \sum_{s_2=0}^{\infty} e^{-j\omega s_2} h(s_2) \\
&\qquad \times \left[\sum_{\tau=-\infty}^{\infty} e^{-j\omega(\tau - s_2 + s_1)} R_x(\tau - s_2 + s_1) \right] \\
&= H(e^{j\omega}) H(e^{-j\omega}) S_x(\omega) = \left| H(e^{-j\omega}) \right|^2 S_x(\omega)
\end{aligned}
$$

∎

Since the spectral density of white noise is

$$
S_n(\omega) = \sigma_n^2, \qquad \omega \in [-\pi, \pi],
$$

the power spectrum of an autoregressive process can be expressed in the form

$$
S_y(\omega) = \frac{\sigma_n^2}{|A(e^{-j\omega})|^2} = \frac{\sigma_n^2}{|1 - \sum_{i=1}^{r} a_i e^{-j\omega i}|^2} \ .
$$

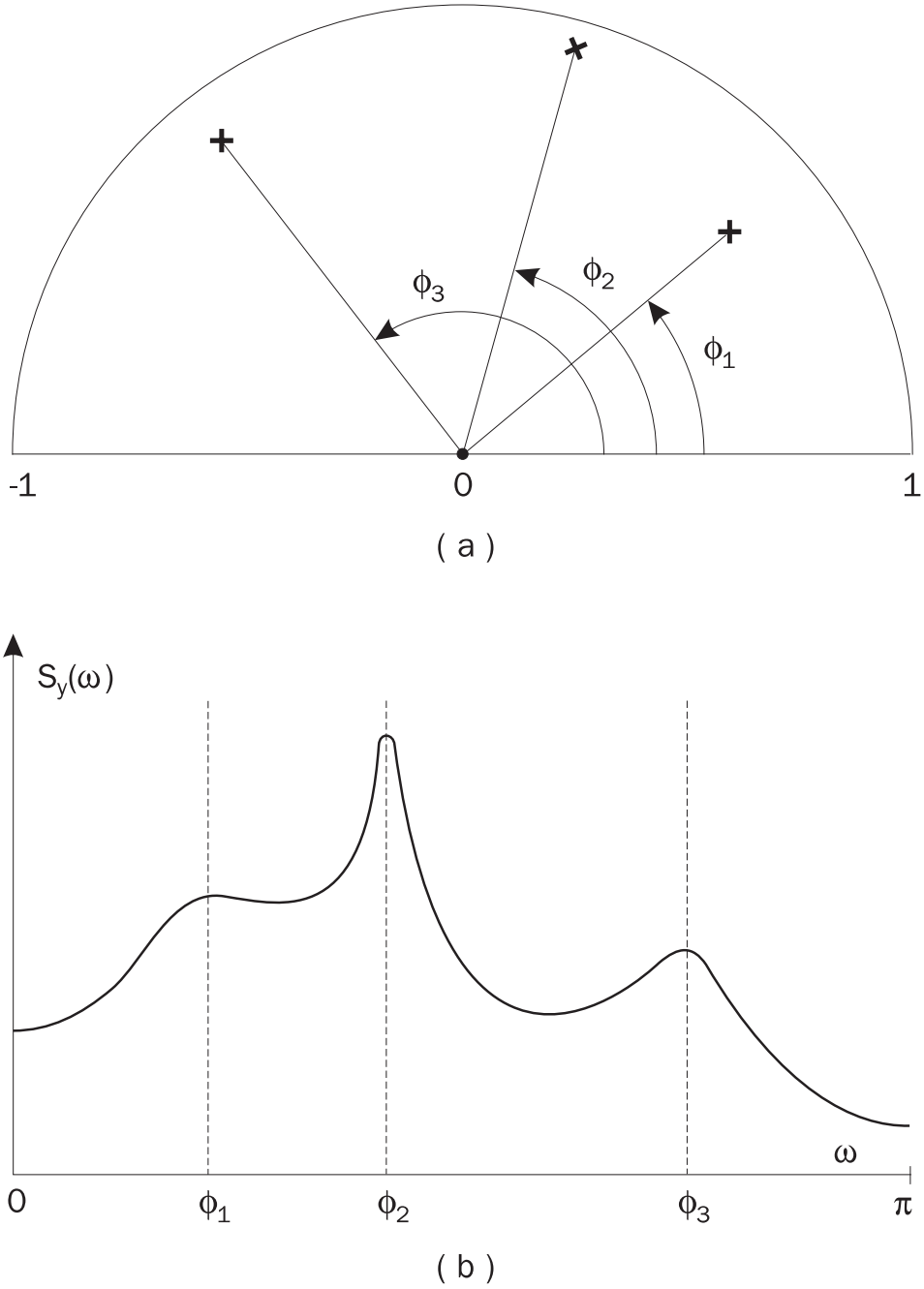
Power spectrum of an AR process

The resonant structure of $S_y(\omega)$ – the number and location of spectral peaks characterizing different frequency components of the analyzed signal – depends on the number and localization of complex poles of the forming filter $H(q^{-1})$.

For well-separated complex poles that are sufficiently close to the unit circle, the following rules of thumb can be used for *qualitative* shape evaluation of the spectral density function:

- Each pair of complex conjugate poles can introduce one spectral peak (resonance).
- The angular frequency coordinate of a spectral peak is approximately equal to the phase angle of the corresponding pole.
- The height of a spectral peak is inversely proportional to the distance from the corresponding pole to the unit circle.

These 'rules' should be used with caution: several complex poles clustered together may create a single spectral peak, and poles with small radii may not create any peak at all. However, irrespective of the pole distribution, the number of spectral resonances may not exceed $r/2$, i.e. half the order of autoregression.



Approximate relationship between the location of noise shaping filter poles in the complex plane (a) and the shape of the spectral density function (b) of the corresponding AR process.

Autoregressive system
Obiekt autoregresyjny

ARX - autoregressive with exogenous input

$$\mathrm{ARX}(r,p): \; y(t) = \sum_{i=1}^{r} a_i y(t-i) + \sum_{i=1}^{p} b_i u(t-i) + n(t)$$

where $u(t)$ denotes an observable input signal.

Shorthand notation:

$$A(q^{-1})y(t) = B(q^{-1})u(t) + n(t)$$

where

$$A(q^{-1}) = 1 - \sum_{i=1}^{r} a_i q^{-i}, \quad B(q^{-1}) = \sum_{i=1}^{p} b_i q^{-i}$$

Please note that $\{n(t)\}$ is the *input* noise, not the *measurement* noise. In the latter case system description would be

$$x(t) = \sum_{i=1}^{r} a_i x(t-i) + \sum_{i=1}^{p} b_i u(t-i)$$

$$y(t) = x(t) + n(t)$$

where $x(t)$ denotes the noiseless system output.

Applications of process identification
Zastosowania identyfikacji procesów

1) Prediction/forecasting / Predykcja/prognozowanie

What is predicted and why?

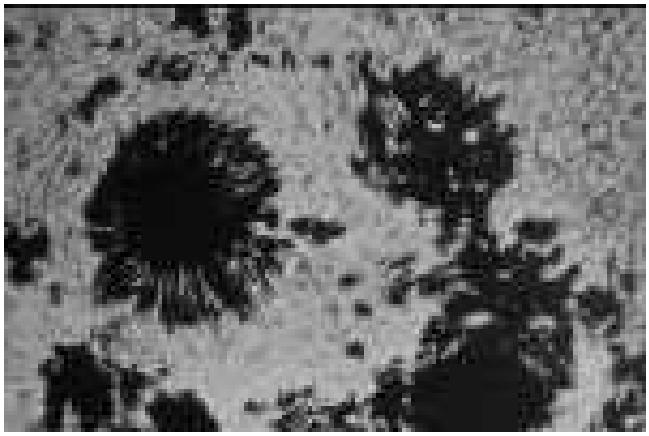
- number of sunspots (liczba plam na Słońcu)
- animal populations (populacje zwierząt)
- rainfall, river flows (opady, pływy rzek)
- econometric figures (wskaźniki ekonometryczne)
- electric power demand (zapotrzebowanie na energię elektryczną)

We usually predict for *control* purposes.

About sunspots
O plamach na Słońcu

Sunspots are seen as small dark spots on the surface of the sun. They are easy to observe and count if the sunlight is strongly filtered. They were first noticed (in Western record) in the year 325 BC by Theophrastus, an Hellenic scientist, and they have been counted on a regular basis since the middle of the 17th century. They come and go in cycles which average about 11 years.

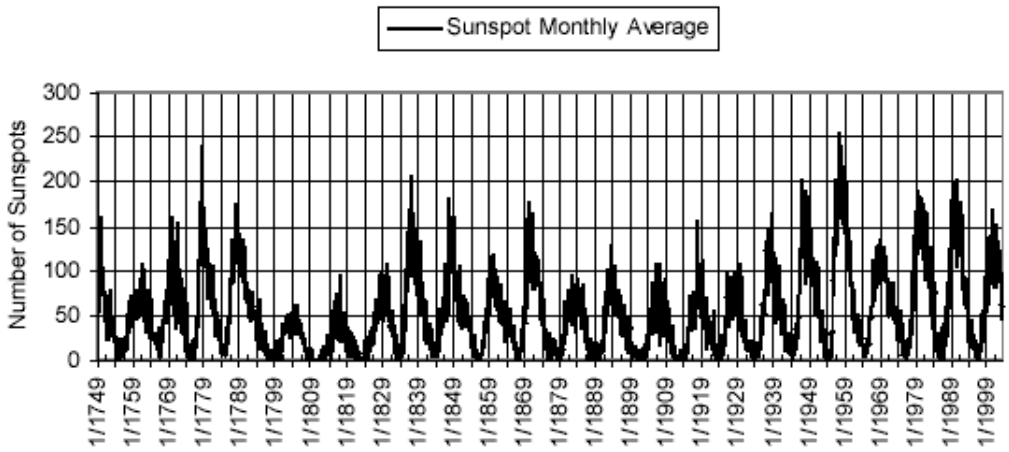
These small dark spots are conceived by modern astrophysicists to be intense bubbles of magnetic energy which somehow cool down the hot gasses within so that they appear dark compared with the surrounding solar atmosphere. These cool bubbles are not really very small, they are quite often the size of the Earth and many times giant spots many times the Earth can be seen. Many more physical facts about them can be can be found on the NASA and NOAA websites, such as at http://www.sec.noaa.gov/primer/primer.html



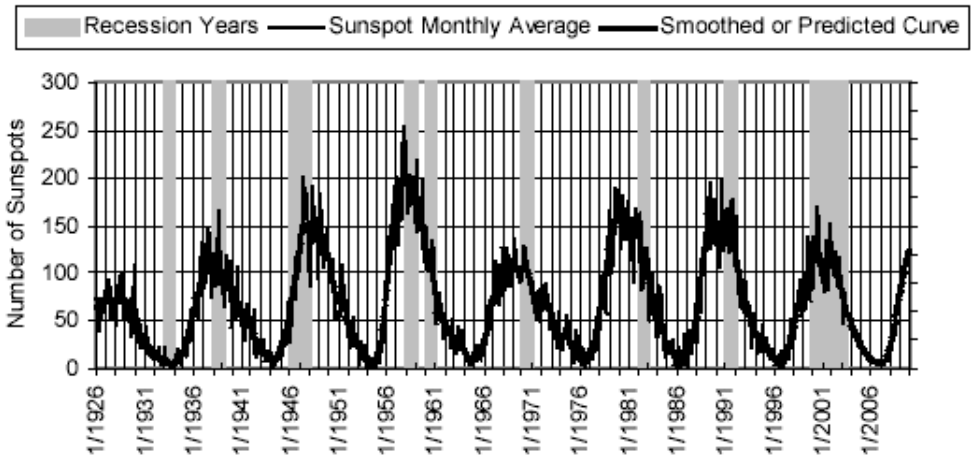
Does it pay to be a scientist ?
Czy opłaca się być naukowcem ?

A. L. Tchijevsky, a Russian professor of Astronomy and Biological Physics, noticed during World War I that particularly severe battles followed solar flares. Since the sunspots were in a peak period during 1916-17, no doubt the war and its various battles were heavily stimulated by the energies which are boiling off the Sun. Intrigued by the connection of human behavior to solar physics, Tchijevsky constructed an Index of Mass Human Excitability. He compiled the histories of 72 countries from 500 BC to 1922 AD to provide a strong database to articulate his correlations. After rating the most significant events, Tchijevsky found that fully 80 of the most significant human events, mostly related to war and violence, occurred during the 5 years or so of maximum sunspot activity.

Tchijevsky went on to observe that the 1917 Russian Revolution occurred during the height of Sunspot Cycle. Unfortunately, this was one of sciences most costly observations, it earned Tchijevsky almost 30 years in Soviet prisons because his theory challenged Marxist dialectics ...



Compiled from the International Sunspot Count by MWM, 2002. Most major human conflicts are easily correlated to the peaks in the count of sunspots and so are major weather disturbances, including severe droughts and floods. MichaelMandeville.com



Compiled from the International Sunspot Count by MWM, 2003. Predicted values 2003-2009 are by the IPS Agency of Australia as of July, 2003. MichaelMandeville.com

Optimal one-step-ahead prediction
Optymalna predykcja jednokrokowa

Suppose that the signal $\{y(t)\}$ is governed by the AR(r) model

$$y(t) = \sum_{i=1}^{r} a_i y(t-i) + n(t)$$

where $\{n(t)\}$ denotes zero-mean white noise.

Find such one-step-ahead prediction $\widehat{y}(t+1|t) = f[\mathcal{Y}(t)]$, based on past signal measurements $\mathcal{Y}(t) = \{y(1), \ldots, y(t)\}$, which minimizes the mean-square prediction error

$$\mathrm{E}\left\{[y(t+1) - \widehat{y}(t+1|t)]^2\right\} \longrightarrow \min$$

Since the random variable $n(t+1)$ is "unpredictable"

$$\widehat{y}(t+1|t) = \sum_{i=1}^{r} a_i y(t+1-i)$$

$$\mathrm{E}\left\{[y(t+1) - \widehat{y}(t+1|t)]^2\right\} = \mathrm{E}[n^2(t+1)] = \sigma_n^2$$

<u>Example</u>

$$y(t) = ay(t-1) + n(t) \Longrightarrow \widehat{y}(t+1|t) = ay(t)$$

Optimal multi-step-ahead prediction
Optymalna predykcja wielokrokowa

Optimal k-step-ahead prediction is a concatenation of k one-step-ahead predictions

$$\widehat{y}(t+k|t) = \sum_{i=1}^{r} a_i \widehat{y}(t+k-i|t)$$

where

$$\widehat{y}(t+k-i|t) = y(t+k-i) \quad \text{for} \quad i \geq k$$

<u>Example</u>

When

$$y(t) = ay(t-1) + n(t)$$

it holds that

$$y(t+2) = a^2 y(t) + n(t+2) + an(t+1)$$

leading to

$$\widehat{y}(t+2|t) = a\widehat{y}(t+1|t) = a^2 y(t)$$

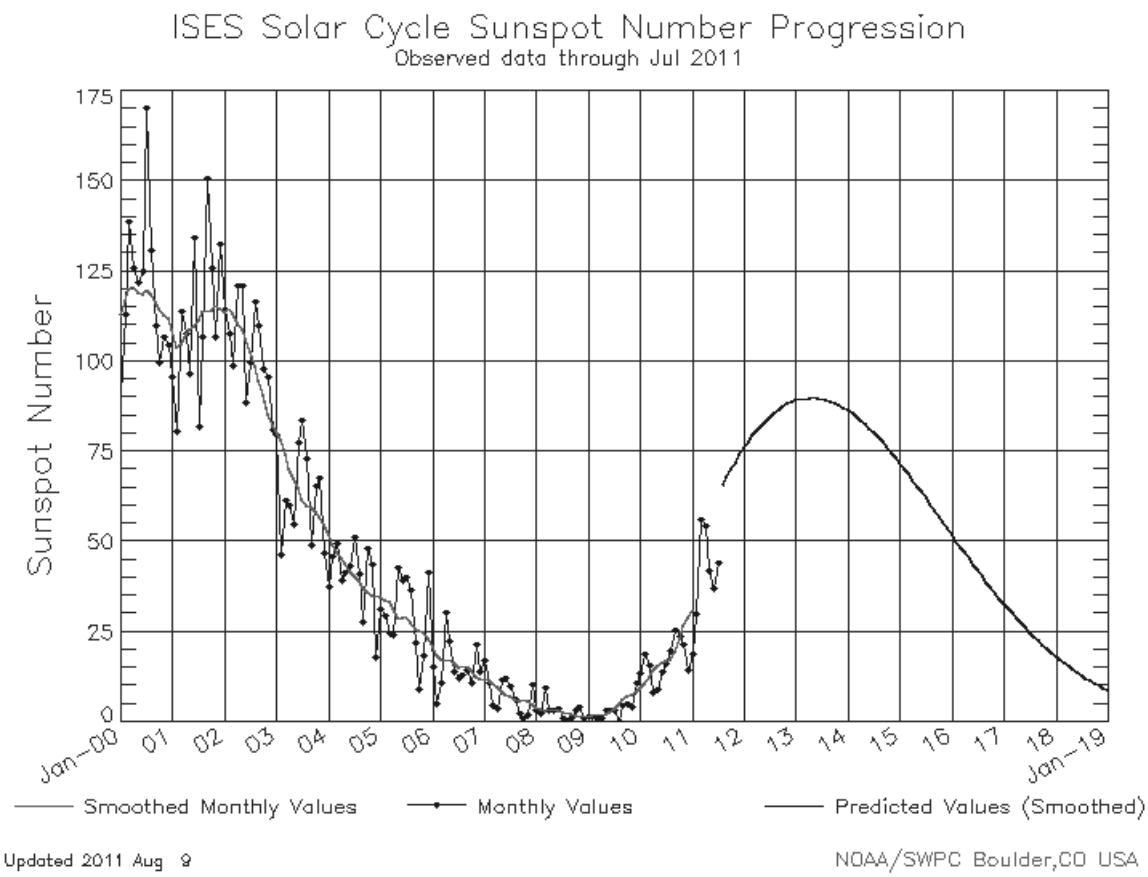
$$\begin{aligned} &\mathrm{E}\left\{[y(t+2) - \widehat{y}(t+2|t)]^2\right\} \\ &\qquad = \mathrm{E}\left\{[n(t+2) + an(t+1)]^2\right\} = (1+a^2)\sigma_n^2 \end{aligned}$$

Optimal multi-step-ahead prediction
Optymalna predykcja wielokrokowa

Generally, for the AR(1) model, one obtains

$$\widehat{y}(t+k|t) = a^k y(t)$$

$$\begin{aligned} \mathrm{E}\left\{[y(t+k) - \widehat{y}(t+k|t)]^2\right\} \\ &= \left[1 + a^2 + \ldots + a^{2(k-1)}\right] \sigma_n^2 \\ &= \frac{1 - a^{2k}}{1 - a^2} \sigma_n^2 \end{aligned}$$



Example of a multi-step-ahead prediction (sunspot numbers)

Prediction for autoregressive systems / Predykcja dla obiektów autoregresyjnych

$$y(t) = \sum_{i=1}^{r} a_i y(t-i) + \sum_{i=1}^{p} b_i u(t-i) + n(t)$$

$$\Downarrow$$

$$\widehat{y}(t+1|t) = \sum_{i=1}^{r} a_i y(t+1-i) + \sum_{i=1}^{p} b_i u(t+1-i)$$

Adaptive prediction / Predykcja adaptacyjne

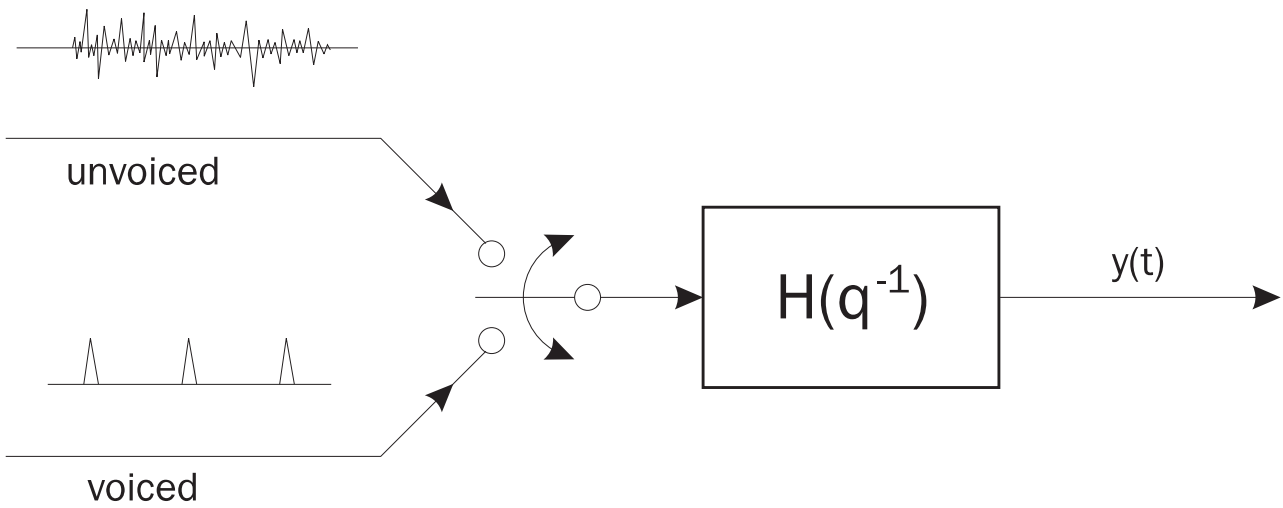
$$\widehat{y}(t+1|t) = \sum_{i=1}^{r} \widehat{a}_i(t) y(t+1-i) + \sum_{i=1}^{p} \widehat{b}_i(t) u(t+1-i)$$

where $\widehat{a}_1(t), \ldots, \widehat{a}_r(t)$ and $\widehat{b}_1(t), \ldots, \widehat{b}_r(t)$ denote current estimates of system coefficients

Applications of process identification
Zastosowania identyfikacji procesów

2) Predictive coding of signals
Predykcyjne kodowanie sygnałów

Voiced speech is produced by exciting the vocal tract (consisting of pharynx and mouth cavity) with quasi-periodic glottal air pulses generated by the vibrating vocal chords. Unvoiced speech is produced by forcing air through a constriction in the vocal tract. the position of speech articulators: velum, tongue, teeth and Physical models of speech production are quite complex and involve a set of partial differential equations based on acoustic theory and low-viscosity (air) fluid mechanics. Despite this, a relatively simple instrumental model of speech generation is successfully used in practice – a linear (slowly time-varying) all-pole filter, representing the vocal tract, excited by a periodic train of pulses (for voiced sounds) or white noise (for unvoiced sounds).



A simple speech production model used for adaptive speech coding.

Principles of LPC coding
Zasady kodowania w systemie LPC

LPC – linear predictive coding

$$H(q^{-1}) = \frac{1}{A(q^{-1})}\,, \quad A(q^{-1}) = 1 - \sum_{i=1}^{r} a_i q^{-i}$$

Since there are up to 5 speech formants, one should pick $r \geq 10$. Under telephone-quality sampling ($F_s \leq$ 10 kHz) the typical choice is $r \in [10, 15]$.

A) Differential coding system (ADPCM)
System kodowania różnicowego

- Divide speech signals into a sequence of frames of length N, each containing approximately 20 ms of speech material (for example, under 10 kHz sampling, one gets N=10 kHz $\times$ 20 ms = 200 Sa).

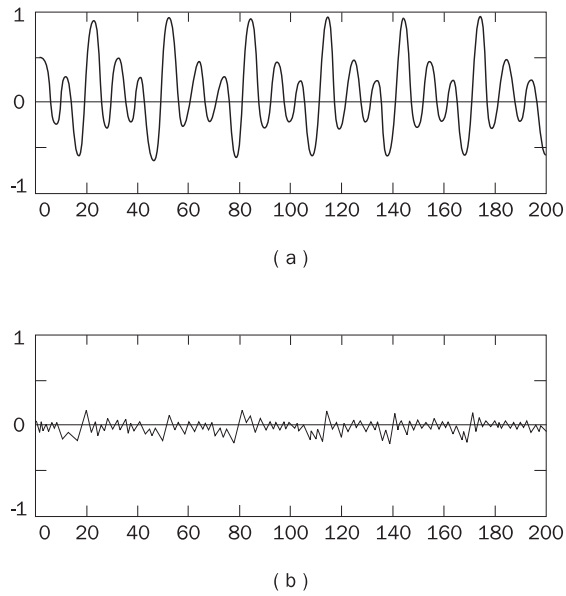
- Fit the AR(r) model to each frame and use the obtained estimates $\widehat{a}_1(N), \ldots, \widehat{a}_r(N)$ to generate residuals

$$e(t) = y(t) - \sum_{i=1}^{r} \widehat{a}_i(N) y(t-i), \quad t = 1, \ldots, N$$

- Send to the receiver the information about filter coefficients $\widehat{a}_1(N), \ldots, \widehat{a}_r(N)$ along with the quantized residual errors

$$e_q(t) = Q[e(t)], \quad t = 1, \ldots, N$$

Since the residuals are usually much smaller than the original signal, they can be encoded using a smaller number of bits per second.



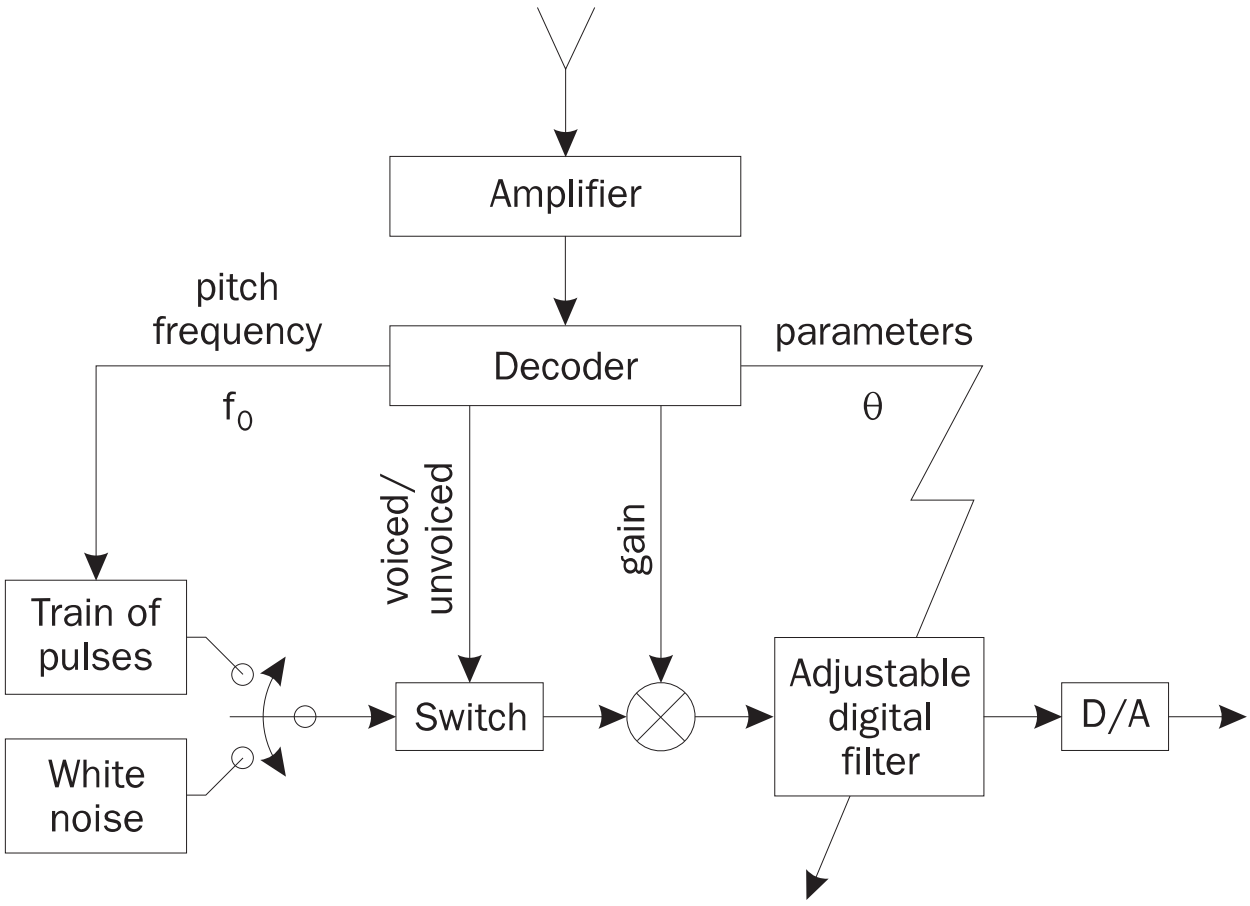
A fragment of a speech signal (a) and the corresponding sequence of residuals obtained by inverse filtering (b).

- At the receiver, synthesize speech signal using inverse filtering

$$\widehat{y}(t) = \sum_{i=1}^{r} \widehat{a}_i(N)\widehat{y}(t-i) + e_q(t), \quad t = 1, \ldots, N$$

As initial values $\widehat{y}(0), \ldots, \widehat{y}(1-r)$ use the last r samples of the signal synthesized in the preceding frame.

B) Full LPC coding
System pełnego kodowania LPC



Block diagram of the receiver in the system with full LPC coding.

In this system the excitation signal $e(t)$ is generated at the receiver based on the information about tonality of the frame (voiced/unvoiced), pitch frequency (for voiced frames only) and signal strength (gain coefficient).

All the remaining steps of signal encoding/decoding are the same as those incorporated in the differential coding system.

C) Code excited linear prediction system (CELP)
System pełnego kodowania LPC

The standard adopted in 1991 for the GSM (Global System for Mobile Communication) transmission system.

- Vocabulary of typical filter coefficients
- Vocabulary of typical excitation signals

In this system, receiver obtains information about the code number of the best approximating filter and the code number of the most suitable excitation. Encoding is optimized using Shannon's principles.

Comparison of coding efficiency (under 8 kHz sampling):

PCM	–	64 kb/s
ADPCM	–	32 kb/s
CELP	–	4.8 kb/s
LPC-10	–	2.4 kb/s

Applications of process identification
Zastosowania identyfikacji procesów

3) Spectrum estimation / Estymacja widma

The power spectral density of a discrete-time wide-sense stationary stochastic process $y(t)$ can be defined as a Fourier transform of its autocorrelation function

$$S_y(\omega) = \sum_{\tau=-\infty}^{\infty} R_y(\tau) e^{-j\omega\tau}$$

$$-\pi < \omega \leq \pi$$

where ω denotes the normalized (dimensionless) angular frequency. Power spectral density function (power spectrum) shows distribution of the signal power (mean energy) over different frequency bands. It is widely used for diagnostic purposes.

<u>Theorem</u>

Consider any stationary process $\{y(t)\}$ with a continuous spectral density function $S_y(\omega)$. Then, for an arbitrarily small threshold $\epsilon > 0$, one can find such autoregressive process $\{y_{\mathrm{AR}}(t)\}$ (of sufficiently high order r) with spectral density function $S_{\mathrm{AR}}(\omega)$, for which it holds

$$|S_y(\omega) - S_{\mathrm{AR}}(\omega)| < \epsilon, \quad \forall \omega \in [0, \pi]$$

Two approaches to spectrum estimation
Dwa podejścia do estymacji widma

Spectral density function $S_y(\omega)$ of a stationary AR(r) process can be *parameterized* in two different ways:

<u>Parameterization 1</u>

$$S_y(\omega) = \sum_{i=-\infty}^{\infty} R_y(i) e^{-j\omega i}$$

This leads to the following nonparametric Blackman-Tukey approach to spectrum estimation

Step 1

Using the available data $\mathcal{Y}(N)$ estimate $M+1$, $M \ll N$, autocorrelation coefficients

$$\widehat{R}_y(i,N) = \frac{1}{N-i} \sum_{t=1}^{N-i} y(t)y(t+i)$$

$$\widehat{R}_y(-i,N) = \widehat{R}_y(i,N)\ , \quad i = 0, \ldots, M$$

Step 2

Estimate the spectral density function of $\{y(t)\}$ using the formula

$$\widehat{S}_y(\omega) = \sum_{i=-M}^{M} \widehat{R}_y(i,N) e^{-j\omega i}$$

Remark

Truncation of the autocorrelation function, a consequence of adopting a finite summation index M, results in distortions known as spectral leakage. It is possible to significantly reduce spectral leakage by replacing the implicit rectangular window with a suitably designed tapered (bell-shaped) window, such as the Hann raised cosine window, for example. The Fourier transform of the windowed estimate of the autocorrelation function is known as the Blackman–Tukey estimate of a signal power spectrum.

Parameterization 2

$$S_y(\omega) = \frac{\sigma_n^2}{\left|1 - \sum_{i=1}^{r} a_i e^{-j\omega i}\right|^2}$$

This leads to the following parametric approach to spectrum estimation

Step 1

Using the available data $\mathcal{Y}(N)$ estimate $r+1$ AR model coefficients: $\widehat{a}_1(N), \ldots, \widehat{a}_r(N)$ and $\widehat{\sigma}_n^2$.

Step 2

Estimate the spectral density function of $\{y(t)\}$ using the formula

$$\widehat{S}_y(\omega) = \frac{\widehat{\sigma}_n^2(N)}{\left|1 - \sum_{i=1}^{r} \widehat{a}_i(N) e^{-j\omega i}\right|^2}$$

The principle of parsimony
Zasada oszczędności

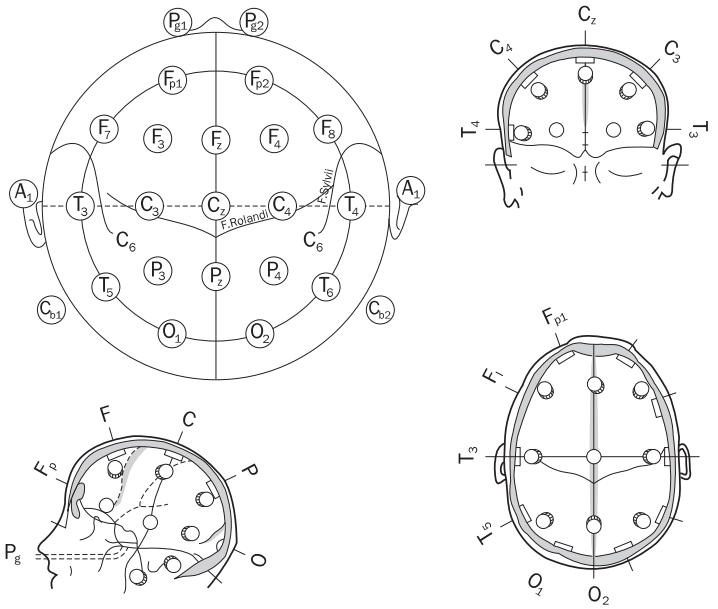
In simple words the principle of parsimony says that 'when describing a dynamic process one should not use extra parameters if not necessary'. Parsimony is clearly reminiscent of a philosophical principle known as an Occam's razor, after William of Occam: *a person should not increase, beyond what is necessary, the number of entities required to explain anything*.

It might seem that the parsimony principle contradicts the common sense observation that 'extra parameters (degrees of freedom) allow one to describe physical phenomena more accurately', which suggests that more complex models can't be inferior to simple ones. Even though agreeable for physical models, i.e. models obtained as a result of physical inference, the above statement is false for instrumental models obtained via system identification, i.e. as a result of statistical inference. The point is that parameter estimates obtained from a limited number of experimental data (finite observation history) are of finite accuracy. Loosely speaking, when the standard deviation of a coefficient estimate exceeds the true coefficient value, the benefits stemming from its inclusion in the process model become problematic from the statistical viewpoint.

Do not build models that require estimation of too many coefficients. General recommendation: $\dim[\boldsymbol{\theta}] \ll N$. Prefer parsimonious parameterizations.

Electroencephalographic (EEG) signals
Sygnały elektroencefalograficzne (EEG)

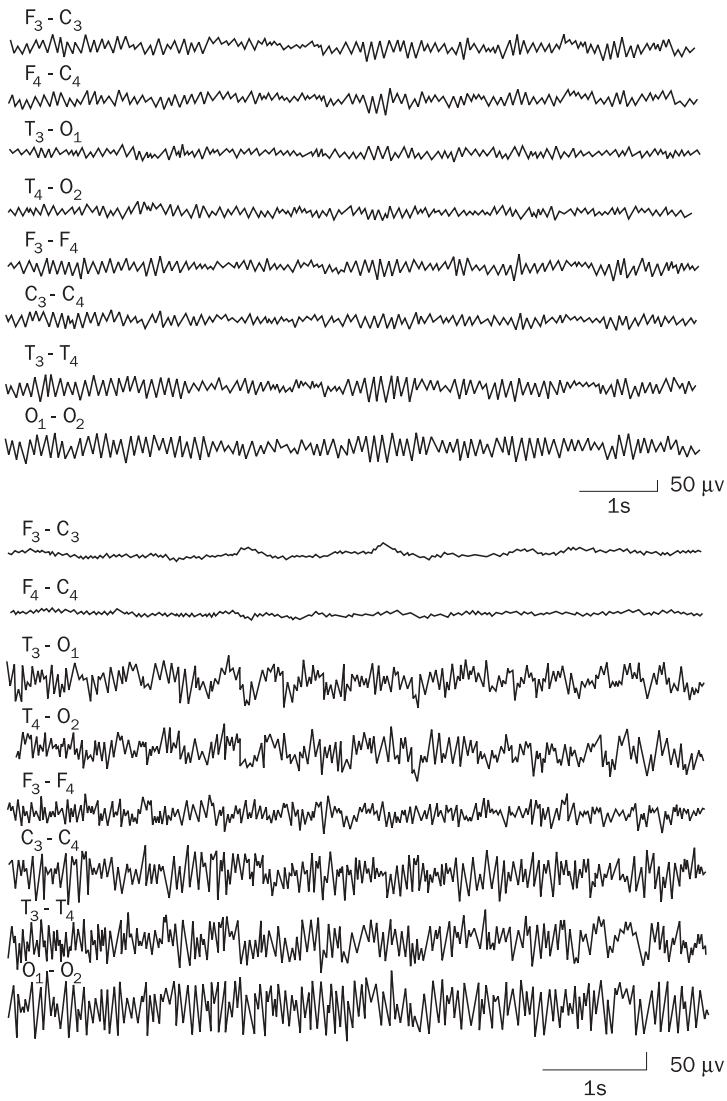
H. Berger, "Uber das Elektroenzephalogramm des Menschen", *Arch. Psychiatr. Nervenkrankheiten*, vol. 87, pp. 527–570, 1929.



Localization of electrodes on the patient's scull

EEG signals carry information about the state of the brain and its reaction to internal and external stimuli. Despite advances made recently in the area of computer-aided tomography (CAT), EEG remains a practically useful tool for studying the functional states of the brain (e.g. sleep-stage analysis) and for diagnosing functional brain disturbances (e.g. epilepsy).

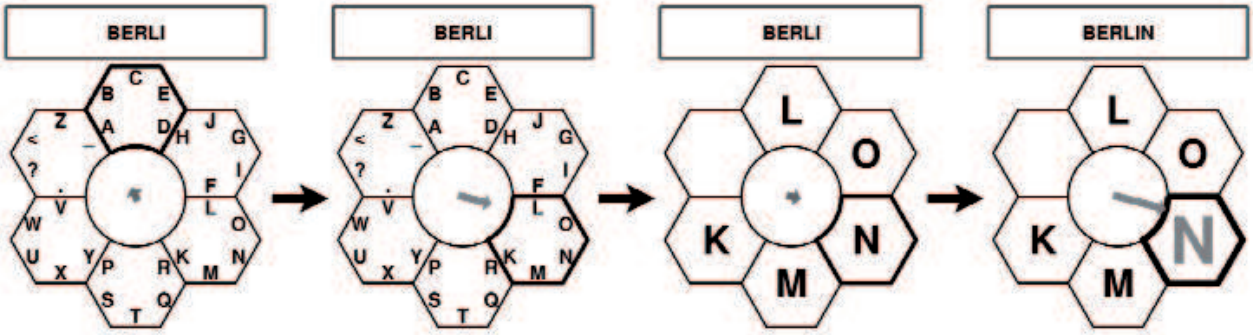
Electroencephalographic (EEG) signals



Example of normal (upper plots) and pathological (lower plots) EEG recordings

According to the traditional clinical nomenclature introduced in a pioneering work of Berger, EEG signals are described and classified in terms of several rhythmic activities called alpha (8–13 Hz), beta (14–30 Hz), delta (1–3 Hz) and theta (4–7 Hz). The normal EEG of an adult in a state called relaxed wakefulness is usually dominated by rhythmic alpha activity and a less pronounced beta activity. The contribution of delta and theta activities is arrhythmic and much smaller.

Brain-Computer Interface (BMI) System
Interfejs mózg-komputer



Hex-o-Spell – mental typewriter
designed by the Fraunhofer Institute.
Typing efficiency : from 2 to 7 symbols per minute.

The man on the photograph *thinks about* moving his left hand ("0") or about moving his right hand ("1").

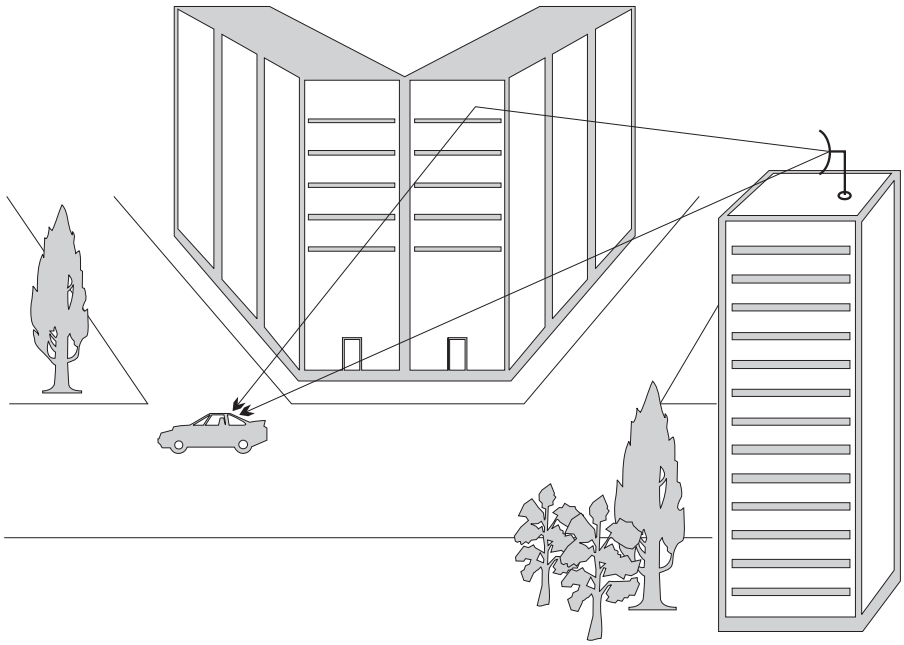
Applications of process identification
Zastosowania identyfikacji procesów

4) Equalization of telecommunication channels
Korekcja kanałów telekomunikacyjnych

In an ideal communication system the signal $y(t)$ measured at the receiver is identical with the transmitted signal $u(t)$. In reality, due to the dispersive nature of the propagation media, communication channels introduce into the transmitted signals a certain degree of distortion and a certain amount of noise. In modern wireless systems using the atmospheric or ionospheric high-frequency (HF) radio channels, distortion is caused by the so-called multipath effect (*efekt wielodrogowości*).

A) Mobile radio channels

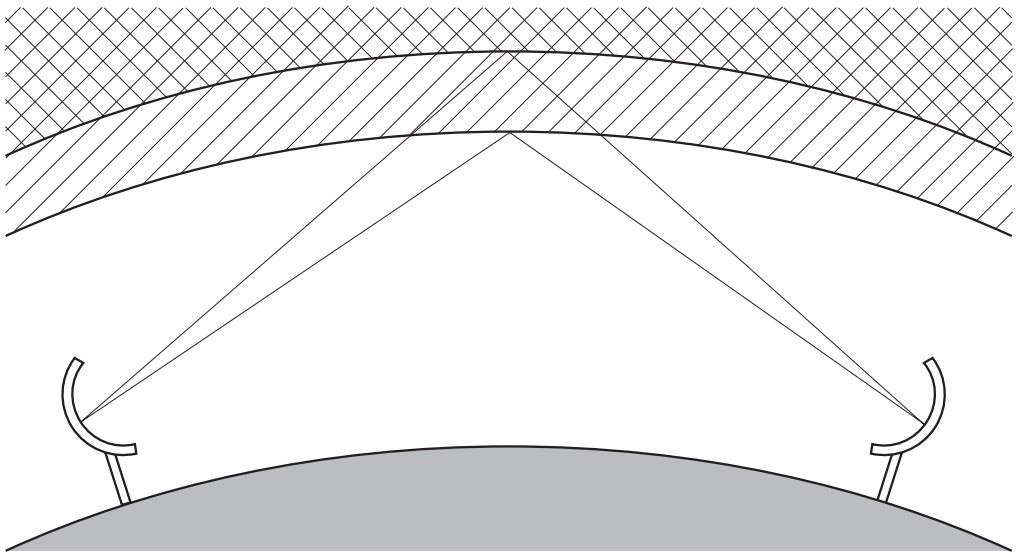
In a typical mobile radio situation one station is fixed in position and the other station is moving. In urban environments the direct line between transmitter and receiver is usually obstructed by buildings, so propagation of the electromagnetic energy to and from the mobile unit is largely by scattering, either by reflection from the flat sides of buildings or by diffraction around obstacles (natural or artificial).



Communication through an HF radio channel.

B) Line-of-sight transmission

In the line-of-sight Hertzian transmission between fixed ground stations, the multipath effect is caused by reflections of electromagnetic waves from different layers of the atmosphere and ionosphere.



Communication through a line-of-sight Hertzian link.

Modeling of wireless channels
Modelowanie kanałów bezprzewodowych

The following linear finite impulse response FIR (Finite Impulse Response) model

$$y(t) = \sum_{i=0}^{s} b_i u(t-i) + n(t),$$

is successfully used to describe the relationship between $y(t)$ and $u(t)$ in a wireless communication system over sufficiently short time intervals. The coefficients $b_0, \ldots, b_s$ characterize the impulse response of the channel along with the transmitter and receiver filters. In digital communication systems, where signals are encoded and transmitted over the channel as a sequence of pulses, one can assume that

$$u(t) = \pm 1,$$

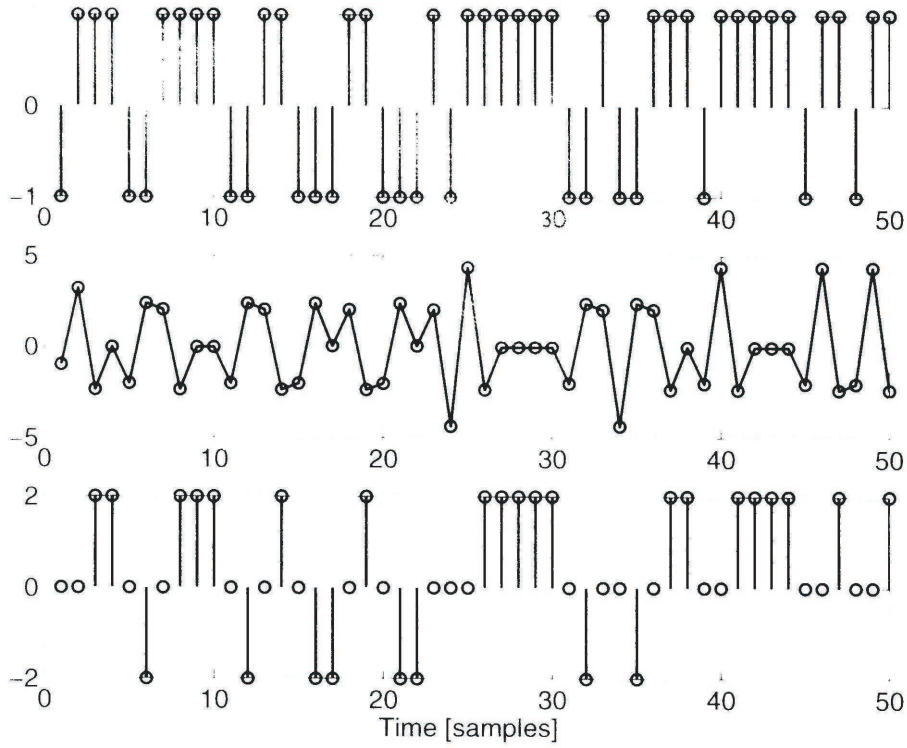
or, if quadrature amplitude modulation is used,

$$u(t) = \pm 1 \pm j.$$

The term $b_0 u(t)$ can be identified as the desired symbol (up to the scale coefficient b_0), while the remaining terms $\sum_{i=1}^{s} b_i u(t-i)$ constitute the so-called intersymbol interference (ISI) (*interferencja międzysymbolowa*) – a distortion which causes spreading and overlapping of adjacent pulses.

Channel equalization
Korekcja kanału

Since in the simplest case the decision device at the output of the data transmission system compares the received signal $y(t)$ with some preassigned threshold, the ISI component, when left unchecked, can seriously degrade the overall system performance.



To overcome this problem the receiver is equipped with a special compensating filter, called a channel equalizer, which can remove most of the intersymbol interference.

Channel equalization
Korekcja kanału

If the channel is invertible and the channel noise is negligible ($\sigma_n^2 \cong 0$), the optimal (in the mean square sense) linear output equalizer, has a pretty obvious form:

$$\widehat{u}(t) = E(q^{-1})y(t)\ , \quad E(q^{-1}) = \frac{1}{\widehat{B}(q^{-1})}$$

$$\widehat{B}(q^{-1}) = \sum_{i=0}^{s} \widehat{b}_i q^{-i}$$

Otherwise, demodulation is based on the Viterbi algorithm, which minimizes, in a computationally efficient manner, the Euclidean distance between the received sequence and all possible transmitted sequences.

Naturally, the quality of such demodulation relies heavily on the knowledge of channel coefficients. To enable receiver estimation of the channel response, a special known sequence of data symbols – the so-called training sequence – is transmitted along with the information sequence. Data is transmitted in fixed-length blocks or bursts, each of which contains its own training sequence.

Guard time (8.25 bits)	Tail (3 bits)	Information sequence (58 bits)	Training sequence (26 bits)	Information sequence (58 bits)	Tail (3 bits)

The user data packet adopted in the ETSC/GSM mobile radio system.

Applications of process identification
Zastosowania identyfikacji procesów

5) Elimination of impulsive disturbances (outliers)
Usuwanie zakłóceń impulsowych (wartości nadmiernych)

The term outlier refers to a sample which does not fit the other, usually neighboring, samples. Most of the outliers can be attributed to some kind of impulsive noise corrupting the measuring equipment, communication channel and/or the storing medium.

Phonograph produced in 1902.

Restoration of old audio files, such as archived gramophone records, is an interesting example of impulsive noise filtering. Clicks, pops and scratches are caused by local groove damage due to aging or mishandling of the record material. The width of noise pulses can be anything between several samples and several hundreds of samples (for old 78 rpm records).

Elimination of clicks

Most of the modern declicking algorithms are based on adaptive prediction. Denote by $\epsilon(t) = y(t) - \widehat{y}(t|t-1)$ the one-step-ahead prediction error and denote by

$$\widehat{\sigma}_\epsilon^2(t) = \frac{1}{M}\sum_{i=1}^{M} \epsilon^2(t-i)$$

the local estimate of the prediction error variance (M can be set to 20, for example). Finally, let $d(t)$ be the binary output of the outlier detector

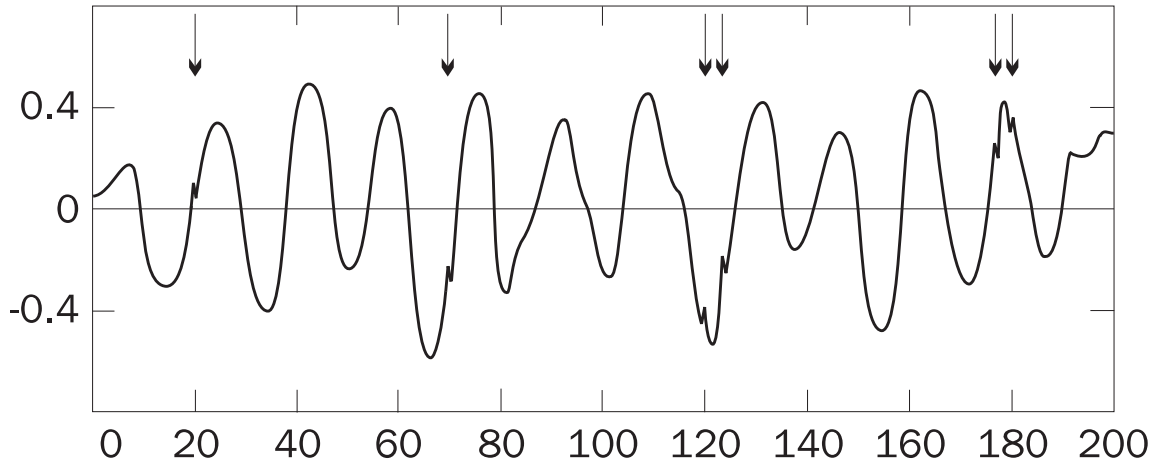
$$d(t) = \begin{cases} 0 & \text{noise impulse absent} \\ 1 & \text{noise impulse present} \end{cases}$$

The following simple decision rule can be used:

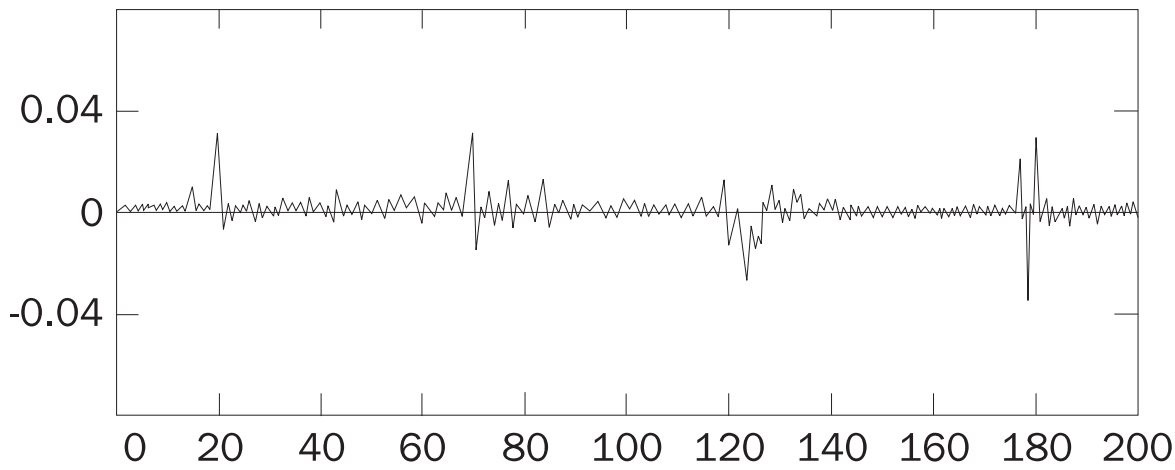
$$\widehat{d}(t) = \begin{cases} 0 & \text{if} \quad |\epsilon(t)| \leq \mu\widehat{\sigma}_\epsilon(t) \\ 1 & \text{if} \quad |\epsilon(t)| > \mu\widehat{\sigma}_\epsilon(t) \end{cases}$$

The sample is classified as an outlier and scheduled for reconstruction if the magnitude of the corresponding prediction error exceeds μ times ($\mu > 0$) its standard deviation. In audio applications the detection threshold μ has to be determined empirically; in most cases the best results are obtained for $\mu \in [3, 5]$.

Elimination of clicks



(a)



(b)

A fragment of an archived audio signal (a) and the corresponding sequence of one-step-ahead prediction errors yielded by the predictor based on the AR(10) model (b). Arrows indicate positions of audible clicks.

Reconstruction of irrevocably distorted samples

Consider an AR(r) signal $\{y(t)\}$. Suppose that one would like to reconstruct a block of m missing samples

$$y(k+1), \ldots, y(k+m)$$

based on $k \geq r$ preceding samples $\{y(1), \ldots, y(k)\}$ and $N-k-m \geq r$ succeeding samples $\{y(k+m+1), \ldots, y(N)\}$.

Reconstruction that minimizes the mean-square estimation error has the form

$$\begin{aligned}
&\widehat{y}(k+1), \ldots, \widehat{y}(k+m) \\
&\quad = \arg \min_{y(k+1),\ldots,y(k+m)} \sum_{t=r+1}^{N} \left[y(t) - \sum_{i=1}^{r} a_i y(t-i) \right]^2 \\
&\quad = \arg \min_{y(k+1),\ldots,y(k+n)} \sum_{t=k+1}^{k+r+m} \left[y(t) - \sum_{i=1}^{r} a_i y(t-i) \right]^2 \\
&\quad = J[y(k+1), \ldots, y(k+m)]
\end{aligned}$$

$$\left. \frac{\partial J[y(k+1), \ldots, y(k+m)]}{\partial y(k+i)} \right|_{\widehat{y}(k+i)} = 0, \quad i = 1, \ldots, m$$

Reconstruction of irrevocably distorted samples

Example

Suppose that the signal $\{y(t)\}$ is governed by the AR(1) model

$$y(t) = ay(t-1) + n(t)$$

and that only the sample $y(k+1)$ is missing ($m = 1$).

$$J[y(k+1)] = [y(k+1) - ay(k)]^2 + [y(k+2) - ay(k+1)]^2$$

$$\left.\frac{\partial J[y(k+1)]}{\partial y(k+1)}\right|_{\widehat{y}(k+1)} = 2[\widehat{y}(k+1) - ay(k)] - 2a[y(k+2) - a\widehat{y}(k+1)]$$

$$= 2(1+a^2)\widehat{y}(k+1) - 2a[y(k) + y(k+2)] = 0$$

$$\widehat{y}(k+1) = \frac{a[y(k) + y(k+2)]}{1+a^2}$$

Note that when linear interpolation is used, one obtains

$$\widehat{y}(k+1) = \frac{y(k) + y(k+2)}{2}$$

Applications of process identification
Zastosowania identyfikacji procesów

6) Active noise and vibration suppression
Aktywne tłumienie hałasu i drgań

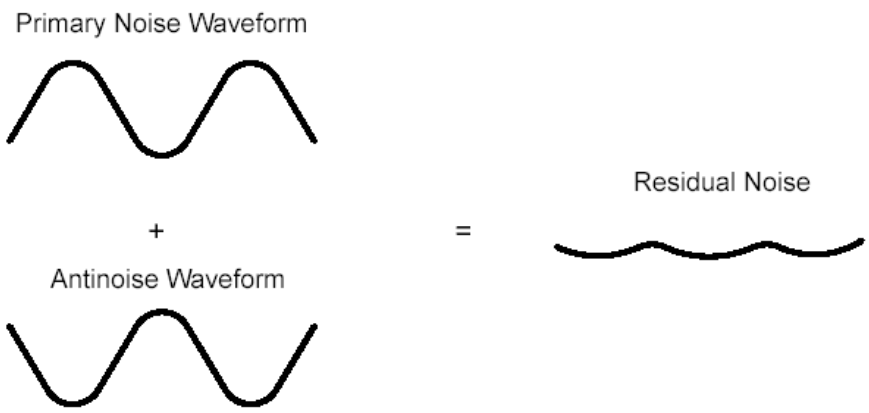
Two approaches to attenuation of acoustic disturbances

A. Passive approach

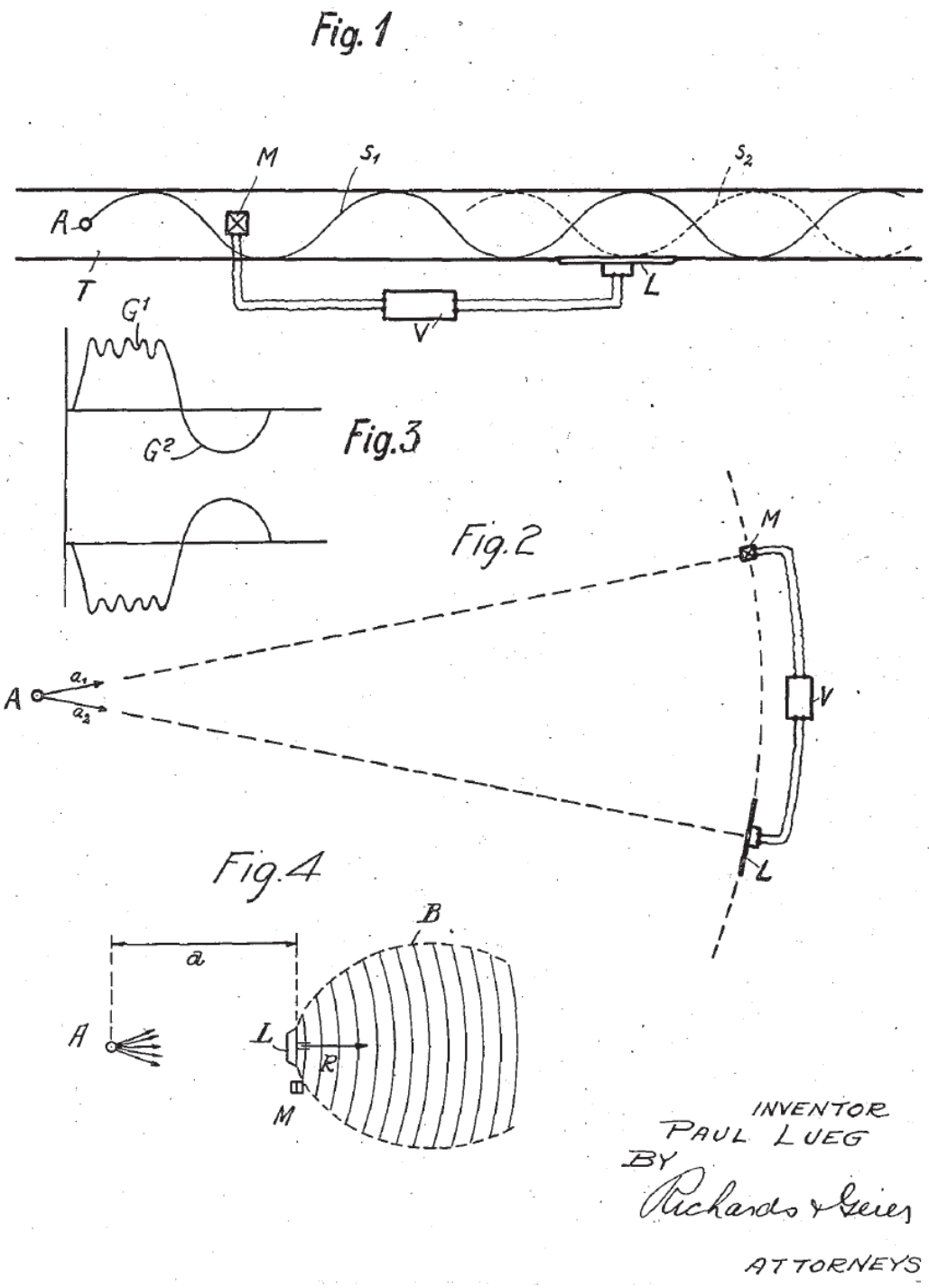
Working principle – absorption of acoustic energy: passive silencers, sound absorbers, acoustic barriers.
Physical limitations – the method is difficult to apply if frequency of acoustic waves is small (say, smaller than 500 Hz). For example, in case of a sinusoidal disturbance with frequency 100 Hz, the width of the acoustic barrier should be greater than 3.5 m (it should be at least comparable with the length of the suppressed acoustic wave).

B. Active approach

Working principle – Young's destructive interference: the sound is cancelled using an "antisound", i.e., the compensating acoustic wave with the same amplitude but reversed phase: "$1 + (-1) = 0$"

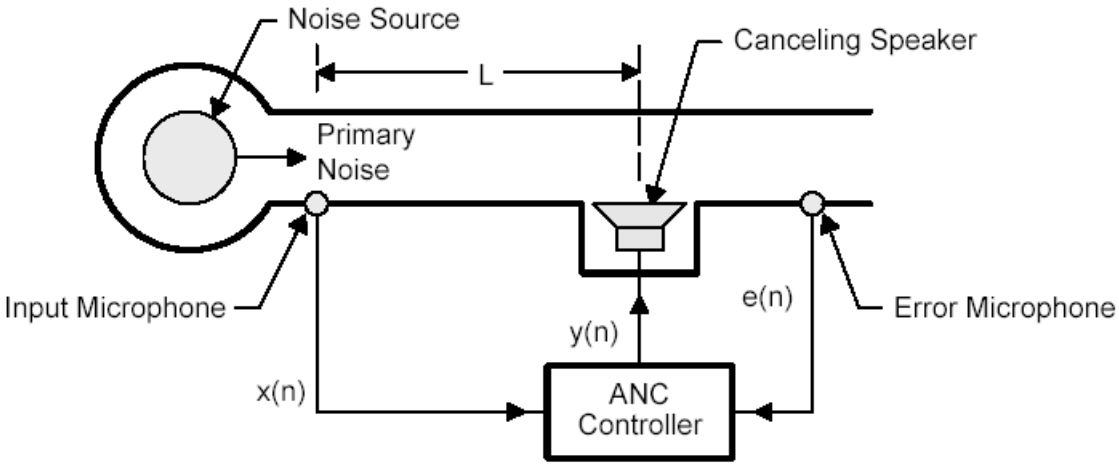


Active noise cancellation

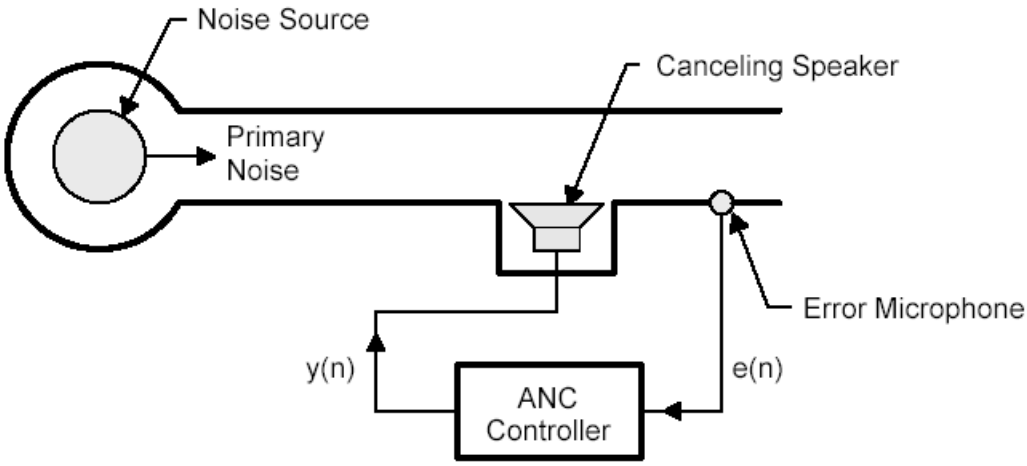


First proposal and patent – Paul Lueg 1934

Two approaches to ANC design
Dwa podejścia do projektowania systemów ANC



Feedforward ANC system.



Feedback ANC system.

Active noise and vibration suppression

Applications of ANC (Adaptive Noise Control) and AVC (Adaptive Vibration Control) systems:

- Active noise-reducing headsets.
- Interior transportation noise and vibration control (cancellation of vibration/noise in automobiles, buses, trucks, railway vehicles, aircraft cabins, operator cabins in industrial machines etc.)
- Interior noise and vibration control in military vehicles (helicopters, tanks, submarines)
- Active noise cancellation for HVAC (Heating, Ventilation and Air-Conditioning) equipment and clean rooms.
- Silencing of noisy industrial machines (fans, mills, transformatory, transformers, saws, pumps); quiet zones in engine rooms.
- Reduction of MRI (Magnetic Resonance Imaging) noise.
- Active mufflers for automobiles, trucks and light machinery.

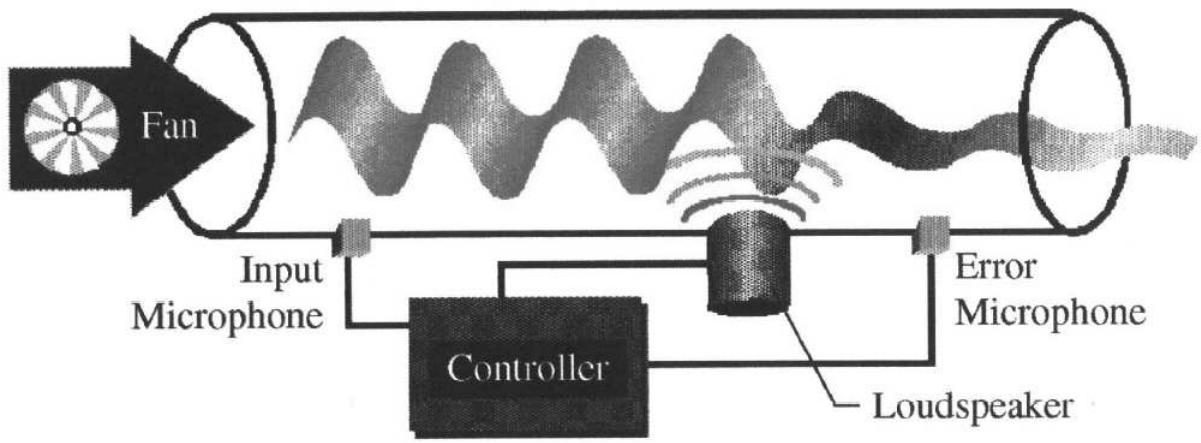
Noise cancellation in acoustic ducts
Tłumienie szumu w duktach akustycznych

ANC systems are used to cancel plane waves, which occur at low frequencies in ducts. Plane waves propagate at frequencies between 20 Hz and the plane wave cut-off frequency f_c, above which the sound wave can propagate in two dimensions. The value of f_c for a cylindrical duct of diameter d is given by the following equation

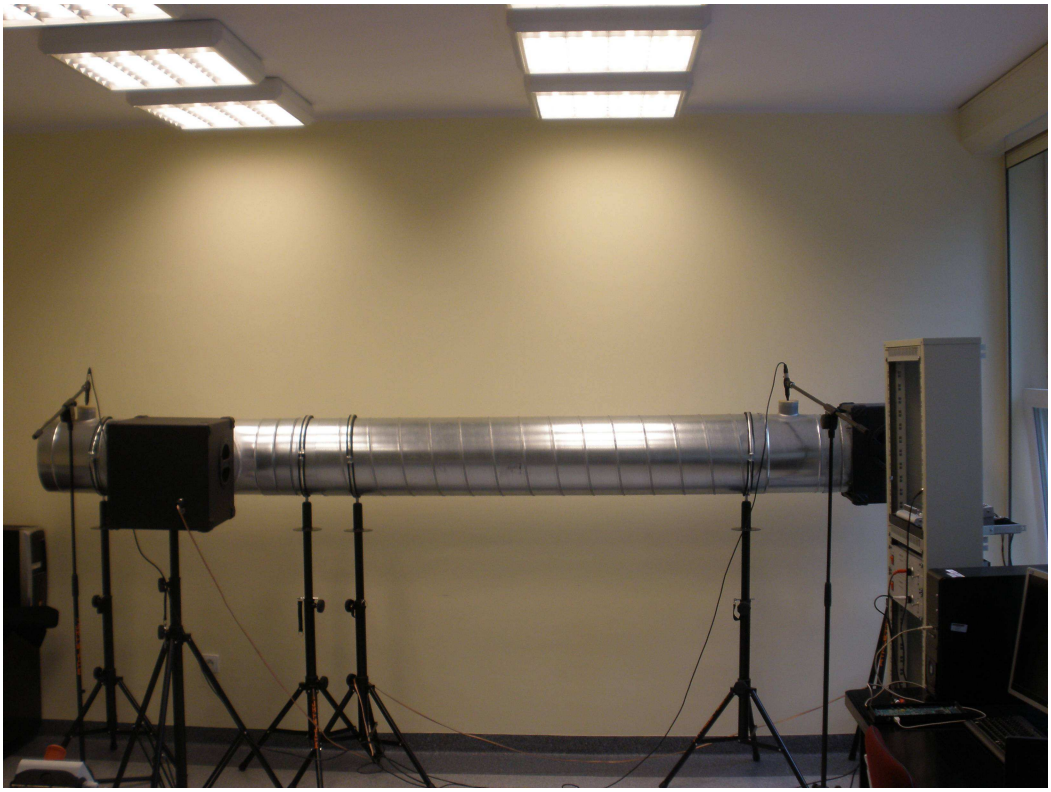
$$f_c = \frac{1.841 v_0}{\pi d} \text{ Hz}$$

where v_0 denotes the speed of sound in air.

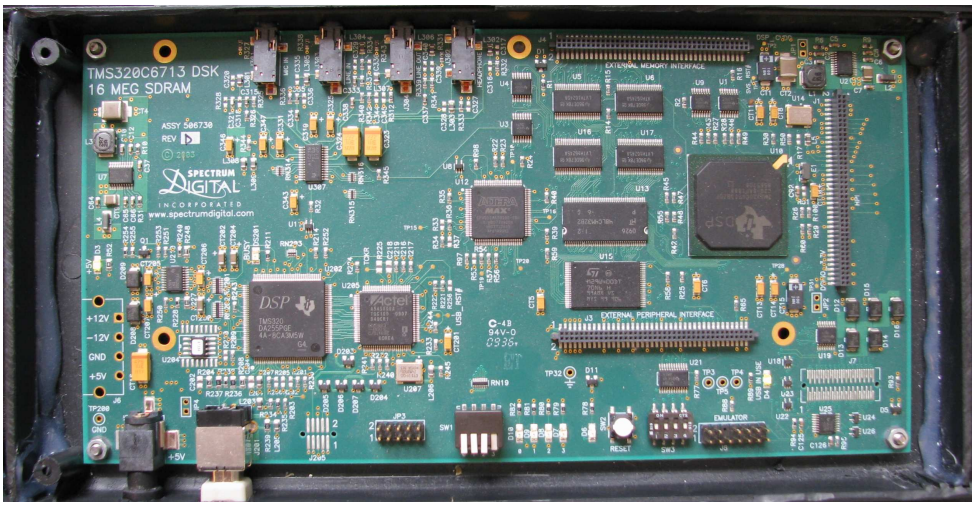
$$v_0 = 343\text{m/s}, d = 0.31\text{m} \Longrightarrow f_c \cong 648\text{Hz}$$



Our laboratory system



Cylindrical duct of diameter $d = 0.31$ m.



Texas Instruments DSP board TMS320C6713.

Applications of process identification
Zastosowania identyfikacji procesów

7) Adaptive control
Sterowanie adaptacyjne

Suppose that one should stabilize a certain technological variable $x(t)$ at a prescribed level x_0. Let $y(t) = x(t) - x_0$. The control objective can be stated as follows

$$u(t): \quad \mathrm{E}[y^2(t)] \longrightarrow \min$$

minimum-variance (MV) control problem

When the controlled system is described by the ARX model

$$y(t+1) = \sum_{i=1}^{r} a_i y(t+1-i) + b_1 u(t) + \sum_{i=2}^{p} b_i u(t+1-i) + n(t+1)$$

the minimum-variance controller takes the form

$$u(t) = -\frac{1}{b_1}\left[\sum_{i=1}^{r} a_i y(t+1-i) + \sum_{i=2}^{p} b_i u(t+1-i)\right]$$

Adaptive minimum-variance regulators

Adaptacyjne regulatory minimalnowariancyjne

$$u(t) = -\frac{1}{\widehat{b}_1(t)}\left[\sum_{i=1}^{r}\widehat{a}_i(t)y(t+1-i) + \sum_{i=2}^{p}\widehat{b}_i(t)u(t+1-i)\right]$$

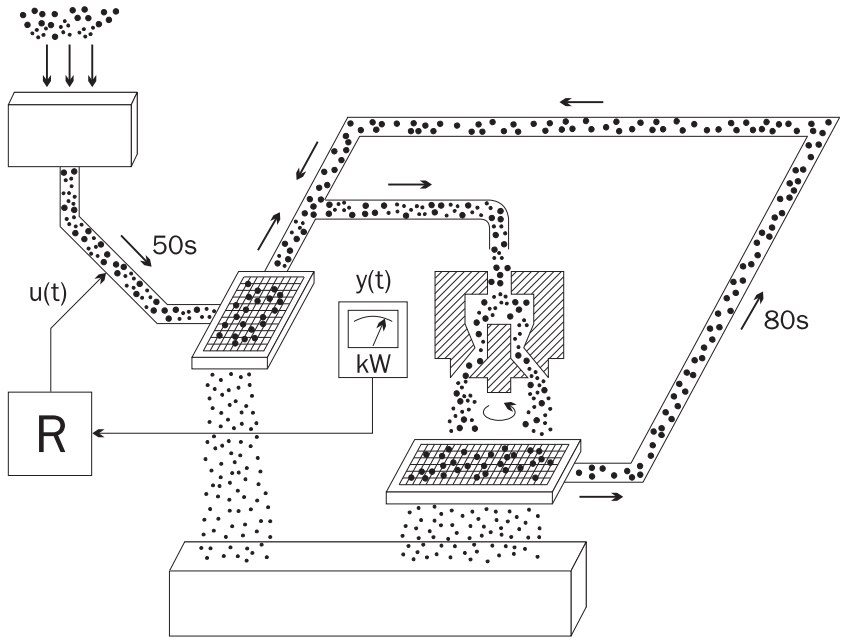
Applications:

- Systems with unknown dynamics or working under unknown conditions.
- Nonstationary systems
- Systems working under nonstationary conditions
- Nonlinear systems working in the presence of set-point changes.

Main advantage: the regulator automatically tunes its coefficients to unknown/changing operating conditions.

Control of the ore crushing plant
Sterowanie instalacją do kruszenia rudy żelaza

One of the early successful applications of self-tuning minimum variance regulators, was for control of the ore crushing plant depicted below.

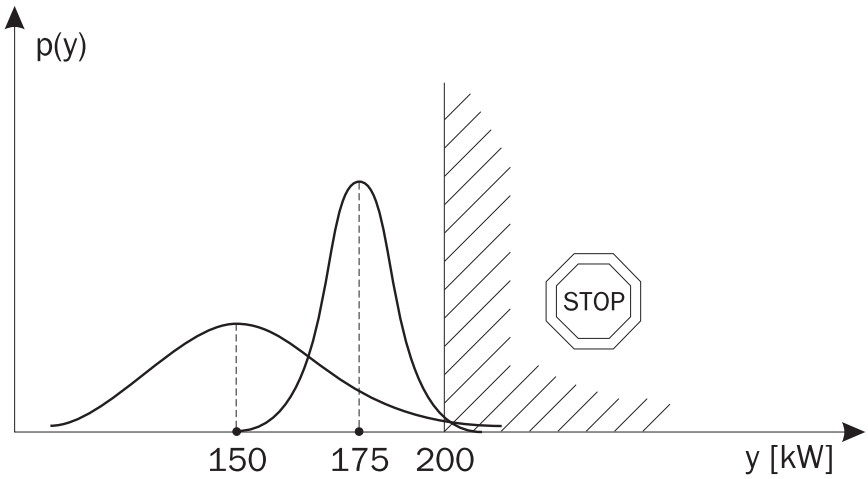


Schematic diagram of an ore crushing plant.

The aim of this plant, consisting of two screens and a crusher driven by an electric motor, is to crush the raw ore material (discharged from the refillable input bin through a system of conveyor belts) to lumps with a maximum dimension of 25 mm. Lumps which already have the desired dimension are separated by the first screen. The rest of the ore is fed into the crusher. The second screen, placed at the output of the crusher, selects the material which needs to be recycled because its grains are the wrong size.

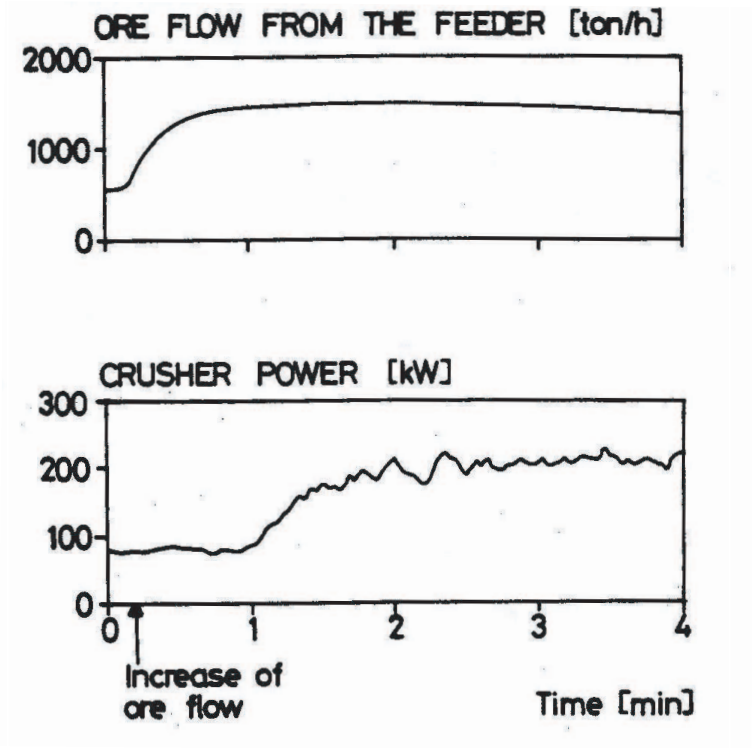
The control objective is to change the flow of the raw material supplied by the electromechanical feeder so as to maximize the power of the crusher without overheating its driving motor. The overload protection mechanism (a special slip coupling) automatically stops the crushing line if the load becomes too high. Since it may take about an hour before the full production rate is restored after a standstill, the probability of such forced stops must be kept sufficiently small if production is to be maintained at a high level.

By minimizing the fluctuations in the instantaneous power of the driving motor around its nominal value, it is possible to bring the set point power closer to the maximum capacity of the crusher for a given probability of an overload. Hence the optimal control of the ore crusher falls pretty naturally into the minimum variance category.



Dependence of the set point of the ore crushing plant on the variance of fluctuations in the instantaneous power: smaller variance allows one to increase the nominal power from 150 kW to 175 kW.

Choice of the sampling interval T_s



$$T_s = \frac{1}{5} \div \frac{1}{15} T_{95\%} \quad \longrightarrow \quad T_s = \frac{1}{10} \times 200\text{s} = 20\text{s}$$

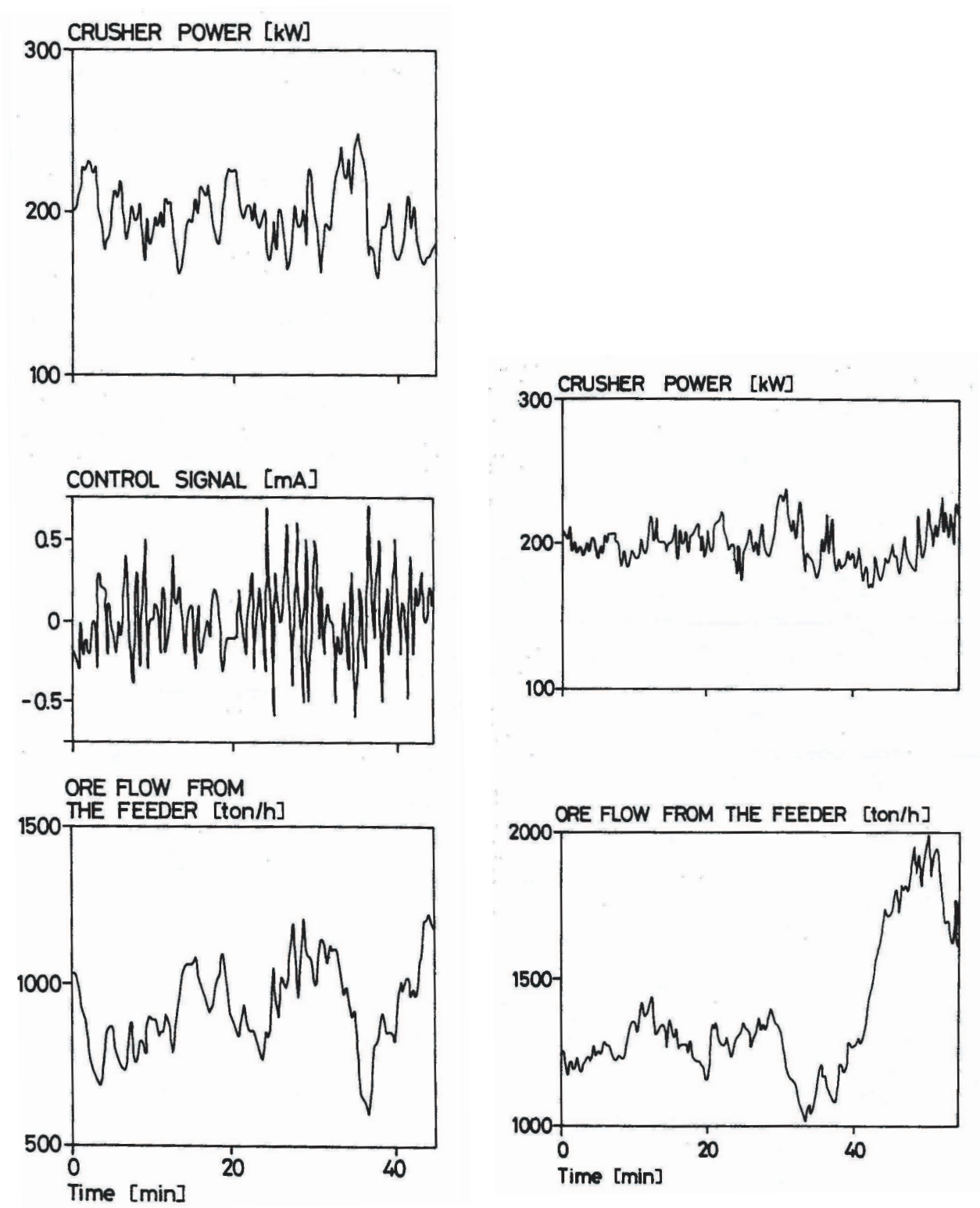
System model

$$y(t) = \sum_{i=1}^{4} a_i y(t-3-i) + \sum_{i=1}^{4} b_i u(t-3-i) + n(t)$$

Sources of system nonstationarity

Nonstationarity of the ore crushing plant originates from variations in crushability of the raw material (depending mainly on the waste rock content), the average lump size and the wear of the jackets (the gap width of the crusher is adjusted every second day and the jackets are replaced every third month).

Two case studies

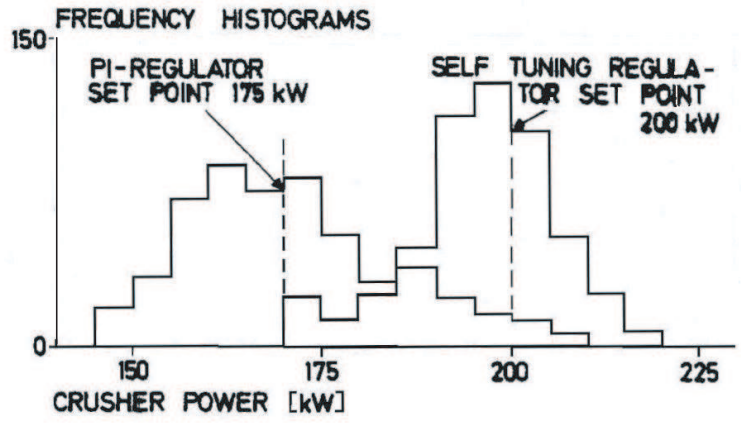


(a) (b)

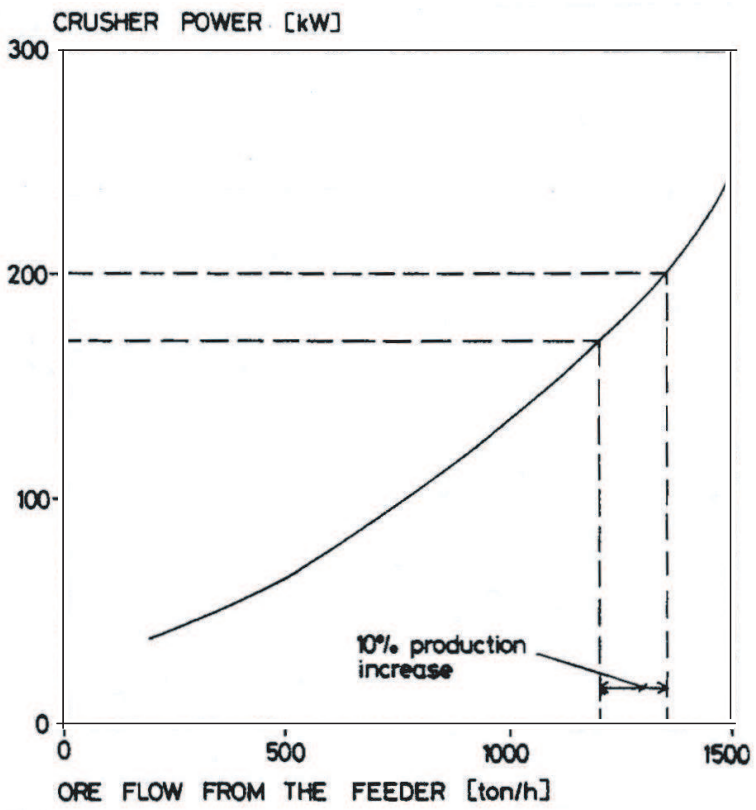
Response of an adaptive MV regulator to a sudden increase (a) and sudden decrease (b) in the instantaneous crusher power.

Adaptive control benefits

Korzyści wynikajce ze sterowania adaptacyjnego



Histograms of the instantaneous power obtained for the PI regulator and adaptive MV regulator.



Ore production versus nominal crusher power.

Let's talk about money
Pomówmy o pieniądzach

Adaptive control made it possible to raise the set point power from 170 kW (PI control) to 200 kW, yielding a production increase of approximately 10%. Since, at the time of implementation, 10 production lines were operated in Kiruna, one obtains

$$10 \times 10\% = 100\% = 1 \text{ production line}$$

The cost of a new production line was at that time equal to 3 millions of USD.

Process identification – basic notions

Identyfikacja procesów – podstawowe pojęcia

Model / Model

$$\mathcal{M}_\phi = \{s(\cdot), \boldsymbol{\phi}\}$$

where $s(\cdot)$ denotes the system equation and $\boldsymbol{\phi}$ denotes the vector of parameters characterizing both dynamic and probabilistic properties of the modeled process. The "true" value of $\boldsymbol{\phi}$ will be denoted by $\boldsymbol{\phi}_0$ and the "true" system model $\mathcal{M}_{\phi_0} = \{s_0(\cdot), \boldsymbol{\phi}_0\}$ will be denoted by $\mathcal{S}$.

Example

$$\mathrm{ARX}(r,p): \quad \boldsymbol{\phi} = \begin{bmatrix} \boldsymbol{\theta} \\ \sigma_n^2 \end{bmatrix}, \quad \boldsymbol{\theta} = [a_1, \ldots, a_r, b_1, \ldots, b_p]^{\mathrm{T}}$$

Class of models / Klasa modeli

$$\mathcal{C}_\phi = \{\mathcal{M}_\phi, \boldsymbol{\phi} \in \Omega_\phi\}$$

where Ω_ϕ denotes the set of admissible values of the vector $\boldsymbol{\phi}$. Usually we will assume that $\mathcal{S} \in \mathcal{C}_\phi$.

Example

$$\mathrm{ARX}(r,p): \quad \Omega_\theta = R^{r+p} \text{ or } R^{r+p}_{\mathrm{stab}}, \quad \Omega_{\sigma_n^2} = R_+$$

Process identification – basic notions
Identyfikacja procesów – podstawowe pojęcia

Measure of fit / Miara jakości dopasowania

$$J_N(\boldsymbol{\phi}) = J(\mathcal{M}_\phi, \Omega(N))$$

where $\Omega(N) = \{\mathcal{Y}(N), \mathcal{U}(N)\}$ denotes the available data set

$$\mathcal{Y}(N) = \{y(1), \ldots, y(N)\}$$
$$\mathcal{U}(N) = \{u(1), \ldots, u(N)\}$$

Estimator / Estymator

$$\widehat{\boldsymbol{\phi}}(N) = \widehat{\boldsymbol{\phi}}(\Omega(N)) = \arg \min_{\boldsymbol{\phi} \in \Omega_\phi} J_N(\boldsymbol{\phi})$$

In statistics, an estimator (*estymator*) is a rule (*przepis*) $\widehat{\boldsymbol{\phi}}(\cdot)$ for calculating an estimate (*estymata*) $\widehat{\boldsymbol{\phi}}(\Omega(N))$ of a given quantity ϕ_0 based on observed data $\Omega(N)$.

Since, due to input and/or measurement noise, the available data set $\Omega(N)$ can be regarded as a realization of a random variable, the estimator itself is also a random variable.

Desired properties of a "good estimator"
Pożądane własności "dobrego estymatora"

Unbiasedness / Brak obciążenia

The estimator $\widehat{\phi}(N)$ is called unbiased (*estymator nieobciążony*) if its expected value is equal to the true value

$$\mathrm{E}_{\Omega(N)}\left[\widehat{\phi}(N)\right] = \phi_0$$

The quantity

$$B[\widehat{\phi}(N)] = \mathrm{E}_{\Omega(N)}\left[\widehat{\phi}(N) - \phi_0\right]$$

is called the estimation bias (*obciążenie estymacji*).

When it holds that $\lim_{N\to\infty} B[\widehat{\phi}(N)] = 0$, the estimator is called asymptotically unbiased (*estymator asymptotycznie nieobciążony*).

Consistency / Zgodność

The estimator $\widehat{\phi}(N)$ is called consistent (*estymator zgodny*) if it converges, in a stochastic sense (e.g. with probability one), to the true parameter value as the number of data points increases to infinity

$$\lim_{N\to\infty} \widehat{\phi}(N) = \phi_0 \quad \mathrm{w.p.1}$$

Desired properties of a "good estimator"
Pożądane własności "dobrego estymatora"

Efficiency / Efektywność

The estimator $\widehat{\boldsymbol{\phi}}(N)$ is called efficient (*estymator najefektywniejszy*) if it is unbiased and if its covariance matrix $\boldsymbol{\Sigma}_\phi$ takes the smallest possible value among covariance matrices of all unbiased estimators:

$$\begin{aligned}
\boldsymbol{\Sigma}_\phi &= \operatorname{cov}\left[\widehat{\boldsymbol{\phi}}(N)\right] \\
&= \mathrm{E}_{\Omega(N)}\left[\left(\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_0\right)\left(\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_0\right)^{\mathrm{T}}\right] \\
&\leq \mathrm{E}_{\Omega(N)}\left[\left(\widehat{\boldsymbol{\psi}}(N) - \boldsymbol{\phi}_0\right)\left(\widehat{\boldsymbol{\psi}}(N) - \boldsymbol{\phi}_0\right)^{\mathrm{T}}\right] \\
&= \operatorname{cov}\left[\widehat{\boldsymbol{\psi}}(N)\right] = \boldsymbol{\Sigma}_\psi
\end{aligned}$$

where $\widehat{\boldsymbol{\psi}}(N)$ is *any* unbiased estimator of $\boldsymbol{\phi}_0$.

$$D_\phi = \left\{\boldsymbol{\phi} : \boldsymbol{\phi}^{\mathrm{T}}\Sigma_\phi^{-1}\boldsymbol{\phi} \leq 1\right\}$$

$$D_\psi = \left\{\boldsymbol{\psi} : \boldsymbol{\psi}^{\mathrm{T}}\Sigma_\psi^{-1}\boldsymbol{\psi} \leq 1\right\}$$

$$\boldsymbol{\Sigma}_\phi \leq \boldsymbol{\Sigma}_\psi \quad \Longleftrightarrow D_\phi \subseteq D_\psi$$

covariance ellipsoid of Σ_ϕ is located inside (possibly touching) the covariance ellipsoid of Σ_ϕ.

Titius-Bode law

Prawo Titiusa-Bodego

Johann Daniel Titius

Johann Elert Bode

The history of the empirical formula known as the Titius–Bode law of planetary distances is an excellent example of the fact that instrumental models can play an important and stimulating role in scientific research. In his translation of the Charles Bonnet's book *Contemplation de la Nature* (1764), the German astronomer Johann Daniel Titius von Wittenberg included his own observations on some intriguing regularities of planetary geometry in our solar system. Assuming that the average distance of Saturn (the most distant of all planets known at that time) from the Sun is equal to 100 units, Titius remarked that the average planetary radii d_n could be described by a very simple mathematical formula

$$d_n = 2 + 3 \cdot 2^n,$$

where n is a planet's "serial number".

n	planet
-∞	Mercury
0	Venus
1	Earth
2	Mars
3	?
4	Jupiter
5	Saturn

Average distance of different planets from the Sun (as known in the middle of the eighteenth century)

When Titius was alive, there was no planet corresponding to $n = 3$. "Can it be that the Grand Constructor left that place empty?" asked Titius. "Never!" was his emphatic answer.

When in 1781 William Herschel discovered Uranus, encircling the Sun at an average distance of 192 units ($d_6 = 196$), confidence in the Titius formula, popularized in the meantime by his fellow astronomer Johann Elert Bode, increased even more. But the orbit corresponding to $n = 3$ still remained empty. In 1801, soon after the international search had begun, the Italian amateur astronomer Giovanni Piazzi spotted an unknown heavenly body later named Ceres. Piazzi continued to trace the trajectory of 'his' planet for almost six weeks before a bad cold forced him to stop observations. When he recovered from his illness Ceres was lost!

The method of least squares (LS)
Metoda najmniejszych kwadratów (NK)

Karl Friedrich Gauss (1777-1855)

Then a German mathematician entered the scene; it was the young but already well-known Carl Friedrich Gauss. Using a new method he called least squares, Gauss was able to take the scarce and inaccurate measurements provided by Piazzi and estimate eight parameters characterizing the orbit of Ceres. The lost planet was soon tracked down. Its average distance from the Sun was equal to 27.67 units, in very good agreement with the law of Titius–Bode ($d_3 = 28$). Shortly after spotting Ceres, other small planets with similarly shaped orbits were discovered: Juno (1804), Vesta (1807), Astraea (1845), etc. The modern solar catalog contains several thousands of these objects and they are known as planetoids.

Least squares identification of ARX systems
Identyfikacja obiektów ARX metodą najmniejszych kwadratów

ARX(r, p):

$$
\begin{aligned}
y(t) &= \sum_{i=1}^{r} a_i y(t-i) + \sum_{i=1}^{p} b_i u(t-i) + n(t) \\
&= \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta} + n(t)
\end{aligned}
$$

where

$$
\boldsymbol{\theta} = \begin{bmatrix} a_1 \\ \vdots \\ a_r \\ b_1 \\ \vdots \\ b_p \end{bmatrix}, \qquad \boldsymbol{\varphi}(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-r) \\ u(t-1) \\ \vdots \\ u(t-p) \end{bmatrix}
$$

parameter vector
wektor parametrów

regression vector
wektor regresji

$$
J_N(\boldsymbol{\theta}) = \sum_{t=1}^{N} \left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta} \right]^2 \longrightarrow \min
$$

In this formulation we assume that the "initial conditions" $y(0), y(-1), \ldots, y(1-r)$ and $u(0), u(-1), \ldots, u(1-p)$ are *known*.

Minimization of a multivariate function
Minimalizacja funkcji wielu zmiennych

$$\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}^{\mathrm{T}}$$

<u>Problem</u>

Find the minimum $\mathbf{x}_0$ of a differentiable function $f(\mathbf{x})$

$$\mathbf{x}_0 = \arg\min_{\mathbf{x}} f(\mathbf{x})$$

<u>Solution</u>

$$\mathbf{x}_0 : \quad \nabla_{\mathbf{x}} f(\mathbf{x})\big|_{\mathbf{x}=\mathbf{x}_0} = \mathbf{0}$$

where

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial f(\mathbf{x})}{\partial x_1} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_n} \end{bmatrix}^{\mathrm{T}}$$

$\mathbf{x}_0$ is a (local) minimum provided that

$$\nabla^2_{\mathbf{x}\mathbf{x}^{\mathrm{T}}} f(\mathbf{x}) = \begin{bmatrix} \frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_1 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f(\mathbf{x})}{\partial x_n x_1} & \dots & \frac{\partial f(\mathbf{x})}{\partial x_n^2} \end{bmatrix}_{\mathbf{x}=\mathbf{x}_0} > \mathbf{O}$$

<u>Useful identities</u>

$$\nabla_{\mathbf{x}}[\mathbf{y}^{\mathrm{T}}\mathbf{x}] = \mathbf{y} \ , \quad \nabla_{\mathbf{x}}[\mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}] = \mathbf{A}\mathbf{x} + \mathbf{A}^{\mathrm{T}}\mathbf{x}$$

Least squares identification of ARX systems

Identyfikacja obiektów ARX metodą najmniejszych kwadratów

$$J_N(\boldsymbol{\theta}) = \sum_{t=1}^{N} \left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2 = \sum_{t=1}^{N} y^2(t)$$
$$- 2\left[\sum_{t=1}^{N} y(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\boldsymbol{\theta} + \boldsymbol{\theta}^{\mathrm{T}}\left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\boldsymbol{\theta}$$

$$\nabla_{\boldsymbol{\theta}} J_N(\boldsymbol{\theta}) = -2\sum_{t=1}^{N} y(t)\boldsymbol{\varphi}(t) + 2\left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\boldsymbol{\theta}$$

$$\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}} J_N(\boldsymbol{\theta}) = 2\left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]$$

$$\mathbf{R}(N) = \sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)$$

regression matrix / macierz regresji

If the regression matrix $\mathbf{R}(N)$ is nonsingular, one obtains

$$\widehat{\boldsymbol{\theta}}(N) = \mathbf{R}^{-1}(N)\mathbf{s}(N), \quad \mathbf{s}(N) = \sum_{t=1}^{N} y(t)\boldsymbol{\varphi}(t)$$

Identifiability of ARX systems
Identyfikowalność obiektów ARX

Note that for *any* vector $\mathbf{x}$ of dimension $n = r + p$ it holds that

$$\mathbf{x}^{\mathrm{T}} \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t) \boldsymbol{\varphi}^{\mathrm{T}}(t) \right] \mathbf{x} = \sum_{t=1}^{N} \left[\mathbf{x}^{\mathrm{T}} \boldsymbol{\varphi}(t) \right]^2 \geq 0$$

which means that the regression matrix $\mathbf{R}(N)$ is "by definition" nonnegative definite.

The LS estimator can be computed provided that the regression matrix is *positive definite* and hence nonsingular

$$\mathbf{R}(N) = \sum_{t=1}^{N} \boldsymbol{\varphi}(t) \boldsymbol{\varphi}^{\mathrm{T}}(t) > \mathbf{O}$$

identifiability condition / warunek identyfikowalności

The identifiability condition is not fulfilled if the elements of the regression vector $\boldsymbol{\varphi}(t)$ are *linearly dependent*, i.e.,

$$\exists \mathbf{x} \neq \mathbf{0} : \quad \mathbf{x}^{\mathrm{T}} \boldsymbol{\varphi}(t) = 0 \,, \quad \forall t$$

We say that under such experimental conditions the ARX model is *not identifiable* (przy takim uwarunkowaniu eksperymentu model ARX jest *nieidentyfikowalny*).

Sources of nonidentifiability
Zródła nieidentyfikowalności

Nonidentifiability *is not* a property of the analyzed system but rather a consequence of defective experimental conditions under which system identification is carried out. There are two well-known sources of system nonidentifiability:

- Application of input signals that are not persistently exciting of sufficiently high order [so that one of the input variables can be expressed as a linear combination of the remaining u-components of $\boldsymbol{\varphi}(t)$].

- Existence of linear feedback between the input and output variables [so that one of the input variables can be expressed as a linear combination of the remaining components of $\boldsymbol{\varphi}(t)$].

Insufficient excitation

<u>Definition</u>

The signal $\{u(t)\}$ is persistently exciting of order n (sygnał pobudzający rzędu n) if:

1. The following limits exist

$$R_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t)u(t-\tau), \quad \tau = 0, 1, \ldots$$

2. The $n \times n$ matrix $\mathcal{I}_n$ defined below is positive definite

$$\mathcal{I}_n = \begin{bmatrix} R_u(0) & & R_u(n-1) \\ \vdots & \ddots & \vdots \\ R_u(n-1) & & R_u(0) \end{bmatrix} > \mathbf{O}$$

3. The $(n+1) \times (n+1)$ matrix $\mathcal{I}_{n+1}$ is singular: $\det[\mathcal{I}_{n+1}] = 0$.

<u>Examples</u>

- The signal $u(t) \equiv 1$ is persistently exciting of order 1.
- The signal $u(t) = (-1)^t$ is persistently exciting of order 1.
- The signal $u(t) = \sin \omega t$ is persistently exciting of order 2 (note: $u(t) = au(t-1) - u(t-2)$, where $a = 2\cos\omega$).
- A mixture of k sinusoidal signals with different frequencies is persistently exciting of order $2k$.
- White noise is persistently exciting of order ∞.
- Pseudo-random binary sequence (PRBS) is persistently exciting of order ∞ (note: PRBS has the largest power among all amplitude-constrained sequences).

Example 1

Check identifiability of the following ARX model

$$\begin{aligned} y(t) &= a_1 y(t-1) + a_2 y(t-2) \\ &\quad + b_1 u(t-1) + b_3 u(t-3) + n(t) \end{aligned}$$

in the case where the input signal has the form

$$u(t) = 5 - u(t-1), \quad t = 1, 2, \ldots$$

Does the answer depend on the initial value $u(0) = u_0$?

Solution

Note that in the case considered

$$\boldsymbol{\varphi}(t) = [y(t-1) \;\; y(t-2) \;\; u(t-1) \;\; u(t-3)]^{\mathrm{T}}$$

and

$$u(t) = u(t-2), \quad \forall t$$

Therefore

$$[0 \;\; 0 \;\; 1 \;\; -1]\boldsymbol{\varphi}(t) = 0, \quad \forall t$$

which means that the model is, under such experimental conditions, not identifiable. The conclusion does not depend on the value of u_0.

Sources of nonidentifiability
Źródła nieidentyfikowalności

Linear feedback

Example 2

Check identifiability of the following ARX model

$$\begin{aligned} y(t) &= a_1 y(t-1) + a_2 y(t-2) \\ &\quad + b_1 u(t-1) + b_2 u(t-2) + n(t) \end{aligned}$$

in the case where the input signal has the form

$$u(t) = k_1 y(t) + k_2 u(t-1), \quad \forall t$$

Solution

Note that in the case considered

$$\boldsymbol{\varphi}(t) = [y(t-1) \;\; y(t-2) \;\; u(t-1) \;\; u(t-2)]^{\mathrm{T}}$$

and

$$u(t-1) = k_1 y(t-1) + k_2 u(t-2), \quad \forall t$$

Therefore

$$[-k_1 \;\; 0 \;\; 1 \;\; -k_2]\boldsymbol{\varphi}(t) = 0, \quad \forall t$$

which means that the model is, under such experimental conditions, not identifiable.

How to restore identifiability ?
Jak przywrócić identyfikowalność ?

Indufficient excitation

- Use sufficiently "rich" excitation signal.

Linear feedback

- Use *dither* signals to destroy linear relationship among regression variables (e.g. add white noise to the output of a linear controller).

- Use linear feedback controllers of sufficiently high order.

- Switch between several controllers, for example:

$$u(t) : \begin{cases} \mathrm{PID}_1 & \text{for} \quad t \in [1, \frac{N}{2}] \\ \mathrm{PID}_2 & \text{for} \quad t \in (\frac{N}{2}, N] \end{cases}$$

  Note: the sum of two singular regression matrices is usually nonsingular.

Estimation of the driving noise variance
Estymacja wariancji szumu tworzącego

$$n(t, \boldsymbol{\theta}) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}$$

modeling errors / błędy modelowania

$$n(t, \widehat{\boldsymbol{\theta}}(N)), \quad t = 1, \ldots, N$$

residual errors / błędy reszt(k)owe

The estimate of the driving noise variance σ_n^2 can be obtained from

$$\begin{aligned}\widehat{\sigma}_n^2(N) &= \frac{1}{N}\sum_{t=1}^{N} n^2(t, \widehat{\boldsymbol{\theta}}(N)) \\ &= \frac{1}{N}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(N)\right]^2\end{aligned}$$

Properties of the least squares estimators
Własności estymatorów najmniejszych kwadratów

"Idealistic" analysis: $\mathcal{S} \in \mathcal{C}_\phi$

Suppose that the true system model has the form

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 + n(t), \quad \mathrm{var}[n(t)] = \sigma_0^2$$

Then it holds that

$$\begin{aligned}\widehat{\boldsymbol{\theta}}(N) &= \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1} \left[\sum_{t=1}^{N} y(t)\boldsymbol{\varphi}(t)\right] \\ &= \boldsymbol{\theta}_0 + \left[\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1} \left[\sum_{t=1}^{N} n(t)\boldsymbol{\varphi}(t)\right]\end{aligned}$$

leading to

$$\begin{aligned}&\widehat{\boldsymbol{\theta}}(N) - \boldsymbol{\theta}_0 \\ &= \left[\frac{1}{N}\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1} \left[\frac{1}{N}\sum_{t=1}^{N} n(t)\boldsymbol{\varphi}(t)\right]\end{aligned}$$

Assumption 1

$\{n(t)\}$ is a sequence of zero-mean, independent and identically distributed random variables with variance σ_0^2 (white noise).

Assumption 2

$\{\boldsymbol{\varphi}(t)\}$ is a stationary ergodic process with positive definite correlation matrix

$$\mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right] = \mathbf{R}_0 > \mathbf{O}$$

The variables $n(t)$ and $\boldsymbol{\varphi}(t)$ are mutually uncorrelated $\forall t$.

As a consequence of assumptions 1 and 2, one obtains

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t) = \mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right] = \mathbf{R}_0$$

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} n(t)\boldsymbol{\varphi}(t) = \mathrm{E}\left[n(t)\boldsymbol{\varphi}(t)\right] = \mathbf{0}$$

$$\text{w.p. } 1$$

$$\begin{aligned}\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}(N) &= \boldsymbol{\theta}_0 + \left\{\mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\right\}^{-1} \mathrm{E}\left[n(t)\boldsymbol{\varphi}(t)\right] \\ &= \boldsymbol{\theta}_0 \quad \text{w.p. } 1\end{aligned}$$

Conclusion: the least squares estimator $\widehat{\boldsymbol{\theta}}(N)$ is consistent

Furthermore

$$\mathrm{E}\left[\widehat{\boldsymbol{\theta}}(N)\right] \cong \boldsymbol{\theta}_0$$

$$\mathrm{cov}\left[\widehat{\boldsymbol{\theta}}(N)\right] \cong \frac{\sigma_0^2 \mathbf{R}_0^{-1}}{N}$$

Conclusion: the least squares estimator $\widehat{\boldsymbol{\theta}}(N)$ is (approximately) unbiased

In an analogous way, one can show that

$$\lim_{N\to\infty} \widehat{\sigma}_n^2(N) = \sigma_0^2 \quad \text{w.p. } 1$$

and

$$\mathrm{E}\left[\widehat{\sigma}_n^2(N)\right] \cong \frac{N-n}{N}\,\sigma_0^2$$

$$\mathrm{var}\left[\widehat{\sigma}_n^2(N)\right] \cong \frac{2\sigma_0^4}{N}$$

where $n = r + p = \dim[\boldsymbol{\theta}]$ denotes the number of estimated coefficients.

Conclusion: the estimator $\widehat{\sigma}_n^2(N)$ is consistent and asymptotically unbiased

Properties of the least squares estimators
Własności estymatorów najmniejszych kwadratów

"Realistic" analysis: $\mathcal{S} \notin \mathcal{C}_\phi$

Suppose that the system governed by

$$y(t) = G_0(q^{-1})u(t) + n(t)$$

is represented using the following (simpler) model

$$y(t) = G(q^{-1}, \boldsymbol{\theta})u(t) + n(t)$$

Suppose that the method of least squares is used to estimate unknown parameters, namely

$$\widehat{\boldsymbol{\theta}}(N) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t, \boldsymbol{\theta})$$

where

$$\varepsilon(t, \boldsymbol{\theta}) = y(t) - \widehat{y}(t|t-1)$$
$$\widehat{y}(t|t-1) = G(q^{-1}, \boldsymbol{\theta})u(t)$$

The best approximating model
Najlepiej aproksymujący model

Suppose that $\{u(t)\}$ and $\{n(t)\}$ are zero-mean, mutually uncorrelated, stationary random processes with spectral density functions $S_u(\omega)$ and $S_n(\omega)$, respectively. Then it holds that

$$\lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} \varepsilon^2(t,\boldsymbol{\theta}) = \mathrm{E}\left[\varepsilon^2(t,\boldsymbol{\theta})\right]$$

$$\lim_{N\to\infty} \widehat{\boldsymbol{\theta}}(N) = \boldsymbol{\theta}_* = \arg\min_{\boldsymbol{\theta}} \mathrm{E}\left[\varepsilon^2(t,\boldsymbol{\theta})\right]$$

Since

$$\varepsilon(t,\boldsymbol{\theta}) = \left[G_0(q^{-1}) - G(q^{-1},\boldsymbol{\theta})\right] u(t) + n(t)$$

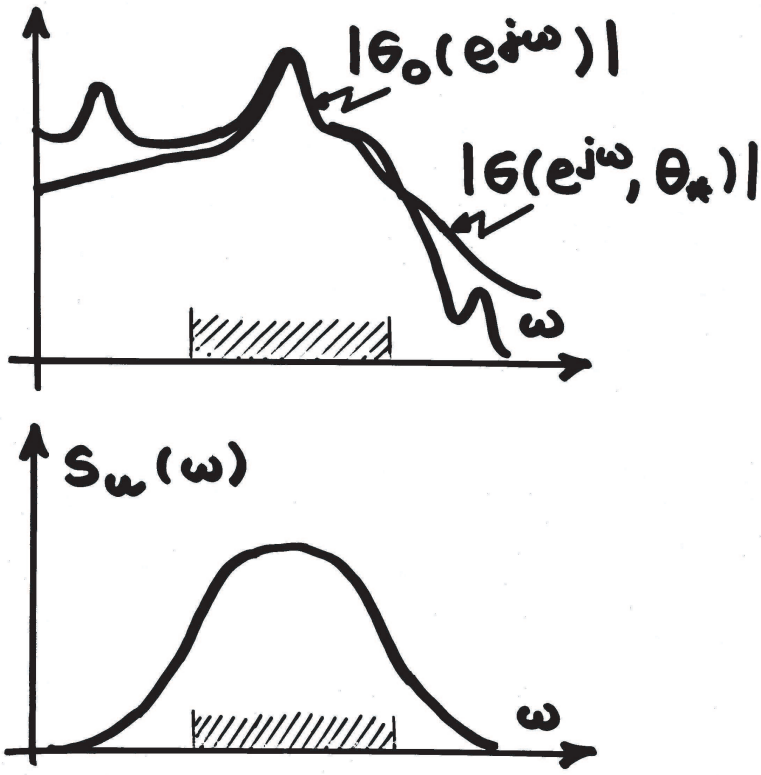
one obtains

$$\mathrm{E}\left[\varepsilon^2(t,\boldsymbol{\theta})\right] = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_\varepsilon(\omega,\boldsymbol{\theta}) d\omega$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} S_n(\omega) d\omega$$

$$+ \frac{1}{2\pi} \int_{-\pi}^{\pi} \left|G_0(q^{-1}) - G(q^{-1},\boldsymbol{\theta})\right|^2 S_u(\omega) d\omega$$

The best approximating model
Najlepiej aproksymujący model



Conclusion: the best approximation results can be achieved in the frequency range "dominated" by the excitation signal $u(t)$

The method of maximum likelihood
Metoda największej wiarygodności

Ronald Aylmer Fisher (1890 - 1962)

$$p(\Omega(N); \boldsymbol{\phi}) = p(\Omega(N)|\boldsymbol{\phi})$$

likelihood function / funkcja wiarygodności

$$\widehat{\boldsymbol{\phi}}(N) = \arg\max_{\boldsymbol{\phi}} p(\Omega(N); \boldsymbol{\phi})$$

The principle of maximum likelihood: find the model that is most likely to have generated the available data.

Zasada największej wiarygodności: znajdź model, dla którego prawdopodobieństwo wygenerowania posiadanego zbioru danych jest największe.

Maximum likelihood estimators
Estymatory największej wiarygodności

Note that the maximum likelihood principle can be reformulated as follows

$$J_N(\boldsymbol{\phi}) = -p(\Omega(N); \boldsymbol{\phi}) \longrightarrow \min$$

and that

$$\widehat{\boldsymbol{\phi}}(N) = \arg\max_{\boldsymbol{\phi}} p(\Omega(N); \boldsymbol{\phi}) = \arg\max_{\boldsymbol{\phi}} L(\Omega(N); \boldsymbol{\phi})$$

$$L(\Omega(N); \boldsymbol{\phi}) = \ln[p(\Omega(N); \boldsymbol{\phi})]$$

log likelihood function / logarytmiczna funkcja wiarygodności

Example 1

Estimate the unknown constant quantity θ based on its N noisy measurements

$$y(t) = \theta + n(t), \quad t = 1, \ldots, N$$

where $\{n(t)\}$ is a sequence of zero-mean independent random variables with Gaussian distribution:

$$n(t) \sim \mathcal{N}(0, \sigma_t^2)$$

$$p(n(t)) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{n^2(t)}{2\sigma_t^2}\right\}$$

Solution

$$\Omega(N) = \{y(1), \ldots, y(N)\}$$

$$p(\Omega(N);\theta) = p(n(1,\theta), \ldots, n(N,\theta)) = \prod_{t=1}^{N} p(n(t,\theta))$$

$$= \prod_{t=1}^{N} \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\left\{-\frac{n^2(t,\theta)}{2\sigma_t^2}\right\}$$

where $n(t,\theta) = y(t) - \theta$.

$$L(\Omega(N);\theta) = -\frac{N}{2}\ln(2\pi) - \frac{1}{2}\sum_{t=1}^{N}\ln(\sigma_t^2)$$

$$-\frac{1}{2}\sum_{t=1}^{N}\frac{[y(t)-\theta]^2}{\sigma_t^2}$$

$$L(\Omega(N);\theta) \to \max \quad \Longrightarrow \quad \sum_{t=1}^{N}\frac{[y(t)-\theta]^2}{\sigma_t^2} \to \min$$

$$\widehat{\theta}(N) = \frac{\displaystyle\sum_{t=1}^{N}\frac{y(t)}{\sigma_t^2}}{\displaystyle\sum_{t=1}^{N}\frac{1}{\sigma_t^2}}$$

Example 2

Estimate the unknown constant quantity θ based on its N noisy measurements

$$y(t) = \theta + n(t), \quad t = 1, \ldots, N$$

where $\{n(t)\}$ is a sequence of zero-mean independent random variables with Laplace distribution:

$$n(t) \sim \mathcal{L}(0, \sigma_n^2)$$

$$p(n(t)) = \frac{\sqrt{2}}{4\sigma_n} \exp\left\{ - \frac{|n(t)|}{\sqrt{2}\,\sigma_n} \right\}$$

Solution

$$p(\Omega(N); \theta) = p(n(1, \theta), \ldots, n(N, \theta)) = \prod_{t=1}^{N} p(n(t, \theta))$$

$$= \prod_{t=1}^{N} \frac{\sqrt{2}}{4\sigma_n} \exp\left\{ - \frac{|n(t, \theta)|}{\sqrt{2}\,\sigma_n} \right\}$$

$$L(\Omega(N); \theta) = - \frac{3N}{2} \ln 2 - N \ln \sigma_n$$

$$- \frac{1}{\sqrt{2}\,\sigma_n} \sum_{t=1}^{N} |y(t) - \theta|$$

$$L(\Omega(N);\theta) \to \max$$

$$\Downarrow$$

$$\sum_{t=1}^{N} |y(t) - \theta| \to \min$$

$$\widehat{\theta}(N) = \mathrm{med}\{y(1), \ldots, y(N)\}$$

where med($\cdot$) denotes the median of a sequence.

Denote by $\{\widetilde{y}(1), \ldots, \widetilde{y}(N)\}$ the original sequence $\{y(1), \ldots, y(N)\}$ rearranged in the nondecreasing order

$$\widetilde{y}(1) \leq \widetilde{y}(2) \leq \ldots \leq \widetilde{y}(N)$$

Then

$$\mathrm{med}\{y(1), \ldots, y(N)\}$$
$$= \begin{cases} \widetilde{y}(n+1) & \text{when} \quad N = 2n+1 \\ \dfrac{\widetilde{y}(n) + \widetilde{y}(n+1)}{2} & \text{when} \quad N = 2n \end{cases}$$

Maximum likelihood squares identification of ARX systems

Identyfikacja obiektów ARX metodą największej wiarygodności

Assumption 3

The process $\{n(t)\}$ is normally distributed:

$$n(t) \sim \mathcal{N}(0, \sigma_n^2)$$

We will find the maximum likelihood estimator of the parameter vector $\boldsymbol{\phi} = [\boldsymbol{\theta}^{\mathrm{T}}, \sigma_n^2]^{\mathrm{T}}$ characterizing the ARX system operated in the open-loop mode

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta} + n(t)$$

In the case considered

$$p(\Omega(N); \boldsymbol{\phi}) = p(\Omega(N)|\boldsymbol{\phi}) = p(\mathcal{Y}(N)|\mathcal{U}(N)\boldsymbol{\phi})p(\mathcal{U}(N))$$

$$\propto p(n(1,\theta), \ldots, n(N,\theta)) = \prod_{t=1}^{N} p(n(t,\theta))$$

$$= \frac{1}{(2\pi\sigma_n^2)^{N/2}} \exp\left\{ -\frac{1}{2\,\sigma_n^2} \sum_{t=1}^{N} \left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2 \right\}$$

$$L[\Omega(N);\boldsymbol{\phi}] = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma_n^2)$$
$$- \frac{1}{2\,\sigma_n^2}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2$$

$$\nabla_{\boldsymbol{\phi}} L[\Omega(N);\boldsymbol{\phi}]_{\boldsymbol{\phi}=\widehat{\boldsymbol{\phi}}(N)} = \mathbf{0}$$

$$\Downarrow$$

$$\frac{\partial L[\Omega(N);\widehat{\boldsymbol{\theta}}(N),\widehat{\sigma}_n^2(N)]}{\partial\boldsymbol{\theta}} = \mathbf{0}$$
$$\frac{\partial L[\Omega(N);\widehat{\boldsymbol{\theta}}(N),\widehat{\sigma}_n^2(N)]}{\partial\sigma_n^2} = \mathbf{0}$$

$$\Downarrow$$

$$\widehat{\boldsymbol{\theta}}(N) = \left[\sum_{t=1}^{N}\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1}\left[\sum_{t=1}^{N}y(t)\boldsymbol{\varphi}(t)\right]$$
$$\widehat{\sigma}_n^2(N) = \frac{1}{N}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(N)\right]^2$$

Conclusion: under Gaussian assumptions the maximum likelihood estimates of ARX model parameters are identical with their least squares counterparts

The Cramér-Rao inequality
Nierówność Craméra-Rao

Harald Cramér (1893-1985)

Calyampudi Radhakrishna Rao (1920)

The Cramér-Rao inequality
Nierówność Craméra-Rao

<u>Theorem</u>

Suppose that the derivatives $\nabla_{\boldsymbol{\phi}} p(\Omega(N); \boldsymbol{\phi})$ and $\nabla^2_{\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{T}}} p(\Omega(N); \boldsymbol{\phi})$ exist and are absolutely integrable, and that $\widehat{\boldsymbol{\phi}}(N)$ is any unbiased estimator of $\boldsymbol{\phi}_0$,i.e., $\mathrm{E}\left[\widehat{\boldsymbol{\phi}}(N)\right] = \boldsymbol{\phi}_0$. Then it holds that

$$\begin{aligned}\mathrm{cov}\left[\widehat{\boldsymbol{\phi}}(N)\right] &= \mathrm{E}\left[\left(\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_0\right)\left(\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_0\right)^{\mathrm{T}}\right] \\ &\geq \mathcal{F}^{-1}(\boldsymbol{\phi}_0) = \mathrm{CRLB}\left[\widehat{\boldsymbol{\phi}}(N)\right]\end{aligned}$$

Cramér-Rao lower bound (CRLB)

where

$$\begin{aligned}\mathcal{F}(\boldsymbol{\phi}_0) &= \mathrm{E}\left[\frac{\partial L[\Omega(N); \boldsymbol{\phi}_0]}{\partial \boldsymbol{\phi}} \frac{\partial L[\Omega(N); \boldsymbol{\phi}_0]}{\partial \boldsymbol{\phi}^{\mathrm{T}}}\right] \\ &= -\mathrm{E}\left[\frac{\partial^2 L[\Omega(N); \boldsymbol{\phi}_0]}{\partial \boldsymbol{\phi}\, \partial \boldsymbol{\phi}^{\mathrm{T}}}\right]_{n \times n}\end{aligned}$$

Fisher information matrix

In its simplest, univariate form, Cramér-Rao inequality states that the variance of any unbiased estimator of a deterministic parameter is at least as high as the inverse of the Fisher information.

Fisher matrix for ARX processes
Macierz Fishera dla procesów ARX

Assumption 1

$\{n(t)\}$ is a sequence of zero-mean, independent and identically distributed Gaussian random variables with variance σ_0^2.

Assumption 2

$\{\boldsymbol{\varphi}(t)\}$ is a stationary ergodic process with positive definite correlation matrix

$$\mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right] = \mathbf{R}_0 > \mathbf{O}$$

The variables $n(t)$ and $\boldsymbol{\varphi}(t)$ are mutually uncorrelated $\forall t$.

$$L[\Omega(N);\boldsymbol{\phi}] = -\frac{N}{2}\ln(2\pi) - \frac{N}{2}\ln(\sigma_n^2) - \frac{1}{2\,\sigma_n^2}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2$$

$$\frac{\partial L[\Omega(N);\boldsymbol{\phi}]}{\partial \boldsymbol{\theta}} = \frac{1}{\sigma_n^2}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]\boldsymbol{\varphi}(t)$$

$$\frac{\partial L[\Omega(N);\boldsymbol{\phi}]}{\partial \sigma_n^2} = -\frac{N}{2\sigma_n^2} + \frac{1}{2\sigma_n^4}\sum_{t=1}^{N}\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2$$

$$\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}]}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}} = -\frac{1}{\sigma_n^2}\sum_{t=1}^{N}\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\varphi}(t)$$

$$\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}]}{\partial\sigma_n^2\partial\sigma_n^2} = \frac{N}{2\sigma_n^4} - \frac{1}{\sigma_n^6}\sum_{t=1}^{N}\left[y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2$$

$$\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}]}{\partial\boldsymbol{\theta}\partial\sigma_n^2} = -\frac{1}{\sigma_n^4}\sum_{t=1}^{N}\left[y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]\boldsymbol{\varphi}(t)$$

$$= \left[\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}]}{\partial\boldsymbol{\theta}^{\mathrm{T}}\partial\sigma_n^2}\right]^{\mathrm{T}}$$

Note that $y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 = n(t)$ and hence

$$\mathrm{E}\left[y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0\right]^2 = \mathrm{E}[n^2(t)] = \sigma_0^2$$

$$\mathrm{E}\left\{\left[y(t)-\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0\right]\boldsymbol{\varphi}(t)\right\} = \mathrm{E}[n(t)\boldsymbol{\varphi}(t)] = \mathbf{0}$$

leading to

$$\mathrm{E}\left[\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}_0]}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^{\mathrm{T}}}\right] = -\frac{N\mathbf{R}_0}{\sigma_0^2}$$

$$\mathrm{E}\left[\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}_0]}{\partial\sigma_n^2\partial\sigma_n^2}\right] = -\frac{N}{2\sigma_0^4}$$

$$\mathrm{E}\left[\frac{\partial^2 L[\Omega(N);\boldsymbol{\phi}_0]}{\partial\boldsymbol{\theta}\partial\sigma_n^2}\right] = \mathbf{0}$$

where $\boldsymbol{\phi}_0 = [\boldsymbol{\theta}_0^{\mathrm{T}}, \sigma_0^2]^{\mathrm{T}}$.

$$\mathcal{F}(\boldsymbol{\phi}_0) = \begin{bmatrix} \frac{N\mathbf{R}_0}{\sigma_0^2} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & \frac{N}{2\sigma_0^4} \end{bmatrix}$$

$$\mathcal{F}^{-1}(\boldsymbol{\phi}_0) = \begin{bmatrix} \frac{\sigma_0^2\mathbf{R}_0^{-1}}{N} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & \frac{2\sigma_0^4}{N} \end{bmatrix}$$

$$\mathrm{cov}\left[\widehat{\boldsymbol{\phi}}(N)\right] \geq \mathcal{F}^{-1}(\boldsymbol{\phi}_0)$$

$$\mathrm{cov}\left[\widehat{\boldsymbol{\theta}}(N)\right] \geq \frac{\sigma_0^2\mathbf{R}_0^{-1}}{N}$$

$$\mathrm{var}\left[\widehat{\theta}_i(N)\right] \geq \frac{\sigma_0^2\left[\mathbf{R}_0^{-1}\right]_{ii}}{N}$$

$$i = 1, \ldots, n$$

$$\mathrm{var}\left[\widehat{\sigma}_n^2(N)\right] \geq \frac{2\sigma_0^4}{N}$$

Conclusion: the maximum likelihood estimators $\widehat{\boldsymbol{\theta}}(N)$ and $\widehat{\sigma}_n^2(N)$ are statistically efficient [see the results obtained earlier for the LS estimators]

<u>Theorem</u>

If an unbiased estimator attains the Cramér-Rao lower bound (i.e., if it is efficient), then it is equal to the maximum likelihood estimator.

Can one beat the Cramér-Rao bound?
Czy można pokonać ograniczenie Craméra-Rao?

Consider a scalar deterministic parameter ϕ. When estimating ϕ based on $\Omega(N)$, what one really cares about is minimization of the mean squared estimation error (MSE)

$$\begin{aligned}\mathrm{MSE}\left[\widehat{\phi}(N)\right] &= \mathrm{E}_{\Omega(N)}\left\{\left[\widehat{\phi}(N)-\phi_0\right]^2\right\} \\ &= B^2\left[\widehat{\phi}(N)\right] + \mathrm{var}\left[\widehat{\phi}(N)\right]\end{aligned}$$

where

$$B\left[\widehat{\phi}(N)\right] = \mathrm{E}_{\Omega(N)}\left[\widehat{\phi}(N)-\phi_0\right]$$

denotes estimation bias.

There exist biased estimators for which it holds that

$$\mathrm{MSE}\left[\widehat{\phi}(N)\right] < \mathrm{CRLB}\left[\widehat{\phi}(N)\right]$$

<u>Example</u>

Let $\{y(t)\}$ denote a sequence of independent and identically distributed Gaussian variables with zero mean and unknown variance σ_y^2.

We will compare two estimators of σ_y^2 based on $\Omega(N) = \{y(1), \ldots, y(N)\}$

unbiased estimator:

$$\widehat{\sigma}_y^2(N) = \frac{1}{N} \sum_{t=1}^{N} y^2(t), \quad B\left[\widehat{\sigma}_y^2(N)\right] = 0$$

$$\begin{aligned} \mathrm{MSE}\left[\widehat{\sigma}_y^2(N)\right] &= \mathrm{var}\left[\widehat{\sigma}_y^2(N)\right] \\ &= \frac{2\sigma_y^2}{N} = \mathrm{CRLB}\left[\widehat{\sigma}_y^2(N)\right] \end{aligned}$$

biased estimator:

$$\widetilde{\sigma}_y^2(N) = \frac{1}{N+2} \sum_{t=1}^{N} y^2(t), \quad B\left[\widetilde{\sigma}_y^2(N)\right] = -\frac{2\sigma_y^2}{N+2}$$

$$\begin{aligned} \mathrm{MSE}\left[\widetilde{\sigma}_y^2(N)\right] &= B^2\left[\widetilde{\sigma}_y^2(N)\right] + \mathrm{var}\left[\widetilde{\sigma}_y^2(N)\right] \\ &= \frac{2\sigma_y^2}{N+2} < \mathrm{CRLB}\left[\widehat{\sigma}_y^2(N)\right] \end{aligned}$$

Note, however, that

$$\lim_{N\to\infty} \frac{\mathrm{CRLB}\left[\widehat{\sigma}_y^2(N)\right]}{\mathrm{MSE}\left[\widetilde{\sigma}_y^2(N)\right]} = \lim_{N\to\infty} \frac{\mathrm{CRLB}\left[\widehat{\sigma}_y^2(N)\right]}{\mathrm{var}\left[\widetilde{\sigma}_y^2(N)\right]} = 1$$

The evil of superefficiency
Zło superefektywności

Let

$$\eta\left[\widehat{\phi}(N)\right] = \frac{\mathrm{CRLB}\left[\widehat{\phi}(N)\right]}{\mathrm{var}\left[\widehat{\phi}(N)\right]}$$

Fisher conjectured that for any asymptotically unbiased estimator it holds that

$$\lim_{N\to\infty} \eta\left[\widehat{\phi}(N)\right] \leq 1$$

He was wrong!

<u>Counterexample</u>

Let ϕ_* denote a fixed point ("trap") in the continuous parameter space Ω_ϕ, and let $\widehat{\phi}(N)$ be a statistically efficient estimator of ϕ. Define

$$\widetilde{\phi}(N) = \begin{cases} \widehat{\phi}(N) & \text{if} \quad |\widehat{\phi}(N) - \phi_*| > \frac{1}{\sqrt[4]{N}} \\ \phi_* & \text{if} \quad |\widehat{\phi}(N) - \phi_*| \leq \frac{1}{\sqrt[4]{N}} \end{cases}$$

Hodges estimator

It can be shown that

$$\lim_{N\to\infty} \eta\left[\widetilde{\phi}(N)\right] = \begin{cases} 1 & \text{if} \quad \phi_0 \neq \phi_* \\ \infty & \text{if} \quad \phi_0 = \phi_* \end{cases}$$

The evil of superefficiency
Zło superefektywności

Hodges estimator belongs to the class of superefficient estimators, i.e., the estimators that asymptotically beat the Cramér-Rao lower bound.

- In an analogous way one can design superefficient estimators for multivariate estimation problems

$$\widetilde{\boldsymbol{\phi}}(N) = \begin{cases} \widehat{\boldsymbol{\phi}}(N) & \text{if} \quad ||\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_*|| > \frac{1}{\sqrt[4]{N}} \\ \boldsymbol{\phi}_* & \text{if} \quad ||\widehat{\boldsymbol{\phi}}(N) - \boldsymbol{\phi}_*|| \leq \frac{1}{\sqrt[4]{N}} \end{cases}$$

- One can set infinitely many "traps" forming the set $A = \{\phi_i^*, -\infty < i < \infty\} \in \Omega_\phi$ – as long as $\mu(A) = 0$, where $\mu(A)$ denotes the Lebesque measure of A.

Superefficient estimators are not practically useful because, under the construction described above, it holds that

$$P(\phi_0 \in A) = 0$$

which means that the probability that one benefits from the estimator's superefficiency is 0.

Recursive estimation algorithms
Rekursywne algorytmy estymacji

$$\widehat{\boldsymbol{\theta}}(t) = g[\Omega(t)]$$

$$\Omega(t) = \{\xi(1), \ldots, \xi(t)\}$$
$$\xi(t) = \{u(t), y(t)\}$$

time-recursive algorithms
algorytmy rekursywne względem czasu

$$\mathbf{x}(t) = f[\mathbf{x}(t-1), \xi(t)]$$

where

$$\mathbf{x}(t) = \begin{bmatrix} \widehat{\boldsymbol{\theta}}(t) \\ \boldsymbol{\eta}(t) \end{bmatrix}$$

and $\boldsymbol{\eta}(t)$ denotes the vector of auxiliary variables ($wektor\ wielkości\ pomocniczych$).

Recursive least squares (RLS) algorithm
Rekursywny algorytm najmniejszych kwadratów

$$\mathbf{R}(t) = \mathbf{R}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)$$
$$\mathbf{s}(t) = \mathbf{s}(t-1) + y(t)\boldsymbol{\varphi}(t)$$

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}(t) &= \mathbf{R}^{-1}(t)\mathbf{s}(t) = \mathbf{R}^{-1}(t)[\mathbf{s}(t-1) + y(t)\boldsymbol{\varphi}(t)] \\
&= \mathbf{R}^{-1}(t)\left[\mathbf{R}(t-1)\widehat{\boldsymbol{\theta}}(t-1) + y(t)\boldsymbol{\varphi}(t)\right] \\
&= \mathbf{R}^{-1}(t)\left\{\mathbf{R}(t)\widehat{\boldsymbol{\theta}}(t-1) + \boldsymbol{\varphi}(t)[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)]\right\} \\
&= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{R}^{-1}(t)\boldsymbol{\varphi}(t)\varepsilon(t)
\end{aligned}$$

where

$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)$$

We will use the following matrix inversion lemma

<u>Lemma</u>

Provided that all inverses below exist, it holds that

$$\left[\mathbf{A} \pm \mathbf{b}\mathbf{b}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} \mp \frac{\mathbf{A}^{-1}\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}}{1 \pm \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}$$

Recursive least squares (RLS) algorithm

Let $\mathbf{P}(t) = \mathbf{R}^{-1}(t)$. Using the matrix inversion lemma with $\mathbf{A} = \mathbf{R}(t-1) = \mathbf{P}^{-1}(t-1)$ and $\mathbf{b} = \boldsymbol{\varphi}(t)$, one obtains

$$\begin{aligned}\mathbf{P}(t) &= \left[\mathbf{P}^{-1}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1} \\ &= \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}\end{aligned}$$

Let

$$\mathbf{k}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t) = \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}$$

The RLS algorithm can be summarized as follows

$$\begin{aligned}\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\ \mathbf{k}(t) &= \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \\ \widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\ \mathbf{P}(t) &= \mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}\end{aligned}$$

$$\boldsymbol{\eta}(t) = \begin{bmatrix} \mathrm{col}\{\mathbf{P}(t-1)\} \\ \boldsymbol{\varphi}(t) \end{bmatrix}$$

where $\mathrm{col}\{\mathbf{A}\}$ denotes a "vectorized" matrix.

Starting the RLS algorithm
Rozpoczynanie pracy algorytmu RLS

Exact initialization: $t = t_0 > \dim[\boldsymbol{\theta}]$

$$\mathbf{R}(t_0) = \sum_{i=1}^{t_0} \boldsymbol{\varphi}(i)\boldsymbol{\varphi}^{\mathrm{T}}(i)$$

$$\mathbf{s}(t_0) = \sum_{i=1}^{t_0} y(t)\boldsymbol{\varphi}(i)$$

$$\widehat{\boldsymbol{\theta}}(t_0) = \mathbf{R}^{-1}(t_0)\mathbf{s}(t_0)$$

Approximate initialization

For any initial conditions $\mathbf{P}(0), \widehat{\boldsymbol{\theta}}(0)$ the RLS estimates can be explicitly (non-recursively) written down in the form

$$\widehat{\boldsymbol{\theta}}(t) = \left[\mathbf{P}^{-1}(0) + \sum_{i=1}^{t} \boldsymbol{\varphi}(i)\boldsymbol{\varphi}^{\mathrm{T}}(i) \right]^{-1} \times \left[\mathbf{P}^{-1}(0)\widehat{\boldsymbol{\theta}}(0) + \sum_{i=1}^{t} y(i)\boldsymbol{\varphi}^{\mathrm{T}}(i) \right]$$

Hence

$$\mathbf{P}^{-1}(0) \longrightarrow 0 \quad \Longrightarrow \quad \widehat{\boldsymbol{\theta}}_{\mathrm{RLS}}(t) \longrightarrow \widehat{\boldsymbol{\theta}}_{\mathrm{LS}}(t)$$

Recommendation: use $\widehat{\boldsymbol{\theta}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = c\mathbf{I}$ where $c > 0$ is a "large number".

Square root algorithms
Algorytmy oparte na pierwiastkowaniu macierzy

Numerical robustness of the RLS algorithm can be significantly improved if, instead of the matrix $\mathbf{P}(t)$, its square root $\widetilde{\mathbf{P}}(t)$ is propagated:

$$\mathbf{P}(t) = \widetilde{\mathbf{P}}(t)\widetilde{\mathbf{P}}^T(t).$$

First, the nonnegative definite character of $\mathbf{P}(t)$ is preserved by virtue of the fact that the product of any square matrix and its transpose is always a nonnegative definite matrix. Second, the numerical conditioning of $\widetilde{\mathbf{P}}(t)$ is generally much better than that of $\mathbf{P}(t)$. Denote by

$$c(\mathbf{A}) = \delta(\mathbf{A}\mathbf{A}^{\mathrm{T}}) = \frac{\lambda_{\max}(\mathbf{A}\mathbf{A}^{\mathrm{T}})}{\lambda_{\min}(\mathbf{A}\mathbf{A}^{\mathrm{T}})}$$

the condition number of a square matrix $\mathbf{A}$.

Observe that

$$c[\mathbf{P}(t)] = c\left[\widetilde{\mathbf{P}}(t)\widetilde{\mathbf{P}}^{\mathrm{T}}(t)\right] = c^2[\widetilde{\mathbf{P}}(t)],$$

i.e. the condition number of $\widetilde{\mathbf{P}}(t)$ is the square root of $c[\mathbf{P}(t)]$.

Square root algorithms

This means that, while numerical operations with $\mathbf{P}(t)$ may encounter difficulties when $c[\mathbf{P}(t)] = 10^k$, the square root filter should function until $c[\mathbf{P}(t)] = 10^{2k}$. This effective double precision of the square root filter allows it to cope better (and longer) with numerical problems caused by a temporary lack of persistence of excitation.

The matrix square roots are not uniquely determined – infinitely many matrices $\widetilde{\mathbf{P}}(t)$ exist which obey the condition stated above. Uniqueness can be enforced by insisting that the square root factors be symmetric or triangular (with positive diagonal elements). However, even in this case, square roots can be computed in many different ways. This explains the existence of a large number of different square root algorithms described in the literature.

One of the first square root algorithms, known to have good numerical properties, was proposed by Potter. The derivation of Potter's algorithm is very simple. We note that the covariance update recursion of can be rewritten in the form

$$\widetilde{\mathbf{P}}(t)\widetilde{\mathbf{P}}^{\mathrm{T}}(t) = \widetilde{\mathbf{P}}(t-1)\left[\mathbf{I} - \frac{\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{\beta(t)}\right]\widetilde{\mathbf{P}}^{\mathrm{T}}(t-1)$$

where

$$\begin{aligned}
\mathbf{r}(t) &= \widetilde{\mathbf{P}}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t) \\
\beta(t) &= 1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{r}(t)
\end{aligned}$$

Potter's algorithm
Algorytm Pottera

It is straightforward to check that

$$\left[\mathbf{I} - \frac{\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{\beta(t)}\right] = \left[\mathbf{I} - \frac{\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{\alpha(t)}\right]\left[\mathbf{I} - \frac{\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{\alpha(t)}\right]$$

where $\alpha(t) = \beta(t) + \sqrt{\beta(t)}$. Therefore the square root $\widetilde{\mathbf{P}}(t)$ can be updated using the following recursion

$$\widetilde{\mathbf{P}}(t) = \widetilde{\mathbf{P}}(t-1)\left[\mathbf{I} - \frac{\mathbf{r}(t)\mathbf{r}^{\mathrm{T}}(t)}{\alpha(t)}\right]$$

The Potter's square root version of the RLS algorithm can be summarized as follows:

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\mathbf{r}(t) &= \widetilde{\mathbf{P}}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t) \\
\beta(t) &= 1 + \mathbf{r}^{\mathrm{T}}(t)\mathbf{r}(t) \\
\alpha(t) &= \beta(t) + \sqrt{\beta(t)} \\
\mathbf{k}(t) &= \frac{\widetilde{\mathbf{P}}(t-1)\mathbf{r}(t)}{\beta(t)} \\
\widetilde{\mathbf{P}}(t) &= \widetilde{\mathbf{P}}(t-1) - \frac{\beta(t)}{\alpha(t)}\,\mathbf{k}(t)\mathbf{r}^{\mathrm{T}}(t)
\end{aligned}$$

Fast least squares (FLS) algorithms
Szybkie algorytmy najmniejszych kwadratów

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t)$$

$$\mathbf{k}(t) = \mathbf{P}(t)\boldsymbol{\varphi}(t) = \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}$$

Is it possible to update the gain vector $\mathbf{k}(t)$ in a way that would not involve matrix computations?

$$\mathbf{k}(t-1) \longrightarrow \mathbf{k}(t)$$

Yes ... but ...

Consider the problem of identification of an AR(r) process

$$y(t) = \sum_{i=1}^{r} a_i y(t-i) + n(t)$$

$$\boldsymbol{\theta} = \begin{bmatrix} a_1 \\ \vdots \\ a_r \end{bmatrix}, \quad \boldsymbol{\varphi}(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-r) \end{bmatrix}$$

Fast least squares (FLS) algorithms
Szybkie algorytmy najmniejszych kwadratów

Example

$$
\begin{aligned}
e_0(t) &= \mathbf{s}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t) - y(t) \\
\mathbf{s}(t) &= \mathbf{s}(t-1) - \mathbf{k}(t)e_0(t) \\
e(t) &= \mathbf{s}^{\mathrm{T}}(t)\boldsymbol{\varphi}(t) - y(t) \\
r(t) &= r(t-1) + e(t)e_0(t) \\
w(t) &= e(t)/r(t) \\
\mathbf{n}(t) &= \mathbf{k}(t) + \mathbf{s}(t)w(t) \\
\begin{bmatrix} \mathbf{m}(t) \\ \mu(t) \end{bmatrix} &= \begin{bmatrix} w(t) \\ \mathbf{n}(t) \end{bmatrix} \\
\eta_0(t) &= \mathbf{d}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t+1) - y(t-r) \\
\mathbf{d}(t) &= \frac{\mathbf{d}(t-1) - \mathbf{m}(t)\eta_0(t)}{1 - \mu(t)\eta_0(t)} \\
\mathbf{k}(t+1) &= \mathbf{m}(t) - \mathbf{d}(t)\mu(t)
\end{aligned}
$$

Ljung, Morf & Falconer (1978)

comparison of computational complexity
(approximate number of multiply/add operations)

RLS $\sim 1.5\, r^2$ FLS $\sim 8\, r$

Estimation of noise variance
Estymacja wariancji szumu

We have shown that the estimate of σ_n^2 can be obtained from

$$\widehat{\sigma}_n^2(t) = \frac{q(t)}{t}$$

where

$$q(t) = \sum_{i=1}^{t} \left[y(i) - \boldsymbol{\varphi}^{\mathrm{T}}(i)\widehat{\boldsymbol{\theta}}(t) \right]^2$$

The quantity $q(t)$ can be evaluated recursively

$$q(t) = q(t-1) + \left[1 - \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{k}(t) \right] \varepsilon^2(t)$$

Note that

$$\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{k}(t) = \frac{\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} < 1$$

Recursive estimation algorithms
Rekursywne algorytmy estymacji

For simplicity we will restrict our attention to identification of autoregressive signals

$$y(t) = \sum_{i=1}^{r} a_{i,r} y(t-i) + n(t) = \boldsymbol{\varphi}_r^{\mathrm{T}}(t)\boldsymbol{\theta}_r + n(t)$$

Suppose that the order r of the autoregressive signal model is not known, and that one would like to establish AR models of orders $1, \ldots, r_{\max}$. Is it necessary to repeat calculations for different orders r? Is it possible to arrange calculations in such a way that the estimates of the coefficients of the model AR(r) would be evaluated by means of "correcting" the estimates obtained earlier for the model AR($r-1$) ?

order-recursive algorithms
algorytmy rekursywne względem rzędu

$$\mathbf{R}_r(N) = \sum_{t=1}^{N} \boldsymbol{\varphi}_r(t)\boldsymbol{\varphi}_r^{\mathrm{T}}(t), \quad \mathbf{s}_r(N) = \sum_{t=1}^{N} y(t)\boldsymbol{\varphi}_r(t)$$

$$\widehat{\boldsymbol{\theta}}_r(N) = \mathbf{R}_r^{-1}(N)\mathbf{s}_r(N)$$

$$r = 1, \ldots, r_{\max}$$

Order-recursive algorithms
Algorytmy rekursywne względem rzędu

Basic observation

The quantities $\mathbf{s}_r(N)$ and $\mathbf{R}_r(N)$ corresponding to different values of r are nested (wielkości te są *zagnieżdżone*)

$$\boldsymbol{\varphi}_r(t) \prec \boldsymbol{\varphi}_{r-1}(t) :$$

$$\boldsymbol{\varphi}_r(t) = \begin{bmatrix} y(t-1) \\ \vdots \\ y(t-r+1) \\ y(t-r) \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{r-1}(t) \\ y(t-r) \end{bmatrix}$$

$$\Downarrow$$

$$\mathbf{s}_r(N) = \begin{bmatrix} \mathbf{s}_{r-1}(N) \\ * \end{bmatrix}$$

$$\mathbf{R}_r(N) = \begin{bmatrix} \mathbf{R}_{r-1}(N) & * \\ * & * \end{bmatrix}$$

Levinson-Durbin algorithm
Algorytm Levinsona-Durbina

Levinson and Durbin derived an order-recursive algorithm which solves the Yule–Walker equations. Yule–Walker equations link parameters of an autoregressive process with its autocorrelation coefficients.

$$\begin{bmatrix} R_y(0) & R_y(1) & \dots & R_y(r-1) \\ R_y(1) & R_y(0) & \dots & R_y(r-2) \\ R_y(2) & R_y(1) & \dots & R_y(r-3) \\ \vdots & & & \vdots \\ R_y(r-1) & R_y(r-2) & \dots & R_y(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_r \end{bmatrix}$$

$$= \begin{bmatrix} R_y(1) \\ R_y(2) \\ R_y(3) \\ \vdots \\ R_y(r) \end{bmatrix}$$

$$\sigma_n^2 = R_y(0) - \sum_{i=1}^{r} a_i R_y(i)$$

<u>Notice</u>

The matrix made up of autocorrelation coefficients is symmetric, nonnegative definite and Toeplitz

Levinson-Durbin algorithm
Algorytm Levinsona-Durbina

Step 1

Based on the available data $\mathcal{Y}(N) = \{y(1), \ldots, y(N)\}$ estimate autocorrelation coefficients $R_y(0), \ldots, R_y(r_{\max})$ of the identified signal

$$\widehat{R}_y(i) = \frac{p_i(N)}{N}$$

$$p_i(N) = \sum_{t=i+1}^{N} y(t)y(t-i)$$

$$i = 1, \ldots, r_{\max}$$

Step 2

For each $i \in [1, r_{\max}]$ solve the Yule-Walker equations obtained by substituting autocorrelation coefficients with their estimates. This is equivalent to solving

$$\begin{bmatrix} p_0(N) & \ldots & p_{i-1}(N) \\ \vdots & & \vdots \\ p_{i-1}(N) & \ldots & p_0(N) \end{bmatrix} \begin{bmatrix} \widehat{a}_{1,i}(N) \\ \vdots \\ \widehat{a}_{i,i}(N) \end{bmatrix} = \begin{bmatrix} p_1(N) \\ \vdots \\ p_i(N) \end{bmatrix}$$

Levinson-Durbin algorithm
Algorytm Levinsona-Durbina

$$\sigma_0(N) = p_0(N)$$

$$k_i(N) = \frac{p_i(N) - \sum_{j=1}^{i-1} \widehat{a}_{j,i-1}(N) p_{i-j}(N)}{\sigma_{i-1}(N)}$$

$$\widehat{a}_{i,i}(N) = k_i(N)$$

$$\widehat{a}_{j,i}(N) = \widehat{a}_{j,i-1}(N) - k_i(N)\widehat{a}_{i-j,i-1}(N)$$
$$j = 1, \ldots, i-1$$

$$\sigma_i(N) = \left[1 - k_i^2(N)\right] \sigma_{i-1}(N)$$
$$i = 1, \ldots, r_{\max}$$

The quantities $k_1, \ldots, k_r$ are known as reflection coefficients (*współczynniki odbicia*) of an AR(r) model. Reflection coefficients allow one to determine autoregressive coefficients and vice versa

$$\{a_1, \ldots, a_r\} \iff \{k_1, \ldots, k_r\}$$

The AR(r) model is stable iff it holds that

$$|k_i| < 1, \quad 0 \le i \le r$$

When the correlation matrix is positive definite, the Levinson-Durbin algorithm always yields a stable AR model.

Relationship to the method of least squares
Związek z metodą najmniejszych kwadratów

Suppose that the available data record is extended with r zero measurements at the beginning and at the end

$$y(0) = y(-1) = \ldots = y(1-r) = 0$$
$$y(N+1) = y(N+2) = \ldots = y(N+r) = 0$$

Then it holds that

$$[\mathbf{R}_r(N)]_{ij} = \left[\sum_{t=1}^{N+r} \boldsymbol{\varphi}_r(t)\boldsymbol{\varphi}_r^{\mathrm{T}}(t)\right]_{ij}$$
$$= \sum_{t=1}^{N+r} y(t-i)y(t-j) = \sum_{s=|i-j|+1}^{N} y(s)y(s-|i-j|)$$
$$= p_{|i-j|}(N)$$

Hence, in such a case, the Levinson-Durbin algorithm yields the same results as the least squares algorithm.

<u>Practical fix</u>

To improve estimation results it is usually recommended to weight the measurements prior to applying the Levinson-Durbin algorithm $y_w(t) = w(t)y(t)$, $t = 1, \ldots, N$, where $\{w(t)\}$ denotes any bell-shaped symmetric window sequence, such as Hann (raised cosine) window.

Ladder/lattice algorithms
Algorytmy drabinkowe/kratowe

Lattice/ladder algorithms estimate reflection coefficients and are made up of identical functional blocks. Hence, they are suitable for hardware implementations.

Example

initial conditions:

$$k_1(0) = \ldots = k_r(0) = 0$$

$$\sigma_0(t) = \sigma_0(t-1) + y^2(t)$$

$$\nu_0(t) = \eta_0(t) = y(t)/\sqrt{\sigma_0(t)}$$

$$k_i(t) = k_i(t-1)\sqrt{1-\nu_{i-1}^2(t)}\sqrt{1-\eta_{i-1}^2(t-1)} + \nu_{i-1}(t)\eta_{i-1}(t-1)$$

$$\nu_i(t) = \frac{\nu_{i-1}(t) - k_i(t)\eta_{i-1}(t-1)}{\sqrt{1-k_i^2(t)}\sqrt{1-\eta_{i-1}^2(t-1)}}$$

$$\eta_i(t) = \frac{\eta_{i-1}(t-1) - k_i(t)\nu_{i-1}(t)}{\sqrt{1-k_i^2(t)}\sqrt{1-\nu_{i-1}^2(t)}}$$

$$i = 1, \min(t, r)$$

Lee, Morf & Friedlander (1981)

Lee-Morf-Friedlander algorithm
Algorytm Lee-Morfa-Friedlandera

Let

$$F(a,b,c) = \frac{a - bc}{\sqrt{1-c^2}\,\sqrt{1-b^2}}$$

$$G(a,b,c) = bc + a\sqrt{1-b^2}\,\sqrt{1-c^2}$$

It can be shown that

$$|a|, |b|, |c| < 1 \quad \Longrightarrow \quad |F(a,b,c)|, |G(a,b,c)| < 1$$

Note that the Lee-Morf-Friedlander algorithm can be written down in the following form

$$\begin{aligned}
k_i(t) &= G[\; k_i(t-1), \nu_{i-1}(t), \eta_{i-1}(t-1) \;] \\
\nu_i(t) &= F[\; \nu_{i-1}(t), k_i(t), \eta_{i-1}(t-1) \;] \\
\eta_i(t) &= F[\; \eta_{i-1}(t-1), k_i(t), \nu_{i-1}(t) \;]
\end{aligned}$$

$$|k_i(t)|, |\nu_i(t)|, |\eta_i(t)| < 1 \quad \forall t$$

The Lee-Morf-Friedlander algorithm always yields a stable AR model.

Identification of nonstationary systems
Identyfikacja obiektów niestacjonarnych

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + n(t)$$

Denote by $\{w(i)\}$ the weighting sequence, a sequence of nonnegative coefficients obeying

$$w(0) = 1 \geq w(1) \geq \ldots \geq 0$$

$$\sum_{i=0}^{\infty} w(i) < \infty$$

which will be used to 'localize' estimation, namely to make its results insensitive (or at least less sensitive) to observations collected in the remote past. To achieve the effect of forgetting or discounting 'old' data, the sum of squares minimized in the method of least squares is replaced with the weighted sum of squares, resulting in the following weighted least squares (WLS) estimator:

$$\widehat{\boldsymbol{\theta}}(t) = \arg\min_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) \tag{1}$$

where

$$J_t(\boldsymbol{\theta}) = \sum_{i=0}^{t-1} w(i) \left[y(t-i) - \boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\theta}\right]^2$$

Weighted least squares estimators
Ważone estymatory najmniejszych kwadratów

Minimization of the weighted measure of fit $J_t(\boldsymbol{\theta})$ can be carried out analogously as minimization of an ordinary sum of squares

$$
\begin{aligned}
J_t(\boldsymbol{\theta}) &= \sum_{i=0}^{t-1} w(i) \left[y(t-i) - \boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\theta}\right]^2 \\
&= \sum_{i=0}^{t-1} w(i) y^2(t-i) \\
&\quad - 2\left[\sum_{i=0}^{t-1} w(i) y(t-i) \boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]\boldsymbol{\theta} \\
&\quad + \boldsymbol{\theta}^{\mathrm{T}}\left[\sum_{i=0}^{t-1} w(i) \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]\boldsymbol{\theta}
\end{aligned}
$$

$$
\begin{aligned}
\nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) &= -2\sum_{i=0}^{t-1} w(i) y(t-i) \boldsymbol{\varphi}(t-i) \\
&\quad + 2\left[\sum_{i=0}^{t-1} w(i) \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]\boldsymbol{\theta}
\end{aligned}
$$

$$
\nabla^2_{\boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}} J_t(\boldsymbol{\theta}) = 2\left[\sum_{i=0}^{t-1} w(i) \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]
$$

This leads to

$$\widehat{\boldsymbol{\theta}}(t) = \mathbf{R}^{-1}(t)\mathbf{s}(t)$$

$$\mathbf{R}(t) = \sum_{i=0}^{t-1} w(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)$$

$$\mathbf{s}(t) = \sum_{i=0}^{t-1} w(i)y(t-i)\boldsymbol{\varphi}(t-i)$$

The two windows most frequently used in practice are a rectangular window (*okno prostokątne*)

$$w(i) = \begin{cases} 1 & 0 \le i < n \\ 0 & \text{elsewhere} \end{cases}$$

and an exponential window (*okno wykładnicze*)

$$w(i) = \lambda^i, \qquad 0 < \lambda < 1$$

In the first case only the n most recent observations (equally weighted) are used for parameter estimation. This approach is often called a sliding window approach as it corresponds to moving a constant-length analysis window along the data.

In the second case the weights decay at an exponential rate, gradually decreasing the influence of old data on the current parameter estimates. This approach is frequently termed exponential forgetting. The constant λ is called a forgetting constant (*stała zapominania*).

Recursive exponentially weighted least squares estimators

Rekursywne estymatory najmniejszych kwadratów z zapominaniem wykładniczym

Note that

$$\begin{aligned}
\mathbf{R}(t) &= \sum_{i=0}^{t-1} \lambda^i \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i) \\
&= \lambda \mathbf{R}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t) \\
\mathbf{s}(t) &= \sum_{i=0}^{t-1} \lambda^i y(t-i)\boldsymbol{\varphi}(t-i) \\
&= \lambda \mathbf{s}(t-1) + y(t)\boldsymbol{\varphi}(t)
\end{aligned}$$

and consequently

$$\begin{aligned}
\widehat{\boldsymbol{\theta}}(t) &= \mathbf{R}^{-1}(t)\mathbf{s}(t) \\
&= \mathbf{R}^{-1}(t)\left[\lambda \mathbf{s}(t-1) + y(t)\boldsymbol{\varphi}(t)\right] \\
&= \mathbf{R}^{-1}(t)\Big\{\mathbf{R}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
&\qquad\qquad + \boldsymbol{\varphi}(t)\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)\right]\Big\} \\
&= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{R}^{-1}(t)\boldsymbol{\varphi}(t)\varepsilon(t)
\end{aligned}$$

Let

$$\mathbf{P}(t) = \mathbf{R}^{-1}(t) = \left[\lambda \mathbf{P}^{-1}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]^{-1}$$

Recursive exponentially weighted least squares algorithm

Rekursywne algorytm najmniejszych kwadratów z zapominaniem wykładniczym

After applying the matrix inversion lemma with $\mathbf{A} = \lambda \mathbf{P}^{-1}(t-1)$ and $\mathbf{b} = \boldsymbol{\varphi}(t)$, one arrives at the following recursive algorithm

$$
\begin{aligned}
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\mathbf{k}(t) &= \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \\
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\mathbf{P}(t) &= \frac{1}{\lambda}\left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)}{\lambda + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}\right]
\end{aligned}
$$

After setting $\widetilde{\mathbf{P}}(t) = \mathbf{P}(t)/\lambda$, the algorithm given above can be also expressed in the form

$$
\begin{aligned}
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\mathbf{k}(t) &= \frac{\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)} \\
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\widetilde{\mathbf{P}}(t) &= \frac{1}{\lambda}\left[\widetilde{\mathbf{P}}(t-1) - \frac{\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)}\right]
\end{aligned}
$$

Vector forgetting
Zapominanie wektorowe

When the conventional exponentially weighted least squares (EWLS) algorithm is used, the rate of forgetting is the same in all directions of the parameter space, i.e. it is implicitly assumed that all process coefficients vary with the same speed. Replacing the single forgetting factor λ in the covariance update equation of the EWLS algorithm with the set of forgetting factors $\lambda_1, \ldots, \lambda_n$, assigned individually for each of n process coefficients, one arrives at the following EWLS algorithm with vector forgetting

$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)$$

$$\mathbf{k}(t) = \frac{\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)}$$

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t)$$

$$\widetilde{\mathbf{P}}(t) = \mathbf{L}\left[\widetilde{\mathbf{P}}(t-1) - \frac{\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)}{1 + \boldsymbol{\varphi}^{\mathrm{T}}(t)\widetilde{\mathbf{P}}(t-1)\boldsymbol{\varphi}(t)}\right]\mathbf{L}$$

where $\mathbf{L} = \mathrm{diag}\{\lambda_1^{-1/2}, \ldots, \lambda_n^{-1/2}\}$, $0 < \lambda_i \leq 1$, $i = 1, \ldots, n$.

Vector forgetting allows one to better incorporate the prior knowledge about the speed of variation of different components of $\boldsymbol{\theta}$. Generally, the faster a given process parameter varies with time, the smaller the forgetting factor to guarantee a good tracking performance.

Square root exponentially weighted least squares algorithm

Algorytm najmniejszych kwadratów z zapominaniem wykładniczym oparty na pierwiastkowaniu macierzy

$$
\begin{aligned}
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\mathbf{r}(t) &= \widetilde{\mathbf{P}}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t) \\
\beta(t) &= \lambda + \mathbf{r}^{\mathrm{T}}(t)\mathbf{r}(t) \\
\alpha(t) &= \beta(t) + \sqrt{\lambda\beta(t)} \\
\mathbf{k}(t) &= \frac{\widetilde{\mathbf{P}}(t-1)\mathbf{r}(t)}{\beta(t)} \\
\widetilde{\mathbf{P}}(t) &= \frac{1}{\sqrt{\lambda}}\left[\widetilde{\mathbf{P}}(t-1) - \frac{\beta(t)}{\alpha(t)}\,\mathbf{k}(t)\mathbf{r}^{\mathrm{T}}(t)\right]
\end{aligned}
$$

Fast exponentially weighted least squares algorithm

Szybki algorytm najmniejszych kwadratów z zapominaniem wykładniczym

$$
\begin{aligned}
e_0(t) &= \mathbf{s}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t) - y(t) \\
\mathbf{s}(t) &= \mathbf{s}(t-1) - \mathbf{k}(t)e_0(t) \\
e(t) &= \mathbf{s}^{\mathrm{T}}(t)\boldsymbol{\varphi}(t) - y(t) \\
r(t) &= \lambda r(t-1) + e(t)e_0(t) \\
w(t) &= e(t)/r(t) \\
\mathbf{n}(t) &= \mathbf{k}(t) + \mathbf{s}(t)w(t) \\
\begin{bmatrix} \mathbf{m}(t) \\ \mu(t) \end{bmatrix} &= \begin{bmatrix} w(t) \\ \mathbf{n}(t) \end{bmatrix} \\
\eta_0(t) &= \mathbf{d}^{\mathrm{T}}(t-1)\boldsymbol{\varphi}(t+1) - y(t-r) \\
\mathbf{d}(t) &= \frac{\mathbf{d}(t-1) - \mathbf{m}(t)\eta_0(t)}{1 - \mu(t)\eta_0(t)} \\
\mathbf{k}(t+1) &= \mathbf{m}(t) - \mathbf{d}(t)\mu(t)
\end{aligned}
$$

Ladder/lattice exponentially weighted least squares algorithm

Drabinkowy/kratowy algorytm najmniejszych kwadratów z zapominaniem wykładniczym

initial conditions:

$$k_1(0) = \ldots = k_r(0) = 0$$

$$\sigma_0(t) = \lambda\sigma_0(t-1) + y^2(t)$$

$$\nu_0(t) = \eta_0(t) = y(t)/\sqrt{\sigma_0(t)}$$

$$k_i(t) = k_i(t-1)\sqrt{1-\nu_{i-1}^2(t)}\sqrt{1-\eta_{i-1}^2(t-1)} + \nu_{i-1}(t)\eta_{i-1}(t-1)$$

$$\nu_i(t) = \frac{\nu_{i-1}(t) - k_i(t)\eta_{i-1}(t-1)}{\sqrt{1-k_i^2(t)}\sqrt{1-\eta_{i-1}^2(t-1)}}$$

$$\eta_i(t) = \frac{\eta_{i-1}(t-1) - k_i(t)\nu_{i-1}(t)}{\sqrt{1-k_i^2(t)}\sqrt{1-\nu_{i-1}^2(t)}}$$

$$i = 1, \min(t, r)$$

Sliding window least squares algorithm
Algorytm najmniejszych kwadratów z oknem prostokątnym

$$\widehat{\boldsymbol{\theta}}(t,n) = \mathbf{R}^{-1}(t,n)\mathbf{s}(t,n)$$

$$\mathbf{R}(t,n) = \sum_{i=0}^{n-1} \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)$$

$$\mathbf{s}(t,n) = \sum_{i=0}^{n-1} y(t-i)\boldsymbol{\varphi}(t-i)$$

The following recursive relationships can be easily verified

$$\begin{aligned}
\mathbf{R}(t,n+1) &= \mathbf{R}(t-1,n) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t) \\
\mathbf{R}(t,n) &= \mathbf{R}(t,n+1) - \boldsymbol{\varphi}(t-n)\boldsymbol{\varphi}^{\mathrm{T}}(t-n) \\
\mathbf{s}(t,n+1) &= \mathbf{s}(t-1,n) + y(t)\boldsymbol{\varphi}(t) \\
\mathbf{s}(t,n) &= \mathbf{s}(t,n+1) - y(t-n)\boldsymbol{\varphi}(t-n)
\end{aligned}$$

Let

$$\mathbf{P}(t,n+1) = \mathbf{R}^{-1}(t,n+1), \quad \mathbf{P}(t,n) = \mathbf{R}^{-1}(t,n)$$

and recall that

$$\left[\mathbf{A} \pm \mathbf{b}\mathbf{b}^{\mathrm{T}}\right]^{-1} = \mathbf{A}^{-1} \mp \frac{\mathbf{A}^{-1}\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}}{1 \pm \mathbf{b}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{b}}$$

Step 1

$$
\begin{aligned}
\varepsilon_+(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1,n) \\
\mathbf{k}(t,n+1) &= \frac{\mathbf{P}(t-1,n)\boldsymbol{\varphi}(t)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1,n)\boldsymbol{\varphi}(t)} \\
\widehat{\boldsymbol{\theta}}(t,n+1) &= \widehat{\boldsymbol{\theta}}(t-1,n) + \mathbf{k}(t,n+1)\varepsilon_+(t) \\
\mathbf{P}(t,n+1) &= \mathbf{P}(t-1,n) \\
&\quad - \frac{\mathbf{P}(t-1,n)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1,n)}{1+\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1,n)\boldsymbol{\varphi}(t)}
\end{aligned}
$$

Step 2

$$
\begin{aligned}
\varepsilon_-(t) &= y(t-n) - \boldsymbol{\varphi}^{\mathrm{T}}(t-n)\widehat{\boldsymbol{\theta}}(t,n+1) \\
\mathbf{k}(t,n) &= \frac{\mathbf{P}(t,n+1)\boldsymbol{\varphi}(t-n)}{1-\boldsymbol{\varphi}^{\mathrm{T}}(t-n)\mathbf{P}(t,n+1)\boldsymbol{\varphi}(t-n)} \\
\widehat{\boldsymbol{\theta}}(t,n) &= \widehat{\boldsymbol{\theta}}(t,n+1) - \mathbf{k}(t,n)\varepsilon_-(t) \\
\mathbf{P}(t,n) &= \mathbf{P}(t,n+1) \\
&\quad + \frac{\mathbf{P}(t,n+1)\boldsymbol{\varphi}(t-n)\boldsymbol{\varphi}^{\mathrm{T}}(t-n)\mathbf{P}(t,n+1)}{1-\boldsymbol{\varphi}^{\mathrm{T}}(t-n)\mathbf{P}(t,n+1)\boldsymbol{\varphi}(t-n)}
\end{aligned}
$$

recursive sliding window least squares algorithm

Effective memory of the weighted least squares algorithm

Efektywna pamięć algorytmu ważonych najmniejszych kwadratów

Suppose that the process made up of regression vectors is stationary: $\{\boldsymbol{\varphi}(t)\}$ is a stationary process with positive definite correlation matrix $\mathrm{E}[\,\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^T(t)\,] = \mathbf{R}_0$.

Then the expected value of the weighted regression matrix obeys

$$\mathrm{E}[\mathbf{R}(t)] = \mathrm{E}\left[\sum_{i=0}^{t-1} w(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^T(t-i)\right] = k_t\mathbf{R}_0,$$

where the quantity

$$k_t = \sum_{i=0}^{t-1} w(i)$$

can be interpreted as the 'window area'. According to the derived formula, the coefficient $k_t \leq t$ characterizes the amount of energy extracted from the input/output data as a result of applying the weighting sequence $\{w(i)\}$. Since, with weighting, the average energy content of the set of t observations is the same as the energy content of the set of k_t observations without weighting, the quantity k_t is usually called the effective number of observations or the effective window width.

Equivalent memory of the weighted least squares algorithm

Równoważna pamięć algorytmu ważonych najmniejszych kwadratów

Suppose additionally that $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0, \forall t$, i.e., that the estimated parameter vector is time-invariant. When estimation is carried out using the method of least squares, one obtains

$$\mathrm{cov}\left[\widehat{\boldsymbol{\theta}}(t)\right] \cong \frac{\sigma_0^2 \mathbf{R}_0^{-1}}{t}$$

The analogous formula for the weighted least squares estimator takes the form

$$\mathrm{cov}\left[\widehat{\boldsymbol{\theta}}(t)\right] \cong \frac{\sigma_0^2 \mathbf{R}_0^{-1}}{l_t}, \quad l_t = \frac{\left[\sum_{i=0}^{t-1} w(i)\right]^2}{\sum_{i=0}^{t-1} w^2(i)}$$

Note that in the stationary case all weighted least squares estimators characterized by the same value of l_t are equivalent from the viewpoint of estimation accuracy, irrespective of the shape of the corresponding windows. Consequently, l_t characterizes the amount of information about $\boldsymbol{\theta}_0$ which is extracted from the input/output data as a result of applying the method of weighted least squares. We shall call l_t the equivalent number of observations

Comparison of two measures of the estimation memory

Porównanie dwóch miar pamięci estymacji

exponential window:

$$k_t = \frac{1-\lambda^t}{1-\lambda} \quad \underset{t\to\infty}{\longrightarrow} \quad \frac{1}{1-\lambda} = k_\infty$$

$$l_t = \frac{(1+\lambda)(1-\lambda^t)}{(1-\lambda)(1+\lambda^t)} \quad \underset{t\to\infty}{\longrightarrow} \quad \frac{1+\lambda}{1-\lambda} = l_\infty$$

Note that for λ close to 1, it holds that

$$\frac{l_\infty}{k_\infty} = 1+\lambda \cong 2$$

rectangular window:

$$k_t = l_t = n, \quad \forall t \geq n$$

It can be easily shown that for any weighting sequence the following inequality is true

$$l_t \geq k_t, \quad \forall t$$

Tracking properties of WLS estimators
Własności śledzące estymatorów WLS

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + n(t)$$

Assumption 1

$\{n(t)\}$ is a sequence of zero-mean, independent and identically distributed random variables with variance σ_0^2.

Assumption 2

$\{\boldsymbol{\varphi}(t)\}$ is a stationary ergodic process, independent of $\{\boldsymbol{\theta}(t)\}$, with positive definite correlation matrix $\mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right] = \mathbf{R}_0 > \mathbf{O}$. The variables $n(t)$ and $\boldsymbol{\varphi}(t)$ are mutually uncorrelated $\forall t$.

Denote by $\bar{\boldsymbol{\theta}}(t) = \mathrm{E}_{\Omega(t)}\left[\widehat{\boldsymbol{\theta}}(t)\right]$ the mean trajectory of parameter estimates. Note that the WLS estimator can be written down in the form

$$\widehat{\boldsymbol{\theta}}(t) = \left[\frac{1}{k_t}\sum_{i=0}^{t-1} w(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]^{-1} \times \left[\frac{1}{k_t}\sum_{i=0}^{t-1} w(i)y(t-i)\boldsymbol{\varphi}(t-i)\right]$$

$$\cong \mathbf{R}_0^{-1}\left[\frac{1}{k_t}\sum_{i=0}^{t-1} w(i)y(t-i)\boldsymbol{\varphi}(t-i)\right]$$

The last transition is valid for sufficiently large values of k_t because, owing to the ergodicity of $\{\boldsymbol{\varphi}(t)\}$, it holds that

$$\lim_{k_t \to \infty} \frac{1}{k_t} \sum_{i=0}^{t-1} w(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i) = \mathbf{R}_0$$

Therefore

$$\begin{aligned}
\bar{\boldsymbol{\theta}}(t) \cong \mathrm{E}\left[\widehat{\boldsymbol{\theta}}(t)\right] &= \frac{1}{k_t}\,\mathbf{R}_0^{-1}\mathrm{E}\left[\sum_{i=0}^{t-1} w(i)y(t-i)\boldsymbol{\varphi}(t-i)\right] \\
&= \frac{1}{k_t}\,\mathbf{R}_0^{-1}\mathrm{E}\left[\sum_{i=0}^{t-1} w(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\boldsymbol{\theta}(t-i)\right] \\
&+ \frac{1}{k_t}\,\mathbf{R}_0^{-1}\mathrm{E}\left[\sum_{i=0}^{t-1} w(i)n(t-i)\boldsymbol{\varphi}(t-i)\right] \\
&= \frac{1}{k_t}\sum_{i=0}^{t-1} w(i)\boldsymbol{\theta}(t-i)
\end{aligned}$$

When examining the tracking capabilities of the WLS algorithm, one is interested primarily in its long-term or steady-state behavior, i.e. the behavior observed when effective and equivalent numbers of observations approach their limiting (steady-state) values

$$k_\infty = \lim_{t\to\infty} k_t, \qquad l_\infty = \lim_{t\to\infty} l_t$$

Tracking properties of WLS estimators
Własności śledzące estymatorów WLS

The steady-state counterpart of the relationship derived above can be obtained by setting $t \to \infty$ or by assuming that the infinite observation history is available at instant t

$$\bar{\boldsymbol{\theta}}(t) \cong \sum_{i=0}^{\infty} h(i)\boldsymbol{\theta}(t-i)$$

$$h(i) = \frac{w(i)}{k_\infty}$$

The sequence $\{h(i)\}$ is called the impulse response associated with the WLS estimator (*odpowiedź impulsowa skojarzona z estymatorem WLS*).

According to the obtained formula, after a sufficiently long time, $\{\bar{\boldsymbol{\theta}}(t)\}$ can be approximately viewed as an output of a linear causal time-invariant filter with impulse response $\{h(i)\}$ excited by the process $\{\boldsymbol{\theta}(t)\}$.

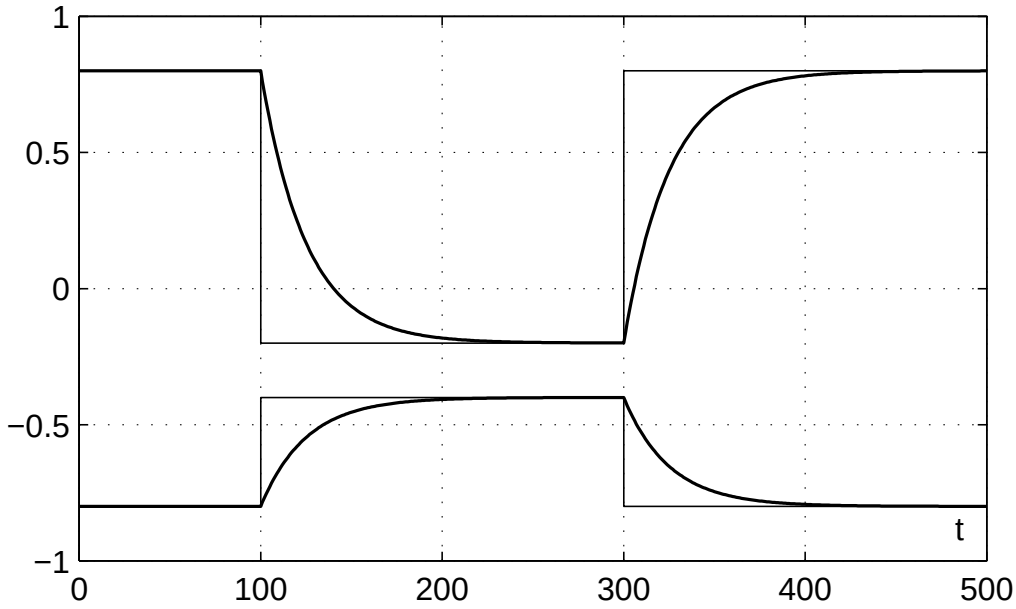
Note that

$$\sum_{i=0}^{\infty} h(i) = 1$$

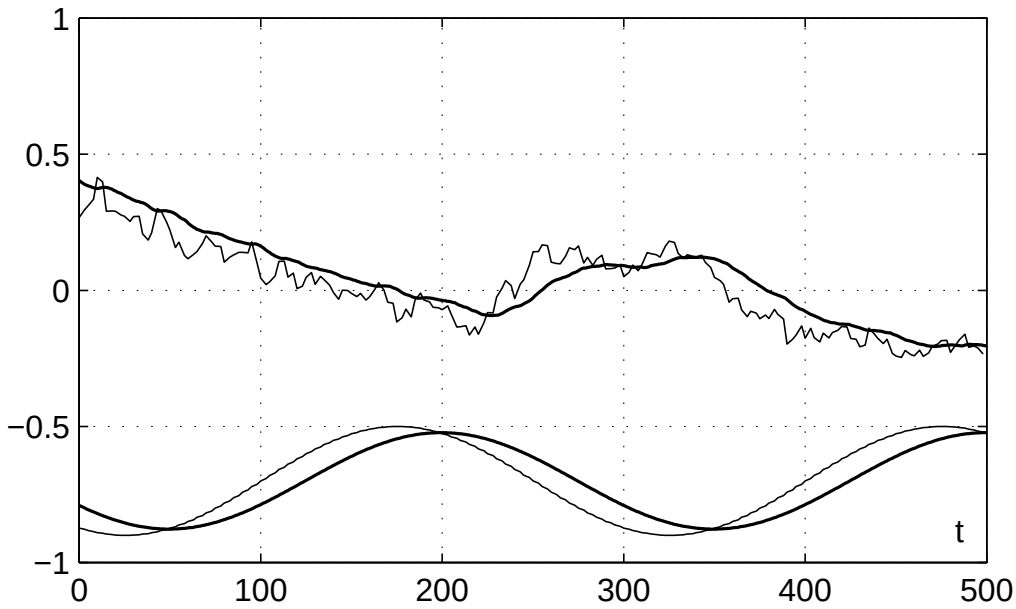
<u>Example</u>

In the exponential forgetting case, one obtains

$$h(i) = \begin{cases} (1-\lambda)\lambda^i & i \geq 0 \\ 0 & i < 0 \end{cases}, \qquad H(q^{-1}) = \frac{1-\lambda}{1-\lambda q^{-1}}$$

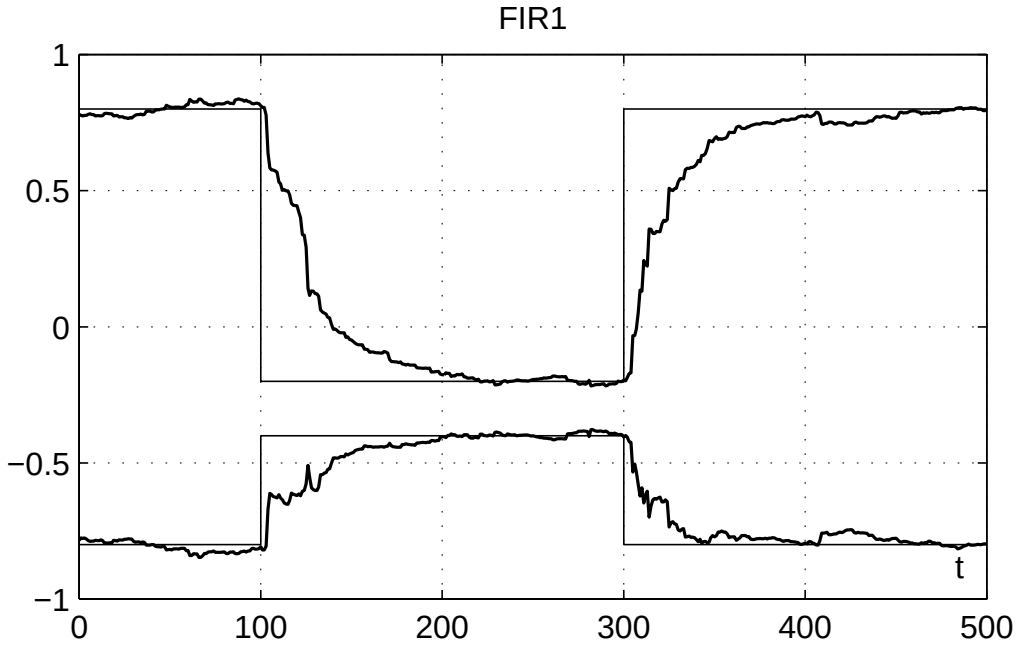


(a)

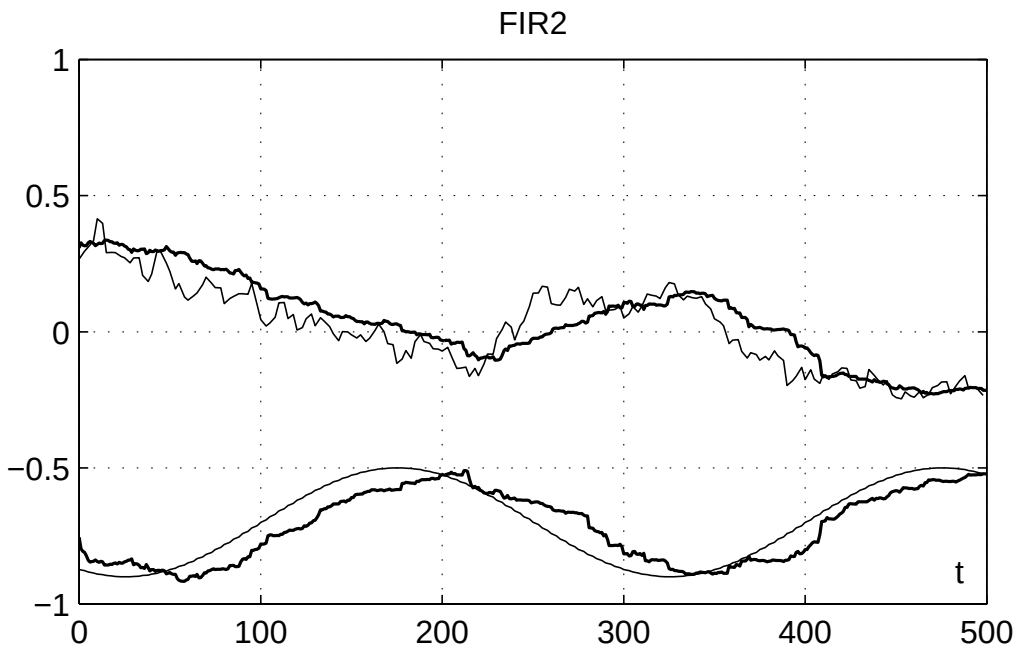


(b)

Output of a linear filter associated with the exponentially weighted least squares estimator for a system with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\lambda = 0.96$).

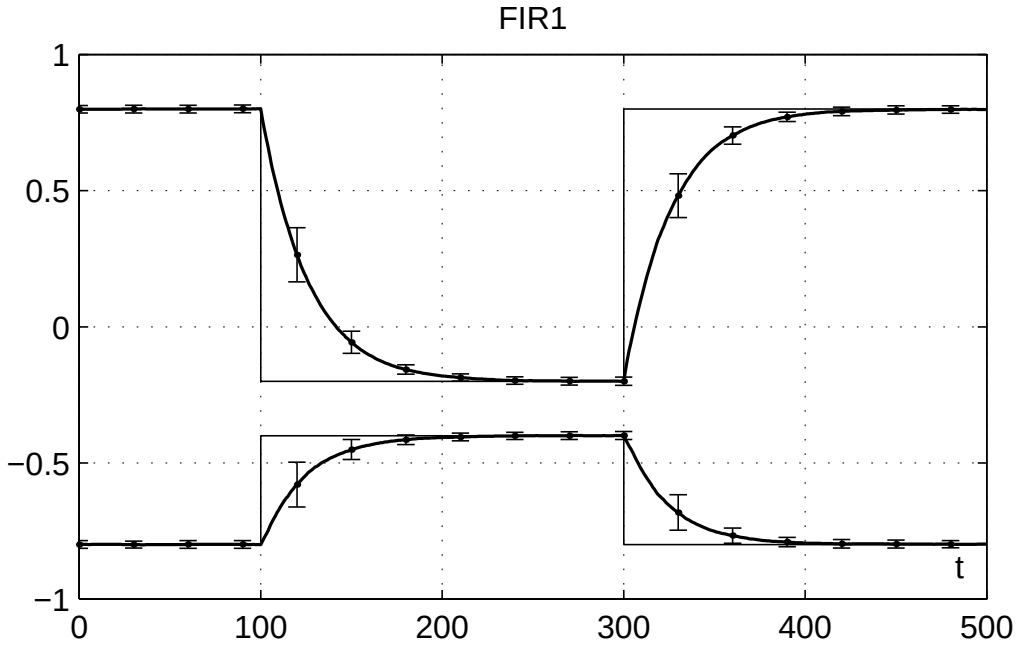


(a)

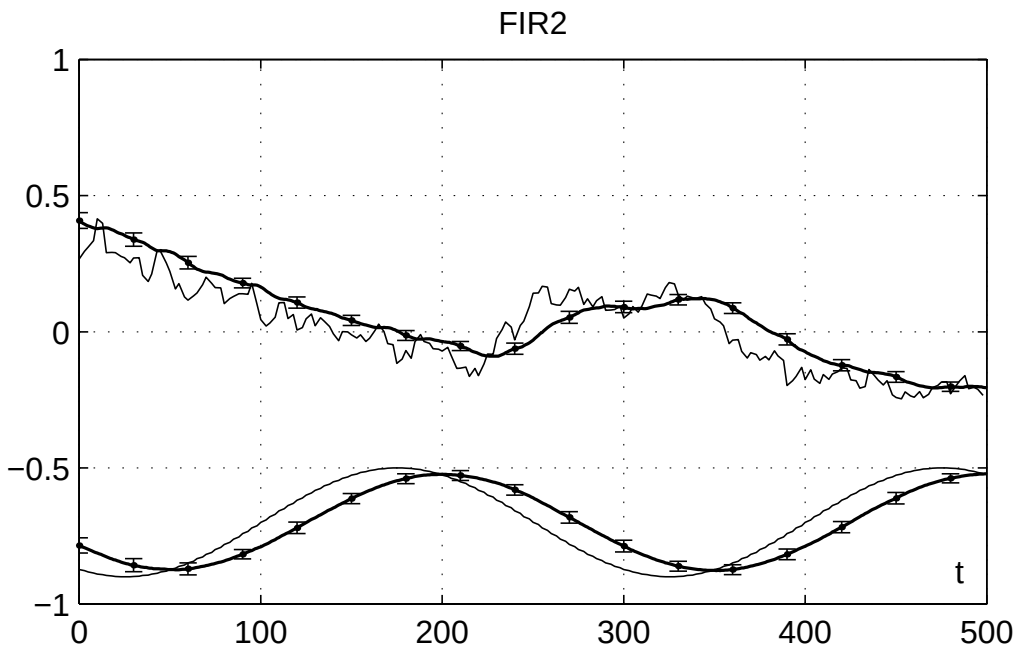


(b)

Parameter estimates yielded by the exponentially weighted least squares algorithm for a single realization of an FIR system with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\lambda = 0.96$).

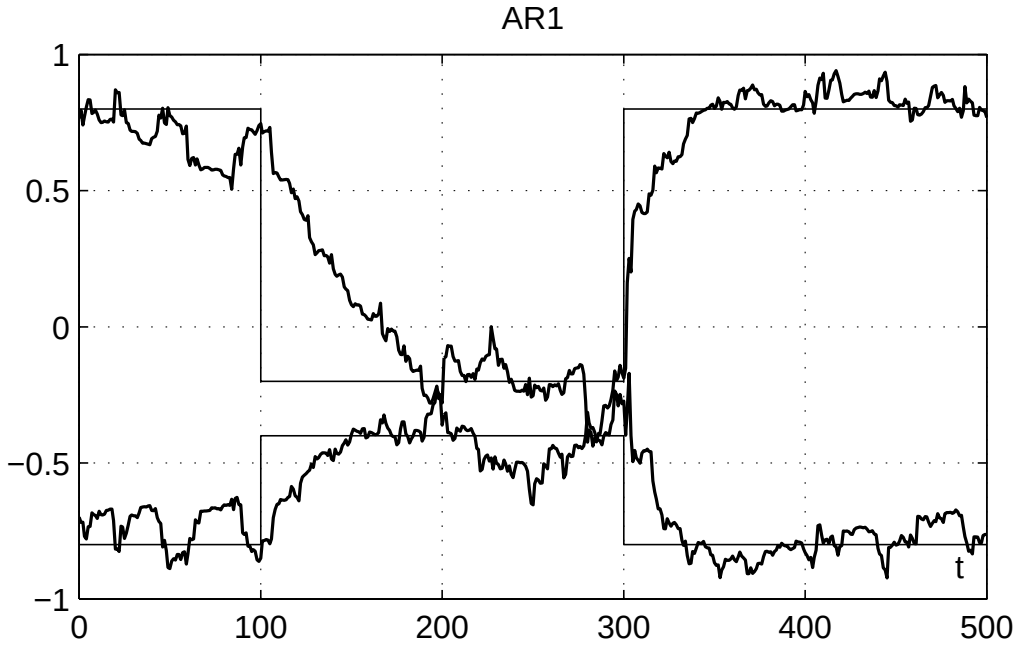


(a)

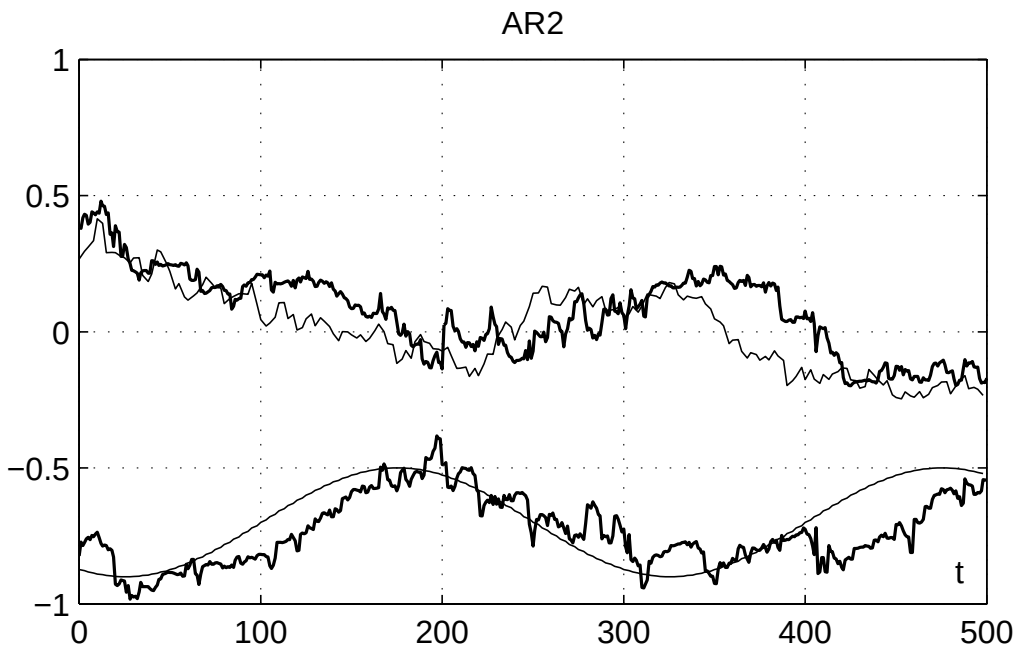


(b)

Average trajectories of parameter estimates yielded by the exponentially weighted least squares algorithm for an FIR system with jump parameter changes (a) and continuous parameter changes (b); vertical bars show standard deviation of the estimates.

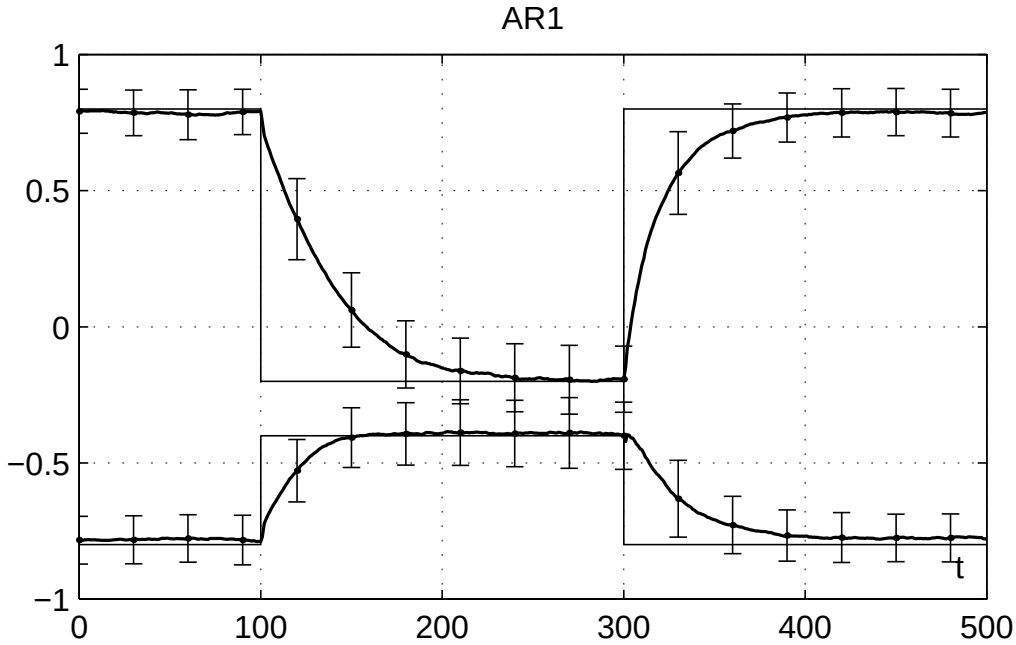


(a)

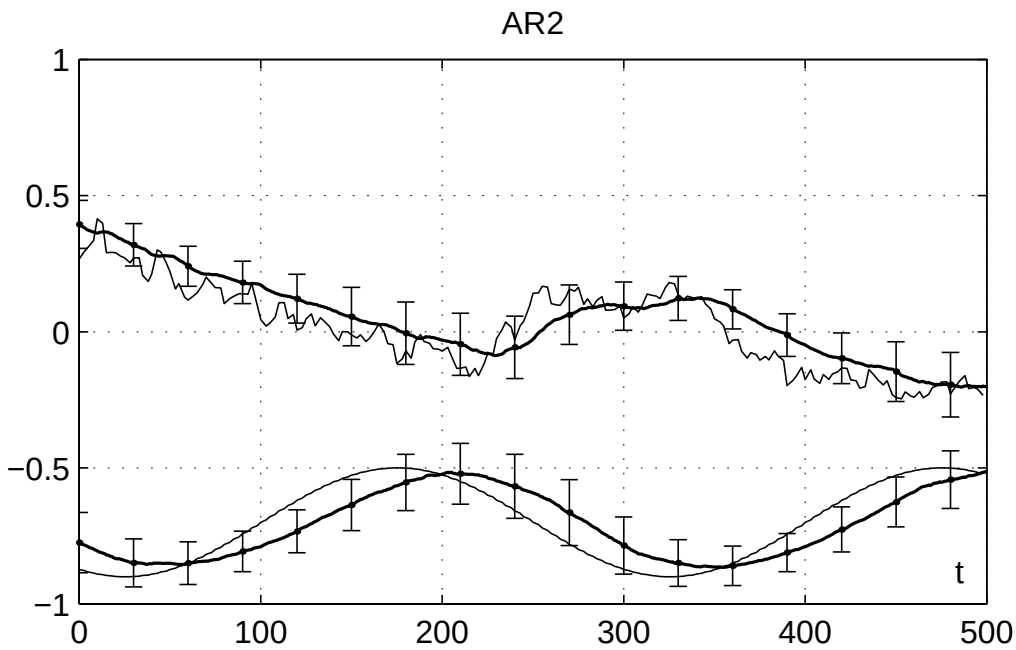


(b)

Parameter estimates yielded by the exponentially weighted least squares algorithm for a single realization of an AR process with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\lambda = 0.96$).



(a)



(b)

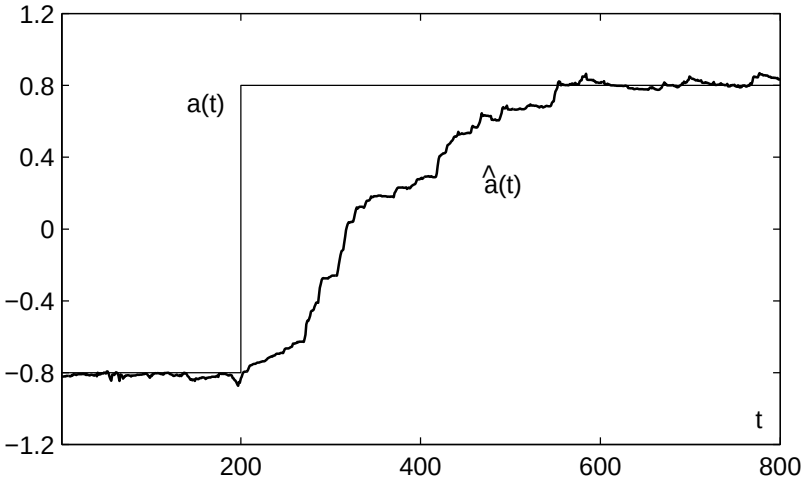
Average trajectories of parameter estimates yielded by the exponentially weighted least squares algorithm for an AR process with jump parameter changes (a) and continuous parameter changes (b); vertical bars show standard deviation of the estimates.

How to choose estimation memory?
Jak dobierać pamięć estymacji?

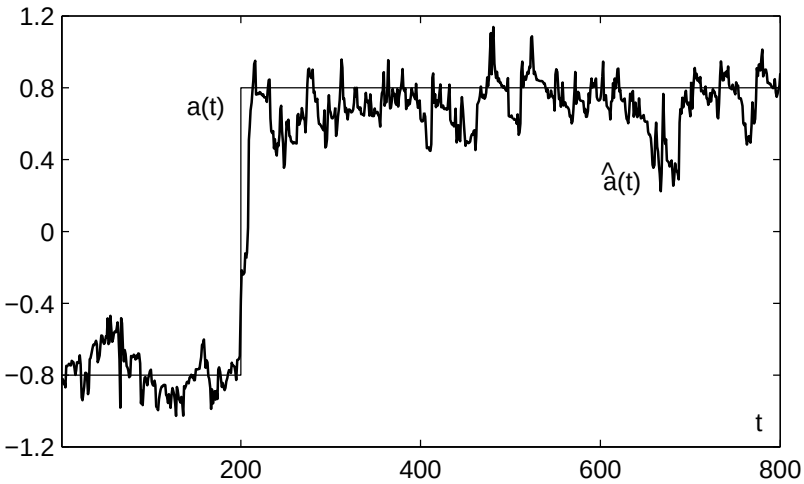
The plots show evolution of an autoregressive coefficient of a first-order AR process

$$y(t) = a(t)y(t-1) + n(t),$$

subject to a jump change at instant $t = 200$, and its exponentially weighted least squares estimates obtained for two values of the forgetting constant, $\lambda = 0.99$ and $\lambda = 0.9$.



$\lambda = 0.99$



$\lambda = 0.9$

How to choose estimation memory?
Jak dobierać pamięć estymacji?

Note that for $\lambda = 0.99$ (long memory) the estimation algorithm provides high estimation accuracy in periods where parameters do not change or change slowly, but is pretty insensitive to parameter changes. On the other hand, $\lambda = 0.9$ (short memory) guarantees fast parameter tracking at the expense of low estimation accuracy attainable under time-invariant conditions.

$$\begin{aligned}\mathrm{MSE}\left[\widehat{\boldsymbol{\theta}}(t)\right] &= \mathrm{E}\left[\| \widehat{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}(t) \|^2\right] \\ &= \mathrm{E}\left[\| \widehat{\boldsymbol{\theta}}(t) - \bar{\boldsymbol{\theta}}(t) \|^2\right] + \| \bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}(t) \|^2\end{aligned}$$

where $\bar{\boldsymbol{\theta}}(t) = \mathrm{E}\left[\widehat{\boldsymbol{\theta}}(t)\right]$ denotes the mean trajectory of parameter estimates.

$$\mathrm{E}\left[\| \widehat{\boldsymbol{\theta}}(t) - \bar{\boldsymbol{\theta}}(t) \|^2\right] \propto \frac{1}{l_\infty} \quad - \quad \text{variance error}$$

$$\| \bar{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}(t) \|^2 \propto l_\infty \quad - \quad \text{bias error}$$

Conclusion: estimation memory of a parameter tracking algorithm should be chosen so as to provide a reasonable balance between the bias component (tracking) and the variance component (accuracy) of the mean square parameter tracking error.

Automatic adjustment of the forgetting constant

Automatyczne strojenie stałej zapominania

So far we have considered constant-memory WLS filters only. An obvious way of preserving identifiability conditions is via appropriate memory scheduling. When the amount of incoming information drops, the process of discarding 'old' measurements should be limited and eventually stopped; this will prevent the regression matrix from getting singular. In other words, the memory of an adaptive filter should be inversely proportional to the average information content of the incoming data.

In the exponentially weighted least squares framework, the variable-memory scheme can easily be realized by replacing the time-invariant forgetting constant λ with a time-varying forgetting factor $\lambda(t)$. This results in

$$\begin{aligned}
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\mathbf{k}(t) &= \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda(t) + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)} \\
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\mathbf{P}(t) &= \frac{1}{\lambda(t)}\left[\mathbf{P}(t-1) - \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)}{\lambda(t) + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}\right]
\end{aligned}$$

where

$$\mathbf{P}^{-1}(t) = \mathbf{R}(t) = \lambda(t)\mathbf{R}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)$$

It is easy to show that the recursive estimation formulas given above can be written in the following explicit form:

$$\widehat{\boldsymbol{\theta}}(t) = \arg\min_{\boldsymbol{\theta}} \sum_{i=0}^{t-1} w_t(i) \left[y(t-i) - \boldsymbol{\varphi}^T(t-i)\boldsymbol{\theta}\right]^2$$

$$= \left[\sum_{i=0}^{t-1} w_t(i)\boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i)\right]^{-1}$$

$$\times \left[\sum_{i=0}^{t-1} w_t(i)y(t-i)\boldsymbol{\varphi}(t-i)\right]$$

where

$$w_t(0) = 1$$

$$w_t(i) = \prod_{j=0}^{i-1} \lambda(t-j), \quad 0 < i \leq t-1$$

Since

$$w_t(i) = \lambda(t)w_{t-1}(i-1), \quad i = 1.\ldots,t$$

the effective memory of the exponentially weighted least squares algorithm with variable forgetting factor

$$k_t = \sum_{i=0}^{t-1} w_t(i)$$

can be evaluated recursively:

$$k_t = \lambda(t)k_{t-1} + 1$$

Automatic adjustment of the forgetting constant

Automatyczne strojenie stałej zapominania

Forgetting constant can be chosen so as to stabilize the weighted sum of residual errors

$$q(t) = \sum_{i=0}^{t-1} w_t(i) \left[y(t-i) - \boldsymbol{\varphi}^T(t-i)\widehat{\boldsymbol{\theta}}(t) \right]^2$$

Note that for a time-invariant system $[\boldsymbol{\theta}(t) = \boldsymbol{\theta}_0]$ and fixed-rate forgetting $[\lambda(t) = \lambda_o,\ 1 - \lambda_o \ll 1]$, it holds that $\widehat{\boldsymbol{\theta}}(t) \cong \boldsymbol{\theta}_0$. Hence

$$y(t-i) - \boldsymbol{\varphi}^T(t-i)\widehat{\boldsymbol{\theta}}(t) \cong n(t-i)$$

leading to

$$\lim_{t\to\infty} \mathrm{E}[q(t)] \cong k_o \sigma_0^2 = q_o$$

where

$$k_o = \sum_{i=0}^{\infty} \lambda_o^i = \frac{1}{1-\lambda_o}$$

denotes the effective memory of the corresponding filter. This means that if $q(t)$ is stabilized at q_o and the intensity of the measurement noise is constant, the corresponding values of $\lambda(t)$ will fluctuate around

$$\lambda_o = 1 - \frac{1}{k_o}$$

If the residual errors increase, e.g. due to faster parameter variation, stabilization of $q(t)$, which can be approximately achieved by setting

$$\lambda(t) = \frac{1}{1 + \dfrac{\epsilon^2(t)}{[1+\gamma(t)]q_o}}$$

$$\gamma(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t)$$

will result in lowering the value of the forgetting constant.

The forgetting factor $\lambda(t)$ obtained from the above formula can take any value between 0 and 1. In practice it may be dangerous to adopt values too far from one, so a lower threshold $\lambda_{\min}$ for $\lambda(t)$ is set. This results in the following algorithm

$$\begin{aligned}
\varepsilon(t) &= y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1) \\
\gamma(t) &= \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}(t-1)\boldsymbol{\varphi}(t) \\
\lambda(t) &= \max\left\{\lambda_{\min},\ \frac{1}{1+\epsilon^2(t)/[(1+\gamma(t))q_o]}\right\} \\
\mathbf{k}(t) &= \frac{\mathbf{P}(t-1)\boldsymbol{\varphi}(t)}{\lambda(t)+\gamma(t)} \\
\widehat{\boldsymbol{\theta}}(t) &= \widehat{\boldsymbol{\theta}}(t-1) + \mathbf{k}(t)\varepsilon(t) \\
\mathbf{P}(t) &= \frac{1}{\lambda(t)}\left[\mathbf{I} - \mathbf{k}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t\right]\mathbf{P}(t-1)
\end{aligned}$$

variable forgetting factor (VFF) algorithm

Parallel estimation approach

Podejście oparte na estymacji równoległej

Suppose that J exponentially weighted least squares algorithms, equipped with different forgetting factors $\lambda_1, \ldots, \lambda_J$, are run in parallel

$$\varepsilon_j(t) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_j(t-1)$$

$$\mathbf{k}_j(t) = \frac{\mathbf{P}_j(t-1)\boldsymbol{\varphi}(t)}{\lambda_j + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}_j(t-1)\boldsymbol{\varphi}(t)}$$

$$\widehat{\boldsymbol{\theta}}_j(t) = \widehat{\boldsymbol{\theta}}_j(t-1) + \mathbf{k}_j(t)\varepsilon_j(t)$$

$$\mathbf{P}_j(t) = \frac{1}{\lambda_j}\left[\mathbf{P}_j(t-1) - \frac{\mathbf{P}_j(t-1)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}_j(t-1)}{\lambda_j + \boldsymbol{\varphi}^{\mathrm{T}}(t)\mathbf{P}_j(t-1)\boldsymbol{\varphi}(t)}\right]$$

$$j = 1, \ldots, J$$

Then one can use the following combined parameter estimate with improved tracking capabilities

$$\widehat{\boldsymbol{\theta}}(t) = \sum_{j=1}^{J} \mu_j(t)\widehat{\boldsymbol{\theta}}_j(t)$$

$$\mu_j(t) = \frac{\eta_j(t)}{\sum_{j=1}^{J}\eta_j(t)}, \quad \eta_i(t) = \left[\sum_{m=0}^{M-1}\varepsilon_j^2(t-m)\right]^{-M/2}$$

where $\mu_j(t), j = 1, \ldots, J$ denote the so-called model credibility coefficients (*współczynniki wiarygodności modeli*) and M denotes the width of the local evaluation interval.

recommendations / zalecenia:

1. The values of M should be large enough to avoid erratic decisions ($\geq$20), but small enough to guarantee adaptivity ($\leq$50).

2. The effective memory lengths of the competing algorithms should form a geometric progression. For example, one can set

$$k_{j+1}(\infty) = 2k_j(\infty), \quad j = 1, \ldots, J-1$$

3. Some of the quantities involved in computation of credibility coefficients may take very large values. Therefore, to avoid problems due to numerical overflow, one can replace the quantities $\eta_j(t)$ with $\eta_j^\circ(t)$:

$$\eta_j^\circ(t) = \exp[\chi_j(t)], \quad \chi_j(t) = \psi_j(t) - \psi_{\max}(t)$$
$$\psi_j(t) = \log[\eta_j(t)], \quad \psi_{\max}(t) = \max_{1 \leq j \leq J} \psi_j(t)$$

Note that for large values of M, the weighted estimation formula will *de facto* reduce itself to

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}_{j^\star(t)}(t), \quad j^\star(t) = \arg \min_{1 \leq j \leq J} \sum_{m=0}^{M-1} \varepsilon_j^2(t-m)$$

Numerical stability of recursive estimation algorithms

Stabilność numeryczna rekursywnych algorytmów estymacji

Many tests have confirmed that the fast least squares (FLS) algorithms are not numerically robust – they tend to accumulate errors due to finite-precision arithmetic, causing the parameter estimates to diverge. The episodes of unstable behavior of an estimation algorithm are often called its bursts (*wybuchy*).

In some cases, bursts can be eliminated by restarting the algorithm each time when the numerical problem of some kind is detected. For example, in the case of the Ljung-Morf-Falconer FLS algorithm, one can use the following Lin's rescue device:

restart the algorithm each time when

$$1 - \mu(t)\eta_0(t) < 0$$

No problems like this were observed for the classical recursive least squares algorithm. Why some of the algorithms burst, while the other ones do not? Is it possible to tell them apart? Is it possible to "fix" the bursting algorithm?

Exponential stability
Stabilność wykładnicza

All time-recursive algorithms can be written down in the following (nonlinear) state space form

$$\mathbf{x}(t) = f\left[\mathbf{x}(t-1), \xi(t)\right]$$
$$\widehat{\boldsymbol{\theta}}(t) = \left[\mathbf{I} \;\; \mathbf{O}\right] \mathbf{x}(t)$$

where

$$\mathbf{x}(t) = \begin{bmatrix} \widehat{\boldsymbol{\theta}}(t) \\ \boldsymbol{\eta}(t) \end{bmatrix}$$

and $\boldsymbol{\eta}(t)$ denotes the vector of auxiliary variables.

From the mathematical viewpoint, the fast least squares algorithms transform data into parameter estimates

$$\Omega(t) \longrightarrow \widehat{\boldsymbol{\theta}}(t)$$

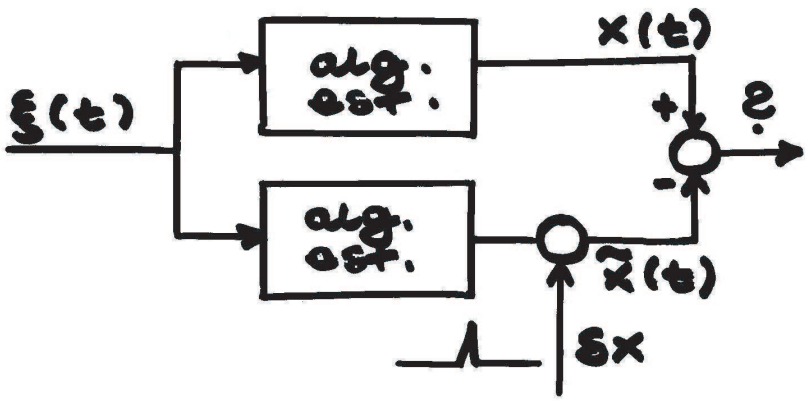
in exactly the same way as the recursive least squares algorithm, i.e., the same data should result in the same parameter estimates. In practice, differences occur because all arithmetic operations are realized with finite precision.

In order to guarantee numerical robustness, recursive estimation algorithms should have the exponential stability (*stabilność wykładnicza*) property: under exact arithmetic the effects of an isolated perturbation, added to the algorithm's state vector, should decay to zero exponentially fast.

Exponential stability condition
Warunek stabilności wykładniczej

Consider two identical recursive identification algorithms, both working with infinite precision. Suppose that one of them is subject to an isolated perturbance added to the state vector at the instant t_0

$$\widetilde{\mathbf{x}}(t) = \mathbf{x}(t), \quad t < t_0$$
$$\widetilde{\mathbf{x}}(t_0) = \mathbf{x}(t_0) + \delta\mathbf{x}$$
$$\widetilde{\mathbf{x}}(t) = f\left[\widetilde{\mathbf{x}}(t-1), \xi(t)\right], \quad t > t_0$$



The algorithm is exponentially stable if for *any* threshold $\delta_0 > 0$, one can find such constants $c > 0$ and $0 < \mu < 1$, that for *every* bounded-norm perturbation $\delta\mathbf{x}$, $\| \delta\mathbf{x} \| \leq \delta_0$, the following inequality holds true

$$\| \widetilde{\mathbf{x}}(t) - \mathbf{x}(t) \| \leq c\mu^{t-t_0}, \quad \forall t \geq t_0$$

Ljung-Morf-Falconer's algorithm is not exponentially stable

Noncausal identification of nonstationary processes
Nieprzyczynowa identyfikacja procesów niestacjonarnych

Qualitative analysis of parameter tracking properties of weighted least squares estimators shows that the expected trajectory of parameter estimates can be regarded, to a certain extent, as a delayed version of the true parameter trajectory. More precisely, $\widehat{\boldsymbol{\theta}}(t)$ can be viewed as an estimate of $\boldsymbol{\theta}(t-\tau_e)$ rather than of $\boldsymbol{\theta}(t)$, where τ_e is an estimation delay (*opóźnienie estymacji*), given approximately by

$$\tau_e \cong \frac{l_\infty}{2}$$

Whether or not the estimation delay is an undesirable phenomenon depends on the particular application at hand. Clearly, in adaptive prediction or control applications, delay (phase) distortions are as harmful as the scale (amplitude) distortions; after all we need a good estimate of $\boldsymbol{\theta}(t)$ and not of $\boldsymbol{\theta}(t-\tau_e)$ in order to predict or control the system's behavior at instant t. There are some other applications, however, called nearly real-time, where the constant delay of the estimated parameter trajectory with respect to the true parameter trajectory can be mitigated by incorporating into the adaptive loop a decision delay (*opóźnienie decyzyjne*) of τ_e samples.

Fixed-interval smoothing
Wygładzanie ze stałym przedziałem

$$\Omega(N) \quad \longrightarrow \quad \{\widehat{\boldsymbol{\theta}}(t), t = 1, \ldots, N\}$$

$$\widehat{\boldsymbol{\theta}}(t) = \mathbf{R}^{-1}(t)\mathbf{r}(t)$$

$$\mathbf{R}(t) = \sum_{i=1}^{N} \lambda^{|t-i|} \boldsymbol{\varphi}(i)\boldsymbol{\varphi}^{\mathrm{T}}(i)$$

$$\mathbf{s}(t) = \sum_{i=1}^{N} \lambda^{|t-i|} y(i)\boldsymbol{\varphi}(i)$$

$$t = 1, \ldots, N$$

exponentially weighted least squares smoother

$$forward - time\ run:$$

$$\begin{aligned}
[\ \widetilde{\mathbf{R}}(0) &= 0, \quad \widetilde{\mathbf{s}}(0) = 0\ ] \\
\widetilde{\mathbf{R}}(t) &= \lambda\widetilde{\mathbf{R}}(t-1) + \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t) \\
\widetilde{\mathbf{s}}(t) &= \lambda\widetilde{\mathbf{s}}(t-1) + y(t)\boldsymbol{\varphi}(t) \\
t &= 1, \ldots, N
\end{aligned}$$

$$backward - time\ run:$$

$$\begin{aligned}
[\ \mathbf{R}(N) &= \widetilde{\mathbf{R}}(N), \quad \mathbf{s}(N) = \widetilde{\mathbf{s}}(N)\ ] \\
\mathbf{R}(t) &= \lambda\mathbf{R}(t+1) + (1-\lambda^2)\widetilde{\mathbf{R}}(t) \\
\mathbf{s}(t) &= \lambda\mathbf{s}(t+1) + (1-\lambda^2)\widetilde{\mathbf{s}}(t) \\
\widehat{\boldsymbol{\theta}}(t) &= \mathbf{R}^{-1}(t)\mathbf{s}(t) \\
t &= N-1, \ldots, 1
\end{aligned}$$

Basis functions approach (BF)
Metoda funkcji bazowych

Consider the analysis frame $T = [1, \ldots, N]$ of length N and let

$$\{f_j(t), j = 1, \ldots, k\}, \quad t \in T$$

be a set of linearly independent discrete-time functions (sequences) defined on T, called basis functions. We shall assume that each time-varying process coefficient can be represented by a linear combination of basis functions

$$\theta_i(t) = \sum_{j=1}^{k} c_{ij} f_j(t), \quad t \in T, \quad i = 1, \ldots, n$$

If some prior knowledge about the nature of process time variation is available, one may attempt to choose the basis functions so as to capture the 'dominant' trends in coefficient changes. In all other cases, selection of the basis has to rely on some general approximation guidelines. The most frequently used general-purpose bases involve the powers of time (Taylor series approximation)

$$f_j(t) = t^{j-1}, \quad j = 1, 2, \ldots$$

and cosines (Fourier series approximation)

$$f_j(t) = \cos \frac{\pi(j-1)t}{N-1}, \quad j = 1, 2, \ldots$$

Basis functions approach
Metoda funkcji bazowych

We will show that the time-varying process

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + n(t)$$

can be described by a time-invariant model. Denote by $\varphi_i(t)$ is the ith component of $\boldsymbol{\varphi}(t)$, and by $\boldsymbol{\psi}(t)$ the $[nk \times 1]$ generalized regression vector associated with the analyzed process

$$\boldsymbol{\psi}(t) = [\varphi_1(t)\mathbf{f}^{\mathrm{T}}(t), \ldots, \varphi_n(t)\mathbf{f}^{\mathrm{T}}(t)]^{\mathrm{T}}$$

where

$$\mathbf{f}(t) = [\, f_1(t), \ldots, f_k(t)\,]^{\mathrm{T}}$$

Furthermore, let

$$\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^{\mathrm{T}}, \ldots, \boldsymbol{\gamma}_n^{\mathrm{T}}]^{\mathrm{T}}, \quad \boldsymbol{\gamma}_i = [c_{i1}, \ldots, c_{ik}]^{\mathrm{T}}$$

be the $[nk \times 1]$ vector of all coefficients used to describe the process time variation.

Basis functions approach
Metoda funkcji bazowych

Using the shorthand notation introduced above the process equation

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + n(t)$$

can be rewritten as

$$y(t) = \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{\gamma} + n(t), \quad t \in T$$

i.e. the time-varying process of order n can be represented by a linear time-invariant model of order nk.

The estimate of $\boldsymbol{\gamma}$ can be obtained using the method of least squares

$$\begin{aligned}
\widehat{\boldsymbol{\gamma}}(N) &= \arg\min_{\boldsymbol{\gamma}} \sum_{t=1}^{N} [y(t) - \boldsymbol{\psi}^{\mathrm{T}}(t)\boldsymbol{\gamma}]^2 \\
&= \left(\sum_{t=1}^{N} \boldsymbol{\psi}(t)\boldsymbol{\psi}^{\mathrm{T}}(t) \right)^{-1} \left(\sum_{t=1}^{N} y(t)\boldsymbol{\psi}(t) \right) \\
&= \mathbf{G}^{-1}(N)\mathbf{h}(N)
\end{aligned}$$

provided, of course, that the regression matrix $\mathbf{G}(N)$ is invertible.

Basis functions approach
Metoda funkcji bazowych

The estimates of time-varying system parameters can be obtained from

$$\begin{aligned}\widehat{\theta}_i(t) &= \sum_{j=1}^{k} \widehat{c}_{ij}(N) f_j(t) \\ &= \widehat{\boldsymbol{\gamma}}_i^{\mathrm{T}}(N)\mathbf{f}(t), \quad t \in T, \quad i = 1, \ldots, n\end{aligned}$$

The estimate of the input noise variance can be obtained from

$$\widehat{\sigma}_n^2(N) = \frac{1}{N} \sum_{t=1}^{N} \left[y(t) - \boldsymbol{\psi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\gamma}}(N\right]^2.$$

Both the parameter estimates and the variance estimate can be evaluated recursively using the algorithms derived for time-invariant systems, provided that the regression vector $\boldsymbol{\varphi}(t)$ is replaced with the generalized regression vector $\boldsymbol{\psi}(t)$.

Gradient search
Poszukiwanie gradientowe

The least mean squares (LMS) approach to system identification or tracking is based on a simple gradient-based error minimization strategy. Denote by $J(\boldsymbol{\theta})$ the performance measure parameterized in terms of $\boldsymbol{\theta}$. Assuming that $J(\boldsymbol{\theta})$ is differentiable with respect to $\boldsymbol{\theta}$, a simple iterative algorithm searching for its minimum can be constructed, using the method of steepest descent:

$$\widehat{\boldsymbol{\theta}}^{(i)} = \widehat{\boldsymbol{\theta}}^{(i-1)} - \frac{1}{2}\,\mu \nabla_{\boldsymbol{\theta}} J(\widehat{\boldsymbol{\theta}}^{(i-1)})$$

where μ denotes a small positive stepsize parameter. The minus sign assures that we move $down$ the local gradient, i.e. towards the minimum of $J(\cdot)$ and the scaling factor $1/2$ is introduced to simplify further results.

Let

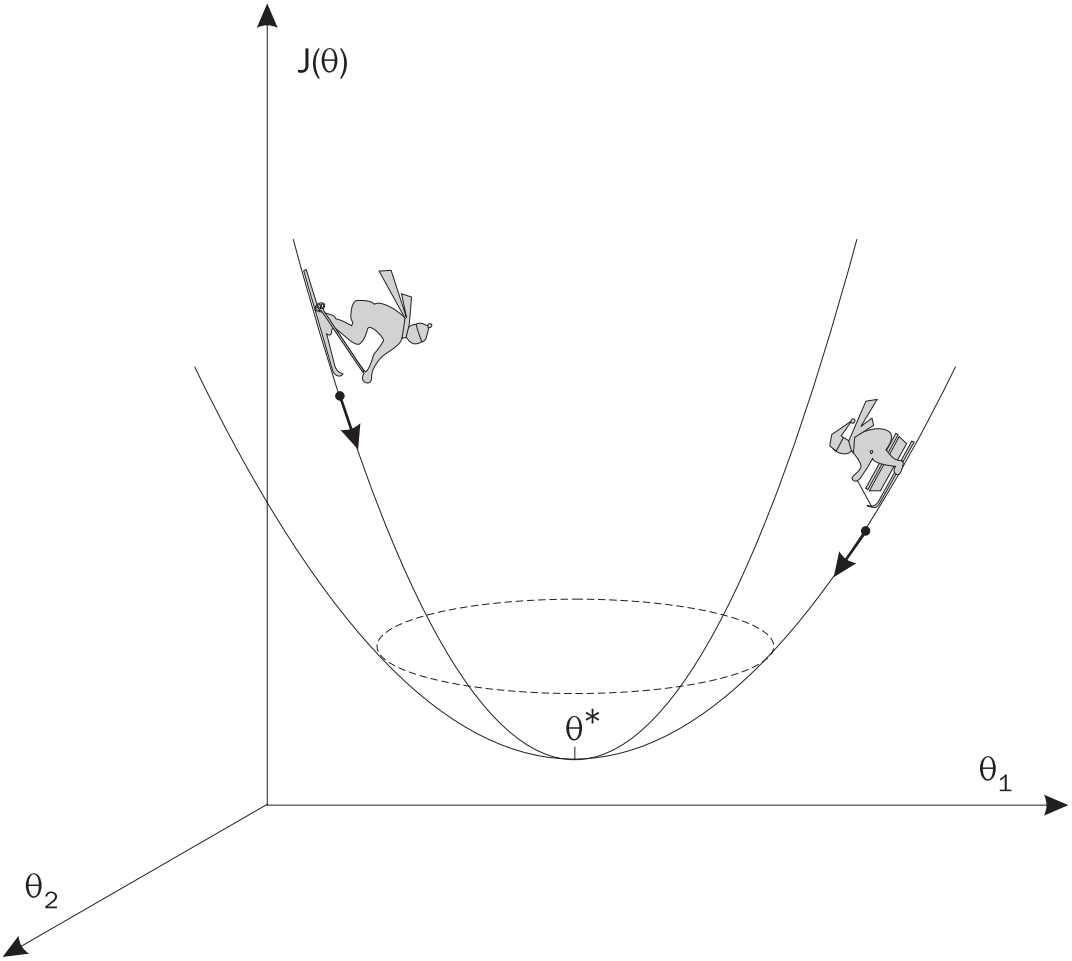
$$\boldsymbol{\theta}_0 = \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

If $J(\boldsymbol{\theta})$ is quadratic around $\boldsymbol{\theta}_0$ and μ is sufficiently small, one can easily show that

$$\widehat{\boldsymbol{\theta}}^{(i)} \quad \underset{i\to\infty}{\longrightarrow} \quad \boldsymbol{\theta}_0$$

provided that the initial estimate $\widehat{\boldsymbol{\theta}}^{(0)}$ is close enough to $\boldsymbol{\theta}_0$.

Gradient search
Poszukiwanie gradientowe



The steepest descent minimization strategy:
always move down the hill.

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 + n(t)$$

If the analyzed system is time invariant and the joint second-order statistic of $\{y(t), \boldsymbol{\varphi}(t)\}$ is known a priori one can set

$$J(\boldsymbol{\theta}) = \mathrm{E}\left\{\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2\right\}$$

Gradient search
Poszukiwanie gradientowe

$$
\begin{aligned}
J(\boldsymbol{\theta}) &= \mathrm{E}\left\{\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2\right\} \\
&= \mathrm{E}[y^2(t)] - 2\mathbf{s}_0^{\mathrm{T}}\boldsymbol{\theta} + \boldsymbol{\theta}^{\mathrm{T}}\mathbf{R}_0\boldsymbol{\theta}
\end{aligned}
$$

where

$$
\mathbf{s}_0 = \mathrm{E}[y(t)\boldsymbol{\varphi}(t)], \quad \mathbf{R}_0 = \mathrm{E}[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)]
$$

$$
\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = -2\mathbf{s}_0 + 2\mathbf{R}_0\boldsymbol{\theta}
$$

$$
\widehat{\boldsymbol{\theta}}^{(i)} = \widehat{\boldsymbol{\theta}}^{(i-1)} + \mu\left[\mathbf{s}_0 - \mathbf{R}_0\boldsymbol{\theta}^{(i-1)}\right]
$$

If the joint second-order statistic of $\{y(t), \boldsymbol{\varphi}(t)\}$ is not known, one can use the following iterative algorithm

$$
\widehat{\boldsymbol{\theta}}^{(i)} = \widehat{\boldsymbol{\theta}}^{(i-1)} + \mu\left[\widehat{\mathbf{s}}_0(N) - \widehat{\mathbf{R}}_0(N)\boldsymbol{\theta}^{(i-1)}\right]
$$

where

$$
\widehat{\mathbf{s}}_0(N) = \frac{1}{N}\sum_{t=1}^{N} y(t)\boldsymbol{\varphi}(t)
$$

$$
\widehat{\mathbf{R}}_0(N) = \frac{1}{N}\sum_{t=1}^{N} \boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)
$$

The method of least mean squares (LMS)
Metoda najmniejszych średnich kwadratów

$$J_t(\boldsymbol{\theta}) = \left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}\right]^2$$

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) - \frac{1}{2}\,\mu\nabla_{\boldsymbol{\theta}} J_t[\widehat{\boldsymbol{\theta}}(t-1)]$$

$$\nabla_{\boldsymbol{\theta}} J_t(\boldsymbol{\theta}) = -2y(t)\boldsymbol{\varphi}(t) + 2\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}$$

$$\Downarrow$$

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \mu\boldsymbol{\varphi}(t)\varepsilon(t)$$

$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)$$

least mean squares (LMS) algorithm

<u>Notice</u>

Since

$$\nabla_{\boldsymbol{\theta}} J_t[\widehat{\boldsymbol{\theta}}(t-1)] = -2\boldsymbol{\varphi}(t)\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)\right]$$

for a system with constant parameters $y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 + n(t)$ it holds that

$$\nabla_{\boldsymbol{\theta}} J_t[\widehat{\boldsymbol{\theta}}(t-1)] = -2\boldsymbol{\varphi}(t)\left\{\boldsymbol{\varphi}^{\mathrm{T}}(t)\left[\boldsymbol{\theta}_0 - \widehat{\boldsymbol{\theta}}(t-1)\right] + n(t)\right\}$$

stochastic gradient

Analysis of the scalar LMS algorithm
Analiza skalarnego algorytmu LMS

Consider a scalar time-invariant system governed by

$$y(t) = \varphi(t)\theta_0 + n(t)$$

where θ_0 denotes an unknown (but constant) coefficient and $\varphi(t)$ is a measurable regression variable. Assume that $\{\varphi(t)\}$ and $\{n(t)\}$ are independent sequences of independent random variables with zero mean and variances σ_φ^2 and σ_0^2, respectively. Finally, assume that

$$\mathrm{E}[\varphi^4(t)] = c_\varphi^4 < \infty$$

Perform the mean and mean square error analysis of the LMS algorithm

$$\widehat{\theta}(t) = \widehat{\theta}(t-1) + \mu\varphi(t)\left[y(t) - \varphi(t)\widehat{\theta}(t-1)\right]$$

convergence in the mean / zbieżność w sensie średnim

Denote by

$$\widetilde{\theta}(t) = \theta_0 - \widehat{\theta}(t)$$

the parameter estimation error yielded by the LMS algorithm. It holds that

$$\widetilde{\theta}(t) = \left[1 - \mu\varphi^2(t)\right]\widetilde{\theta}(t-1) - \mu\varphi(t)n(t)$$

Analysis of the scalar LMS algorithm
Analiza skalarnego algorytmu LMS

We will evaluate expectations of both sides of the error equation

$$\mathrm{E}[\widetilde{\theta}(t)] = (1 - \mu\sigma_\varphi^2)\mathrm{E}[\widetilde{\theta}(t-1)]$$

Therefore if

$$|1 - \mu\sigma_\varphi^2| < 1 \quad \Longrightarrow \quad 0 < \mu < \frac{2}{\sigma_\varphi^2}$$

the mean value of the estimation error will asymptotically decay to zero irrespective of the imposed initial conditions, i.e.

$$\mathrm{E}[\widehat{\theta}(t)] \quad \underset{t\to\infty}{\longrightarrow} \quad \theta_0$$

mean square convergence / zbieżność w sensie średniokwadratowym

To study the mean square convergence of the LMS algorithm, we will square and take expectations of both sides of the error equation

$$\widetilde{\theta}(t) = \left[1 - \mu\varphi^2(t)\right]\widetilde{\theta}(t-1) - \mu\varphi(t)n(t)$$

This leads to

$$\sigma_\theta^2(t) = f(\mu)\sigma_\theta^2(t-1) + \mu^2\sigma_\varphi^2\sigma_0^2$$
$$f(\mu) = 1 - 2\mu\sigma_\varphi^2 + \mu^2 c_\varphi^4$$

Analysis of the scalar LMS algorithm
Analiza skalarnego algorytmu LMS

Hence, to guarantee stability of the LMS algorithm in the mean square sense, one should choose the stepsize μ so as to fulfill

$$|f(\mu)| < 1 \quad \Longrightarrow \quad f(\mu) < 1 \quad \Longrightarrow \quad 0 < \mu < \frac{2\sigma_\varphi^2}{c_\varphi^4}$$

The second transition follows from the fact that $f(\mu) = \mathrm{E}[(1 - \mu\varphi^2(t))^2] \geq 0$.

Note that the mean square stability condition is more stringent than the mean stability condition since, according to the Schwartz inequality,

$$c_\varphi^4 = \mathrm{E}[\varphi^4(t)] \geq (\mathrm{E}[\varphi^2(t)])^2 = \sigma_\varphi^4$$

If the distribution of $\varphi(t)$ is Gaussian then

$$c_\varphi^4 = 3\sigma_\varphi^4$$

leading to

$$0 < \mu < \frac{2}{3\sigma_\varphi^2}$$

Here the mean square stability bound is three times smaller than the mean stability bound.

Analysis of the scalar LMS algorithm
Analiza skalarnego algorytmu LMS

If the stability condition is met, the asymptotic solution of the mean square error equation given above can be obtained in the form

$$\sigma_\theta^2(\infty) = \lim_{t\to\infty} \sigma_\theta^2(t) = \frac{1}{1-\mu\,\frac{c_\varphi^4}{2\sigma_\varphi^2}}\,\frac{\mu}{2}\,\sigma_0^2$$

$$\cong \frac{\mu}{2}\,\sigma_0^2$$

General observations

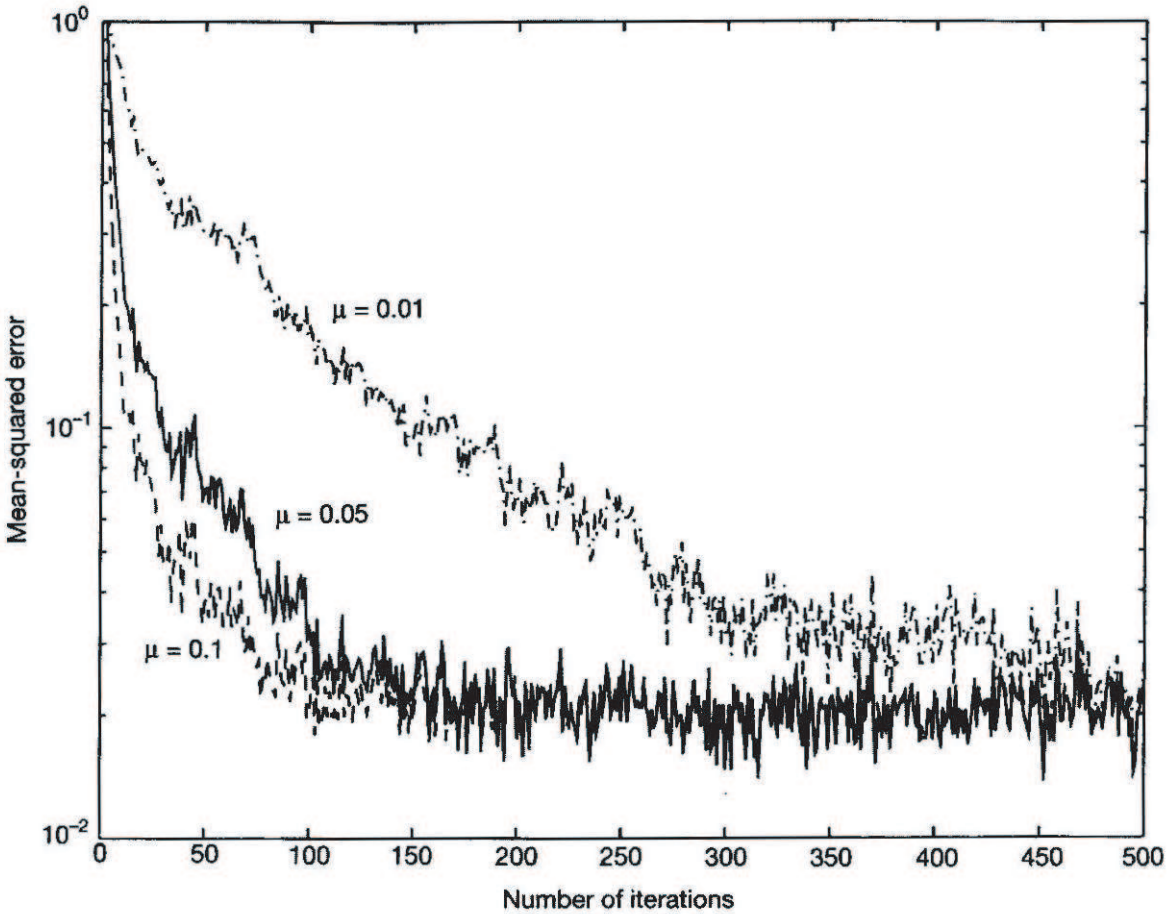
1. The LMS algorithm is stable provided that the stepsize parameter μ is sufficiently small.

2. Under time-invariant conditions the mean square parameter estimation error approaches a constant value proportional to μ.

Analysis of the scalar LMS algorithm
Analiza skalarnego algorytmu LMS

$$y(t) = ay(t-1) + n(t)$$

$$a = -0.99 \qquad \sigma_n^2 = 0.94$$



Experimental learning curves of adaptive predictor for different values of the step-size patameter μ (ensemble averaging was performed over 100 independent trials of the experiment).

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS

Consider a scalar time-invariant system governed by

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}_0 + n(t)$$

where $\boldsymbol{\theta}_0$ denotes an unknown coefficient and $\boldsymbol{\varphi}(t)$ is a measurable regression vector. Assume that $\{\boldsymbol{\varphi}(t)\}$ and $\{n(t)\}$ are independent sequences of independent random variables with zero mean and covariances $\mathbf{R}_0$ and σ_0^2, respectively. Finally, assume that

$$\mathrm{E}[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)] = \mathbf{C}_0 < \infty$$

We will perform error analysis of the LMS algorithm

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \mu\boldsymbol{\varphi}(t)\left[y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)\right]$$

convergence in the mean / zbieżność w sensie średnim

Denote by

$$\widetilde{\boldsymbol{\theta}}(t) = \boldsymbol{\theta}_0 - \widehat{\boldsymbol{\theta}}(t)$$

the parameter estimation error yielded by the LMS algorithm. It holds that

$$\widetilde{\boldsymbol{\theta}}(t) = \left[\mathbf{I} - \mu\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\widetilde{\boldsymbol{\theta}}(t-1) - \mu\boldsymbol{\varphi}(t)n(t)$$

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS

$$\mathrm{E}[\widetilde{\boldsymbol{\theta}}(t)] = [\mathbf{I} - \mu\mathbf{R}_0]\,\mathrm{E}[\widetilde{\boldsymbol{\theta}}(t-1)]$$

To get more insights into the conditions and geometry of convergence, we shall analyze the mean error equation in a different system of coordinates. Denote by $\mathbf{Q}$ a unitary matrix, made up of the eigenvectors of $\mathbf{R}_0$, converting $\mathbf{R}_0$ into a diagonal form:

$$\mathbf{Q}^{\mathrm{T}}\mathbf{Q} = \mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{I}, \qquad \mathbf{Q}^{\mathrm{T}}\mathbf{R}_0\mathbf{Q} = \boldsymbol{\Lambda}_o$$

where

$$\boldsymbol{\Lambda}_o = \mathrm{diag}\{\lambda_1(\mathbf{R}_0), \ldots, \lambda_n(\mathbf{R}_0)\}$$

is a diagonal matrix made up of the eigenvalues of $\mathbf{R}_0$.

Let $\boldsymbol{\gamma}(t) = \mathbf{Q}^{\mathrm{T}}\widetilde{\boldsymbol{\theta}}(t)$. Then one obtains

$$\mathrm{E}[\boldsymbol{\gamma}(t)] = [\mathbf{I} - \mu\boldsymbol{\Lambda}_0]\,\mathrm{E}[\boldsymbol{\gamma}(t-1)]$$

which can be rewritten in the following explicit, decoupled form

$$\mathrm{E}[\gamma_i(t)] = [1 - \mu\lambda_i(\mathbf{R}_0)]^t\,\mathrm{E}[\gamma_i(0)], \qquad i = 1, \ldots, n$$

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS

Since both the stepsize μ and the eigenvalues of the covariance matrix $\mathbf{R}_o$ must be positive, the mean stability/convergence condition can be put in the form

$$|1 - \mu\lambda_{\max}(\mathbf{R}_0)| < 1$$

where $\lambda_{\max}(\mathbf{R}_0)$ denotes the maximum eigenvalue of $\mathbf{R}_0$

$$\lambda_{\max}(\mathbf{R}_0) = \max\{\lambda_1(\mathbf{R}_0), \ldots, \lambda_n(\mathbf{R}_0)\}.$$

This leads to

$$0 < \mu < \frac{2}{\lambda_{\max}(\mathbf{R}_0)}$$

Remark 1

The mean convergence condition stated above may be difficult to verify in practice. Note that

$$\lambda_{\max}(\mathbf{R}_0) \leq \sum_{i=1}^{n} \lambda_i(\mathbf{R}_0) = \mathrm{tr}\{\mathbf{R}_0\}.$$

Hence the bound derived above can be replaced with the more conservative constraint

$$0 < \mu < \frac{2}{\mathrm{tr}\{\mathbf{R}_0\}}$$

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS

This condition is much easier to check since

$$\mathrm{tr}\{\mathbf{R}_0\} = \mathrm{tr}\left\{\mathrm{E}[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)]\right\} = \mathrm{E}[\|\,\boldsymbol{\varphi}(t)\,\|^2]$$

Remark 2

Analysis of the diagonalized error equation reveals another important feature of the LMS algorithm – its sensitivity to the covariance structure of the regression vector $\boldsymbol{\varphi}(t)$. Even though, when the stability conditions are met, all components of $\boldsymbol{\gamma}(t)$ converge exponentially to zero, the actual speed of convergence depends on the eigenvalue spread of the covariance matrix $\mathbf{R}_0$; convergence takes place faster in those directions in the parameter space which are associated with large-eigenvalue eigenvectors of $\mathbf{R}_0$ and slower in directions determined by small-eigenvalue eigenvectors. If the *eigenvalue disparity* index of $\mathbf{R}_0$ is large,

$$\delta(\mathbf{R}_0) = \frac{\lambda_{\max}(\mathbf{R}_0)}{\lambda_{\min}(\mathbf{R}_0)} \gg 1$$

the variation of the convergence speed over different components of γ_i may be significant.

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS

mean square convergence / zbieżność w sensie średniokwadratowym

The mean square stability condition takes the form

$$0 < \mu < \frac{2\lambda_{\min}(\mathbf{R}_0)}{\lambda_{\max}(\mathbf{C}_0)}$$

If the stability condition is met, the asymptotic solution of the mean square error equation can be obtained in the form

$$\mathbf{\Sigma}_\theta(\infty) = \lim_{t\to\infty} \operatorname{cov}\left[\widehat{\boldsymbol{\theta}}(t)\right] \cong \frac{\mu}{2}\,\sigma_0^2\mathbf{I}$$

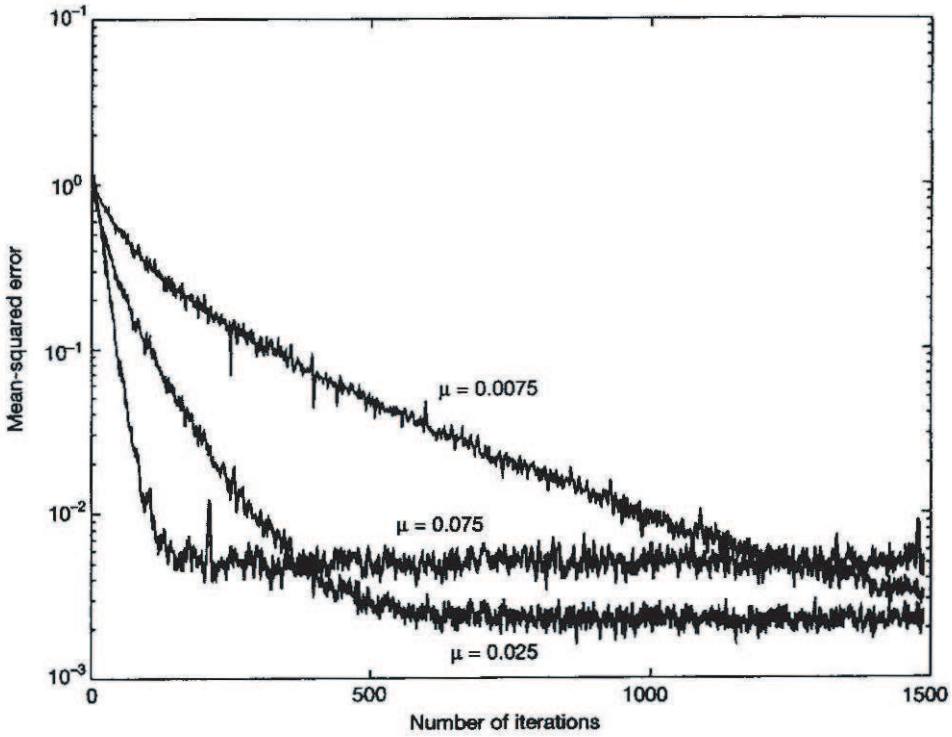
estimation memory of the LMS algorithm / pamięć estymacji algorytmu LMS

For weighted least squares estimators it holds that

$$\lim_{t\to\infty} \operatorname{cov}\left[\widehat{\boldsymbol{\theta}}(t)\right] \cong \frac{\sigma_0^2\mathbf{R}_0^{-1}}{l_\infty}$$

$$\operatorname{tr}\left\{\frac{\mu}{2}\,\sigma_0^2\mathbf{I}\right\} = \operatorname{tr}\left\{\frac{\sigma_0^2\mathbf{R}_0^{-1}}{l_\infty}\right\} \quad \Rightarrow \quad l_\infty = \frac{2\operatorname{tr}\{\mathbf{R}_0^{-1}\}}{n\mu}$$

Analysis of the vector LMS algorithm
Analiza wektorowego algorytmu LMS



Experimental learning curves of the LMS algorithm (11-tap adaptive equalizer) for different values of the step-size patameter μ
(ensemble averaging was performed over 200 independent trials of the experiment).

Tracking properties of LMS estimators
Własności śledzące estymatorów LMS

$$y(t) = \boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + n(t)$$

Assumption 1

$\{n(t)\}$ is a sequence of zero-mean, independent and identically distributed random variables with variance σ_0^2.

Assumption 2

$\{\boldsymbol{\varphi}(t)\}$ is a sequence of independent and identically distributed random vectors, independent of $\{\boldsymbol{\theta}(t)\}$, with positive definite correlation matrix $\mathrm{E}\left[\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right] = \mathbf{R}_0 > \mathbf{O}$. The variables $n(t)$ and $\boldsymbol{\varphi}(t)$ are mutually independent $\forall t$.

Since for the LMS estimator it holds that

$$\widehat{\boldsymbol{\theta}}(t) = \left[\mathbf{I} - \mu\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\right]\widehat{\boldsymbol{\theta}}(t-1) + \mu\boldsymbol{\varphi}(t)\boldsymbol{\varphi}^{\mathrm{T}}(t)\boldsymbol{\theta}(t) + \mu\boldsymbol{\varphi}(t)n(t)$$

leading to

$$\bar{\boldsymbol{\theta}}(t) = \mathrm{E}\left[\widehat{\boldsymbol{\theta}}(t)\right] = (\mathbf{I} - \mu\mathbf{R}_0)\,\bar{\boldsymbol{\theta}}(t-1) + \mu\mathbf{R}_0\boldsymbol{\theta}(t)$$

Tracking properties of LMS estimators
Własności śledzące estymatorów LMS

Note that in the steady state the recursive relationship between $\bar{\boldsymbol{\theta}}(t)$ and $\boldsymbol{\theta}(t)$ can be written down in the explicit form

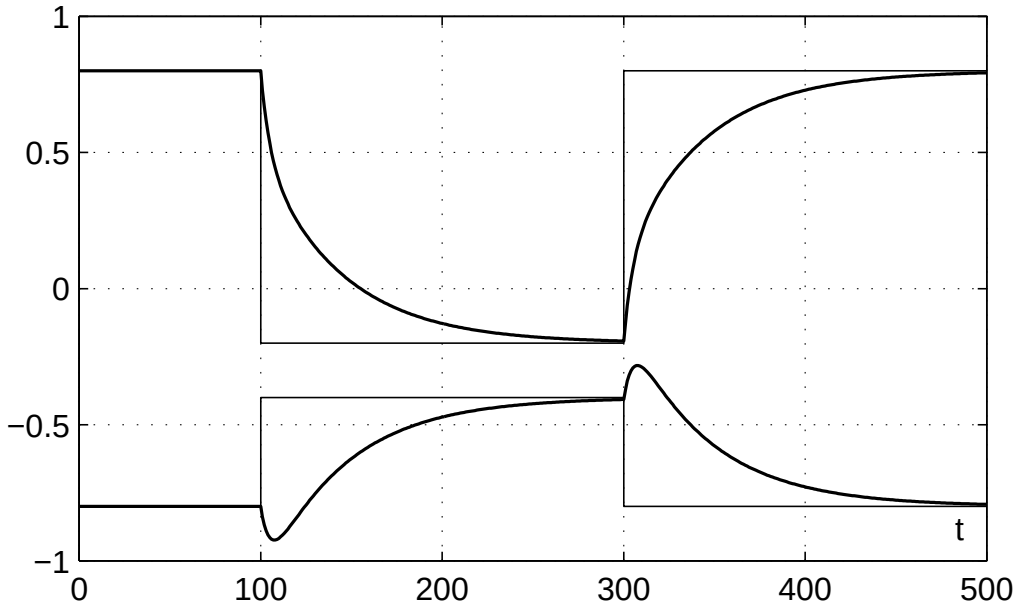
$$\bar{\boldsymbol{\theta}}(t) \cong \sum_{i=0}^{\infty} \mathbf{H}(i)\boldsymbol{\theta}(t-i)$$

$$\mathbf{H}(i) = \mu\left(\mathbf{I} - \mu\mathbf{R}_0\right)^i \mathbf{R}_0$$

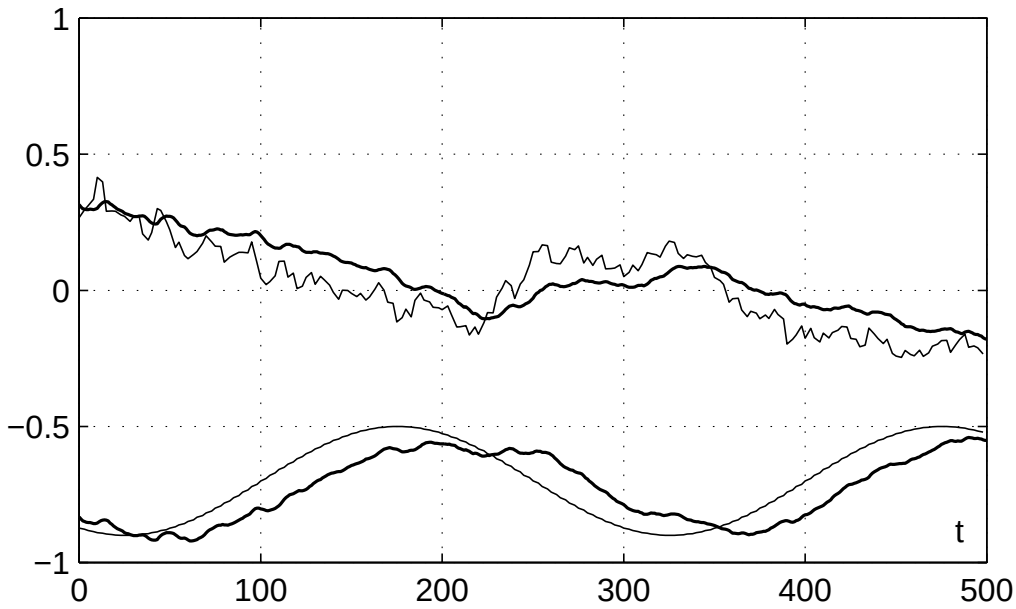
The sequence $\{\mathbf{H}(i)\}$ is called the impulse response associated with the LMS estimator (*odpowiedź impulsowa skojarzona z estymatorem LMS*).

According to the obtained formula, after a sufficiently long time, $\{\bar{\boldsymbol{\theta}}(t)\}$ can be approximately viewed as an output of a linear causal time-invariant filter with impulse response $\{\mathbf{H}(i)\}$ excited by the process $\{\boldsymbol{\theta}(t)\}$.

Unlike the WLS case, where a sequence of scalar coefficients $\{h(i)\}$ was sufficient to characterize evolution in the mean of all components of $\widehat{\boldsymbol{\theta}}(t)$, the impulse response $\{\mathbf{H}(i)\}$ is made up of matrices depending both on the stepsize μ (which is a user-dependent quantity) and on $\mathbf{R}_0$ (which depends on the input data).

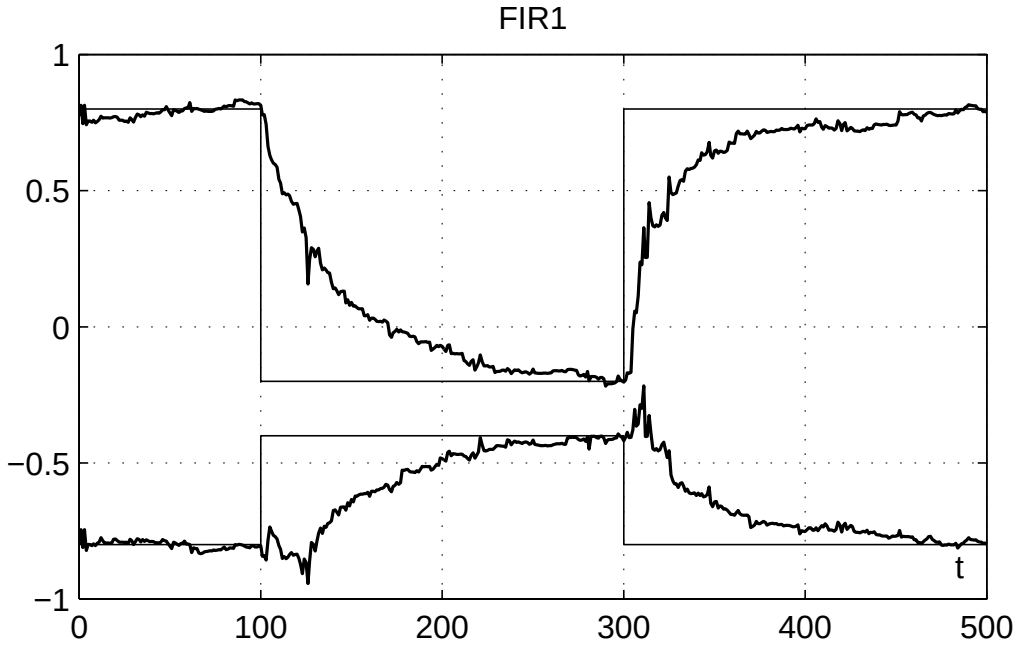


(a)

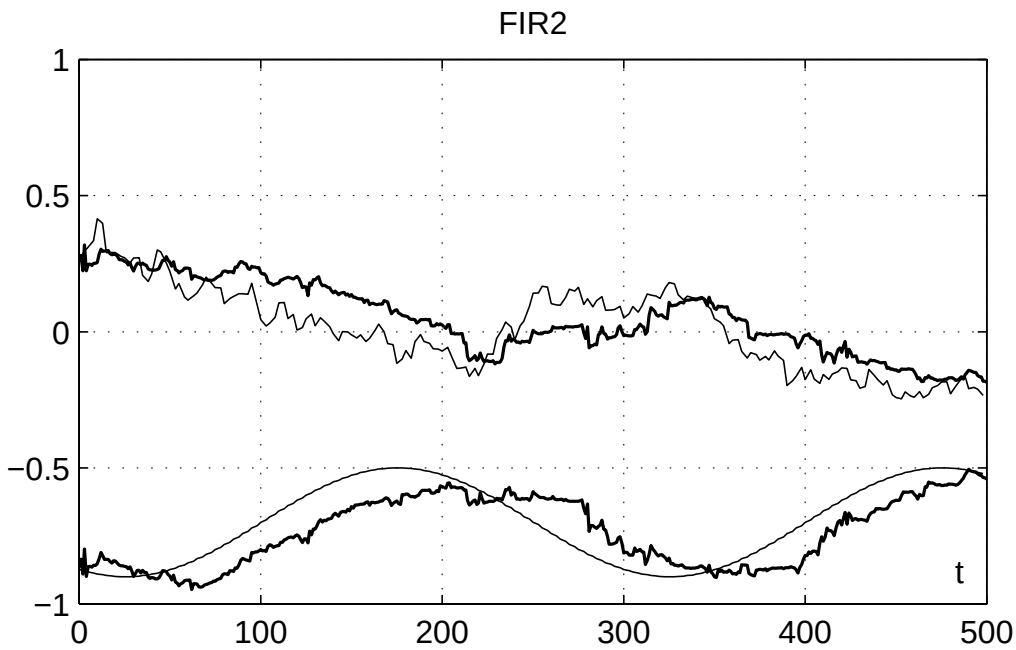


(b)

Output of a linear filter associated with the least mean squares estimator for a system with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\mu = 0.04$).

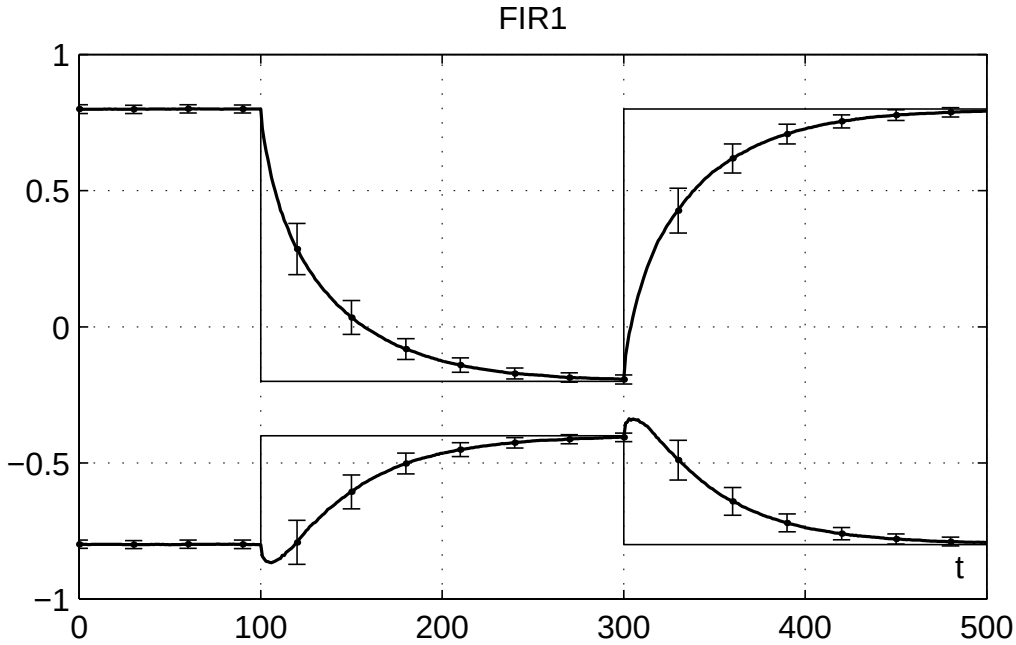


(a)

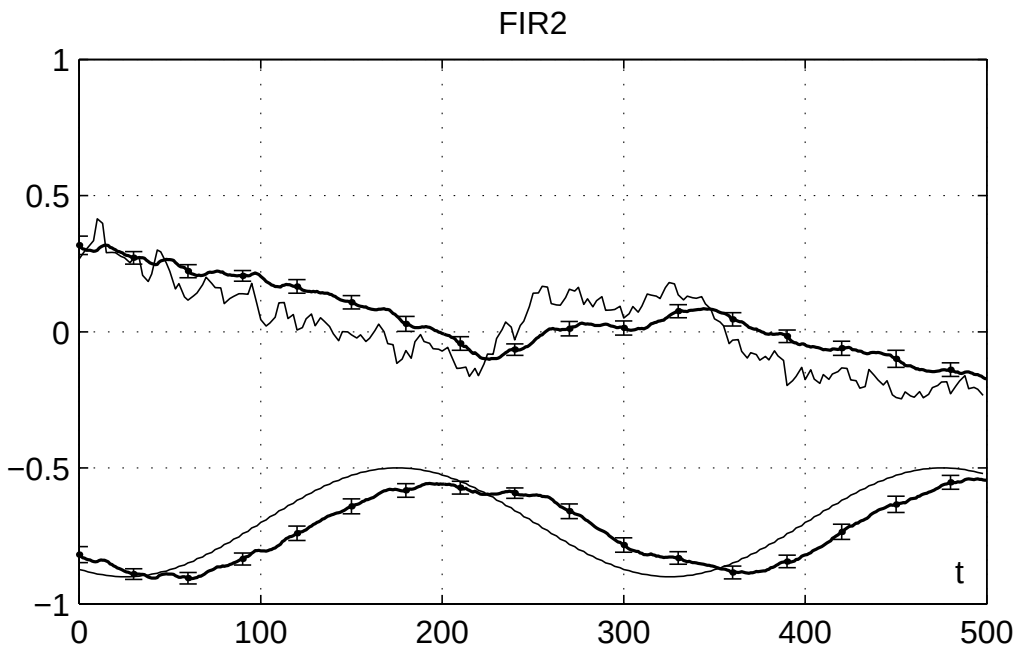


(b)

Parameter estimates yielded by the least mean squares algorithm for a single realization of an FIR system with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\mu = 0.0.4$).

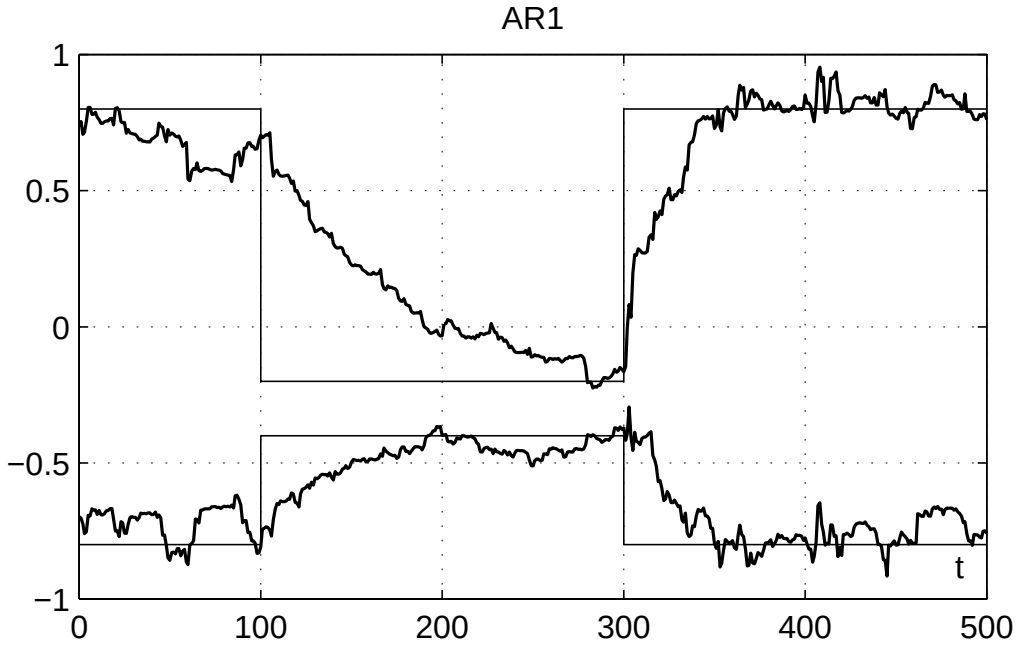


(a)

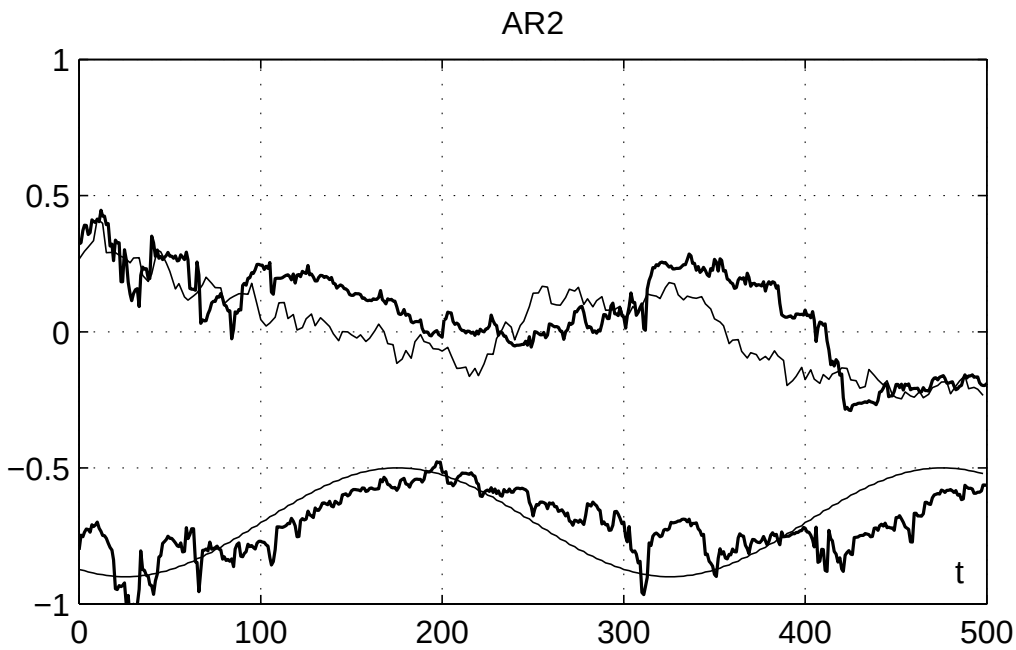


(b)

Average trajectories of parameter estimates yielded by the least mean squares algorithm for an FIR system with jump parameter changes (a) and continuous parameter changes (b); vertical bars show standard deviation of the estimates.

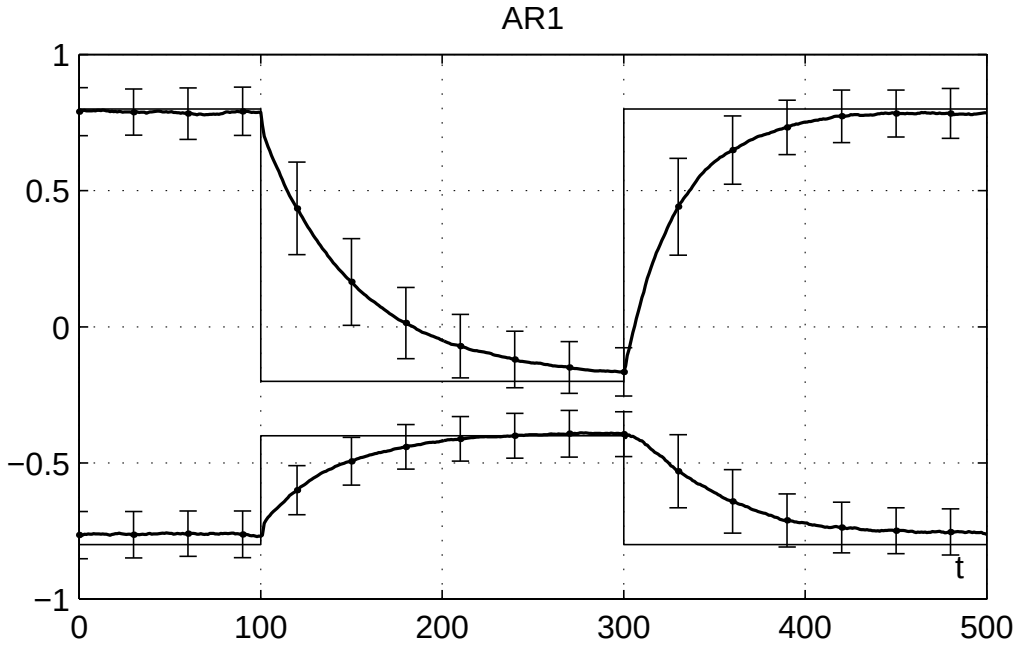


(a)

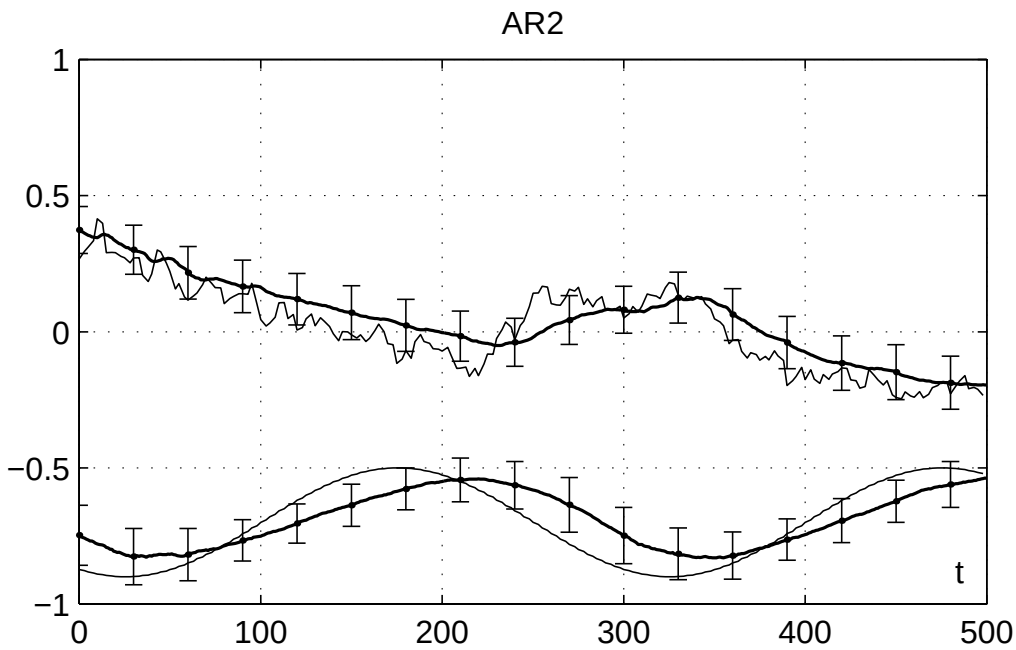


(b)

Parameter estimates yielded by the least mean squares algorithm for a single realization of an AR process with jump parameter changes (a) and continuous parameter changes (b); the equivalent memory of the estimation algorithm was set to $l_\infty = 50$ ($\mu = 0.04$).



(a)



(b)

Average trajectories of parameter estimates yielded by the least mean squares algorithm for an AR process with jump parameter changes (a) and continuous parameter changes (b); vertical bars show standard deviation of the estimates.

LMS algorithm – summary
Algorytm LMS – podsumowanie

advantages / zalety

- Very low computational complexity - the number of multiply/add operations $(2n+1)$ is proportional to the number of estimated parameters (n).
- Good parameter tracking properties

disadvantages / wady

- Slow initial convergence (compared to weighted least squares algorithms).
- Dependence of the estimation memory on characteristics of the identified system.
- Dependence of parameter tracking properties on the eigenstructure of the matrix $\mathbf{R}_0$

Normalized LMS algorithms
Unormowane algorytmy LMS

The memory of an LMS algorithm depends on the magnitude of the regression vector, i.e. it is not scale invariant. Actually, note that the LMS algorithm with stepsize μ, running on a set of measurements scaled by a factor η,

$$y'(t) = \eta y(t), \qquad \boldsymbol{\varphi}'(t) = \eta \boldsymbol{\varphi}(t)$$

will yield identical results as an LMS algorithm with a stepsize μ/η^2 and fed with the original measurements. Quite obviously, this is an undesirable property. Since scaling does not affect the information content of the data, one has the right to expect identification results to be identical for the same stepsize μ.

The scale-invariant version of the LMS algorithm can be obtained by replacing the stepsize μ with a normalized stepsize

$$\frac{\mu}{\| \boldsymbol{\varphi}(t) \|^2}$$

To avoid numerical problems, arising when the squared norm $\| \boldsymbol{\varphi}(t) \|^2$ becomes very small or zero, the denominator in the expression presented above is usually replaced with $\epsilon + \| \boldsymbol{\varphi}(t) \|^2$ where ϵ is a small positive constant – a 'safety valve' preventing the normalized stepsize from getting too large.

The resulting algorithm,

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \frac{\mu}{\epsilon + \| \boldsymbol{\varphi}(t) \|^2} \boldsymbol{\varphi}(t)\varepsilon(t)$$
$$\varepsilon(t) = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(t-1)$$

is usually called the normalized LMS (NLMS).

Another useful variant of a scale-invariant LMS algorithm, employing the concept of exponential forgetting, is

$$\widehat{\boldsymbol{\theta}}(t) = \widehat{\boldsymbol{\theta}}(t-1) + \frac{\mu}{r(t)} \boldsymbol{\varphi}(t)\varepsilon(t)$$
$$r(t) = \lambda r(t-1) + \| \boldsymbol{\varphi}(t) \|^2$$

where $0 < \lambda < 1$, is called the trace LMS algorithm (TLMS). The name stems from the fact that $r(t)$ is identical with the trace of the exponentially weighted regression matrix $\mathbf{R}(t)$ in the exponentially weighted least squares approach:

$$r(t) = \sum_{i=0}^{t-1} \lambda^i \| \boldsymbol{\varphi}(t-i) \|^2$$
$$= \mathrm{tr} \left\{ \sum_{i=0}^{t-1} \lambda^i \boldsymbol{\varphi}(t-i)\boldsymbol{\varphi}^{\mathrm{T}}(t-i) \right\} = \mathrm{tr}\{\mathbf{R}(t)\}$$

The mean convergence conditions for NLMS and TLMS algorithms take the form $0 < \mu < 2$ and $0 < \mu < 2/(1-\lambda)$, respectively.

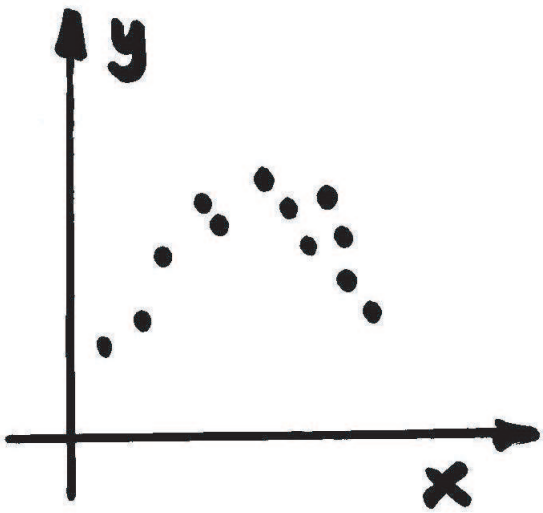
Identification of the model structure
Identyfikacja strukturalna

Example

Based on the set of 12 input/output measurements $\Omega(12) = \{x(1), y(1), \ldots, x(12), y(12)\}$ of a static nonlinear system governed by

$$y(t) = f[x(t)] + n(t), \quad t = 1, \ldots, 12$$

where $\{n(t)\}$ denotes white noise, determine the best polynomial model of the nonlinear characteristic $f[\cdot]$



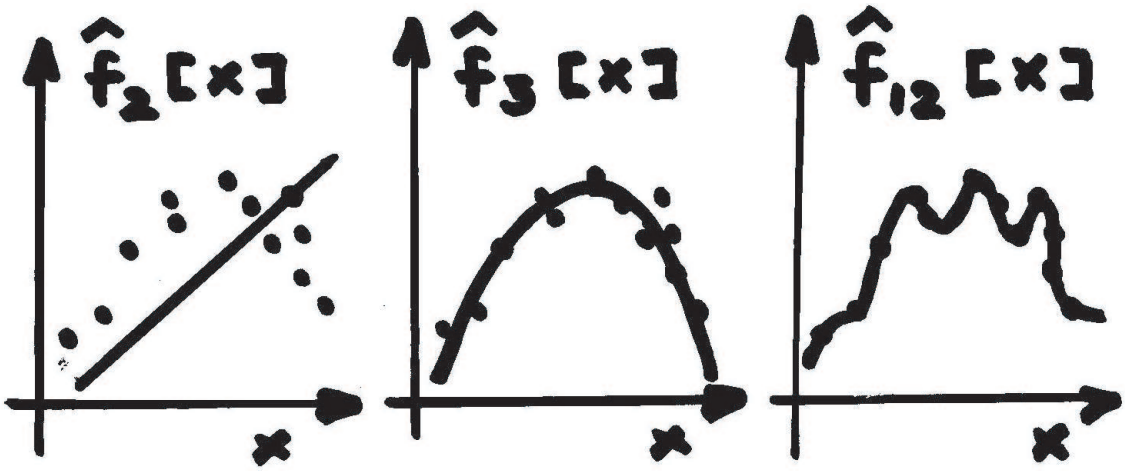
$$f_k[x] = \sum_{i=1}^{k} a_i^k x^{i-1}$$

$k = ?$

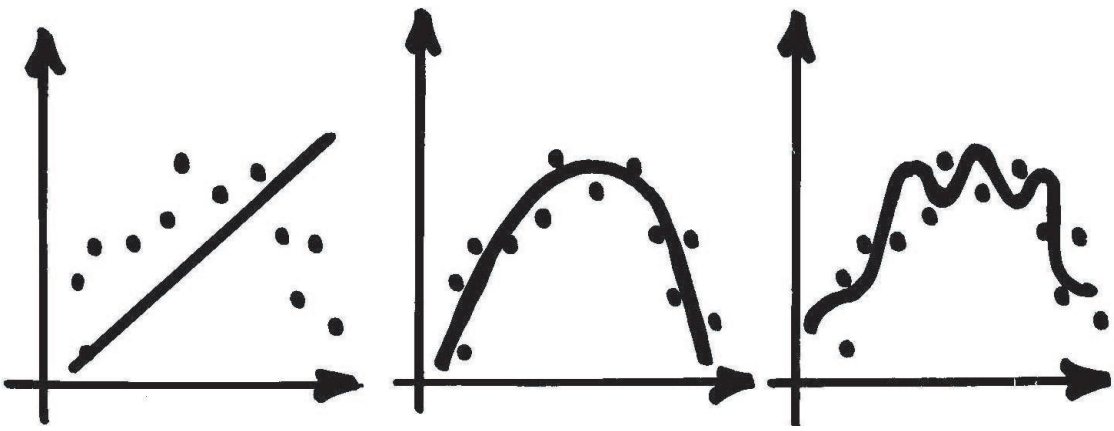
$$\widehat{f}_k[x] = \sum_{i=1}^{k} \widehat{a}_i^k(12) x^{i-1}$$

where

$$\{\widehat{a}_i^k(12)\} = \arg\min_{\{a_i^k\}} \sum_{t=1}^{12} \left[y(t) - \sum_{i=1}^{k} a_i^k x^{i-1}(t) \right]^2$$



Which model would you choose?



Hint: check your model using another (independent) set of measurements

Preliminary considerations
Rozważania wstępne

Consider two "nested" system descriptions

$$y(t) = \boldsymbol{\theta}_1^{\mathrm{T}} \boldsymbol{\varphi}_1(t) + n_1(t)$$
$$y(t) = \boldsymbol{\theta}_2^{\mathrm{T}} \boldsymbol{\varphi}_2(t) + n_2(t)$$

where $\boldsymbol{\varphi}_1(t)$ is a sub-vector of $\boldsymbol{\varphi}_2(t)$, i.e.

$$\boldsymbol{\varphi}_1(t) \prec \boldsymbol{\varphi}_2(t), \quad \forall t$$

Denote by $\widehat{\boldsymbol{\theta}}_1(N)$ and $\widehat{\boldsymbol{\theta}}_2(N)$ the least squares estimates of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, respectively

$$\widehat{\boldsymbol{\theta}}_i(N): \quad \sum_{t=1}^{N} \left[y(t) - \boldsymbol{\theta}_i^{\mathrm{T}} \boldsymbol{\varphi}_i(t)\right]^2 \longrightarrow \min, \quad i = 1, 2$$

Then it holds that

$$\sum_{t=1}^{N} \left[y(t) - \widehat{\boldsymbol{\theta}}_2^{\mathrm{T}}(N) \boldsymbol{\varphi}_2(t)\right]^2 \leq \sum_{t=1}^{N} \left[y(t) - \widehat{\boldsymbol{\theta}}_1^{\mathrm{T}}(N) \boldsymbol{\varphi}_1(t)\right]^2$$

Conclusion: minimization of the sum of squared residual errors is not a good way of choosing the model order – it will always select the model of the largest order

The method of hypothesis testing
Metoda testowania (weryfikacji) hipotez

Consider two "nested" system descriptions

$$y(t) = \boldsymbol{\theta}_1^{\mathrm{T}} \boldsymbol{\varphi}_1(t) + n_1(t)$$
$$y(t) = \boldsymbol{\theta}_2^{\mathrm{T}} \boldsymbol{\varphi}_2(t) + n_2(t)$$

where

$$n_1(t) \sim \mathcal{N}(0, \sigma_1^2), \quad n_2(t) \sim \mathcal{N}(0, \sigma_2^2)$$
$$\boldsymbol{\varphi}_1(t) \prec \boldsymbol{\varphi}_2(t), \quad \boldsymbol{\theta}_1 \prec \boldsymbol{\theta}_2$$

Denote by $\boldsymbol{\delta}$, $\dim[\boldsymbol{\delta}] = \dim[\boldsymbol{\theta}_2] - \dim[\boldsymbol{\theta}_1] = n_2 - n_1$, the vector made up of superfluous coefficients that are present in $\boldsymbol{\theta}_2$, but absent from $\boldsymbol{\theta}_1$. We will consider two, mutually exclusive, hypotheses:

null hypothesis / hipoteza zerowa (podstawowa)

$$\mathcal{H}_0 : \quad \boldsymbol{\delta} = \mathbf{0}$$

alternative hypothesis / hipoteza alternatywna (uzupełniająca)

$$\mathcal{H}_1 : \quad \boldsymbol{\delta} \neq \mathbf{0}$$

Statistical test
Test statystyczny

Our decision rule will have the following form

$$\eta[\Omega(N)] \begin{cases} \leq \eta_0 & \text{accept } \mathcal{H}_0 \\ > \eta_0 & \text{reject } \mathcal{H}_0 \end{cases}$$

where $\eta[\Omega(N)]$ is called test statistic (*statystyka decyzyjna*) and η_0 is called decision threshold (*próg decyzyjny*).

Assuming that the null hypothesis is true, what is the probability of observing a value for the test statistic that is not larger than the decision threshold ?

Zakładając słuszność hipotezy zerowej, jakie jest prawdopodobieństwo tego, że statystyka decyzyjna przyjmie wartość nie większą od progu decyzyjnego ?

Let

$$\boldsymbol{\phi}_i = \begin{bmatrix} \boldsymbol{\theta}_i \\ \sigma_i^2 \end{bmatrix}, \quad \widehat{\boldsymbol{\phi}}_i(N) = \begin{bmatrix} \widehat{\boldsymbol{\theta}}_i(N) \\ \widehat{\sigma}_i^2(N) \end{bmatrix}, \quad i = 1, 2$$

where $\widehat{\boldsymbol{\phi}}_1(N)$ and $\widehat{\boldsymbol{\phi}}_2(N)$ are maximum likelihood estimators

$$\widehat{\boldsymbol{\phi}}_i(N) = \arg\max_{\boldsymbol{\phi}_i} p\left(\Omega(N)|\boldsymbol{\phi}_i\right)$$

Likehood ratio test
Test ilorazu wiarygodności

We will adopt the following test statistic

$$\eta[\Omega(N)] = 2\log\frac{p\left(\Omega(N)|\widehat{\boldsymbol{\phi}}_2(N)\right)}{p\left(\Omega(N)|\widehat{\boldsymbol{\phi}}_1(N)\right)}$$
$$= 2L\left[\Omega(N);\widehat{\boldsymbol{\phi}}_2(N)\right] - 2L\left[\Omega(N);\widehat{\boldsymbol{\phi}}_1(N)\right]$$

It can be shown that

$$p\left(\eta[\Omega(N)] \mid \mathcal{H}_0\right) \cong \chi^2(n_2 - n_1)$$

where $\chi^2(n)$ denotes the chi-squared distribution with n degrees of freedom (*rozkład chi-kwadrat o n stopniach swobody*).

<u>Fact</u>

Consider a sequence of independent random variables $X_i \sim \mathcal{N}(0,1), i = 1, \ldots n$. The random variable

$$Y = \sum_{i=1}^{n} X_i^2$$

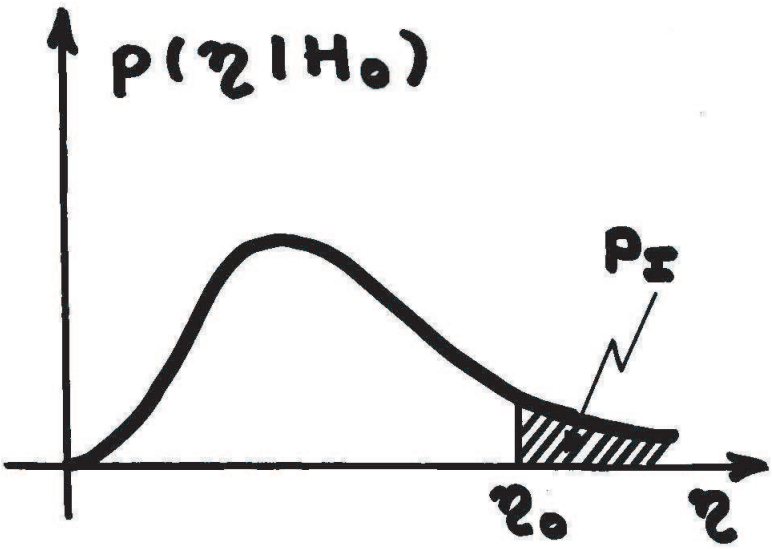
is distributed according to the chi-squared distribution with n degrees of freedom: $Y \sim \chi^2(n)$.

Likehood ratio test
Test ilorazu wiarygodności

We will select decision threshold η_0 so as to make the Type I error (*prawdopodobieństwo błędu pierwszego rodzaju*) appropriately small, namely

$$\begin{aligned} p_{\mathrm{I}} &= P(\text{reject } \mathcal{H}_0 \mid \mathcal{H}_0 \text{ is valid}) \\ &= P\left(\eta[\Omega(N)] > \eta_0 \mid \mathcal{H}_0\right) = \epsilon \end{aligned}$$

where ϵ denotes significance level of a test (*poziom istotności testu*) – the upper bound of the probability of rejecting the null hypothesis over all cases covered by the null hypothesis.



<u>Example</u>

$$n_2 - n_1 = 1, \epsilon = 0.01 \quad \Longrightarrow \quad \eta_0 = 6.6$$

Likehood ratio test for ARX models

Test ilorazu wiarygodności dla modeli ARX

$$L\left[\Omega(N);\boldsymbol{\phi}_i\right] = -\frac{N}{2}\,\log(2\pi\sigma_i^2) - \frac{1}{2\sigma_i^2}\sum_{t=1}^{N}\left[y(t)-\boldsymbol{\varphi}_i^{\mathrm{T}}(t)\boldsymbol{\theta}_i\right]^2$$

$$L\left[\Omega(N);\widehat{\boldsymbol{\phi}}_i(N)\right] = -\frac{N}{2}\,\log[2\pi\widehat{\sigma}_i^2(N)] - \frac{N}{2}$$

where

$$\widehat{\sigma}_i^2(N) = \frac{1}{N}\sum_{t=1}^{N}\left[y(t)-\boldsymbol{\varphi}_i^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_i(N)\right]^2$$

$$\widehat{\boldsymbol{\theta}}_i(N) = \left[\sum_{t=1}^{N}\boldsymbol{\varphi}_i(t)\boldsymbol{\varphi}_i^{\mathrm{T}}(t)\right]^{-1}\left[\sum_{t=1}^{N}\boldsymbol{\varphi}_i(t)y(t)\right]$$

This leads to

$$\eta[\Omega(N)] = N\log\frac{\widehat{\sigma}_1^2(N)}{\widehat{\sigma}_2^2(N)}$$

$$\text{if}\ \ \log\frac{\widehat{\sigma}_1^2(N)}{\widehat{\sigma}_2^2(N)} \le \frac{\eta_0}{N}\ \text{choose}\ \mathcal{M}_1,\ \text{otherwise choose}\ \mathcal{M}_2$$

Order selection procedure
Procedura wyboru rzędu

Consider k "nested" system descriptions

$$y(t) = \boldsymbol{\theta}_1^{\mathrm{T}} \boldsymbol{\varphi}_1(t) + n_1(t)$$
$$\vdots$$
$$y(t) = \boldsymbol{\theta}_k^{\mathrm{T}} \boldsymbol{\varphi}_k(t) + n_k(t)$$

where

$$\boldsymbol{\varphi}_1(t) \prec \boldsymbol{\varphi}_2(t) \prec \ldots \prec \boldsymbol{\varphi}_k(t)$$
$$\boldsymbol{\theta}_1 \prec \boldsymbol{\theta}_2 \prec \ldots \prec \boldsymbol{\theta}_k$$

1. Based on the available measurements $\Omega(N)$, evaluate $\widehat{\boldsymbol{\theta}}_i(N), \widehat{\sigma}_i^2(N), i = 1, \ldots, k$.

2. Compare the obtained models $\mathcal{M}_i, i = 1, \ldots, k$, pairwise and choose the most adequate one, i.e., the model that passes the log likelihood test against all its competitors.

<u>Problem</u>

Since the likelihood ratio test is not transitive, the most adequate model may not exist.

The method of cross-validation
Metoda walidacji skrośnej

Postulate

The quality of models should be evaluated based on a data set that is different from the one used for the purpose of parameter estimation.

Denote by $\widetilde{\Omega}(N)$ the evaluation data set – obtained under the same experimental conditions as $\Omega(N)$, but independent of $\Omega(N)$.

Consider k models $\mathcal{M}_i, i = 1, \ldots, k$, and the corresponding parameter estimates

$$\widehat{\boldsymbol{\phi}}_i[\Omega(N)] = \begin{bmatrix} \widehat{\boldsymbol{\theta}}_i[\Omega(N)] \\ \widehat{\sigma}_i^2[\Omega(N)] \end{bmatrix}, \quad i = 1, \ldots, k$$

maximum likelihood approach:

$$\mathrm{E}_{\widetilde{\Omega}(N)} \left\{ L \left[\widetilde{\Omega}(N); \widehat{\boldsymbol{\phi}}_i[\Omega(N)] \right] \right\} \quad \xrightarrow{i} \quad \max$$

least squares approach:

$$\mathrm{E}_{\widetilde{\Omega}(N)} \left\{ \sum_{t=1}^{N} \left[\widetilde{y}(t) - \widetilde{\boldsymbol{\varphi}}_i^{\mathrm{T}}(t) \widehat{\boldsymbol{\theta}}_i[\Omega(N)] \right]^2 \right\} \quad \xrightarrow{i} \quad \max$$

The method of cross-validation
Metoda walidacji skrośnej

If only one set of experimental data $\Omega(N) = \{u(1), y(1), \ldots, u(N), y(N)\}$ is available, one can "split" it into:

$\Omega_1(N_1) = \{u(1), y(1), \ldots, u(N_1), y(N_1)\}$ – the set used exclusively for the purpose of parameter estimation

$\Omega_2(N_2) = \{u(N_1+1), y(N_1+1), \ldots, u(N), y(N)\}$ – the set used exclusively for the purpose of model evaluation

$$\Omega_1(N_1) \cup \Omega_2(N_2) = \Omega(N)$$
$$N_1 + N_2 = N$$

The corresponding model selection criterion has the form

$$L\left[\Omega_2(N_2); \widehat{\boldsymbol{\phi}}_i[\Omega_1(N_1)]\right] \quad \xrightarrow{i} \quad \max$$

or

$$\sum_{t=N_1+1}^{N} \left[y(t) - \boldsymbol{\varphi}_i^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_i[\Omega_1(N_1)]\right]^2 \quad \xrightarrow{i} \quad \max$$

<u>Drawback</u>

Creating the evaluation data set, one reduces the number of data that can be used for parameter estimation purposes.

Akaike's criteria
Kryteria Akaikego

1) Final prediction error (FPE) criterion
   Kryterium końcowego błędu predykcji

Consider a family of ARX models subject to a stationary excitation and denote $\mathrm{E}[\boldsymbol{\varphi}_i(t)\boldsymbol{\varphi}_i^{\mathrm{T}}(t)] = \mathbf{R}_i, i = 1, \ldots, k$. Let $\widehat{\boldsymbol{\theta}}_i(N), i = 1, \ldots, k$, be the least squares estimates. Finally, for the time being, suppose that the model order is not underestimated, i.e.,

$$\boldsymbol{\theta}_0 \prec \boldsymbol{\theta}_i, \quad i = 1, \ldots, k$$

We will use the following measure of fit

$$\widetilde{\sigma}_i^2(N) = \frac{1}{N}\,\mathrm{E}_{\widetilde{\Omega}(N)}\left\{\sum_{t=1}^{N}\left[\widetilde{y}(t) - \widetilde{\boldsymbol{\varphi}}_i^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_i(N)\right]^2\right\}$$

Note that

$$\begin{aligned}\widetilde{\sigma}_i^2(N) &= \frac{1}{N}\sum_{t=1}^{N}\mathrm{E}_{\widetilde{\Omega}(N)}\left\{\left[(\widehat{\boldsymbol{\theta}}_i(N) - \boldsymbol{\theta}_i^0)^{\mathrm{T}}\widetilde{\boldsymbol{\varphi}}_i(t)\right]^2\right\} \\ &\quad + \frac{1}{N}\sum_{t=1}^{N}\mathrm{E}_{\widetilde{\Omega}(N)}\left[\widetilde{n}^2(t)\right] \\ &= \frac{1}{N}\sum_{t=1}^{N}\left\{\mathrm{tr}\left[(\widehat{\boldsymbol{\theta}}_i(N) - \boldsymbol{\theta}_i^0)(\widehat{\boldsymbol{\theta}}_i(N) - \boldsymbol{\theta}_i^0)^{\mathrm{T}}\mathbf{R}_i\right] + \sigma_0^2\right\}\end{aligned}$$

FPE criterion
Kryterium FPE

Since

$$\mathrm{E}_{\Omega(N)}\left[(\widehat{\boldsymbol{\theta}}_i(N)-\boldsymbol{\theta}_i^0)(\widehat{\boldsymbol{\theta}}_i(N)-\boldsymbol{\theta}_i^0)^{\mathrm{T}}\right] \cong \frac{\sigma_0^2\mathbf{R}_i^{-1}}{N}$$

one obtains

$$\mathrm{E}_{\Omega(N)}\left[\widetilde{\sigma}_i^2(N)\right] \cong \left[1+\frac{n_i}{N}\right]\sigma_0^2$$

where $n_i = \dim[\boldsymbol{\theta}_i]$.

Since it holds that

$$\mathrm{E}_{\Omega(N)}\left[\widehat{\sigma}_i^2(N)\right] \cong \left[1-\frac{n_i}{N}\right]\sigma_0^2$$

one can estimate the quantity $\widetilde{\sigma}_i^2(N)$ using the formula

$$\widehat{\widetilde{\sigma}_i^2}(N) = \frac{1+\frac{n_i}{N}}{1-\frac{n_i}{N}}\,\widehat{\sigma}_i^2(N) = \mathrm{FPE}(i)$$

This leads to the following selection rule

$$\mathrm{FPE}(i) \quad \xrightarrow{i} \quad \min$$

What happens when the model order is underestimated?

Akaike's criteria
Kryteria Akaikego

1) Akaike's information criterion (AIC)
   Kryterium iformacyjne Akaikego

Let $\widehat{\boldsymbol{\phi}}_i(N), i = 1, \ldots, k$, be the maximum likelihood estimates. Akaike showed that the quantity

$$\mathrm{E}_{\widetilde{\Omega}(N)} \left\{ L \left[\widetilde{\Omega}(N); \widehat{\boldsymbol{\phi}}_i[\Omega(N)] \right] \right\}$$

can be estimated using the following formula

$$L \left[\Omega(N); \widehat{\boldsymbol{\phi}}_i(\Omega(N)) \right] - \dim[\boldsymbol{\phi}_i]$$

For ARX systems (under Gaussian assumptions), one obtains

$$L \left[\Omega(N); \widehat{\boldsymbol{\phi}}_i(N) \right] = \mathrm{const} - \frac{N}{2} \log \widehat{\sigma}_i^2(N)$$

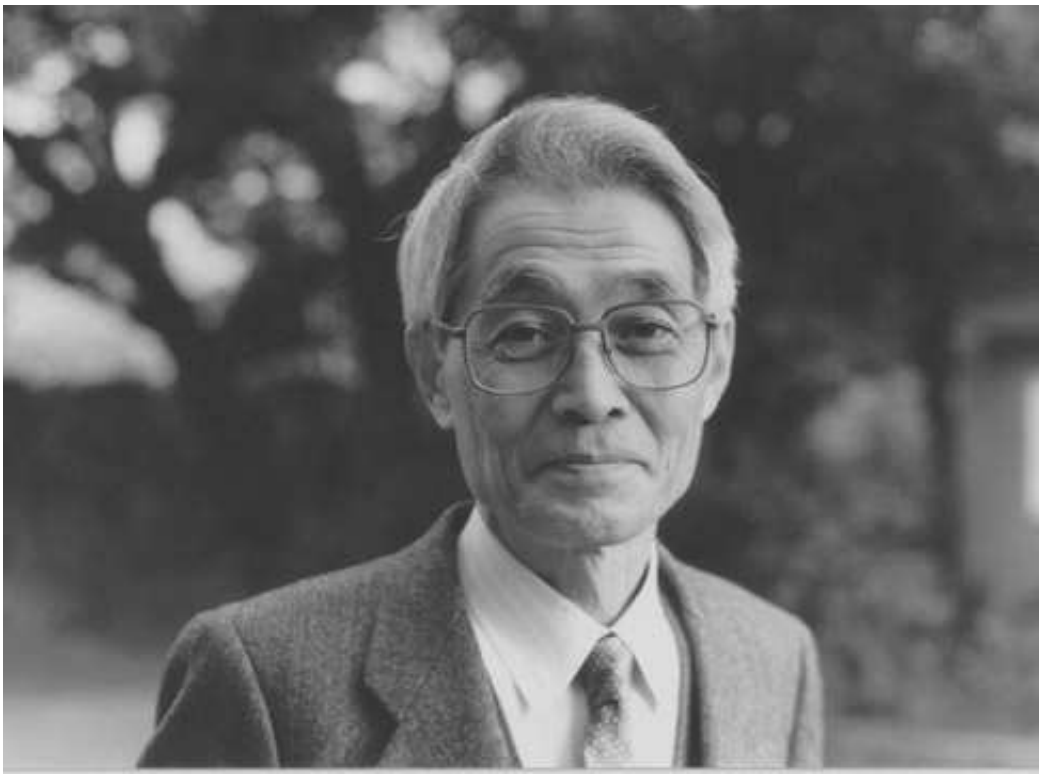
which leads to the following selection rule

$$\mathrm{AIC}(i) = N \log \widehat{\sigma}_i^2(N) + 2n_i \quad \xrightarrow{i} \quad \min$$

Akaike's criteria
Kryteria Akaikego

For ARX systems the FPE and AIC criteria are asymptotically equivalent because, for $n_i \ll N$, it holds that

$$\log \mathrm{FPE}(n_i) = \log \widehat{\sigma}_i^2(N) + \log\left[1 + \frac{n_i}{N}\right] - \log\left[1 - \frac{n_i}{N}\right]$$
$$\cong \log \widehat{\sigma}_i^2(N) + \frac{2n_i}{N} = \frac{1}{N}\,\mathrm{AIC}(n_i)$$



Hirotugu Akaike (1932)

H. Akaike, "A new look at the statistical model identification", IEEE Transactions on Automatic Control, 1974

15849 citations according to Google Scholar

Rissanen's criterion
Kryterium Rissanena

When can one claim that ske/he "understands" behavior of the identified system?

- When one can accurately predict future behavior of the system
- When one can efficiently encode the experimental data

Example

Suppose that one would like to encode the sequence

$$\{a_i\} = \{1, 4, 10, 22, 46, \ldots\}$$

Encoding can be made very efficiently after noting that

$$a_{i+1} = 2a_i + 2, \quad i = 1, 2, \ldots$$

In coding we want to transmit or store sequences of elements of a finite set $A = \{a_1, \ldots, a_m\}$ in terms of binary sequences 0 and 1. The set A is called the *alphabet*, and its elements are called *symbols*. The sequences are called *messages*. A code is a one-to-one function taking each symbol in the alphabet into a finite binary string called the *codeword*.

Rissanen's criterion
Kryterium Rissanena

If the applied code has the so-called prefix property (no codeword is a prefix of another) the codewords in the encoded sequence can be separated and recognized without a comma. The code length is defined as the number of bits of the resulting binary string.

predictive least squares (PLS) criterion

Rissanen has shown that, for ARX models, the length of the optimally assigned code is proportional to the sum of squares of the corresponding one-step-ahead prediction errors. This leads to the following selection rule

$$\mathrm{PLS}(i) = \sum_{t=n_i+1}^{N} \left[y(t) - \widehat{y}_i(t|t-1)\right]^2 \quad \xrightarrow{i} \quad \min$$

where

$$\widehat{y}_i(t|t-1) = \boldsymbol{\varphi}_i^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}_i(t-1)$$

minimum description length (MDL) criterion

Another selection rule proposed by Rissanen, based on the same principle, has the form

$$\mathrm{MDL}(i) = N \log \widehat{\sigma}_i^2(N) + n_i \log N \quad \xrightarrow{i} \quad \min$$

Bayesian approach to identification of the model structure

Bayesowskie podejście do identyfikacji strukturalnej

Consider k system descriptions of the form

$$y(t) = \boldsymbol{\varphi}_i^{\mathrm{T}}(t)\boldsymbol{\theta}_i(t) + n_i(t), \quad i = 1, \ldots, k$$

Suppose that i – the model order – is a realization of a random variable with a noninformative prior distribution

$$\pi(i) = \frac{1}{k}, \quad i = 1, \ldots, k$$

<u>Problem</u>

Based on the available measurements $\Omega(N)$ find the best one-step-ahead predictor of $y(N+1)$, i.e., predictor that minimizes

$$\mathrm{E}\left\{[y(N+1) - \widehat{y}(N+1|N)]^2\right\}$$

It can be shown that the optimal solution has the form

$$\widehat{y}(N+1|N) = \sum_{i=1}^{k} \mu_i(N)\widehat{y}_i(N+1|N)$$

where $\widehat{y}_i(N+1|N) = \boldsymbol{\varphi}_i^{\mathrm{T}}(N+1)\widehat{\boldsymbol{\theta}}_i(N)$ and $\mu_i(N) = p(i|\Omega(N))$ denotes posterior probability of i given $\Omega(N)$.

Bayesian approach to identification
Bayesowskie podejście do identyfikacji

Observe that, for the set of nested models $\boldsymbol{\varphi}_1(t) \prec \ldots \prec \boldsymbol{\varphi}_k(t)$, $\forall t$, the optimal predictor can be rewritten in the form

$$\widehat{y}(N+1|N) = \boldsymbol{\varphi}_k^{\mathrm{T}}(N+1)\bar{\boldsymbol{\theta}}_k(N)$$

where $\bar{\boldsymbol{\theta}}_k(t)$ is the k-dimensional vector of averaged estimates

$$\bar{\boldsymbol{\theta}}_k(N) = \sum_{i=1}^{k} \mu_i(N) \begin{bmatrix} \widehat{\boldsymbol{\theta}}_i(N) \\ \mathbf{0}_{k-i} \end{bmatrix}$$

Bayesian "redefinition" of the AIC and PLS criteria

Use

$$\mu_i(N) \propto \exp\left\{-\frac{\mathrm{AIC}(i)}{2}\right\}$$

or

$$\mu_i(N) \propto [\mathrm{PLS}(i)]^{-N/2}$$

The proportionality coefficients should be chosen so that

$$\sum_{i=1}^{k} \mu_i(N) = 1$$

Model validation
Badanie adekwatności modeli

Model validation is based on analysis of residual errors

$$e(t) = n[t, \widehat{\boldsymbol{\theta}}(N)] = y(t) - \boldsymbol{\varphi}^{\mathrm{T}}(t)\widehat{\boldsymbol{\theta}}(N)$$

błędy resztowe

When the model is "satisfactory"?

- When the sequence of residual errors $e(t)$ can be regarded as a realization of white noise (zero-mean uncorrelated random variables) [signals & systems].

- When the quantities $e(t)$ and $u(t-i)$ can be regarded as mutually uncorrelated for every value of $i > 0$ [systems].

$$\gamma_1(i) = \frac{\mathrm{E}[e(t)e(t-i)]}{\mathrm{E}[e^2(t)]}$$

$$\widehat{\gamma}_1(i; N) = \frac{\sum_{t=1}^{N} e(t)e(t-i)}{\sum_{t=1}^{N} e^2(t)}$$

$$\gamma_2(i) = \frac{\mathrm{E}[e(t)u(t-i)]}{\sqrt{\mathrm{E}[e^2(t)]\mathrm{E}[u^2(i)]}}$$

$$\widehat{\gamma}_2(i; N) = \frac{\sum_{t=1}^{N} e(t)u(t-i)}{\sqrt{\sum_{t=1}^{N} e^2(t) \sum_{t=1}^{N} u^2(t)}}$$

Model validation
Badanie adekwatności modeli

Remark

The quantity

$$\gamma = \frac{\mathrm{E}[XY]}{\sqrt{\mathrm{E}[X^2]\mathrm{E}[Y^2]}}$$

is called the normalized correlation between random variables X and Y (*unormowany współczynnik korelacji zmiennych losowych*). It can be shown that $|\gamma| \leq 1$.

1. Elementary correlation tests

Autocorrelation test

For any $i > 0$ consider the following null and alternative hypotheses

$$\mathcal{H}_0 : \quad \gamma_1(i) = 0, \qquad \mathcal{H}_1 : \quad \gamma_1(i) \neq 0$$

It can be shown that

$$p\left(\sqrt{N}\widehat{\gamma}_1(i;N) \mid \mathcal{H}_0\right) = \mathcal{N}(0,1)$$

and

$$P\left(|\widehat{\gamma}_1(i;N)| \leq \frac{2}{\sqrt{N}} \mid \mathcal{H}_0\right) \cong 0.95$$

Elementary correlation tests
Elementarne testy korelacyjne

Hence, after adopting significance level $\epsilon = p_{\mathrm{I}} = 0.05$, one arrives at the following test

$$|\widehat{\gamma}_1(i;N)| \begin{cases} \leq \frac{2}{\sqrt{N}} & \text{accept } \mathcal{H}_0 \\ > \frac{2}{\sqrt{N}} & \text{reject } \mathcal{H}_0 \end{cases}$$

<u>Cross-correlation test</u>

For any $i > 0$ consider the following null and alternative hypotheses

$$\mathcal{H}_0: \quad \gamma_2(i) = 0, \qquad \mathcal{H}_2: \quad \gamma_2(i) \neq 0$$

It can be shown that

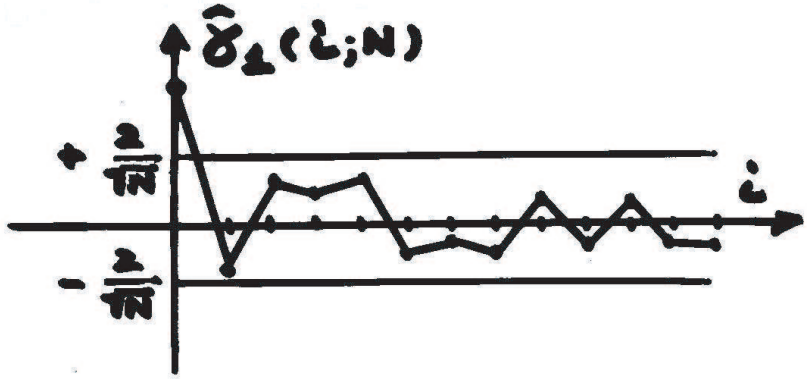
$$p\left(\sqrt{N}\widehat{\gamma}_2(i;N) \mid \mathcal{H}_0\right) = \mathcal{N}(0,1)$$

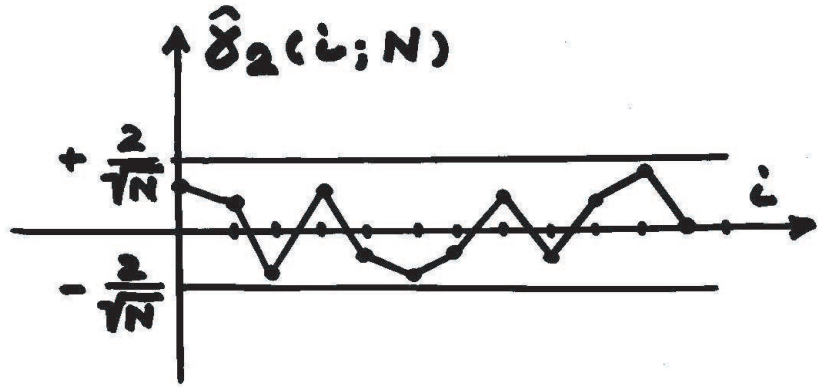
which leads to the following test

$$|\widehat{\gamma}_2(i;N)| \begin{cases} \leq \frac{2}{\sqrt{N}} & \text{accept } \mathcal{H}_0 \\ > \frac{2}{\sqrt{N}} & \text{reject } \mathcal{H}_0 \end{cases}$$

Elementary correlation tests

Elementarne testy korelacyjne





<u>Warning</u>

Suppose that the autocorrelation test is extended to K normalized correlations, i.e., one requires that

$$|\widehat{\gamma}_1(i;N)| \leq \frac{2}{\sqrt{N}}, \quad i = 1, \ldots, K$$

Then the probability of accepting the null hypothesis $\mathcal{H}_0$: $\gamma_1(1) = \ldots = \gamma_1(K) = 0$, given that this hypothesis is true, is equal to $(1-\epsilon)^K$, where ϵ is the adopted significance level. Note that for $\epsilon = 0.05$ one obtains $(1-\epsilon)^5 = 0.729$ and $(1-\epsilon)^{50} = 0.097$. The same remark applies to the cross-correlation test.

Model validation
Badanie adekwatności modeli

2. Group correlation tests

<u>Autocorrelation test</u>

Let

$$\boldsymbol{\gamma}_1 = \begin{bmatrix} \gamma_1(1) \\ \vdots \\ \gamma_1(K) \end{bmatrix}, \quad \widehat{\mathbf{r}}_1(N) = \sqrt{N} \begin{bmatrix} \widehat{\gamma}_1(1;N) \\ \vdots \\ \widehat{\gamma}_1(K;N) \end{bmatrix}$$

Consider the following null and alternative hypotheses

$$\mathcal{H}_0: \quad ||\boldsymbol{\gamma}_1|| = 0, \qquad \mathcal{H}_1: \quad ||\boldsymbol{\gamma}_1|| \neq 0$$

It can be shown that

$$p\left(\widehat{\mathbf{r}}_1(N) \mid \mathcal{H}_0\right) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$p\left(||\widehat{\mathbf{r}}_1(N)||^2 \mid \mathcal{H}_0\right) = \chi^2(K)$$

This leads to the following "group" correlation test

$$||\widehat{\mathbf{r}}_1(N)||^2 \begin{cases} \leq \eta_0 & \text{accept } \mathcal{H}_0 \\ > \eta_0 & \text{reject } \mathcal{H}_0 \end{cases}$$

where η_0 is the decision threshold chosen so that

$$p_{\mathrm{I}} = P\left(||\widehat{\mathbf{r}}_1(N)||^2 > \eta_0 \mid \mathcal{H}_0\right) = \epsilon$$

Group correlation tests
Grupowe testy korelacyjne

<u>Cross-correlation test</u>

Let

$$\boldsymbol{\gamma}_2 = \begin{bmatrix} \gamma_2(1) \\ \vdots \\ \gamma_2(K) \end{bmatrix}, \quad \widehat{\mathbf{r}}_2(N) = \sqrt{N} \begin{bmatrix} \widehat{\gamma}_2(1;N) \\ \vdots \\ \widehat{\gamma}_2(K;N) \end{bmatrix}$$

Consider the following null and alternative hypotheses

$$\mathcal{H}_0: \quad ||\boldsymbol{\gamma}_2|| = 0, \qquad \mathcal{H}_1: \quad ||\boldsymbol{\gamma}_2|| \neq 0$$

It can be shown that

$$p\left(\widehat{\mathbf{r}}_2(N) \mid \mathcal{H}_0\right) = \mathcal{N}(\mathbf{0}, \mathbf{I})$$
$$p\left(||\widehat{\mathbf{r}}_2(N)||^2 \mid \mathcal{H}_0\right) = \chi^2(K)$$

This leads to the following "group" correlation test

$$||\widehat{\mathbf{r}}_2(N)||^2 \begin{cases} \leq \eta_0 & \text{accept } \mathcal{H}_0 \\ > \eta_0 & \text{reject } \mathcal{H}_0 \end{cases}$$

THE END