Model Identification and Data Analysis

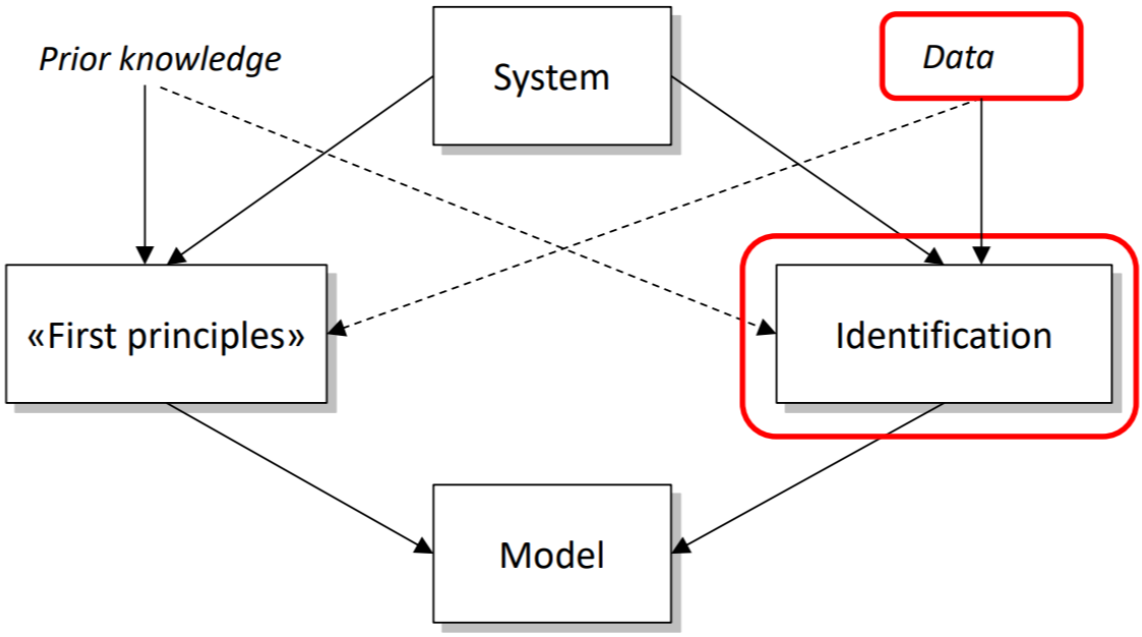
github.com/martinopiaggi/polimi-notes

2023-2024

Contents

1 Introduction

The goal of this course is to provide a framework for black-box modelling of I/O dynamical systems and time series. In engineering you usually start from principles and obtain the model. Here we use the data as a source for information to build the model



1.1 Stochastic process

The purpose of making predictions/identification on the future behavior of a signal requires taking into account the uncertainty in the model. This leads us to the need for a new concept, the stochastic process SP, which extends the concept of random variables to signals.

A stochastic process SP is a n infinite sequence of random variables that depend on the outcome of a random experiment.

In this context is important to refresh this stuff:

- **Mean**: $E[v]$. Key property of expected value $E[\alpha_1 v_1 + \alpha_2 v_2] = \alpha_1 E[v_1] + \alpha_2 E[v_2]$
- **Variance**: $E[(v - E[v])^2] \geq 0$
- **Covariance**:

$$\gamma(t_1, t_2) = E[(v(t_1, s) - m(t_1))(v(t_2, s) - m(t_2))]$$

The covariance function is a useful statistical measure that indicates how two samples of the **same** stochastic process change together ("covary"). If (for example) the SP is periodic the correlation is very high. Remember also that the covariance between two functions is:

$$\gamma(\tau) = E[(y_1(t) - \mu_1)(y_2(t + \tau) - \mu_2)]$$

While the formula to compute the **correlation** function $\tilde{\gamma}(\tau)$ is:

$$\tilde{\gamma}(\tau) = E[v_1(t)v_2(t-\tau)] = \gamma(\tau) + \bar{\mu}_1\bar{\mu}_2$$

$$\tilde{\gamma}(\tau) = E[y(t)y(t-\tau)] = \gamma(\tau) + \bar{\mu}^2$$

We will focus on **Stationary** Stochastic Process (SSP) $y(t)$ which have the following characteristics:

- $m(t) = m \forall t$ "any realization will be a constant outcome"
- $\gamma_y(t_1, t_2)$ depends only on $\tau = t_2 - t_1$ and can therefore be indicated with $\gamma_y(\tau)$. This basically means "it's not important when you take the samples but only the absolute shift between the two". Also (always if $y(t)$ is stationary) the covariance γ_y has these proprieties:
  - $\gamma(0) = E[(v(t,s) - m)^2] \geq 0$ positivity
  - $|\gamma(\tau)| \leq \gamma(0), \forall \tau$ not increasing
  - $\gamma(\tau) = \gamma(-\tau), \forall \tau$ even

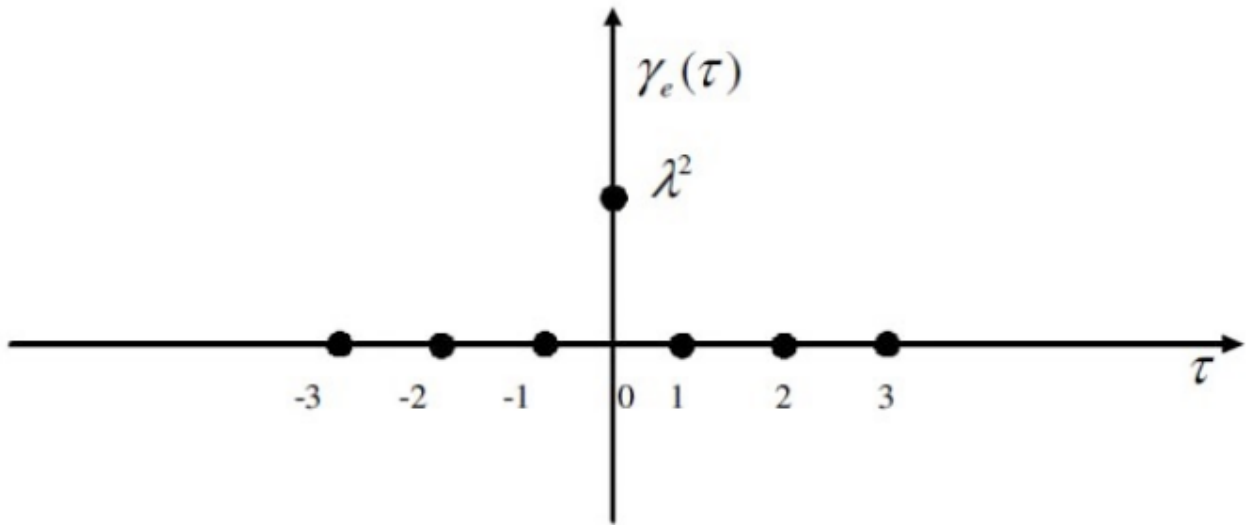
To recap the property about eh $\gamma_y(t_1, t_2)$: the properties of a SSP don't depend on time t. In other words, what happens at time t is representative also of what happens at time $t + \tau$ (no probabilistic difference).

1.2 White noise

A White Noise (WN) is a SSP with $\gamma(\tau) = 0$ for all $\tau \neq 0$.

A WN is an unpredictable signal meaning that there is **NO CORRELATION** between $\eta(t_1)$ and $\eta(t_2)$. Usually, a WN is defined as $\eta(t) \sim WN(0, \lambda^2)$. Having them with zero-mean is not mandatory.

The fact that the covariance function is zero everywhere except for the origin is another way to say that the past is not informative to know the future (**whiteness** property). We will use white noise to represent **uncertainty** in the model.



Unpredictable by definition.

2 MA, AR and ARMA processes

We will see these families of SSP:

- **MA (Moving Average):** Use past errors to predict future values.
- **AR (Autoregressive):** Use past values to predict future values.
- **ARMA (Autoregressive Moving Average):** Combine both AR and MA elements

2.1 Moving averages

With $e(t) \sim wn(0, \lambda^2)$ we can define the moving average process of order n $MA(n)$ in this way:

$$y(t) = c_0 \cdot e(t) + c_1 e(t-1) + c_2 \cdot e(t-2) + \cdots + c_n e(t-n)$$

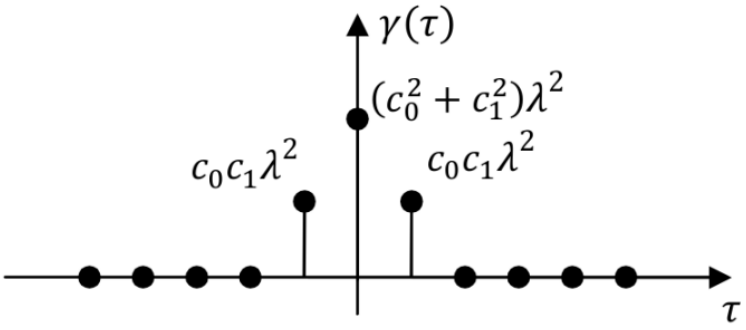
$MA(n)$ is still stationary . A generalization of $MA(n)$ is:

$$MA(\infty) = \sum_{i=0}^{+\infty} c_i e(t-i), e(t) \sim \text{wn}(0, \lambda^2)$$

but the series must converge! The properties to remember here are:

$$m_y(t) = m_y(0) = 0$$

$$\gamma_y(\tau) = \begin{cases} \left({c_0}^2 + {c_1}^2 + {c_2}^2 + ... + {c_n}^2\right) \cdot \lambda^2 & \text{if } \tau = 0 \\ (c_0 c_1 + c_1 c_2 + ... + c_{n-1} c_n) \cdot \lambda^2 & \text{if } \tau = \pm 1 \\ (c_0 c_2 + c_1 c_3 + ... + c_{n-2} c_n) \cdot \lambda^2 & \text{if } \tau = \pm 2 \\ \vdots & \\ (c_0 c_n) \cdot \lambda^2 & \text{if } \tau = n \\ 0 & \text{if } |\tau| > n \end{cases}$$



So for example, for $y(t) = e(t) + c \cdot e(t-1)$:

$$\gamma_y(0) = 1 + c^2$$

and

$$\gamma_y(1) = \gamma_y(-1) = c$$

2.1.1 AR

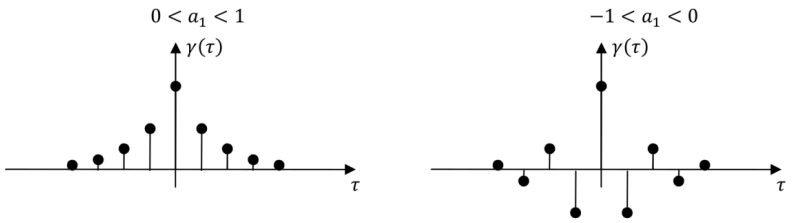
Autoagressive models will allow us to have $\gamma(\tau) \neq 0$ for all τ using a finite set of coefficients.

$$y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + e(t)$$

with $e(t) \sim wn(0, \lambda^2)$. Where n is the order of the process and $e(t) \sim WN(0, \lambda^2)$. Covariance function with $n = 1$ (case of $AR(1)$):

$$\begin{cases} \frac{1}{1-a^2}\lambda^2 & \tau = 0 \\ \gamma(\tau) = a\gamma(\tau - 1) & |\tau| > 0 \end{cases}$$

well defined if $|a| < 1$.



The $MA(\infty)$ representation of an AR process refers to representing an autoregressive process as an infinite moving average MA process. performing the recursion.

The intuition behind this representation is that an AR process relies on its past values which, in turn, also depend on their own past values, and so on. If you keep substituting back these dependencies, you end up with a representation that expresses the current value of the process as the sum of the effects of all past and current shock terms ϵ_t, each weighted by a ψ_i coefficient.

$$\begin{array}{rl} y(t) & = \alpha q(t-1) + e(t) \\ y(t) & = q(t-1) + e(t-1) + e(t-2) \\ & = e(t) + ae(t-1) + a^2 q(t-2) \\ y(t) & = e(t) + ae(t-1) + a^2[\alpha(t-3) + e(t-2)] \\ & = e(t) + ae(t-1) + a^2 e(t-2) + a^3 q(t-3) \\ \vdots & \vdots \\ y(t) & = e(t) + ae(t-1) + a^2 e(t-2) + ... = \sum_{i=0}^{\infty} a^i e(t-i) \quad \leftarrow \text{MA}(\infty) \end{array}$$

The $MA(\infty)$ representation is mainly of theoretical interest to write the steady-state solution of an AR process but in practical time series modeling is useless.

It's interesting to notice that the constraint $|a| < 1$ is need to check to bound the variance value, which must be finite.

$$\sum_{i=0}^{+\infty} c_i^2 = \sum_{i=0}^{+\infty} a^{2i}$$

The variance must be convergent if $|a| < 1$. This is a critical condition for the stability and stationarity of the $AR(1)$ process. ### ARMA

An ARMA process is a combination of Autoregressive (AR) and Moving Average (MA) models.

$$v(t) = a_1 v(t-1) + ... + a_{n_a} v(t-n_a) + c_0 \eta(t) + ... + c_1 \eta(t-1) + ... + c_{n_c} \eta(t-n_c)$$

$ARMA(n_a, n_c)$ $processes include all other AR and MA models as special cases. Also we can say that given an $ARMA$ process (n_a, n_b) the covariance function:

- if $n_a > n_b$: $\gamma_y(\tau)$ becomes recursive for $\tau = n_a$
- if $n_a \leq n_b$: $\gamma_y(\tau)$ becomes recursive for $\tau = n_b + 1$

2.1.2 ARMAX

ARMAX models extends ARMA including an **exogenous inputs**. ARMAX(m, n, p, k) with k pure input/output (I/O) delay. p order of the exogenous part:

$$\begin{aligned} y(t) = a_1 y(t-1) + a_2 y(t-2) + \cdots + a_m y(t-m) + \\ + c_0 t(t) + c_1 e(t-1) + \cdots\cdots c_n e(t-n) + \\ + b_0 u(t-k) + b_1 u(t-k-1) + \cdots b_p u(t-k-p) \end{aligned}$$

The concept of $e(t)$ is a mathematical artifact representing the stochastic nature of measurements. In reality, $e(t)$ does not exist, but when is the output of a system, both deterministic and stochastic effects need to be considered. The deterministic effect is caused by an input u, while the stochastic effect is due to the **inherent errors in measurements**.

2.2 Processes with non-null term

TLDR: the key takeaway is that incorrect results are obtained when dealing with non-zero mean processes if one fails to recognize their non-zero mean status or doesn't read the exercise carefully.

Generally computations with non-zero mean signals are more complex compared to the zero-mean case. However, a tip to make the computation more efficient is to introduce unbiased versions of the processes $\tilde{y}$ and $\tilde{e}$, which are obtained by removing the means directly. Unbiased versions are still useful because:

$$\gamma_{\tilde{y}}(\tau) = \gamma_{\gamma}(\tau) \quad \forall \tau$$

$$\begin{cases} \tilde{y}(t) = y(t) - m_y \\ \tilde{e}(t) = e(t) - m_e \end{cases}$$

By writing the equation for $\tilde{y}$, the terms can be simplified to make the computations more manageable.

3 Frequency Domain

3.1 Transfer functions

The transfer function $W(z)$, being a ratio of two polynomials, acts as a **digital filter**. It is instrumental in transforming WN into an ARMA process and is applied in analyzing stationary processes through digital filter theory.

Shift Operators in Time Domain Analysis

- Backward shift operator z^{-1} shifts a signal back in time
- Forward shift operator z shifts a signal forward in time.

Properties of z and z^{-1} include linearity, recursive application, and linear composition. Using these properties, we can rewrite an AR model equation in terms of z and $z-1$, revealing the relationship between the operators and the AR model. The resulting equation shows that the output y_t is the **ratio** of two polynomials of z:

$$y(t) = \frac{c_0 + c_1 z^{-1} + \cdots c_n z^n}{1 - a_1 z^{-1} + \cdots a_m z^{-m}} e(t) = \frac{C_{(z)}}{A(z)} e(t) = W(z) e^{it}(t)$$

3.2 Asymptotically stability

Stability analysis involves examining the poles of the transfer function:

$$y(t) = \frac{C(z)}{A(z)} e(t)$$

- Zeros are values of z where the transfer function equals zero
- Poles are the inverse of the transfer function's roots (roots of $A(z)$ in case of no cancellations).

The locations of poles and zeros, particularly within the unit circle in the complex plane, are indicative of system stability.

- To compute zeros, set the numerator of the transfer function equal to zero.
- For poles, set the denominator equal to zero.

A system is considered asymptotically **stable** if all **poles** lie strictly inside the unit circle, and it is deemed a minimum phase system if both poles and zeros reside within the unit circle.

3.3 Canonical representation of SSP

ARMA (and so MA and AR) processes can be represented in various domains:

- **Time domain representation**

$$y(t) = a_1 y(t-1) + \cdots a_m y(t-m) + c_o e(t) + \cdots c_n e(t-n)$$

- **Operatorial representation**

$$y(t) = \frac{C(z)}{A(z)} e(t)$$

- **Probabilistic representation**

$$y(t) = m_y, \gamma_y(\tau)$$

- **Frequency domain**

$$y(t) = m_y, \Gamma_y(\omega)$$

They are all equivalent but not **unique**. The uniqueness aspect it will be problematic, and so before introducing prediction theory, we need to better understand the non-uniqueness property and find ways to make the representation unique.

Taken the process in operatorial representation:

$$y(t) = \frac{C(z)}{A(z)} e(t)$$

We can adopt this "convention" to be sure about the uniqueness of representation:

1) **monic**: the term with the highest power has coefficient equal to 1
2) **coprime**: no common factors to that can be simplified
3) have **same degree**
4) have **roots inside the unit circle**

3.4 The spectral factorization theorem

In the analysis of discrete-time signals through the use of the transfer function $W(z)$, if $y(t) = W(z)v(t)$ and:

- $v(t)$ is stationary
- $W(z)$ is stable

$y(t)$ is well defined and stationary.

3.5 Final value theorem

The final value theorem or theorem of the gain is:

$E[y(t)] = W(1) \cdot E[u(t)]$

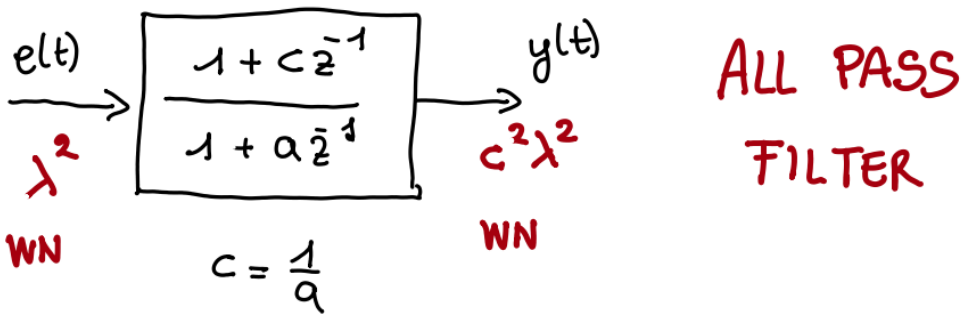
NB: The final value theorem can be applied to $W(z)$ that are not in canonical form.

Final value theorem is useful when we want to define de-biased process:

1. Find mean with gain theorem:
$$E[y(t)] = W(1) \cdot \mu = \bar{y}$$
2. Define de-biased processes:
$$\tilde{y}(t) = y(t) - \bar{y}$$
$$\tilde{e}(t) = e(t) - \mu$$
3. Do whatever you want, like computing the covariance $\gamma_{\tilde{y}}(\tau) = \gamma_{\gamma}(\tau) \quad \forall \tau$

3.6 All Pass Filter

In the context of transfer functions $W(z)$ and digital filters, an all pass filter is a particular process which transform a white noise in a white noise with a different variance.



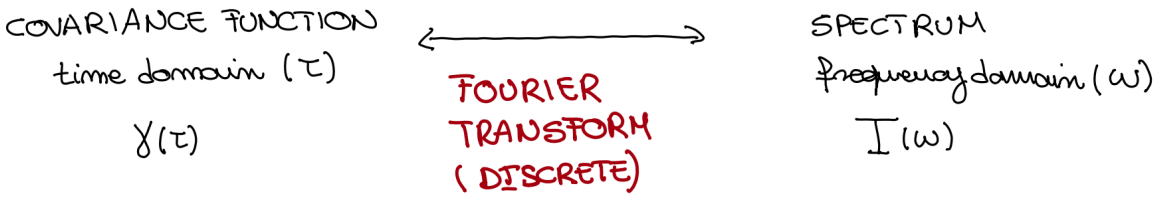
An APF is useful for converting a system to canonical form: we can do this by aggregating coefficients and multiplying by a fraction of identical polynomials (so that we essentially multiply by one) and then we redefine a new white noise process.

Apart the variance everything remains the same; for example if input $u(t)$ and output $y(t)$:

- If $u(t)$ is constant, $y(t)$ is constant.
- If $u(t)$ is sinusoidal, $y(t)$ is sinusoidal.

3.7 Spectral representation

Spectral density of a stationary stochastic process

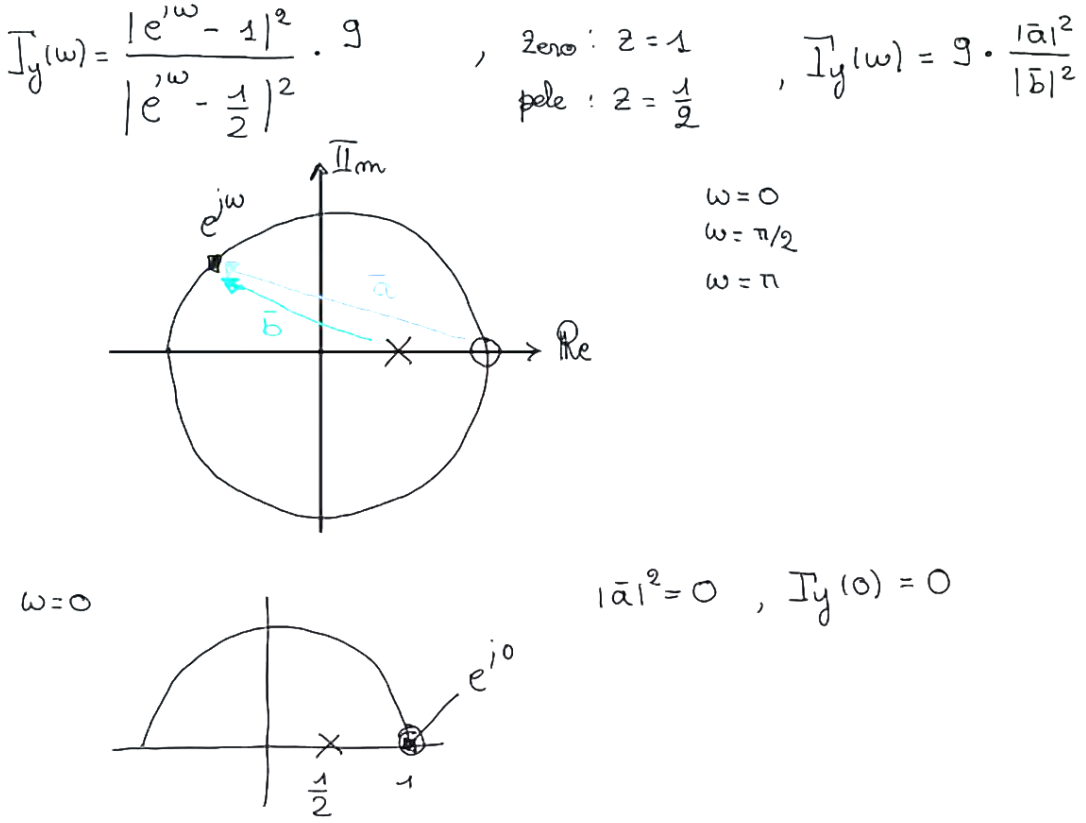


$$\Gamma_y(w) = \sum_{\tau=-\infty}^{\infty} \gamma_y(\tau) \cdot e^{-jw\tau}$$

For a stationary process $y(t)$, $\Gamma_y(\omega)$:

- positive
- real
- even $\Gamma(-\omega) = \Gamma(\omega)$
- periodic with period $T = 2\pi$.

Its graph can be represented in the upper half plane of a 2D plot, as the imaginary part is always zero.



The "general rule" regarding the effect of the poles and zeros of the transfer function is that if the generic point $e^{j\omega}$ is:

- Near **zeroes**, the $\Gamma_y(\omega)$ exhibits attenuation. For a zero on the unit circle there is the so called **blocking property** of the transfer function, which refers to the phenomenon where a zero on the unit circle in the z-domain causes the system to completely attenuate that specific frequency component of the input signal. As the frequency moves away from this zero, the attenuation decreases.
- Near **poles**, the $\Gamma_y(\omega)$ is high. The frequency response is amplified. If a pole is near the unit circle, it signifies a strong frequency response at the angle ω corresponding to the pole's position.

3.8 Spectrum Properties

For stationary processes, the following properties hold for the power spectral density $\Gamma(\omega)$:

1. Scalar Multiple If $y(t) = ax(t)$, then:
$$\Gamma_y(\omega) = a^2\Gamma_x(\omega)$$
2. Sum of Uncorrelated Processes: If $z(t) = x(t) + y(t)$ and $x(t)$ and $y(t)$ are uncorrelated, then:
$$\Gamma_z(\omega) = \Gamma_x(\omega) + \Gamma_y(\omega)$$
3. Inverse transform returns the covariance function based on the spectral density, noted as the inverse discrete Fourier transform.

$$\gamma_y(z) = F^{-1}(\Gamma_y(\omega)) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\Gamma_y(\omega)d\omega$$

4. The spectrum of a WN is constant and equal to its variance:

$$\Gamma_e(\omega) == Var[e(t)] = \lambda^2$$

3.8.1 Example

If $v(t) = ax(t) + by(t)$ and $x(t)$ and $y(t)$ are uncorrelated, then:

$$\Gamma_v(\omega) = a^2\Gamma_x(\omega) + b^2\Gamma_y(\omega)$$

3.9 Fundamental theorem of spectral analysis

Very useful theorem to compute the spectrum:

$$\Gamma_y(\omega) = W(z = e^{j\omega})W(z = e^{-j\omega})\Gamma_u(\omega)$$

It's possible to apply the fundamental theorem of spectrum analysis also if $W(z)$ is not in the canonical form :)

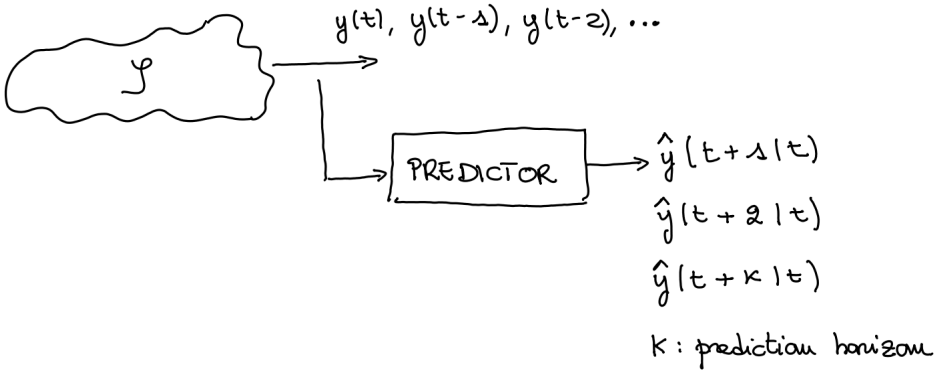
Recall of useful formulas for spectrum qualitative study are:

- $e^{j\omega} = \cos\omega + j\sin\omega$
- $e^{-j\omega} = \cos\omega - j\sin\omega$
- $e^{-j\omega} + e^{+j\omega} = 2\cos(\omega)$

Remember that is always possible to write the spectral density function as a real-valued function composed by cosine terms easier to interpret.

4 Model prediction

We want to find the predictor $\hat{y}(t)$ of process $y(t)$ from the past values of y.



Steps:

0) analysis of the system
1) evaluation of the canonical representation of the system
2) computation of the predictor
3) evaluate the prediction error

Practically the optimal predictor from data is:

$$\hat{y}(t+n|t) = \frac{F(z)}{C(z)}y(t)$$

This because:

$$W(z) = \frac{C(z)}{A(z)} = \text{long division} = E(z) + \frac{z^{-u}F(z)}{A(z)}$$

where $E(z)$ is the result of the long division, while $F(z)$ is the rest.

So:

$$y(t+k) = \frac{C(z)}{A(z)}e(t+k) = \left[E(z) + z^{-k}\frac{F(z)}{A(z)}\right]e(t+k)$$

In practice what happens is that the $E(z)e(t+k)$ part will be always be discarded and treated as the prediction error, while this $z^{-k}\frac{F(z)}{A(z)}e(t+k)$ can be use.

Actually to be used the predictor with respect to the data we will use this $e(t) = \frac{A(z)}{Cz}y(t)$ to re-write the equation into:

$$\hat{y}(t+n|t) = \frac{F(z)}{A(z)}e(t) = \frac{F(z)}{A(z)} \cdot \frac{A(z)}{C(z)} \cdot y(t) = \frac{F(z)}{C(z)}y(t)$$

Considerations:

- $\hat{y}(t+k,s)$ is a stochastic process, **stationary** (because the denominator $C(z)$ still has roots inside the unit circle).
- The quality of the prediction get worse with increasing k.
  - For $k=1$ the variance of the error coincides with the variance of the white noise
  - For $k \to \infty$ the variance of the error coincides with the variance of the process and the predictor becomes the expected value of the process

$$var[e(t)] \leq var[\epsilon(t|t-k)] < var[y(t)]$$

4.0.1 1-step predictor

If you use the formula previously presented for a 1 step ARMA predictor with null mean you will always obtain $E(z) = 1$. This aspect permits to make some simplification in the final formula and just use these "shortcuts":

$$\hat{y}(t|t-1) = \frac{C(z) - A(z)}{A(z)}e(t)$$

from data:

$$\hat{y}(t|t-1) = \frac{C(z) - A(z)}{C(z)}y(t)$$

And since:

$$\epsilon(t|t-1) = y(t) - \hat{y}(t|t-s) = y(t) - \frac{C_m - A_m}{C_m}y(t) = \left(\frac{C_m - C_m + A_m}{C_m}\right)y(t)$$

the prediction error will always be:

$$\epsilon(t|t-1) = \frac{A_m(z)}{C_m(z)} \cdot y(t)$$

4.0.2 AR

The usual $AR(1)$ process:

$$v(t) = av(t-1) + \eta(t)$$

The transfer function is:

$$W(z) = \frac{1}{1 - az^{-1}}$$

So the optimal 1-step ahead predictor is:

$$\hat{v}(t|t-1) = \frac{a}{z}v(t) = av(t-1)$$

With the optimal 2-steps ahead predictor we can see that:

$$\hat{v}(t|t-2) = \frac{a^2}{z^2}v(t) = a^2v(t-2)$$

By generalization we can say that for an $AR(1)$ process the optimal predictor is always:

$$\hat{v}(t+r|t) = a^r v(t)$$

Another notorious predictor is that for $AR(n)$ the 1-step predictor will be always:

$$\hat{v}(t|t-1) = a_1v(t-1) + a_2v(t-2)\dots a_nv(t-n)$$

4.0.3 ARMA with not null mean

Remember that for processes with non-null term we have some troubles? Also when computing predictors with not null mean we have to make some *sbatti*:

$$y(t+k) = \hat{y}(t+k) + my$$
$$\hat{y}(t+k|t) = \hat{y}(t+k|t) + my$$

To make the predictor we have to just consider the unbiased process, so first of all compute the mean with gain theorem:

$$E[y(t)] = W(1) \cdot \mu = my$$

Then define **de-biased** processes:

$$\tilde{y}(t) = y(t) - my$$
$$\tilde{e}(t) = e(t) - \mu$$

Redefine the process using unbiased elements:

$$\tilde{y}(t+k|t) = \frac{C(z)}{A(z)} \cdot \tilde{e}(t)$$

And apply the generic formulas for the predictor. At the end we can make the prediction adjustment for non-zero-mean:

$$\hat{y}(t+k|t) = \tilde{y}(t+k|t) + my$$

$$\hat{y}(t+k|t) = \frac{F(z)}{C(z)}\tilde{y}(t) + my = \frac{F(z)}{C(z)}(y(t) - my) + my$$
$$= \frac{F(z)}{C(z)}y(t) - \frac{F(z)}{C(1)}my + my$$

and the final formula is:

$$\hat{y}(t+k|t) = \frac{F(z)}{C(z)} \cdot y(t) + \left(1 - \frac{F(1)}{C(1)}\right) \cdot my$$

4.1 ARX and ARMAX

We want to predict the generic ARMAX process:

$$y(t) = \frac{C(z)}{A(z)}e(t) + \frac{B(z)}{A(z)}u(t-d)$$

where d is the pure delay.

The ARMAX predictor from data:

$$\widehat{y}(t|t-k) = \frac{F_{k(z)}}{C(z)}y(t-k) + \frac{B(z)E(z)}{C(z)}u(t-d)$$

The ARMAX 1-step predictor ($E(z) = 1$) from data:

$$\widehat{y}(t|t-1) = \frac{C(z) - A(z)}{C(z)}y(t-1) + \frac{B(z)}{C(z)}u(t-d)$$

Demonstration

Assuming already canonical $\frac{C(z)}{A(z)}$ and $u(t)$ known process (so completely **deterministic**) we only need predictor for stochastic part. Said this we compute the predictor in a similar way as before:

$$y(t) = \frac{C(z)}{A(z)}e(t) + \frac{B(z)}{A(z)}u(t-d)$$

Now, let's consider the long division between $C(z)$ and $A(z)$:

$$\frac{C(z)}{A(z)} = E(z) + \frac{F(z)}{A(z)}$$

Which can be written as:

$$A(z) = \frac{C(z) - F(z)}{E(z)}$$

Let's put it in the ARMAX equation:

$$\left(\frac{C(z) - F(z)}{E(z)}\right)y(t) = C(z)e(t) + B(z)u(t-d)$$

Let's rewrite it as:

$$C(z)y(t) = F(z)y(t) + C(z)E(z)e(t) + B(z)E(z)u(t-d)$$

Let's divide by $C(z)$:

$$y(t|t-k) = \frac{F(z)}{C(z)}y(t-k) + \frac{B(z)E(z)}{C(z)}u(t-d) + E(z)e(t)$$

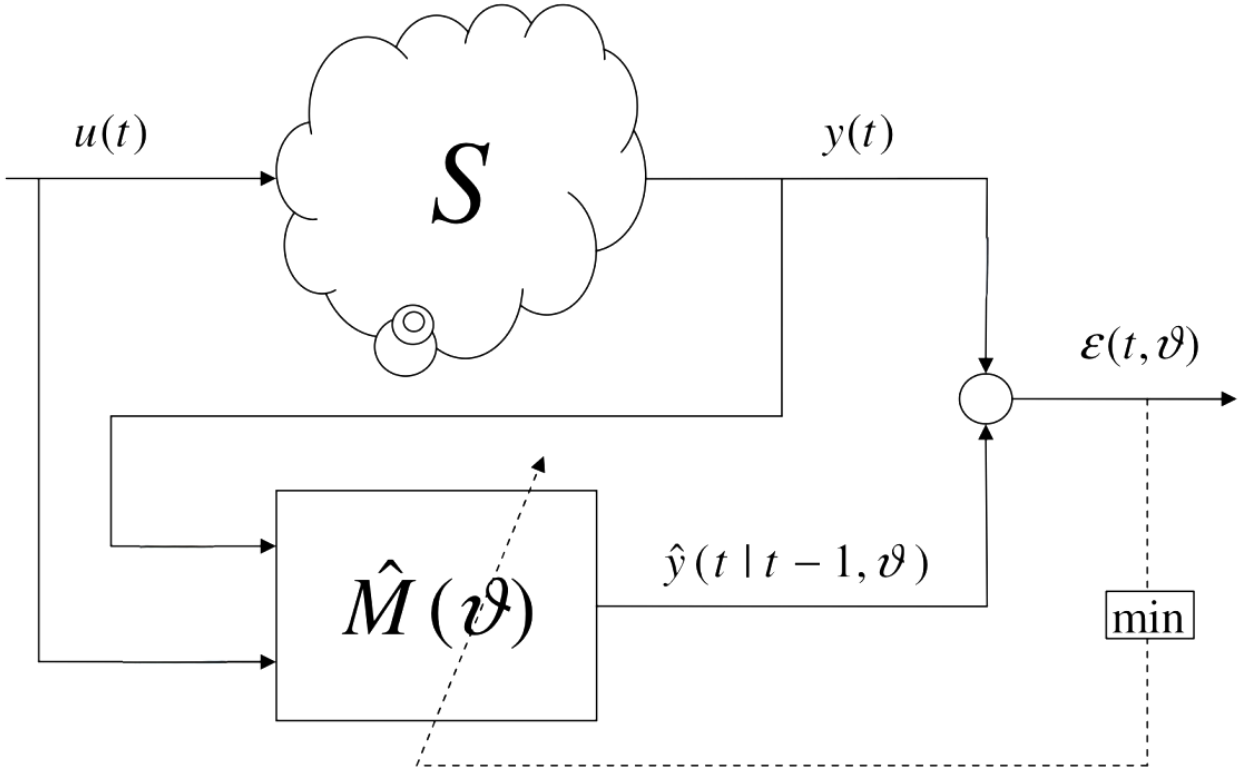
Last term as usual is unpredictable at time $t-k$:

$$\widehat{y}(t|t-k) = \frac{F_{k(z)}}{C(z)}y(t-k) + \frac{B(z)E(z)}{C(z)}u(t-d)$$

5 Model identification

An identification problem is essentially a parametric optimization problem: we have a set of data $y(1), y(2), \dots (u(1), u(2), \dots)$ and we want to find the best model that approximate these data focusing on **parametric** identification of dynamic systems.

The concept of making a model relying on data as the source is the foundation of what we will see in Machine Learning .



Choice of $J^N(\theta) > 0$

Predictive approach

$$J(\theta) = \mathbb{E}\left[(y(t+1) - \hat{y}(t+1|t,\theta))^2\right]$$

Ideal objective:

- A good model should return
- Low variance of the 1-step

- Ahead prediction error

Sample version of the objective:

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

Why 1-step ahead prediction? and not 2-step or k-step ... ? If $\hat{y}(t+1|t)$ is the optimal predictor of $y(t)$, the variance of $\epsilon(t+1|t)$ is:

$$\text{var}[\epsilon(t+1|t)] = \text{var}[e(t)] = \lambda^2$$

We obtain λ^2 (or better, an estimate of λ^2) for free .

$$J^N(\theta) : \mathbb{R}^{n_\theta} \to \mathbb{R}^+$$

Two relevant situations:

1. $J^N(\theta)$ quadratic (AR, ARX)
2. $J^N(\theta)$ non-quadratic (ARMA, MA, ARMAX)

We made a few assumptions: - S lies within a specific model set - n_a, n_b, n_c fixed a-priori - data could have been given by somebody else (could be non-informative)

5.1 Identification of AR/ARX Models

Generic expression of $AR(X)$:

$$M(\theta) : y(t) = \frac{B(z)}{A(z)}u(t-d) + \frac{1}{A(z)}e(t), \quad e(t) \sim WN(0, \lambda^2)$$

with:

$$A(z) = 1 - a_1 z^{-1} - a_2 z^{-2} - ... - a_m z^{-m}$$

$$B(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + ... + b_{p-1} z^{-(p+1)}$$

$$C(z) = 1 \quad \text{(ARX model! No MA part)}$$

$\theta = [a_1\, a_2 \,...\, a_m\, b_0\, b_1 \,...\, b_{p-1}]^T$ are the parameters $\in \mathbb{H}$ (the parameters domain):

$$\theta = \begin{vmatrix} a_1 \\ \vdots \\ a_{n_a} \\ b_1 \\ \vdots \\ b_{n_b} \end{vmatrix} \quad \varphi(t) = \begin{vmatrix} y(t-1) \\ \vdots \\ y(t-n_a) \\ u(t-1) \\ \vdots \\ u(t-n_b) \end{vmatrix}$$

Express the 1-step ahead predictor:

$$\hat{y}(t|t-1,\theta) = \phi(t)^T\theta$$

Define the prediction error criterion as we said before:

$$J_N(\vartheta) = \frac{1}{N}\sum_{i=1}^{N}\left(\mathrm{y}(i) - \hat{y}(i \mid i-1,\vartheta)\right)^2 = \frac{1}{N}\sum_{t=1}^{N}\left(y(t) - \varphi(t)^T\vartheta\right)^2$$

A good model should return a low empirical variance of the prediction error. Substitute the predictor expression:

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \phi(t)^T\theta)^2$$

Take the derivative with respect to θ and set to zero:

$$\frac{\partial J_N(\theta)}{\partial\theta} = \frac{d}{d\theta}[J_N(\theta)] = \frac{d}{d\theta}\left[\frac{1}{N}\sum_{t=1}^{N}(y(t) - \varphi(t)^T\theta)^2\right] = 0$$

$$\begin{aligned}\frac{\partial J(\theta)}{\partial\theta} &= -\frac{1}{N}\sum_{t=1}^{N}2\left(y(t) - \theta'\varphi(t)\right)\varphi(t)^T \\ &= -\frac{2}{N}\left(\sum_{t=1}^{N}y(t)\varphi(t)^T - \sum_{t=1}^{N}\theta'\varphi(t)\varphi(t)^T\right) = 0\end{aligned}$$

Rearrange to get the normal equations:

$$\left[\frac{1}{N}\sum_{t=1}^{N}\phi(t)\phi(t)^T\right]\theta = \frac{1}{N}\sum_{t=1}^{N}\phi(t)y(t)$$

$$\left[\sum_{t=1}^{N}\varphi(t)\varphi(t)^T\right]\hat{\theta}_N = \sum_{t=1}^{N}\varphi(t)y(t)$$

we obtain the Least Squares method for AR(X) models family:

$$\hat{\theta} = \left[\sum_{t=1}^{N}\varphi(t)\varphi(t)^T\right]^{-1}\sum_{t=1}^{N}y(t)\varphi(t)^T$$

To check if $\hat{\theta}_N$ corresponds to a minimum of the cost function $J_N(\theta)$, we need to examine the first and second derivatives of $J_N(\theta)$.

The first derivative is given by:

$$\frac{\partial J_N(\theta)}{\partial\theta} = -\frac{2}{N}\sum_{t=1}^{N}\phi(t)(y(t) - \phi(t)^T\theta)$$

Setting this to zero gives us the condition for a stationary point.

The second derivative is:

$$\frac{\partial^2 J_N(\theta)}{\partial \theta^2} = \frac{2}{N}\sum_{t=1}^{N}\phi(t)\phi(t)^T$$

For $\hat{\theta}_N$ to be a minimum, the Hessian must be positive definite:

$$x^T M x > 0 \quad \forall x \neq 0$$

Applying this to our Hessian:

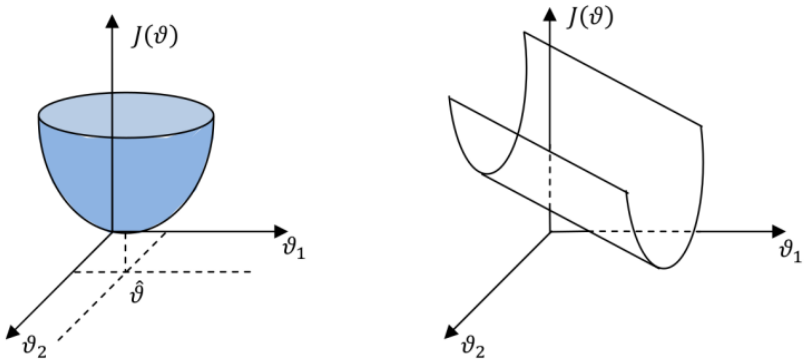
$$x^T \left(\frac{2}{N}\sum_{t=1}^{N}\phi(t)\phi(t)^T\right) x > 0 \quad \forall x \neq 0$$

This can be rewritten as:

$$\frac{2}{N}\sum_{t=1}^{N}(x^T\phi(t))^2 > 0$$

Which is always true for non-zero x, confirming that we can say that LS estimates converge to Δ (set of global minimum of $J(\theta)$):

1. $\mathbb{E}[\varphi(t)\varphi(t)^T]$ non-singular $\rightarrow \Delta = \{\theta^*\}$ and we have **identifiability** 2. $\mathbb{E}[\varphi(t)\varphi(t)^T]$ singular $\rightarrow J(\theta)$ has ∞ global minima, including θ^*



Actually if multiple global minimums are found, it's an error since it's known that only one system originated the data. So multiple minimums are not feasible solutions and this can happen because:

- **over**: the chosen models are too complex for the system
- **under**: data isn't representative enough

At the end we say that **identifiability** is when there is only **one** solution. Since AR has no MA part ($C(z) = 1$), explicit minimization is possible:

$$\hat{\vartheta}_N = \left(\sum_{t=1}^{N}\varphi(t)\varphi'(t)\right)^{-1}\sum_{t=1}^{N}y(t)\varphi'(t)$$

In the above formula, the crucial thing is that the matrix $\left[\sum_{t=1}^{N} \varphi(t)\varphi'(t)\right]$ is invertible: it must be positive semi-definite.

A necessary condition for the invertibility is that the input $u(t)$ is **persistently exciting** of order k which is a property which interest the $k \times k$ matrix:

$$\begin{vmatrix} \gamma_{uu}(0) & \gamma_{uu}(1) & \gamma_{uu}(2) & \cdots \\ \gamma_{uu}(1) & \gamma_{uu}(0) & \gamma_{uu}(1) & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{vmatrix}$$

This matrix must be invertible $\leftarrow$ is a necessary condition to make invertible $\bar{R}$! Note that a WN is **persistently exciting** of any order:

$$\begin{vmatrix} \gamma_{uu}(0) & \gamma_{uu}(1) & \gamma_{uu}(2) & \cdots \\ \gamma_{uu}(1) & \gamma_{uu}(0) & \gamma_{uu}(1) & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{vmatrix} = \begin{vmatrix} \lambda^2 & 0 & 0 & \cdots \\ 0 & \lambda^2 & 0 & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{vmatrix} = \lambda^2 I$$

5.2 ARMA/ARMAX model identification: Maximum Likelihood

TLDR: ARMA (or ARMAX) identification differs from AR (or ARX) because there is no more linearity in the parameters. Traditional methods like Least Squares cannot be directly applied, we must use an iterative numerical method such as gradient descent can be used with an initial estimate θ_1 and an update rule that computes the next guess for θ based on the previous iteration.

ARMA/ARMAX generic formula:

$$M(\theta) : y(t) = \frac{B(z)}{A(z)}u(t-d) + \frac{C(z)}{A(z)}e(t), \quad e(t) \sim WN(0, \lambda^2)$$

Where $A(z)$, $B(z)$, and $C(z)$ are polynomials in z^{-1} and $\theta = [a_1\, a_2 \ldots a_m\, b_0\, b_1 \ldots b_{p-1}, c-1\, c_2 \ldots c_n]^T$.

Applying the 1 step predictor, we will immediately notice that from this:

$$J_N(\theta) = \frac{1}{N}\sum_{t=1}^{N}(y(t) - \phi(t)^T\theta)^2$$

$$\varepsilon(t|t-1,\theta) = y(t) - \hat{y}(t|t-1,\theta) =$$

$$= y(t) - \frac{C(z) - A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d)$$

$$= \left[1 - \frac{C(z) - A(z)}{C(z)}\right]y(t) - \frac{B(z)}{C(z)}u(t-d)$$

$$= \frac{A(z)}{C(z)}y(t) - \frac{B(z)}{C(z)}u(t-d)$$

The prediction error has a nonlinear dependence on the coefficients of the $C(z)$ polynomial $(c_1, c_2, ..., c_n)$. This nonlinearity arises because $C(z)$ appears in the denominator of both terms. This means that the square of ϵ (J_n) depends quadratically to $C(z)$.

Because of this, ARMA/ARMAX models need iterative approaches with update rules based on gradient and Hessian information. This update/iterative rule comes in 3 main main flavors in this course:

- **Newton's rule**:

$$\theta^{(i+1)} = \theta^{(i)} - [\text{Hessian}]^{-1} \cdot \text{gradient}$$

- **Gradient descent**: gradient descent in neural networks is more effective since the are a lot of parameters.

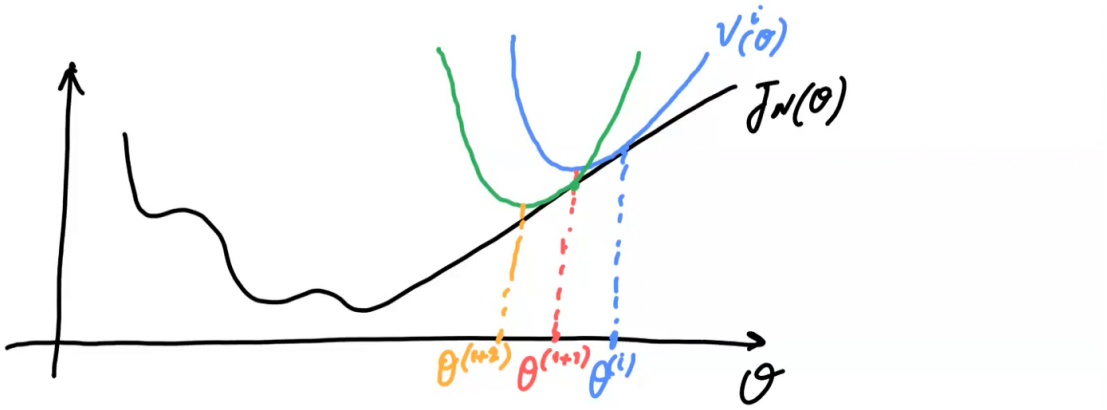
$$\theta^{(i+1)} = \theta^{(i)} - \eta \cdot \text{gradient}$$

- **Quasi-Newton's rule**: in the system identification framework, it's often used quasi-newton's rule.

$$\theta^{(i+1)} = \theta^{(i)} - [\frac{\text{aproximate}}{\text{Hessian}}]^{-1}\text{gradient}$$

5.2.1 Newton's rule

Newton method basically involves taking the quadratic expansion of the cost function and finding the minimum of the resulting paraboloid to update the parameter.



If you consider a quadratic approximation, the approximating function can be obtained by the Taylor development.

$$V(\theta) = \varepsilon(t|t-1,\theta)|_{\theta=\theta^{(i)}} + (\theta-\theta^{(i)})^\top \left.\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}\right|_{\theta=\theta^{(i)}} + \frac{1}{2}(\theta-\theta^{(i)})^\top \left.\frac{\partial^2 \varepsilon(t|t-1,\theta)}{\partial \theta^2}\right|_{\theta=\theta^{(i)}} (\theta-\theta^{(i)})$$

The optimal update rule use that quadratic approximation:

$$\theta^{(i+1)} = \theta^{(i)} - \left[\left.\frac{\partial^2 J_N(\theta)}{\partial \theta^2}\right|_{\theta=\theta^{(i)}}\right]^{-1} \left.\frac{\partial J_N(\theta)}{\partial \theta}\right|_{\theta=\theta^{(i)}}$$

So we need both the **gradient** and the **hessian**. The gradient of the cost function is given by:

$$\frac{\partial J_N(\theta)}{\partial \theta} = \frac{2}{N}\sum_{t=1}^{N} \varepsilon(t|t-1,\theta) \cdot \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}$$

The Hessian is computed (derivative of first derivative, and so a derivative of a product):

$$\frac{\partial^2 J_N(\theta)}{\partial \theta^2} = \frac{2}{N}\sum_{t=1}^{N} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}\left(\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}\right)^T + \frac{2}{N}\sum_{t=1}^{N} \varepsilon(t|t-1,\theta)\frac{\partial^2 \varepsilon(t|t-1,\theta)}{\partial \theta^2}$$

The second-order term is usually neglected for simplicity (it's a good approximation):

$$\frac{\partial^2 J_N(\theta)}{\partial \theta^2} \approx \frac{2}{N} \sum_{t=1}^{N} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \left(\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \right)^T$$

With this approximation we can say that the hessian is always positive and so it's useful so that we always have the "positive concavity"

"I'm sure to take the right paraboloid"

and the Newton's law becomes:

$$\theta^{(i+1)} = \theta^{(i)} - \left[\sum_{t=1}^{N} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}^T \right]^{-1} \left[\sum_{t=1}^{N} \varepsilon(t|t-1,\theta) \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \right]$$

Now that we have the expression we need to compute the protagonist $\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}$: the crucial thing here is that it's a vector of partial derivatives computed with respect to each parameter in $\theta = [a_1, a_2, ..., a_m, b_1, ..., c_1, ...]^T$

We need to compute derivative polynomials considering θ at step i. $\theta = [a_1, a_2, ...a_m, b_1, ..., c_1..., c_n]$.

3 auxiliary signals are defined to simplify computations:

$$\alpha(t) = -\frac{1}{C(z)} y(t)$$

$$\beta(t) = -\frac{1}{C(z)} u(t)$$

$$\gamma(t) = -\frac{1}{C(z)} \varepsilon(t|t-1,\theta)$$

All of these are time signals : the derivative of a signal respect to a parameter is another signal (time dependant). Note that $\gamma(t)$ depends by the current value of θ

So that the gradient of the prediction error with respect to the parameter vector θ can be expressed as:

$$\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} = [\alpha(t-1)\alpha(t-2)\dots\alpha(t-m)\beta(t-1)\dots\dots\gamma(t-1)\dots\gamma(t-m)]^T$$

After computing the signals $\alpha(t,\theta^{(i)}), \beta(t,\theta^{(i)}), \gamma(t,\theta^{(i)})$ and so the minimum we also need the hessian (which is a matrix).

$$\frac{\partial^2 \varepsilon(t|t-1,\theta)}{\partial \theta^2}$$

But actually using the "approximate hessian" we just do :

$$\frac{\partial^2 J_N(\theta)}{\partial \theta^2} \approx \frac{2}{N} \sum_{t=1}^{N} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \left(\frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \right)^T$$

(so no hessian). Skipping the endless computation, at the end the update rule:

$$\theta^{(i+1)} = \theta^{(i)} - \left[\sum_{t=1}^{N} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta} \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}^T\right]^{-1} \left[\sum_{t=1}^{N} \varepsilon(t|t-1,\theta) \frac{\partial \varepsilon(t|t-1,\theta)}{\partial \theta}\right]$$

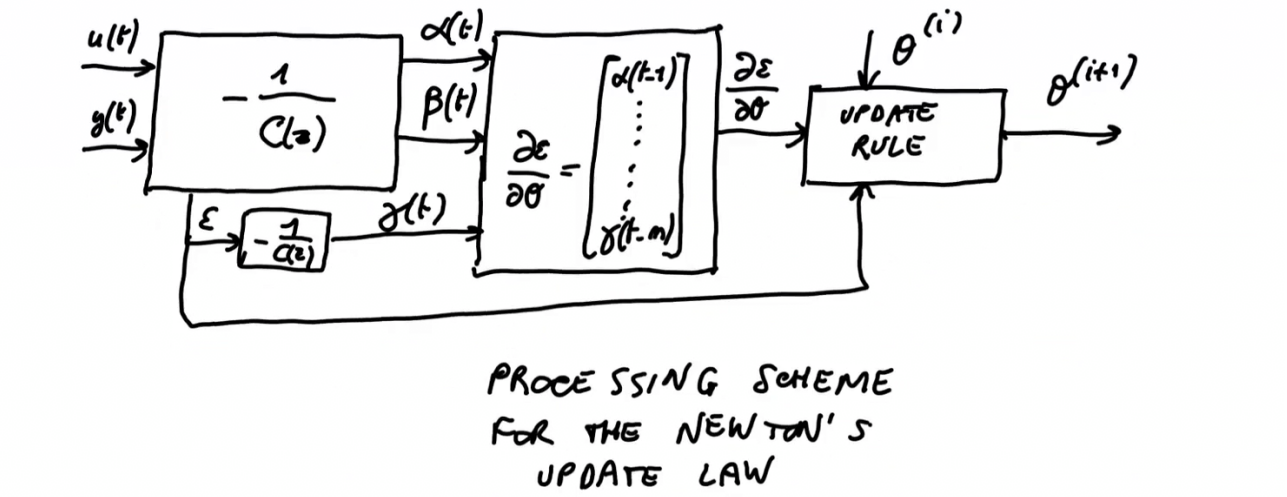
$$\vartheta^{(i+1)} = \vartheta^{(i)} - \left[\frac{1}{N}\sum_{t=1}^{N} \frac{\partial \varepsilon(t)}{\partial \theta} \left(\frac{\partial \varepsilon(t)}{\partial \theta}\right)^T\right]^{-1} \left[\frac{1}{N}\sum_{t=1}^{N} \frac{\partial \varepsilon(t)}{\partial \theta} \varepsilon_\vartheta(t)\right]$$

which is (recall):

$$\theta^{(i+1)} = \theta^{(i)} - [\text{Hessian}]^{-1} \cdot \text{gradient}$$

The iterative process continues until a convergence criterion is met, such as a small change in the parameter estimates or a maximum number of iterations is reached.

This approach allows for the identification of ARMA and ARMAX models, which are nonlinear in parameters, unlike the simpler AR and ARX models that can be solved using least squares.



PROCESSING SCHEME FOR THE NEWTON'S UPDATE LAW

5.3 Dummy identification exercise

In exam can happen:

$$y(t) = e(t) + \frac{1}{2}e(t-1)$$

where $e(t) \sim WN(0,1)$ which we would like to identify with a model M :

$$y(t) = \eta(t) + b\eta(t-1)$$

where $\eta(t) \sim WN(0,\lambda^2)$

We must not be a noobs and use the brain and recognize without moving the pen that with $b^* = \frac{1}{2}$ we have $\lambda^{2*} = 1$.

The equality of models is trivial:

$$\varepsilon(t|t-1) = \frac{A_m(z)}{C_m(z)}y(t) = \frac{1}{1+bz^{-1}}y(t) = \frac{1}{1+bz^{-1}}(1+\frac{1}{2}z^{-1})e(t)$$

$$\varepsilon(t|t-1) = \frac{1+\frac{1}{2}z^{-1}}{1+bz^{-1}}e(t)$$

For $\varepsilon(t)$ to be white noise, equal to $e(t)$ and have $\lambda^{2*} = 1$ we need:

$$b^* = \frac{1}{2}$$

5.4 Non-parametric identification

So far if we want to infer something about the underling stochastic process we need to learn the model first and then make a **parametric identification**.

Now let's try a non-parametric one directly estimating from data m_y, $\gamma_y(\tau)$ and $\Gamma_y(\omega)$ without first identifying a full model of $W(z)$.

But before diving in the estimators, we we need to define what we mean by "good estimator". We will use two definitions:

- **Correctness:** An estimator is considered correct (or unbiased) if its expected value is equal to the true parameter it is estimating, indicating that the mean of many independent estimates would converge to the true parameter. $E[\hat{\mu}_n] = \mu$ where μ is the real mean.
- **Consistency:** An estimator is consistent if the probability of the estimates being close to the true parameter increases as the sample size grows, meaning that with more data, the estimates become progressively more precise. $Var[\hat{s}_N] \to 0$ as $N \to \infty$.

5.4.1 Estimation of Mean

The most natural estimator is $\hat{\mu}_n$, where n refers to the number of samples.

$$\hat{\mu}_n = \frac{1}{N}\sum_{i}^{N} y(i)$$

This estimator grows more informative with additional data, capturing the dynamics of ARMA processes without relying on full historical memory.

5.4.2 Estimation of Covariance

$$\gamma_y(\tau) = \frac{1}{N-\tau}\sum_{t=1}^{N-\tau} y(t)y(t+\tau)$$

The estimators is consistent in case of stationary ARMA processes but it's only asymptotically correct.

5.4.3 Estimation of Spectrum

We can estimate this:

$$\Gamma_y(w) = \sum_{z=-\infty}^{\infty} \gamma(z)e^{-jwz}$$

with this:

$$\hat{\Gamma}_N(w) = \sum_{\tau=-(N-1)}^{N-1} \hat{\gamma}_N(\tau)e^{-jwz}$$

Estimating the spectrum $\hat{\Gamma}_N(\omega)$ is complex as it is derived from the estimated covariance function, making it an indirect estimation.

- This estimator is not initially correct but becomes asymptotically correct with a large dataset.
- Consistency is challenging as it's shown that the error variance does not tend to zero even with infinite samples:

$$\mathbb{E}[(\hat{\Gamma}_N(\omega) - \Gamma_\gamma(\omega))^2] \xrightarrow[n\to\infty]{} \Gamma(\omega)^2$$

The best we can do to address this issue is applying the so called Bartlett method: average the spectrum by dividing the dataset into r parts and computing the spectrum for each part:

$$\hat{\Gamma}^{(i)}(\omega), \quad i, ..., r$$

Then we average these estimators:

$$\bar{\hat{\Gamma}}(\omega) = \frac{1}{r}\sum_{i=1}^{r} \hat{\Gamma}_{\hat{N}}^{(i)}(\omega)$$

Under the assumption that the data of different sub-series are uncorrelated (hence the requirement $N \gg r$), the variance is approximately:

$$\mathrm{Var}[\bar{\hat{\Gamma}}(\omega)] \approx \frac{1}{r^2}\Gamma^2(\omega)$$

The uncertainty is now "significantly" reduced.

5.4.4 Data preprocessing

What if $y(t)$ is non-stationary? So these two are possible causes of non-stationarity that we address in this course:

- trend
- seasonality

To work with non-stationary processes, we first need to estimate possible trends or seasonalities. Then we can remove them and work with the reminder **SSP (Stationary Stochastic Process)**.

5.4.4.1 Trend removal

$$y(t) = \tilde{y}(t) + kt + m$$

So in order to work with $\tilde{y}(t)$ we first estimate k and m. Then we remove the trend from the data set. We can also say:

$$\mathbb{E}[y(t) - kt - m] = \mathbb{E}[\tilde{y}(t)] = 0$$

Inspired by the above equality, we can find $\hat{m}$ and $\hat{k}$ as the argument of the minimum with respect to m and k :

$$(\hat{m}, \hat{k}) = \mathrm{argmin}_{m,k} \frac{1}{N}\sum_{t=1}^{N} \left(y(t) - k(t-m)\right)^2$$

This expression is a classic least squares optimization, which aims to determine the "best fit" by finding the line (or trend) that best fits the data, where k can be thought of as a scaling factor and m as a time shift. As the LS cost function, the expression forms a paraboloid, which again has a single global minimum making possible to find the best linear trend estimate in the data.

5.4.4.2 Seasonality

$$y(t) = \tilde{y}(t) + s(t)$$

where $s(t) = s(t + k\mathbb{T})$ where $\mathbb{T}$ is the period. In the same way we need to estimate $s(t)$. The underlying idea is :

$$\hat{S}(t) = \frac{1}{M}\sum_{k=0}^{M-1} y(t + hT) = \frac{1}{M}\sum_{k=0}^{M-1} \widetilde{y}(t + hT) + \frac{1}{M}\sum_{k=0}^{M-1} S(t + hT)$$

But remember that we had also to estimate T period.

6 Model validation

Given an optimal model $M(\hat{\theta}_n)$ the goal is to validate its optimality also considering its model order n. Using measured samples, one can estimate the prediction error variance through $J_k(a)$, which evaluates the performance of the predictive model:

$$J_k(a) = \frac{1}{k}\sum_{t=0}^{t=k} \left(y(t+1) - \hat{y}(t+1|t)\right)^2$$

At the end we can say that even with measured samples (so finite data) **asymptotically** we will converge to the minimum of the asymptotic cost since:

$$J_N(\vartheta) \xrightarrow[N\to\infty]{} \bar{J}(\vartheta) = \mathbf{E}[\varepsilon_\vartheta(t)^2]$$

which implies that:

$$\min\{J_N(\vartheta)\} \xrightarrow[N\to\infty]{} \min\{\bar{J}(\vartheta)\}$$

6.1 PEM converges to real S

Now let's prove that if $S \in M(\theta)$ PEM guarantees that $M(\hat{\theta}_N) \to S$ or alternatively let's prove that $\theta^o \epsilon \Delta$.

$$\begin{aligned}
\varepsilon(t|t-1,\theta) &= y(t) - \hat{y}(t|t-1,\theta) \\
&= y(t) - \hat{y}(t|t-1,\theta^0) + \hat{y}(t|t-1,\theta^0) - \hat{y}(t|t-1,\theta) \\
\mathbb{E}[E(t|t-1,\theta)^2] = \mathbb{E}[(e(t) + \hat{y}(t|t-1,\theta^0) - \hat{y}(t|t-1,\theta))^2] &= \\
E[e(t)^2] + E[(\hat{y}(t|t-1,\theta^0) - \hat{y}(t|t-1,\theta))^2] + 0 & \\
= \lambda^2 + 0 &
\end{aligned}$$

Last passage is possible since $\theta = \theta^0$ in the case of the global optimum (as said above, PEM is guarantee to converge to the minimum of the asymptotic cost so $\theta = \theta^0$). This theorem is very, very significant: if the system that you want to model and learn from data is inside the model set, you will converge asymptotically (so with a large data set) to exactly that system.

$$\hat{\theta}_N \xrightarrow[N\to\infty]{} \theta^o$$

It means that the prediction error algorithm that we are using in this course is actually good, meaning that they will lead you to the right model (at least in this very ideal case where the system is in the model set).

Yeah indeed, it's very ideal that $S \in M(\theta)$. Let's summary the possible cases:

- $S \in M(\theta)$ and Δ is a singleton : for $n \to \infty$ $M(\hat{\theta}_n)$ will converge to S. The identification problem is properly defined, ensuring a singular, accurate solution that aligns with the system's dynamics.
- $S \in M(\theta)$ and Δ is not a singleton : for $n \to \infty$ $M(\hat{\theta}_n)$ will not necessary converge to S, but it's guarantee $\hat{\vartheta}_N$ tends to one of the values in Δ (multiple optimal minima subset). This most scenarios case, where multiple models equally represent the system.
- $S \notin M(\theta)$ and Δ is a singleton: for $n \to \infty$ $M(\hat{\theta}_n)$ will converge to $M(\theta_n^*)$. In this case the model with $\hat{\vartheta}_N$ is the best proxy of the true system in the selected family. This case is when there is an insufficiency of the model set to capture the system's complexity.
- $S \notin M(\theta)$ and Δ is not a singleton: for $n \to \infty$ $M(\hat{\theta}_n)$ no guarantees. In this case the model with $\hat{\vartheta}_N$ tends to one of the best proxies of the true system in the selected family.

6.2 Model order selection

How can we select the best value of n namely the best model complexity? Remember that $J_n^n(\theta)$ in function of n is always non increasing but it's not the optimal predictor!

So in general an $ARX(n, n)$ is a better fit to the data than an $ARX(n-1, n-1)$ model, but we can't simply continue to use more parameters, we would **overfit** it.

6.2.1 Whiteness test on residuals

If we simply compute the performance index for multiple increasing values of n we will see that is monotonically decreasing with n: we can use the whiteness test on residuals to understand when "it's enough" and that further decreasing will not reflect in better performance for new unseen data (overfit).

Actually this method is not robust and it's not used.

6.2.2 Cross-validation

In cross-validation, the dataset is partitioned into an **identification set** (or training), which is used to construct the model, and a **validation set**, which is utilized to assess the model's performance. The validation portion of the data is "reserved" and not used in the model-building phase, potentially leading to data "wastage".

6.2.3 Identification with model order penalties

For better model selection that help prevent over-fitting we can directly control model complexity.

6.2.3.1 FPE (Final Prediction Error) Identification with model order penalties:

$$FPE = \frac{N+n}{N-n} J_N(\hat{\theta}_N)^{(n)}$$

We are giving a penalty to the models with high complexity. The FPE functions is not monotonically decreasing, and the complexity corresponding to its minimum value can be chosen as complexity of the model.

6.2.3.2 AIC (Akaike Information Criteria)

$$AIC = \frac{2n}{N} + \ln\left(J_N(\hat{\theta}_N)^{(n)}\right)$$

For high values of N, this is equivalent to FPE.

6.2.3.3 MDL (Minimum Description Length) Asymptotically is similar to AIC but with higher penalization since $ln(N) > 2$.

$$MDL = \ln(N)\frac{n}{N} + \ln\left(J_N(\hat{\theta}_N)^{(n)}\right)$$

In general case $S \notin M(\theta)$ we usually prefer to (slightly) overfit, so generally AIC is preferred.