Parametric Methods

- Autoregressive (AR)
- Moving Average (MA)
- Autoregressive - Moving Average (ARMA)

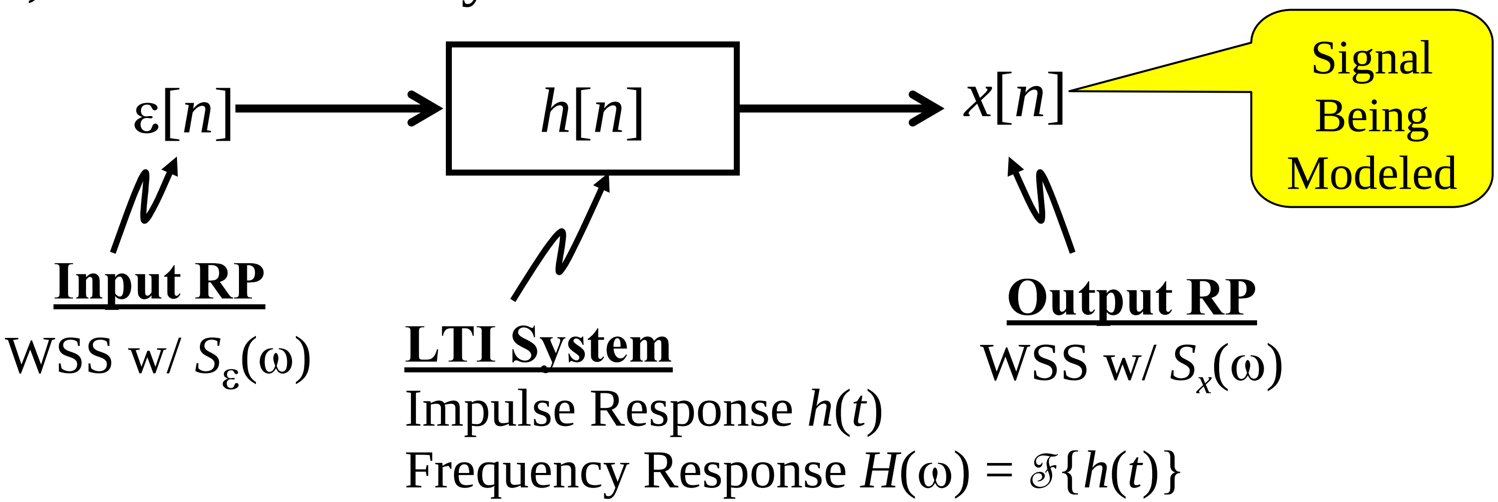
LO-2.5, P-13.3 to 13.4 (skip 13.4.3 – 13.4.5)

Time Series Models

"Time Series" = "DT Random Signal"

Motivation for Time Series Models

Recall the result we had that related output PSD to input PSD for a linear, time-invariant system:



$$\boxed{S_x(\omega) = |H(\omega)|^2 S_\varepsilon(\omega)}$$

If the input ε[n] is white with power σ^2 then: $\boxed{S_x(\omega) = |H(\omega)|^2 \sigma^2}$

Then… Shape of output PSD is <u>completely</u> set by $H(\omega)$!!!

Time Series Models (Parametric Models)

Thus, under this model… knowing the LTI system's transfer function (or frequency response) tells everything about the PSD.

The transfer function of an LTI system is completely determined by a set of parameters $\{b_k\}$ and $\{a_k\}$:

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{k=1}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

If (…if, if , if!!!) we *can* assure ourselves that the random processes we are to process can be **modeled** as the output of a LTI system driven by white noise, then….

"Estimating Parameters" = "Estimating PSD"

Note: We'll Limit Discussion to Real-Valued Processes

Parametric PSD Models

The most general parametric PSD model is then:

$$S_x(\omega) = \sigma^2 \frac{\left| 1 + \sum_{k=1}^{q} b_k e^{-j\omega k} \right|^2}{\left| 1 + \sum_{k=1}^{p} a_k e^{-j\omega k} \right|^2}$$

Model Parameters

$$\boxed{\sigma^2, \quad \{a_k\}_{k=1}^{p}, \quad \{b_k\}_{k=1}^{q}}$$

The output of the LTI system gives a time-domain model for the process:

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \sum_{k=0}^{q} b_k \varepsilon[n-k]$$

$$(b_0 = 1)$$

There are three special cases that are considered for these models:

- Autoregressive (AR)
- Moving Average (MA)
- Autoregressive Moving Average (ARMA)

Autoregressive (AR) PSD Models

If the LTI system's model is constrained to have only poles, then:

$$H(z) = \frac{1}{A(z)} = \frac{1}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

TF has only Poles

$$\underbrace{x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \varepsilon[n]}_{\text{Output depends “regressively” on itself}} \quad (b_0 = 1)$$

Order of the model is p: called AR(p) model

$$\boxed{S_{AR}(\omega) = \frac{\sigma^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-j\omega k}\right|^2}}$$

Poles Give Rise to PSD Spikes

Examples: **LO Fig. 2.11 & Fig. 2.12**

Moving Average (MA) PSD Models

If the LTI system's model is constrained to have only zeros, then:

$$H(z) = B(z) = 1 + \sum_{k=1}^{q} b_k z^{-k}$$

TF has only Zeros

$$\underbrace{x[n] = -\sum_{k=0}^{q} b_k \varepsilon[n-k],}_{} \quad b_0 = 1$$

Output is an "<u>average</u>" of values inside a <u>moving</u> window

Order of the model is q: called <u>MA(q) model</u>

$$\boxed{S_{MA}(\omega) = \sigma^2 \left| 1 + \sum_{k=1}^{q} b_k e^{-j\omega k} \right|^2}$$

Zeros Give Rise to PSD Nulls

Examples: **LO Fig. 2.13 & Fig. 2.14**

Autoregressive Moving Average (ARMA)

If the LTI system's model is allowed to have Poles & Zeros, then:

$$H(z) = \frac{B(z)}{A(z)} = \frac{1 + \sum_{k=1}^{q} b_k z^{-k}}{1 + \sum_{k=1}^{p} a_k z^{-k}}$$

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \sum_{k=0}^{q} b_k \varepsilon[n-k]$$

$$(b_0 = 1)$$

Order of the model is *p*,*q* : called ARMA(*p*,*q*) model

$$\boxed{S_x(\omega) = \sigma^2 \frac{\left|1 + \sum_{k=1}^{q} b_k e^{-j\omega k}\right|^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-j\omega k}\right|^2}}$$

Poles & Zeros
Give Rise to PSD
Spikes & Nulls

ACF Model of a Process

So far we've seen relationships between:

- PSD Model
- Time-Domain Model

These models impart a corresponding model to the ACF:

Let the process obey an ARMA(p,q) model

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \sum_{k=0}^{q} b_k \varepsilon[n-k]$$

To get ACF: multiply both sides of this by $x[n-k]$ & take E{}:

$$E\{x[n]x[n-k]\} = -\sum_{l=1}^{p} a_l E\{x[n-l]x[n-k]\} + \sum_{l=0}^{q} b_l E\{\varepsilon[n-l]x[n-k]\}$$

$$\Rightarrow \quad r_x[k] = -\sum_{l=1}^{p} a_l r_x[k-l] + \sum_{l=0}^{q} b_l r_{x\varepsilon}[k-l]$$

Need This!

ACF Model of a Process (cont.)

To evaluate this – write $x[n]$ as output of filter with input $\varepsilon[n]$:

$$
\begin{aligned}
r_{x\varepsilon}[k] &= E\{x[n]\varepsilon[n+k]\} \\
&= E\left\{\varepsilon[n+k]\sum_{l=-\infty}^{\infty} h[n-l]\varepsilon[l]\right\} \\
&= \sum_{l=-\infty}^{\infty} h[n-l]E\{\varepsilon[n+k]\varepsilon[l]\} \\
&= \sum_{l=-\infty}^{\infty} h[n-l]\sigma^2\delta[n+k-l] \\
&= h[-k]\sigma^2
\end{aligned}
$$

We have assumed a causal filter for a model:

$$\boxed{r_{x\varepsilon}[k] = 0 \quad k > 0}$$

ACF Model of a Process (cont.)

Using this result gives the **Yule-Walker Equations for ARMA**:

$$r_x[k] = \begin{cases} -\displaystyle\sum_{l=1}^{p} a_l r_x[k-l] + \sigma^2 \sum_{l=0}^{q} b_{l+k} h[l] & k = 0, 1, \ldots, q \\ -\displaystyle\sum_{l=1}^{p} a_l r_x[k-l] & k \geq q+1 \end{cases}$$

(ARMA)

These equations are the key to estimating the model parameters!!!

We now look at simplifications of these for the AR & MA cases.

ACF Model for an AR Process

Specializing to the AR case, we set $q = 0$ and get:

$$r_x[k] = \begin{cases} -\sum_{l=1}^{p} a_l r_x[k-l] + \sigma^2 h[0] & k = 0 \\ -\sum_{l=1}^{p} a_l r_x[k-l] & k \geq 1 \end{cases}$$

Now, we see that

$$\underbrace{h[0] = \lim_{z\to\infty} H(z)} = \lim_{z\to\infty} \frac{1+\sum_{k=1}^{q} b_k z^{-k}}{1+\sum_{k=1}^{p} a_k z^{-k}} = 1$$

"Initial Value Theorem" for Z-Transform

Yule-Walker Equations (AR)

$$\boxed{r_x[k] = \begin{cases} -\sum_{l=1}^{p} a_l r_x[k-l] + \sigma^2 & k = 0 \\ -\sum_{l=1}^{p} a_l r_x[k-l] & k \geq 1 \end{cases}}$$

ACF Model for an AR Process (cont.)

If we look at $k = 0, 1, \ldots p$ for these AR Yule-Walker equations, we get p+1 simultaneous equations that can be solved for the p+1 model parameters of $\{a_i\}_{i=1,\ldots,p}$ and σ^2:

Yule-Walker Equations (AR)

$$\begin{bmatrix} r_x[0] & r_x[1] & \cdots & r_x[p-1] \\ r_x[1] & r_x[0] & \cdots & \vdots \\ \vdots & \cdots & \ddots & r_x[1] \\ r_x[p-1] & \cdots & r_x[1] & r_x[0] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = -\begin{bmatrix} r_x[1] \\ r_x[2] \\ \vdots \\ r_x[p] \end{bmatrix}$$

$$\sigma^2 = r_x[0] + \sum_{l=1}^{p} a_l r_x[l]$$

If we know the $p \times p$ AC Matrix, then we can solve these equations for the model parameters!!!

ACF Model for an MA Process

Specializing to the MA case, we set p = 0 and get:

$$r_x[k] = \begin{cases} \sigma^2 \sum_{l=0}^{q-k} b_{l+k} h[l] & k = 0, 1, \ldots, q \\ 0 & k \geq q+1 \end{cases}$$

But… for the MA case the system is a FIR filter and we have

$$h[k] = \begin{cases} b_k, & k = 0, 1, \cdots, q \\ 0, & otherwise \end{cases}$$

Yule-Walker Equations (MA)

$$\boxed{r_x[k] = \begin{cases} \sigma^2 \sum_{l=0}^{q-|k|} b_{l+k} b_l & |k| = 0, 1, \ldots, q \\ 0 & |k| \geq q+1 \end{cases}}$$

Parametric PSD Estimation

As mentioned above, the idea here is to find a good estimate of the model parameters and then use those to get an estimate of the PSD. The basic idea holds regardless if it is ARMA, AR, or MA.

However, the derivation of the parameter estimates is quite hard for the ARMA and MA cases. So… we consider only the AR case – but even there we rely on intuition to some degree.

There has been a HUGE amount of research on how to estimate the AR model parameters. EE522 discusses this to some extent; here we simply state a few particular methods.

Parametric PSD Estimation (cont.)

Here is the general AR method: Given data $\{x[n],\ 0 \le n \le N\text{-}1\}$

1. Estimate the $p \times p$ AC Matrix from the data:

$$\{x[n], 0 \le n \le N-1\} \quad \Rightarrow \quad \{\hat{r}[k], 0 \le k \le p\}$$

2. Solve the AR Yule-Walker Equations for the AR Model

$$\begin{bmatrix} \hat{r}_x[0] & \hat{r}_x[1] & \cdots & \hat{r}_x[p-1] \\ \hat{r}_x[1] & \hat{r}_x[0] & \cdots & \vdots \\ \vdots & \cdots & \ddots & \hat{r}_x[1] \\ \hat{r}_x[p-1] & \cdots & \hat{r}_x[1] & \hat{r}_x[0] \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} = - \begin{bmatrix} \hat{r}_x[1] \\ \hat{r}_x[2] \\ \vdots \\ \hat{r}_x[p] \end{bmatrix}$$

$$\hat{\sigma}^2 = \hat{r}_x[0] + \sum_{l=1}^{p} \hat{a}_l \hat{r}_x[l]$$

3. Compute the PSD estimate from the model

$$\boxed{\hat{S}_{AR}(\omega) = \frac{\hat{\sigma}^2}{\left| 1 + \sum_{k=1}^{p} \hat{a}_k e^{-j\omega k} \right|^2}}$$

Parametric PSD Estimation – AR Case (cont.)

Two common methods (but there are many others):

"Autocorrelation" Method

Estimate the ACF using: $\hat{r}_x[k] = \frac{1}{N} \sum_{i=0}^{N-1-k} x[i]x[i+|k|], \quad 0 \le k \le p$

"Covariance" Method

Estimate using: $\hat{c}_{jk} = \frac{1}{N-p} \sum_{n=p}^{N-1} x[n-j]x[n-k], \quad 0 \le j,k \le p$

Solve Using:

$$\begin{bmatrix} \hat{c}_{11} & \hat{c}_{12} & \cdots & \hat{c}_{1p} \\ \hat{c}_{21} & \hat{c}_{22} & \cdots & \hat{c}_{2p} \\ \vdots & \cdots & \ddots & \vdots \\ \hat{c}_{p1} & \hat{c}_{p2} & \cdots & \hat{c}_{pp} \end{bmatrix} \begin{bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \vdots \\ \hat{a}_p \end{bmatrix} = - \begin{bmatrix} \hat{c}_{10} \\ \hat{c}_{20} \\ \vdots \\ \hat{c}_{p0} \end{bmatrix}$$

$$\hat{\sigma}^2 = \hat{c}_{00} + \sum_{l=1}^{p} \hat{a}_l \hat{c}_{0l}$$

Least Squares Method & Linear Prediction

There is another method that is often used that comes at the problem from a little different direction.

Recall: The above idea was based on the Yule-Walker equations, which are in terms of the ACF (which is unknown in practice!!)

➔ Thus we need to estimate the ACF to use this view

Least Squares provides a different way to estimate the AR parameters.

Recall: The output of an AR model is given by

$$\varepsilon[n] \longrightarrow \boxed{\frac{1}{1+\sum_{k=1}^{p} a_k z^{-k}}} \longrightarrow x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \varepsilon[n]$$

LS Method & Linear Prediction (cont.)

If we re-arrange this output equation we get:

$$x[n] - \underbrace{\left[-\sum_{k=1}^{p} a_k x[n-k] \right]}_{\hat{x}[n]} = \varepsilon[n]$$

Prediction Error

Prediction of $x[n]$

There are lots of applications where linear prediction is used:

- Data Compression
- Noise Cancellation
- Target Tracking
- Etc.

Goal: Find a set of prediction coefficients $\{a_k\}$ such that the $\underbrace{\text{sum of squares}}_{\text{Least Squares!!!}}$ of the prediction error is minimized

$$\boxed{\text{minimize} \quad V = \frac{1}{N\text{-}p} \sum_{n=p}^{N-1} \varepsilon^2[n]}$$

LS Method & Linear Prediction (cont.)

To choose the $\{a_k\}$ to minimize V we differentiate and set = 0

$$\frac{\partial V}{\partial a_l} = \frac{2}{N-p} \sum_{n=p}^{N-1} \frac{\partial \varepsilon[n]}{\partial a_l} \varepsilon[n] = \frac{2}{N-p} \sum_{n=p}^{N-1} x[n-l]\varepsilon[n]$$

Now we use:

$$\varepsilon[n] = x[n] - \underbrace{\left[-\sum_{k=1}^{p} a_k x[n-k] \right]}_{\hat{x}[n]} = \sum_{k=0}^{p} a_k x[n-k]; \quad a_0 = 1$$

$$\frac{\partial V}{\partial a_l} = \frac{2}{N-p} \sum_{n=p}^{N-1} x[n-l] \left[\sum_{k=0}^{p} a_k x[n-k] \right]$$

$$= \frac{2}{N-p} \sum_{k=0}^{p} a_k \sum_{n=p}^{N-1} x[n-l]x[n-k] = 0; \quad 1 \le l \le p$$

LS Method & Linear Prediction (cont.)

So to solve the LS Linear Prediction problem we need:

$$\frac{2}{N-p}\sum_{k=0}^{p} a_k \sum_{n=p}^{N-1} x[n-l]x[n-k] = 0; \quad 1 \le l \le p \qquad (\star)$$

Define:

1. Matrix $\boldsymbol{\Gamma}$ with elements λ_{lk}
2. Vector $\boldsymbol{\lambda}$ with elements λ_{l0}
3. Vector $\mathbf{a}$ with elements $a_1, \ldots, a_p$

where

$$\lambda_{lk} = \frac{1}{N-p}\sum_{n=p}^{N-1} x[n-l]x[n-k]; \quad 1 \le l,k \le p$$

Then $(\star)$ can be written as (exploiting that $a_0 = 1$):

$$\boldsymbol{\Gamma}\mathbf{a} + \boldsymbol{\lambda} = \mathbf{0} \quad \Rightarrow \quad \mathbf{a} = -\boldsymbol{\Gamma}^{-1}\boldsymbol{\lambda}$$