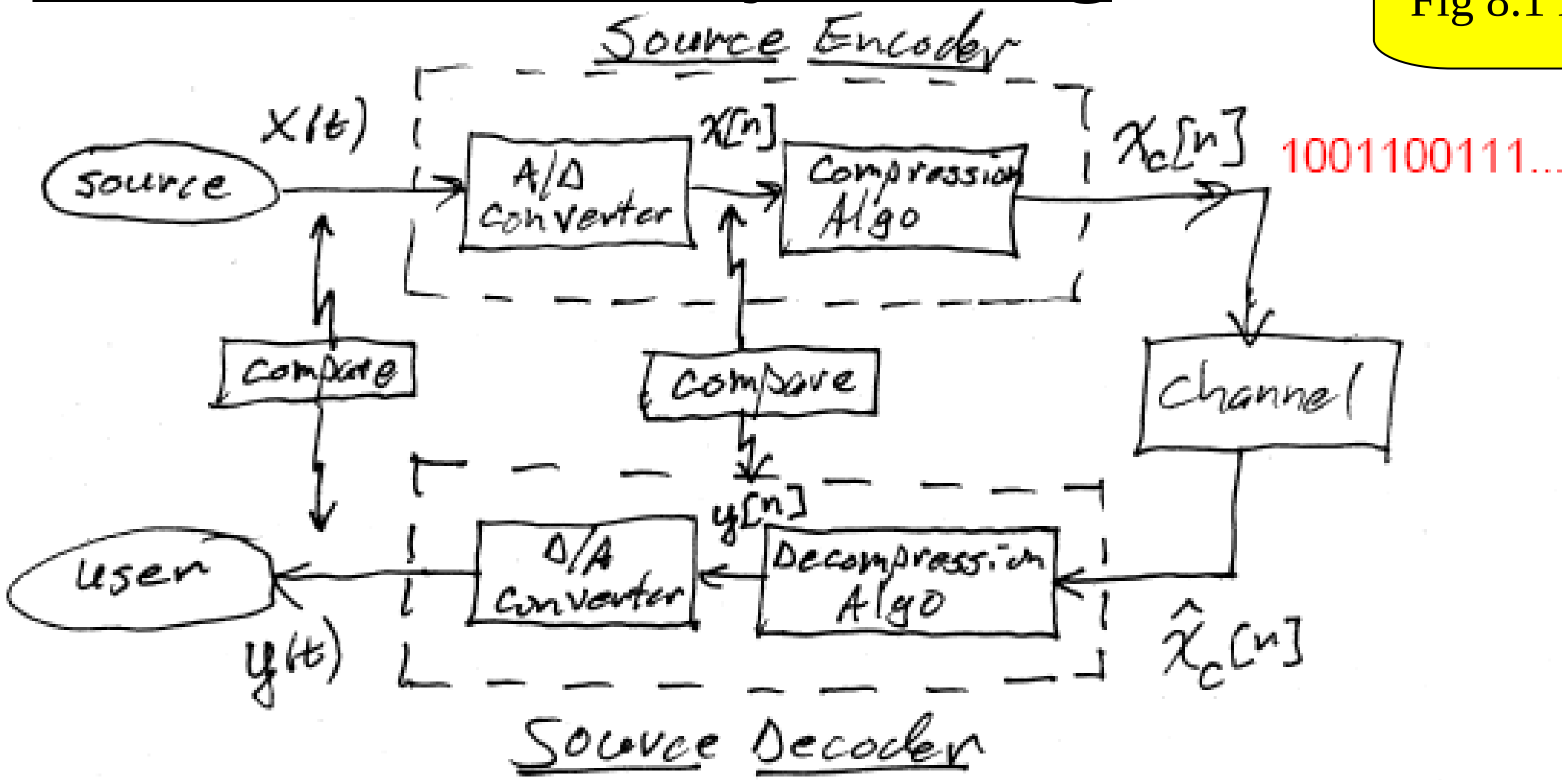
Ch. 8 Math Preliminaries for Lossy Coding

Overview of Lossy Coding

Slight variation on Fig 8.1 in textbook:



- Source signal shown is function of time *t*
  - Speech, Music, Etc.
- Source signal could be function of space *x*,*y*
  - Images
- … Or could be a function of space & time
  - Video

Sometimes compare $x(t)$ to $y(t)$

Nowadays generally focus on comparing $x[n]$ to $y[n]$

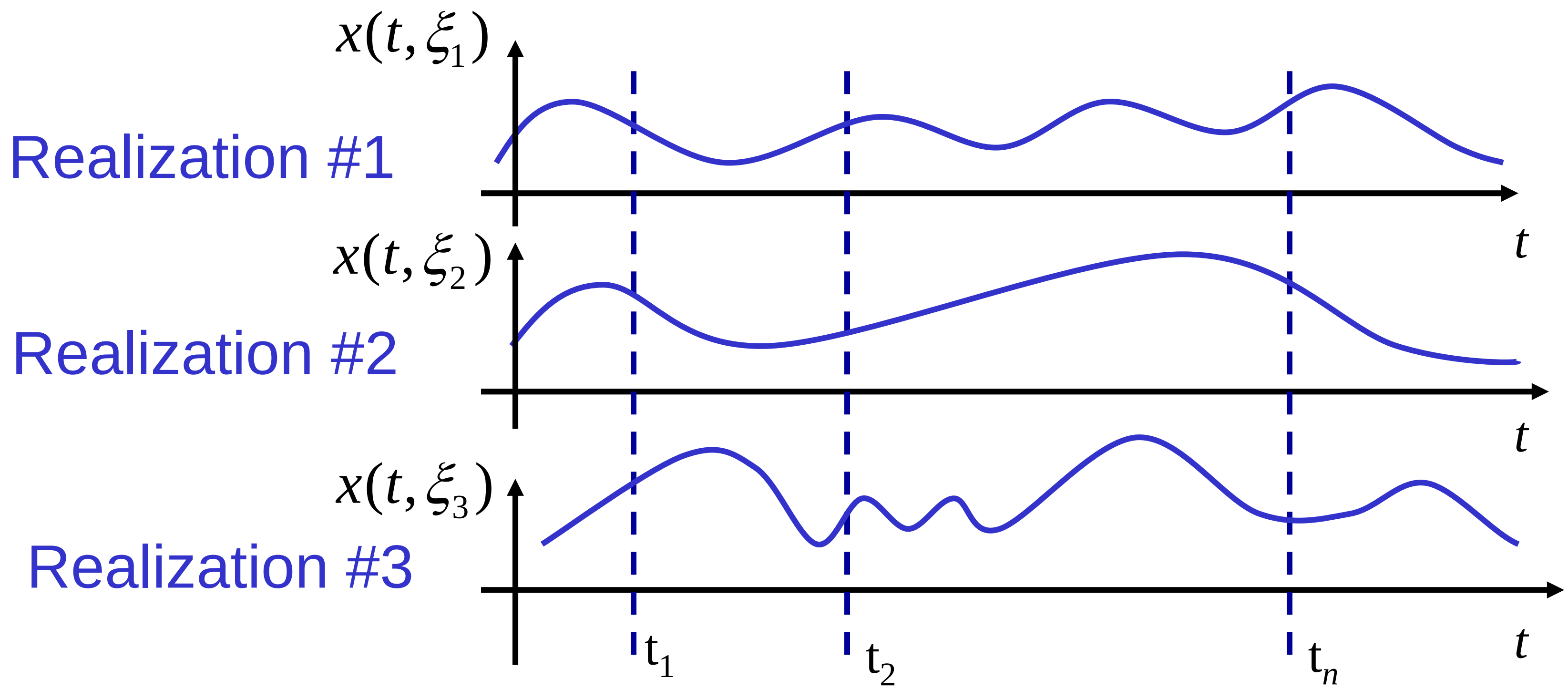
General Goal of Lossy Compression

- Make $y[n]$ (result of compressing then decompressing) as close to the original signal $x[n]$
- While using the smallest possible # of bits to represent $x_c[n]$
- We'll need probabilistic models for the signals $x(t)$ (or $x[n]$) and $y(t)$ (or $y[n]$)
- Model as random processes that take values over a continuum
  - $x(t)$ and $y(t)$ are CT random processes
  - $x[n]$ and $y[n]$ are DT random processes

Need a Prob. Density Function (PDF)

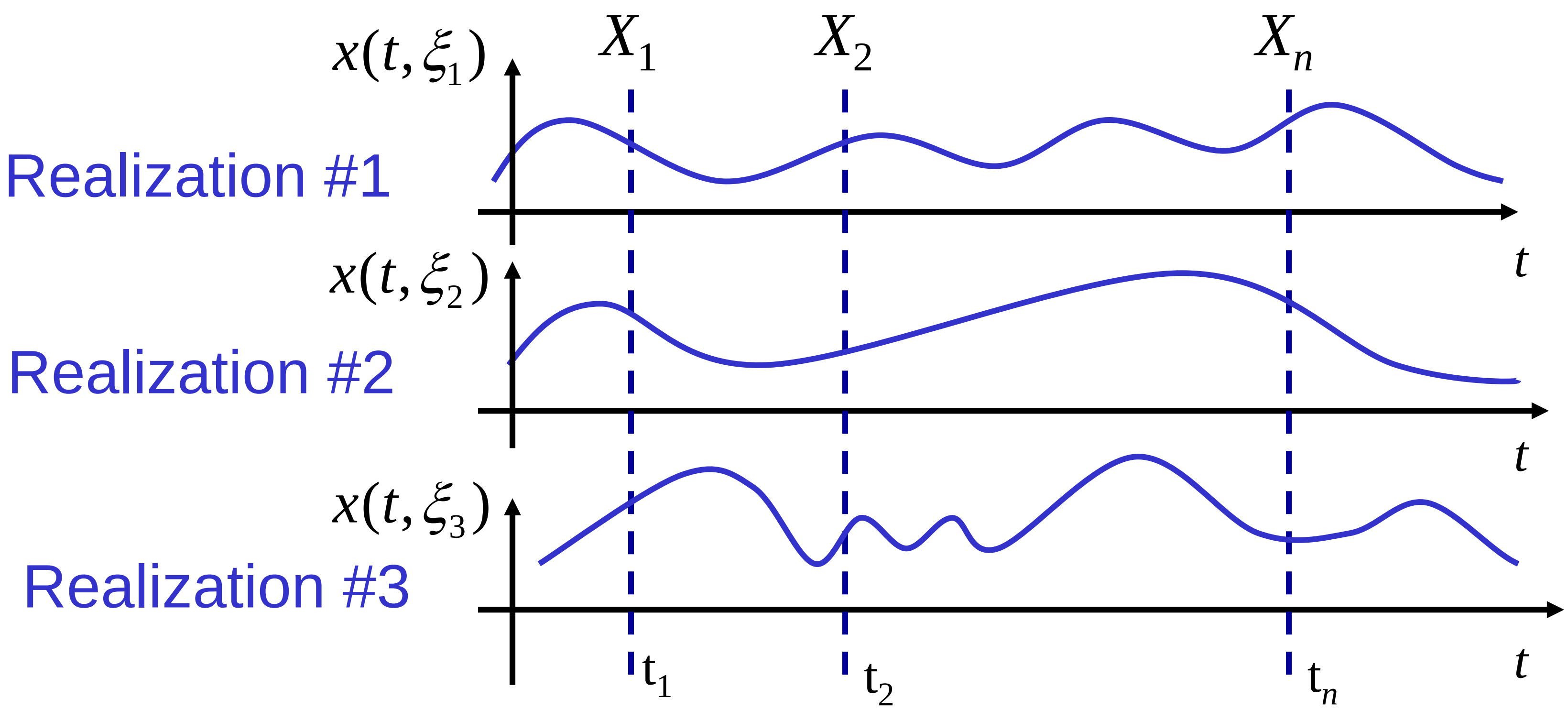
Random Processes: Collection of Functions

- Just as an RV is viewed as a collection of values that occur with a specified probability….
- A random process is viewed as a collection of functions that occur with specified probability.
  - The collection is called the "Ensemble"
  - Each function in the collection is called a "Realization"



Random Processes: Sequence of RVs

- At each time, say t_i, the RP is an RV $X_i = x(t_i,\xi)$
- In general, X_i is a continuous RV so we need a PDF
- In general, this PDF depends on time t_i: $f_X(x,t)$ **1st Order PDF**
- To give some complete probabilistic characterization of an RP we need joint PDFs $f_X(x_1, x_1, \ldots, x_n, t_1, t_1, \ldots, t_n)$ ***n*th Order PDF**



Wide Sense Stationary (WSS) Processes

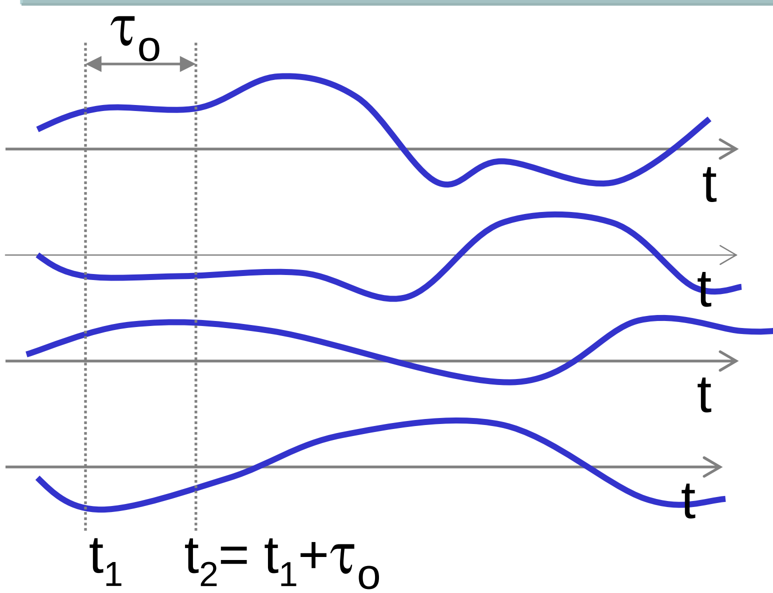
- We will limit ourselves to WSS processes and will only make use of the 1st order PDF and the autocorrelation function (ACF)
  - Or equivalently, the Power Spectral Density (PSD)
- The ACF is $E\{x(t)x(t+\tau)\}$ and in general depends on both t & τ
  - But for WSS the ACF depends only the τ: $R(\tau) = E\{x(t)x(t+\tau)\}$

$R_X(\tau) =$

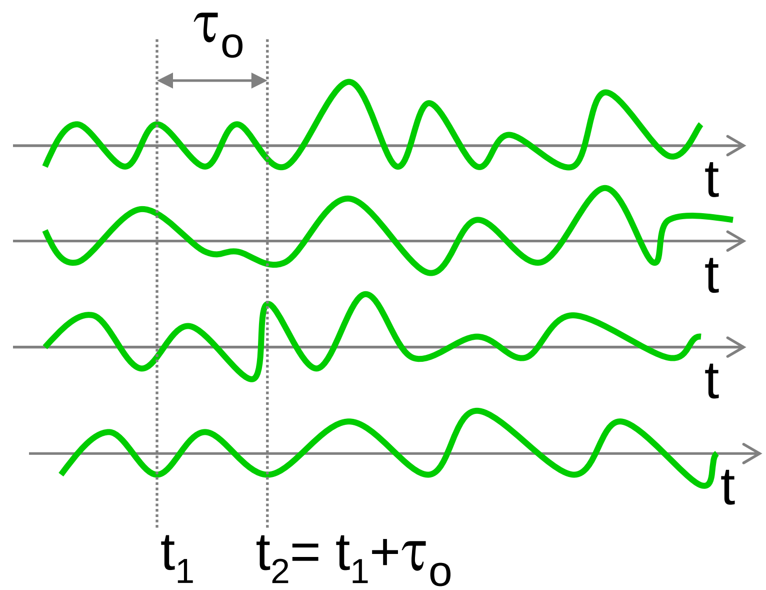
- **Positive** value if — **$x(t)$ & $x(t+\tau)$ are highly likely to have the <u>same sign</u>**
- **Negative** value if — **$x(t)$ & $x(t+\tau)$ are highly likely to have <u>opposite signs</u>**
- **Near Zero** if — **Product $x(t)x(t+\tau)$ is ≈ equally likely pos. or neg.**

- A WSS process must have these two properties
  - Its ACF depends only on τ
  - Its mean is constant
- $R_X(0) = E\{x^2(t)\} =$ constant
- Variance = Power: $\sigma^2 = E\{[x(t) - \text{Mean}_x]^2\}$

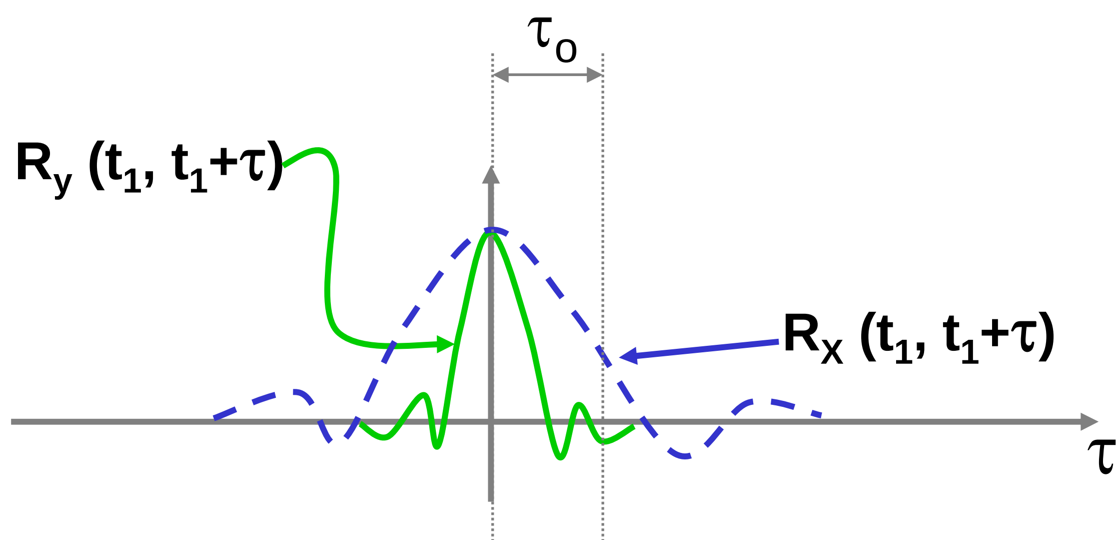
Four Realizations of x(t)



Four Realizations of y(t)



Note: Both x(t) and y(t) have the same 1st Order PDF… Yet they have VERY different ACFs



Ex. #1: D-T White Noise

Let x[k] be a sequence of RV's where…
each RV x[k] in the sequence is
uncorrelated with all the others:

$$E\{ x[k]\, x[m] \} = 0 \qquad \text{for } k \neq m$$

This DEFINES a DT White Noise
Also called "Uncorrelated Process"

Physically, uncorrelated means that knowing x[k] provides no insight into what value x[m] (for m≠k) will be likely to take (roll a die; the value you get provides no insight into what you expect to get on any future roll)

Ex. #1: D-T White Noise

<u>TASK :</u> We have a model…. Find the mean, ACF, and check if WSS (also find variance of process)

MEAN of Process : $E\{x[k]\} = 0$ **CONSTANT**

By definition !

ACF: $R_X(k_1,k_2)=E\{x[k_1] \cdot x[k_2]\}$

By our definition of white noise,
….this is 0 if $k_1 \neq k_2$

Ex. #1: D-T White Noise

Now for $k_1=k_2=k$:

$$R_x(k_1,k_2)\Big|_{k=k_1=k_2} = E\left\{x^2[k]\right\} = \sigma^2$$

By definition of variance for zero-mean case

Thus,

$$R_x(k_1,k_2) = \begin{cases} \sigma^2, & k_1 = k_2 \\ 0, & k_1 \neq k_2 \end{cases}$$

$$= \sigma^2 \delta[\underbrace{k_1 - k_2}]$$

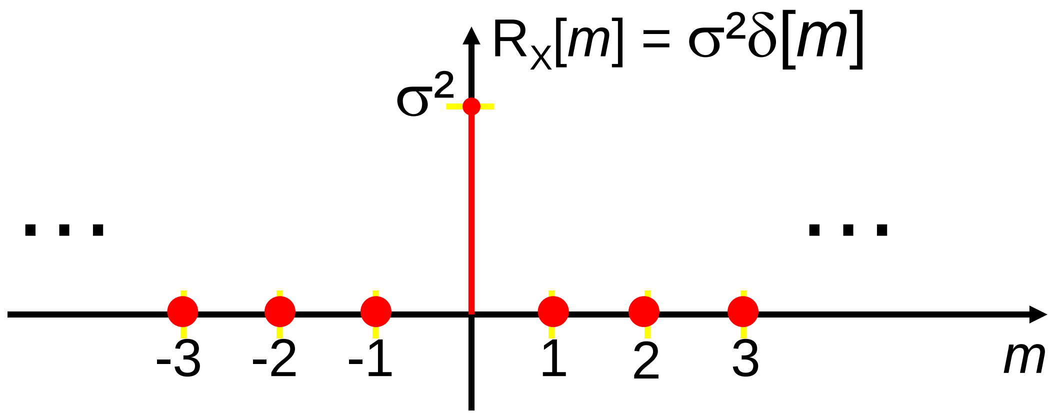
$= m$
(like τ for cont-time ACF)

ACF for DT White RP

★★ $\Rightarrow R_x(m) = \sigma^2 \delta[m]$ ★★

Ex. #1: D-T White Noise

ACF displays lack of correlation between any pair of any time instants:



Now since we have constant mean and ACF depends only on $m = k_2 - k_1$ $\Rightarrow$ **WSS**

Ex. #1: D-T White Noise

<u>Variance</u>

$$\sigma_x^2 = R_x[0] - \bar{x}^2$$

=0

$$= R_x[0]$$

$$= \sigma^2$$

For this case:
Variance of the **<u>Process</u>** = Variance of the **<u>RV</u>**

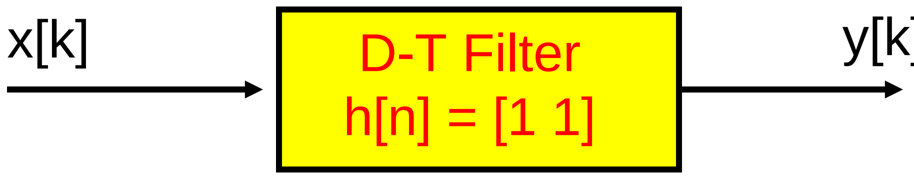
Ex. #2: Filtered D-T RP

Start with White RP x[k] in previous example

Recall : Zero Mean Process

$$R_X[m] = \sigma^2 \delta[m]$$

$\Rightarrow$ **WSS**



$y[k] = \underbrace{x[k] + x[k-1]}$

Two–Tap FIR filter
Taps = [1 1]

Ex. #2: Filtered D-T RP

TASK: Is y[k] WSS?

$\Rightarrow$ need to find mean & ACF

MEAN: Using filter output expressions gives

$$E\{y[k]\} = E\{x[k] + x[k-1]\}$$
$$= E\{x[k]\} + E\{x[k-1]\}$$

$\Rightarrow$ E {y[k]} = 0

Ex. #2: Filtered D-T RP

ACF :

Plug in Eq. for output

$$R_y(k_1,k_2) = E\{y[k_1]y[k_2]\}$$

$$= E\{(x[k_1]+x[k_1-1])(x[k_2]+x[k_2-1])\}$$

$$= \underbrace{E\{x[k_1]x[k_2]\}}_{R_x(k_2-k_1)} + \underbrace{E\{x[k_1]x[k_2-1]\}}_{R_x(k_2-k_1-1)}$$

$$+ \underbrace{E\{x[k_1-1]x[k_2]\}}_{R_x(k_2-k_1+1)} + \underbrace{E\{x[k_1-1]x[k_2-1]\}}_{R_x((k_2-1)-(k_1-1))}$$

Ex. #2: Filtered D-T RP

$$R_y(k_1,k_2) = \underbrace{R_x(k_2 - k_1)}_{\sigma^2\delta[m]} + \underbrace{R_x(k_2 - k_1 - 1)}_{\sigma^2\delta[m-1]}$$

$$+ \underbrace{R_x(k_2 - k_1 + 1)}_{\sigma^2\delta[m+1]} + \underbrace{R_x\big((k_2 - 1) - (k_1 - 1)\big)}_{\sigma^2\delta[m]}$$

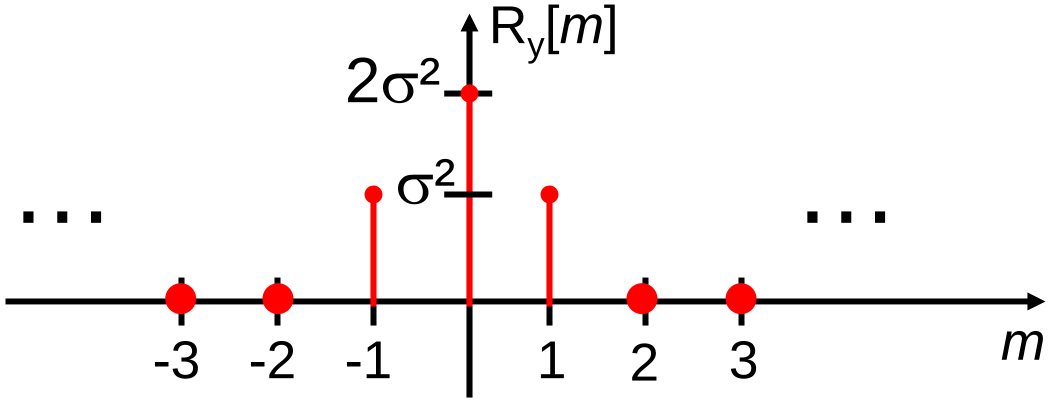
$$\text{where} \quad m = k_2 - k_1$$

y[k] is WSS

ACF for 2-Tap Filtered White RP

☆☆ $\Rightarrow R_Y(m) = \sigma^2[2\delta[m] + \delta[m-1] + \delta[m+1]]$ ☆☆

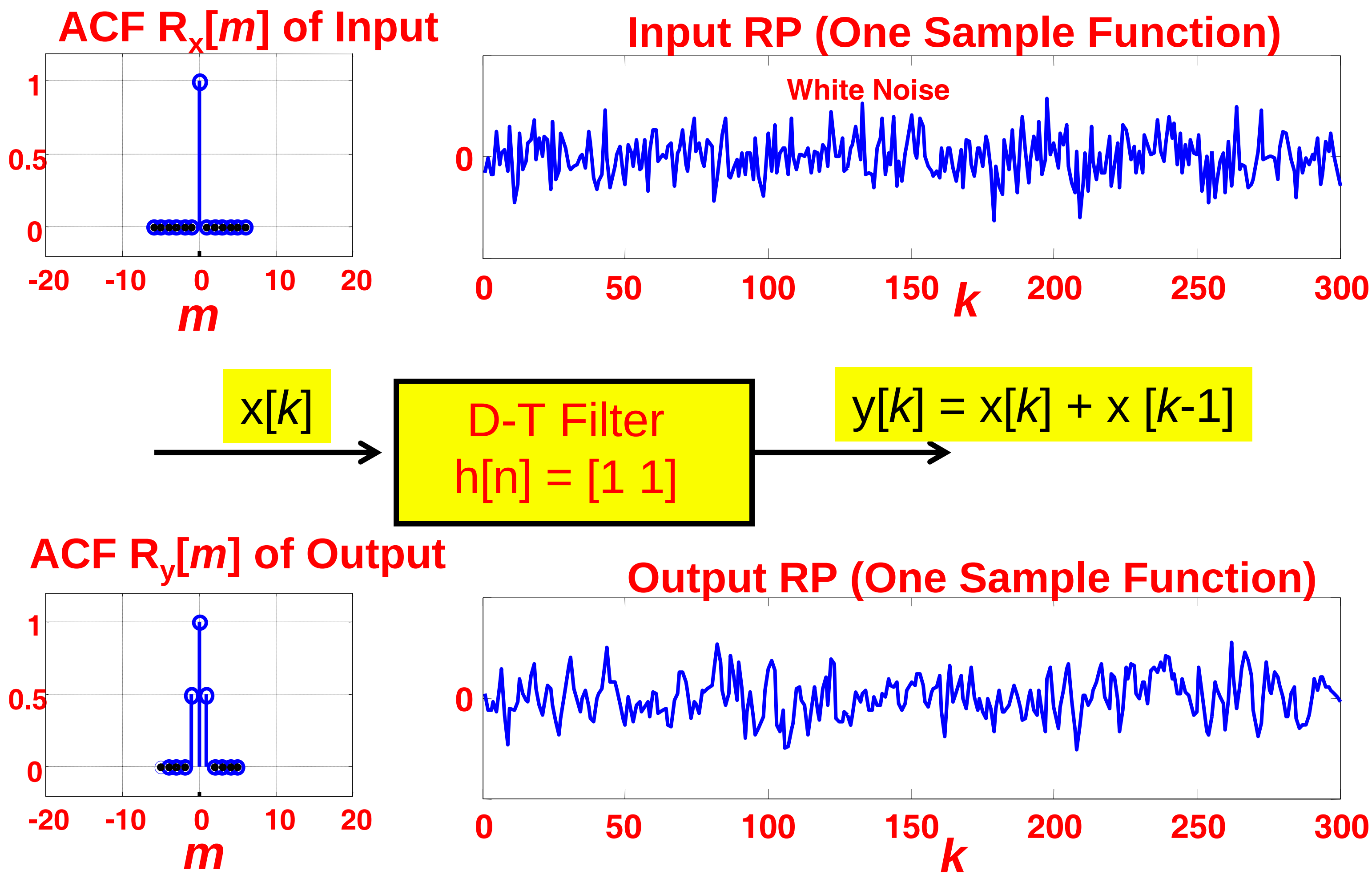
Ex. #2: Filtered D-T RP



Note : Filter introduces correlation between adjacent samples - but still no correlation for samples 2 or more samples apart (for this filter)

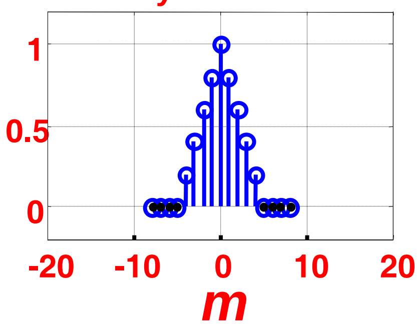
Big Picture: Filtered RP

Filters can be used to change the correlation structure of a RP:

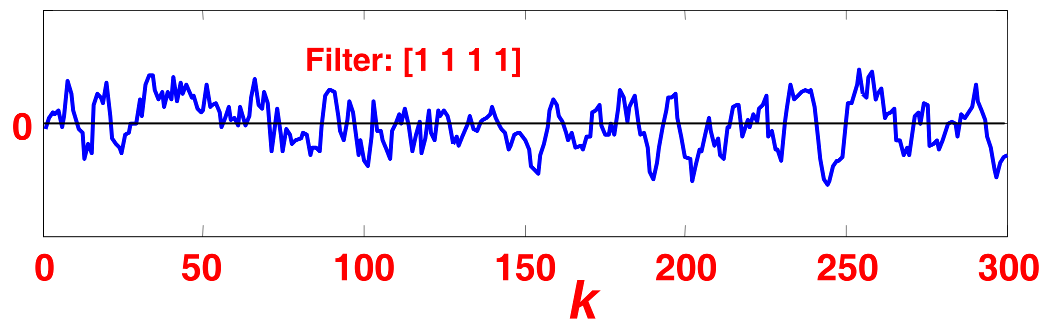


Big Picture: Filtered RP (cont)

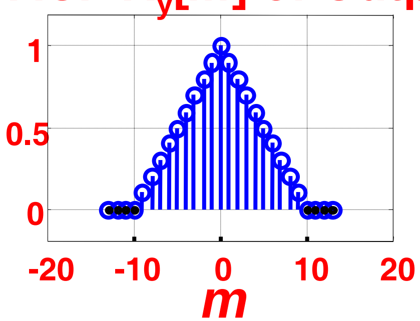
ACF $R_y[m]$ of Output



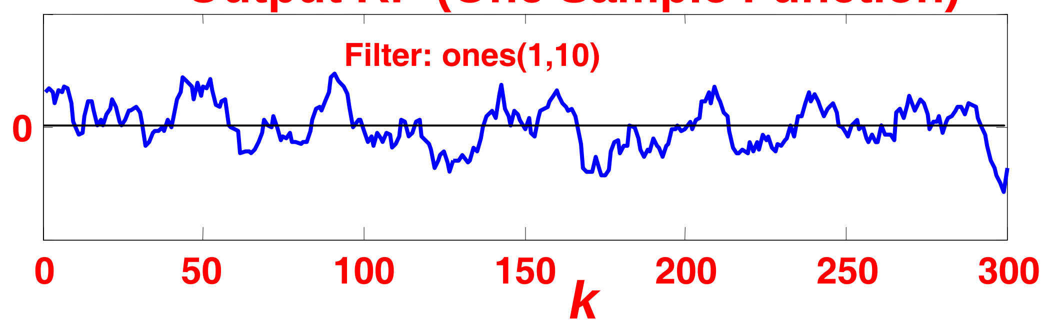
Output RP (One Sample Function)



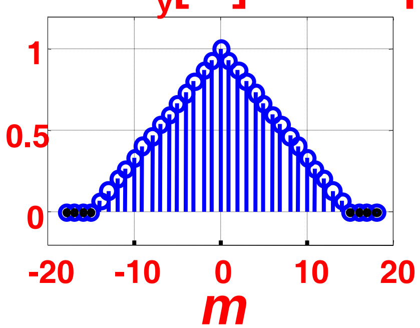
ACF $R_y[m]$ of Output



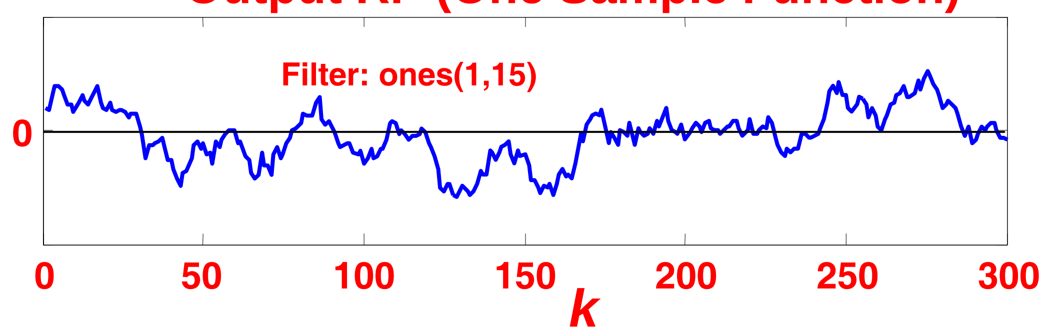
Output RP (One Sample Function)



ACF $R_y[m]$ of Output



Output RP (One Sample Function)



Filtered RPs: Insight

Our study of the ACFs of filtered random processes and the degree of "smoothness" of the sample functions shows the following general result:

Narrow ACF	↔	**Rapid Fluctuations**
Broad ACF	↔	**Slow Fluctuations**

Power Spectral Density of a Random Process

For a random Process: each realization (sample function) of process x(t) has different FT and therefore a different PSD.

We again rely on averaging to give the **"Expected" PSD** or **"Average" PSD**…

But… **Usually just call it "PSD"**.

Define PSD for WSS RP

We define PSD of WSS process x(t) to be :

$$S_X(\omega) = \lim_{T \to \infty} E\left\{ \frac{|X_T(\omega)|^2}{T} \right\} \quad \bigstar$$

This definition isn't very useful for analysis so we <u>seek an alternative form</u>

The **<u>Wiener-Khinchine Theorem</u>** provides this alternative!!!

Weiner- Khinchine Theorem

Let x(t) be a WSS process w/ ACF $R_X(\tau)$ and w/ PSD $S_X(\omega)$ as defined in (★)… Then $R_X(\tau)$ and $S_X(\omega)$ form a FT pair :

$$S_X(\omega) = \mathcal{F}\{ R_X(\tau) \}$$

or Equivalently

$$R_X(\tau) \leftrightarrow S_X(\omega)$$

Computing Power from PSD

From it's name – Power Spectral Density – we know what to expect :

$$P_x = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_x(\omega)\, d\omega$$

$$P_x = \int_{-\infty}^{\infty} S_x(\omega) \frac{d\omega}{2\pi} \Rightarrow S_x(\omega)$$

Watts

Hz

[Watts / Hz]

PSD for DT Processes

Not much changes – mostly, just use DTFT instead of CTFT!!

$$S_X(\Omega) = \text{DTFT}\{ R_X[m] \}$$

Periodic in Ω with period 2π

$$P_X = \frac{1}{2\pi} \int_{-\pi}^{\pi} S_X(\Omega)\, d\Omega$$

Need only look at -π ≤ Ω < π

Big Picture of PSD & ACF

Narrow ACF ↔ Broad PSD

Less Correlated Sample-to-sample ↔ Process exhibits Rapid fluctuations

i.e. **<u>High Frequencies</u> have <u>Large</u> power content**

Broad ACF ↔ Narrow PSD

More Correlated Sample-to-sample ↔ Process exhibits Slow fluctuations

i.e. **<u>High Frequencies</u> have <u>Small</u> power content**

<<See “Big Picture: Filtered RP” back a few Charts >>

White Noise

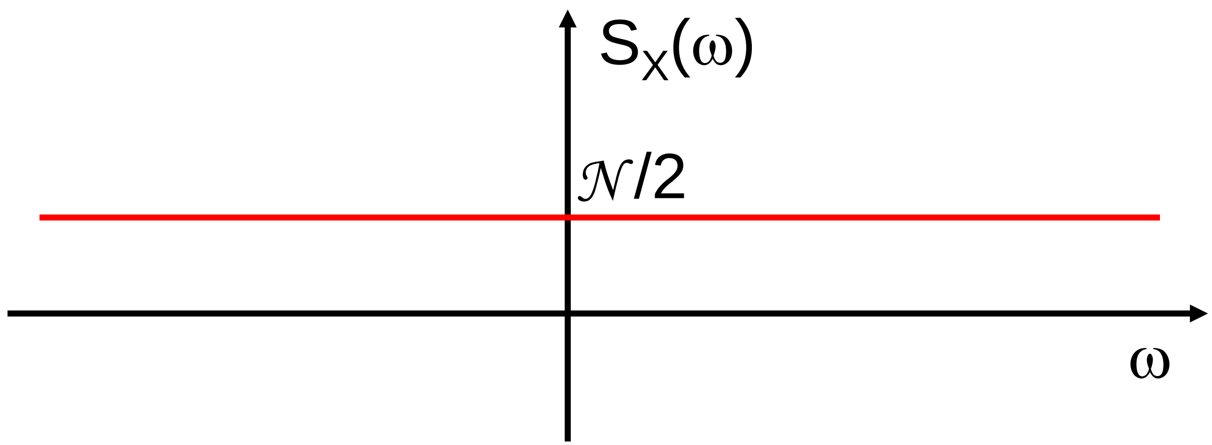
The term **"White Noise"** refers to a WSS process whose **PSD is flat** over all frequencies

C-T White Noise

$$S_X(\omega) = \frac{\mathcal{N}}{2} \quad \forall \omega$$

White Noise Has Broadest Possible PSD

Convention to use this form (i.e. w/ division by 2)



C-T White Noise

<u>NOTE</u> : C-T white noise has **<u>infinite Power</u>** :

$$\int_{-\infty}^{\infty} \mathcal{N}/2 \, d\omega \to \infty$$

Can't **really** exist in practice but still a **very** useful <u>Model for Analysis of Practical Scenarios</u>

C-T White Noise

Q : what is the ACF of C-T white Noise ?

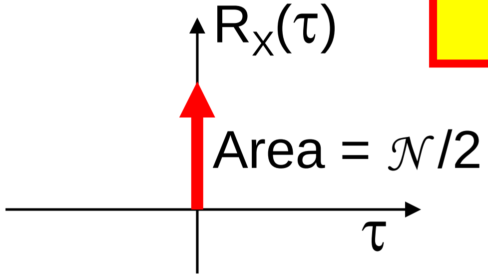
A: Take the IFT of the flat PSD :

$$R_X(\tau) = \mathfrak{F}^{-1}\{\mathcal{N}/2\}$$

$$= \frac{\mathcal{N}}{2}\,\delta(\tau)$$

Delta function !

Narrowest ACF



$x(t_1)$ & $x(t_2)$ are <u>uncorrelated</u> for <u>any</u> $t_1 \neq t_2$

Also….

$$P_X = R_X(0) = \mathcal{N}/2\,\delta(0) \rightarrow \infty$$

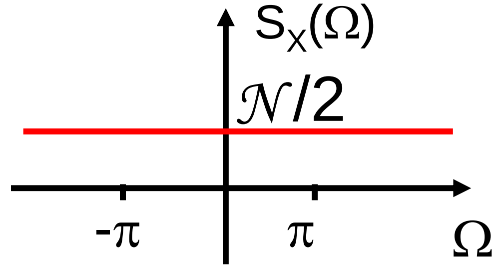
Infinite Power.. It Checks!

D-T White Noise

PSD is:

$S_X(\Omega) = \mathcal{N}/2 \ \forall \ \Omega$

…but focus on $\Omega \in [-\pi, \pi]$



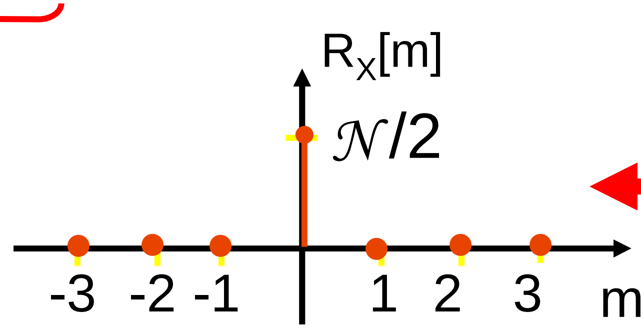
Broadest Possible PSD

ACF is:

$R_X[m] = \text{IDTFT}\,\{\mathcal{N}/2\}$

$= \mathcal{N}/2 \ \delta[m]$

Delta sequence



Narrowest ACF

$x[k_1]$ & $x[k_2]$ are uncorrelated for <u>any</u> $k_1 \neq k_2$

D-T White Noise

<u>Note</u>:

$$P_x = R_x[0] = \frac{\mathcal{N}}{2} \quad \text{watts}$$

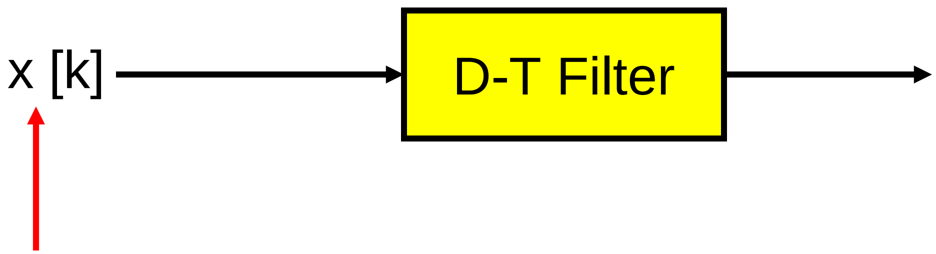
$$P_x = \frac{1}{2\pi}\int_{-\pi}^{\pi} \frac{\mathcal{N}}{2}\, d\Omega = \frac{\mathcal{N}}{2} \quad \text{watts}$$

D-T White Noise has **<u>Finite</u> Power**
(unlike C-T White Noise)

Example #2 of PSD

Example 2: “FILTERED D-T RANDOM PROCESS”

< See Also: “Filtered RPs” back a few charts >

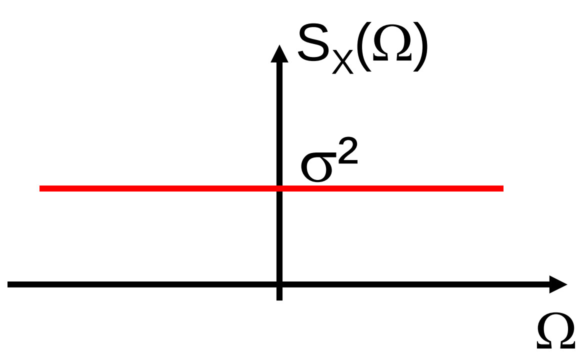


$y[k] = x[k] + x[k+1]$

Zero mean
White noise

$\Rightarrow R_X[m] = \sigma^2 \delta[m]$ (Input ACF)

$\Rightarrow S_X(\Omega) = \sigma^2 \quad \forall \Omega$ (Input PSD)



Example #3 of PSD

For this case we showed earlier that for <u>this</u> filter output the ACF is :

$$R_Y[m] = \sigma^2 \{ 2\delta[m] + \delta[m-1] + \delta[m+1] \}$$

So the <u>Output PSD</u> is:

$$S_Y(\Omega) = \sigma^2 [2 + e^{-j\Omega} + e^{-j\Omega}]$$

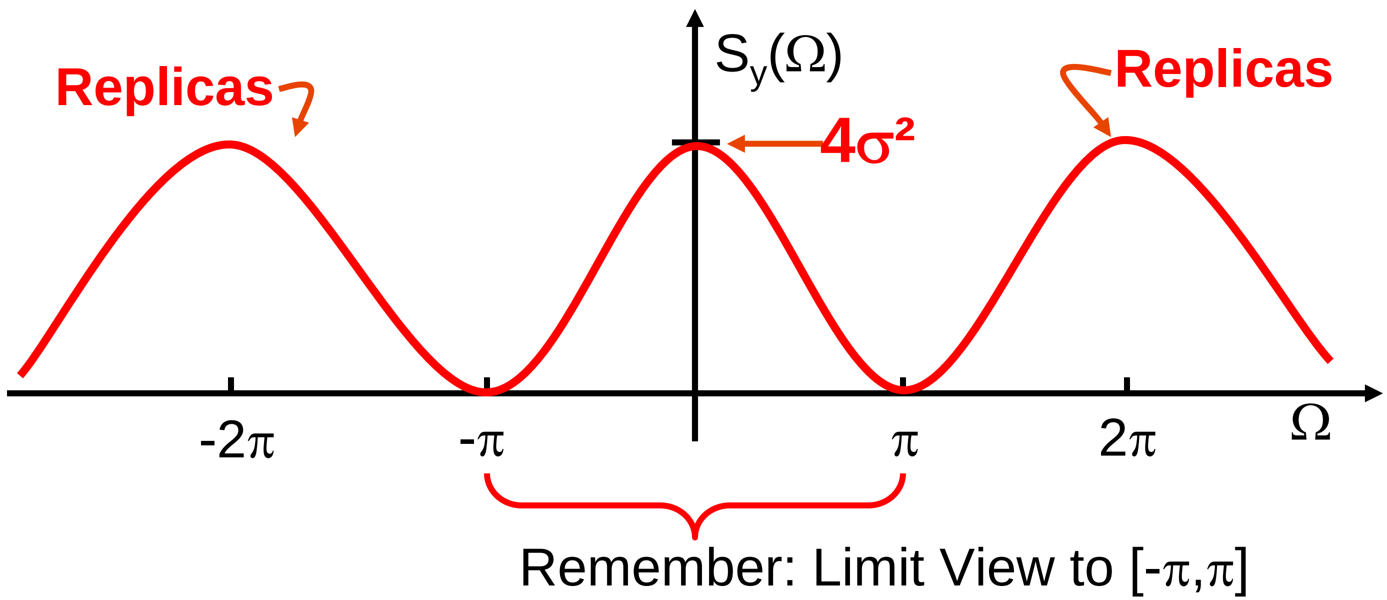
$$\mathbf{= 2\sigma^2 [\cos(\Omega) + 1]}$$

Use the result for DTFT of $\delta[m]$ and also time-shift property

$= 2 \cos(\Omega)$ By Euler

Example #3 of PSD

$S_Y(\Omega) = 2\sigma^2 [\cos(\Omega) + 1]$



<u>General Idea…Filter Shapes Input PSD:</u>

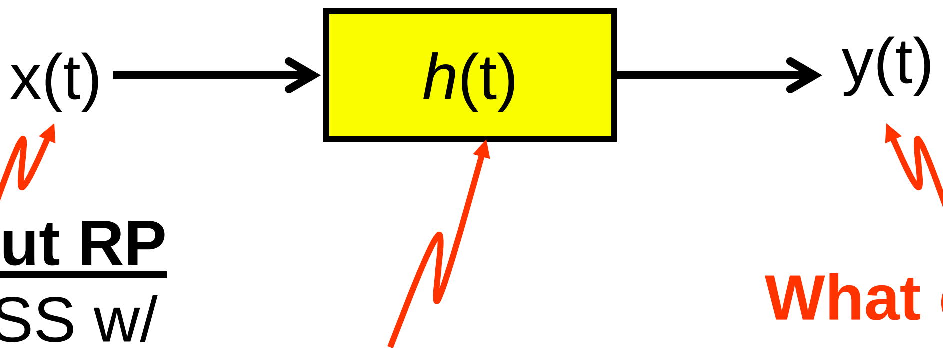
Here it suppresses High Frequency power

RPs Through LTI Systems

We already saw that passing DT white noise through a FIR filter reshapes the ACF and PSD.

Here we learn the <u>General Theory:</u>

(extremely useful for Modeling Practical RP's)



<u>Input RP</u>

WSS w/

$R_x(\tau)$

$S_x(\omega)$

<u>LTI System</u>

Impulse Response $h(t)$

Frequency Response $H(\omega) = \mathfrak{F}\{h(t)\}$

<u>Output RP</u>

What does it look like?

RPs & LTI Systems: Results

To describe output RP y(t) we look at its:

(i) Mean
(ii) ACF and
(iii) PSD

Results First (Proof Later)

(i) Mean:

$$E\{y(t)\} = H(0)E\{x(t)\}$$

Comment: Means are viewed as the DC Value of a RP – it makes sense that the Filter's DC Response, H(0), transfers "input-DC" to "output-DC"

RPs & LTI Systems: Results

(ii) ACF: $R_y(\tau) = h(\tau)*h(-\tau)*R_x(\tau)$

Comments:

(1) Implicit in this is **"WSS into LTI gives WSS out"**

(2) The "second-order" dependence on h(.) comes from the ACF being a "second-order" characteristic

(3) ACF is a time-domain characteristic so it makes sense that convolution is involved.

RPs & LTI Systems: Results

(iii) PSD:

$$S_y(\omega) = |H(\omega)|^2 S_x(\omega)$$

Comments: (1) Again, 2nd-order dependence on $H(\omega)$ comes from PSD being a 2nd-order characteristic

(2) PSD is a Frequency-domain characteristic so it makes sense that the frequency response $H(\omega)$ is involved.

Ex: Filtered White Noise

Earlier we looked at figures showing how five different (but similar) filters impact the output ACF.
Recall that in those examples the input was **D-T white noise** **$\Rightarrow R_X[m]=\sigma^2\delta[m]$**. Thus the output ACF's are just the convolution: **$\sigma^2 h[m]*h[-m]$**.

The filters in the previous case all had rectangular impulse responses, which when convolved like this
give the triangular ACF's shown in the previous figures.

Note also: rectangular FIR filters are low-pass filters whose cut-off frequency gets lower as the filter length increases.

Ex: Filtered White Noise

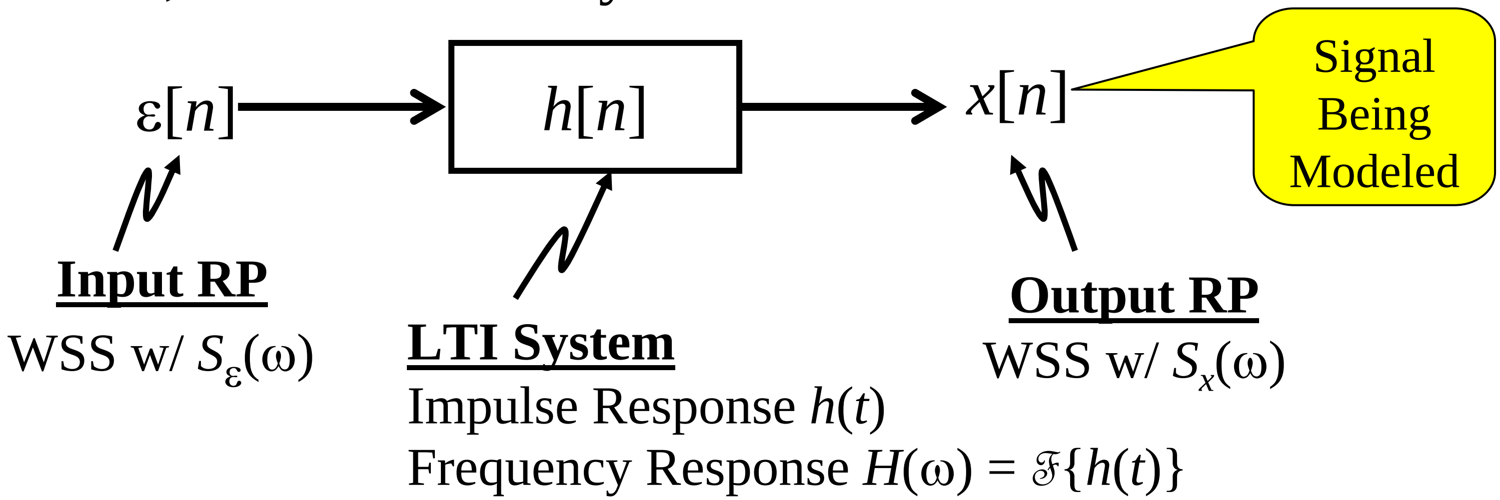
Thus , Since $$S_y(\Omega) = |H(\Omega)|^2 \underbrace{S_x(\Omega)}_{=\ \mathcal{N}/2 \text{ for White Noise}}$$

PSD's of processes that are outputs of longer rectangle filters have narrower PSD's

Wide ACF	↔	**Narrow PSD**	➡	Process has slow fluctuations
Narrow ACF	↔	**Broad PSD**	➡	Process has fast Fluctuations

Linear System Models for RPs: ARMA, AR, MA)

From the result we just saw that relates output PSD to input PSD for a linear, time-invariant system:



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

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

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

RP Models via 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…. We can characterize the RP by the model parameters

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 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

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

Poles & Zeros Give Rise to PSD Spikes & Nulls

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

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

Output is an "average" of values inside a moving window

Order of the model is q: called MA(q) model

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

Zeros Give Rise to PSD Nulls

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

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + \varepsilon[n] \quad (b_0 = 1)$$

Output depends "regressively" on itself

Since *x*[*n*] depends only its past *p* values it is a *p*th order Markov Model

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

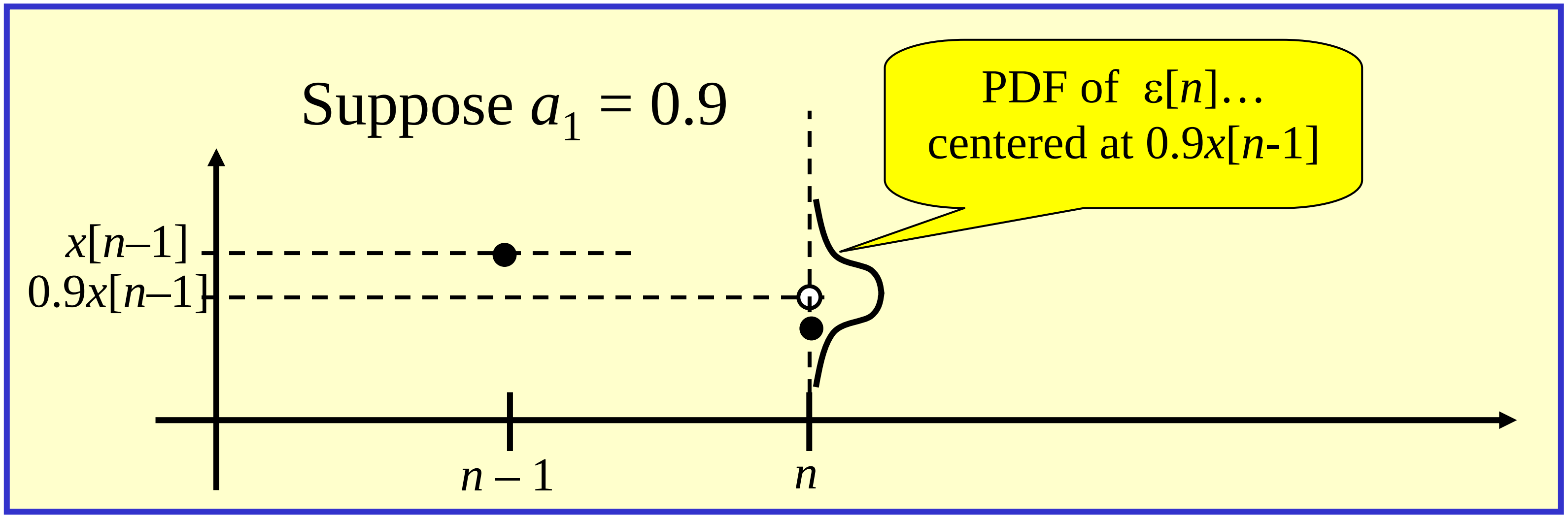
$$S_{AR}(\omega) = \frac{\sigma^2}{\left| 1 + \sum_{k=1}^{p} a_k e^{-j\omega k} \right|^2}$$

Poles Give Rise to PSD Spikes

Ex. First-Order AR Model

For an AR(1) process the defining time-domain model is

$$x[n] = \underbrace{a_1 x[n-1]}_{\text{Conditionally-Deterministic Part}} + \underbrace{\varepsilon[n]}_{\text{Random Part}}$$



ACF: $$R(k) = \left[\frac{\sigma_\varepsilon^2}{1-a_1^2}\right] a_1^k$$

Exponential… at each step the correlation is reduced by a factor of a_1

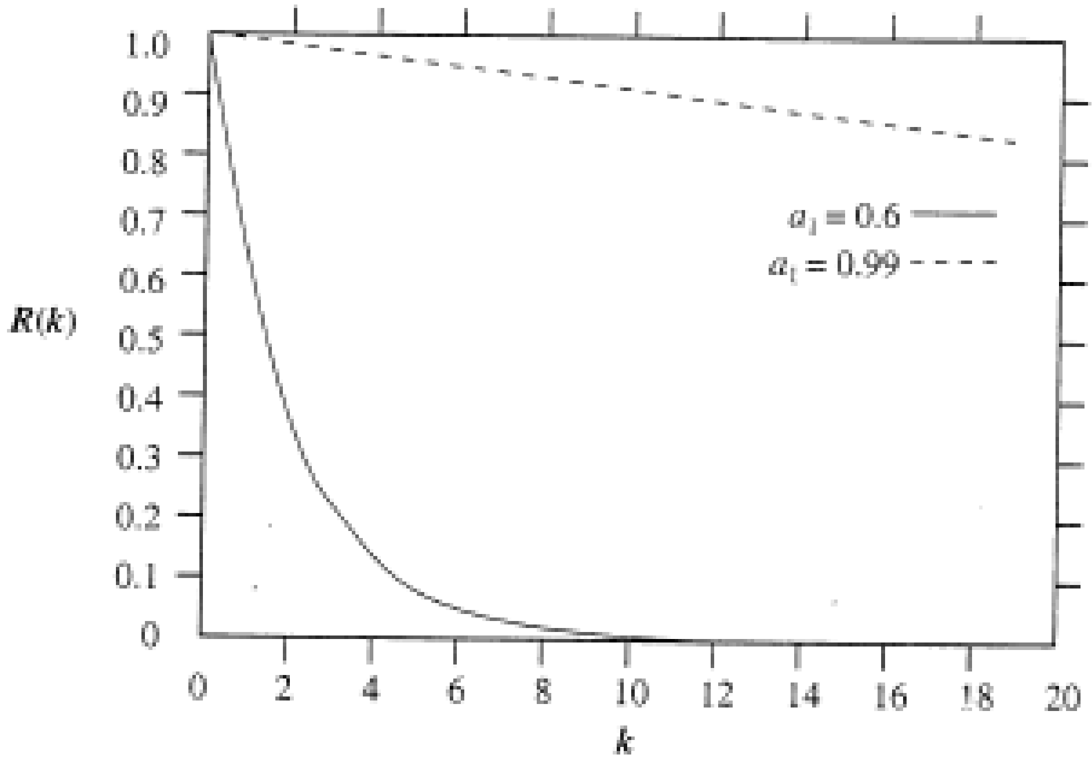


FIGURE 7.6 **Autocorrelation function of an AR(1) process with two values of a_1.**

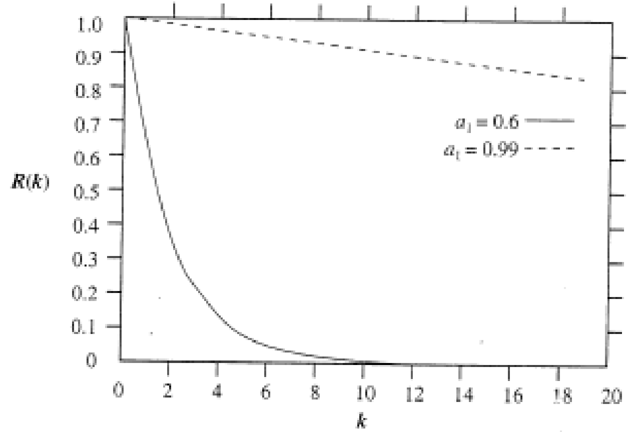


FIGURE 7.6 Autocorrelation function of an AR(1) process with two values of a_1.

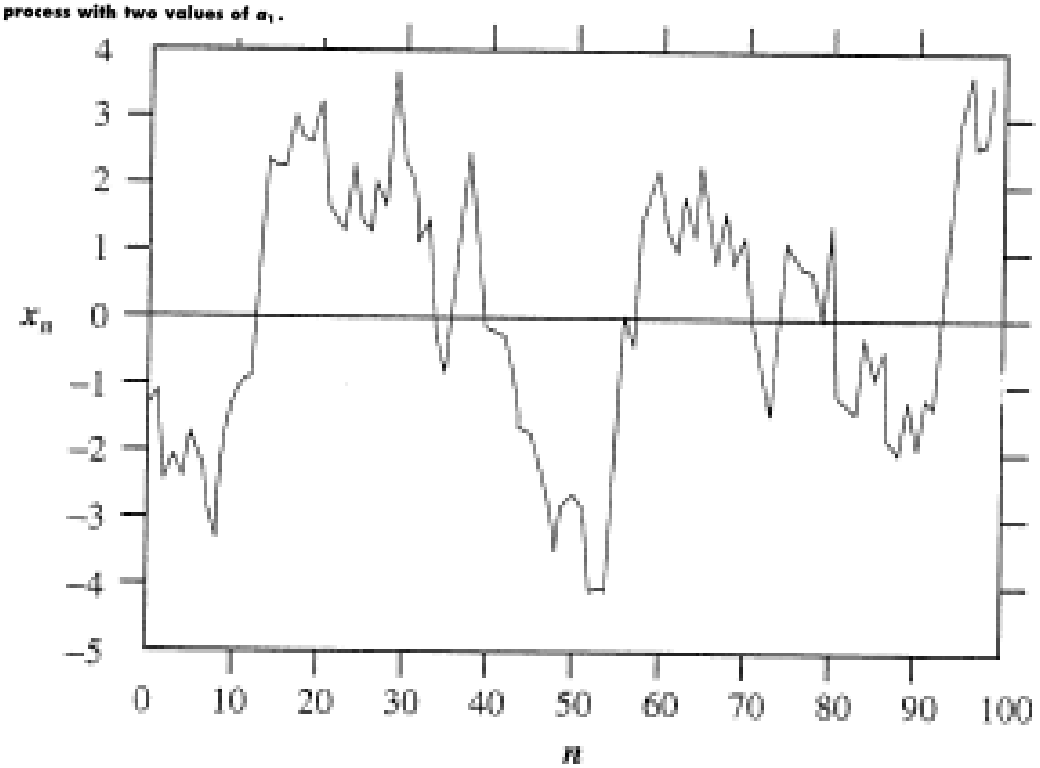


FIGURE 7.7 Sample function of an AR(1) process with $a_1 = 0.99$.

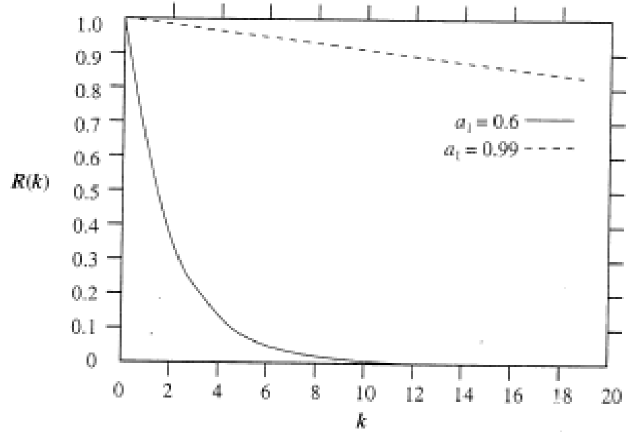


FIGURE 7.6 Autocorrelation function of an AR(1) process with two values of a_1.

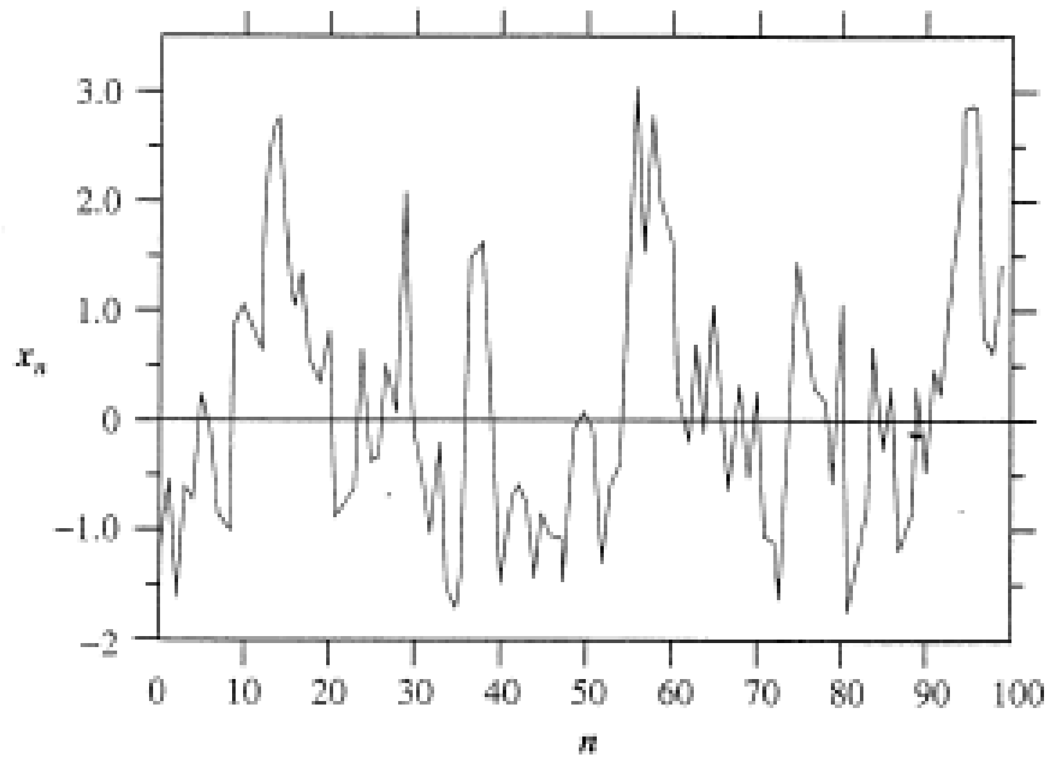


FIGURE 7.8 Sample function of an AR(1) process with $a_1 = 0.6$.

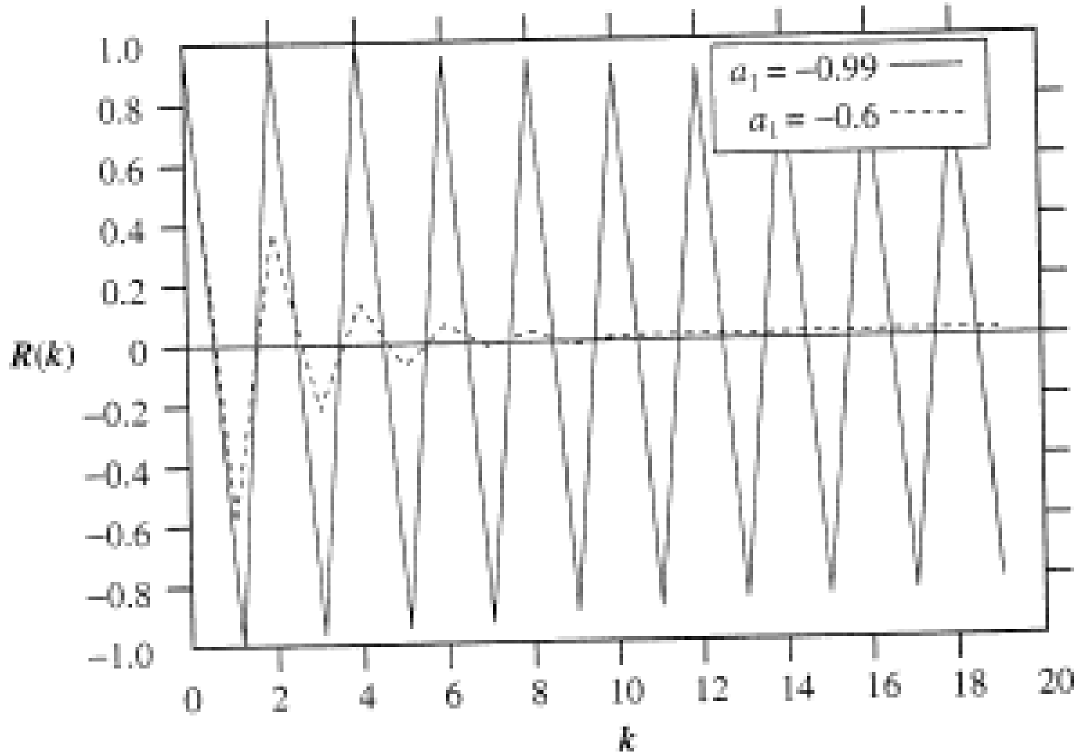


FIGURE 7.11 Autocorrelation function of an AR(1) process with two negative values of a_1.

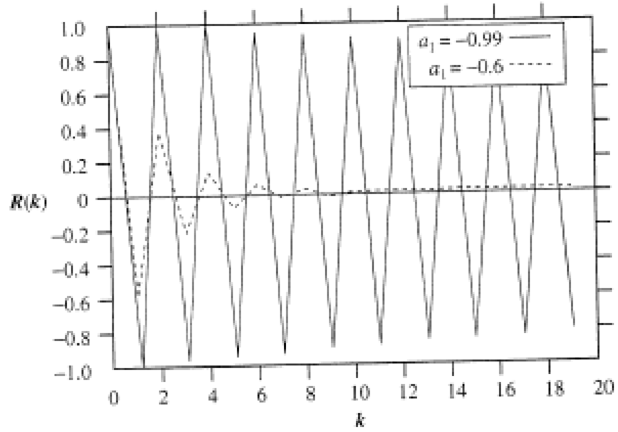


FIGURE 7.11 Autocorrelation function of an AR(1) process with two negative values of a_1.

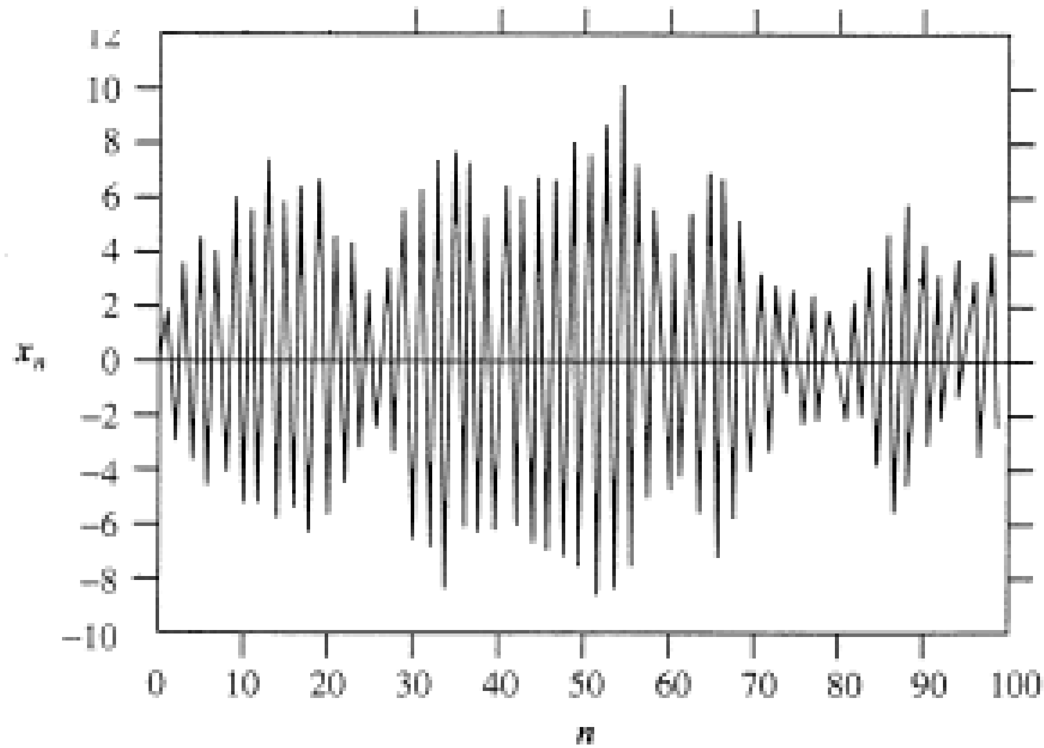


FIGURE 7.9 Sample function of an AR(1) process with $a_1 = -0.99$.

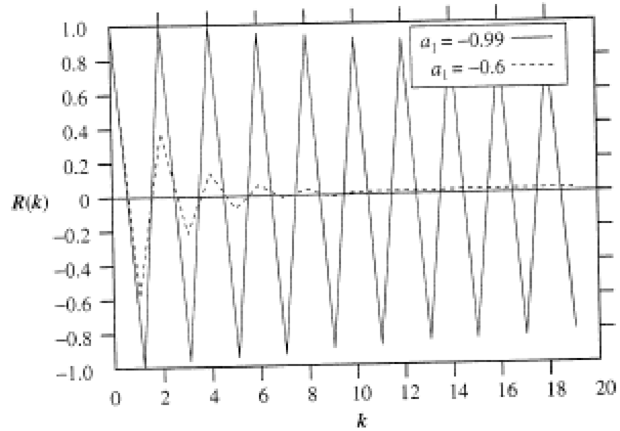


FIGURE 7.11 Autocorrelation function of an AR(1) process with two negative values of

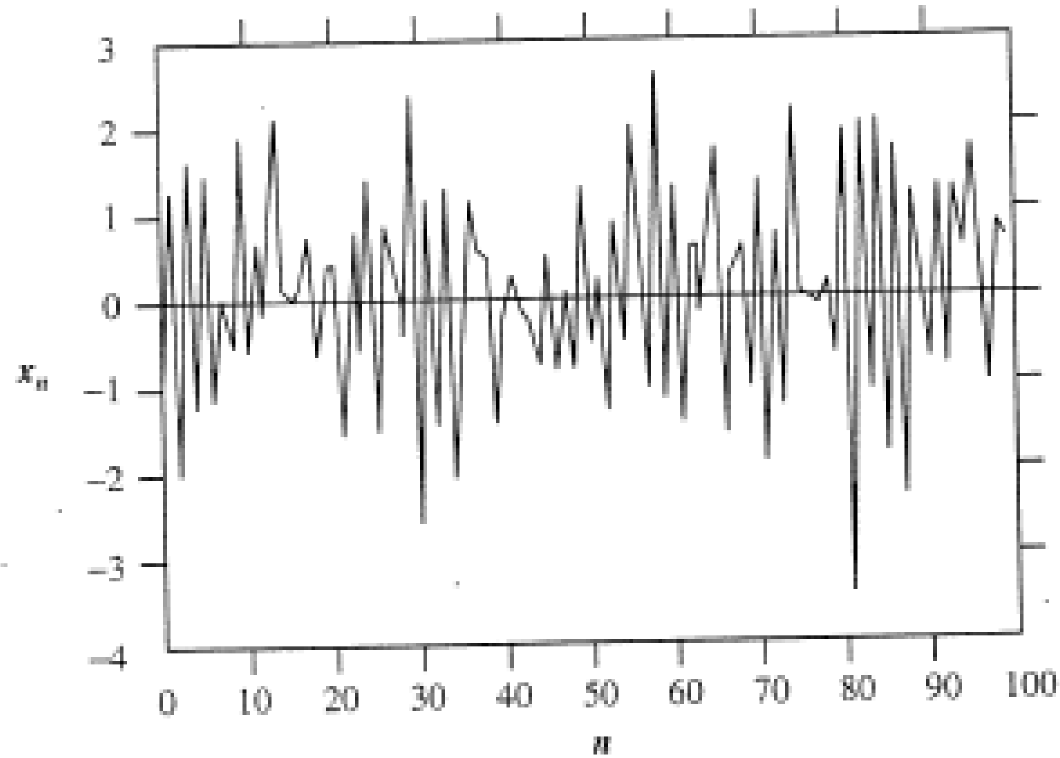


FIGURE 7.10 Sample function of an AR(1) process with $a_1 = -0.6$.

Linear Prediction & AR

Recall the AR model structure:

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

If we re-arrange this output equation we get:

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

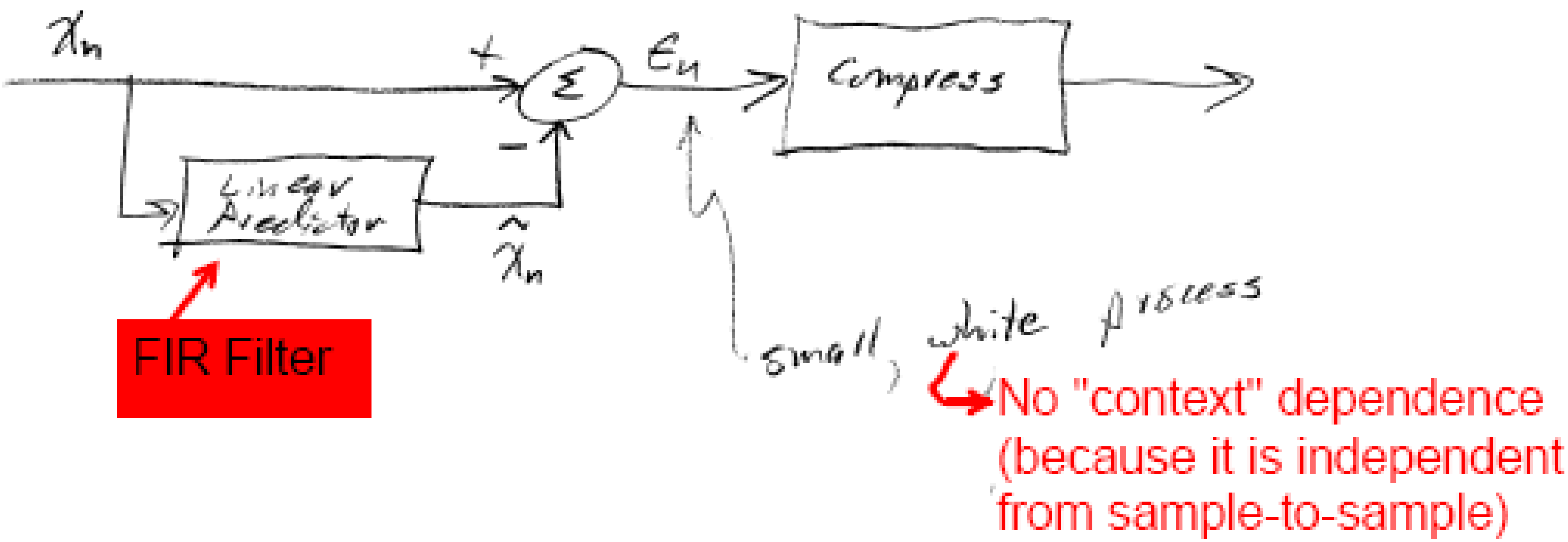
Prediction Error ($\varepsilon[n]$)

Prediction of $x[n]$ based on p past values ($\hat{x}[n]$)

Exploting Linear Prediction for Compression

There are lots of applications where linear prediction is used:

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



As we will see later, the prediction is easier to compress for two reasons

1. **It has had its "context dependence" removed**
2. **It is limited to a smaller dynamic range**