ELEG-636: Statistical Signal Processing

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

Spring 2010

Course Objectives & Structure

Objective: Given a discrete time sequence $\{x(n)\}$, develop

- Statistical and spectral signal representation
- Filtering, prediction, and system identification algorithms
- Optimization methods that are
  - Statistical
  - Adaptive

Course Structure:

- Weekly lectures [notes: www.ece.udel.edu/∼arce]
- Periodic homework (theory & Matlab implementations) [15%]
- Midterm & Final examinations [85%]

Textbook:

- Haykin, Adaptive Filter Theory.

Course Objectives & Structure

- Broad Applications in Communications, Imaging, Sensors.
- Emerging application in
  - Brain-imaging techniques
  - Brain-machine interfaces,
  - Implantable devices.
- Neurofeedback presents real-time physiological signals from MRIs in a visual or auditory form to provide information about brain activity. These signals are used to train the patient to alter neural activity in a desired direction.
- Traditionally, feedback using EEGs or other mechanisms has not focused on the brain because the resolution is not good enough.

Outline

Outline I

Blind Deconvolution

Motivation:

- Adaptive equalizers typically require a training period during which they operate on known signals/statistics.

Blind Deconvolution

Motivation:

- Adaptive equalizers typically require a training period during which they operate on known signals/statistics.
- This known signal training is not always appropriate, e.g., in mobile communications

Blind Deconvolution

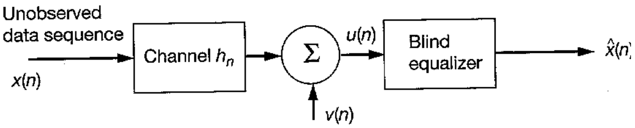
Motivation:

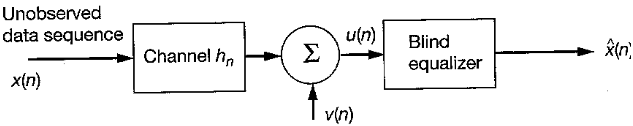
- Adaptive equalizers typically require a training period during which they operate on known signals/statistics.
- This known signal training is not always appropriate, e.g., in mobile communications
  - Cost is too high (time/bandwidth)
  - Multipathing or other time varying interference

Blind Deconvolution

Motivation:

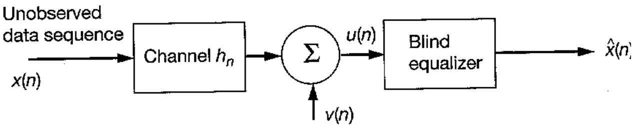
- Adaptive equalizers typically require a training period during which they operate on known signals/statistics.
- This known signal training is not always appropriate, e.g., in mobile communications
  - Cost is too high (time/bandwidth)
  - Multipathing or other time varying interference
- In such cases, we must use blind equalization


Unobserved
data sequence
x(n)
Channel h_n
Σ
u(n)
v(n)
Blind
equalizer
x̂(n)



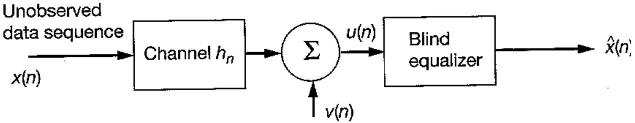
System components and assumptions:

- Channel introduces distortion (dominant)



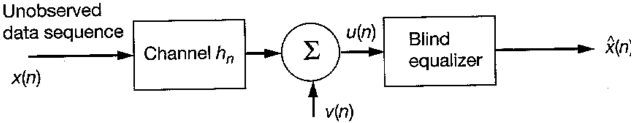
System components and assumptions:

- Channel introduces distortion (dominant)
- System has additive noise (not dominant)



System components and assumptions:

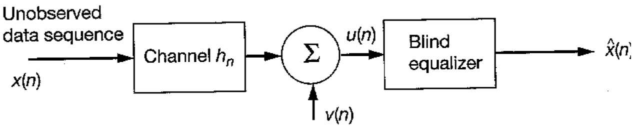
- Channel introduces distortion (dominant)
- System has additive noise (not dominant)
- Assume a baseband model of communications
    - Multilevel pulse amplitude modulation (M-ary PAM)



System components and assumptions:

- Channel introduces distortion (dominant)
- System has additive noise (not dominant)
- Assume a baseband model of communications
  - Multilevel pulse amplitude modulation (M-ary PAM)
  - Received signal:

$$u(n) = \sum_k h_k x(n-k)$$



System components and assumptions:

- Channel introduces distortion (dominant)
- System has additive noise (not dominant)
- Assume a baseband model of communications
  - Multilevel pulse amplitude modulation (M-ary PAM)
  - Received signal:

$$u(n) = \sum_k h_k x(n-k)$$

  - Dominating interference is due to intersymbol interference (ISI) from channel distortion
    $\Rightarrow$ The noise is ignored

Also assume that:

$$h \neq 0 \text{ for } n < 0 \quad \text{(noncausal)}$$

$$\sum_k h_k^2 = 1 \quad \text{(to keep the variance of the output constant)}$$

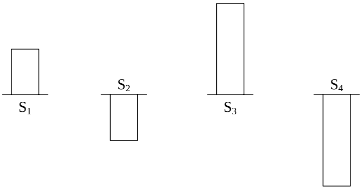
Also assume that:

$$h \neq 0 \text{ for } n < 0 \quad \text{(noncausal)}$$

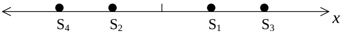
$$\sum_k h_k^2 = 1 \quad \text{(to keep the variance of the output constant)}$$

Example (4-ary PAM modulation)

A 4–ary PAM modulation scheme uses 4 signals



Or



To solve the equalization problem, we need a statistical model of the data.

To solve the equalization problem, we need a statistical model of the data. Assume,

1. The data is white

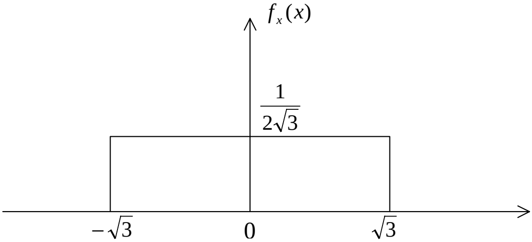
$$E\{x(n)\} = 0$$
$$E\{x(n)x(k)\} = \begin{cases} 1 & k = n \\ 0 & \text{otherwise} \end{cases}$$

To solve the equalization problem, we need a statistical model of the data. Assume,

1. The data is white

$$E\{x(n)\} = 0$$
$$E\{x(n)x(k)\} = \begin{cases} 1 & k = n \\ 0 & \text{otherwise} \end{cases}$$

2. The pdf of $x(n)$ is symmetric and uniform



Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$y(n) = \sum_i w_i u(n-i)$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$\begin{aligned} y(n) &= \sum_i w_i u(n-i) \\ &= \sum_i \sum_k w_i h_k x(n-i-k) \end{aligned}$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$\begin{aligned} y(n) &= \sum_i w_i u(n-i) \\ &= \sum_i \sum_k w_i h_k x(n-i-k) \qquad [\text{let } k = l - i] \end{aligned}$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$\begin{aligned} y(n) &= \sum_i w_i u(n-i) \\ &= \sum_i \sum_k w_i h_k x(n-i-k) \qquad [\text{let } k = l - i] \\ &= \sum_l x(n-l) \sum_i w_i h_{l-i} \end{aligned}$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$\begin{aligned} y(n) &= \sum_i w_i u(n-i) \\ &= \sum_i \sum_k w_i h_k x(n-i-k) \qquad [\text{let } k = l - i] \\ &= \sum_l x(n-l) \sum_i w_i h_{l-i} \\ &= \sum_l x(n-l)\delta_l \\ &= x(n) \end{aligned}$$

Deconvolution Objective: If $\{w_i\}$ are the coefficients of the ideal inverse filter, then

$$\sum_i w_i h_{l-i} = \delta_l = \begin{cases} 1 & l = 0 \\ 0 & \text{else} \end{cases}$$

If this is the case, the output of the equalizer is

$$\begin{aligned} y(n) &= \sum_i w_i u(n-i) \\ &= \sum_i \sum_k w_i h_k x(n-i-k) \qquad [\text{let } k = l - i] \\ &= \sum_l x(n-l) \sum_i w_i h_{l-i} \\ &= \sum_l x(n-l)\delta_l \\ &= x(n) \end{aligned}$$

Problem: h_n is not known $\Rightarrow$ the exact inverse can not be used

Solution: Use an iterative procedure to find the filter.

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i)$$

where a $2L+1$ tap filter is used

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i)$$

where a $2L+1$ tap filter is used

Setting $\hat{w}_i(n) = 0$ for $|i| > L$, we can write

$$y(n) = \sum_{i} \hat{w}_i(n)u(n-i)$$

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i)$$

where a $2L+1$ tap filter is used

Setting $\hat{w}_i(n) = 0$ for $|i| > L$, we can write

$$\begin{aligned} y(n) &= \sum_i \hat{w}_i(n)u(n-i) \\ &= \sum_i w_i u(n-i) + \sum_i [\hat{w}_i(n) - w_i]u(n-i) \end{aligned}$$

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i)$$

where a $2L+1$ tap filter is used

Setting $\hat{w}_i(n) = 0$ for $|i| > L$, we can write

$$\begin{aligned} y(n) &= \sum_i \hat{w}_i(n)u(n-i) \\ &= \sum_i w_i u(n-i) + \sum_i [\hat{w}_i(n) - w_i]u(n-i) \\ &= x(n) + v(n) \end{aligned}$$

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n)u(n-i)$$

where a $2L+1$ tap filter is used

Setting $\hat{w}_i(n) = 0$ for $|i| > L$, we can write

$$\begin{aligned} y(n) &= \sum_i \hat{w}_i(n)u(n-i) \\ &= \sum_i w_i u(n-i) + \sum_i [\hat{w}_i(n) - w_i]u(n-i) \\ &= x(n) + v(n) \qquad \left[\text{since } x(n) = \sum_i w_i u(n-i)\right] \end{aligned}$$

Solution: Use an iterative procedure to find the filter.

Let the output at iteration n be given by

$$y(n) = \sum_{i=-L}^{L} \hat{w}_i(n) u(n-i)$$

where a $2L+1$ tap filter is used

Setting $\hat{w}_i(n) = 0$ for $|i| > L$, we can write

$$\begin{aligned} y(n) &= \sum_i \hat{w}_i(n) u(n-i) \\ &= \sum_i w_i u(n-i) + \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= x(n) + v(n) \qquad \left[\text{since } x(n) = \sum_i w_i u(n-i)\right] \end{aligned}$$

Note $v(n) = \sum_i [\hat{w}_i(n) - w_i] u(n-i)$ is the residual ISI

Interpretation: $v(n) = \sum_i [\hat{w}_i(n) - w_i]u(n-i)$ is the convolution noise (residual ISI) resulting from the fact that ideal filter was not used

Interpretation: $v(n) = \sum_i [\hat{w}_i(n) - w_i]u(n-i)$ is the convolution noise (residual ISI) resulting from the fact that ideal filter was not used

Estimation Approach: Apply the output $y(n) = x(n) + v(n)$ to a zero memory nonlinear estimator

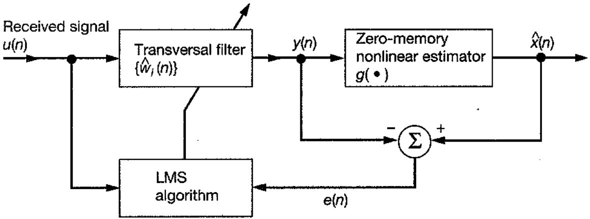
$$\hat{x}(n) = g[y(n)]$$

Interpretation: $v(n) = \sum_i [\hat{w}_i(n) - w_i] u(n-i)$ is the convolution noise (residual ISI) resulting from the fact that ideal filter was not used

Estimation Approach: Apply the output $y(n) = x(n) + v(n)$ to a zero memory nonlinear estimator

$$\hat{x}(n) = g[y(n)]$$

Complete System:

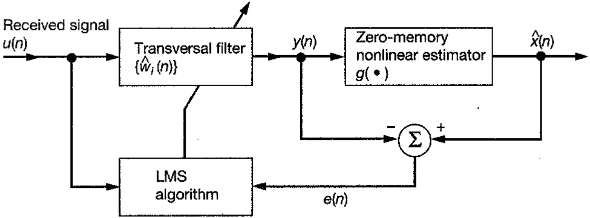


Interpretation: $v(n) = \sum_i [\hat{w}_i(n) - w_i] u(n-i)$ is the convolution noise (residual ISI) resulting from the fact that ideal filter was not used

Estimation Approach: Apply the output $y(n) = x(n) + v(n)$ to a zero memory nonlinear estimator

$$\hat{x}(n) = g[y(n)]$$

Complete System:



- The nonlinear estimate $g[y(n)]$ can be used to update the equalizer to produce a better estimate at time $n+1$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Note that in this case, the cost function is

$$J(n) = E\{e^2(n)\}$$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Note that in this case, the cost function is

$$\begin{aligned} J(n) &= E\{e^2(n)\} \\ &= E\{(\hat{x}(n) - y(n))^2\} \end{aligned}$$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Note that in this case, the cost function is

$$\begin{aligned} J(n) &= E\{e^2(n)\} \\ &= E\{(\hat{x}(n) - y(n))^2\} \\ &= E\{(g[y(n)] - y(n))^2\} \end{aligned}$$

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Note that in this case, the cost function is

$$\begin{aligned} J(n) &= E\{e^2(n)\} \\ &= E\{(\hat{x}(n) - y(n))^2\} \\ &= E\{(g[y(n)] - y(n))^2\} \end{aligned}$$

Observation: $J(n)$ is a nonconvex function of the filter weights ($g[\cdot]$ is nonlinear)

Define the equalizer error to be

$$e(n) = \hat{x}(n) - y(n)$$

Use $e(n)$ in the LMS algorithm to update the equalizer weights

$$\hat{w}_i(n+1) = \hat{w}_i(n) + \mu u(n-i)e(n) \qquad i = 0, \pm 1, \pm 2, \cdots, \pm L$$

Note that in this case, the cost function is

$$\begin{aligned} J(n) &= E\{e^2(n)\} \\ &= E\{(\hat{x}(n) - y(n))^2\} \\ &= E\{(g[y(n)] - y(n))^2\} \end{aligned}$$

Observation: $J(n)$ is a nonconvex function of the filter weights ($g[\cdot]$ is nonlinear)

- The cost function can have numerous local minima

To solve the optimization, first evaluate the convolution noise,

$$u(n) = \sum_k h_k x(n-k)$$

To solve the optimization, first evaluate the convolution noise,

$$\begin{aligned} u(n) &= \sum_k h_k x(n-k) \\ \Rightarrow u(n-i) &= \sum_k h_k x(n-i-k) \end{aligned}$$

To solve the optimization, first evaluate the convolution noise,

$$\begin{aligned} u(n) &= \sum_k h_k x(n-k) \\ \Rightarrow u(n-i) &= \sum_k h_k x(n-i-k) \end{aligned}$$

Next, use this in

$$y(n) = \underbrace{\sum_i w_i u(n-i)}_{x(n)} + \underbrace{\sum_i [\hat{w}_i(n) - w_i] u(n-i)}_{v(n)}$$

To solve the optimization, first evaluate the convolution noise,

$$\begin{aligned} u(n) &= \sum_k h_k x(n-k) \\ \Rightarrow u(n-i) &= \sum_k h_k x(n-i-k) \end{aligned}$$

Next, use this in

$$y(n) = \underbrace{\sum_i w_i u(n-i)}_{x(n)} + \underbrace{\sum_i [\hat{w}_i(n) - w_i] u(n-i)}_{v(n)}$$

where recall

$$\begin{aligned} w_i &\equiv \text{perfect equalizer weights} \\ \hat{w}_i(n) &\equiv \text{finite approximate equalizer with } \hat{w}_i(n) = 0 \text{ for } |i| > L \end{aligned}$$

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$v(n) = \sum_i [\hat{w}_i(n) - w_i] u(n-i)$$

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$\begin{aligned} v(n) &= \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= \sum_i \sum_k h_k [\hat{w}_i(n) - w_i] x(n-i-k) \end{aligned}$$

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$\begin{aligned} v(n) &= \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= \sum_i \sum_k h_k [\hat{w}_i(n) - w_i] x(n-i-k) \\ &= \sum_l x(l) \nabla(n-l) \end{aligned}$$

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$\begin{aligned} v(n) &= \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= \sum_i \sum_k h_k [\hat{w}_i(n) - w_i] x(n-i-k) \\ &= \sum_l x(l) \nabla(n-l) \end{aligned}$$

where the last line follows from letting $n-i-k = l$ and defining

$$\nabla(n) = \sum_k h_k [\hat{w}_{n-k}(n) - w_{n-k}]$$

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$\begin{aligned} v(n) &= \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= \sum_i \sum_k h_k [\hat{w}_i(n) - w_i] x(n-i-k) \\ &= \sum_l x(l) \nabla(n-l) \end{aligned}$$

where the last line follows from letting $n - i - k = l$ and defining

$$\nabla(n) = \sum_k h_k [\hat{w}_{n-k}(n) - w_{n-k}]$$

- $\nabla(n)$ is the residual impulse response of the channel due to imperfect equalization

Then using $u(n-i) = \sum_k h_k x(n-i-k)$

$$\begin{aligned} v(n) &= \sum_i [\hat{w}_i(n) - w_i] u(n-i) \\ &= \sum_i \sum_k h_k [\hat{w}_i(n) - w_i] x(n-i-k) \\ &= \sum_l x(l) \nabla(n-l) \end{aligned}$$

where the last line follows from letting $n-i-k=l$ and defining

$$\nabla(n) = \sum_k h_k [\hat{w}_{n-k}(n) - w_{n-k}]$$

- $\nabla(n)$ is the residual impulse response of the channel due to imperfect equalization
- $\nabla(n)$ is small in value, but long and oscillatory

Since the convolution noise is given by

$$v(n) = \sum_{l} x(l)\nabla(n-l)$$

Since the convolution noise is given by

$$
\begin{aligned}
v(n) &= \sum_l x(l)\nabla(n-l) \\
\Rightarrow E\{v(n)\} &= \sum_l E\{x(l)\}\nabla(n-l) = 0
\end{aligned}
$$

Since the convolution noise is given by

$$\begin{aligned} v(n) &= \sum_{l} x(l)\nabla(n-l) \\ \Rightarrow E\{v(n)\} &= \sum_{l} E\{x(l)\}\nabla(n-l) = 0 \end{aligned}$$

Also

$$E\{v(n)v(n-j)\} = E\left\{\sum_{l} x(l)\nabla(n-l)\sum_{m} x(m)\nabla(n-m-j)\right\}$$

Since the convolution noise is given by

$$\begin{aligned} v(n) &= \sum_l x(l)\nabla(n-l) \\ \Rightarrow E\{v(n)\} &= \sum_l E\{x(l)\}\nabla(n-l) = 0 \end{aligned}$$

Also

$$\begin{aligned} E\{v(n)v(n-j)\} &= E\left\{\sum_l x(l)\nabla(n-l)\sum_m x(m)\nabla(n-m-j)\right\} \\ &= \sum_l \sum_m \nabla(n-l)\nabla(n-m-j)E\{x(l)x(m)\} \end{aligned}$$

Since the convolution noise is given by

$$\begin{aligned} v(n) &= \sum_l x(l)\nabla(n-l) \\ \Rightarrow E\{v(n)\} &= \sum_l E\{x(l)\}\nabla(n-l) = 0 \end{aligned}$$

Also

$$\begin{aligned} E\{v(n)v(n-j)\} &= E\left\{\sum_l x(l)\nabla(n-l)\sum_m x(m)\nabla(n-m-j)\right\} \\ &= \sum_l\sum_m \nabla(n-l)\nabla(n-m-j)E\{x(l)x(m)\} \\ &= \sum_l \nabla(n-l)\nabla(n-l-j) \end{aligned}$$

Since $\nabla(n)$ is long and oscillatory

$$E\{v(n)v(n-j)\} = \sum_{l} \nabla(n-l)\nabla(n-l-j)$$

tends to average to 0 for $j \neq 0$.

Since $\nabla(n)$ is long and oscillatory

$$E\{v(n)v(n-j)\} = \sum_{l} \nabla(n-l)\nabla(n-l-j)$$

tends to average to 0 for $j \neq 0$. Thus

$$E\{v(n)v(n-j)\} = \begin{cases} \sigma^2 & j = 0 \\ 0 & j \neq 0 \end{cases}$$

where

$$\sigma^2 = \sigma^2(n) = \sum_{l} \nabla^2(n-l)$$

Since $\nabla(n)$ is long and oscillatory

$$E\{v(n)v(n-j)\} = \sum_{l} \nabla(n-l)\nabla(n-l-j)$$

tends to average to 0 for $j \neq 0$. Thus

$$E\{v(n)v(n-j)\} = \begin{cases} \sigma^2 & j = 0 \\ 0 & j \neq 0 \end{cases}$$

where

$$\sigma^2 = \sigma^2(n) = \sum_{l} \nabla^2(n-l)$$

Observation: $v(n) = \sum_{l} x(l)\nabla(n-l)$ is a weighted sum of i.i.d. RVs

Since $\nabla(n)$ is long and oscillatory

$$E\{v(n)v(n-j)\} = \sum_l \nabla(n-l)\nabla(n-l-j)$$

tends to average to 0 for $j \neq 0$. Thus

$$E\{v(n)v(n-j)\} = \begin{cases} \sigma^2 & j = 0 \\ 0 & j \neq 0 \end{cases}$$

where

$$\sigma^2 = \sigma^2(n) = \sum_l \nabla^2(n-l)$$

Observation: $v(n) = \sum_l x(l)\nabla(n-l)$ is a weighted sum of i.i.d. RVs $\Rightarrow$ $v(n)$ is Gaussian (central limit theorem)

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

$$E\{x(n)v(n-j)\} = E\{x(n)\sum_{l} x(l)\nabla(n-l-j)\}$$

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

$$\begin{aligned} E\{x(n)v(n-j)\} &= E\{x(n)\sum_l x(l)\nabla(n-l-j)\} \\ &= \sum_l \nabla(n-l-j)E\{x(n)x(l)\} \end{aligned}$$

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

$$
\begin{aligned}
E\{x(n)v(n-j)\} &= E\{x(n)\sum_{l} x(l)\nabla(n-l-j)\} \\
&= \sum_{l} \nabla(n-l-j)E\{x(n)x(l)\} \\
&= \nabla(-j)
\end{aligned}
$$

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

$$\begin{aligned} E\{x(n)v(n-j)\} &= E\{x(n)\sum_l x(l)\nabla(n-l-j)\} \\ &= \sum_l \nabla(n-l-j)E\{x(n)x(l)\} \\ &= \nabla(-j) \\ &\ll \sum_l \nabla^2(l) \end{aligned}$$

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

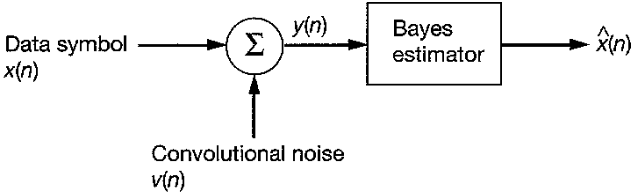
$$\begin{aligned} E\{x(n)v(n-j)\} &= E\{x(n)\sum_l x(l)\nabla(n-l-j)\} \\ &= \sum_l \nabla(n-l-j)E\{x(n)x(l)\} \\ &= \nabla(-j) \\ &\ll \sum_l \nabla^2(l) = \sigma^2 \end{aligned}$$

Lastly, consider the cross correlation of $x(n)$ and $v(n)$,

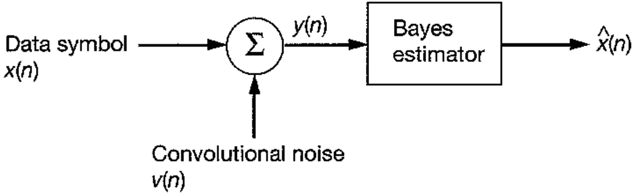
$$\begin{aligned} E\{x(n)v(n-j)\} &= E\{x(n)\sum_l x(l)\nabla(n-l-j)\} \\ &= \sum_l \nabla(n-l-j)E\{x(n)x(l)\} \\ &= \nabla(-j) \\ &\ll \sum_l \nabla^2(l) = \sigma^2 \end{aligned}$$

Observation: Thus we can say $x(n)$ and $v(n)$ are essentially independent

Finally, consider the nonlinear estimation of $x(n)$



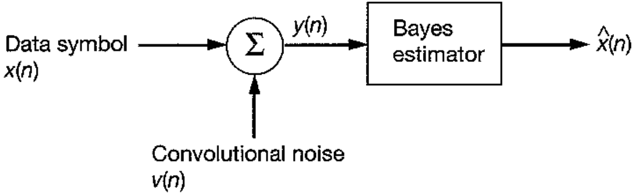
Finally, consider the nonlinear estimation of $x(n)$



Component statistics:

- $x(n)$ is uniformly distributed with zero mean and unit variance

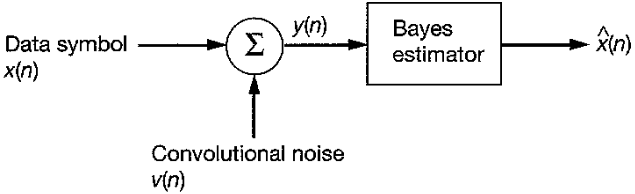
Finally, consider the nonlinear estimation of $x(n)$



Component statistics:

- $x(n)$ is uniformly distributed with zero mean and unit variance
- $v(n)$ is white Gaussian noise with zero mean and variance $\sigma^2(n)$

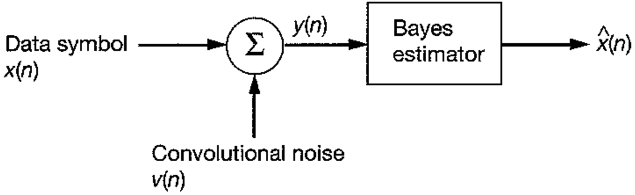
Finally, consider the nonlinear estimation of $x(n)$



Component statistics:

- $x(n)$ is uniformly distributed with zero mean and unit variance
- $v(n)$ is white Gaussian noise with zero mean and variance $\sigma^2(n)$
- $x(n)$ and $v(n)$ are independent

Finally, consider the nonlinear estimation of $x(n)$



Component statistics:

- $x(n)$ is uniformly distributed with zero mean and unit variance
- $v(n)$ is white Gaussian noise with zero mean and variance $\sigma^2(n)$
- $x(n)$ and $v(n)$ are independent

Estimation Approach: Utilize Bayes estimation, which exploits knowledge of the distributions

The estimation risk is defined by

$$R = E\{c(x, \hat{x}(n))\}$$

The estimation risk is defined by

$$\begin{aligned} R &= E\{c(x,\hat{x}(n))\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x,\hat{x}(n)) f_{xy}(x,y)\,dy\,dx \end{aligned}$$

The estimation risk is defined by

$$
\begin{aligned}
R &= E\{c(x, \hat{x}(n))\} \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x, \hat{x}(n)) f_{xy}(x, y)\, dy\, dx
\end{aligned}
$$

Recall that if $c(x, \hat{x}(n)) = (x - \hat{x}(n))^2$, then the risk is minimized by

The estimation risk is defined by

$$\begin{aligned} R &= E\{c(x,\hat{x}(n))\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x,\hat{x}(n)) f_{xy}(x,y)\,dy\,dx \end{aligned}$$

Recall that if $c(x,\hat{x}(n)) = (x-\hat{x}(n))^2$, then the risk is minimized by

$$\begin{aligned} \hat{x} &= \int_{-\infty}^{\infty} x f_x(x|y)\,dx \\ &= E\{x|y\} \qquad \text{[conditional expectation]} \end{aligned}$$

The estimation risk is defined by

$$\begin{aligned} R &= E\{c(x, \hat{x}(n))\} \\ &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} c(x, \hat{x}(n)) f_{xy}(x, y)\,dydx \end{aligned}$$

Recall that if $c(x, \hat{x}(n)) = (x - \hat{x}(n))^2$, then the risk is minimized by

$$\begin{aligned} \hat{x} &= \int_{-\infty}^{\infty} x f_x(x|y)\,dx \\ &= E\{x|y\} \qquad \text{[conditional expectation]} \end{aligned}$$

where we can use the fact that

$$f_x(x|y) = \frac{f_y(y|x) f_x(x)}{f_y(y)}$$

Employ the current model,

$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Employ the current model,

$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Thus since x and v are independent, $f_y(y|x) = f_v(y - c_0 x)$

Employ the current model,

$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Thus since x and v are independent, $f_y(y|x) = f_v(y - c_0 x)$ and

$$\hat{x} = \int_{-\infty}^{\infty} x f_x(x|y) dx$$

Employ the current model,

$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Thus since x and v are independent, $f_y(y|x) = f_v(y - c_0 x)$ and

$$\begin{aligned} \hat{x} &= \int_{-\infty}^{\infty} x f_x(x|y) dx \\ &= \frac{1}{f_y(y)} \int_{-\infty}^{\infty} x f_y(y|x) f_x(x) dx \end{aligned}$$

Employ the current model,

$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Thus since x and v are independent, $f_y(y|x) = f_v(y - c_0 x)$ and

$$\begin{aligned}
\hat{x} &= \int_{-\infty}^{\infty} x f_x(x|y) dx \\
&= \frac{1}{f_y(y)} \int_{-\infty}^{\infty} x f_y(y|x) f_x(x) dx \\
&= \frac{1}{f_y(y)} \int_{-\infty}^{\infty} x f_v(y - c_0 x) f_x(x) dx
\end{aligned}$$

Employ the current model,

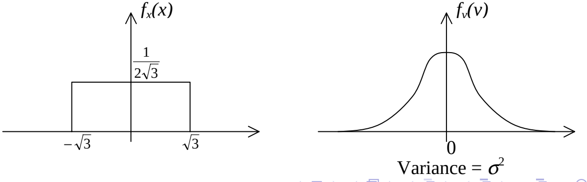
$$y = c_0 x + v$$

where $c_0 < 1$ is a scaling factor included to ensure $E\{y^2\} = 1$.

Thus since x and v are independent, $f_y(y|x) = f_v(y - c_0 x)$ and

$$\begin{aligned}
\hat{x} &= \int_{-\infty}^{\infty} x f_x(x|y) dx \\
&= \frac{1}{f_y(y)} \int_{-\infty}^{\infty} x f_y(y|x) f_x(x) dx \\
&= \frac{1}{f_y(y)} \int_{-\infty}^{\infty} x f_v(y - c_0 x) f_x(x) dx
\end{aligned}$$

where



Evaluating this yields (Berllini, 1988)

$$\hat{x} = \frac{1}{c_0 y} - \frac{\sigma}{c_0} \frac{Z(y_1) - Z(y_2)}{Q(y_1) - Q(y_2)}$$

Evaluating this yields (Berllini, 1988)

$$\hat{x} = \frac{1}{c_0 y} - \frac{\sigma}{c_0} \frac{Z(y_1) - Z(y_2)}{Q(y_1) - Q(y_2)}$$

where

$$y_1 = \frac{1}{\sigma}(y + \sqrt{3}c_0)$$

$$y_2 = \frac{1}{\sigma}(y - \sqrt{3}c_0)$$

Evaluating this yields (Berllini, 1988)

$$\hat{x} = \frac{1}{c_0 y} - \frac{\sigma}{c_0} \frac{Z(y_1) - Z(y_2)}{Q(y_1) - Q(y_2)}$$

where

$$y_1 = \frac{1}{\sigma}(y + \sqrt{3}c_0)$$

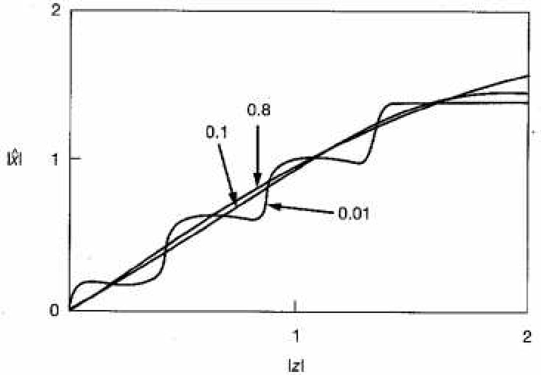
$$y_2 = \frac{1}{\sigma}(y - \sqrt{3}c_0)$$

and

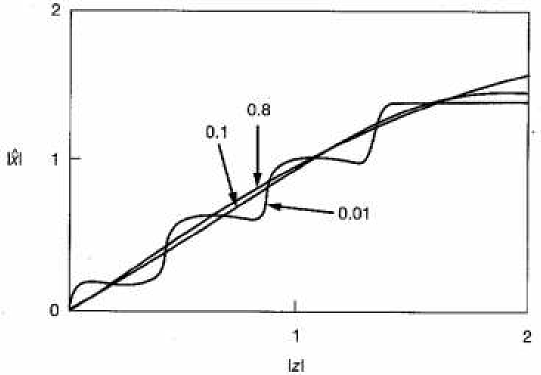
$$Z(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} \qquad \text{[normalized Gaussian pdf]}$$

$$Q(y) = \frac{1}{\sqrt{2\pi}} \int_y^\infty e^{-\frac{u^2}{2}} du \qquad \text{[normalized Gaussian cdf]}$$

Results for an 8 level PAM system:



Results for an 8 level PAM system:



- Plotted for three different noise to (signal + noise) ratios
- Note that this tends to a step function as the noise goes to zero

Implementation Point: When the blind equalizer has converged, the algorithm is switched to decision-directed mode.

Implementation Point: When the blind equalizer has converged, the algorithm is switched to decision-directed mode.

In the case of binary symbols,

$$x(n) = \begin{cases} +1 & \text{for symbol 1} \\ -1 & \text{for symbol 2} \end{cases}$$

Implementation Point: When the blind equalizer has converged, the algorithm is switched to decision-directed mode.

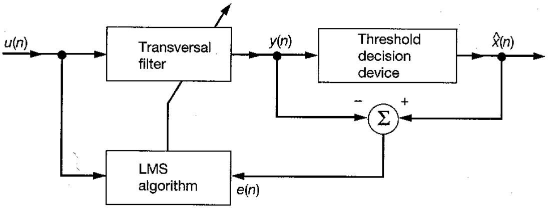
In the case of binary symbols,

$$x(n) = \begin{cases} +1 & \text{for symbol 1} \\ -1 & \text{for symbol 2} \end{cases}$$

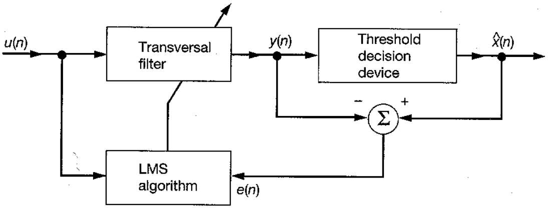
which results in the sample estimate

$$\Rightarrow \hat{x}(n) = \mathrm{sgn}(y(n)) = \begin{cases} +1 & \text{if } y(n) \geq 0 \\ -1 & \text{else} \end{cases}$$

Final System:



Final System:



Final Observation: This system works well as long as what?