Helsinki University of Technology
Signal Processing Laboratory

S-38.411 Signal Processing in Telecommunications I

Spring 2000

Lecture 9: Nonlinear receivers 2: Viterbi algorithm

Prof. Timo I. Laakso

timo.laakso@hut.fi, Tel. 451 2473

http://wooster.hut.fi/studies.html

Timetable

L1 Introduction; models for channels and communication systems
L2 Channel capacity
L3 Transmit and receive filters for bandlimited AWGN channels
L4 Optimal linear equalizers for linear channels 1
L5 Optimal linear equalizers for linear channels 2
L6 Adaptive equalizers 1
L7 Adaptive equalizers 2
L8 Nonlinear receivers 1: DFE equalizers
L9 Nonlinear receivers 2: Viterbi algorithm
L10 GL1: DSP for Fixed Networks / *Matti Lehtimäki, Nokia Networks*
L11 GL2: DSP for Digital Subscriber Lines / *Janne Väänänen, Tellabs*
L12 GL3: DSP for CDMA Mobile Systems / *Kari Kalliojärvi, NRC*
L13 Course review, questions, feedback
E 24.5. (Wed) 9-12 S4 **Exam**

Contents of Lecture 9

Nonlinear receivers 1: Viterbi algorithm

I. Maximum Likelihood Sequence Detection (MLSD) in AWGN Channels
II. MLSD in linear channels
III. Viterbi algorithm

Helsinki University of Technology
Signal Processing Laboratory

I. Maximum Likelihood Sequence Detection in AWGN Channels

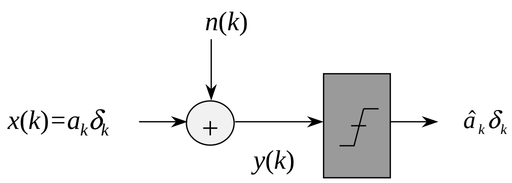
MLSD in AWGN Channels

- In the previous lecture, decision feedback (DFE) was introduced as modification to *linear* equalizer
- Reduces ISI without noise enhancement
- Basic limitations of DFE:
  - ISI cancellation results in *loss of signal energy*
  - symbol-by-symbol detection

  $\rightarrow$ DFE cannot be optimal in the sense of minimum BER
- In this lecture:

  Derive the optimal (ML) method for detecting a symbol sequence in a linear channel and find its efficient implementation using the Viterbi algorithm

MLSD in AWGN Channels...



- Let us reconsider the symbol detection problem in discrete-time *AWGN channels* (symbol-rate sampling)
- When there is no ISI, symbol-by-symbol detection *is* optimal in the sense of minimum error probability

MLSD in AWGN Channels...

- Maximize the probability that the received symbol is the right one (*posterior probability*):

$$P(\text{Symbol } A_m \text{ transmitted} \,|\, r(k) \text{ received})$$
$$= P(A_m | r(k)) = \text{MAX}$$

- Maximum *a posteriori* (MAP) criterion
- Bayes rule for conditional probabilities:

$$P(A_m | r(k)) = \frac{f(r(k) | A_m) P(A_m)}{f(r(k))}$$

- $f(r(k))$ = probability distribution of $r(k)$

MLSD in AWGN Channels...

- When the symbol probabilities $P(A_m)$ are the same for all symbols, $m = 1,\ldots,M$, MAP is the same as Maximum Likelihood (ML) criterion:

$$f(r(k) | A_m) = \text{MAX}$$

- AWGN channel: $r(k) = a_k + n(k)$
- Gaussian probability distribution:

$$f(r(k) | A_m) = \frac{1}{\sqrt{\pi N_0}} e^{-(r(k) - A_m)^2 / N_0}$$

$$\sigma_r^2 = \sigma_n^2 = N_0 / 2$$

MLSD in AWGN Channels...

- Gaussian distribution: ML criterion is equivalent to minimizing *Euclidian (quadratic) distance metric*
- Decide symbol a_k so that $(r(k) - A_m)^2 = \text{MIN}$
- Binary PAM:

$$[r(k) - (1)]^2 < [r(k) - (-1)]^2 \quad \Rightarrow \quad \hat{a}_k = 1$$
$$[r(k) - (1)]^2 \geq [r(k) - (-1)]^2 \quad \Rightarrow \quad \hat{a}_k = -1$$

- Simplified rule:

$$r(k) > 0 \quad \Rightarrow \quad \hat{a}_k = 1$$
$$r(k) \leq 0 \quad \Rightarrow \quad \hat{a}_k = -1$$

MLSD in AWGN Channels...

- Consider then the detection of a *sequence* of K symbols
  $r(k) = a_k + n(k), \; k = 1,2,\ldots,K$
- The received and transmitted sequences (signals) can be considered as K-length *vectors*

$$\mathbf{r} = [r(k) \quad r(k-1) \quad \cdots \quad r(k-K+1)]^{\mathrm{T}}$$
$$\mathbf{a}_m = [a_k \quad a_{k-1} \quad \cdots \quad a_{k-K+1}]^{\mathrm{T}}$$
$$\mathbf{n} = [n(k) \quad n(k-1) \quad \cdots \quad n(k-K+1)]^{\mathrm{T}}$$

- *Vector* AWGN channel: $\mathbf{r} = \mathbf{a}_m + \mathbf{n}$
- $m = 1 \ldots M$ (M different possible symbol vectors)

MLSD in AWGN Channels...

- The decision of a *K*-length symbol sequence can be based on choosing the best symbol vector from the possible ones
- ML criterion: maximize probability

$$f(\mathbf{r}|\mathbf{a}_m) = \prod_{l=1}^{K} f(r(k-l)|a_{k-l})$$
$$= \frac{1}{(\pi N_0)^{K/2}} e^{-\sum_{l=1}^{K}(r(k-l)-a_{k-l})^2/N_0}$$
$$= \frac{1}{(\pi N_0)^{K/2}} e^{-|\mathbf{r}-\mathbf{a}_m|^2/N_0}$$

MLSD in AWGN Channels...

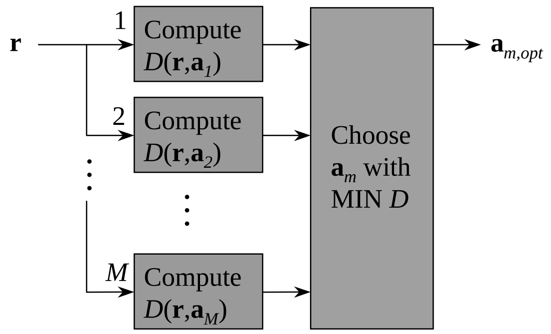
- The ML solution is equivalent to using *K*-dimensional *distance metric*: Choose the sequence $\mathbf{a}_m$ that minimizes

$$D(\mathbf{r},\mathbf{a}_m) = \sum_{l=1}^{K}\left(r(k-l) - a_{k-l}\right)^2$$
$$= |\mathbf{r} - \mathbf{a}_m|^2$$

- Implementation with a bank of *M* sequence tests
- One for each $\mathbf{a}_m$, $m = 1,\ldots, M$

MLSD in AWGN Channels...

- MLSD implementation: test all possible sequences $\mathbf{a}_m$, $m = 1,\ldots,M$ and choose the one with minimum distance metric!

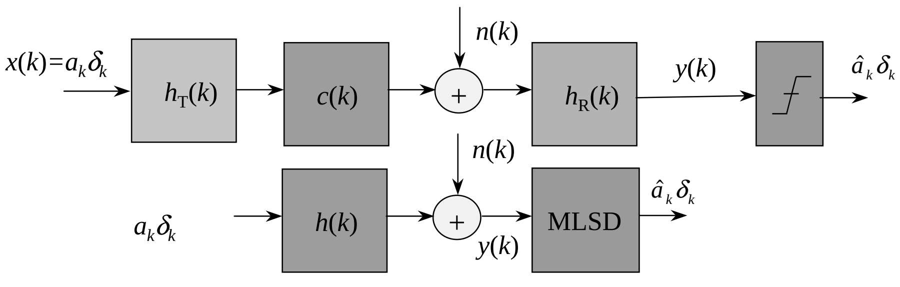


MLSD in AWGN Channels...

- Direct implementation of MLSD is laborious
- Binary PAM: $M = 2^K$ different sequences $\mathbf{a}_m$ possible
- Alternative: the computation can be made in a more efficient iterative manner (Viterbi algorithm)
- Let us study linear channel first!

II. MLSD in Linear Channels

MLSD in Linear Channels



- Combine Tx, channel and Rx filters into one $h(k)$:
- Crucial assumptions:
  1) Noise is AWGN *after* Rx filter:
  2) $h(k)$ is finite length (L) $\rightarrow$ ISI over L symbols only

MLSD in Linear Channels...

- Received discrete-time signal:

$$r(k) = h(k) * a_k + n(k)$$

- Linear filter input is Gaussian $\rightarrow$ the output is too
- The conditional probability of the received signal *vector* can be expressed as in the AWGN case:

$$f(\mathbf{r}|\mathbf{a}_m) = \frac{1}{(\pi N_0)^{K/2}} e^{-|\mathbf{r}-\mathbf{q}_m|^2/N_0}$$

where $\mathbf{q}_m$ is a K-length vector containing terms of the convolution $h(k) * a_k$

MLSD in Linear Channels...

- The ML solution for *linear* channel:

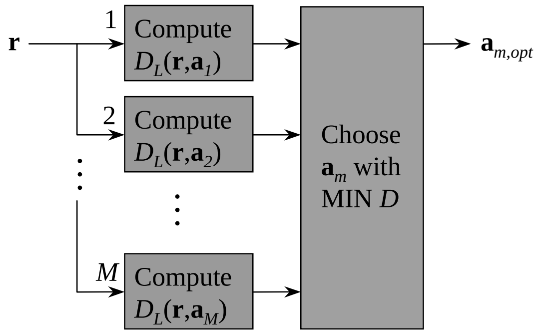
Choose the sequence $\mathbf{a}_m$ (which has a_k as its elements) that minimizes the distance metric

$$D_L(\mathbf{r}, \mathbf{a}_m) = |\mathbf{r} - \mathbf{q}_m|^2$$
$$= \sum_{l=1}^{K} \left(r(k-l) - q(k-l)\right)^2$$

where $q(k-l) = \left[h(k) * a_k\right]_{k-l}$

MLSD in Linear Channels...

- MLSD implementation for *linear* channel: test all possible *M* sequences and choose the one with minimum distance metric!



MLSD in Linear Channels...

Properties of MLSD in linear channel:

- MLSD makes a *joint decision* of a *block* of *K* symbols
- Channel estimate needed to compute the distance metric
- All the signal energy considered in the decision (energy at right symbol instant + ISI)
- Optimal in the ML sense (min BER)
- But: laborious to implement
  (binary PAM: $M = 2^K$ comparisons!)

III. Viterbi Algorithm

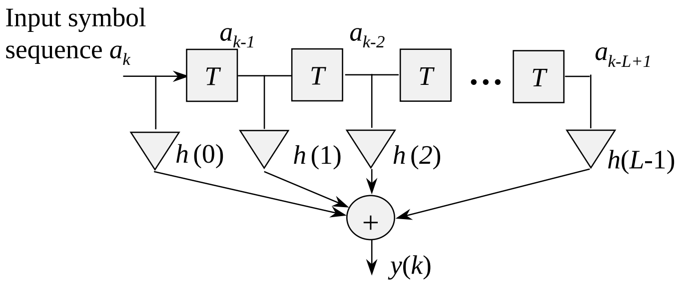
Viterbi Algorithm

- MLSD involves computation of distances between received signal vector and possible symbol sequences
- The distance computation is *redundant*: because the sequences contain same subsequences, the same squared differences are computed several times
- Strategy for reducing computations:

  1) Start computing distance metric from one end of the sequence

  2) Cancel possible subsequences on the way so that those that cannot be the best are eliminated

Viterbi Algorithm...



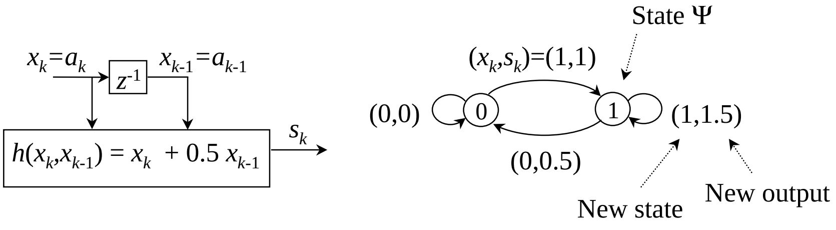
- *Finite-state machines:* a linear discrete-time channel model with L taps (FIR filter) has a *memory of length L-1*
- The next output depends on the past L values of the input
- Binary PAM: the channel has 2^{L-1} *states*

Viterbi Algorithm...

- Consider a two-tap filter channel model with impulse response (no noise):

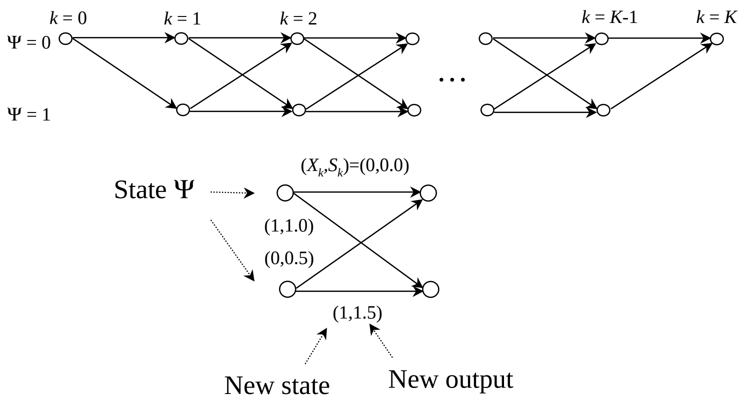
$$h(k) = \delta_k + 0.5\delta_{k-1}$$

- *Markov model* for state transitions:



Viterbi Algorithm...

- *Trellis diagram* for the 2-tap channel model:



Viterbi Algorithm...

- The trellis contains 2^{L-1} = 2 different *states*
- The sequence is K symbols long
- There are $M = 2^K$ different possible sequences, of which the one closest to the received sequence should be found
- Each possible symbol sequence corresponds to a certain *path* in the trellis, which has a certain *length* (= distance from the received (sub)sequence)
- Each connection of two states is a *branch* which has a certain length (= increase in total length of path)

Viterbi Algorithm...

- The problem of Viterbi algorithm:

  How to use the trellis to search the received signal sequence once and find the optimum ML symbol sequence with minimum number of computations?
- Algorithm:
  - proceed symbol by symbol and compute length of new branches
  - determine the overall lengths of remaining paths
  - cancel unnecessary paths and keep *surviving* paths only

Viterbi Algorithm...

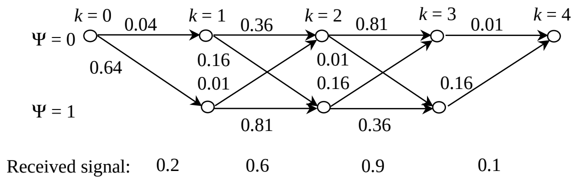
- How many paths need to be stored at each step to find the optimum solution?
- Answer: N paths for a system with N states

  (Binary PAM, L-tap channel: $N = 2^{L-1}$)

Viterbi Algorithm...

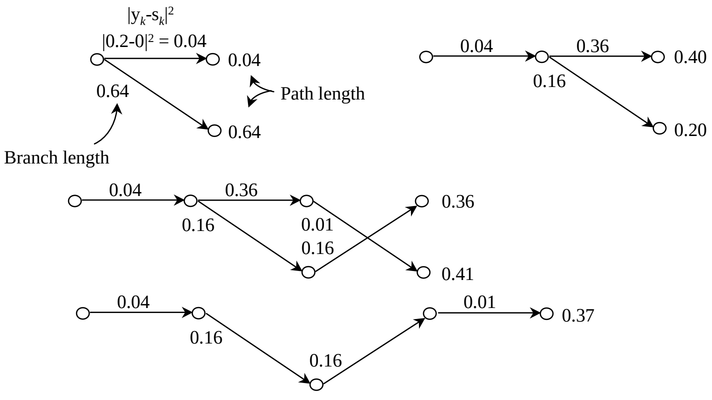
Example (No. 9-25 Lee-Messerschmitt):

- 2-tap filter, channel $h(k) = \delta_k + 0.5\ \delta_{k-1}$
- Sequence length $K = 4$



Viterbi Algorithm...

- This is how it goes:



Decisions 0 1 0 0

Viterbi Algorithm...

- Basic Viterbi gives the detected sequence only after processing the whole sequence of K symbols (long delay!)
- The early symbols usually do not change after processing a certain number (ca. $5L$) symbols
- Modification: decide early symbols after processing up to the *truncation depth d* (<<K)
  - reduces delay and computations
  - suboptimal solution in general

Viterbi Algorithm...

- Viterbi algorithm is (almost) optimal ML solution (better than linear or DFE equalizer)
- Viterbi algorithm can be applied when
  - the delay of block processing is acceptable
  - the complexity $(L\text{-}1)^K$ is tolerable (short enough impulse response)
- Example: GSM mobile phone receiver uses Viterbi for K = 148 bit block reception (26 bit training sequence in the middle, 2 x 58 bits data, 2x3 extra bits).

Summary

Today we discussed:

Viterbi algorithm

I. Maximum Likelihood Sequence Detection (MLSD) in AWGN Channels

II. MLSD in linear channels

III. Viterbi algorithm

Next week: Guest lecture!