TSKS04 Digital Communication

Continuation Course

Lecture 7

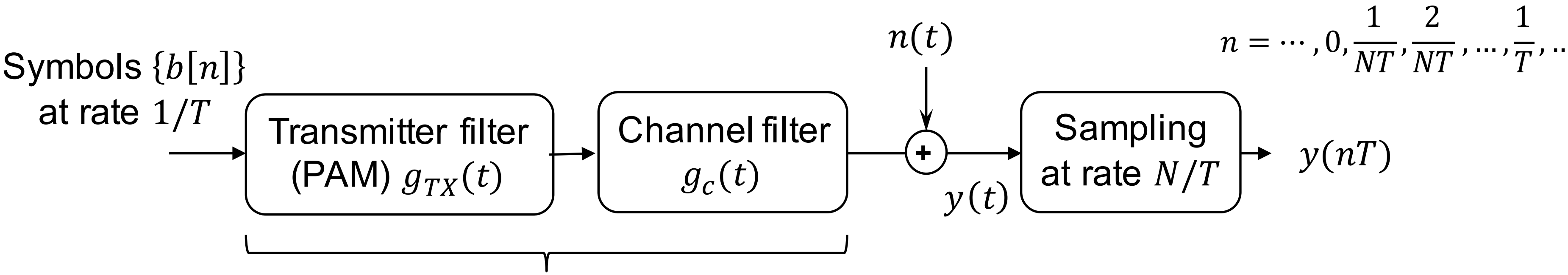
Channel Equalization, ML Sequence Estimation

Emil Björnson

Department of Electrical Engineering (ISY)

Division of Communication Systems

Channel Equalization (1/4)



Effective pulse shape:
$p(t) = (g_{TX} * g_c)(t)$

Received analog signal:

$$y(t) = \sum_m b[m]p(t - mT) + n(t)$$

How to detect $\{b[n]\}$ from the received signal?

Terminology: *Channel equalization*

- Reverse distortion caused by channel
- In particular: Inter-symbol interference and noise

Channel Equalization (2/4)

Hypothesis: Sequence $\{b[n]\}$

- Step 1: Filter $y(t)$ with $h_{RX}(t) = p^*(-t)$ (a matched filter)
- Step 2: Sample $(y * h_{RX})(t)$ at $t = nT$ to get $z[n]$
- Step 3: Correlate with hypothesis sequence $\{b[n]\}$

ML criterion:

$$\hat{\boldsymbol{b}} = \mathrm{argmax}_{\boldsymbol{b}} \ln L(y|\boldsymbol{b})$$

where

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\left(\sum_m Re\{z[m]b^*[m]\} - \frac{\|s_{\boldsymbol{b}}\|^2}{2}\right)$$

Channel Equalization (3/4)

Have solved the equalization problem?

- Yes, we can try all $\{b[n]\}$ and see which one that maximizes $L(y|\boldsymbol{b})$

Example: M-ary constellation and sequence $b[0], \ldots, b[L-1]$

- M^L different sequences
- $L = 10$ and $M = 16$: $16^{10} \approx 1.1$ trillion sequences

Unreasonable complexity (and delay)!

What can we do to reduce complexity?

Channel Equalization (4/4)

ML criterion:

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\left(Re\{< y, s_b >\} - \frac{\|s_{\boldsymbol{b}}\|^2}{2}\right)$$

Can we update ML rule symbol by symbol?

- First term: $Re\{< y, s_b >\} = \sum_m Re\{z[m]b^*[m]\}$ Yes!
- Second term:

$$\|s_{\boldsymbol{b}}\|^2 = \sum_m \left(|b[m]|^2 h[0] + \sum_{n:n<m} 2Re\{b[n]b^*[m]h(m-n)\} \right)$$

where $h(n) = r_p(Tn)$ and $r_p(\tau) = \int_{-\infty}^{\infty} p(t)p^*(t-\tau)dt$ Yes!

Iterative ML Sequence Estimation (1/2)

ML Estimation of $\{b[n]\}$

- Assume that $h(n) \neq 0$ only for $n = 0, \pm 1, \ldots, \pm L$
- Maximize the log-likelihood

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\sum_m \underbrace{\left(Re\{z[m]b^*[m]\} - \frac{|b[m]|^2 h[0]}{2} - \sum_{n=m-L}^{m-1} Re\{b[n]b^*[m]h(m-n)\} \right)}$$

Call this: $\lambda_m(b[m], s[m])$

Current symbol

L *previous symbols:*
$s[m] = \{b[m-1], \ldots, b[m-L]\}$

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\sum_m \lambda_m(b[m], s[m])$$

Iterative ML Sequence Estimation (2/2)

Log-likelihood function:

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\sum_m \lambda_m(b[m], s[m])$$

- Current "state" $s[m] = \{b[m-1], \dots, b[m-L]\}$
- Interpretation: $\lambda_m(b[m], s[m])$ likelihood of different choices of $s[m+1]$

Alternative notation: $\lambda_m(b[m], s[m]) = \lambda_m(s[m] \rightarrow s[m+1])$

$$\ln L(y|\boldsymbol{b}) = \frac{1}{\sigma^2}\sum_m \lambda_m(s[m] \rightarrow s[m+1])$$

Viterbi Algorithm

Efficient implementation of ML sequence estimation (MLSE)

- Uses principle of optimality
  - If two or more sequences reach a certain state $s[m]$, only the most likely is of interest
  - Add and compare steps
- We only need to save one sequence per state: “the survivor”

Complexity comparison (${N}$-length sequence)

- Check all sequences: $N \cdot M^N$ λ_m computations
- Viterbi algorithm: $N \cdot (M^L)^2$ λ_m computations

LINKÖPING UNIVERSITY

www.liu.se