Department of Electrical Engineering
University of Arkansas

# ELEG 5693 Wireless Communications
# Ch. 5 Equalization

**Dr. Jingxian Wu**
wuj@uark.edu

# OUTLINE

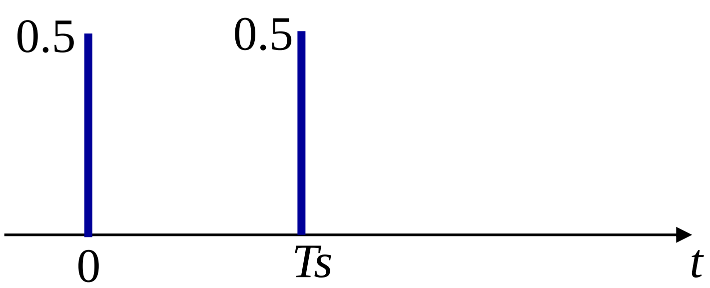
- **Introduction**
- **Maximum Likelihood Sequence Estimation (MLSE)**

# INTRODUCTION

- **Frequency selective fading**
  - Frequency domain: Signal bandwidth >> channel coherence bandwidth
  - Time domain: symbol period << rms delay spread
    - Relative arrival time of the multipath components is no longer negligible!

$$h(t,\tau)=\sum_{l=1}^{L}h_l(t)\times\delta(\tau-\tau_l)$$

    - $t$: time variation, $\tau$ : relative delay between multipath component.
  - Power delay profile:
    - The relative power and delay of the multipath component.
    - E.g.




- If the first multipath component arrives at receiver at time 0, then the second multipath component arrives at receiver at time *Ts*.

# INTRODUCTION: ISI

- **Intersymbol interference (ISI)**
  - E.g. two resolvable multipath components with relative delay *Ts*.
    - 1st multipath component: $h_0(1), h_0(2), h_0(3), \cdots, h_0(k), h_0(k+1), \cdots$
    - 2nd multipath component: $h_1(1), h_1(2), h_1(3), \cdots, h_1(k), h_1(k+1), \cdots$
  - Tx symbol:

$$x_1, x_2, x_3, \cdots, x_k, x_{k+1}, \cdots$$

  - In the channel

    the 1st multipath: $h_0(1)x_1, h_0(2)x_2, h_0(3)x_3, \cdots$

    the 2nd multipath: $0, h_1(2)x_1, h_1(3)x_2, h_1(4)x_3, \cdots,$

  - Rx symbols are the combination of the two multipath components

$$y_1 = h_0(1)x_1 + 0 + n_1$$

$$y_2 = h_0(2)x_2 + h_1(2)x_1 + n_2$$

AWGN

$$y_3 = h_0(3)x_3 + h_1(3)x_2 + n_3$$

$$y_k = h_0(k)x_k + h_1(k)x_{k-1} + n_k$$

# INTRODUCTION: ISI

- **ISI (Cont'd)**

$$y_k = h_0(k)x_k + h_1(k)x_{k-1} + n_k$$

  - The received symbol is a combination of two or more information symbols!
    - $x_k$ : desired symbol. $x_{k-1}$ : Interference from multipath component.
  - Intersymbol interference:
    - Interference caused by multipath components of frequency selective fading.

- **ISI model**
  - *L* resolvable multipath components with relative delay *Ts*.

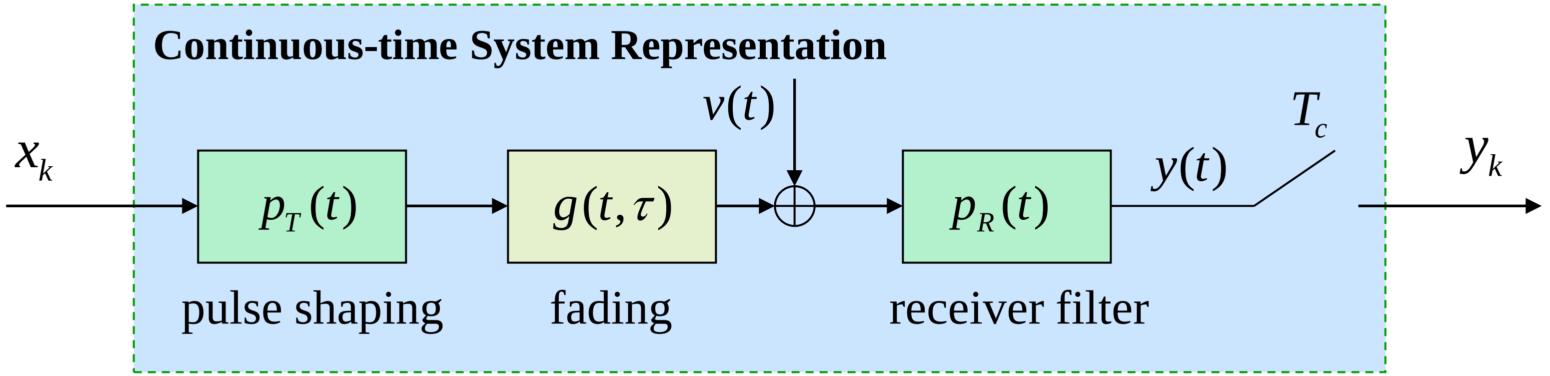
$$y_k = h_0(k)x_k + h_1(k)x_{k-1} + \cdots + h_{L-1}(k)x_{k-L+1} + n_k$$

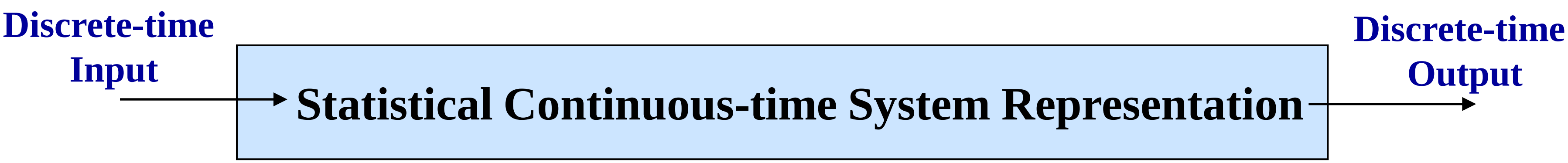
$$y_k = \sum_{l=0}^{L-1} h_l(k)x_{k-l} + n_k$$

UNIVERSITY OF ARKANSAS

# INTRODUCTION: DISCRETE-TIME MODEL

- **What if the relative delay between multipath component is not an integer multiply of symbol period $T_s$ ?**
  - Discrete-time model can convert the channel impulse response into desired form
- **Discrete-time model**




**Can we replace the continuous-time system with an equivalent discrete-time system without changing system input-output relationship?**

UNIVERSITY OF ARKANSAS

# INTRODUCTION: DISCRETE-TIME MODEL

- **Composite impulse response**

$$h(t,\tau) = p_T(\tau) \otimes g(t,\tau) \otimes p_R(\tau)$$

$v(t) \otimes p_R(t)$

– System model

$$x(t) = \sum_{k=-\infty}^{+\infty} x_k \delta(t-kT_s), \qquad y(t) = x(t) \otimes h(t,\tau) = \sum_{k=-\infty}^{+\infty} x_k h(t,\tau-kT_s) + n(t)$$

– After sampling

$$y(mT_s) = \sum_{k=-\infty}^{+\infty} x_k h(mT_s,(m-k)T_s) + n(mT_s)$$

– Represent with discrete-time variables

$$h_l(m) = h(mT_s, lT_s)$$

$$y_m = \sum_{k=-\infty}^{+\infty} x_k h_{m-k}(m) + n_m$$

$$y_m = \sum_{l=0}^{L} h_l(m) x_{k-l} + n_m$$

UNIVERSITY OF ARKANSAS

# INTRODUCTION: EQUALIZATION

- **What is equalization?**
  - Signal processing operation employed at receiver to mitigate the effects of ISI.
  - Adaptive equalization
    - Time-varying frequency-selective fading ➔ Channel impulse response varies with time
    - The equalizer should vary with time as well! ➔ adaptive equalization.
- **A possible equalization method:**
  - At time 1, $y_1 = h_0(1)x_1 + 0 + n_1$ (no ISI) ➔ $\hat{x}_1 = y_1 / h_0(1)$
  - At time 2, $y_2 = h_0(2)x_2 + h_1(2)x_1 + n_2$ ➔ $\hat{x}_2 = \left[y_2 - h_1(2)\hat{x}_1\right] / h_0(1)$
  - At time 3, $y_3 = h_0(3)x_3 + h_1(3)x_2 + n_3$ ➔ $\hat{x}_3 = \left[y_3 - h_1(3)\hat{x}_2\right] / h_0(3)$

Problem: If $\hat{x}_1$ is in error, all the remaining symbols will be affected!
Error propagation.

# INTRODUCTION: CLASSIFICATION

- **Classification**
  - Linear equalization: only linear operations are employed in equalizer.
    - Zero Forcing (ZF), least mean square error (LMS), minimum mean square error (MMSE), etc
    - Advantages: simple
    - Not commonly used in wireless communication system.
  - Non-linear equalization:
    - Decision feedback equalization (DFE), Maximum Likelihood Sequence Estimation (MLSE), Delayed Decision Feedback Equalization (DDFE), Reduced State Sequence Estimation (RSSE), Turbo Equalization, ……
    - More complex than linear equalization
    - Better performance
    - Widely used in wireless communication systems.
- **MLSE is the optimum equalizer**
  - It has the best performance among all the equalizers
  - It has the highest computational complexity.

# OUTLINE

- Introduction

- **Maximum Likelihood Sequence Estimation (MLSE)**

# MLSE

- **System model**
  - Symbols are transmitted in block (assume N symbols per block)

$$\mathbf{x} = [x_1, x_2, \cdots, x_N]$$

  - The *k*th symbol at receiver

$$y_k = \sum_{l=0}^{L-1} h_l(k) x_{k-l} + n_k$$

  - $n_k$ is AWGN with variance $\sigma^2$ ➔ $y_k - \sum_{l=0}^{L-1} h_l(k) x_{k-l}$ is Gaussian distributed

$$p(y_k \mid \mathbf{x}) = p(n_k) = \frac{1}{\pi\sigma^2} \exp\left[ -\frac{1}{\sigma^2} \left| y_k - \sum_{l=0}^{L-1} h_l(k) x_{k-l} \right|^2 \right]$$

# MLSE

- **Likelihood function**
  - White noise ➔ $n_j$ is independent of $n_i$ ➔ $p(n_i, n_j) = p(n_i)\,p(n_j)$

$$p(y_1, y_2, \cdots, y_N \mid \mathbf{x}) = p(n_1)\,p(n_2) \times \cdots \times p(n_N)$$

$$= \frac{1}{(\pi\sigma^2)^N} \prod_{k=1}^{N} \exp\left[-\frac{1}{\sigma^2}\left|y_k - \sum_{l=0}^{L-1} h_l(k)\,x_{k-l}\right|^2\right]$$

$$= \frac{1}{(\pi\sigma^2)^N} \exp\left[-\frac{1}{\sigma^2} \sum_{k=1}^{N}\left|y_k - \sum_{l=0}^{L-1} h_l(k)\,x_{k-l}\right|^2\right]$$

- **Maximum Likelihood**
  - Find $\mathbf{x} = [x_1, x_2, \cdots, x_N]$ that maximize $p(y_1, y_2, \cdots, y_N \mid \mathbf{x})$
  - Equivalent to: minimize

$$J = \sum_{k=1}^{N}\left|y_k - \sum_{l=0}^{L-1} h_l(k)\,x_{k-l}\right|^2$$

# MLSE: COST FUNCTION

- **Maximum Likelihood Sequence Estimation**
  - Based on the information of the received symbols $\mathbf{y} = [y_1, y_2, \cdots, y_N]$ and the channel impulse response, find the sequence $\mathbf{x} = [x_1, x_2, \cdots, x_N]$ that minimizes the following cost function

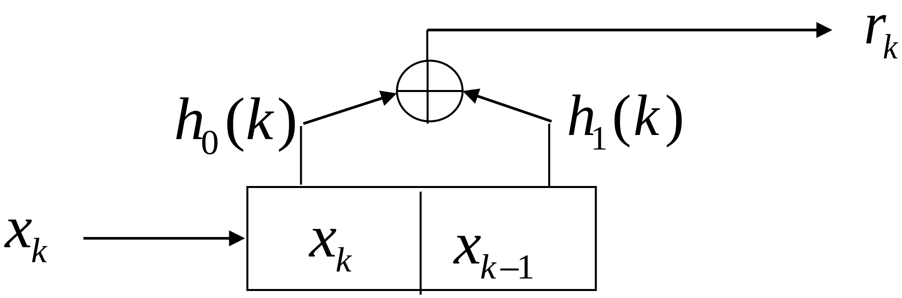
$$\boxed{J = \sum_{k=1}^{N} \left| y_k - r_k \right|^2} \qquad \text{where} \quad r_k = \sum_{l=0}^{L-1} h_l(k) x_{k-l}$$

- **Most direct way: exhaustive searching**
  - M-ary modulation
    - Modulation constellation size: $M$ ➔ each $x_k$ can take one of $M$ possible values.
  - $N$ symbols per block
  - There are totally possibilities of $\mathbf{x} = [x_1, x_2, \cdots, x_N]$
  - If N = 100, M = 8, then we need to search possibilities

# MLSE: STATE REPRESENTATION

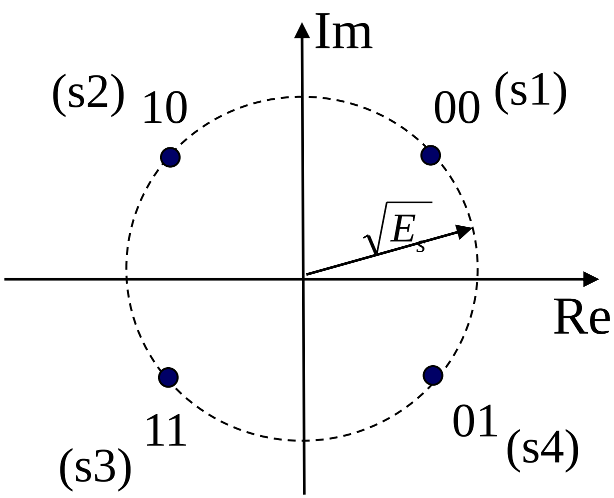
- **State representation of frequency selective fading**

$$y_k = h_0(k)x_k + h_1(k)x_{k-1} + n_k = r_k + n_k$$




Cost function: $$J = \sum_{k=1}^{N} \left| y_k - r_k \right|^2$$

- **E.g. QPSK**
  - How many states?



# MLSE: STATE REPRESENTATION

- **State representation**
  - E.g. QPSK (four symbols: s1, s2, s3, s4), L = 2
    - M^(L-1) = 4 states: a: (s1), b: (s2), c: (s3), d: (s4).

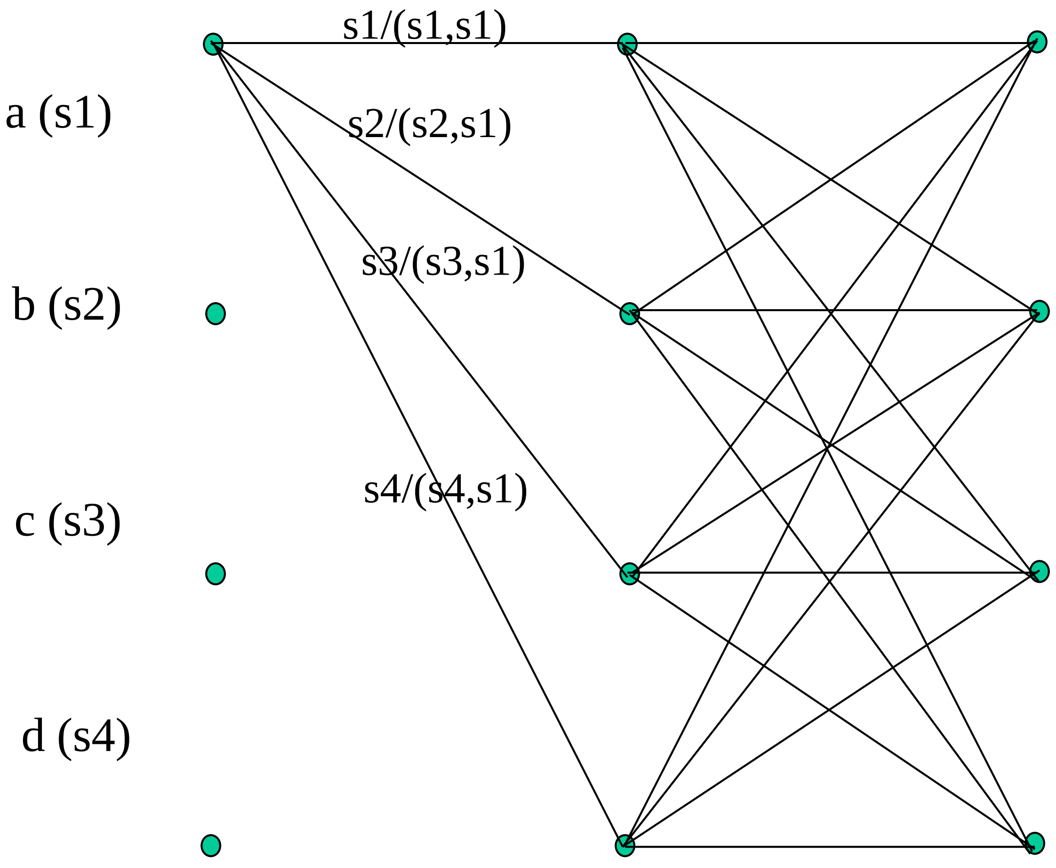
| Input | Current state $x_{k-1}$ now | Register contents | Next state $x_{k-1}$ after shift |
|---|---|---|---|
| s1 | a (s1) | (s1, s1) | a (s1) |
| s1 | b (s2) | (s1, s2) | a (s1) |
| s1 | c (s3) | (s1, s3) | a (s1) |
| s1 | d (s4) | (s1, s4) | a (s1) |
| s2 | a (s1) | (s1, s1) | b (s2) |
| s2 | b (s2) | (s1, s2) | b (s2) |
| s2 | c (s3) | (s1, s3) | b (s2) |
| s2 | d (s4) | (s1, s4) | b (s2) |

# MLSE: STATE REPRESENTATION

- **State representation**

| Input | Current state $x_{k-1}$ now | Register contents | Next state $x_{k-1}$ after shift |
|---|---|---|---|
| s3 | a (s1) | (s3, s1) | c (s3) |
| s3 | b (s2) | (s3, s2) | c (s3) |
| s3 | c (s3) | (s3, s3) | c (s3) |
| s3 | d (s4) | (s3, s4) | c (s3) |
| s4 | a (s1) | (s4, s1) | d (s4) |
| s4 | b (s2) | (s4, s2) | d (s4) |
| s4 | c (s3) | (s4, s3) | d (s4) |
| s4 | d (s4) | (s4, s4) | d (s4) |

# MLSE: TRELLIS REPRESENTATION




input/(register contents)

# MLSE: TRELLIS REPRESENTATION

- **Example**
  - For a BPSK system with channel length three, how many states do we have? Draw the trellis diagram.

# MLSE: VITERBI ALGORITHM

- **Viterbi algorithm**
  - Each state transition corresponds to one pair of $(x_k, x_{k-1})$
    - E.g. (state a → state c) ➔ $(x_k = s3, x_{k-1} = s1)$
  - Thus, for each state transition, we can calculate

$$r_k = h_0(k)x_k + h_1(k)x_{k-1}$$

  - For each transition, we can find out the Euclidean distance between $y_k$ and $r_k$

$$J_k = \left|y_k - r_k\right|^2 = \left|y_k - h_0(k)x_k - h_1(k)x_{k-1}\right|^2$$

  - Utilizing the trellis diagram, find out the path with the minimum accumulated Euclidean distance

$$J = \sum_{k=1}^{N} J_k = \sum_{k=1}^{N} \left|y_k - r_k\right|^2$$

# MLSE: VITERBI ALGORITHM

- **Example**
  - BPSK $s_1 = -1$ $s_2 = 1$
  - Channel

| | | |
|---|---|---|
| $h_0(1)$ <br> $0.3+0.5j$ | $h_0(2)$ <br> $0.4+0.6j$ | $h_0(3)$ <br> $0.5+0.7j$ |
| $h_1(1)$ <br> $0.8+0.4j$ | $h_1(2)$ <br> $0.7+0.2j$ | $h_1(3)$ <br> $0.6+0.3j$ |

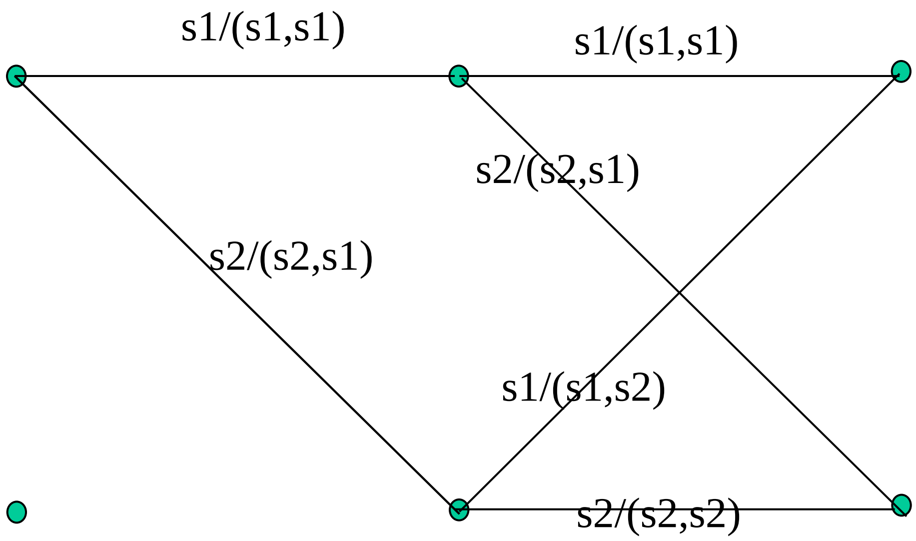
  - Rx symbols

| $y_1$ | $y_2$ | $y_3$ |
|---|---|---|
| $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |

  - Using Viterbi algorithm finding out what are the sending symbols $x_1$ $x_2$ $x_3$ ?

# MLSE: VITERBI ALGORITHM

- **Trellis diagram**
  - # of states M^(L-1) = 2^(2-1) = 2



# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |


a: (s1, s1) → $r_1 = -1\cdot h_0 - 1\cdot h_1 = -1.1-0.9j$

$$J_1 = |y_1 - r_1|^2 = |0-0.3j-(-1.1-0.9j)|^2 = 2.65$$

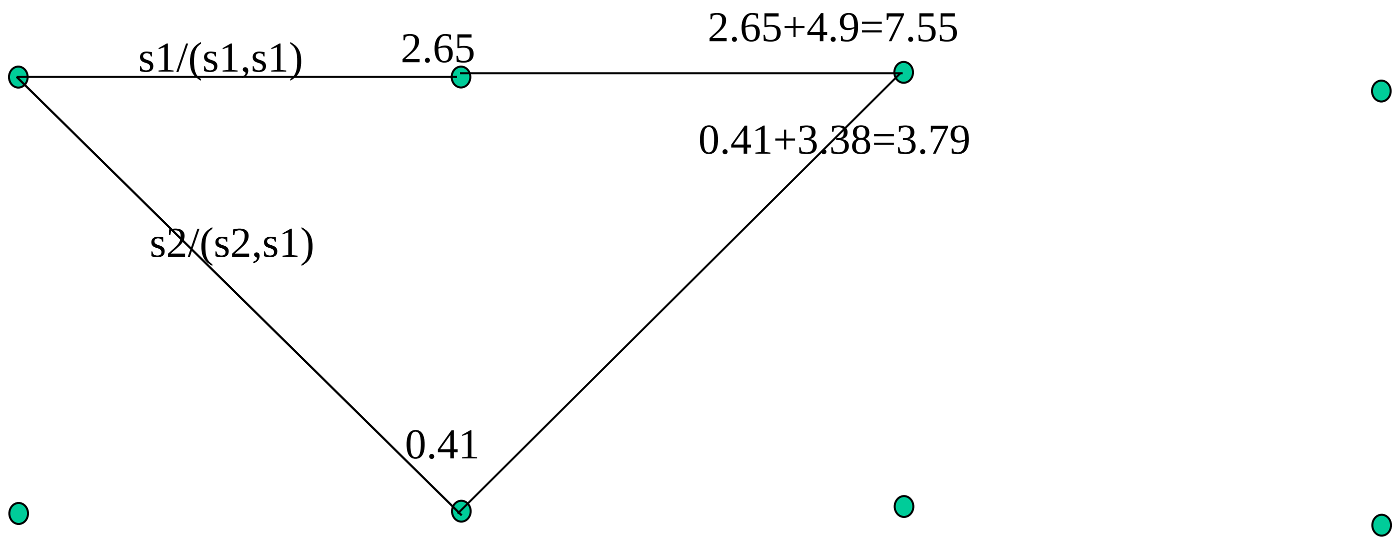
b: (s2, s1) → $r_1 = 1\cdot h_0 - 1\cdot h_1 = -0.5+0.1j$

$$J_1 = |y_1 - r_1|^2 = 0.41$$

UNIVERSITY OF ARKANSAS

# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




a→a: (s1, s1) → $r_1 = -h_0 - h_1 = -1.1 - 0.8j$

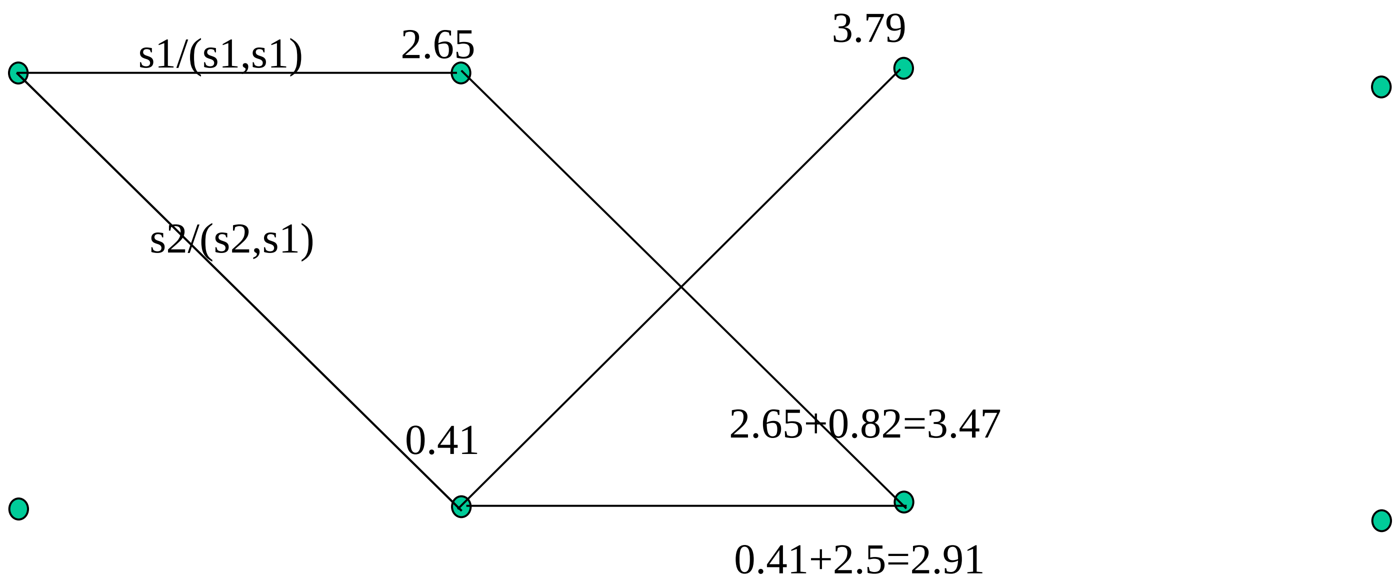
$$J_1 = \left|y_1 - r_1\right|^2 = \left|-0.4 + 1.3j - (-1.1 - 0.8j)\right|^2 = 4.9$$

b→a: $r_1 = -h_0 + h_1 = 0.3 - 0.4j$

$$J_1 = \left|-0.4 + 1.3j - (0.3 - 0.4j)\right|^2 = 3.38$$

# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




a→b: (s2, s1) → $r_1 = h_0 - h_1 = -0.3+0.4j$

$$J_1 = |y_1 - r_1|^2 = |-0.4+1.3j-(-0.3+0.4j)|^2 = 0.82$$

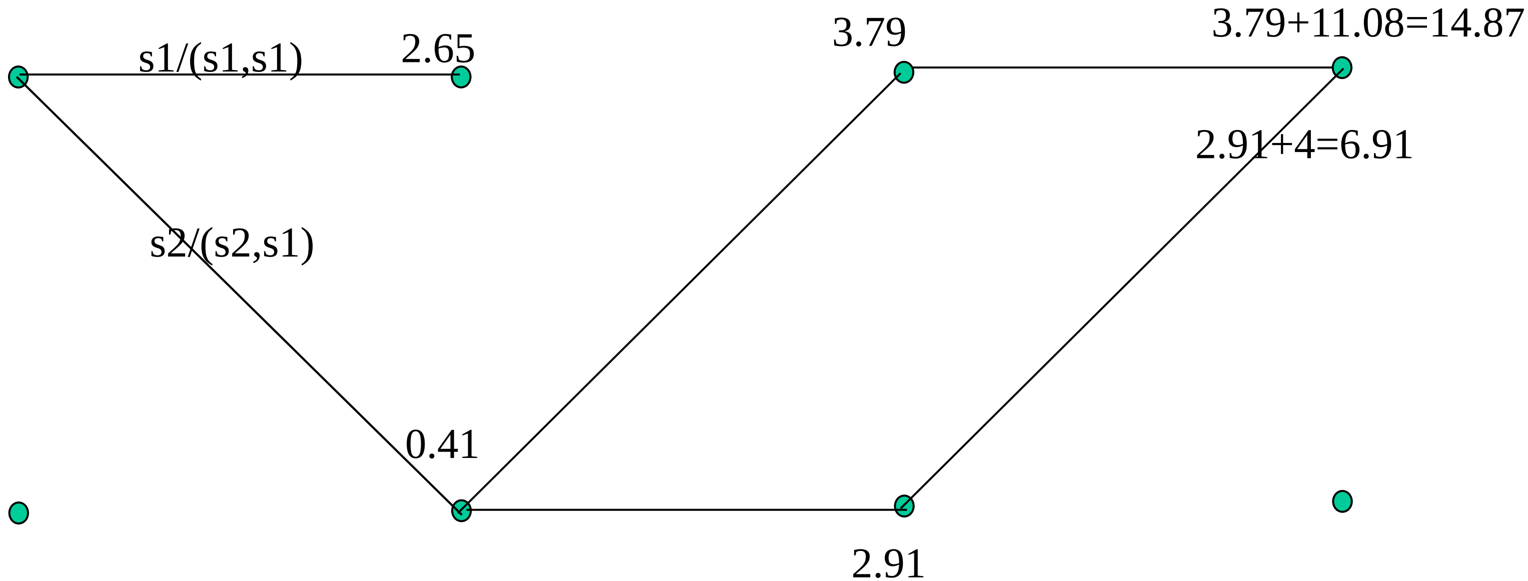
b→a: $r_1 = h_0 + h_1 = 1.1+0.8j$

$$J_1 = |-0.4+1.3j-(1.1+0.8j)|^2 = 2.5$$

UNIVERSITY OF ARKANSAS

# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




a→b: (s2, s1) → $r_1 = -h_0 - h_1 = -1.1-1.0j$

$$J_1 = |y_1 - r_1|^2 = |1.7+0.8j-(-1.1-1.0j)|^2 = 11.08$$

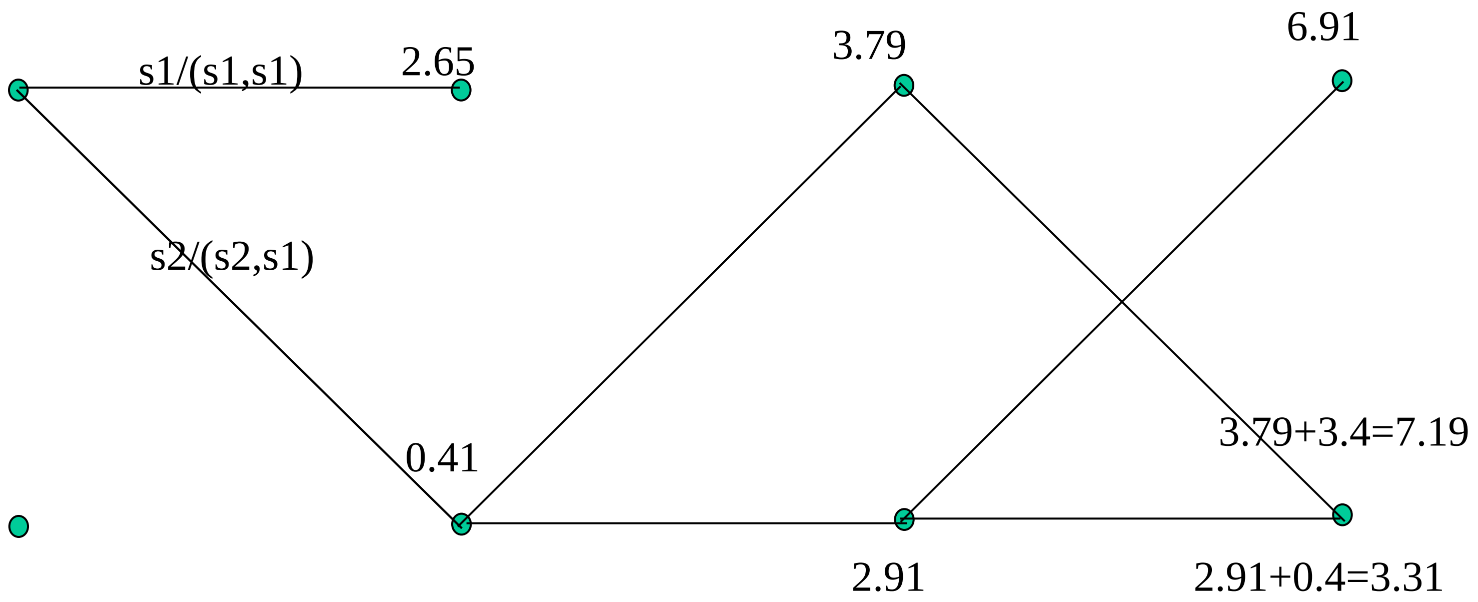
b→a: $r_1 = -h_0 + h_1 = 0.1-0.4j$

$$J_1 = |1.7+0.8j-(0.1-0.4j)|^2 = 4$$

UNIVERSITY OF ARKANSAS

# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




a→b: (s2, s1) → $r_1 = h_0 - h_1 = -0.1+0.4j$

$$J_1 = |y_1 - r_1|^2 = |1.7+0.8j-(-0.1+0.4j)|^2 = 3.4$$

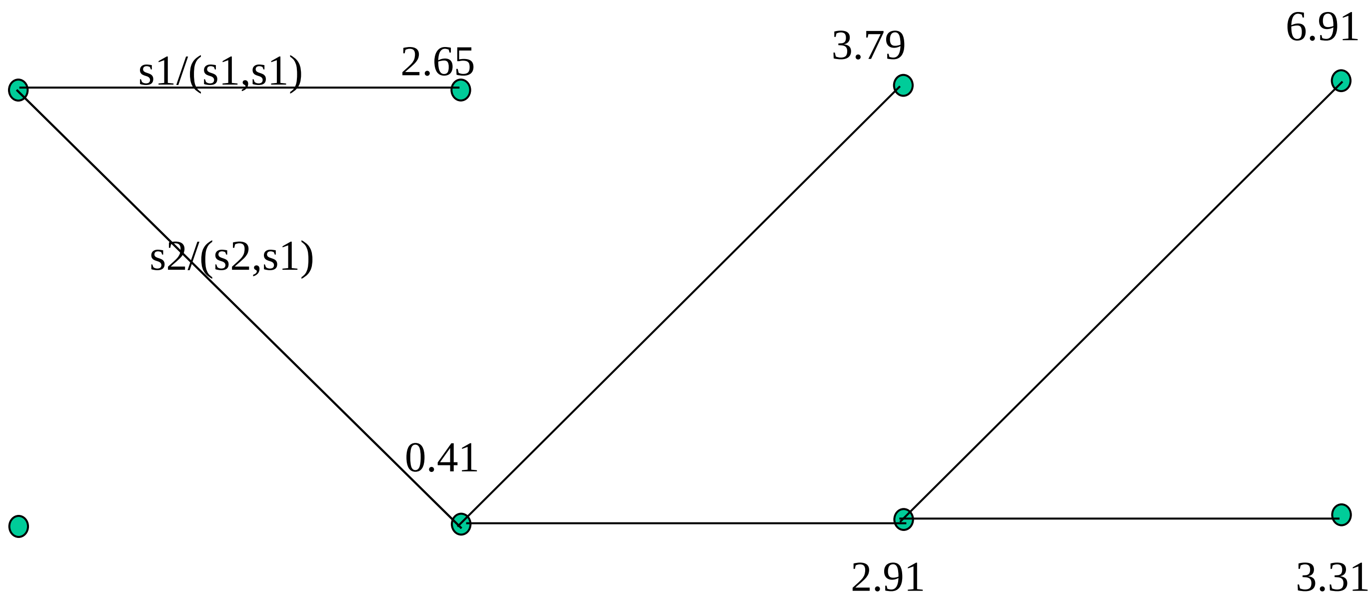
b→a: $r_1 = h_0 + h_1 = 1.1+1j$

$$J_1 = |1.7+0.8j-(1.1+1j)|^2 = 0.4$$

UNIVERSITY OF ARKANSAS

# MLSE: VITERBI ALGORITHM

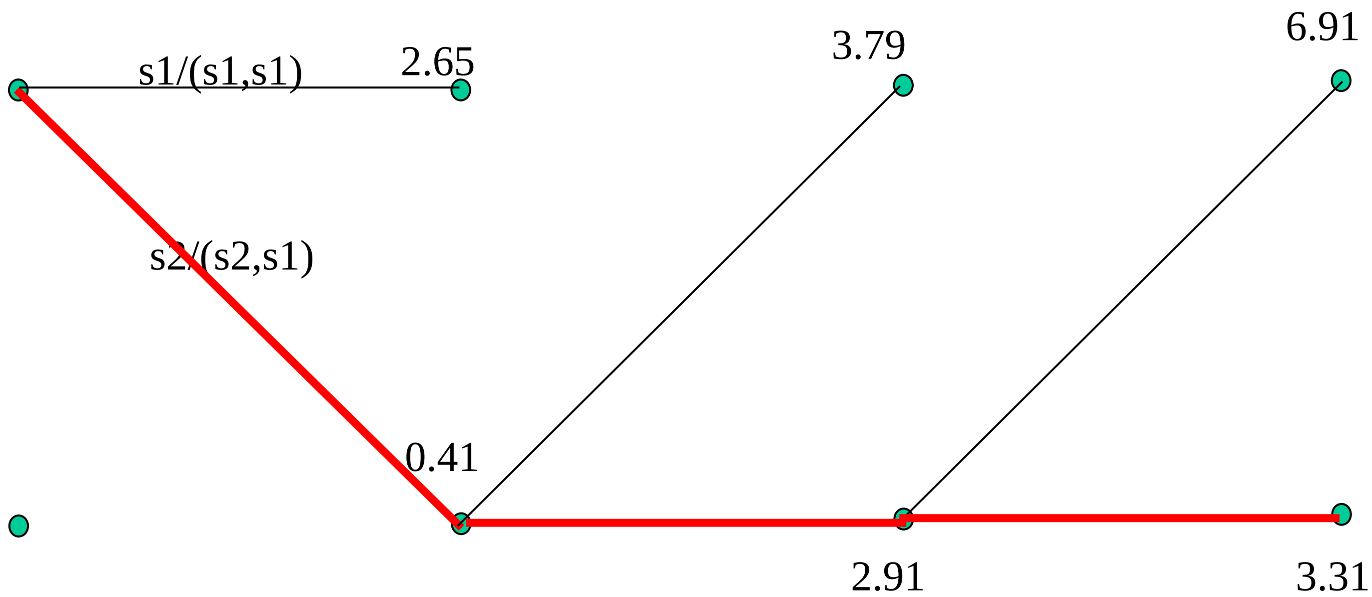
| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




UNIVERSITY OF ARKANSAS

# MLSE: VITERBI ALGORITHM

| | | | |
|---|---|---|---|
| $\mathbf{h}_0$ | $0.3+0.5j$ | $0.4+0.6j$ | $0.5+0.7j$ |
| $\mathbf{h}_1$ | $0.8+0.4j$ | $0.7+0.2j$ | $0.6+0.3j$ |
| $\mathbf{y}$ | $0-0.3j$ | $-0.4+1.3j$ | $1.7+0.8j$ |




**1** **1** **1**

# MLSE

- **States**
  - M = 4, L = 3
    - # of states:
  - M = 8, L = 2
    - # of states:
  - M = 8, L = 3
    - # of states:
  - M = 8, L = 4
    - # of states:

**# of states = M^(L-1)**