08 November 2016 06:31

Running Example.

$$h(m) = \begin{cases} 3/2, & m = 0 \\ -1/2, & m = 1 \\ -1/2, & m = -1 \\ 0, & \text{otherwise} \end{cases}$$

$\boxed{L=1}$

BPSK Modulation $b(n) \in \{+1, -1\}$

$M = 2$.

No. of states $M^L = 2$

$$S(n) = [b(n-1)]$$

Branch metric

Branch metric

$$\lambda_n(b(n), s(n)) = \lambda_n(S_n \to S_{n+1}) = \mathrm{Re}\{ b^*(n) z(n) \} - \frac{h(0)}{2} |b(n)|^2$$
$$- \mathrm{Re}\{ h(1) \; b^*(n) \, b(n-1) \}$$

$Z(n) \to$ Matched filter output

Since BPSK, ony $\mathrm{Re}\{Z(n)\} \equiv y(n)$ is relevent

Also $b(n)$ are real

$|b(n)| = 1$ for all n

$\frac{h(0)}{2} |b(n)| =$ constant for all n

$\Downarrow$

can be ignored in argmax

Modified (Equivalent branch metric)

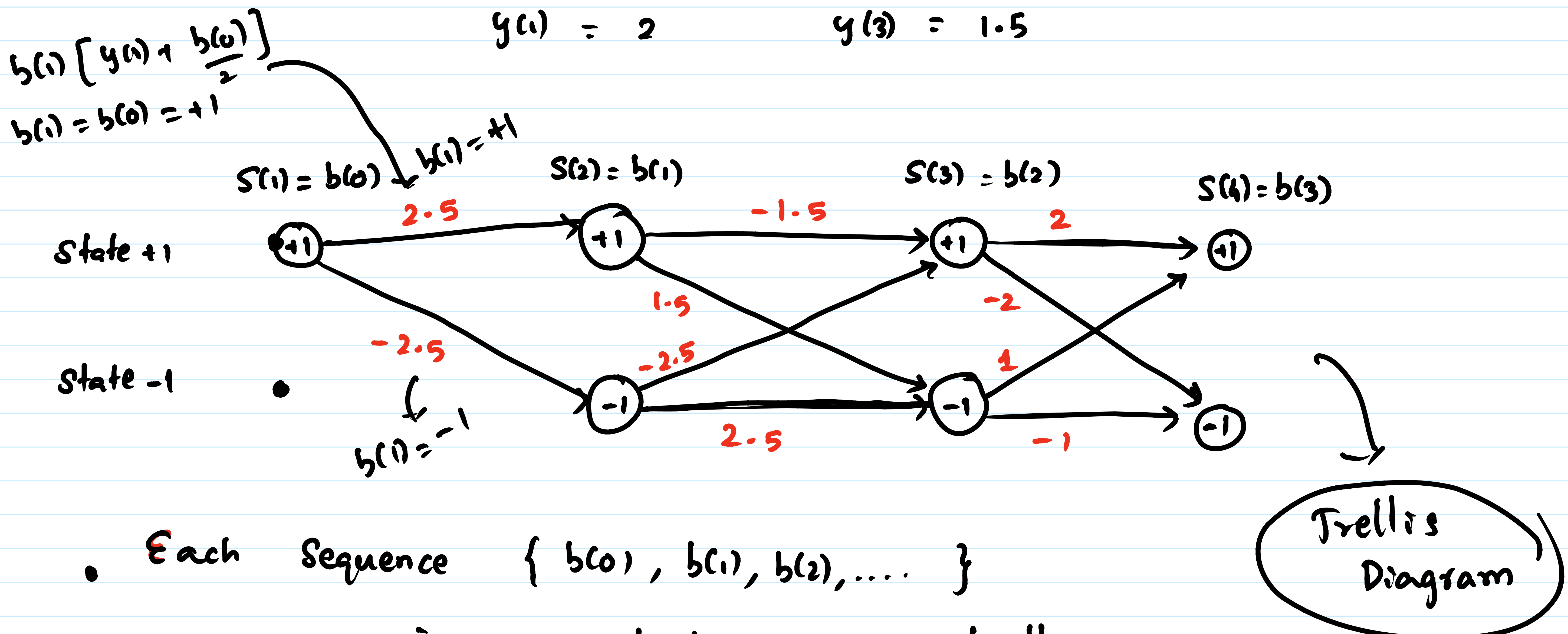
$$m_n(b(n), S_n) = m_n(S_n \to S_{n+1}) = b(n) \left[y(n) + \frac{1}{2} b(n-1) \right]$$

$$m_n(b(n), S_n) = m_n(S_n \to S_{n+1}) = b(n)\left[y(n) + \frac{1}{2}b(n-1)\right]$$

Suppose we start with initialization $b(0) = +1$

Also let $y(0) = -1$ $\quad y(2) = -2$

$y(1) = 2$ $\quad y(3) = 1.5$



- Each Sequence $\{b(0), b(1), b(2), \ldots\}$ is a path in the trellis

- Each state transition has a metric (cost)
  $\lambda_n(S_n \rightarrow S_{n+1})$

- MLSE $\rightarrow$ Find the path along the trellis with the highest metric

Note : M branches come out of each state

M branches merge at each state

Consider (partial) sequences merging at a state

Example $S(3) = +1$

(partial)
Two sequences $\{ \overset{b(0)}{+1}, \overset{b(1)}{+1}, \overset{b(2)}{+1} \}$ merge at $S(3) = +1$

$\{+1, -1, +1\}$

Accumulated metric

upto time 2

$+1, +1, +1 \qquad +2.5 - 1.5 = 1$

$+1, -1, +1 \qquad -2.5 - 2.5 = -5$

Partial sequences merging at a state

can be <u>compared</u> <u>directly</u>

↳ Partial seq. with smaller metric

can <u>not</u> be MLSE

& can be removed from

further consideration

SURVIVOR : Partial Sequence with highest metric at each state

+1, +1, +1 is survivor at state $s(3) = +1$

Number of survivors at each time

= Number of states $= M^L$

Need to "extend" only the survivors further along the trellis

Just like fixing initialization $(b(0) = +1)$

we can also fix final $(b(N) = +1)$

Rest $b(1), b(2), \ldots b(N-1)$ depend on data

Fixing $b(N) = +1$, fixes the "ending" state

SURVIVOR at the END state is ML Solution (MLSE)

Complexity $= N \cdot M^L$

linear in block length

linear in number of states (exponential in channel memory length L)

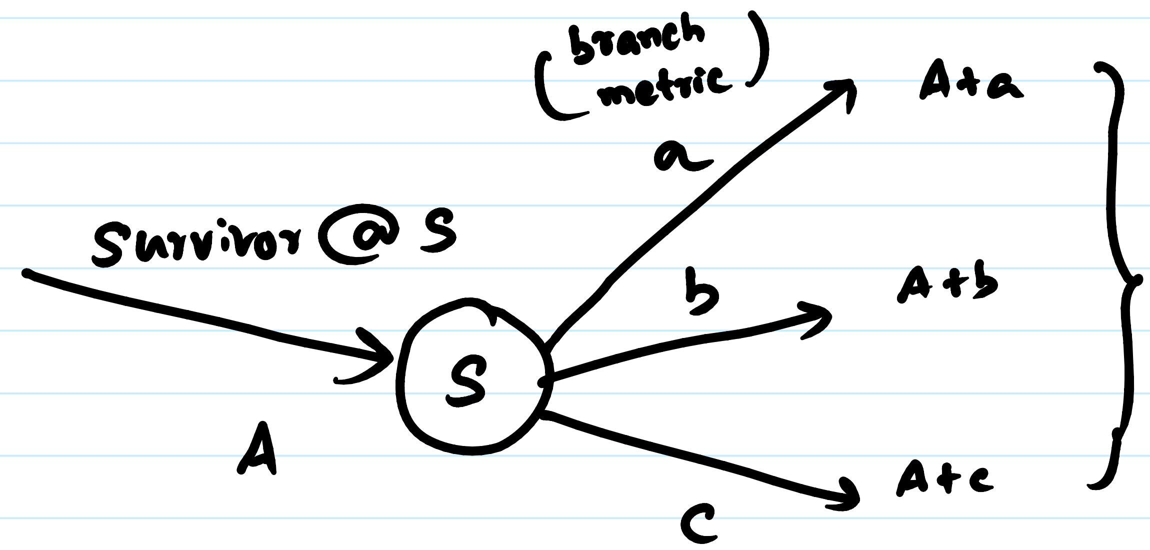
Usually $L << N$

Complexity is much less than brute force Computation (M^N)

Viterbi Algorithm (Two steps)

① ADD step

(at each time, each state)

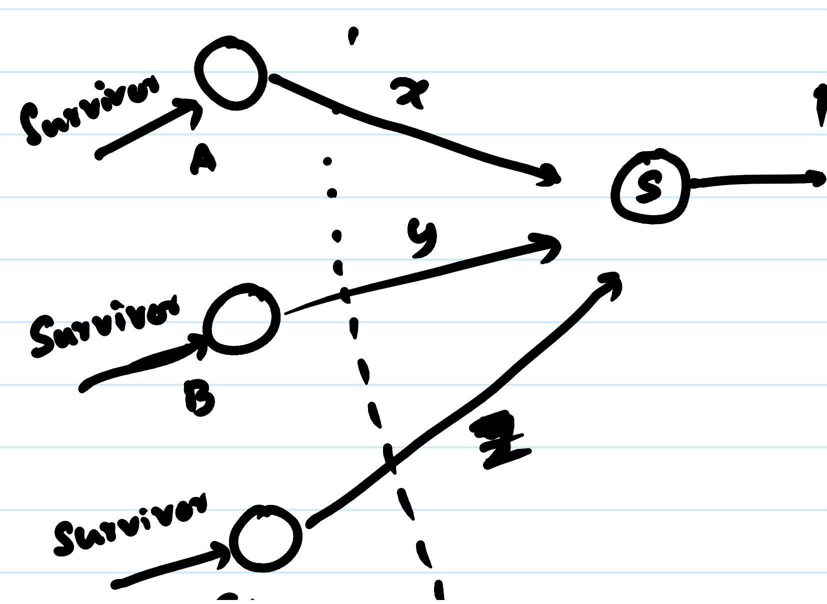


Accum. metrics going into next state from s

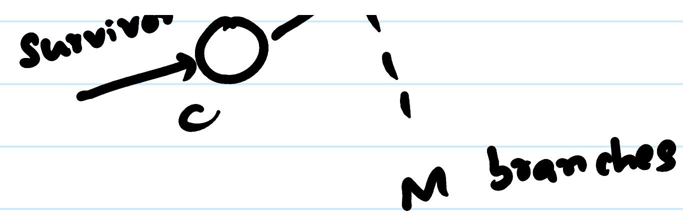
(M branches)

Accum. metric of survivor = A

② COMPARE step

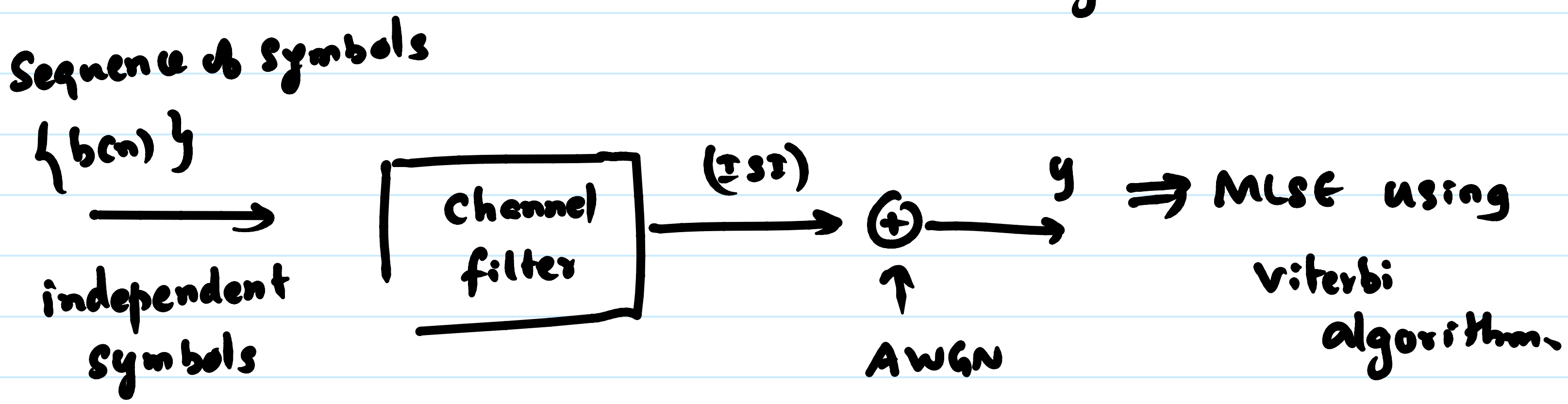


Pick largest of $A+x$, $B+y$, $C+z$ and obtain survivor at state S



Summary

- Sending a stream of symbols over ISI channels
- Symbols need to be <u>jointly</u> demodulated for optimal ML solution
- Obtained optimal MLSE solution using Trellis diagrams and Viterbi algorithm



Related problems where Viterbi algorithm can be used for ML solution

① Correlated symbols over AWGN

Sequence of symbols $\{b(n)\}$ $\rightarrow \oplus \rightarrow y \rightarrow$ Viterbi algo. for optimal ML solution

AWGN $\uparrow$

Correlated e.g. (convolutional code)

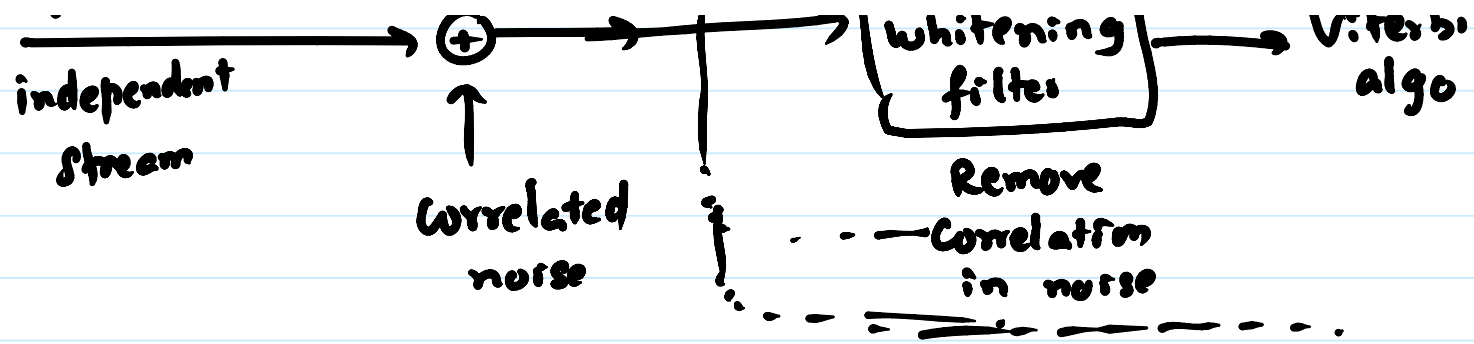
Current symbol depends on previous L symbols & current input

Memory is introduced by encoder

② Independent stream over colored Gaussian noise

$\{b(n)\}$ independent $\rightarrow \oplus \rightarrow y \rightarrow$ [whitening filter] $\rightarrow$ Viterbi algo

Receiver Processing



③ Correlated symbols over ISI channels with colored noise

Low complexity (sub optimal) linear equalization

→ other popular alternative to MLSE

(Read books!)