ML Sequence Demodulation [Ch. 14]:

- Goal: Show that "orthogonal modulation with memory" enables MLWD with $\mathcal{O}(K_b)$ complexity.
- Key Idea: Leverage trellis structure induced by the FSM.
- This problem is solved by the Viterbi algorithm, an instance of *dynamic programming*.
- For simplicity, we restrict our discussion to rate-1 coding (i.e., $R = 1$).

Recall MLWD:

$$
\begin{aligned}
\hat{\underline{I}} &= \arg \max_{i \in \{0,\dots,2^{K_b}-1\}} T_i \\
&= \arg\max_i \sqrt{E_b} \sum_{k=1}^{N_f} \mathrm{Re}\big[\tilde{d}_i^{(k)*} Q^{(k)}\big] - \frac{E_b}{2} \sum_{k=1}^{N_f} |\tilde{d}_i^{(k)}|^2 \\
&= \arg\min_i \sum_{k=1}^{N_f} \Big| Q^{(k)} - \sqrt{E_b}\, \tilde{d}_i^{(k)} \Big|^2
\end{aligned}
$$

Observations:

- Brute force minimization requires $\mathcal{O}(2^{K_b})$ operations.
- The total number of trellis edges is only about $2N_sK_b$, suggesting that an $\mathcal{O}(K_b)$ scheme might exist.

Some definitions:

$$\Omega_{k_s}^{(l)} \triangleq \Big\{ i = \underline{I} \text{ s.t. } \sigma^{(l)} = k_s \Big\} \text{ for } k_s \in \{1, \ldots, N_s\}$$

(i.e., the set of sequences in state k_s at time l)

$$\Delta_E^{(l,k_s)} \triangleq \min_{i \in \Omega_{k_s}^{(l)}} \sum_{k=1}^{l-1} \Big| Q^{(k)} - \sqrt{E_b}\, \tilde{d}_i^{(k)} \Big|^2$$

(i.e., the lowest time-$(l-1)$ accumulated cost among sequences in state k_s at time l.)

$$\hat{\underline{I}}^{(l,k_s)} \triangleq \text{ partial bit sequence } [m_1, \ldots, m_{l-1}] \text{ minimizing } \Delta_E^{(l,k_s)}$$

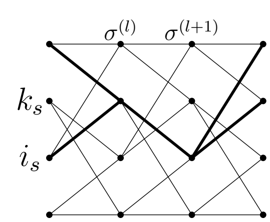
With a terminated trellis, the ML solution $\hat{\underline{I}}$ becomes

$$\hat{\underline{I}} = \hat{\underline{I}}^{(N_f+1,1)}.$$

Can partition the set $\Omega_{i_s}^{(l+1)}$ as follows:

$$\Omega_{i_s}^{(l+1)} = \Big\{ i = \underline{I} \text{ s.t. } \sigma^{(l+1)} = i_s \Big\}$$
$$= \bigcup_{k_s=1}^{N_s} \underbrace{\Big\{ i = \underline{I} \text{ s.t. } \sigma^{(l)} = k_s \,\&\, \sigma^{(l+1)} = i_s \Big\}}_{\triangleq\, \Omega_{k_s,i_s}^{(l,l+1)}}$$

Note that all sequences in $\Omega_{k_s,i_s}^{(l,l+1)}$ produce the same symbol at time l, i.e., $\forall i \in \Omega_{k_s,i_s}^{(l,l+1)},\ \tilde{d}_i^{(l)} = \tilde{d}_{k_s,i_s}$.



$$\tilde{d}_{k_s,i_s} \triangleq \begin{cases} a\big(g_2(k_s, m_s)\big) & \text{if } \exists\, m_s \text{ s.t. } i_s = g_1(k_s, m_s) \\ \infty & \text{else} \end{cases}$$

Main idea behind Viterbi algorithm:

$$
\begin{aligned}
\Delta_E^{(l+1,i_s)} &= \min_{i \in \Omega_{i_s}^{(l+1)}} \sum_{k=1}^{l} \left| Q^{(k)} - \sqrt{E_b}\tilde{d}_i^{(k)} \right|^2 \\
&= \min_{i \in \Omega_{i_s}^{(l+1)}} \left\{ \sum_{k=1}^{l-1} \left| Q^{(k)} - \sqrt{E_b}\tilde{d}_i^{(k)} \right|^2 + \left| Q^{(l)} - \sqrt{E_b}\tilde{d}_i^{(l)} \right|^2 \right\} \\
&= \min_{k_s} \min_{j \in \Omega_{k_s,i_s}^{(l,l+1)}} \left\{ \sum_{k=1}^{l-1} \left| Q^{(k)} - \sqrt{E_b}\tilde{d}_j^{(k)} \right|^2 + \left| Q^{(l)} - \sqrt{E_b}\tilde{d}_j^{(l)} \right|^2 \right\} \\
&= \min_{k_s \in \{1,\dots,N_s\}} \left\{ \Delta_E^{(l,k_s)} + \left| Q^{(l)} - \sqrt{E_b}\tilde{d}_{k_s,i_s} \right|^2 \right\}
\end{aligned}
$$

$$
\underline{\hat{I}}^{(l+1,i_s)} = \left[\underline{\hat{I}}^{(l,k_s^*)}, m_s^* \right] \text{ where } \begin{cases} k_s^* \triangleq \text{ minimizing } k_s. \\ m_s^* \triangleq \text{ bit value taking } k_s^* \text{ to } i_s. \end{cases}
$$

Viterbi Algorithm Summary:

$\Delta_E^{(1,1)} = 0, \quad \Delta_E^{(1,k_s)}\big|_{k_s>1} = \infty, \quad \underline{\hat{I}}^{(1,\cdot)} = [\,]$

for $l = 1, \dots, N_f$,

  for $i_s = 1, \dots, N_s$,

$$
\Delta_E^{(l+1,i_s)} = \min_{k_s \in \{1,\dots,N_s\}} \left\{ \Delta_E^{(l,k_s)} + \left| Q^{(l)} - \sqrt{E_b}\tilde{d}_{k_s,i_s} \right|^2 \right\}
$$

$$
\underline{\hat{I}}^{(l+1,i_s)} = \left[\underline{\hat{I}}^{(l,k_s^*)}, m_s^* \right] \text{ for } \begin{cases} k_s^* \triangleq \text{ minimizing } k_s. \\ m_s^* \triangleq \text{ bit taking } k_s^* \text{ to } i_s. \end{cases}
$$

  end

end