Partial Response and Viterbi Detection

Prof. David Johns
University of Toronto

(johns@eecg.toronto.edu)
(www.eecg.toronto.edu/~johns)

Partial-Response Motivation

Disadvantage — Feed-Forward Equalizer

- An FFE boosts the noise in areas where received signal power is low
- Example:

\[ A_k = \pm 1 \]

\[ H_{tc}(z) \rightarrow FFE \rightarrow H_1(z) \rightarrow \text{output data} \]

\[ H_{tc} \times H_1 \]

\[ f \]

• Noise is boosted at high frequencies.
**Partial-Response Motivation**

**Disadvantage — Decision Feedback Equalizer**

- A DFE does not make use of all the impulse response.

\[ H_{fe}(z) = 1 + 0.9z^{-1} + 0.1z^{-2} \]

- Since \( \hat{A}_k = A_k \), impulse response is \( \delta(k) \)
- If single input, better to look for \[ 1, 0.9, 0.1, 0, 0 \] than \[ 1, 0, 0, 0, 0 \]
- Postcursor ISI may have significant signal power.

**Partial-Response Motivation**

- Rather than equalizing to a Nyquist pulse, equalize to a partial-response signal
- Equalize to \( 1 + z^{-1} \) in DFE example
- Less noise boost
- More of the impulse response used to determine transmitted signal
- Need to look at a string of received symbols rather than symbol-by-symbol detection — MLSD
- Disadvantage — extra complexity and may not recover full dynamic range loss
Nyquist Criterion for Zero ISI

- **Nyquist’s First Criterion** for zero ISI

\[
\begin{align*}
    h(kT) &= \begin{cases} 
        1 & k = 0 \\ 
        0 & k \neq 0 
    \end{cases} \\
    \sum H\left(\omega - \frac{2\pi n}{T}\right) &= T
\end{align*}
\]

Minimum Bandwidth System with Zero ISI

- **A brickwall low-pass spectrum with a cutoff frequency of** \(1/(2T)\) (**sinc** impulse response)

- However — impulse response decays at a rate of \(1/t\) due to the frequency discontinuity in \(H(f)\).

- Excessive ISI if **any** timing perturbation occurs
Non-Minimum Bandwidth System

• One way to overcome jitter problem is to use more than the minimum bandwidth.

• A popular class of non-minimum bandwidth solutions are — Cosine Roll-Off Filters

• Can still transmit and receive only one of two symbols.

• But are minimum bandwidth systems practical? Yes. — use partial-response signaling.

Partial-Response Signaling

• By relaxing the zero-ISI criterion of Nyquist, the maximum symbol rate of 2 symbols/hertz can be achieved.

• Allow a controlled amount of ISI by digitally FIR filtering the data — results in more signal levels.

• Three popular FIR filters:

  1 + z\(^{-1}\)  duobinary - class 1  zero at \(f_s/2\)
  1 − z\(^{-1}\)  dicode  zero at dc
  1 − z\(^{-2}\)  modified duobinary - class 4  zeros at dc, \(f_s/2\)

  (also called PR4 or PRIV)
Duobinary \((1+z^{-1})\)

- Impulse response decays at a rate of \(1/t^2\) since \(H(f)\) is continuous but its first derivative is not.
- However, it transmits signal power at dc.

Dicode \((1-z^{-1})\)

- Impulse response decays at a rate of \(1/t\) due to the frequency discontinuity in \(H(f)\).
- However, it does not transmit any signal power at dc.
Modified Duobinary \((1-z^{-2})\) — class 4

- decays at a rate of \(1/t^2\) since \(H(f)\) is continuous.
- It does not transmit signal power at either dc or \(f_s/2\).

Class-4 Partial Response Signaling Scheme

- Spectral nulls at DC and \(f_s/2\)
- Can be encoded/decoded by two interleaved dicode encoder/decoder each operating at half the rate.

- Thus, we need only decode a dicode and use two interleaved identical blocks to decode PRIV.
- If binary inputs, 3 level output — BPR4 or BPRIV
- (If 4 level inputs, 9 level output — QPR4 or QPRIV)
Magnetic Recording Similarities

• At low densities, a magnetic read signal is inherently 1-D encoded (i.e. a dicode).

• At higher densities, high-frequency roll-off important (modelled as a Lorentzian pulse).

• If equalized to a 1-D channel, high-frequency noise is amplified.

• Find a good approximation to channel so that the boost required by equalizer is kept small.

Magnetic Recording Similarities

• Magnetic recording channel often modelled as Lorentzian pulse

![Frequency Response of a Lorentzian Magnetic Read Channel](image)
Magnetic Recording Similarities

- Similar to \((1 - z^{-1})(1 + z^{-1})^n\) partial-response channel.

![Frequency Response of \((1-D)(1+D)^n\) PRS Scheme](image)

SNR Degradation for Dicode

- Now 3 levels being sent rather than just two.

<table>
<thead>
<tr>
<th>Uncoded binary signal</th>
<th>“1-D” encoded ternary signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Δ</td>
</tr>
<tr>
<td>-1</td>
<td>ΔΔ</td>
</tr>
</tbody>
</table>

- Thus, a bit-by-bit detection results in SNR performance degradation (about 2-3 dB loss).
- However, the 3 levels have some redundancy included.
- SNR performance can be recovered in detection by employing Maximum-Likelihood Sequence Estimation (MLSE) detection schemes
- The **Viterbi Algorithm** is an efficient way of realizing MLSE detection
**Trellis Introduction**

- **State Diagram**
  - Input: 0, 1
  - Output: ±1, 0
  - States: 0/0, 1/0, 1/+1, 0/-1

- **Trellis Diagram**
  - States: k-1, k, k+1
  - Transitions: 0/0 → 1/0 → 1/+1 → 0/-1 → 0/0

---

**Trellis Representation of Dicode (1-z⁻¹)**

- A trellis can be used to describe an encoder.
- Example:

  - **Trellis**
    - States: k-1, k, k+1
    - Transitions: 0/0 → 1/0 → 1/+1 → 0/-1 → 0/0

---

**Encoder**

- **Data**: 0, 1
- **Encoded Data**: ±1, 0
- **Output**: ±1, 0

---

University of Toronto
Transmit Trellis for Dicode \((1-z^{-1})\)

- Note that following a ‘+1’ output, there can be an arbitrary number of zeros followed by a ‘-1’.
- In other words, if two ‘+1’ symbols are detected with no ‘-1’ between them, an error occurred in transmission.
- Similar for a ‘-1’ output.

Conventional Bit-by-Bit Detection

- Note the error in bit 3 received.
- Error can be detected since a “-1” must be next non-zero symbol after a “+1”.
- Did the error most likely occur in symbol 2, 3 or 4?
Viterbi Algorithm (VA)

- VA is an iterative method for determining the most likely sequence sent — maximum likelihood detector.

- Accomplished by creating a **receive trellis** having **branch metrics** proportional to the difference squared between received signal and each ideal symbol value.

- The most likely sequence is the shortest path through the receive trellis.

- **State metrics** and **path memory** also stored to reduce search time through trellis — they are the length of shortest path and path taken at each node.

Viterbi Algorithm Example

- Data: 1 1 0 1 0 0
- Received Data: +1 0 -1 +1 -1 0
- Encoded Data: 0/0, 1/0, 1/+1, 0/-1
- Channel: 0.00 0.64 0.04 0.73 0.64 0.04 0.73 0.64
- Noise: +0.80 -0.30 -0.40 -0.90 -0.80 -0.10
- Large noise effect on channel.

Data Received: 1 1 0 1 0 0
Encoded Data Received: 0 +1 0 -1 +1 -1 0

University of Toronto
Detailed Received Trellis Description (BPR4)

- Transmitted signal — one of three values, $\pm a, 0$
- Received signal — $y_k$.

state metrics $\rightarrow m_{0_{k-1}} \quad m_{0_k}$

\[ (y_k - 0)^2 \]

\[ (y_k + a)^2 \quad (y_k - a)^2 \quad \text{branch metrics} \]

state metrics $\rightarrow m_{1_{k-1}} \quad m_{1_k}$

\[ (y_k - 0)^2 \]

\[ (m_{0_{k-1}} + y_k)^2 \quad (m_{1_{k-1}} + (y_k + a)^2) \]

\[ m_{0_k} = \min \{ (m_{0_{k-1}} + y_k)^2, (m_{1_{k-1}} + (y_k + a)^2) \} \]

\[ m_{1_k} = \min \{ (m_{1_{k-1}} + y_k^2), (m_{0_{k-1}} + (y_k - a)^2) \} \]
Simplifications to Remove Multiplications

• Remove \( y_k^2 \) terms since it occurs in both terms and we are only interested in finding the minimum path (don’t need the absolute length of the path).

• State-metrics can now be either positive or negative.

\[
\begin{align*}
m_{0k-1} & \quad m_{0k} \\
(y_k - 0)^2 & \quad (y_k - a)^2 \\
(y_k + a)^2 & \quad (y_k - a)^2 \\
m_{1k-1} & \quad m_{1k} \\
\end{align*}
\]

\[
\begin{align*}
m_{0k-1} & \quad m_{0k} \\
0 & \quad 0 \\
(2ay_k + a^2) & \quad (-2ay_k + a^2) \\
m_{1k-1} & \quad m_{1k} \\
\end{align*}
\]

Simplifications to Remove Multiplications

• Divide all branches by \( 2a \) (assume \( a > 0 \))

• Simply scales state metrics.

\[
\begin{align*}
m_{0k-1} & \quad m_{0k} \\
0 & \quad 0 \\
(2ay_k + a^2) & \quad (-2ay_k + a^2) \\
m_{1k-1} & \quad m_{1k} \\
\end{align*}
\]

\[
\begin{align*}
m_{0k-1} & \quad m_{0k} \\
0 & \quad 0 \\
\left(y_k + \frac{a}{2}\right) & \quad \left(-y_k + \frac{a}{2}\right) \\
m_{1k-1} & \quad m_{1k} \\
\end{align*}
\]

• Equations:

\[
\begin{align*}
m_0 &= \min\{m_{0k-1}, (m_{1k-1} + y_k + a)\} \\
m_1 &= \min\{m_{1k-1}, (m_{0k-1} - y_k + a)\}
\end{align*}
\]
Difference Metric Algorithm

- [Wood and Peterson, Trans. on Comm., May 1986]

- We are not interested in absolute state-metric values — only which state-metric is smaller.

- Store only the difference in the state-metrics, $\Delta m_k$

  \[ \Delta m_k = m_{0k} - m_{1k} \]

- We shall see that while absolute state-metric values increase in time, their difference does not.

- This “difference metric algorithm” results in less complex realizations for both digital and analog realizations in cases where there are only two state-metrics.

Difference Metric Algorithm

- Subtract off $m_{1k-1}$ from input state-metrics and add it into each branch metric instead.

- Now, $m_{1k-1}$ can be subtracted off branch metrics since it is the same in all branches.
Difference Metric Algorithm

• Different path choices.

\[ \Delta m_{k-1} \]

\[ \left( \begin{array}{c} y_k + \frac{a}{2} \\ -y_k + \frac{a}{2} \end{array} \right) \]

\[ \left( \begin{array}{c} 0 \\ 0 \end{array} \right) \]

\[ \left( \begin{array}{c} m_0 \\ m_1 \end{array} \right) \]

\[ \Delta m_{k-1} < y_k + \frac{a}{2} \quad \text{and} \quad \Delta m_{k-1} < y_k - \frac{a}{2} \]

\text{Result} \quad \Delta m_k = y_k - \frac{a}{2}

\[ \Delta m_{k-1} \]

\[ \left( \begin{array}{c} y_k + \frac{a}{2} \\ -y_k + \frac{a}{2} \end{array} \right) \]

\[ \left( \begin{array}{c} 0 \\ 0 \end{array} \right) \]

\[ \left( \begin{array}{c} m_0 \\ m_1 \end{array} \right) \]

\[ \Delta m_{k-1} > y_k + \frac{a}{2} \quad \text{and} \quad \Delta m_{k-1} > y_k - \frac{a}{2} \]

\text{Result} \quad \Delta m_k = y_k + \frac{a}{2}

Difference Metric Algorithm

• Different path choices.

\[ \Delta m_{k-1} \]

\[ \left( \begin{array}{c} y_k + \frac{a}{2} \\ -y_k + \frac{a}{2} \end{array} \right) \]

\[ \left( \begin{array}{c} 0 \\ 0 \end{array} \right) \]

\[ \left( \begin{array}{c} m_0 \\ m_1 \end{array} \right) \]

\[ \Delta m_{k-1} < y_k + \frac{a}{2} \quad \text{and} \quad \Delta m_{k-1} > y_k - \frac{a}{2} \]

\text{Result} \quad \Delta m_k = \Delta m_{k-1}

\[ \Delta m_{k-1} \]

\[ \left( \begin{array}{c} y_k + \frac{a}{2} \\ -y_k + \frac{a}{2} \end{array} \right) \]

\[ \left( \begin{array}{c} 0 \\ 0 \end{array} \right) \]

\[ \left( \begin{array}{c} m_0 \\ m_1 \end{array} \right) \]

\[ \Delta m_{k-1} > y_k + \frac{a}{2} \quad \text{and} \quad \Delta m_{k-1} < y_k - \frac{a}{2} \]

\text{IMPOSSIBLE}

\text{Result} \quad \text{do nothing}
Difference Metric Algorithm

- Equations:

\[
\begin{align*}
\Delta m_k &= \begin{cases} 
  y_k + \frac{a}{2} & \Delta m_{k-1} > y_k + \frac{a}{2} \\
  \Delta m_{k-1} & y_k - \frac{a}{2} < \Delta m_{k-1} < y_k + \frac{a}{2} \\
  y_k - \frac{a}{2} & \Delta m_{k-1} < y_k - \frac{a}{2}
\end{cases}
\]

- These equations describe an adjustable threshold device.
- Used in digital PR4 implementations.
- They are also simple to implement in analog.

Typical Digital Implementation

- 6-bit flash A/D requires 63 comparators + decoding logic.
- A/D converter might consume around 300mW (or more)
- FIR equalizer might be 1mW/MHz/tap — 8-tap at 100MHz = 800mW
**Typical Digital Implementation**

- Digital equalizer requires multi-bit multiplies in feed-forward equalizer (power hungry).
- If decision feedback is used, it will need to use early estimates of output (cannot wait for MLSE to finish).
- Digital difference algorithm requires some minor adders and digital comparators.
- Digital path memory logic consists of about 16 serial/parallel shift registers.

**Typical Analog Implementation**

- Analog equalizer needed (less power than digital but more challenging).
- Digital path memory is the same as in fully digital realization.

---

![Diagram](image-url)
**Typical Analog Implementation**

- Analog difference algorithm is very small (about the size of 2 comparators)

- Thus, power is saved and speed can be increased over 6-bit A/D converter.

- Note that dynamic range in analog parts need only be around 6 bits (i.e. 40dB)

**Analog Implementation**

- Path memory consists of two serial/parallel in/out shift registers.

- Threshold Device

<table>
<thead>
<tr>
<th>$v_a/2$</th>
<th>$v_a/2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\Delta v_a/2$</td>
</tr>
</tbody>
</table>

- Viterbi Detector

<table>
<thead>
<tr>
<th>$v_a/2$</th>
<th>$v_a/2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\Delta v_a/2$</td>
</tr>
</tbody>
</table>

- Mux

- DC level shifter

- State “0”

- State “1”

- to path memory

- CLK

- S/H
**Some Practical Limitations**

- In a digital implementation, performance is degraded by limiting the number of bits used in A/D conversion.
- Typically use about 6-bit A/D converters (easily achievable in an all analog implementation).
- Truncating the trace-back length (path memory) also degrades the performance.
- Typically use length of 16 for little loss in performance.

**Why an Analog Implementation?**

- Avoids using a pre-stage A/D converter.
- Combines the A/D and VA into one stage with a complexity near to a 2-bit A/D (*Special-purpose A/D converter*).
- Consumes less power.
- Operates faster.
- 6-bit accuracy is enough (Moderate Precision Circuitry).
- Low-dynamic range requirement (Low-Voltage Operation).
- The difference algorithm updates only one sampled data without using previous samples (no *accumulative* analog errors)
Simulation and Experimental Results

- Simulation and experimental (discrete prototype) results confirm validity of the analog approach and its robustness against imperfections.

![Graphs showing Signal to Noise Ratio (SNR) and Bit Error Rate (BER)]

Offsets and mismatches are described in percentage of $a$.

Measured BER Performance of the Detector (Path Memory = 17)

Input-Interleaved Algorithm

- The implementation above updates $\Delta m_k$ once the comparator outputs are known.

- Thus, **critical speed path is 2 sample-and-holds.** (Sample input and compare, then, perhaps, update $\Delta m_k$ with another sample-and-hold).

- The input-interleaved algorithm reduces the critical speed path to a single sample-and-hold (i.e. can operate at twice the speed).

- It uses two sample-and-holds at the input and switches which one the input goes to if $\Delta m_k$ needs to be updated.
**Input Interleaved Algorithm**

- According to the update mechanism, $\Delta m_{k-1}$ is a DC-shifted version of a previously-sampled input signal

$$\Delta m_{k-1} = y_{k-j} \pm 0.5$$

- We can use the previously held input signal plus appropriate sign of DC shift for $\Delta m_k$.

- When $\Delta m_k$ needs to be updated, switch input on to other sample-and-hold capacitor and use the just sampled input and a sign-bit for new $\Delta m_k$.

**Input-Interleaved Algorithm**

- Toggle between two S/Hs which store $y_k$ and $y_{k-j}$

- Use a flip-flop to properly switch the DC signal

- Speed improvement, as no additional S/H is required ($y_{k-j}$ has already been stored)
BiCMOS Integrated-Circuit Implementation

BiCMOS Integrated-Circuit Implementation

University of Toronto
**BiCMOS Integrated-Circuit Implementation**

- Path memory consists of 2x12 multiplexed-input D flip-flops

  ![Path Memory Diagram]

- Clock phases

  ![Clock Phases Diagram]

- Compared to other analog implementations
  
  [Matthews and Spencer, JSSC, Dec. 93]

  - Less complex (Individual state metrics are not calculated)
  - Less prone to imperfections (Feedback signals are only digital)
  - Fully differential
  - Faster (Master-slave S/Hs are not used)

  [Yamasaki, ISSCC, 1994]

  - No details given
Experimental Results

- Process: 0.8 µm BiCMOS
- Area (dicode): ~0.25 mm²
  - Analog: ~0.06 mm²
  - Digital: ~0.1 mm²
  - Bypass capacitors, ...
- Power consumption (dicode):
  - 3.3V power supply
  - ~12mW at 50MHz
  - ~15mW at 100MHz
**Experimental Results**

- Setup

![Diagram](slide)

- Measured Bit-Error-Rate (BER) performance

![Graph](slide)

**A General Implementation Approach**

- Analog implementations are useful if the preceding signal processing is simple
  - Magnetic recording
  - Data transmission over unshielded cables
- Simplifications are only possible in some special cases (i.e. PR4)
- This general approach can be used in
  - More general PRS schemes (i.e. EPR4, EEPR4)
  - Convolutional codes
  - Multi-level digital communication
  - Irregular trellises
**A General Implementation Approach**

- This approach takes full advantage of the ability of simple analog circuits in realizing the ACS function

\[
m_i(k) = \text{Max}_j \left\{ m_j(k-1) - e_{ji}(k) \right\} \quad i = 1, 2, \ldots, N
\]

\[
j = 1, 2, \ldots, M
\]

- Branch metrics, \(e_{ij}\), are usually expressed in terms of linear combinations of the received samples and DC signals

**Circuit Realization**

- A generalized differential cell is employed to realize the ACS function

- Using degenerated differential pairs in V/I conversions makes the linear combinations simple to realize
• Optional DC currents added to the error signals reduce the unnecessary DC voltage drops across the resistors

---

**Circuit Realization**

• Branch currents in the differential cells are comparison results

• To achieve high speeds, ping-pong S/Hs are preferred to master-slave S/Hs in feeding the state metrics back

• Algorithmic growth of state metrics is overcome by a fast Common-Mode Feedback (CMFB) circuit

• Fast CMFB minimizes the signal swings of the state metrics — this approach is usually not practical in digital realizations
Design Example: Binary Dicode

Integrated Circuit Implementation

- A chip containing Viterbi decoders for a binary dicode and an Extended PR4 (EPR4, (1-D)(1+D)^2) has been fabricated in a 0.8\,\mu m BiCMOS process
- Based on simulations, fast speed (>100 MHz) can be achieved with \sim 15mW/state (Excluding path memory)
- The area is \sim 0.03mm^2/state (Excluding path memory)
- In a CMOS implementation, lower g_m causes some degradation in:
  - Obtaining simple low-impedance nodes
  - ACS performance due to the high dependency of v_{GS} to the drain current
Integrated Circuit Implementation

Preliminary Experimental Results

- Results on dicode

- High-frequency tests have been conducted up to 80 MHz (Off-chip path memory)
Summary

• The use of partial-response signals allows one to send closer to the maximum rate of 2 symbols/hertz.

• Making use of partial-response signalling reduces the need for large equalization boost.

• The difference algorithm is efficient for PR4 signals.

• Analog realizations of the difference algorithm save silicon and power over digital realizations (however, an analog equalizer is needed).

• Input-interleaved algorithm increases the speed for an analog implementation.

• A general analog Viterbi approach was discussed. (However, it makes use of bipolar transistors).