HYBRID CONCATENATED CONVOLUTIONAL CODES FOR DEEP SPACE MISSION

Presented by Dr. DEEPAK MISHRA OSPD/ODCG/SNPA

MATLAB EXPO 2015
Objective: To find out suitable channel codec for future deep space mission.

Outline:

- Interleaver Design.
- Puncturing.
- Decoding Algorithms.
- Various Turbo code like structure like Hybrid Concatenated codes.
- Validation and verification using MATLAB
  - HDL Co Simulation
  - Hardware in loop Simulation
Digital Communication System

- Information Source & Input Transducer
- Source Encoder
- Channel Encoder
- Digital Modulator
- Channel
- Digital Demodulator
- Synchronization
- Output Transducer
- Source Decoder
- Channel Decoder
- Transmitted Signal
- Received Signal

MATLAB EXPO 2015
Shannon theorem

Channel Coding theorem

\[ \frac{H(z)}{TS} \leq \frac{C}{TC} \]

Shannon Source Coding theorem

\[ L \geq \mathcal{H}(s) \]

Where

\[ L = \sum_{k=0}^{K-1} p_k l_k \]

\[ H(S) = - \sum p_i \log(p_i) \]

Information Capacity Theorem

\[ C = B \log_2 \left( 1 + \frac{P}{N_0 B} \right) \text{ bits/sec} \]
Interleaver Design
Classical Block diagram
Interleaver Design

Earlier Concepts

Design

Earlier Interleaver design is separate design and there is no relation between interleaver design and Channel coder design.

Purpose

To distribute the error through out the frame. It has the job of spreading out long bursts of errors.
Interleaver Design

Purpose
To provide Interleaver gain (decorrelation gain) to decoder.

Different properties:-
1) S distance properties
2) mod-k properties
3) symmetric properties
S distance Property

An interleaver with the spread or S distance property will, after interleaving, separate all neighboring elements at least S interleaver index distance a part, i.e., 

\[ S \min(|\pi(i) - \pi(j)|, (|\pi^{-1}(i) - \pi^{-1}(j)|)), \text{ for all } i, j \in I, |i-j|=1. \]

The performance of Channel codes improves as the S distance increases.

\[ S < \lceil \sqrt{N/2} \rceil, \text{ where } N \text{ is the size of the interleaver} \]
Input /output Position Plot of a 192 bit poor S distance random Interleaver

Input /output Position Plot of a 192 bit good s distance random Interleaver
Mod -K Property used in applications where k-1 parity bits are punctured from each constituent code. We call this a pure mod-k interleaver where the modulus rule applies to all elements of the interleaver.

Therefore a pure mod-k interleaver has \( i \mod k = j \mod k \), where the interleaver maps \( i \rightarrow j \).
Symmetric Property

A disadvantage of most of the interleavers types mentioned previously is that they require both an interleave and a deinterleave sequence. Since an interleave and a deinterleave sequence are normally different, separate hardware or look up tables are usually required for each sequence. We can solve this problem by using a symmetric interleaver, where the interleaver and deinterleaver sequences are identical.
Standardization of turbo codes by the Consulting Committee for Space Data System (CCSDS) organization was remarkable efficient process, because there are relatively few parameters must be determined to define a turbo code.

### CCSDS Turbo Codes Standard

<table>
<thead>
<tr>
<th>Code type</th>
<th>Systematic parallel concatenation turbo code</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Number of components codes</strong></td>
<td>2 (plus an uncoded component to make the code systematic).</td>
</tr>
<tr>
<td><strong>Type of component codes</strong></td>
<td>Recursive convolutional codes.</td>
</tr>
<tr>
<td><strong>Number of states of each convolutional component code</strong></td>
<td>16</td>
</tr>
<tr>
<td><strong>Nominal Code Rates</strong></td>
<td>$r = 1/2, 1/3, 1/4, \text{ or } 1/6$ (selectable).</td>
</tr>
<tr>
<td><strong>Interleaver length $k$</strong></td>
<td>1784, 3568, 7136, or 8920</td>
</tr>
<tr>
<td><strong>Interleaver type</strong></td>
<td>Algorithmic</td>
</tr>
</tbody>
</table>
CCSDS 101.0-B-6 Standard Turbo Encoder
CCSDS Compline Turbo Encoder and Decoder

encoded
Signal From Workspace1

Tx
Error Rate Calculation
Rx

Model Parameters
Double-click to set model parameters

encoded

CCSDS compline Turbo Encoder

Int
Out1

AWGN

Int
Out1
CCSDS compline Turbo Decoder

In1
Out1

In1
Out1

encoded

finaldatad
To Workspace1

pcccBER

Error Rate Calculation

CCSDS Compline Turbo codes
The interleaver permutation laws $\pi : Z_k \rightarrow Z_k$ were proposed by Berrou et al [Berrou et al (1993)], and are generated by the following algorithm [1]:

- First express $k$ as $k = k_1 k_2$ where $k_1 = 8$ and $k_2 = k/8$.
- Then, for each $s$ from 0 to $(k - 1)$, compute $\pi(s)$ by using the following equations:

$$
\begin{align*}
    m &= s \mod 2 \\
    i &= \text{floor}\left(\frac{s}{2k_2}\right) \\
    j &= \text{floor}\left(\frac{s}{2}\right) - ik_2 \\
    t &= (19i + 1) \mod 4 \\
    q &= t \mod 8 + 1 \\
    c &= (P_q j + 21m) \mod k_2 \\
    \pi(s) &= 2t + 8c + 1 - m
\end{align*}
$$

Where $P_q$ denotes one of the following eight prime integers:

$p_1 = 31; p_2 = 37; p_3 = 43; p_4 = 47; p_5 = 53; p_6 = 59; p_7 = 61; p_8 = 67$. 

MATLAB EXPO 2015
Input / Output distribution of CCSDS Interleaver
The proposed interleaver is based on Gaussian distribution function. The selective criteria of these models are minimum distance and multiplicities for all suggested algorithmic interleavers and polynomials for turbo codes.

```matlab
m = mod((h-1),2);
i = floor((h-1)/446);
j = floor((h-1)/2) - (i*223);
t = mod((19*i+1),4);
q = mod(t,8)+1;
c = mod((9*j+113*m),223);
position1 = (2*(t+(c*4)+1))-m;
pot(h) = position1;
```
Input / Output Distribution of Proposed Interleaver
The advantages of proposed interleaver with respect to CCSDS interleaver are following.

- There is no need to store 8 prime integers value in hardware. Since it has only fixed value.
- The S distance property is better compare to CCSDS interleaver.
- The minimum hamming distance $d_{\text{min}}$ and its multiplicity values, $A_{\text{min}}$, is better compare to CCSDS standard.

<table>
<thead>
<tr>
<th>Frame length</th>
<th>Code rate</th>
<th>Feedback Polynomial</th>
<th>Feed forward Polynomial</th>
<th>Interleaver Model</th>
<th>$d_{\text{min}}$</th>
<th>$A_{\text{min}}$</th>
<th>$W_{\text{min}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1784</td>
<td>1/2</td>
<td>10011</td>
<td>11011</td>
<td>CCSDS</td>
<td>17</td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td>1784</td>
<td>1/2</td>
<td>10011</td>
<td>11011</td>
<td>PROPOSED</td>
<td>23</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Proposed Interleaver Advantages

The improvement factor in terms of error floor using the proposed interleaver compare to CCSDS interleaver in turbo code performance can be given as

\[ \Delta d_{\text{min}} = 10 \log \frac{d_{\text{min(proposed interleaver)}}}{d_{\text{min(CCSDS interleaver)}}} = 10 \log \frac{23}{17} = 1.31 \text{dB} \]

The BER performance of PSCCs can be approximated by

\[ BER \approx \frac{1}{2} \frac{w_{\text{min}}}{k} \text{erfc} \left( \sqrt{d_{\text{min}} \frac{k E_b}{n N_o}} \right) \]
SIMULATION APPROACH
function y = interleaver(u)
    s=zeros(1,1784); % Initialize output array
    y=zeros(1,1784); % Initialize output array
    p=zeros(1,8); % Initialize parameter array
    pot=zeros(1,1784); % Initialize temporary array
    p(1)= 31; p(2) = 37; p(3) = 43; p(4) = 47; % Set parameters
    r=length(u); % Length of input array
    for h=1:r;
        m=mod((h-1),2); % Calculate modulus
        i=floor((h-1)/446); % Calculate index
        j = floor((h-1)/2) - (i*223); % Calculate index
        t=mod((19*i+1),4); % Calculate modulus
        q = mod(t,8) +1; % Calculate modulus
        c=mod((p(q)*j+21*q),223); % Calculate temporary value
        position1=(2^t+(c^4)+1)-2; % Calculate index
        pot(h) =position1;
        y(h,1)=u(position1); % Assign value
    end;
Proposed Interleaver Advantages
Puncturing
MATLAB EXPO 2015

Turbo Encoder
CCSDS Complline Turbo codes

Turbo Decoder

For Iterator

Select Rows

Insert Zero

Multiport Selector

Internal Deinterleaver

Hard Decision

Internal Interleaver

APP Decoder

Decoder1

Decoder2

NotUsed

L(u)
L(c)
L(u)
L(c)

L(u)
L(c)
L(u)
L(c)

L(u)
L(c)
L(u)
L(c)

Lc1
Lc2
Lc3

Out

In1

MATLAB EXPO 2015
Turbo Decoder without Transmission of Systematic Bits
Result shows that deletion of parity bit will be preferred over systematic bits.
Decoder in case of Turbo code
1. Decoding algorithm

- MAP Algorithm (max of posteriori probability).
- Log-MAP Algorithm
- Near Log-MAP Algorithm
- SOVA Algorithm (Soft output Viterbi algorithm)

Performance of MAP Algorithm is better compare to other algorithm, However Log-MAP and SOVA algorithm is easier to implement in Hardware (i.e. In log domain multiplication become addition and division become subtraction.)

May 13, 2015
Decoding Algorithm for Hybrid Concatenated Codes

Decoding algorithm

- Log-MAP Algorithm (max of posteriori probability).
- SOVA Algorithm (Soft output Viterbi algorithm)

These two algorithm are practically use for implementation of Concatenated decoders. However decoding complexity of HCCC is still higher. Further modification on Log-MAP algorithm know as Linear Log-MAP Algorithm.
Decoding using Log-MAP Algorithm
Decoding using SOVA Algorithm

MATLAB EXPO 2015
Log-MAP vs SOVA

$G = [7, 5]$, Unpunctured(1/3), $Fram = 1024$, iter = 8

The plot shows the performance comparison between SOVA and Log-MAP for Turbo codes with a specific $G$ matrix and frame size.
**Linear Log-MAP Algorithm**

\[
\max^* (\lambda_i) \triangleq \ln\left(\sum_i e^{\lambda_i}\right)
\]

\(\lambda_i\) is a real number. This operation with multiple arguments can be decomposed into a recursive form using a max* operator with only two arguments, such as

\[
\max^* (\lambda_1, \lambda_2, \lambda_3, \lambda_4) = \max^* \left( \max^* (\lambda_1, \lambda_2), \max^* (\lambda_3, \lambda_4) \right)
\]

Applying the Jacobian logarithm, a two-input max* operator can be expressed in the form

\[
\max^* (\lambda_1, \lambda_2) = \max (\lambda_1, \lambda_2) + \ln(1 + e^{-|\lambda_1 - \lambda_2|})
\]

\[
\triangleq \max (\lambda_1, \lambda_2) + f_c(|\lambda_1 - \lambda_2|)
\]
Linear Log-MAP Algorithm

A further enhancement is the more complex linear-log-MAP algorithm, which offers one of the best trade-offs in terms of complexity and performance among the different max* variants. It achieves an approximation very close to that of the log-MAP max* implementation by using a linear correction function.

\[
\text{max}^* (\lambda_1, \lambda_2) \approx \text{max} (\lambda_1, \lambda_2) + \begin{cases} 
0, & \text{if } |\lambda_2 - \lambda_1| > T \\
\alpha (|\lambda_2 - \lambda_1| - T), & \text{if } |\lambda_2 - \lambda_1| \leq T
\end{cases}
\]

We found that, parameters \(a = -0.24904\) and \(T = 2.5068\) minimize the total squared error between the exact correction function and its linear approximation, when using floating-point operations to implement the decoder.
## Simulation Parameter

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eb/No</td>
<td>0-7 dB</td>
</tr>
<tr>
<td>Block Length</td>
<td>1784</td>
</tr>
<tr>
<td>Interleaver</td>
<td>Random Interleaver</td>
</tr>
<tr>
<td>Iteration</td>
<td>6</td>
</tr>
</tbody>
</table>
Simulation of turbo decoder using Linear Log MAP using $a = -0.24904$ and $T = 2.5068$
PERFORMANCE COMPARISON BETWEEN LOG MAP AND LINEAR LOG MAP

Eb/No (dB) vs. BER

- **Log-MAP**
- **Linear Log-MAP**
Serial concatenated Encoder and Decoder

Convolutional Encoder
Random Interleaver
Convolutional Encoder
Unipolar to Bipolar Converter

Random Interleaver
Random Deinterleaver
APP Decoder
L(u)
L(c)
Outer Decoder
APP Decoder
L(u)
L(c)
Inner Decoder
Lin0/1
-K-
-K-
Deinterlacer
Deinterlacer
Add

MATLAB EXPO 2015
Hybrid concatenated Encoder and Decoder

Hybrid concatenated codes

MATLAB EXPO 2015
COMPARITIVE PERFORMANCE OF SCCC, PCCC AND HCCC

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>SCCC</th>
<th>PCCC</th>
<th>HCCC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CROSS OVER POINT

Turbo codes
Hybrid Concatenated codes with respect to different interleaver
# RESULTS

Packet size: 1784 Bits, BER=10^{-6}

<table>
<thead>
<tr>
<th>S.No</th>
<th>Interleaver Type</th>
<th>1&lt;sup&gt;st&lt;/sup&gt; iteration (Eb/No)</th>
<th>2&lt;sup&gt;nd&lt;/sup&gt; iteration (Eb/No)</th>
<th>3&lt;sup&gt;rd&lt;/sup&gt; iteration (Eb/No)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Pseudo random</td>
<td>4.4</td>
<td>3.3</td>
<td>2.1</td>
</tr>
<tr>
<td>2</td>
<td>Matrix</td>
<td>5.5</td>
<td>4.8</td>
<td>3.9</td>
</tr>
<tr>
<td>3</td>
<td>Helical</td>
<td>5.2</td>
<td>4.7</td>
<td>3.8</td>
</tr>
<tr>
<td>4</td>
<td>CircularA</td>
<td>5.6</td>
<td>5.0</td>
<td>4.3</td>
</tr>
<tr>
<td>5</td>
<td>Algebraic</td>
<td>5.1</td>
<td>4.2</td>
<td>3.5</td>
</tr>
</tbody>
</table>
# Turbo Coded Digital QPSK Modulator for Human Space Program

<table>
<thead>
<tr>
<th>Specifications</th>
<th>70.000 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carrier Frequency</td>
<td></td>
</tr>
<tr>
<td>Modulation type</td>
<td>QPSK</td>
</tr>
<tr>
<td>Information rate</td>
<td>1.024 Mbps</td>
</tr>
<tr>
<td>Input Signal interface</td>
<td>LVDS</td>
</tr>
<tr>
<td>Forward Error Correction</td>
<td>CCSDS ½ rate Turbo Encoder</td>
</tr>
<tr>
<td>Scrambler</td>
<td>CCSDS compatible Scrambler</td>
</tr>
<tr>
<td>Input Power from TCXO</td>
<td>0 dBm±0.5 dB</td>
</tr>
<tr>
<td>Output Power (Nominal)</td>
<td>-12 ± 1 dBm</td>
</tr>
<tr>
<td>Pulse Shaping Characteristic</td>
<td>35% Root Raised Cosine</td>
</tr>
<tr>
<td>Out of band attenuation</td>
<td>30 dB</td>
</tr>
<tr>
<td>Impedance</td>
<td>50 Ω</td>
</tr>
<tr>
<td>Amplitude Unbalance</td>
<td>0.4 dB (max)</td>
</tr>
<tr>
<td>Phase Unbalance</td>
<td>≤ 30</td>
</tr>
<tr>
<td>Input Interface</td>
<td>9 pin D-type connector</td>
</tr>
<tr>
<td>Output Interface (RF)</td>
<td>SMA</td>
</tr>
<tr>
<td>DC Current at 5 V ± 5%</td>
<td>160 mA</td>
</tr>
<tr>
<td>DC Current at -5 V ± 5%</td>
<td>150 mA</td>
</tr>
</tbody>
</table>
Modulator Block Diagram
Validation and Verification

- Raw data
- Encoded Data
  - Turbo Encoded Data
  - AWGN
  - -K-
- CCSDS compliant Turbo Decoder
- BER
- Error Rate Calculation
  - Tx
  - Rx

Model Parameters
Double-click to set model parameters
Turbo Decoder
Simulation vs Hardware

Matlab Simulation

Eye pattern of the modulated data

Eye opening = 57.5%

Hardware
Simulation vs Hardware

Matlab Simulation

Hardware
DVM Model
Hardware Setup
Hardware Spectrum
Demodulation
Verification of coding gain
APSK Modulator Block Diagram
## DESIGN PARAMETER

<table>
<thead>
<tr>
<th>S.No.</th>
<th>PARAMETER</th>
<th>UNITS</th>
<th>SPECIFICATIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Carrier Frequency</td>
<td>MHz</td>
<td>70.00</td>
</tr>
<tr>
<td>2</td>
<td>Modulation type</td>
<td>-</td>
<td>(4+12) APSK</td>
</tr>
<tr>
<td>3</td>
<td>Information rate</td>
<td>Mbps</td>
<td>1.024</td>
</tr>
<tr>
<td>4</td>
<td>Input Signal Interface</td>
<td>-</td>
<td>Differential</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>TURBO Convolutional code as per CCSDS STANDARD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Code rate= 1/3 &amp; 1/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Frame Length: 1784</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Backward vector G0=10011</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Forward vector G1=11011</td>
</tr>
<tr>
<td>5</td>
<td>Forward Error Correction</td>
<td>-</td>
<td>CCSDS COMPLINE scrambler</td>
</tr>
<tr>
<td>6</td>
<td>Scrambler</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Transmission Data Rate</td>
<td>Mbps</td>
<td>2.048</td>
</tr>
<tr>
<td>8</td>
<td>Output Power (Nominal)</td>
<td>dBm</td>
<td>-10 ± 1</td>
</tr>
</tbody>
</table>
CCSDS Standard APSK Modulator in Simulink
Noise Characterization at 13db Eb/No
Noise Characterization at 9 dB

Scatter Plot

In-phase Amplitude vs. Quadrature Amplitude
Simulation Result at 1784 bits frame length

Uncoded 16-APSK performance
Turbo coded 16-APSK performance

6.2 dB coding gain
Characterization of Turbo Encoder and Decoder
Hardware Constellation of 16 APSK
HDL Cosimulation
FPGA IN LOOP VERIFICATION

CCSDS Compliant Turbo Encoder and Decoder

- encoded
- Signal From Workspace1
- finaldata
- To Workspace2
- ENCODER_IN
- ENCODER_OUT
- FIL
- QPSK
- M-PSK Modulator Baseband
- AWGN
- M-PSK Demodulator Baseband

Model Parameters
- Double-click to set model parameters

Out1
- CCSDS Compliant Turbo Decoder

MATLAB EXPO 2015
Verification of Turbo decoder
Cosimulation and FPGA IN LOOP VERIFICATION
Result achieved

- **New interleaver design**: Performance is better than 1.31 dB compared to CCSDS compliant interleaver.

- **Puncturing**: Suitable puncturing scheme for Turbo-like codes.

- **Decoding scheme**: Optimize decoding scheme for hardware implementation.

- **Encoder and Decoder selection**: Suitable structure for future deep space mission.
Use of MATLAB to achieve the results

- Co simulation of proposed interleaver with CCSDS Interleaver
- Performance comparison of various puncturing channel codec.
- Comparison of various decoding algorithms.
- Simulation of various Turbo code like structure.
- Verification and Validation of codec without actual hardware.
Questions?
Dr. Deepak Mishra
Scientist/Engineer - SF
Onboard Signal Processing Division (OSPD)
Optical & Digital Communication Group (ODCG)
deepakmishra@sac.isro.gov.in

Space Applications Centre
Indian Space Research Organisation
Ahmedabad