column distance function \( (CDF) \) is defined as:

\[
\begin{align*}
\{0 \neq n : (\lambda)^m\} & \text{ with } \\
\{i \neq n : (i\lambda + i\lambda)^m\} & = \text{ the } p
\end{align*}
\]

This is the minimum distance between any two code words in the code. Since a convolutional code is linear, the code is linear.

\[
\text{minimum free distance}
\]

\[
1 - Y = \text{max}_m = m
\]

\[
\text{maximal memory order} = \text{length of the longest}
\]

\[
\sum_{i=1}^{m} i = m
\]

\[
\text{total memory} = \text{total number of memory elements}
\]

\[
\sum_{i=1}^{m} i = m
\]

Convolutional Codes Definitions

must be saved

In formula: number of information bits that

\[
(1 + m)u = y
\]

Single information bit

encoder outputs that can be affected by a

\[
\text{in time and cost} = m + 1 = y
\]

\[
\text{any input bit}
\]

a single output stream that can be affected by

\[
\text{in time and cost} = m + 1 = y
\]

\[
\text{any input bit}
\]
\[ f X (t) = \sum_{t=0}^{\infty} f_{X}(t) \]

**Weight Enumerator**

CDF of convolutional code with constraint length \( \lambda \) is the minimum distance \( d \) of a convolutional code.

The information block is nonzero.

So, \( d \) is the minimum-weight code word over

\[ \{0 \neq 0[n]: \exists [\lambda m] m \} \]
\[ \{0 \neq 0[n]: \exists [\lambda m] m \} \]
\[ \min \{0 \neq 0[n]: \exists [\lambda m] m \} \]
\[ \min \{0 \neq 0[n]: \exists [\lambda m] m \} \]
\[ \min \{0 \neq 0[n]: \exists [\lambda m] m \} \]
\[ \min \{0 \neq 0[n]: \exists [\lambda m] m \} \]

W denotes the weight of a Wiener filter.

Order \( i \), \( k \) is defined as

The convolution distance function of

denote the distance of the information sequence \( n \).

\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]

The convolution distance function of

denote the distance of the information sequence \( n \).

\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ (n) \]

\[ (n) \]
\[ (n) \]
\[ (n) \]
\[ \delta - \sum_{\mathcal{C}_m} \mathcal{C}_m \sum_{\mathcal{C}_l} \mathcal{C}_l - 1 = \nabla^{(Y)} \]

and where \( \nabla^{(Y)} \) is the cofactor of path \( Y \)

where \( (\mathcal{L}^{(Y)}) \) are surrounding loop pairs

\[ \delta + \sum_{\mathcal{C}_m} \mathcal{C}_m \sum_{\mathcal{C}_l} \mathcal{C}_l - \]

\[ \sum_{\mathcal{C}_m} \mathcal{C}_m \sum_{\mathcal{C}_l} \mathcal{C}_l - 1 = \nabla \]

where \( \nabla \) is the graph determinant

\[ \frac{\nabla}{\nabla^{(Y)}} = (X'X)_L \]

\( \text{Gains} \)

\( \mathcal{G}_m \) is the set of corresponding path

\( \mathcal{G}_m \) is the set of all loops that do

\( \mathcal{C}_l \) is the set of all loops that do

\( \mathcal{T} \) is the set of all loops that do