Design of Rate-Compatible Serially Concatenated Convolutional Codes

Alexandre Graell i Amat
Universitat Pompeu Fabra
Barcelona, Catalonia, Spain
Politecnico di Torino
Torino, Italy

Fredrik Brännström
Dept. of Signals and Systems
Chalmers Univ. of Technology
Göteborg, Sweden

Lars K. Rasmussen
Inst. for Telecommun. Research
University of South Australia
Mawson Lakes, Australia

1. Motivation
2. System Model
3. Upper Bounds
4. EXIT Chart Analysis
5. Numerical Examples
6. Conclusions
Motivation

- Turbo-like Codes approach capacity to within a few fractions of a decibel.
Motivation

- Turbo-like Codes approach capacity to within a few fractions of a decibel.
- There is still practical need for improvements in terms of:
  - Versatility: adaptive modulation
  - Throughput: high code rates
  - Simplicity: Low decoding complexity and short block length
Motivation

- Turbo-like Codes approach capacity to within a few fractions of a decibel.
- There is still practical need for improvements in terms of:
  - Versatility: adaptive modulation
  - Throughput: high code rates
  - Simplicity: Low decoding complexity and short block length

Objective:
Design a rate-compatible serially concatenated convolutional code with low decoding complexity and good performance in both the error floor and the waterfall regions over a wide range of code rates!
The outer code $C_O$ is concatenated in serial with the inner code $C_I$ through an interleaver $\pi$ of size $N$. 
The outer code $C_O$ is concatenated in serial with the inner code $C_I$ through an interleaver $\pi$ of size $N$.

The outer code $C_O$ consists of an encoder $C_a$ and a puncturer $P_a$. 

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005
System Model: Classical Serially Concatenated Code

- The outer code \( C_O \) is concatenated in serial with the inner code \( C_I \) through an interleaver \( \pi \) of size \( N \).

- The outer code \( C_O \) consists of an encoder \( C_a \) and a puncturer \( P_a \).

- The inner code \( C_I \) consists of an encoder \( C_b \) and a puncturer \( P_b^s \) for the systematic bits and a puncturer \( P_b^p \) for the parity bits.
System Model: Classical Serially Concatenated Code

- The outer code $C_O$ is concatenated in serial with the inner code $C_I$ through an interleaver $\pi$ of size $N$.

- The outer code $C_O$ consists of an encoder $C_a$ and a puncturer $P_a$.

- The inner code $C_I$ consists of an encoder $C_b$ and a puncturer $P_b^s$ for the systematic bits and a puncturer $P_b^p$ for the parity bits.

- The outer code has code rate $R_O$ and the inner code has code rate $R_I$, resulting in an overall code rate $R = R_O R_I$. 

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005
The outer code $C_O$ is concatenated in serial with the inner code $C_I$ through an interleaver $\pi$ of size $N$.

- The outer code $C_O$ consists of an encoder $C_a$ and a puncturer $P_a$.
- The inner code $C_I$ consists of an encoder $C_b$ and a puncturer $P^s_b$ for the systematic bits and a puncturer $P^p_b$ for the parity bits.
- The outer code has code rate $R_O$ and the inner code has code rate $R_I$, resulting in an overall code rate $R = R_O R_I$.
- In standard rate-compatible codes, the puncturing in $C_I$ is such that $R_I \leq 1$, limiting $R \leq R_O$. 
The outer code $C_O$ is concatenated in serial with the inner code $C_I$ through an interleaver $\pi$ of size $N$.

The outer code $C_O$ consists of an encoder $C_a$ and a puncturer $P_a$.

The inner code $C_I$ consists of an encoder $C_b$ and a puncturer $P^s_b$ for the systematic bits and a puncturer $P^p_b$ for the parity bits.

The outer code has code rate $R_O$ and the inner code has code rate $R_I$, resulting in an overall code rate $R = R_O R_I$.

In standard rate-compatible codes, the puncturing in $C_I$ is such that $R_I \leq 1$, limiting $R \leq R_O$.

For high $R$ the increasing value of $R_O$ causes an interleaver gain penalty resulting in a high error floor.
The inner code may be punctured beyond the unitary rate, resulting in $R > R_O$. 

- The inner code may be punctured beyond the unitary rate, resulting in $R > R_O$.
- The interleaver gain for low rates is also kept for high rates by moving the heavy puncturing from the outer code to the inner code.
The inner code may be punctured beyond the unitary rate, resulting in $R > R_O$.

The interleaver gain for low rates is also kept for high rates by moving the heavy puncturing from the outer code to the inner code.

The performance depends on the puncturing patterns for $P_a$, $P_b^s$, and $P_b^p$. 
The inner code may be punctured beyond the unitary rate, resulting in $R > R_O$.

The interleaver gain for low rates is also kept for high rates by moving the heavy puncturing from the outer code to the inner code.

The performance depends on the puncturing patterns for $\mathcal{P}_a$, $\mathcal{P}_b^s$, and $\mathcal{P}_b^p$.

**Our objective:**

Design $\mathcal{P}_a$, $\mathcal{P}_b^s$, and $\mathcal{P}_b^p$ to give good performance in both the error floor (EF) and the waterfall (WF) regions over a wide range of code rates!
**System Model: Classical Serially Concatenated Code**

The inner code may be punctured beyond the unitary rate, resulting in $R > R_O$.

The interleaver gain for low rates is also kept for high rates by moving the heavy puncturing from the outer code to the inner code.

The performance depends on the puncturing patterns for $P_a$, $P_b^s$, and $P_b^p$.

**Our objective:**
Design $P_a$, $P_b^s$, and $P_b^p$ to give good performance in both the error floor (EF) and the waterfall (WF) regions over a wide range of code rates!

**Our solution:**
Employ upper bounds based on uniform interleavers for the EF region and extrinsic information transfer (EXIT) charts analysis for the WF region.
System Model: Equivalent Serially Concatenated Code
System Model: Equivalent Serially Concatenated Code
System Model: Equivalent Serially Concatenated Code

- $C_1$ and $C_2$ are the parity parts of $C_a$ and $C_b$, respectively.
System Model: Equivalent Serially Concatenated Code

- $C_1$ and $C_2$ are the parity parts of $C_a$ and $C_b$, respectively.
- $P$ is the parity part of $P_a$.
System Model: Equivalent Serially Concatenated Code

- $C_1$ and $C_2$ are the parity parts of $C_a$ and $C_b$, respectively.
- $P$ is the parity part of $P_a$.
- $P_0$ and $P_1$ include a deinterleaved version of the puncturer $P_b^s$. 

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005
System Model: Equivalent Serially Concatenated Code

- $C_1$ and $C_2$ are the parity parts of $C_a$ and $C_b$, respectively.
- $P$ is the parity part of $P_a$.
- $P_0$ and $P_1$ include a deinterleaved version of the puncturer $P_b^s$.
- $P_2$ is identical to $P_b^p$, i.e., $x_2 = x^p$. 
Upper Bound on the Error Probability

- Optimize $\mathcal{P}_2$ in $C_L$, separate from $C_U$, to minimize the EF based on upper bounds using the IOWEF: $w \rightarrow x_2$. 


Upper Bound on the Error Probability

- Optimize $P_2$ in $C_L$, separate from $C_U$, to minimize the EF based on upper bounds using the IOWEF: $w \rightarrow x_2$.

- Optimize $\{P_0, P_1\}$ in $C_U$, separate from $C_L$, to minimize the EF based on upper bounds using the IOWEF: $v \rightarrow \{x_0, x_1\}$.


Upper Bound on the Error Probability

- Optimize $\mathcal{P}_2$ in $\mathcal{C}_L$, separate from $\mathcal{C}_U$, to minimize the EF based on upper bounds using the IOWEF: $w \rightarrow x_2$.

- Optimize $\{\mathcal{P}_0, \mathcal{P}_1\}$ in $\mathcal{C}_U$, separate from $\mathcal{C}_L$, to minimize the EF based on upper bounds using the IOWEF: $v \rightarrow \{x_0, x_1\}$.

- In both cases using a search algorithm that works incrementally, fulfilling the rate-compatibility constraint [1, 2].

---


Example Code

- $C_1$ and $C_2$ is the rate-1, 4-state, $(5/7)$ convolutional code.
Example Code

- $C_1$ and $C_2$ is the rate-1, 4-state, (5/7) convolutional code.
- $\mathcal{P}$ is chosen to puncture every other bit.
Example Code

- $C_1$ and $C_2$ is the rate-1, 4-state, $(5/7)$ convolutional code.
- $\mathcal{P}$ is chosen to puncture every other bit.
- Overall code rate: $R = \left( \rho_0 + \frac{1}{2}\rho_1 + \frac{3}{2}\rho_2 \right)$, where $0 \leq \rho_k \leq 1$ for $k = 0, 1, 2$. 

• $C_1$ and $C_2$ is the rate-1, 4-state, (5/7) convolutional code.
• $\mathcal{P}$ is chosen to puncture every other bit.
• Overall code rate: $R = \left(\rho_0 + \frac{1}{2}\rho_1 + \frac{3}{2}\rho_2\right)$, where $0 \leq \rho_k \leq 1$ for $k = 0, 1, 2$.
• $\mathcal{P}_0$ is chosen not to puncture any bits, i.e., $\rho_0 = 1$ and the overall code is fully systematic to assure invertibility.
Example Code

- $C_1$ and $C_2$ is the rate-1, 4-state, $(5/7)$ convolutional code.
- $P$ is chosen to puncture every other bit.
- Overall code rate: $R = \left( \rho_0 + \frac{1}{2} \rho_1 + \frac{3}{2} \rho_2 \right)$, where $0 \leq \rho_k \leq 1$ for $k = 0, 1, 2$.
- $P_0$ is chosen not to puncture any bits, i.e., $\rho_0 = 1$ and the overall code is fully systematic to assure invertibility.
- Achievable code rates: $1/3 \leq R \leq 1$. 

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005

28
Example Code in Error Floor Region

- Overall code rate: $R = \left( \rho_0 + \frac{1}{2} \rho_1 + \frac{3}{2} \rho_2 \right)$, where $0 \leq \rho_k \leq 1$ for $k = 0, 1, 2$. 
Example Code in Error Floor Region

- Overall code rate: \( R = (\rho_0 + \frac{1}{2}\rho_1 + \frac{3}{2}\rho_2) \), where \( 0 \leq \rho_k \leq 1 \) for \( k = 0, 1, 2 \).
- The period of the puncturing pattern in \( \mathcal{P}_2 \) is chosen to 300, hence the period of the puncturing pattern in \( \mathcal{P}_1 \) is 100.
Example Code in Error Floor Region

- Overall code rate: \( R = \left( \rho_0 + \frac{1}{2} \rho_1 + \frac{3}{2} \rho_2 \right) \), where \( 0 \leq \rho_k \leq 1 \) for \( k = 0, 1, 2 \).

- The period of the puncturing pattern in \( \mathcal{P}_2 \) is chosen to 300, hence the period of the puncturing pattern in \( \mathcal{P}_1 \) is 100.

- This means that \( \rho_1 = \frac{d_1}{100} \) for \( d_1 = 0, 1, \ldots, 100 \) and \( \rho_2 = \frac{d_2}{300} \) for \( d_2 = 0, 1, \ldots, 300 \).
Example Code in Error Floor Region

- Overall code rate: $R = \left( \rho_0 + \frac{1}{2} \rho_1 + \frac{3}{2} \rho_2 \right)$, where $0 \leq \rho_k \leq 1$ for $k = 0, 1, 2$.
- The period of the puncturing pattern in $\mathcal{P}_2$ is chosen to 300, hence the period of the puncturing pattern in $\mathcal{P}_1$ is 100.
- This means that $\rho_1 = \frac{d_1}{100}$ for $d_1 = 0, 1, \ldots, 100$ and $\rho_2 = \frac{d_2}{300}$ for $d_2 = 0, 1, \ldots, 300$.
- Achievable code rates: $R = \frac{200}{L}$, where $L = 200, 201, \ldots, 600$. 

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005
• Overall code rate: \( R = \left( \rho_0 + \frac{1}{2} \rho_1 + \frac{3}{2} \rho_2 \right) \), where \( 0 \leq \rho_k \leq 1 \) for \( k = 0, 1, 2 \).

• The period of the puncturing pattern in \( \mathcal{P}_2 \) is chosen to 300, hence the period of the puncturing pattern in \( \mathcal{P}_1 \) is 100.

• This means that \( \rho_1 = \frac{d_1}{100} \) for \( d_1 = 0, 1, \ldots, 100 \) and \( \rho_2 = \frac{d_2}{300} \) for \( d_2 = 0, 1, \ldots, 300 \).

• Achievable code rates: \( R = 200/L \), where \( L = 200, 201, \ldots, 600 \).

• For example, there are 101 ways of choosing \( \{\rho_1, \rho_2\} \) for \( R = 1/2 \).
Overall code rate: \( R = \left( \rho_0 + \frac{1}{2}\rho_1 + \frac{3}{2}\rho_2 \right) \), where \( 0 \leq \rho_k \leq 1 \) for \( k = 0, 1, 2 \).

The period of the puncturing pattern in \( \mathcal{P}_2 \) is chosen to 300, hence the period of the puncturing pattern in \( \mathcal{P}_1 \) is 100.

This means that \( \rho_1 = \frac{d_1}{100} \) for \( d_1 = 0, 1, \ldots, 100 \) and \( \rho_2 = \frac{d_2}{300} \) for \( d_2 = 0, 1, \ldots, 300 \).

Achievable code rates: \( R = \frac{200}{L} \), where \( L = 200, 201, \ldots, 600 \).

For example, there are 101 ways of choosing \( \{\rho_1, \rho_2\} \) for \( R = 1/2 \).

Maximizing \( d_2 \) minimizes the required SNR in the EF for all code rates!
Required SNR to Reach $P_b = 10^{-9}$ in the EF
The EXIT function for the outer code is independent of the SNR:

\[ I_{E(v)} = T_v(I_{A(v)}) = T_v(I_{E(w)}) \]

**EXIT Chart Analysis: Classical Serially Concatenated Convolutional Code**

- The EXIT function for the outer code is independent of the SNR:

\[ I_{E(v)} = T_v(I_{A(v)}) = T_v(I_{E(w)}) \]

- The EXIT function for the inner code depends on the SNR and the code rate:

\[ I_{E(w)} = T_w(I_{A(w)}, RE_b/N_0) = T_w(I_{E(v)}, RE_b/N_0) \]

EXIT Chart Analysis: Classical Serially Concatenated Convolutional Code

- The EXIT function for the outer code is independent of the SNR:

\[ I_{E(v)} = T_v(I_{A(v)}) = T_v(I_{E(w)}) \]

- The EXIT function for the inner code depends on the SNR and the code rate:

\[ I_{E(w)} = T_w(I_{A(w)}, RE_b/N_0) = T_w(I_{E(v)}, RE_b/N_0) \]

- Plot \( I_{E(w)} \) versus \( I_{A(w)} \) in the same plot as \( I_{A(v)} \) versus \( I_{E(v)} \) to create the EXIT chart [3].

EXIT Chart Analysis for $\rho_1 = 20/100$ and $\rho_2 = 20/300$, i.e., $R = 5/6$
EXIT Chart Analysis for $\rho_1 = 20/100$ and $\rho_2 = 20/300$, i.e., $R = 5/6$
The EXIT functions for both the upper code and the lower code depend on the SNR and the code rate:

\[
I_E(v) = T_v(I_A(v), RE_b/N_0) = T_v(I_E(w), RE_b/N_0)
\]

\[
I_E(w) = T_w(I_A(w), RE_b/N_0) = T_w(I_E(v), RE_b/N_0)
\]
EXIT Chart Analysis: Equivalent Serially Concatenated Convolutional Code

- The EXIT functions for both the upper code and the lower code depend on the SNR and the code rate:

\[ I_E(v) = T_v(I_A(v), RE_b/N_0) = T_v(I_E(w), RE_b/N_0) \]
\[ I_E(w) = T_w(I_A(w), RE_b/N_0) = T_w(I_E(v), RE_b/N_0) \]

- Plot \( I_E(w) \) versus \( I_A(w) \) in the same plot as \( I_A(v) \) versus \( I_E(v) \) to create an EXIT chart.
EXIT Chart Analysis for $\rho_1 = 20/100$ and $\rho_2 = 20/300$, i.e., $R = 5/6$
Design Approach for the Example Code in the Waterfall Region

- Find all 101 EXIT functions for the upper code, $\rho_1 = \frac{d_1}{100}$ for $d_1 = 0, 1, \ldots, 100$,

$$I_{E(v)} = T_v(I_A(v), I_A(x), \rho_1).$$

Design Approach for the Example Code in the Waterfall Region

- Find all 101 EXIT functions for the upper code, \( \rho_1 = \frac{d_1}{100} \) for \( d_1 = 0, 1, \ldots, 100 \),

\[
I_{E(v)} = T_v(I_A(v), I_A(x), \rho_1).
\]

- Find all 301 EXIT functions for the lower code, \( \rho_2 = \frac{d_2}{300} \) for \( d_2 = 0, 1, \ldots, 300 \),

\[
I_{E(w)} = T_w(I_A(w), I_A(x), \rho_2).
\]

Design Approach for the Example Code in the Waterfall Region

- Find all 101 EXIT functions for the upper code, $\rho_1 = \frac{d_1}{100}$ for $d_1 = 0, 1, \ldots, 100$,

  \[ I_E(v) = T_v(I_A(v), I_A(x), \rho_1). \]

- Find all 301 EXIT functions for the lower code, $\rho_2 = \frac{d_2}{300}$ for $d_2 = 0, 1, \ldots, 300$,

  \[ I_E(w) = T_w(I_A(w), I_A(x), \rho_2). \]

- Project all combinations of upper and lower EXIT functions onto an EXIT chart and find the required SNR to reach $P_b = 10^{-5}$.

---

EXIT Functions for the Lower Code

\[ I_E(w) = T_w(I_A(w), I_A(x), \frac{50}{300}) \]

\[ I_E(w) = T_w(I_A(w), I_A(x), \frac{200}{300}) \]
Required SNR to Reach $P_b = 10^{-9}$ in the EF and $P_b = 10^{-5}$ in the WF

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005
Design Approach for the EF and the WF

- Find the rate-compatible puncturing patterns in $P_1$ and $P_2$, separately.
Design Approach for the EF and the WF

- Find the rate-compatible puncturing patterns in $P_1$ and $P_2$, separately.
- Calculate the upper bounds for all combinations of $\rho_1$ and $\rho_2$ and find the required SNR to reach $P_b = 10^{-9}$. 
Design Approach for the EF and the WF

- Find the rate-compatible puncturing patterns in $P_1$ and $P_2$, separately.
- Calculate the upper bounds for all combinations of $\rho_1$ and $\rho_2$ and find the required SNR to reach $P_b = 10^{-9}$.

Maximizing $d_2$ minimizes the required SNR in the EF for all code rates!
Find the rate-compatible puncturing patterns in $P_1$ and $P_2$, separately.

Calculate the upper bounds for all combinations of $\rho_1$ and $\rho_2$ and find the required SNR to reach $P_b = 10^{-9}$.

Maximizing $d_2$ minimizes the required SNR in the EF for all code rates!

Project all combinations of upper and lower EXIT functions onto an EXIT chart and find the required SNR to reach $P_b = 10^{-5}$. 
Find the rate-compatible puncturing patterns in $P_1$ and $P_2$, separately.

Calculate the upper bounds for all combinations of $\rho_1$ and $\rho_2$ and find the required SNR to reach $P_b = 10^{-9}$.

Maximizing $d_2$ minimizes the required SNR in the EF for all code rates!

Project all combinations of upper and lower EXIT functions onto an EXIT chart and find the required SNR to reach $P_b = 10^{-5}$.

Choosing $d_2 = d_1$, as long as possible, minimizes the required SNR in the WF for all code rates!
Puncturing Strategy for the Example Code

\[ L = d_0 + d_1 + d_2 = \frac{200}{R} \]

Minimum, EF optimized, WF optimized, Compromise

Design of Rate-Compatible Serially Concatenated Convolutional Codes
Alexandre Graell i Amat, Fredrik Brännström, and Lars K. Rasmussen, 2005

54
Minimum Required SNR ($N = 3000$)
Performance after 10 and 20 iterations, Optimized for the WF ($N = 3000$)
Performance after 10 and 20 iterations, Optimized for the WF ($N = 24600$)
Conclusions

- Design criteria for serially concatenated convolutional codes:
  - Use upper bounds to optimize the rate-compatible puncturing patterns for the upper and lower code separately to give good performance in the error floor region.

Conclusions

- Design criteria for serially concatenated convolutional codes:
  - Use upper bounds to optimize the rate-compatible puncturing patterns for the upper and lower code separately to give good performance in the error floor region.
  - Employ EXIT chart analysis to optimize the permeability rates for the upper and lower code to give good performance in the waterfall region.


Conclusions

- Design criteria for serially concatenated convolutional codes:
  - Use upper bounds to optimize the rate-compatible puncturing patterns for the upper and lower code separately to give good performance in the error floor region.
  - Employ EXIT chart analysis to optimize the permeability rates for the upper and lower code to give good performance in the waterfall region.

**Outcome:**

A rate-compatible serially concatenated convolutional code with low decoding complexity and good performance in both the error floor and the waterfall regions over a wide range of code rates!


