The Progressive Edge Growth Algorithm

Telecommunications Laboratory
Alex Balatsoukas-Stimming

Technical University of Crete

July 22, 2009

Outline

Progressive Edge Growth

Progressive Edge-Growth Algorithm:

for $j = 0$ to $n - 1$ **do**
begin
for $k = 0$ to $d_{s_j} - 1$ **do**
begin
if $k = 0$
$E_{s_j}^0 \longleftarrow$ edge (c_i, s_j), where $E_{s_j}^0$ is the first edge incident to s_j and c_i is a check node such that it has the lowest check-node degree under the current graph setting $E_{s_0} \cup E_{s_1} \cup \cdots \cup E_{s_{j-1}}$.
else
expand a subgraph from symbol node s_j up to depth l under the current graph setting such that the cardinality of $\mathcal{N}_{s_j}^l$ stops increasing but is less than m, or $\bar{\mathcal{N}}_{s_j}^l \neq \emptyset$ but $\bar{\mathcal{N}}_{s_j}^{l+1} = \emptyset$, then $E_{s_j}^k \longleftarrow$ edge (c_i, s_j), where $E_{s_j}^k$ is the kth edge incident to s_j and c_i is a check node picked from the set $\bar{\mathcal{N}}_{s_j}^l$ having the lowest check-node degree.
end
end

Modified Progressive Edge Growth

PEG Algorithm for Establishing Edges of V_s^p:

for $j = 0$ to $m - 1$ **do**
begin
 for $k = 0$ to $d_{s_j} - 1$ **do**
 begin
  if $k = 0$
   $E_{s_j}^0 \longleftarrow$ edge (c_j, s_j), where $E_{s_j}^0$ is the first edge incident to s_j. This edge corresponds to the "1" in the diagonal line of matrix H^p.
  else
   expand a subgraph from symbol node s_j up to depth l under the current graph setting such that $\bar{\mathcal{N}}_{s_j}^l \cap \{c_0, c_1, \ldots, c_{j-1}\} \neq \varnothing$ but $\bar{\mathcal{N}}_{s_j}^{l+1} \cap \{c_0, c_1, \ldots, c_{j-1}\} = \varnothing$, or the cardinality of $\mathcal{N}_{s_j}^l$ stops increasing, then $E_{s_j}^k \longleftarrow$ edge (c_i, s_j), where $E_{s_j}^k$ is the kth edge incident to s_j and c_i is a check node picked from the set $\bar{\mathcal{N}}_{s_j}^l \cap \{c_0, c_1, \ldots, c_{j-1}\}$ having the lowest check-node degree.
 end
end

Simulations

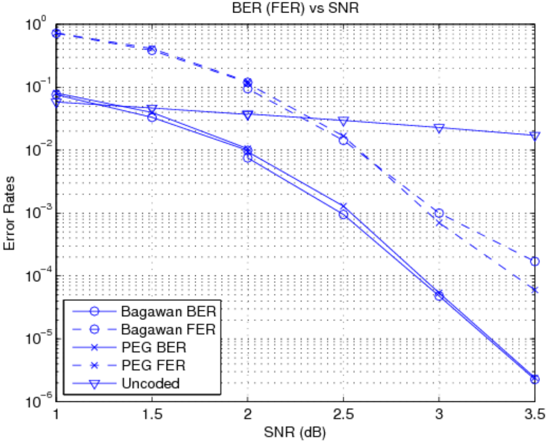

Figure: Rate-1/2, $n = 504$ PEG Code without Gaussian elimination used for quick encoding (all-zero codeword)

Simulations

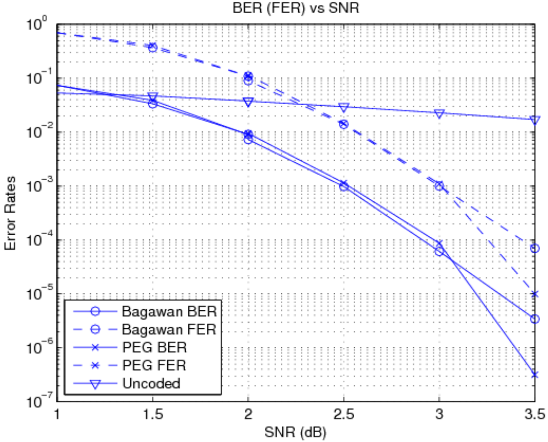

Figure: Rate-1/2, $n = 504$ PEG Code with Gaussian elimination used for quick encoding (all-zero codeword)

Simulations

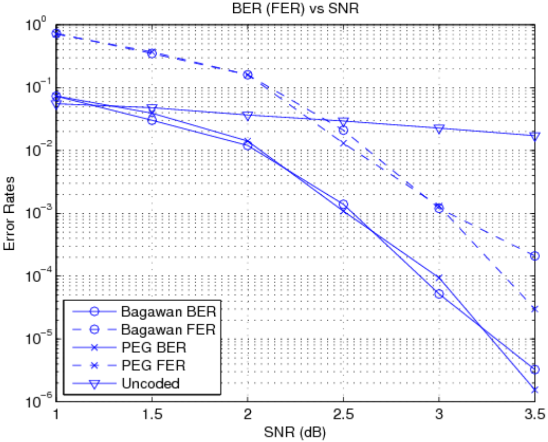

Figure: Rate-1/2, $n = 504$ PEG Code with Gaussian elimination used for quick encoding