Iterative Decoders with Deliberate Message Flips

Approaching Maximum Likelihood Decoding

Bane Vasic[1], Predrag Ivanis[2], David Declercq[3] Khoa LeTrung[3] and Elza Dupraz[4]

[1]University of Arizona, Tucson and [2]University of Belgrade [3]ETIS Lab, ENSEA, Université de Cergy-Pontoise-CNRS [4]Telecom, Bretagne

i-RISC Workshop, Cork, Ireland
Sept 2, 2015

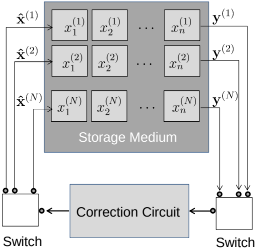
This work was supported in part by the NSF under Grants CCF-0963726, CCF–1314147 and ECCS-1500170, Fulbright Scholar Program, and by the Seventh Framework Programme of the European Union, under Grant Agreement number 309129 (i–RISC project)

Outline

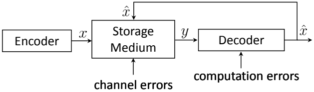
Outline

Memory Architecture

- Each k-bit user information is stored as a codeword of an (n, k) linear block code of length n and code rate $R = k/n$.
- The memory elements are unreliable and fail transiently and independently of each other – the von Neumann failure model, BSC(α_M)
- The correction circuit is also unreliable.



Memory



Equivalent Channel

**“Nothing in this presentation is novel.
The only novelty is random errors.
Even this joke is not novel.”**
Anonymous

“Is noise always bad?”
A Kirkland & Ellis lawyer

All Components are Faulty

Shashi's Theorem (Vasić and Chilappagari, 2007)

Let G be a $(\gamma, \rho, \alpha, (3/4 + \epsilon))$ expander for any $\epsilon > 0$. Our memory architecture can tolerate constant fraction of errors in all the components if

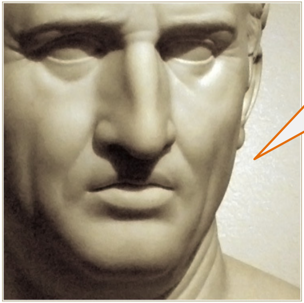
$$\alpha_M + \gamma(\rho - 2)\alpha_{\oplus} + \alpha_\gamma < \alpha(1 + 4\epsilon)(4\epsilon)/2$$

Philosophy

- Probabilistic behavior of an iterative decoder due to random deliberate errors in its logic gates can be exploited to our advantage and lead to an improved performance and reduced hardware redundancy.
- Iterative decoding can be viewed as a recursive procedure for Bethe free energy function minimization, and the randomness in a message update may help the decoder to escape from local minima.
- The decoder operates in stochastic gradient descent fashion, but the random perturbations do not require any additional hardware as they are built-in the faulty hardware itself. Thus the decoder harvests good deeds of logic gate errors.

Philosophy

Marcus Tullius Cicero



Clavus clavo eicitur !

One nail drives out another

Error errore eicitur !

Authors

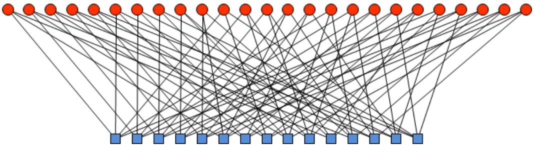
One error drives out another

Ingredients

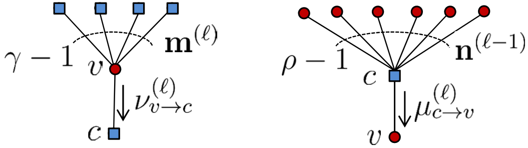
- Low-density parity check (LDPC) code and an iterative decoder
  - (γ, ρ)-regular LDPC codes
  - Tanner graph $G = (V \cup C, E)$, $|V| = n$ $|C| = n\gamma/\rho$ and $|E| = n\gamma$
- Noisy Gallager-B message updates
- Rewinding decoder, $\mathcal{F}^{\circlearrowleft^r}(L_R)$
  - If a codeword is not found after L_R iterations, $L_R \ll L$, the decoding algorithm is re-initialized with the word received from the channel.
  - Instead of running the whole L iterations, the decoder instead runs $r = L/L_R$ very short rounds.
- Critical gates must be perfect.

An example of a code graph G of an $(3, 5)$-regular LDPC code

$$H = \begin{pmatrix}
1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0\\
0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0\\
0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0\\
0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1&0&0&0&0&1\\
1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0\\
0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&1&0&0\\
0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0\\
0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&1\\
0&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&0\\
1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0\\
0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0\\
0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0\\
0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&0&0&1&0&0&1&0&0&0
\end{pmatrix}$$



Noisy Gallager-B



$$\nu^{(\ell)}_{v\to c} = \Phi(y_v, \mathbf{m}^{(\ell)}) + e^{(\ell)}_{MAJ}$$
$$\mu^{(\ell)}_{c\to v} = \Psi(\mathbf{n}^{(\ell-1)}) + e^{(\ell)}_{\oplus}$$

- In this talk: Φ and Ψ require only two types of logic gates: $(\gamma - 1)$-input majority logic (MAJ) gates and $(\rho - 1)$-input XOR gates.
- $e^{(\ell)}_{MAJ}$ and $e^{(\ell)}_{\oplus}$ are independent Bernoulli random variables with parameters α_{MAJ} and $\alpha_{\oplus}$.

Rewinding decoder, $\mathcal{F}^{\circlearrowleft^r}(L_R)$

- If a codeword is not found after L_R iterations, $L_R \ll L$, the decoding algorithm is re-initialized with the word received from the channel.
- Instead of running the whole L iterations, the decoder instead runs $r = L/L_R$ very short rounds.

$$\mathcal{F}^{\circlearrowleft^r}(L_R) = \underbrace{\mathcal{F}(L_R)\Diamond\mathcal{F}(L_R)\Diamond\cdots\Diamond\mathcal{F}(L_R)}_{r}.$$

Benefits of Rewinding

- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

Benefits of Rewinding

- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

Benefits of Rewinding

- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

Benefits of Rewinding

- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

Benefits of Rewinding

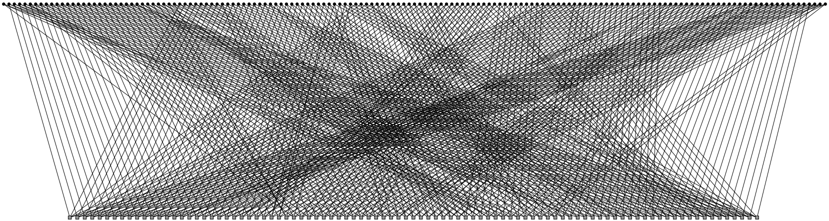
- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

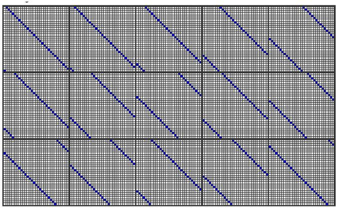
Benefits of Rewinding

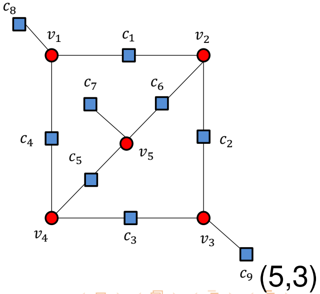
- Note that the plain noisy decoder with no rewinding, $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L)$.
- To allow the decoder to benefit from errors, large number of iterations is needed under some conditions of gate unreliability.
- However, too many logic gate errors can overwhelm the decoder, and lead to miscorrection.
- Rewinding is a key feature which prevents the accumulation of errors in the messages.
- Applicable to other decoders such as Probabilistic Gradient Descent Bit Flipping (not in this talk).

Outline

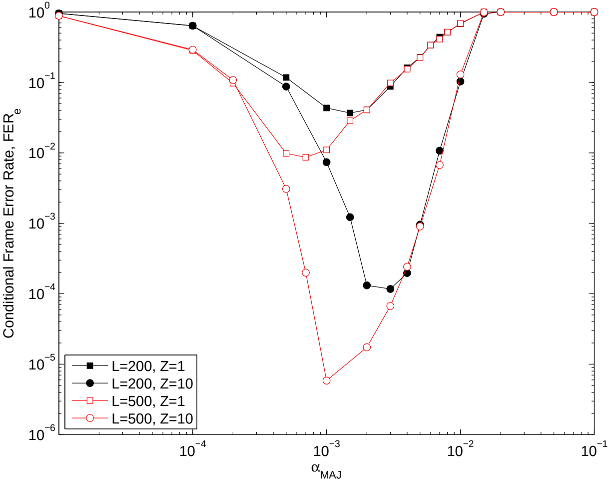
(5, 3) Trapping Sets in the Tanner (155, 64, 20) Code





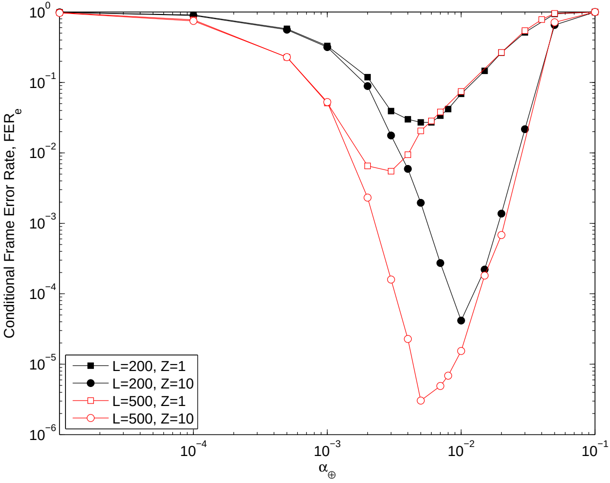


Conditional FER of the Tanner (155,64) code



Nosy MAJ

Conditional FER of the Tanner (155,64) code

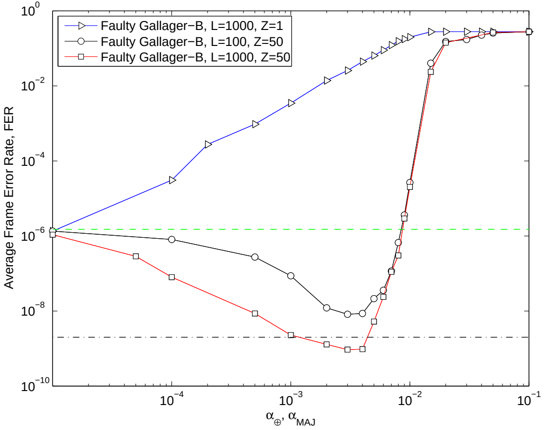


Nosy XOR

Optimal $\alpha_{\oplus} > 0$, $\alpha_{MAJ} > 0$

- FER is dominated by the most critical three-bit error patterns.
- The particular low-weight error pattern cannot be decoded by perfect hardware, but it can be decoded with non-zero probability for a wide range of gate error probabilities α_{MAJ} and $\alpha_{\oplus}$.
- After sufficient number of iterations, the minimum FER is not achieved for perfect gates but for some nonzero value of the gate error rates α_{MAJ} and $\alpha_{\oplus}$.
- For a broad range of gate error rates, our decoder actually benefits from logic gate errors.

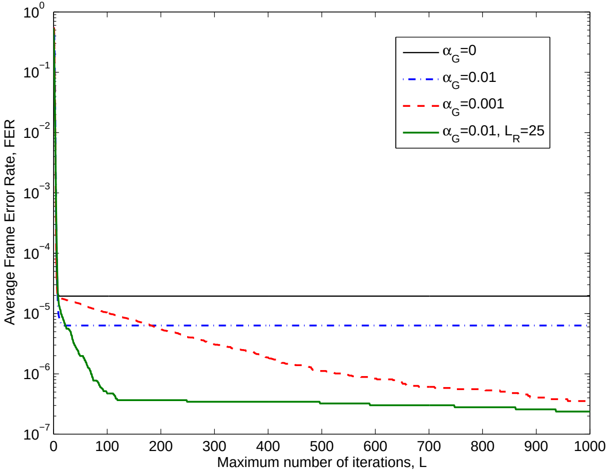
Average FER



— perfect Gallager-B — perfect min-sum decoder

- Both XOR and MAJ gates have the same failure rate $\alpha_M = 2 \times 10^{-3}$

The impact of L

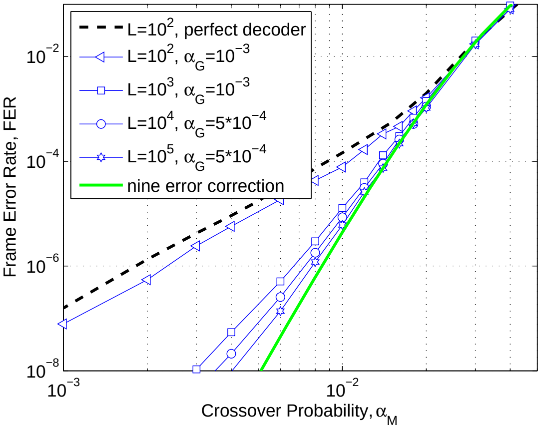


$$\alpha_M = 5 \times 10^{-3}$$

The Impact of the Number of Iterations L

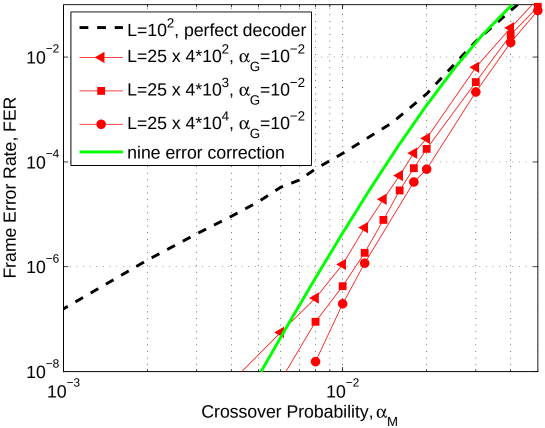
- Increasing the maximum number of iterations, L, reduces the probability that the error pattern remains uncorrected.
- The impact of L is especially noticeable for high reliability gates
  - Hardware errors cannot help much in annihilating trapping sets because the state transition probabilities in $\mathcal{W}$ are small for most transitions other than those that already exist in the perfect decoder.
  - Consequently, the convergence to the attractor of the codeword takes longer (the average convergence time also grows with n).
- On the other hand, increasing Z is has stronger effect when gates are very noisy.
- Prolonging the second stage of the decoding algorithm greatly improves the performance of a decoder made of the less reliable hardware.

Plain Decoder $\mathcal{F}$



- For all L, the decoder $\mathcal{F}$ outperforms the ideal decoder $\overline{\mathcal{F}}$.
- For large L, $\mathcal{F}$ approaches the nine-error correction decoder.

Rewinding Decoder $\mathcal{F}^{\circlearrowleft}$



- For $\alpha_G = 10^{-2}$, the rewind decoder $\mathcal{F}^{\circlearrowleft}$ performs beyond the $\lfloor \frac{d_{min}-1}{2} \rfloor$ bound.
- The total number of iterations $L = r \times L_R$ is kept the same as for $\mathcal{F}$.

Lessons Learned for Two Gate Error Rate Regimes

- For <u>more reliable</u> logic gates, large L is needed before the decoder start benefiting from the positive effects of logic gate errors.
  - Strategy: The perfect syndrome checker is used in the first twenty iterations, as the average *FER* rapidly decreases only at the beginning of the decoding.
  - Both the final bit-estimation circuit as well as the syndrome checker are turned-off and sufficient number of iterations is allowed before they are turned on again.
  - Clearly, this strategy results in energy saving as the perfect gates are used in a reduced number of iterations.
- For <u>less reliable</u> gates, errors correctable by the perfect decoder may turn uncorrectable, or lead to miscorrections as they may lead to large deviations from the trajectory of the perfect decoder.
  - A solution for this case is to rewind the decoder.
  - The higher the gate error rate, the lower optimal rewind period L_R

Outline

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages $\boldsymbol{\mu}$.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\mu}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\mu}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages $\boldsymbol{\mu}$.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\mu}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages $\boldsymbol{\mu}$.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\mu}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages μ.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft 1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\mu}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages μ.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages $\boldsymbol{\mu}$.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamical System of a Noisy Decoder

- We consider $\mathcal{F}(L) = \mathcal{F}^{\circlearrowleft^1}(L_R = L)$ – no rewinding.

$$\boldsymbol{\mu}^{(\ell)} = \Upsilon_C(\boldsymbol{\mu}^{(\ell-1)}) + \mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$$

- $\boldsymbol{\mu}^{(\ell)} = (\mu_c^{(\ell)})_{c \in C}$ – the outgoing messages from check nodes.
- Υ_C is the composition of Φ and Ψ, and define the dynamical system of the perfect decoder, $\overline{\mathcal{F}}$.
- $\mathbf{e}_{\boldsymbol{\mu}}^{(\ell)}$ (of length $= m\rho$) are the realization of errors at time ℓ that affect computation of messages $\boldsymbol{\mu}$.
- Since Φ is the function of the memory output $\mathbf{y}$, Υ_C also depends on $\mathbf{y}$, thus the transition probabilities depend on the the channel error vector $\mathbf{e}$, and for a given decoder we have an ensemble of Markov chains $\{\mathcal{W}_{\mathbf{e}}\}_{\mathbf{e} \in \{0,1\}^n}$.

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0,1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \mathtt{Pr}\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1-\alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0, 1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \Pr\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1 - \alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ

α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0,1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \mathtt{Pr}\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1-\alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0, 1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \mathtt{Pr}\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1 - \alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0,1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \mathtt{Pr}\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1-\alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0, 1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \mathtt{Pr}\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1 - \alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

Dynamics of Perfect and Noisy Decoders, $\overline{\mathcal{F}}$ and $\mathcal{F}$

- For a given a decoder $\mathcal{F}$ and error pattern $\mathbf{e}$ in the memory elements, consider the Markov chain $\mathcal{W}_{\mathbf{e}}$, with the state space $S = \{0, 1\}^{|C|\rho}$, and the transition probability matrix $P = (p_{\varepsilon,\delta})_{\varepsilon,\delta \in S}$.
- The transition probabilities between states, $p_{\varepsilon,\delta} = \Pr\{\boldsymbol{\mu}^{(\ell)} = \delta | \boldsymbol{\mu}^{(\ell-1)} = \varepsilon\}$, depend on $\alpha_{\oplus}$ and α_{MAJ}.
- Due to independence of logic gate errors, their effect can be combined into a single conditional probability α_G.
- Let $\bar{\delta} = \Upsilon_C(\varepsilon)$ be the state of the perfect decoder $\overline{\mathcal{F}}$ reached from the state ε.

$$p_{\varepsilon,\delta} = p_{\varepsilon,\bar{\delta}} \alpha_G^{d_{\bar{\delta},\delta}} (1 - \alpha_G)^{|C|\rho - d_{\bar{\delta},\delta}}$$

$d_{\bar{\delta},\delta}$ – the Hamming distance between the binary vectors $\bar{\delta}$ and δ
α_G – the probability that a single XOR gate output in a noisy decoder $\mathcal{F}$ is different from the corresponding XOR gate output in the perfect decoder $\overline{\mathcal{F}}$

α_G

Gate failure probability

$$\alpha_G = (1-\alpha_\oplus)\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) + \alpha_\oplus\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1})$$

$$\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) = \mathtt{Pr}\{\text{odd number of MAJ gate errors}\}$$

$$\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1}) = \mathtt{Pr}\{\text{even number of MAJ gate errors}\}$$

Approximation for low logic gate error rates

$$\alpha_G \approx (\rho-1)\alpha_{MAJ} + \alpha_\oplus$$

α_G

Gate failure probability

$$\alpha_G = (1-\alpha_\oplus)\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) + \alpha_\oplus\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1})$$

$$\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) = \mathtt{Pr}\{\text{odd number of MAJ gate errors}\}$$
$$\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1}) = \mathtt{Pr}\{\text{even number of MAJ gate errors}\}$$

Approximation for low logic gate error rates

$$\alpha_G \approx (\rho-1)\alpha_{MAJ} + \alpha_\oplus$$

α_G

Gate failure probability

$$\alpha_G = (1-\alpha_\oplus)\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) + \alpha_\oplus\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1})$$

$$\frac{1}{2}(1-(1-2\alpha_{MAJ})^{\rho-1}) = \texttt{Pr}\{\text{odd number of MAJ gate errors}\}$$
$$\frac{1}{2}(1+(1-2\alpha_{MAJ})^{\rho-1}) = \texttt{Pr}\{\text{even number of MAJ gate errors}\}$$

Approximation for low logic gate error rates

$$\alpha_G \approx (\rho-1)\alpha_{MAJ} + \alpha_\oplus$$

Classification of States of $\mathcal{W}_{\mathbf{e}}$

- For a given decoder $\mathcal{F}$, a codeword $\mathbf{x} \in \mathcal{C}$, and the error pattern $\mathbf{e}$ in the memory elements, we define Markov chain $\mathcal{W}_{\mathbf{e}}$ with the set of states

$$S = S_{\mathbf{x}} \cup S_{\sim\mathbf{x}} \cup S_{\sim\mathcal{C}},$$

  - $S_{\mathbf{x}}$ — subset of S for which all parity check are satisfied, and the variable node decisions form the codeword $\mathbf{x}$
  - $S_{\sim\mathbf{x}}$ — set of states for which all parity check are satisfied, and the variable node decisions form a codeword different from $\mathbf{x}$.
  - $S_{\sim\mathcal{C}}$ — set of states for which the variable node decisions are not codewords.

Classification of States of $\mathcal{W}_{\mathbf{e}}$

- For a given decoder $\mathcal{F}$, a codeword $\mathbf{x} \in \mathcal{C}$, and the error pattern $\mathbf{e}$ in the memory elements, we define Markov chain $\mathcal{W}_{\mathbf{e}}$ with the set of states

$$S = S_{\mathbf{x}} \cup S_{\sim \mathbf{x}} \cup S_{\sim \mathcal{C}},$$

  - $S_{\mathbf{x}}$ — subset of S for which all parity check are satisfied, and the variable node decisions form the codeword $\mathbf{x}$
  - $S_{\sim \mathbf{x}}$ — set of states for which all parity check are satisfied, and the variable node decisions form a codeword different from $\mathbf{x}$.
  - $S_{\sim \mathcal{C}}$ — set of states for which the variable node decisions are not codewords.

Classification of States of $\mathcal{W}_{\mathbf{e}}$

- For a given decoder $\mathcal{F}$, a codeword $\mathbf{x} \in \mathcal{C}$, and the error pattern $\mathbf{e}$ in the memory elements, we define Markov chain $\mathcal{W}_{\mathbf{e}}$ with the set of states

$$S = S_{\mathbf{x}} \cup S_{\sim\mathbf{x}} \cup S_{\sim\mathcal{C}},$$

  - $S_{\mathbf{x}}$ — subset of S for which all parity check are satisfied, and the variable node decisions form the codeword $\mathbf{x}$
  - $S_{\sim\mathbf{x}}$ — set of states for which all parity check are satisfied, and the variable node decisions form a codeword different from $\mathbf{x}$.
  - $S_{\sim\mathcal{C}}$ — set of states for which the variable node decisions are not codewords.

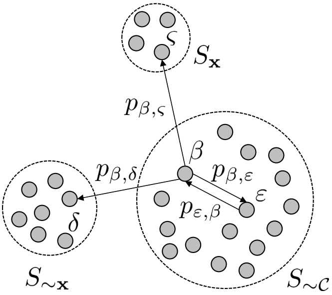
Classification of States of $\mathcal{W}_{\mathbf{e}}$

- For a given decoder $\mathcal{F}$, a codeword $\mathbf{x} \in \mathcal{C}$, and the error pattern $\mathbf{e}$ in the memory elements, we define Markov chain $\mathcal{W}_{\mathbf{e}}$ with the set of states

$$S = S_{\mathbf{x}} \cup S_{\sim\mathbf{x}} \cup S_{\sim\mathcal{C}},$$

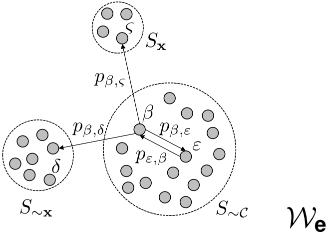
  - $S_{\mathbf{x}}$ — subset of S for which all parity check are satisfied, and the variable node decisions form the codeword $\mathbf{x}$
  - $S_{\sim\mathbf{x}}$ — set of states for which all parity check are satisfied, and the variable node decisions form a codeword different from $\mathbf{x}$.
  - $S_{\sim\mathcal{C}}$ — set of states for which the variable node decisions are not codewords.

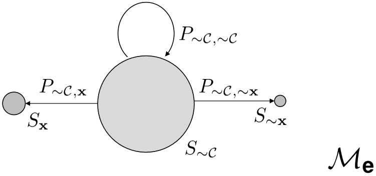
Closeness to a Codeword



- When the perfect syndrome checker is turned on, and if the Markov chain is the state $\beta \in S_{\mathbf{x}} \cup S_{\sim\mathbf{x}}$, the decoding is terminated, and the Markov chain stays in β.
- Thus, the states in $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are absorbing.

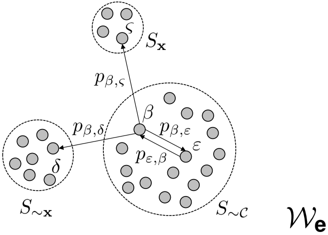
Reduced Chain

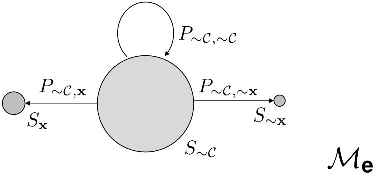




All the states in $S_{\mathbf{x}}$ from $\mathcal{W}_{\mathbf{e}}$ are lumped into a single state in $\mathcal{M}_{\mathbf{e}}$. Similarly, the states $S_{\sim\mathbf{x}}$ are are also lumped into a single state

Reduced Chain





All the states in $S_{\mathbf{x}}$ from $\mathcal{W}_{\mathbf{e}}$ are lumped into a single state in $\mathcal{M}_{\mathbf{e}}$. Similarly, the states $S_{\sim\mathbf{x}}$ are are also lumped into a single state

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

Absorbing Chain Transition Probability Matrix

- Note that $P_{\sim\mathcal{C},\mathbf{x}}$, $P_{\sim\mathcal{C},\sim\mathbf{x}}$ and $P_{\sim\mathcal{C},\sim\mathcal{C}}$ do not depend on ℓ due to homogeneity.

$$P = \begin{pmatrix} 1 & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ P_{\sim\mathcal{C},\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathbf{x}} & P_{\sim\mathcal{C},\sim\mathcal{C}} \end{pmatrix} = \begin{pmatrix} I_2 & \mathbf{0} \\ R & Q \end{pmatrix}$$

- The transition diagram of the transient states is a strongly connected graph
- Q does not have any nonzero entries
- The transition probabilities from transient to absorbing states $S_{\mathbf{x}}$ and $S_{\sim\mathbf{x}}$ are given by the matrix $R = (R_{\mathbf{x}}, R_{\sim\mathbf{x}})$

$$R_{\mathbf{x}} = P_{\sim\mathcal{C},\mathbf{x}},$$
$$R_{\sim\mathbf{x}} = P_{\sim\mathcal{C},\sim\mathbf{x}}.$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e} \in \{0,1\}^n} \Pr\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\Pr\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1 - \alpha_M)^{n - w(\mathbf{e})}$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e}\in\{0,1\}^n} \Pr\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\Pr\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1-\alpha_M)^{n-w(\mathbf{e})}$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e}\in\{0,1\}^n} \Pr\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\Pr\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1-\alpha_M)^{n-w(\mathbf{e})}$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e}\in\{0,1\}^n} \Pr\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\Pr\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1-\alpha_M)^{n-w(\mathbf{e})}$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e}\in\{0,1\}^n} \mathtt{Pr}\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\mathtt{Pr}\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1-\alpha_M)^{n-w(\mathbf{e})}$$

$FER_{\mathbf{e}}^{(\ell)}$ and $MER_{\mathbf{e}}^{(\ell)}$

- Let us assume that all-zero codeword is transmitted $\mathbf{x} = \mathbf{0}$

$$FER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}} \cup S_{\sim\mathcal{C}}\}$$
$$MER_{\mathbf{e}}^{(\ell)} = \Pr\{\boldsymbol{\mu}^{(\ell)} \in S_{\sim\mathbf{0}}\}$$

- Average performance

$$FER(\mathcal{D}) = \sum_{\mathbf{e}\in\{0,1\}^n} \mathtt{Pr}\{\mathbf{e}\} \times FER_{\mathbf{e}}(\mathcal{D})$$
$$\mathtt{Pr}\{\mathbf{e}\} = \alpha_M^{w(\mathbf{e})}(1-\alpha_M)^{n-w(\mathbf{e})}$$

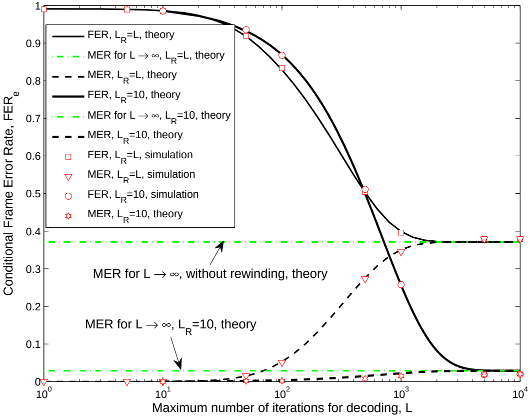
$FER_{\mathbf{e}}^{(\infty)} = MER_{\mathbf{e}}^{(\infty)}$

Theorem

For the noisy Gallager-B decoding algorithm $\mathcal{D} = \mathcal{F}(L, L)$ on any LDPC code $\mathcal{C}$, and sufficiently large L

$$FER_{\mathbf{e}}^{(L)}(\mathcal{D}) \approx MER_{\mathbf{e}}^{(L)}(\mathcal{D}).$$

Illustration on a (5,1,5) Repetition Code



Conditional FER for error pattern **e**=(11000) as a function of Z

$\alpha_{\oplus MAJ} = 0.01$, $L_R = 25$

Outline

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  - (i) codewords - which are the fixed points
  - (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  - (i) codewords - which are the fixed points
  - (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  (i) codewords - which are the fixed points
  (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system $(\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$ includes
  - (i) codewords - which are the fixed points
  - (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  (i) codewords - which are the fixed points
  (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  - (i) codewords - which are the fixed points
  - (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  - (i) codewords - which are the fixed points
  - (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

Discussion

- For a perfect decoder, $\alpha_{MAJ} = 0$ and $\alpha_{\oplus} = 0$, the transitions between states are deterministic, and the attractor basin of a dynamical system ($\boldsymbol{\mu}^{(\ell)} = \Upsilon_V(\boldsymbol{\mu}^{(\ell-1)})$) includes
  (i) codewords - which are the fixed points
  (ii) trapping sets - which can be either fixed points or cycle attractors.
- Perfect decoder may oscillate between these states, thus failing to converge to a codeword.
- On the other hand, in a noisy decoder every state can be reached with a nonzero probability.
- Thus, the faulty decoder will eventually converge to a codeword - either correct or incorrect one.
- When the decoding algorithm have small probability of miscorrection in the first decoding iterations, it is better to use the rewinding decoder with $r = L/L_R$ rounds, $L_R \ll L$.

End of Talk

Thank you!

References I

B. Vasić and S. K. Chilappagari. An information theoretical framework for analysis and design of nanoscale fault-tolerant memories based on low-density parity-check codes. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 54(11): 2438–2446, Nov. 2007.