

Stabilité des Mémoires de Taylor-Kuznetsov construites à partir d'un Décodeur LDPC de type Gallager B

Elsa DUPRAZ¹, David DECLERCQ¹, Bane VASIĆ²

¹ETIS - ENSEA / Univ. Cergy-Pontoise / CNRS UMR-8051, Cergy-Pontoise, France

²Dept. of Electrical and Computer Eng., University of Arizona, Tucson, AZ 85721, USA

elsa.dupraz@ensea.fr, david.declercq@ensea.fr
vasic@email.arizona.edu

Résumé – Dans cet article, nous nous intéressons à la construction de mémoires fiables sur du *hardware* bruité. Nous considérons l'architecture de mémoire proposée par Taylor et Kuznetsov, dans laquelle un décodeur LDPC est régulièrement appliqué au mot de code stocké dans la mémoire pour tenter de corriger les erreurs introduites par le *hardware*. Nous supposons que l'architecture utilise un décodeur LDPC de type Gallager B bruité. Contrairement aux résultats de [1, 2] qui se limitent à l'analyse des performances du décodeur LDPC bruité seul, nous étudions ici la stabilité de l'ensemble de l'architecture. Pour cela, nous exprimons l'évolution au cours du temps de la probabilité d'erreur dans la mémoire. Cela définit une suite de probabilités d'erreur dont nous analysons les propriétés de croissance et de convergence. Nous utilisons ensuite ces propriétés pour définir un seuil de dégradation qui permet de prédire le niveau maximum de bruit du *hardware* que la mémoire peut tolérer pour rester stable. Nous validons enfin l'analyse théorique avec des résultats de simulation.

Abstract – This paper addresses the problem of constructing reliable memories from unreliable components. We consider the memory construction proposed by Taylor and Kuznetsov in which a codeword stored in a faulty memory is regularly updated by an LDPC decoder to overcome the memory degradation. We assume that the LDPC decoder used in the system is a faulty Gallager B decoder. Compared to [1, 2] which analyze only the faulty decoder, we analyze here the stability of the whole memory construction. We introduce a sequence of errors probabilities at successive time instants and determine the properties and the fixed points of the sequence. From the fixed-point analysis, we define a threshold that predicts the noise level which can be tolerated for the memory to be stable. We finally represent the stability regions of the Taylor-Kuznetsov memory with respect to the decoder noise parameters and validate the theoretical results with Monte-Carlo simulations.

1 Introduction

Ces dernières années, les composants électroniques sont devenus de plus en plus petits, ce qui les a rendus de plus en plus sensibles au bruit. On peut donc supposer que le *hardware* va devenir moins fiable et pourrait introduire des erreurs dans les opérations de calcul et de stockage. En particulier, le bruit du *hardware* pourrait provoquer une dégradation des informations contenues dans les mémoires. Ainsi, après un temps long, les informations stockées risquent d'être très éloignées des données d'origine.

Pour lutter contre cette dégradation, Taylor [3] et Kuznetsov [4] ont proposé de stocker les données en mémoire sous la forme d'un mot de code obtenu à partir d'un code de type Low Density Parity Check (LDPC). À intervalle régulier, le mot de code est extrait de la mémoire et on lui applique un décodeur LDPC pour tenter de corriger les erreurs introduites par le *hardware*. Étant donné que le décodeur LDPC fonctionne sur le même *hardware* que la mémoire, il est lui aussi affecté par le bruit.

Des travaux récents [1, 2] se sont intéressés aux performances de l'architecture de Taylor et Kuznetsov. Ils se sont cependant

limités à l'analyse de l'influence du bruit sur les performances du décodeur LDPC seul, et n'ont pas considéré la mémoire dans son ensemble. En particulier, on voudrait pouvoir prédire si la mémoire est stable, ce qui sera le cas si on peut toujours reconstruire les données originales à partir des informations effectivement contenues dans la mémoire, même après un temps long. Cela implique qu'à chaque instant, le décodeur LDPC est capable de compenser le bruit du *hardware*, et qu'ainsi le niveau d'erreur dans la mémoire n'est jamais trop élevé.

Dans cet article, nous allons nous intéresser à l'architecture de Taylor et Kuznetsov construite à partir d'un décodeur LDPC de type Gallager B, et nous allons proposer une analyse théorique de la stabilité de la mémoire. Pour cela, nous décrirons tout d'abord l'architecture et nous présenterons les modèles d'erreurs qui représentent l'effet du bruit du *hardware* dans la mémoire et dans le décodeur LDPC (voir Partie 2). Ensuite, nous exprimerons l'évolution du niveau d'erreur dans l'information stockée en mémoire (voir Partie 3). Cette évolution correspond à une suite de probabilités d'erreurs exprimées à des instants discrets du temps. Nous étudierons les propriétés de croissance et de convergence de cette suite, et nous utiliserons ces propriétés pour caractériser la stabilité de la mémoire

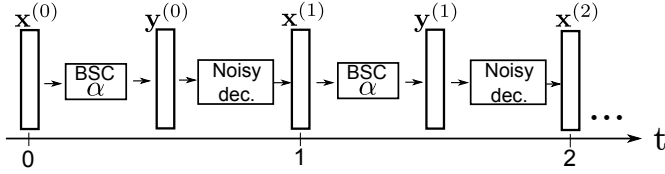


Figure 1: Architecture de la mémoire

(voir Partie 4). Pour finir, nous présenterons des résultats de simulation qui montrent que l'analyse théorique permet bien de prédire la stabilité de la mémoire (voir Partie 5).

2 Architecture de la mémoire

Dans cette partie, nous présentons l'architecture de mémoire proposée par Taylor et Kuznetsov, qui comporte notamment un décodeur LDPC de type Gallager B. Nous décrivons également les modèles d'erreur que nous considérons pour représenter l'effet du bruit du *hardware* dans la mémoire et dans le décodeur.

2.1 Architecture

Dans l'architecture proposée dans [3, 4], les données sont stockées dans la mémoire sous forme d'un mot de code obtenu à partir d'un code LDPC. On considère ici un code LDPC binaire de longueur n représenté par sa matrice de parité H de dimension $n \times m$. Dans le graphe de Tanner du code LDPC, les Noeuds Variable (VN) sont représentés par $v \in \{1, \dots, n\}$, et les Noeuds Check (CN) sont représentés par $c \in \{1, \dots, m\}$. Le degré d'un VN v est noté d_v , et le degré d'un CN c est noté d_c . L'ensemble des CN (respectivement VN) connectés au VN v (respectivement CN c) est représenté par $\mathcal{N}(v)$ (respectivement $\mathcal{N}(c)$).

La structure de la mémoire est décrite sur la Figure 1. On considère des instants discrets $t = 0, \dots, T$. À l'instant initial $t = 0$, le mot de code $\mathbf{x}^{(0)}$, binaire, de longueur n , est écrit dans la mémoire. Le vecteur contenu dans la mémoire à l'instant t est noté $\mathbf{x}^{(t)}$, et $x_v^{(t)}$ représente la v -ième composante de $\mathbf{x}^{(t)}$. Entre deux instants successifs t et $t + 1$, le vecteur $\mathbf{x}^{(t)}$ subit la dégradation de la mémoire, qui est représentée par un canal binaire symétrique (BSC) de paramètre α . On obtient un vecteur dégradé $\mathbf{y}^{(t)}$. Pour tenter de corriger les erreurs introduites par le BSC, on applique alors un décodeur LDPC à $\mathbf{y}^{(t)}$. Le vecteur $\mathbf{x}^{(t+1)}$ obtenu en sortie du décodeur est stocké dans la mémoire à l'instant $t + 1$.

Ici, nous allons supposer que l'architecture utilise un décodeur LDPC de type Gallager B. Ce décodeur, qui travaille avec des messages binaires, présente l'intérêt d'être peu complexe. Étant donné que le décodeur fonctionne sur le même *hardware* que la mémoire, nous supposons qu'il est également affecté par le bruit du *hardware*. Dans la suite, nous allons donc présenter le décodeur Gallager B, ainsi que le modèle d'erreur que nous considérons pour représenter le bruit dans le décodeur.

2.2 Décodeur Gallager B bruité

Le décodeur Gallager B que nous utilisons dans l'architecture fonctionne en L itérations notées $\ell = 1, \dots, L$. Nous décrivons tout d'abord la version non-bruitée de ce décodeur. À l'itération ℓ , le message du CN c au VN v est noté $\gamma_{c \rightarrow v}^{(\ell)}$, et le message du VN v au CN c est noté $\eta_{v \rightarrow c}^{(\ell)}$. Le décodeur utilisé à l'instant t est initialisé avec $\eta_{c \rightarrow v}^{(0)} = y_v^{(t)}$. À l'itération ℓ , on calcule les messages suivants aux CN $c \in \{1, \dots, m\}$:

$$\forall v \in \mathcal{N}(c), \quad \gamma_{c \rightarrow v}^{(\ell)} = \sum_{v' \in \mathcal{N}(c) \setminus v} \eta_{v' \rightarrow c}^{(\ell-1)} \quad (1)$$

L'expression précédente correspond à une somme de XOR. On calcule ensuite les messages aux VN $v \in \{1, \dots, n\}$:

$$\forall c \in \mathcal{N}(v), \quad (2)$$

$$\eta_{v \rightarrow c}^{(\ell)} = \begin{cases} y_v^{(t)} \oplus 1 & \text{si } |\{c' \in \mathcal{N}(v) \setminus c : \gamma_{c' \rightarrow v}^{(\ell)} = y_v^{(t)} \oplus 1\}| > b, \\ y_v^{(t)} & \text{sinon,} \end{cases} \quad (3)$$

où b est un paramètre du décodeur.

Dans la version bruitée du décodeur, on suppose que le *hardware* introduit des erreurs en sortie des fonctions (1) et (2). On note p_b la probabilité d'erreur binaire sur les messages. Les versions bruitées des messages sont représentées par $\tilde{\gamma}_{c \rightarrow v}^{(\ell)}$ et $\tilde{\eta}_{v \rightarrow c}^{(\ell)}$. Elles sont obtenues à partir des messages non-bruités par les relations

$$\tilde{\gamma}_{c \rightarrow v}^{(\ell)} = \gamma_{c \rightarrow v}^{(\ell)} \oplus e_{c \rightarrow v}^{(\ell)}, \quad (4)$$

$$\tilde{\eta}_{v \rightarrow c}^{(\ell)} = \eta_{v \rightarrow c}^{(\ell)} \oplus e_{v \rightarrow c}^{(\ell)}. \quad (5)$$

Dans ces expressions, $e_{c \rightarrow v}^{(\ell)}$ et $e_{v \rightarrow c}^{(\ell)}$ sont des variables aléatoires telles que $P(e_{c \rightarrow v}^{(\ell)} = 1) = p_b$ et $P(e_{v \rightarrow c}^{(\ell)} = 1) = p_b$.

Le rôle du décodeur LDPC est de compenser les erreurs introduites par le *hardware*. À chaque instant t , même grand, on veut en effet pouvoir reconstruire l'information originale $\mathbf{x}^{(0)}$ à partir de $\mathbf{x}^{(t)}$. Pour étudier cette condition, nous exprimons tout d'abord les probabilités d'erreurs dans les $\mathbf{x}^{(t)}$ successifs. Nous analysons ensuite la stabilité de la mémoire à partir des propriétés de la suite des probabilités d'erreur.

3 Probabilités d'erreur dans la mémoire

Dans cette partie, nous exprimons la suite des probabilités d'erreur dans les $\mathbf{x}^{(t)}$ aux instants $t = 0, \dots, T$. Nous caractérisons ensuite ses propriétés de croissance et de convergence.

3.1 Suite de probabilités d'erreur

Pour calculer les probabilités d'erreur dans la mémoire, on suppose que l'on dispose d'une fonction $P_e(\beta)$ qui correspond à la probabilité d'erreur en sortie du Gallager B bruité. La valeur β représente la probabilité d'erreur en entrée du décodeur. La fonction $P_e(\beta)$ peut-être obtenue à partir d'une évolution de densité [5]. L'expression de $P_e(\beta)$ dépend des valeurs de L et

de p_b , mais on les omet volontairement dans la notation dans un souci de simplification.

La proposition suivante exprime les probabilités d'erreurs dans la mémoire aux instants $t = 1, \dots, T$.

Proposition 1. Soit $\beta^{(t)}(\alpha) = P(y_v^{(t)} = 1 | x_v^{(0)} = 0)$, la probabilité d'erreur dans $\mathbf{y}^{(t)}$ à l'instant t . On a

$$\begin{aligned} \beta^{(1)}(\alpha) &= \alpha, \\ \forall t > 1, \beta^{(t)}(\alpha) &= (1 - \alpha)P_e \left(\beta^{(t-1)}(\alpha) \right) \\ &\quad + \alpha \left(1 - P_e \left(\beta^{(t-1)}(\alpha) \right) \right). \end{aligned} \quad (6)$$

L'expression de $\beta^{(t)}(\alpha)$ est donnée par la concaténation de deux effets. Le premier effet correspond à la probabilité d'erreur à l'instant $t - 1$, représentée par $P_e \left(\beta^{(t-1)}(\alpha) \right)$. Le deuxième effet est dû la dégradation de la mémoire donnée par le BSC de paramètre α . On peut remarquer que, étant donné que les modèles d'erreur et les fonctions utilisées dans le décodeur sont symétriques au sens de [6], les probabilités d'erreur dans les $\mathbf{x}^{(t)}$ ne dépendent pas du mot de code $\mathbf{x}^{(0)}$ initialement stocké dans la mémoire.

Pour que la mémoire soit stable, on veut que, à chaque instant t , la probabilité d'erreur $\beta^{(t)}(\alpha)$ soit relativement faible. Pour pouvoir caractériser cette condition, nous étudions maintenant les propriétés de croissance et de convergence de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$.

3.2 Propriétés de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$

Les propriétés de croissance de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$ sont données par la proposition suivante.

Proposition 2. Si la fonction $\beta \rightarrow P_e(\beta)$ est croissante avec β , alors la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$ est croissante avec t .

La condition sur la croissance de la fonction $P_e(\beta)$ est vérifiée pour la plupart des décodeurs LDPC, et en particulier pour le Gallager B bruité.

La proposition précédente montre que la probabilité d'erreur dans la mémoire ne peut qu'augmenter. On espère cependant que la suite de probabilité d'erreur va converger vers un point fixe qui correspondra à un niveau d'erreur pour lequel il sera toujours possible de reconstruire les données initiales $\mathbf{x}^{(0)}$ à partir de $\mathbf{x}^{(t)}$. Les points fixes de la suite peuvent être obtenus grâce à la proposition suivante.

Proposition 3. Les points fixes de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$ sont les valeurs $\bar{\beta}$ solutions de l'équation

$$P_e(\bar{\beta}) = \frac{\bar{\beta} - \alpha}{1 - 2\alpha}. \quad (7)$$

À partir du résultat de la proposition 3, on peut montrer que $\bar{\beta} = 1/2$ est toujours un point fixe de la suite. Il s'agit cependant d'un "mauvais" point fixe, pour lequel on ne pourra évidemment pas reconstruire l'information originale $\mathbf{x}^{(0)}$. On

espère donc que la suite aura d'autres points fixes correspondant à des niveaux d'erreur relativement faibles. Nous définissons maintenant la stabilité de la mémoire à partir de l'analyse des points fixes de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$.

4 Conditions de stabilité

La définition suivante donne les conditions de stabilité pour la mémoire.

Définition 1. On fixe les paramètres α et p_b . On note \mathcal{B} l'ensemble des points fixes de la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$, $\bar{\beta} = 1/2$ étant exclu. On note β^* le seuil du décodeur LDPC Belief Propagation (BP) sans bruit [7]. On dit que la mémoire est stable pour les paramètres α et p_b si les deux conditions suivantes sont vérifiées

1. Stabilité : L'ensemble \mathcal{B} n'est pas vide.
2. Admissibilité : L'ensemble \mathcal{B} est tel que $\max \mathcal{B} \leq \beta^*$.

La condition de stabilité impose qu'il existe au moins un point fixe pour la suite qui soit différent de $1/2$. La condition d'admissibilité garantit que ce point fixe correspond à un niveau d'erreur peu élevé. Ainsi, on assure que quelque soit t , on pourra reconstruire $\mathbf{x}^{(0)}$ en appliquant un décodeur de type BP à $\mathbf{x}^{(t)}$.

L'existence ou non de points fixes différents de $1/2$ pour la suite $\{\beta^{(t)}(\alpha)\}_{t=1}^{+\infty}$ dépend des valeurs de α et de p_b . On peut donc définir un niveau maximum de dégradation α qui garantira la stabilité de la mémoire, comme décrit dans la définition suivante.

Définition 2. Pour une valeur de p_b fixée, on définit le seuil de dégradation $\bar{\alpha}$ par

$$\bar{\alpha} = \arg \max_{\alpha} \{ \text{La mémoire est stable} \}. \quad (8)$$

Avec cette définition, on sait que la mémoire sera stable pour tout $\alpha \leq \bar{\alpha}$. On peut noter que ce seuil est différent du seuil du décodeur LDPC [7, 8], qui est défini pour une seule instance de décodage, et ne peut pas prendre en compte la dynamique de la mémoire.

À partir de la définition du seuil de dégradation, nous traçons maintenant des régions de stabilité $\bar{\alpha}$ en fonction de p_b . La Figure 2 représente ces régions pour une famille de codes réguliers avec $d_v = 3$ et $d_c = 6$, pour $b = 2$ et pour différentes valeurs du nombre d'itérations L du décodeur. Les points fixes $\bar{\beta}$ sont calculés numériquement à partir de l'équation (7), et les seuils de dégradation sont obtenus à partir de la Définition 2. Comme attendu, quand le nombre d'itérations augmente, la taille de la région de stabilité augmente également, au prix d'une augmentation de la complexité du décodeur. On remarque également que quand le bruit dans le décodeur, caractérisé par p_b , est trop important, il devient impossible d'obtenir une mémoire stable.

Dans la partie suivante, nous vérifions la validité de l'analyse de la stabilité à l'aide de résultats de simulation.

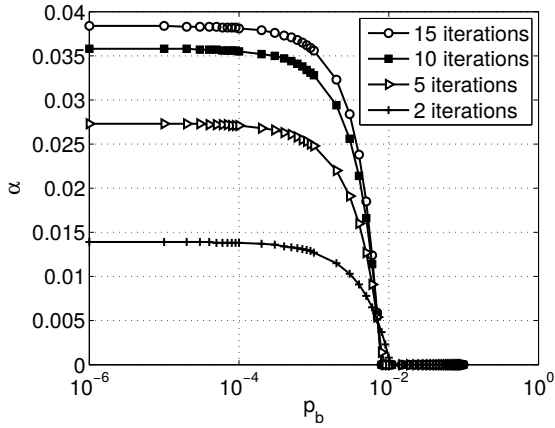


Figure 2: Régions de stabilité pour le Gallager B, famille de codes avec $d_v = 3$, $d_c = 6$

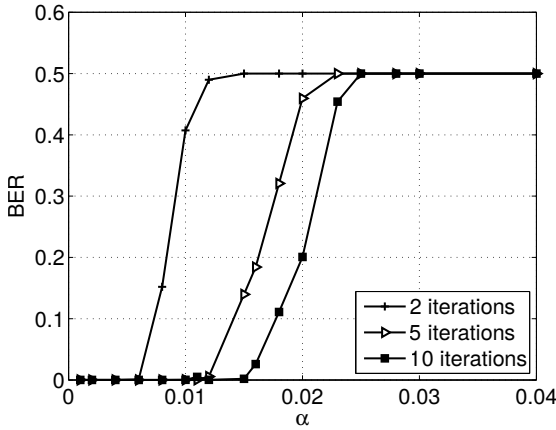


Figure 3: BER en fonction de α pour les mémoires après $T = 100$ instants temporels

5 Résultats de simulation

Nous présentons ici les résultats de simulation pour l'architecture de mémoire construite à partir du décodeur Gallager B. On choisit un code LDPC de rendement $1/2$, de longueur $n = 504$, avec $m = 252$. Il s'agit d'un code régulier avec $d_v = 3$ et $d_c = 6$. On fixe $p_b = 10^{-3}$ et on s'intéresse au taux d'erreur binaire (BER) dans $\mathbf{y}^{(T)}$ après $T = 100$ instants temporels. La figure 3 représente le BER en fonction de α , pour $b = 2$ et pour différentes valeurs de L , le nombre d'itérations du décodeur.

Quand le niveau de dégradation α est faible, le BER est également très faible. Comme attendu, lorsque α devient grand, on observe une brusque augmentation du BER, jusqu'à atteindre la valeur $1/2$. Pour $p_b = 10^{-3}$, l'analyse théorique de la partie 4 prédit des valeurs de seuil de dégradation $\bar{\alpha} = 0.013$ pour $L = 2$, $\bar{\alpha} = 0.025$ pour $L = 5$, $\bar{\alpha} = 0.033$ pour $L = 10$. Dans les simulations, les valeurs de α pour lesquelles on observe l'augmentation brusque du BER sont plus faibles que les valeurs prédites par l'analyse théorique. La différence est probablement due à la longueur relativement faible du code que l'on

utilise ici.

6 Conclusion

Dans cet article, nous avons proposé une analyse de la stabilité de l'architecture de mémoire de Taylor et Kuznetsov construite à partir d'un décodeur LDPC Gallager B. Nous avons notamment fourni des régions de stabilité qui déterminent l'ensemble des paramètres de bruit du hardware pour lesquels la mémoire est stable. Dans nos travaux futurs, nous souhaiterions nous intéresser à des modèles d'erreur plus réalistes, qui pourraient par exemple prendre en compte les dépendances temporelles dans le bruit introduit par le *hardware*. Nous voudrions également pouvoir comparer les décodeurs en prenant en compte leur complexité.

References

- [1] B. Vasic 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 Circuits Systems I, Regular Papers*, 54(11):2438–2446, Nov. 2007.
- [2] S. Brkic, P. Ivanis, and B. Vasic. Analysis of one-step majority logic decoding under correlated data-dependent gate failures. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2599–2603. IEEE, 2014.
- [3] M. Taylor. Reliable information storage in memories designed from unreliable components. *Bell System Technical Journal*, 47:2299–2337, Dec. 1968.
- [4] A. Kuznetsov. Information storage in a memory assembled from unreliable components. *Problems of Information Transmission*, 9:254–264, 1973.
- [5] C-H. Huang, Y. Li, and L. Dolecek. Gallager B LDPC decoder with transient and permanent errors. In *IEEE International Symposium on Information Theory Proceedings*, pages 3010–3014, July 2013.
- [6] L.R. Varshney. Performance of LDPC codes under faulty iterative decoding. *IEEE Transactions on Information Theory*, 57(7):4427–4444, July 2011.
- [7] T. J. Richardson and R. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, Feb. 2001.
- [8] C. Kameni Ngassa, V. Savin, E. Dupraz, and D. Declercq. Density Evolution and Functional Threshold for the Noisy Min-Sum Decoder. *Accepted at IEEE Transactions on Communications*, December 2014.