

ARCHITECTURE D'UN DECODEUR DE VITERBI A DECISION PONDEREE EN SORTIE UTILISANT LA METHODE DE MEMORISATION DE CHEMIN "REA"

E. Angui, P. Adde, S. Faudeil, C. Berrou.

Laboratoire Circuits Intégrés Télécom
Ecole Nationale Supérieure des Télécommunications de Bretagne
BP 832-29285 Brest Cedex (France)

RESUME

Cet article présente l'architecture d'un décodeur de Viterbi utilisant la méthode du "Register Exchange Algorithm" et capable de fournir des décisions pondérées à sa sortie. L'algorithme de pondération mis en oeuvre dans cette architecture, est basé sur celui de Battail ou de Hagenauer-Hoehner et peut être réalisé sur un circuit intégré monolithique de taille environ double de celle d'un décodeur de Viterbi classique. L'architecture proposée est utilisée dans la conception des décodeurs associés aux turbo-codes qui forment une nouvelle classe de codes convolutifs dont les performances, en terme de taux d'erreurs binaires, sont proches de la limite théorique de Shannon.

ABSTRACT

This paper presents the architecture of a Viterbi decoder using the Register Exchange Algorithm which is able to provide soft (weighted) decisions. The weighting algorithm used in this architecture is based on Battail or Hagenauer-Hoehner algorithm and leads to a real-time circuit, the size of which is roughly twice the size of the classical Viterbi decoder. The proposed architecture is implemented in decoders of a new class of convolutional codes called turbo-codes, whose performances in terms of Bit Error Rate (BER) are close to the Shannon limit.

1- Introduction

L'algorithme de Viterbi [1] permet de trouver, dans un treillis, le chemin suivi par un codeur convolutif, selon le critère du maximum de vraisemblance. C'est un algorithme optimal qui délivre, en sortie du treillis, des informations binaires. Cependant, dans bien des cas, le résultat du décodage seul ne suffit pas. Il est utile de lui affecter une mesure de fiabilité afin d'augmenter les performances des dispositifs situés en aval de la liaison numérique. Hagenauer [2] a montré tout l'intérêt (gain de 2 à 4 dB) de travailler en décision souple à tous les niveaux d'une chaîne de communication numérique (concaténation de codes, système GSM, égaliseur, ...etc...). L'exemple de la concaténation série de deux codes est illustré figure 1.

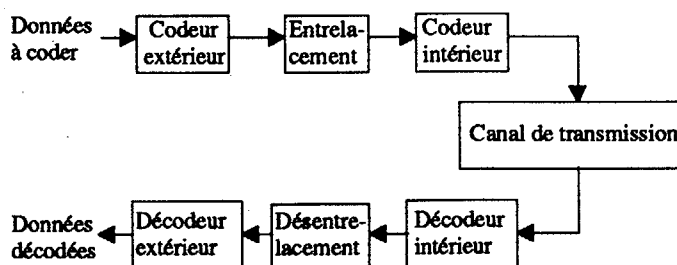


Fig. 1: Concaténation série de deux codes.

Le décodeur intérieur utilise généralement, à son entrée, des informations quantifiées qui sont délivrées par un convertisseur analogique-numérique. Les performances du système global sont améliorées si le décodeur intérieur peut

également fournir des décisions pondérées à l'entrée du décodeur extérieur via le désentrelaceur.

Nous avons proposé un algorithme de pondération [3] [4] dont le principe s'inspire de celui de Battail [5] ou Hagenauer-Hoehner [6], doté d'une architecture basée sur la méthode du "trace-back" facilement intégrable. Pour des codes convolutifs de petites longueurs de contrainte $K=v+1 < 5$, la méthode du "Register Exchange Algorithm" (REA) peut remplacer avantageusement, en terme de compromis vitesse-surface, celle du "trace-back".

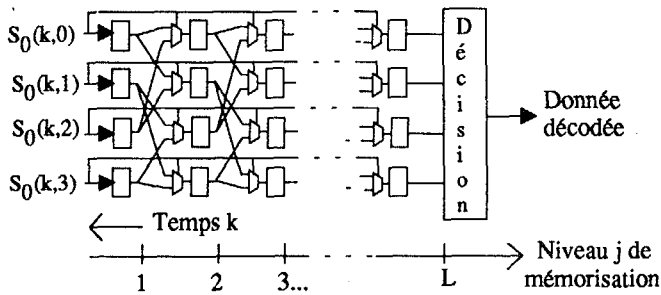
2- Méthode du "Register Exchange Algorithm" (REA)

Pour chaque noeud du treillis, la méthode "REA" permet la mise à jour et la mémorisation de la suite des survivants (bits des décisions partielles) dans un registre à décalage de longueur L , appelée longueur de troncature (L est de l'ordre de 4 ou 5 fois K , pour un code de rendement $R=1/2$). Un exemple d'architecture liée à cette méthode [7] est présenté figure 2. Des multiplexeurs intercalés entre deux registres consécutifs permettent la mise à jour du contenu de chacun des registres, à chaque niveau j de mémorisation. A la fin du treillis, la sortie du registre à décalage, relative à un noeud m considéré, constitue la donnée décodée par ce noeud. Un dispositif de décision (par exemple, le vote majoritaire) placé à la sortie des 2^V registres à décalage permet d'effectuer l'estimation binaire finale de la donnée codée à l'émission.

Le principal inconvénient de la méthode "REA" vient du fait que cette technique réalise physiquement le câblage du treillis, ce qui se caractérise par un réseau d'interconnexion



dont la complexité augmente très rapidement avec le nombre d'états du décodeur. Pour de faibles valeurs de v (de 2 à 4), le nombre de noeuds formant le treillis est faible et ce handicap peut alors être minimisé. On profite alors pleinement de la rapidité de fonctionnement qu'offre cette méthode de mémorisation.



$S_j(k,m)$ = signe de la décision au niveau j pour un noeud m considéré à un instant k .

Fig. 2: Architecture d'un treillis $T(2^v, L)$ à $v=2$ construit selon le principe du "Register Exchange Algorithm" (REA).

3- Mesure de la fiabilité dans l'algorithme de Viterbi

Nous considérons le cas des codes convolutifs binaires pour lesquels le décodeur doit sélectionner, à chaque instant k de calcul, un chemin parmi deux pour chacun des 2^v noeuds du treillis. La mesure de vraisemblance (métrique), ainsi que les décisions partielles (choix du chemin survivant) prises au niveau de chaque noeud, permettent d'affecter une fiabilité à la décision prise en sortie du décodeur.

L'algorithme de Viterbi s'appuie sur la mesure des vraisemblances cumulées en chaque noeud. La différence de ces vraisemblances, aussi appelée mesure de concurrence, peut constituer alors un élément de pondération de premier ordre.

Soient, à l'instant k , $M_S(k,m)$ et $M_C(k,m)$ respectivement les métriques cumulées du chemin survivant et de l'autre chemin, que nous appelons concurrent, arrivant au noeud m . La différence $M_C(k,m) - M_S(k,m) \geq 0$ mesure donc le degré de proximité entre le chemin survivant relatif au noeud m et son chemin concurrent, et renseigne ainsi sur la fiabilité du choix effectué au niveau de ce noeud. En effet, plus cette différence est grande, plus le choix du survivant est fiable. Dans le cas où cette différence est nulle, il est clair qu'il n'y a aucune raison de sélectionner un chemin plutôt que l'autre, ce qui se traduit par un choix arbitraire et donc pouvant être erroné.

Soit $s(k,m)$ le signe (± 1) de la décision partielle, conditionnelle au choix du survivant pour le noeud m à l'instant k . On lui affecte une valeur signée (ou décision pondérée), notée $a(k,m)$, codée sur n bits:

$$a(k,m) = s(k,m) \cdot [M_C(k,m) - M_S(k,m)] = s(k,m) \cdot la(k,m) \quad (1)$$

Soient k l'instant courant, $a_j(k,m)$ et $a'_j(k,m)$ respectivement les valeurs signées du chemin survivant et de

son concurrent à un niveau de mémorisation j quelconque du treillis. Ces décisions sont relatives à l'instant $k-j$:

$$\begin{aligned} a'_j(k,m) &= s'_j(k,m) \cdot |a'_j(k,m)|, \\ a_j(k,m) &= s_j(k,m) \cdot |a_j(k,m)| \quad \text{avec } j=1, \dots, L \end{aligned} \quad (2)$$

La modification de la méthode "REA" permettant de pondérer les décisions en sortie du décodeur consiste, pour chaque noeud m , à:

1/ remplacer la décision (ferme) partielle $s(k,m)$ par la valeur signée $a(k,m)$ dans un registre (de taille n),

2/ effectuer la mise à jour du contenu du registre à décalage (de longueur L) associé au noeud m , pour tout niveau j , par un calcul qui tient compte des valeurs signées, $a_j(k,m)$ et $a'_j(k,m)$, en les traitant comme étant conditionnelles à la décision pondérée, $a(k,m)$, qui vient d'être prise.

A partir d'un calcul de probabilité rigoureux effectué par Battail [4], nous avons simplifié l'expression de la valeur révisée de $a_j(k,m)$, notée $\hat{a}_j(k,m)$, pour aboutir à deux expressions simples qui dépendent des signes $s_j(k,m)$ et $s'_j(k,m)$:

cas 1: Si $s_j(k,m) \cdot s'_j(k,m) < 0$, alors

$$\hat{a}_j(k,m) = s_j(k,m) \cdot \min[|a_j(k,m)|, |a'_j(k,m)|] \quad (3a)$$

cas 2: Si $s_j(k,m) \cdot s'_j(k,m) > 0$, alors

$$\hat{a}_j(k,m) = s_j(k,m) \cdot \min[|a_j(k,m)| + |a'_j(k,m)|] \quad (3b)$$

Fort heureusement, si l'on néglige de traiter le cas 2, la pondération ne perd rien de sa qualité. En effet, lorsque les deux chemins (survivant et concurrent) donnent le même signe de décision (au niveau j), la révision n'a plus une grande importance. Ainsi, seul le cas 1 qui ne nécessite pas de connaître les poids $|a'_j(k,m)|$ sur le chemin concurrent est traité. Toutes ces simplifications conduisent à une procédure de révision proche de celle proposée par Hagenauer-Hoehner.

En plus de l'encombrement d'un circuit classique de décodage selon l'algorithme de Viterbi, cette procédure de révision (3a) requiert:

- $L \cdot 2^v \cdot (n-1)$ bits pour mémoriser les poids $la_j(k,m)$, avec $j=1, \dots, L$ et $m=0, \dots, 2^v-1$;
- $L \cdot 2^v$ comparateurs binaires pour $s_j(k,m)$ et $s'_j(k,m)$;
- $L \cdot 2^v$ comparateurs numériques et multiplexeurs, sur $(n-1)$ bits, pour éventuellement minorer $|a'_j(k,m)|$ par $la(k,m)$.

Cette méthode de pondération réclame donc un grand nombre d'opérations de révisions et aboutit à un décodeur trop encombrant. Nous proposons dans le chapitre suivant une adaptation de l'algorithme de pondération, conduisant à un circuit de faible encombrement.

4- Algorithme de pondération a posteriori et architecture associée

Afin de réduire le coût de la procédure de révision, l'algorithme de pondération a posteriori comporte deux étapes distinctes: tout d'abord une prise de décision ferme (binaire) puis une affectation d'un poids à cette décision.

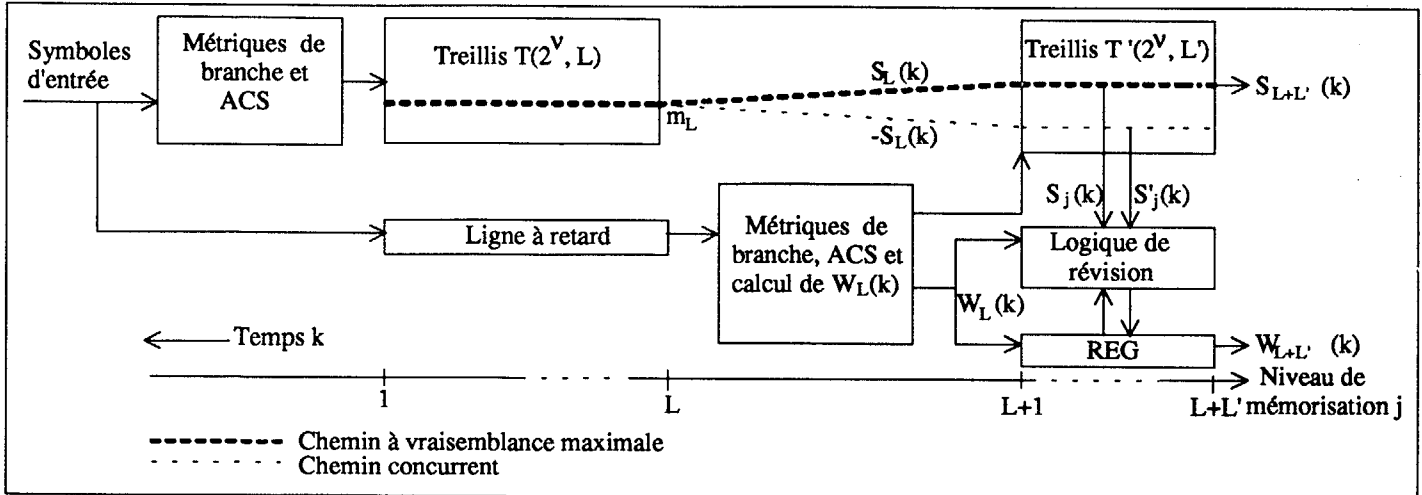


Fig. 3: Architecture d'un décodeur de Viterbi à décision pondérée en sortie.

Une première partie du treillis référencée $T(2^V, L)$ permet, au bout d'une latence L (relative à l'instant $k-L$), de réaliser une estimation binaire, notée $s_L(k)$, de la donnée codée à l'émission, et de lui associer un état décodé, noté m_L . La seconde partie du treillis, référencée $T'(2^V, L')$ permet, le chemin à vraisemblance maximale étant connu et unique, d'effectuer une révision à partir d'un seul noeud. Le nombre total d'opérations intervenant dans la révision est alors divisé par 2^V .

L'architecture du décodeur de Viterbi associée à cette pondération est représentée sur la figure 3. Elle comprend:

- dans la première partie, un décodeur avec ses circuits classiques à savoir: une unité de calcul des métriques de branches, une unité de blocs d'addition-comparaison-sélection (ACS) et un treillis T construit selon la méthode "REA";
- une ligne à retard de longueur L transportant les symboles d'entrée jusqu'au niveau du treillis T' ;
- dans la deuxième partie, les circuits classiques (dupliqués) permettant de reconstruire un treillis T' (selon la méthode "REA") à partir des symboles d'entrée transportés.

Le chemin à vraisemblance maximale, issu de m_L , est poursuivi dans T' , en même temps que son chemin concurrent, aussi issu de m_L , et initialisé par le signe opposé à $s_L(k)$. Le poids $w_L(k)$ est obtenu en sélectionnant uniquement les deux métriques cumulées $M_S(k-L, m_L)$ et $M_C(k-L, m_L)$ du noeud m_L et en prenant la valeur absolue de leur différence:

$$w_L(k) = |M_C(k-L, m_L) - M_S(k-L, m_L)| = |a(k-L, m_L)| \quad (4).$$

Le poids $w_L(k)$, codé sur $(n-1)$ bits, alimente, à chaque instant k ($k > L$), un registre à décalage, référencé REG, de taille $L' \times (n-1)$ et agit de manière concomitante comme un poids de révision selon l'expression (3a).

A l'instant courant k et au niveau j ($L+1 \leq j \leq L+L'$), soit $w_j(k)$ le contenu de REG. L'opération de révision du contenu de REG est effectuée, pour tout niveau j , sous la forme suivante:

si $s_j(k) \neq s'_j(k)$ et si $w_j(k) > w_L(k)$, alors $w_j(k)$ est remplacé par $w_L(k)$ 3).

Cette opération est faite en parallèle pour tous les L' contenus du REG et requiert L' comparateurs et

multiplexeurs. Un ensemble de tels opérateurs à un niveau j de révision est décrit sur la figure 4.

Le résultat final du décodage, relatif à l'instant $k-(L+L')$, est donné sur n bits après donc une latence $L+L'$, par la décision signée $s_{L+L'}(k)$ sur 1 bit, et par le poids $w_{L+L'}(k)$ sur $n-1$ bits.

Il n'est pas nécessaire de donner à L et à L' la même valeur. De très bons résultats sont obtenus lorsque $L'=L/2$ de sorte que le décodeur à sortie pondérée occupe, en tout, moins du double de la taille du décodeur classique. Par ailleurs, du fait de la structure série (et "pipe-line") du décodeur complet, l'algorithme de pondération ne pénalise pas le décodeur de Viterbi, pour ce qui concerne sa rapidité de traitement des données.

Puisque le treillis T a une longueur suffisante, le chemin à vraisemblance maximale dans le treillis T' ne doit plus dépendre, ou si peu, des symboles reçus à l'instant courant. En d'autres termes, la décision ferme (signée) issue de T à l'instant k doit normalement être retrouvée, en sortie de T' , à l'instant $k+L'$. Nous appelons "incohérence" le cas contraire. On peut donc affecter à chaque donnée décodée, dès la sortie de T , un poids provisoire rangé dans un registre. Ce poids est ensuite révisé, au maximum L' fois, tout au long de son parcours dans le registre et constitue, à la sortie du registre, le poids définitif affecté à la décision binaire correspondante. Il est souhaitable, dans le cas d'une incohérence, de réduire ou d'annuler ce poids issu de la révision car celui-ci provient d'une procédure inappropriée.

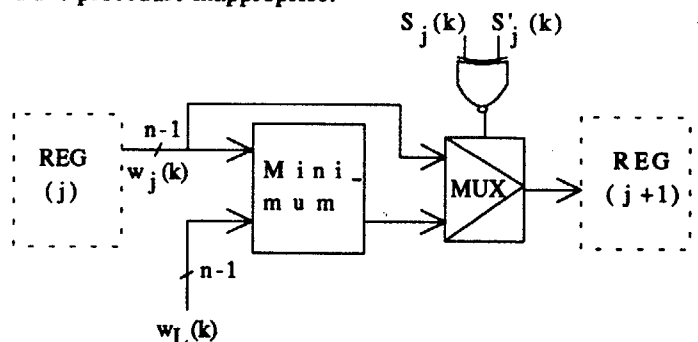


Fig. 4: Logique de révision à un niveau de mémorisation j.

L'algorithme de pondération que nous préconisons est donc basé sur l'observation et le traitement des mesures de concurrence, sur le chemin à vraisemblance maximale. Si ce



dernier n'est pas le bon chemin, le concurrent, dans le treillis T' , est, à un moment ou un autre, le bon chemin et c'est alors que l'opération de révision est particulièrement bénéfique.

Par rapport à la technique du "trace-back", celle du "REA" n'apporte pas de modifications significatives dans la mise en oeuvre de l'algorithme de pondération a posteriori. Toutefois, les signes de décisions partielles subissent une opération permanente de mise à jour dans le treillis "REA" de sorte que ces décisions peuvent être modifiées. De ce fait, la solution envisagée avec l'architecture "trace-back" qui consiste, dès le niveau $j=0$, à calculer les poids $|a(k,m)|$ pour chaque noeud m et à les transporter grâce à une mémoire jusqu'au niveau $j=L$, est trop pénalisante (et n'est donc pas envisageable) lorsque l'on utilise la méthode du "REA". En effet, il faut acheminer les valeurs signées $a(k,m)$ au lieu des poids (un bit supplémentaire étant requis). Le calcul répété des métriques au niveau $j=L$, écarte le besoin de mémorisation et rend disponibles, en même temps que les 2^V survivants pour le treillis T' , les 2^V chemins concurrents dont l'un est nécessaire à l'opération de révision.

5- Conclusion

Dans [4], nous avons examiné la qualité de l'algorithme de pondération a posteriori à travers les résultats du taux d'erreurs binaires dans un schéma de concaténation série de deux codes convolutifs. Les résultats de simulation montrent que l'algorithme est quasi-optimal en ce sens que l'on obtient sensiblement le même taux d'erreurs à la sortie du décodeur extérieur en lui appliquant, sur son entrée, soient les sorties pondérées du décodeur intérieur, soient des symboles qui suivent une loi de distribution gaussienne ayant le même taux d'erreurs binaires que les sorties pondérées.

Les codes convolutifs ont un pouvoir de correction lié à la mémoire de codage v . Plus v est grande, plus fort est le pouvoir de correction si bien que le cas $v=6$ (impliquant tout de même un décodeur de Viterbi à 64 états) est recommandé par le CCSDS (*). Toutefois, des codes à faible valeur v (2 à 4) peuvent être associés dans une concaténation parallèle pour former une nouvelle classe de codes baptisée turbo-codes [8], qui présentent des performances supérieures à tout autre code correcteur existant.

La figure 5 illustre un exemple des performances d'un turbo-décodeur conçu selon un principe de décodage itératif qui consiste en l'association en cascade de plusieurs décodeurs, à décision pondérée en sortie, dotés de l'architecture que nous proposons.

Références bibliographiques

- [1] G. D. Forney, "The Viterbi algorithm," Proc. IEEE, Vol. 61, n°3, pp. 268-278, Mar. 1973.
- [2] J. Hagenauer, "Soft-in/Soft-out, the benefits of using soft values in all stages of digital receivers," Proc. of the 3rd Int. Workshop on Digital Signal Process., ESTEC, Noordwijk, Sept. 1992.
- [3] C. Berrou, P. Adde, "Procédé de décodage d'un code convolutif à maximum de vraisemblance et pondération des décisions, et décodeur correspondant," Dépôt de brevet français n° 91 05279 du 23 Avril 1991.

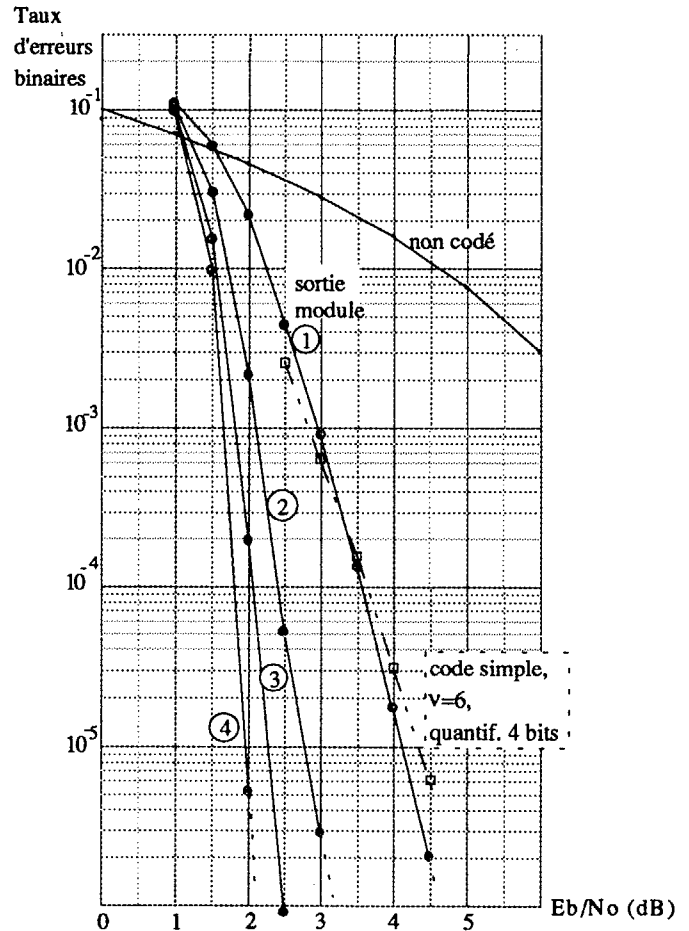


Fig. 5: Taux d'erreurs binaires en sortie d'un turbo-décodeur constitué de 4 modules après transmission sur canal gaussien. Caractéristiques du turbo-code: $R=1/2$ (concaténation parallèle de deux codes convolutifs de rendements 4/7 et 4/5), $v=4$, $L=33$, $L'=25$, quantification sur 4 bits et entrelacement 64×32 .

- [4] C. Berrou, P. Adde, E. Angui, S. Faudeil, "A low complexity soft-output Viterbi decoder architecture," ICC'93, Geneva, pp. 737-740, May 1993.
- [5] G. Battail, "Pondération des symboles décodés par l'algorithme de Viterbi," Annales des Télécoms, N° 1-2, pp. 31-38, Jan.-Fév. 1987.
- [6] J. Hagenauer, P. Hoehner, "A Viterbi algorithm with soft-decision outputs and its applications," IEEE Globecom'89, Dallas, pp. 47.11-47.17, Nov. 1989.
- [7] E. Paaske, S. Pedersen and J. Sparso, "An area-efficient path memory structure for VLSI implementation of high speed Viterbi decoder," Integration, The VLSI journal 12, pp. 78-91, 1991.
- [8] C. Berrou, A. Glavieux, "Near Shannon limit error correcting coding and decoding: Turbo-codes," I.C.C. '93, Geneva, pp. 1064-1070, May 1993.

(*) Consultative Committee for Space Data Systems, recommendation for space data system standards, telemetry channel coding, CCSDS 101.0-B-2 Blue book, Jan. 1987.