

①9 RÉPUBLIQUE FRANÇAISE
INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE
PARIS

①1 N° de publication :

2 972 877

(à n'utiliser que pour les
commandes de reproduction)

②1 N° d'enregistrement national :

11 52106

⑤1 Int Cl⁸ : H 03 M 13/35 (2012.01), H 03 M 13/29, 13/27, 13/23,
H 04 L 1/00

⑫

DEMANDE DE BREVET D'INVENTION

A1

②2 Date de dépôt : 15.03.11.

③0 Priorité :

④3 Date de mise à la disposition du public de la
demande : 21.09.12 Bulletin 12/38.

⑤6 Liste des documents cités dans le rapport de
recherche préliminaire : *Se reporter à la fin du
présent fascicule*

⑥0 Références à d'autres documents nationaux
apparentés :

⑦1 Demandeur(s) : CASSIDIAN SAS Société par actions
simplifiée — FR et TELECOM SUDPARIS — FR.

⑦2 Inventeur(s) : FLOREA ALINA ALEXANDRA,
NGUYEN HANG, MARTINOD LAURENT et MOLKO
CHRISTOPHE.

⑦3 Titulaire(s) : CASSIDIAN SAS Société par actions sim-
plifiée, TELECOM SUDPARIS.

⑦4 Mandataire(s) : CABINET PONTET ALLANO &
ASSOCIES SELARL.

⑤4 PROCÉDE D'ENCODAGE CORRECTEUR D'ERREUR, PROCÉDE DE DECODAGE ET DISPOSITIFS ASSOCIES.

⑤7 La présente invention concerne un procédé d'encoda-
ge correcteur d'erreur (300) pour encoder en parallèle des
données numériques (30) dites sources, se présentant sous
la forme d'une trame (102), lesdites données pouvant être
classées en N classes (102ⁱ), où N est un entier au moins
égal à 2.

Le procédé d'encodage selon l'invention comprend :

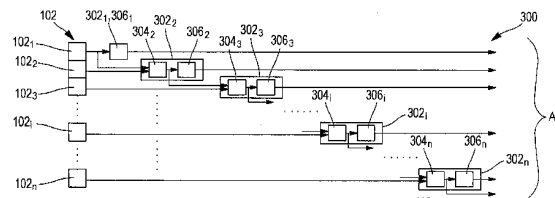
- une première étape d'encodage convolutionnel récursif
et systématique (3061) de données à encoder, formées par
les données de la classe 1 (102₁); et

- une mise en oeuvre des étapes suivantes, pour chaque
n variant de 1 à M, où M est un entier positif inférieur ou égal
à N-1 :

- n^{ème} mélange (304_{n+1}) d'un ensemble formé par les
données de la classe n+1 (102_{n+1}) et les données systéma-
tique d'un encodage précédent;

(n + 1)^{ème} encodage (306_{n+1}) convolutionnel récursif et
systématique de données à encoder, formées par le résultat
du n^{ème} mélange.

L'invention concerne également un procédé de décoda-
ge de données encodées suivant le procédé d'encodage
selon l'invention, ainsi qu'un dispositif d'encodage et un dis-
positif de décodage associés.



FR 2 972 877 - A1



« Procédé d'encodage correcteur d'erreur, procédé de décodage et dispositifs associés»

5

Domaine technique

La présente invention concerne un procédé d'encodage correcteur d'erreur.

Elle concerne également un procédé de décodage adapté à décoder des données ayant été encodées à l'aide du procédé d'encodage correcteur d'erreur selon l'invention.

Elle concerne aussi un dispositif d'encodage pour la mise en œuvre du procédé d'encodage correcteur d'erreur selon l'invention, ainsi qu'un dispositif de décodage pour la mise en œuvre du procédé de décodage selon l'invention.

15

Le domaine de l'invention est celui de l'encodage de données numériques, destinées à être transmises notamment en présence d'un bruit de transmission, et du décodage desdites données numériques après transmission.

L'invention concerne plus particulièrement mais de manière non limitative le domaine de l'optimisation de la transmission de données numériques, par exemple via un réseau radio à large bande.

Etat de la technique antérieure

En télécommunications, on utilise des procédés d'encodage correcteur d'erreur (on peut parler de FEC, pour l'anglais « Forward Error Correction ») pour protéger contre les erreurs qui vont provenir de la transmission, des données, dites sources, à transmettre. Pour cela, on ajoute aux données sources de la redondance afin de permettre au destinataire de détecter et de corriger une partie des erreurs.

30

L'encodage correcteur d'erreur est suivi d'une modulation en vue de la transmission, c'est pourquoi on parle généralement de schéma de codage et de modulation (MCS, pour l'anglais « Modulation and Coding Scheme »), pour désigner tout à la fois l'encodage correcteur d'erreur et la modulation.

On connaît dans l'art antérieur un procédé d'encodage correcteur d'erreur communément appelé « turbo code ». Il s'agit d'un procédé

d'encodage correcteur d'erreur, mettant en œuvre en parallèle au moins deux étapes indépendantes d'encodage convolutif systématique de l'ensemble des données à encoder, et au moins une étape d'entrelacement temporel modifiant l'ordre de prise en compte des données pour chacune des étapes
5 d'encodage. Les turbo codes sont par exemple présentés dans le brevet français FR 2 675 971. Le décodage met en œuvre un algorithme de décodage itératif basé sur l'algorithme de Bahl, Cocke, Jelinek et Raviv, et une recherche de maximum a posteriori.

10 Un inconvénient des turbo codes est cependant que l'ensemble des données sources est protégé de façon égale.

Les codes UEP (pour l'anglais « Unequal Error Protection », c'est-à-dire à protection d'erreur inégale), nés avec la technologie GSM, apportent une réponse à cet inconvénient en permettant de regrouper dans différentes
15 classes, en fonction de leur importance, les données numériques d'une trame, et de protéger chaque classe en fonction de son niveau de priorité (on attribue un niveau de priorité d'autant plus élevé que la donnée est importante).

Ce principe permet d'optimiser des ressources en transmission ainsi que
20 la largeur de bande de fréquence utilisée.

Un inconvénient des codes UEP connu est que chaque classe est traitée séparément. On commence par séparer les différentes classes, avant de les encoder séparément. Les données encodées de chaque classe sont ensuite modulées séparément. Après transmission, les données d'une même trame
25 sont donc décorréliées. Cela implique une perte en terme de ressources car on a par exemple besoin :

- d'en-têtes supplémentaires (ou en anglais « headers », données supplémentaires servant à définir un paquet de données, par exemple les données d'une classe dans le cas où les différentes
30 classes sont traitées de façon indépendante), et
- de traitements supplémentaires pour resynchroniser après transmission les données des différentes classes d'une même trame.

En outre, ces étapes de resynchronisation engendrent des retards en
35 réception.

Une telle perte de ressource va à l'encontre de la demande actuelle en débit de transmission plus élevé, en capacité de réseau plus élevée, et en délai de transmission plus court.

5 Un but de la présente invention est de proposer des procédés et dispositifs d'encodage/de décodage correcteur d'erreur qui ne présentent pas les inconvénients de l'art antérieur.

Un autre but de la présente invention est de proposer des procédés et dispositifs d'encodage/de décodage correcteur d'erreur qui minimisent les
10 délais de transmission et réception, notamment pour des applications telles que la transmission de son ou de vidéo.

Un autre but de la présente invention est de proposer des procédés et dispositifs d'encodage/de décodage correcteur d'erreur qui sont moins gourmands en ressource que les procédés et dispositifs de l'art antérieur.

15 Un autre but de la présente invention est de proposer des procédés et dispositifs d'encodage/de décodage correcteur d'erreur qui nécessitent moins de débits en transmission que les procédés et dispositifs de l'art antérieur.

Enfin, un but de la présente invention est de proposer des procédés et dispositifs d'encodage/de décodage correcteur d'erreur qui nécessitent moins
20 de capacité de réseau que les procédés et dispositifs de l'art antérieur.

Exposé de l'invention

L'invention permet d'atteindre au moins l'un de ces buts par un procédé d'encodage correcteur d'erreur pour encoder en parallèle des données
25 numériques dites sources, se présentant sous la forme d'une trame, lesdites données pouvant être classées en N classes, où N est un entier au moins égal à 2.

Le procédé d'encodage selon l'invention comprend :

– une première étape d'encodage convolutionnel récursif et systématique de
30 données à encoder, formées par les données de la classe 1 ; et
– une mise en œuvre des étapes suivantes, pour chaque n variant de 1 à M, où M est un entier positif inférieur ou égal à N-1 :

– n^{ème} mélange d'un ensemble formé par les données de la classe n+1 et les données systématique d'un encodage
35 précédent ;

- $(n+1)^{\text{ème}}$ encodage convolutionnel récursif et systématique de données à encoder, formées par le résultat du $n^{\text{ème}}$ mélange.

5 On peut aussi prévoir plusieurs étapes d'encodage intermédiaires des mêmes données avant l'ajout de nouvelles informations à encoder.

On parle donc d'un $n^{\text{ème}}$ mélange d'un ensemble formé par les données de la classe $n+1$ et les données systématiques d'une étape d'encodage précédente car

- 10
- l'étape d'encodage précédente peut être l'étape d'encodage n ;
 - l'étape d'encodage précédente peut être une étape d'encodage intermédiaire.

Le terme « donnée systématique », ainsi que le terme « donnée de parité » utilisé dans la suite, sont des termes relatives aux codes convolutionnels récursifs et systématiques, et sont connues de l'homme de l'art. Ils correspondent aux deux sorties d'un code convolutionnel récursif et systématique.

15

Dans tout le texte, les données systématiques et les données de parité peuvent comprendre des bits de terminaison (on parle en anglais de «tail bits»)

20

La donnée systématique est de préférence identique à la donnée à encoder, tandis que la donnée de parité peut correspondre à au moins une donnée de redondance.

Les données systématiques d'une étape d'encodage forment les données à encoder par ladite étape encodage.

25

L'invention prévoit avantageusement d'ajouter avant certaines étapes d'encodage de nouvelles informations à encoder.

On réalise ainsi un procédé d'encodage correcteur d'erreur de type UEP, c'est-à-dire à protection non-uniforme, chaque classe pouvant bénéficier d'une protection différente à l'égard des erreurs apparaissant notamment au cours de la transmission sur un canal.

30

L'invention reprend le principe général des turbo codes, puisqu'on retrouve des étapes d'encodage successives et des étapes de mélange en vue d'un autre encodage des mêmes données. Cependant, on a modifié le schéma

35

connu pour aboutir à un schéma d'encodage dans lequel différentes données numériques sources d'une même trame sont plus ou moins protégées.

La protection différente provient d'un nombre d'informations de redondance (ou données encodées) différent, en fonction du nombre de fois
5 où les données de la classe ont été encodées.

Chaque classe peut en effet être encodée un nombre de fois différent, selon un nombre d'encodages réalisés prenant en compte les données de cette classe. Les données d'une classe peuvent être prises en compte pour un encodage, en tant que données de la classe 1, en tant que données de la
10 classe n+1 et/ou en tant que données à encoder formées par le résultat du n^{ème} mélange.

Le procédé selon l'invention est adapté à traiter des trames entières.

La protection des données peut être appelée hiérarchique, des données plus importantes, autrement dit à plus haut niveau de priorité, pouvant être
15 mieux protégées.

La structure peut être adaptée à tout type de trame, peu importe notamment le nombre de classes.

On réalise un encodage UEP qui s'applique directement à une trame de données numériques entière.

Chaque classe d'une trame peut ainsi être encodée avec un schéma
20 d'encodage dédié et différent du schéma d'encodage appliqué à une ou plusieurs autres classes de la même trame.

Le procédé selon l'invention permet ainsi de réaliser un encodage avec moins de ressources que les procédés de l'état de la technique.

Par ailleurs, le procédé selon l'invention permet de réaliser un encodage
25 plus rapide, en consommant moins d'énergie par rapport aux procédés de l'état de la technique et en réduisant le retard en réception des applications tel que la parole et la vidéo.

Enfin, les données encodées avec le procédé selon l'invention peuvent
30 être transmises avec moins de débits en transmission et moins de capacité de réseau que les données encodées avec les procédés et dispositifs de l'art antérieur, à protection égale.

Le procédé selon l'invention permet que les différentes classes d'une même trame soient encodées par un unique procédé d'encodage correcteur
35 d'erreur, à l'inverse des procédés d'encodage UEP connus dans lesquels

chaque classe est encodée indépendamment des autres classes de la même trame.

Il n'est plus nécessaire de séparer en plusieurs flux de données les données des différentes classes d'une même trame, pour les encoder
5 séparément.

Le procédé selon l'invention permet ainsi d'éviter la transmission d'informations de synchronisation, et donc d'optimiser les ressources et les délais du réseau de transmission.

Le procédé selon l'invention permet ainsi de réduire un retard en
10 réception, en particulier pour des applications telles que la transmission de son (par exemple parole) ou de vidéo.

Le procédé selon l'invention permet ainsi d'éviter une étape de resynchronisation après transmission.

Le procédé selon l'invention permet aussi de simplifier la modulation des
15 données ayant été encodées, toutes les classes d'une trame pouvant être modulées ensemble. Il permet d'appliquer un unique schéma de modulation.

L'au moins une étape de mélange peut apporter une distribution aléatoire des données numériques dans le résultat final.

20 Les données numériques peuvent être toute donnée numérique, notamment des données numériques représentant une vidéo ou une voix.

Le procédé d'encodage est suivi de préférence d'une modulation appropriée au canal de transmission utilisé.

Le schéma d'encodage et de modulation pouvant alors être obtenu est
25 particulièrement robuste aux erreurs.

On peut prévoir que certaines données de la trame ne soient pas encodées.

On peut prévoir de mettre en œuvre un poinçonnage suite à la mise en œuvre d'une étape d'encodage. Cela peut impliquer au moins une étape de
30 dépointonnage lors d'un décodage. Le dépointonnage consiste à retrouver des données de la même taille que des données avant un poinçonnage correspondant, par exemple en introduisant des zéros dans les données poinçonnées.

De préférence, un niveau de priorité est attribué à chacune des classes, les classes 1 à N étant rangées dans l'ordre décroissant des niveaux de priorité.

Le procédé selon l'invention permet ainsi que chaque classe bénéficie
5 d'une protection adaptée. Il permet ainsi éviter la transmission de plus d'informations de redondance qu'il est nécessaire, ce qui permet d'optimiser les ressources du réseau de transmission tout en obtenant une qualité optimale en réception, les informations les plus importantes ayant été hautement protégées.

10

Chaque étape de mélange peut consister en une simple concaténation.

Avantageusement, chaque étape de mélange consiste en un entrelacement.

Un entrelacement peut consister à organiser d'une manière non
15 contiguë des données reçues. On peut envisager tout type d'entrelacement connu, notamment les entrelacements développés dans le cadre des turbo codes.

En général, des erreurs lors d'une transmission sur un canal se produisent en rafales plutôt que de manière indépendante. Si le nombre
20 d'erreurs dépasse la capacité de l'encodage correcteur d'erreur, il ne parvient pas à récupérer les données sources. L'entrelacement est généralement utilisé pour contribuer à résoudre ce problème en modifiant l'ordre de prise en compte de mêmes données numériques dans plusieurs encodages, créant ainsi une distribution plus uniforme des erreurs.

25

Selon un mode de réalisation privilégié du procédé d'encodage selon l'invention, on choisit M égal à N-1, de façon que les classes 1 à N soient encodées.

Ainsi, toutes les données numériques source peuvent être protégées.

30

De préférence, on obtient à l'issue de la mise en œuvre de toute les étapes du procédé d'encodage selon l'invention, c'est-à-dire en sortie, des données de parité correspondant à chacune des étapes d'encodage et une donnée systématique correspondant à la (M+1)^{ème} étape d'encodage.

35

L'invention concerne également un dispositif d'encodage pour la mise en œuvre du procédé d'encodage correcteur d'erreur selon l'invention, apte à encoder des données numériques dites sources se présentant sous la forme d'une trame, lesdites données pouvant être classées en N classes. Ledit

5 dispositif d'encodage selon l'invention comprend :

- un premier module d'encodage convolutionnel récursif et systématique agencé pour encoder des données à encoder formées par les données de la classe 1 ; et
- et soit n variant de 1 à M, où M est un entier positif inférieur ou égal à N-1,

10 M ensembles formés chacun par un n^{ème} mélangeur suivi d'un (n+1)^{ème} module d'encodage convolutionnel récursif et systématique, le n^{ème} mélangeur étant agencé pour recevoir les données de la classe n+1 et les données systématiques d'un module d'encodage précédent, et le (n+1)^{ème} module d'encodage étant agencé pour encoder des données à encoder formées par la

15 sortie du n^{ème} mélangeur.

L'invention concerne également un procédé de décodage de données numériques, agencé pour décoder des données numériques encodées conformément au procédé d'encodage selon l'invention.

20

De préférence, on obtient à l'issue de la mise en œuvre de toute les étapes du procédé d'encodage selon l'invention des données de parité correspondant à chacune des étapes d'encodage, et la donnée systématique de la (M+1)^{ème} étape d'encodage.

25 Ces données obtenues sont avantageusement transmises via un canal de transmission.

On reçoit donc après transmission des données dites reçues pouvant être affectées d'erreurs intervenues en particulier au cours de la transmission.

30 Avantageusement, le procédé de décodage selon l'invention est appliqué à ces données reçues.

Pour des raisons de clarté de l'exposé, dans tout le texte on désigne de la même façon une donnée avant et après transmission.

35 De façon particulièrement avantageuse, le procédé de décodage selon l'invention est tel que, pour tout j, k, l compris entre M+1 et 1:

- à chaque $j^{\text{ème}}$ étape d'encodage du procédé d'encodage selon l'invention, correspond une étape de décodage j (410_j), adaptée à décoder des données encodées résultant de la $j^{\text{ème}}$ étape d'encodage ;
- à l'issue de chaque étape de décodage j , on obtient d'une part des données dites « soft » correspondant à une estimation des données de la classe j , d'autre part des données dites extrinsèques ;
5 et on met en œuvre les étapes suivantes :
 - décodage k ; puis
 - décodage $l \neq k$ en fonction d'au moins une donnée extrinsèque fournie par au
10 moins une étape de décodage autre, utilisée comme donnée a priori.

Les données dites "a priori" représentent de préférence des probabilités sur des données encodées reçues du canal.

- Ces probabilités sont disponibles avant tout décodage courant desdites données encodées reçues, ces valeurs probabilistiques venant d'une source
15 autre que les données encodées reçues du canal.

Une donnée a priori utilisée pour décoder les données encodées résultant de la $k^{\text{ème}}$ étape d'encodage peut être relative aux données de parité et aux données systématiques de cette $k^{\text{ème}}$ étape d'encodage.
20

Les données extrinsèques d'un bit B désignent avantageusement les informations produites par un décodeur (en se basant sur les informations encodées reçues canal et le cas échéant des données a priori), à l'exception des informations canal et a priori du bit B en question.

- Ces données extrinsèques peuvent représenter la probabilité d'avoir reçu ce bit B en fonction des valeurs de tous les autres bits adjacents de la même trame.
25

- On peut se référer notamment à l'ouvrage suivant : Todd K Moon, "Error Correction Coding – Mathematical Methods and Algorithms", John Wiley & Sons 2005.
30

Les données extrinsèques résultant de la $j^{\text{ème}}$ étape d'encodage sont de préférence relatives aux données de parité et aux données systématiques de cette $j^{\text{ème}}$ étape d'encodage.

Les données extrinsèques comprennent de préférence des données dites « a priori » fournissant une donnée supplémentaire pour estimer les données des autres classes.

5 A chaque étape de décodage, on peut utiliser une donnée a priori.

Pour l'étape de décodage effectuée en premier dans l'ordre chronologique, la donnée a priori est mise à zéro. Ensuite, chaque étape de décodage permet d'obtenir une donnée a priori utilisée pour une autre étape de décodage.

10

Chaque classe peut bénéficier d'une protection différente à l'égard des erreurs. Une classe fortement protégée bénéficiera d'un taux d'erreur d'autant moins important lors du décodage.

15 Un décodage $l \neq k$ en fonction d'au moins une donnée extrinsèque fournie par au moins une étape de décodage autre, utilisée comme donnée a priori, permet que les différentes classes encodées bénéficient de l'encodage des autres classes encodées.

20 On peut ainsi atteindre plus rapidement, pour une classe moins protégée, un taux d'erreur binaire donné. L'invention permet ainsi une économie d'énergie, de redondance et de délai.

Chaque étape de décodage peut mettre en œuvre un décodage itératif, c'est-à-dire tout type d'algorithme basé sur la recherche du maximum a posteriori pour l'estimation de probabilités a posteriori (MAP). Ce maximum a posteriori peut être calculée avec l'algorithme BCJR (algorithme de Bahl, Cocke, Jelinek et Raviv), avec une dérivation du MAP, notamment selon un décodage dit LOG MAP utilisant un rapport de vraisemblance (on parle en anglais du "Log Likelihood Probabilities Ratios"), ou un décodage dit MAX LOG MAP, plus appropriée pour l'implémentation matérielle.

30

Avant leur utilisation comme donnée a priori pour une étape de décodage j , on peut effectuer un certain traitement des données extrinsèques. L'idée est de retrouver des données de la même dimension et dans le même ordre que les données en sortie de l'étape d'encodage correspondant.

35

De préférence, les estimations des données de chaque classe sont progressivement extraites des données soft. Il s'agit d'extraire au bon moment certaines données relatives aux classes respectives.

5 Une étape de décodage peut être réalisée non successivement pour toute classe, indépendamment de l'ordre d'encodage.

La première étape d'encodage peut être réalisée pour une classe intermédiaire, et les étapes de décodage précédentes et suivantes peuvent être réalisées dans tout ordre avantageux, notamment selon un taux d'erreur
10 prédéfini à atteindre pour chaque classe.

Avantageusement, une étape de décodage peut être réalisée successivement pour toutes les classes n , où n est décroissant, variant de $M+1$ à 1.

15

De préférence, le procédé de décodage selon l'invention comprend en outre une étape initiale de démultiplexage réalisant la séparation des données de parité de chaque classe.

Les données de parité de chaque classe peuvent alors être utilisées
20 chacune pour une étape de décodage correspondante.

Pour tout j compris entre $M+1$ et 2, le procédé de décodage selon l'invention peut comprendre en outre après chaque étape de décodage j les opérations suivantes :

- 25 - démêlage $j-1$ des données dites extrinsèques, le démêlage $j-1$ réalisant une fonction inverse de celle mise en œuvre à l'étape de mélange $j-1$, pour fournir des données démêlées ;
- démultiplexage des données démêlées pour séparer des données a priori relatives à la classe j dites données extraites et des données a priori
30 relatives aux classes 1 à $j-1$ dites a priori utiles ;
- fourniture des données dites a priori utiles pour être utilisées à l'étape de décodage $j-1$.

Dans tout le texte, lorsqu'on parle démêlage, on se réfère à un mélange donné, le démêlage consistant à retrouver l'ordre des données avant ledit mélange.

5 Pour tout j compris entre $M+1$ et 2 , le procédé de décodage selon l'invention peut comprendre en outre après chaque étape de décodage j les opérations suivantes :

- démêlage $j-1$ des données soft, le démêlage $j-1$ réalisant une fonction inverse de celle mise en œuvre à l'étape de mélange $j-1$, pour fournir
10 des données soft démêlées;

- démultiplexage des données soft démêlées pour séparer des données soft relatives à la classe j dites données soft extraites et des données soft relatives aux classes 1 à $j-1$.

Les données soft extraites servent pour une estimation des données de
15 la classe j .

On peut prévoir en outre une étape spécifique d'estimation des données soft extraites pour retrouver les valeurs de la classe j .

Au moins une étape de décodage j peut être réitérée au moins une fois,
20 en fonction de données a priori correspondant à des données extrinsèques fournies par au moins une étape de décodage des données d'une autre classe.

A chaque réitération d'une étape de décodage, on peut utiliser une donnée a priori.

Chaque étape de décodage peut être réitérée au moins 1 fois, par
25 exemple entre 1 et 5 fois.

L'étape de décodage ainsi réitérée peut ensuite être suivie de nouvelles étapes de décodage de données des classes suivantes ou précédentes.

Avant leur utilisation pour une réitération de l'étape de décodage j , on peut effectuer un certain traitement des données extrinsèques avant d'en
30 utiliser certaines au moins comme données a priori. L'idée est de retrouver des données de la même dimension et dans le même ordre que les données en sortie de l'étape d'encodage correspondant.

On réalise ainsi au moins un rebouclage. Le procédé de décodage selon
35 l'invention peut donc être considéré comme itératif, chaque nouvelle itération

d'une étape de décodage pouvant améliorer l'estimation des données de la classe correspondante.

On peut ainsi utiliser des informations d'autres classes pour améliorer le décodage d'une classe.

5 Chaque classe bénéficie d'une protection différente à l'égard des erreurs. Une classe fortement protégée bénéficiera d'un taux d'erreur d'autant moins important lors du décodage. Lors du décodage, l'au moins un rebouclage permet d'exploiter le fait que lors de l'encodage, des données correspondant à chacune des classes sont mélangées. Les différentes classes encodées
10 peuvent donc bénéficier de l'encodage des autres classes encodées.

On peut ainsi atteindre plus rapidement, pour une classe moins protégée, un taux d'erreur binaire donné. L'invention permet ainsi une économie d'énergie, de redondance binaire et de délai de transmission.

15 Au moins une étape de décodage j peut être réitérée au moins une fois en fonction de données a priori correspondant à des données extrinsèques fournies par au moins une étape de décodage des données d'une autre classe, et pour j compris entre $M+1$ et 2 et t strictement inférieur à j , on réitère l'étape de décodage j en fonction de données a priori obtenues aux étapes de
20 décodage t à $j-1$.

Chaque étape de décodage peut donc être réitérée en utilisant des données a priori obtenues à des étapes de décodage de classes de priorité plus élevée.

25 Dans ce cas, lesdites données a priori peuvent comprendre des informations extrinsèques relatives aux classes 1 à $j-1$ et des informations relatives aux données de parité des classes 1 à $j-1$.

On peut ensuite réitérer les étapes de décodage $j-1$ à t .

30 Au moins une étape de décodage j peut être réitérée au moins une fois en fonction de données a priori correspondant à des données extrinsèques fournies par au moins une étape de décodage des données d'une autre classe, et pour j compris entre M et 1 et t strictement supérieur à j , on réitère l'étape de décodage j en fonction de données a priori obtenues aux étapes de décodage t à $j+1$.

Chaque étape de décodage peut donc être réitérée en utilisant des données a priori obtenues à des étapes de décodage de classes de priorité moins élevée.

Dans ce cas, lesdites données a priori peuvent comprendre des informations extrinsèques relatives aux classes t à $j+1$ et des informations relatives aux données de parité des classes t à $j+1$.

On peut ensuite réitérer les étapes de décodage $j+1$ à t .

De préférence, le procédé de décodage selon l'invention comprend les étapes suivantes :

- on réitère l'étape de décodage $M+1$ en fonction de données a priori obtenues aux étapes de décodage 1 à M ;

- on réitère les étapes de décodage M à 1 en utilisant des données a priori correspondant à des données extrinsèques fournies par l'étape de décodage précédente (selon l'ordre chronologique), de façon que les étapes de décodage $M+1$ à 1 composent une phase de décodage ; et la phase de décodage est réitérée au moins une fois.

Ainsi, on effectue un rebouclage sur l'ensemble des étapes de décodage, et dans chaque phase de décodage les étapes de décodage sont effectuées successivement pour toutes les classes n , où n est décroissant, variant de $M+1$ à 1.

A partir de la deuxième itération, l'étape de décodage $M+1$ peut être réalisée en fonction de données extrinsèques fournies par l'étape de décodage 1.

On peut ainsi utiliser des informations de toutes les autres classes pour améliorer le décodage d'une classe.

L'invention concerne également un dispositif de décodage adapté à la mise en œuvre du procédé de décodage selon l'invention.

Le dispositif de décodage selon l'invention comprend $M+1$ modules de décodage, chaque module de décodage j (où j est un entier compris entre 1 et $M+1$ inclus) étant apte à décoder des données encodées résultant de la $j^{\text{ème}}$ étape d'encodage du procédé d'encodage selon l'invention, et chaque module

de décodage j fournissant des données dites extrinsèques aptes à être utilisées comme données a priori par un autre module de décodage, et au moins une donnée dite « soft » pour une estimation de la classe j.

- 5 L'invention trouve une application dans tous les domaines de transmission de données et tout système de transmission, que ce soit une transmission filaire ou non filaire. Il peut s'agir notamment du domaine :
- des communications radio terrestres,
 - des communications radio aérospatiales,
 - 10 - de la transmission de données en robotique ou en électronique,
 - des applications audio et/ou vidéo.

L'invention concerne également un produit programme d'ordinateur comprenant les instructions pour réaliser les étapes du procédé d'encodage selon l'invention lorsqu'il est exécuté par un appareil informatique.

15

L'invention concerne également un produit programme d'ordinateur comprenant les instructions pour réaliser les étapes du procédé de décodage selon l'invention lorsqu'il est exécuté par un appareil informatique.

20

Description des figures et modes de réalisation

D'autres avantages et particularités de l'invention apparaîtront à la lecture de la description détaillée de mises en œuvre et de modes de réalisation nullement limitatifs, et des dessins annexés suivants :

25

- la figure 1 illustre sous forme de diagramme un exemple dit « en parallèle » de procédé d'encodage selon l'invention,
 - la figure 2 illustre sous forme de diagramme un exemple dit « en parallèle » de procédé de décodage selon l'invention,
 - 30 - la figure 3 illustre un mode de réalisation particulier du procédé d'encodage dit « en parallèle » selon l'invention,
 - la figure 4 illustre un mode de réalisation particulier du procédé de décodage dit « en parallèle » selon l'invention,
 - la figure 5 illustre des courbes de taux d'erreur binaire obtenu avec un
- 35 procédé de décodage selon l'invention.

Dans tout le texte, un multiplexage peut désigner une concaténation, un entrelacement ou toute autre opération réalisée pour ranger des données dans une trame binaire unidimensionnelle ou multidimensionnelle.

5 Dans tout le texte, lorsqu'on parle démultiplexage, on se réfère à un multiplexage donné, le démultiplexage étant l'opération inverse dudit multiplexage.

Dans tout le texte, lorsqu'on parle désentrelacement, on se réfère à un entrelacement donné, le désentrelacement consistant à retrouver l'ordre des
10 données avant ledit entrelacement.

Les moyens pour mettre en œuvre chacune des étapes du procédé selon l'invention sont connus de l'homme du métier, c'est pourquoi on se limitera ici à la description détaillée d'exemples de procédés selon l'invention.

15

La figure 1 est une représentation sous forme de diagramme d'un exemple de procédé d'encodage en parallèle conformément au procédé selon l'invention.

Chaque encodeur met en œuvre un code convolutionnel récursif et
20 systématique.

Dans l'exemple représenté sur la figure 1, une trame de données 102 est encodée. Les données de la trame 102 sont classées dans n classes 102₁-102_n. A chacune des classes 102_i est associé un niveau de priorité. Dans
25 l'exemple présent, de manière non limitative, le niveau de priorité de la classe 102₁ est plus grand que le niveau de priorité de la classe 102₂, et ainsi de suite, la classe de niveau de priorité le plus faible étant la classe 102_n.

Le procédé 300, représenté sur la figure 3, comprend une première étape de codage 302₁ qui se limite à l'encodage 306₁ des données de la classe
30 102₁.

Cette étape de codage 302₁ fournit en sortie des données de parité qui sont des données de redondance pour permettre de retrouver les données d'entrée (et des données dites systématiques qui correspondent aux données d'entrée).

Cette étape 302_1 est suivie d'une deuxième étape de codage 302_2 réalisant un entrelacement 304_2 des données de la classe 102_1 avec les données de la classe 102_2 .

On utilise le terme « suivi », bien que chaque étape de codage i puisse
5 être effectuée simultanément, avant ou après l'encodage 306_{i-1} .

L'étape de codage 302_2 comprend un encodage 306_2 des données entrelacées fournies par l'entrelacement 304_2 .

Cette étape de codage 302_2 fournit en sortie des données de parité qui
10 sont des données de redondance pour permettre de retrouver les données d'entrée (et des données dites systématiques qui correspondent aux données d'entrée).

Le procédé 100 comprend après l'étape 302_2 une étape de codage 302_3 et ainsi de suite jusqu'à l'étape 302_n . Chacune des étapes 302_i pour $i \geq 3$
15 comprend les opérations suivantes :

- un entrelacement 304_i des données systématiques fournies à l'étape 302_{i-1} , avec les données sources de la classe 102_i ; et
- un encodage 306_i des données entrelacées fournies par l'entrelacement 304_i .

20 Chaque étape 302_i pour $i \geq 2$ fournit en sortie les données de parité et des données systématiques qui correspondent ici aux données entrelacées obtenues à l'entrelacement 304_i .

En sortie, la trame encodée A est obtenue par multiplexage des données de parité fournies à chaque étape 302_i , $i=1 \rightarrow n$, et de la donnée
25 systématique fournie par l'étape 302_n .

On remarque que chaque étape d'entrelacement 304_{i+1} , peut aussi être mise en œuvre en même temps que l'étape d'encodage 306_i .

Les données de la trame 102 sont modulées et transmises ensemble
30 sous la forme des données A , car elles n'ont pas été séparées préalablement à la mise en œuvre d'un procédé d'encodage selon l'invention.

Les données A sont de préférences modulées puis transmises sur un canal de transmission.

Après transmission, on reçoit les données A qui peuvent être entachées d'erreurs.

La figure 2 est une représentation sous forme de diagramme d'un
5 exemple de procédé de décodage en série 400 conformément au procédé
selon l'invention, dans le cas où chaque encodeur met en œuvre un code
convolutionnel récursif et systématique.

Dans l'exemple représenté sur la figure 2, des données A sont
décodées. Ces données A ont été encodées conformément au procédé
10 d'encodage en parallèle 300 selon l'invention.

On peut prévoir qu'après chaque encodage, on ait en outre réalisé un
poinçonnage. Dans ce cas, le décodage comprend des étapes de
dépointonnage. Ce cas particulier n'est pas représenté ici, mais il n'est pas
15 exclu du domaine de l'invention.

Le procédé 400 comprend une étape préliminaire de démultiplexage
402 permettant de séparer parmi les données A les données de parité $404_{1 \rightarrow n}$
obtenues respectivement aux étapes d'encodage $306_{1 \rightarrow n}$ du procédé
20 d'encodage en parallèle 300, et une donnée dites systématique 406
correspondant aux données à encoder à la dernière étape d'encodage du
procédé d'encodage en parallèle 300.

Une première étape de décodage 408_n des données de la classe 102_n
comprend les étapes suivantes :

- 25 - un décodage 410_n des données de parité 404_n de la classe 102_n ,
utilisant la donnée systématique 406 et une donnée a priori
(initialement égale à zéro), et fournissant des données dites
extrinsèques et des données dites soft pour une estimation des
données de la classe 102_n ;
- 30 - un désentrelacement 412_n des données dites extrinsèques, pour fournir
des données désentrelacées, le désentrelacement 412_n mettant en
œuvre une fonction d'entrelacement inverse de la fonction
d'entrelacement mise en œuvre à l'entrelacement 304_n du procédé 300
d'encodage en parallèle ;

- un démultiplexage 414_n des données désentrelacées pour séparer des données a priori dites utiles qui vont être utilisées à l'étape de décodage suivante, et des données dites a priori relatives aux données de la classe 102_n.

5 Les données soft pour une estimation des données de la classe 102_n subissent une étape de désentrelacement 416_n mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_n du procédé 300 d'encodage en parallèle. Une étape de démultiplexage 418_n permet d'isoler des probabilités pour les
10 données de la classe 102_n (pour chaque bit, probabilité de valoir 0 ou 1). On peut prévoir en outre une étape d'estimation des données de la classe 102_n.

Le démultiplexage 414_n est suivi d'une nouvelle étape de décodage 408_{n-1} des données de la classe 102_{n-1} comprenant les étapes suivantes :

- un décodage 410_{n-1} des données de parité 404_{n-1} de la classe 102_{n-1},
15 utilisant :
 - les données a priori utiles obtenues à l'étape de décodage précédente, et
 - une donnée canal estimée (données parité et systématique pour les codes convolutionnels récurrents et systématiques, dont le systématique est formé par la donnée systématique de
20 l'étape de décodage précédente ayant subi une étape de désentrelacement 420_{n-1} mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_n du procédé 300 d'encodage en parallèle et en ayant retiré les données systématiques
25 correspondant à la classe 102_n).

Le décodage 410_{n-1} fournit des données dites extrinsèques correspondant aux classes 102_{1->n-1} et des données pour une estimation des données de la
30 classe 102_{n-1}.

L'étape de décodage 408_{n-1} comprend ensuite les étapes suivantes :

- un désentrelacement 412_{n-1} des données dites extrinsèques, pour fournir des données désentrelacées, le désentrelacement 412_{n-1} mettant en œuvre une fonction d'entrelacement inverse de la fonction

d'entrelacement mise en œuvre à l'entrelacement 304_{n-1} du procédé 300 d'encodage en parallèle ;

- un démultiplexage 414_{n-1} des données désentrelacées pour séparer des données a priori dites utiles relatives aux données de la classe 102_{1->n-2} qui vont être utilisées à l'étape de décodage suivante, et des données dites a priori relatives aux données de la classe 102_{n-1}.

Les données soft pour une estimation des données de la classe 102_{n-1} subissent une étape de désentrelacement 416_{n-1} mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_{n-1} du procédé 300 d'encodage en parallèle. Une étape de démultiplexage 418_{n-1} permet d'isoler des probabilités pour les données de la classe 102_{n-1} (pour chaque bit, probabilité de valoir 0 ou 1). On peut prévoir en outre une étape d'estimation des données de la classe 102_{n-1}.

Le procédé 300 comprend après l'étape 408_{n-1} une étape décodage 408_{n-2} et ainsi de suite jusqu'à l'étape 408₂. Chacune des étapes 408_i pour n-2 ≥ i ≥ 2 comprend les opérations suivantes :

- un décodage 410_i des données de parité 404_i de la classe 102_i, utilisant :
 - les données a priori utiles obtenues à l'étape de décodage précédente, et
 - une donnée canal estimée (formée par la parité et le systématique pour les codes convolutionnel récurrents et systématiques, dont le systématique est formé par la donnée systématique de l'étape de décodage précédente ayant subi une étape de désentrelacement 420_i mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_{i+1} du procédé 300 d'encodage en parallèle et en ayant retiré les données systématiques correspondant aux classes 102_{i+1->n}).

Le décodage 410_i fournit des données dites extrinsèques et des données pour une estimation des données de la classe 102_i, dites données soft.

L'étape de décodage 408_i comprend ensuite les étapes suivantes :

- un désentrelacement 412_i des données dites extrinsèques, pour fournir des données désentrelacées, le désentrelacement 412_i mettant en

œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_i du procédé 300 d'encodage en parallèle ;

- un démultiplexage 414_i des données désentrelacées pour séparer des données a priori dites utiles relatives aux données de la classe 102_{1->i-1} qui vont être utilisées à l'étape de décodage suivante, et des données dites a priori relatives aux données de la classe 102_i.

Les données pour une estimation des données de la classe 102_i subissent une étape de désentrelacement 416_i mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304_i du procédé 300 d'encodage en parallèle. Une étape de démultiplexage 418_i permet d'isoler des probabilités pour les données de la classe 102_i (pour chaque bit, probabilité de valoir 0 ou 1). On peut prévoir en outre une étape d'estimation des données de la classe 102_i.

Le procédé 400 comprend après l'étape 408₂ une étape de décodage 408₁ comprenant les étapes suivantes :

- un décodage 410₁ des données de parité 404₁ de la classe 102₁, utilisant :
 - les données a priori utiles obtenues à l'étape de décodage précédente, et
 - une donnée canal estimée (formée par la parité et le systématique pour les codes convolutionnel récurrents et systématiques, dont le systématique est formé par la donnée systématique de l'étape de décodage précédente ayant subi une étape de désentrelacement 420₁ mettant en œuvre une fonction d'entrelacement inverse de la fonction d'entrelacement mise en œuvre à l'entrelacement 304₂ du procédé 300 d'encodage en parallèle et en ayant retiré les données systématiques correspondant aux classes 102_{2->n}).

Le décodage 410₁ fournit des données dites extrinsèques et des données correspondant à une estimation des données de la classe 102₁, dites données soft.

On appelle phase de décodage les étapes décrites, depuis le décodage 410_n.

Le procédé de décodage 400 adapté à l'encodage en parallèle comprend également un rebouclage, qui consiste à utiliser des données extrinsèques fournies à une étape de décodage pour réitérer une étape de décodage
5 précédente de la phase de décodage.

Les données extrinsèques utilisées pour une réitération d'une étape de décodage sont entrelacées pour retrouver des données de la même dimension et dans le même ordre que les données en sortie de l'étape d'encodage correspondant.

10

Le rebouclage comprend les étapes suivantes :

- entrelacement 422_1 des données extrinsèques fournies par l'étape de décodage 410_1 et des données dites a priori relatives aux données de la classe 102_2 , pour obtenir des données entrelacées fournies par
15 l'entrelacement 422_1 , l'entrelacement 422_1 mettant en œuvre une fonction d'entrelacement similaire à la fonction d'entrelacement mise en œuvre à l'entrelacement 304_2 du procédé 300 d'encodage en parallèle ;
- pour i allant de 2 à $n-2$, i étapes d'entrelacement 422_i des données dites a priori relatives aux données de la classe 102_i et des données entrelacées fournies par l'entrelacement 422_{i-1} , l'entrelacement 422_i
20 mettant en œuvre une fonction d'entrelacement similaire à la fonction d'entrelacement mise en œuvre à l'entrelacement 304_{i+1} du procédé 300 d'encodage en parallèle ;
- entrelacement 422_{n-1} des données entrelacées fournies par
25 l'entrelacement 422_{n-2} , avec une donnée de la taille des données a priori relatives aux données de la classe 102_n mais mise à zéro, l'entrelacement 422_{n-1} mettant en œuvre une fonction d'entrelacement similaire à la fonction d'entrelacement mise en œuvre à l'entrelacement 304_n du procédé 300 d'encodage en parallèle ;
- prise en compte des données entrelacées de l'entrelacement 422_{n-1} en tant que donnée a priori lors d'une deuxième itération de l'étape de décodage 410_n .
30

Cette deuxième itération de l'étape de décodage 410_n peut être suivie d'une troisième itération de toutes les autres étapes d'une phase de décodage
35 telle que décrite précédemment.

Un tel rebouclage peut être mis en œuvre plusieurs fois, par exemple trois à quinze fois. On améliore à chaque fois l'estimation des données de chacune des classes. Après au moins cinq rebouclages, il n'est plus toujours intéressant d'effectuer plus de rebouclages, le gain sur la précision de l'estimation étant négligeable en comparaison du temps supplémentaire nécessaire pour une autre itération.

On va ensuite décrire, en référence à la figure 3, un mode de réalisation particulier de procédé d'encodage correcteur d'erreur 300 selon l'invention.

Chaque encodage met en œuvre un code convolutionnel récursif et systématique.

Un tel code permet d'obtenir des données encodées formées par des données dites « de parité » (redondance) et des données dites « systématiques » (identiques aux données à encoder).

Les données numériques dites sources 30 sont formées par une trame 102 comprenant trois classes 102_1 , 102_2 et 102_3 .

Le procédé 300 selon l'invention comprend une étape initiale 70 de séparation des données de chacune des classes 102_1 , 102_2 et 102_3 .

Les données de la classe 102_1 sont désignées par le symbole a_1 .

Les données de la classe 102_2 sont désignées par le symbole a_2 .

Les données de la classe 102_3 sont désignées par le symbole a_3 .

Le procédé 300 selon l'invention comprend une première étape d'encodage 306_1 des données de la classe 102_1 .

On obtient des données de parité P_1 , c'est-à-dire des données de redondance relatives aux données a_1 .

Les données P_1 obtenues sont appelées « parité de la classe 102_1 ».

Le procédé 300 selon l'invention comprend ensuite (ou simultanément) une étape 304_2 d'entrelacement des données a_1 avec les données a_2 de la classe 102_2 .

On obtient des données entrelacées b_1 .

Les données entrelacées b_1 sont ensuite encodées au cours d'une étape d'encodage 306_2 , qui fournit des données de parité P_2 , c'est-à-dire des données de redondance relatives aux données b_1 .

Les données b_1 étant formées par les données a_1 et a_2 mélangées, on
5 augmente le nombre de données de redondance disponibles et correspondant aux données a_1 .

Les données P_2 obtenues sont appelées « parité des classes 102_1 et 102_2 ».

10 Le procédé 300 selon l'invention comprend ensuite (ou simultanément) une étape 304_3 d'entrelacement des données b_1 avec les données a_3 de la classe 102_3 .

On obtient des données entrelacées b_2 .

15 Les données entrelacées b_2 sont ensuite encodées au cours d'une étape d'encodage 306_3 , qui fournit des données de parité P_3 , c'est-à-dire des données de redondance relatives aux données b_2 .

Les données b_2 étant formées par les données a_1 , a_2 et a_3 mélangées, on augmente le nombre de données de redondance disponibles et
20 correspondant aux données a_1 et a_2 .

Les données P_3 obtenues sont appelées « parité des classes 102_1 , 102_2 et 102_3 ».

On obtient en sortie les données A regroupant l'ensemble des parités
25 P_1 , P_2 et P_3 , ainsi qu'une sortie S_3 dite systématique correspondant aux données b_2 à encoder lors de la dernière étape d'encodage 306_3 . La sortie systématique est due à l'utilisation des codes convolutionnels récurrents et systématiques.

30 On va maintenant décrire, en référence à la figure 4, un mode de réalisation particulier du procédé de décodage 400 selon l'invention, correspondant au procédé d'encodage de la figure 3, et dans le cas où chaque encodeur met en œuvre un code convolutionnel récurrent et systématique.

Une première étape de démultiplexage 402 permet de séparer, parmi les données A reçues, les parités P_1 , P_2 , P_3 , et la sortie systématique S_3 .

Le procédé 400 selon l'invention comprend un premier décodage comprenant une étape de décodage 410₃ de la parité P_3 , en fonction de la sortie systématique S_3 et d'une donnée a priori initialement mise à zéro.

On obtient une sortie $L_{\text{soft}}(b_2)$, et des données dites extrinsèques $L_{\text{ext}}(b_2)$.

La sortie $L_{\text{soft}}(b_2)$ permet une estimation des données b_2 .

Dans tout le texte, L_{soft} , L_{ext} et L_{priori} correspondent aux probabilités logarithmiques pour chaque bit des données de valoir 0 ou 1, suite à une utilisation avantageuse pour cette réalisation particulière de l'algorithme de décodage appelé MAX LOG MAP.

D'une part, on met en œuvre les étapes suivantes :

- désentrelacement 416₃ de la sortie $L_{\text{soft}}(b_2)$, le désentrelacement 416₃ mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304₃ ;
- démultiplexage 418₃ pour séparer les données $L_{\text{soft}}(a_3)$ et $L_{\text{soft}}(b_1)$.

La sortie $L_{\text{soft}}(a_3)$ correspond à une estimation des données a_3 de la classe 102₃.

La sortie $L_{\text{soft}}(b_1)$ correspond à une estimation des données b_1 .

En effet, les données b_2 correspondent aux données a_3 entrelacées avec les données b_1 .

Les données dites extrinsèques $L_{\text{ext}}(b_2)$ comprennent notamment des informations relatives à une estimation des données de la classe 102₃.

D'autre part, on met en œuvre les étapes suivantes :

- désentrelacement 412₃ de la sortie $L_{\text{ext}}(b_2)$, le désentrelacement 412₃ mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304₃ ;
- démultiplexage 414₃ pour séparer les données $L_{\text{priori}}(a_3)$ et $L_{\text{priori}}(b_1)$.

Les données $L_{\text{priori}}(a_3)$ correspondent aux probabilités logarithmiques pour chaque bit des données de la classe 102_3 de valoir 0 ou 1.

Les données $L_{\text{priori}}(b_1)$ sont utilisées comme informations à priori à l'étape de décodage suivante.

5

Le démultiplexage 414_3 est suivi d'un deuxième décodage comprenant une étape de décodage 410_2 de la parité P_2 , en fonction de $L_{\text{priori}}(b_1)$ et de la sortie systématique S_3 à laquelle on a appliqué un désentrelacement 420_2 mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304_3 , et un démultiplexage pour séparer les informations systématiques correspondant aux données b_1 et les données de la classe a_3 . Seules les données systématiques b_1 sont utiles pour ce décodage.

On obtient une sortie $L_{\text{soft}}(b_1)$, et des données dites extrinsèques $L_{\text{ext}}(b_1)$.

15

La sortie $L_{\text{soft}}(b_1)$ permet une estimation des données b_1 .

D'une part, on met en œuvre les étapes suivantes :

- désentrelacement 416_2 de la sortie $L_{\text{soft}}(b_1)$, le désentrelacement 416_2 mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304_2 ;
- démultiplexage 418_2 pour séparer les données $L_{\text{soft}}(a_2)$ et $L'_{\text{soft}}(a_1)$.

20

La sortie $L_{\text{soft}}(a_2)$ correspond à une estimation des données a_2 de la classe 102_2 .

25

Les données dites extrinsèques $L_{\text{ext}}(b_1)$ comprennent des informations relatives à une estimation des données des classes 102_1 et 102_2 .

D'autre part, on met en œuvre les étapes suivantes :

- désentrelacement 412_2 de la sortie $L_{\text{ext}}(b_1)$, le désentrelacement 412_2 mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304_2 ;
- démultiplexage 414_2 pour séparer les données $L_{\text{priori}}(a_2)$ et $L_{\text{priori}}(a_1)$.

30

35

Les données $L_{\text{priori}}(a_2)$ correspondent aux probabilités pour chaque bit des données de la classe 102_2 de valoir 0 ou 1.

Les données $L_{\text{priori}}(a_1)$ sont utilisées comme informations à priori à l'étape de décodage suivante.

5

Le démultiplexage 414_2 est suivi d'un troisième décodage comprenant une étape de décodage 410_1 de la parité P_1 , en fonction de $L_{\text{priori}}(a_1)$ et de la sortie systématique S_3 à laquelle on a appliqué un désentrelacement 420_2 puis 420_1 , mettant en œuvre une fonction de désentrelacement inverse de la fonction d'entrelacement mise en œuvre à l'étape d'entrelacement 304_3 respectivement 304_2 , et les démultiplexages appropriés pour obtenir les données systématiques de la classe a_1 .

10

On obtient des données extrinsèques $L_{\text{ext}}(a_1)$ et une estimation des données de la classe 102_1 , $L_{\text{soft}}(a_1)$.

15

Le procédé de décodage 400 présente un rebouclage comprenant les étapes suivantes :

20

- entrelacement 422_1 des données $L_{\text{ext}}(a_1)$ et $L_{\text{priori}}(a_2)$, pour obtenir une donnée entrelacée $L'_{\text{ext}}(b_1)$, et mettant en œuvre une fonction d'entrelacement similaire à la fonction d'entrelacement mise en œuvre à l'entrelacement 304_2 du procédé 300 d'encodage en parallèle ;
- entrelacement 422_2 des données $L'_{\text{ext}}(b_1)$ et $L'_{\text{ext}}(a_3)$, pour obtenir une donnée entrelacée $L_{\text{priori}}(b_2)$, et mettant en œuvre une fonction d'entrelacement similaire à la fonction d'entrelacement mise en œuvre à l'entrelacement 304_3 du procédé 300 d'encodage en parallèle ($L'_{\text{ext}}(a_3)$ étant une donnée de la taille de a_3 mais prenant des valeurs nulles) ;
- nouvelle itération de l'étape de décodage 410_3 , en prenant en compte comme donnée a priori $L_{\text{priori}}(b_2)$;
- nouvelle itération des étapes suivant l'étape de décodage 410_3 .

25

30

Ce rebouclage permet que chaque classe bénéficie de la précision de décodage obtenue pour les autres classes.

Au final, les classes peu protégées peuvent être décodées avec une meilleure précision que si elles avaient été encodées séparément des classes mieux protégées.

35

Les encodages décrits utilisent par exemple des générateurs polynomiaux.

La taille des différentes classes traitées peut varier.

On peut prévoir que certaines classes ne sont pas encodées.

5

On a illustré à la figure 5, des courbes de taux d'erreur binaire pouvant être obtenus avec un procédé de décodage selon l'invention.

Le taux d'erreur binaire est le nombre de bits erronés dans les estimations des données d'une classe encodées, divisé par le nombre total de bits analysés par le procédé de décodage selon l'invention. Il est donc sans unité.

Le taux d'erreur binaire est souvent exprimé en fonction d'un rapport signal à bruit. A la figure 5, l'axe des abscisses correspond à un taux d'erreur binaire, l'axe des ordonnées correspond au rapport E_b/N_0 en dB, c'est-à-dire le rapport en dB d'une énergie par bit sur la densité spectrale de puissance du bruit.

On a pris l'exemple où :

- on a mis en œuvre après l'encodage, une modulation QPSK (pour l'anglais « Quadrature Phase-Shift Keying ») sur un canal AWGN (pour l'anglais « Additive White Gaussian Noise ») ;
- la trame 102 ne comprend que deux classes 102_1 et 102_2 encodées.

A la figure 5, on a un rapport 2/3 entre la taille de la classe 102_2 moins protégée et la taille de la trame, et une taille de trame de 900 bits.

La courbe 11 représente le taux d'erreur binaire associé au décodage de la classe 102_1 , dès la première itération de l'étape de décodage des données de la classe 102_1 .

La courbe 12 représente le taux d'erreur binaire associé au décodage de la classe 102_2 , dès la première itération de l'étape de décodage des données de la classe 102_2 .

La courbe 11' représente le taux d'erreur binaire associé au décodage de la classe 102_1 , à la deuxième itération de l'étape de décodage des données de la classe 102_1 .

La courbe 12' représente le taux d'erreur binaire associé au décodage de la classe 102₂, à la deuxième itération de l'étape de décodage des données de la classe 102₂.

Les courbes sont présentées pour leur allure, et non pour leurs valeurs
5 particulières.

On voit donc que :

- la classe 102₁, qui est la première classe à avoir été encodée, atteint dès la première itération un très bon taux d'erreur binaire, puisqu'on dispose de nombreuses informations de redondance pour retrouver les données de la
10 classe 102₁ ;

- les données de la classe 102₁ encodées à la première étape d'encodage bénéficient d'un gain de décodage semblable à celui qu'on obtient dans un décodage de type « turbo » dès la deuxième itération;

- à la première itération, le taux d'erreur binaire associé aux données de la
15 classe 102₂ est assez faible, car on ne dispose que de peu d'informations de redondance pour retrouver les données de la classe 102₂ ;

- après une itération, le taux d'erreur binaire associé aux données de la classe 102₂ est nettement amélioré, et se rapproche du taux d'erreur binaire obtenu pour le décodage des données de la classe 102₁, bénéficiant
20 notamment du gain de décodage « turbo » .

L'influence d'une classe plus fortement encodée sur une classe moins encodée dépend notamment du rapport entre la taille de la première classe et la taille de la deuxième classe, en nombre de bits.

Après cinq itérations on peut par exemple obtenir un taux d'erreur
25 binaire de 10^{-2} pour un rapport signal à bruit inférieur de 2 dB, avec un gain de 2,5 dB entre la première et la dernière itération.

Cette propriété de l'invention est particulièrement intéressante, car on voit que chaque classe bénéficie de la précision de décodage obtenue pour les autres classes et de l'effet « turbo » .

30 Ainsi, une classe donnée peut être moins protégée que dans l'art antérieur, pour un taux d'erreur binaire donné.

On voit alors qu'on peut transmettre moins de données de redondance que dans l'art antérieur, pour obtenir un taux d'erreur binaire donné.

On augmente ainsi la capacité d'un canal de transmission pour une
35 couverture donnée.

On augmente ainsi la portée d'un canal de transmission pour une capacité donnée.

Bien sûr, l'invention n'est pas limitée aux exemples qui viennent d'être
5 décrits et de nombreux aménagements peuvent être apportés à ces exemples sans sortir du cadre de l'invention.

On peut par exemple envisager tout type de décodage, mettant en œuvre en particulier différentes phases de rebouclage.

On peut par exemple combiner l'invention avec des techniques qui
10 existent déjà, par exemple avec les techniques de poinçonnage, qui consistent à effacer des bits de la trame déjà encodée pour augmenter le ratio de codage. Dans ce cas, on peut réduire la redondance du code pour chaque classe.

On peut également combiner l'invention avec des techniques de l'art
15 antérieur consistant à séparer les données d'une même trame, mais chaque paquet de données regroupant plusieurs classes et pouvant être traité selon l'invention.

REVENDICATIONS

- 5 **1.** Procédé d'encodage correcteur d'erreur (300) pour encoder en parallèle des données numériques (30) dites sources, se présentant sous la forme d'une trame (102), lesdites données pouvant être classées en N classes ($102_1, 102_2, 102_3, 102_i$), où N est un entier au moins égal à 2, caractérisé en ce qu'il comprend :
- 10 - une première étape d'encodage convolutionnel récursif et systématique (306_1) de données à encoder, formées par les données de la classe 1 (102_1); et
- une mise en œuvre des étapes suivantes, pour chaque n variant de 1 à M, où M est un entier positif inférieur ou égal à N-1 :
- n^{ème} mélange (304_{n+1}) d'un ensemble formé par les données de la classe n+1 (102_{n+1}) et les données systématique d'un encodage précédent ;
- 15 - (n+1)^{ème} encodage convolutionnel récursif et systématique (306_{n+1}) de données à encoder, formées par le résultat du n^{ème} mélange.
- 20 **2.** Procédé d'encodage (300) selon la revendication 1, caractérisé en ce qu'un niveau de priorité est attribué à chacune des classes ($102_1, 102_2, 102_3, 102_i$), les classes 1 à N étant rangées dans l'ordre décroissant des niveaux de priorité.
- 25 **3.** Procédé d'encodage (300) selon la revendication 1 ou 2, caractérisé en ce que chaque étape de mélange (304_{n+1}) consiste en un entrelacement.
- 30 **4.** Dispositif d'encodage pour la mise en œuvre du procédé (300) d'encodage correcteur d'erreur en parallèle selon l'une quelconque des revendications précédentes, apte à encoder des données numériques (30) dites sources se présentant sous la forme d'une trame (102), lesdites données pouvant être classées en N classes ($102_1, 102_2, 102_3, 102_i$), et caractérisé en ce qu'il comprend :
- 35 - un premier module d'encodage convolutionnel récursif et

systematique agencé pour encoder des données à encoder formées par les données de classe 1 (102_1) ; et

- et soit n variant de 1 à M , où M est un entier positif inférieur ou égal à $N-1$, M ensembles formés chacun par un $n^{\text{ème}}$ mélangeur suivi d'un $(n+1)^{\text{ème}}$ module d'encodage convolutionnel récursif et systematique, le $n^{\text{ème}}$ mélangeur étant agencé pour recevoir les données de la classe $n+1$ (102_{n+1}) et les données systematiques d'un module d'encodage précédent, et le $(n+1)^{\text{ème}}$ module d'encodage étant agencé pour encoder des données à encoder formées par la sortie du $n^{\text{ème}}$ mélangeur.

5. Procédé de décodage de données numériques (400), caractérisé en ce qu'il est agencé pour décoder des données numériques encodées conformément au procédé selon l'une quelconque des revendications 1 à 3.

6. Procédé de décodage (400) selon la revendication 5, caractérisé en ce que pour tout j, k, l compris entre $M+1$ et 1:

- à chaque $j^{\text{ème}}$ étape d'encodage (306_j) du procédé selon l'une quelconque des revendications 1 à 3, correspond une étape de décodage j (410_j), adaptée à décoder des données encodées résultant de la $j^{\text{ème}}$ étape d'encodage ;

- à l'issue de chaque étape de décodage j , on obtient d'une part des données dites « soft » pour une estimation des données de la classe j , d'autre part des données dites extrinsèques ;

et en ce qu'on met en œuvre les étapes suivantes :

- décodage k ; puis

- décodage $l \neq k$ en fonction d'au moins une donnée extrinsèque fournie par au moins une étape de décodage autre, utilisée comme donnée a priori.

7. Procédé de décodage (400) selon la revendication 6, caractérisé en ce qu'il comprend en outre une étape initiale de démultiplexage (402) réalisant la séparation des données de parité de chaque classe.

- 8.** Procédé de décodage (400) selon la revendication 6 ou 7, caractérisé en ce que pour j compris entre $M+1$ et 2, après chaque étape de décodage j (410_j) le procédé comprend en outre les opérations suivantes :
- 5 - démêlage $j-1$ (412_j) des données dites extrinsèques, le démêlage $j-1$ réalisant une fonction inverse de celle mise en œuvre à l'étape de mélange $j-1$ (304_j), pour fournir des données démêlées ;
- démultiplexage des données démêlées pour séparer des données a priori relatives à la classe j dites données extraites et des données
- 10 a priori relatives aux classes 1 à $j-1$ dites a priori utiles ;
- fourniture des données dites a priori utiles pour être utilisées à l'étape de décodage $j-1$ (410_{j-1}).
- 9.** Procédé de décodage (400) selon l'une quelconque des revendications 6 à 8, caractérisé en ce que pour j compris entre
- 15 $M+1$ et 2, après chaque étape de décodage j (410_j) le procédé comprend en outre les opérations suivantes :
- démêlage $j-1$ (412_j) des données soft, le démêlage $j-1$ réalisant une fonction inverse de celle mise en œuvre à l'étape de mélange $j-1$ (304_j), pour fournir des données soft démêlées;
- 20 - démultiplexage des données soft démêlées pour séparer des données soft relatives à la classe j dites données soft extraites et des données soft relatives aux classes 1 à $j-1$.
- 10.** Procédé de décodage (400) selon l'une quelconque des revendications 6 à 9, caractérisé en ce qu'au moins une étape de décodage j (410_j) est réitérée au moins une fois, en fonction de données a priori correspondant à des données extrinsèques fournies
- 25 par au moins une étape de décodage des données d'une autre classe.
- 30
- 11.** Procédé de décodage (400) selon la revendication 10, caractérisé en ce que pour j compris entre $M+1$ et 2 et t strictement inférieur à j , on réitère l'étape de décodage j (410_j) en fonction de données a

priori obtenues aux étapes de décodage t à $j-1$.

- 5 **12.** Procédé de décodage (400) selon la revendication 10, caractérisé en ce que pour j compris entre M et 1 et t strictement supérieur à j , on réitère l'étape de décodage j (410_j) en fonction de données a priori obtenues aux étapes de décodage t à $j+1$.
- 10 **13.** Procédé de décodage (400) selon la revendication 11, caractérisé en ce qu'une phase de décodage comprend les étapes suivantes :
- 15 - on réitère l'étape de décodage $M+1$ (410_{M+1}) en fonction de données a priori obtenues aux étapes de décodage 1 à M ;
- on réitère les étapes de décodage M à 1 en utilisant des données a priori correspondant à des données extrinsèques fournies par l'étape de décodage précédente, de façon que les
- étapes de décodage $M+1$ à 1 ($410_{M+1 \rightarrow 1}$) composent une phase de décodage ;
- et en ce que la phase de décodage est réitérée au moins une fois.
- 20 **14.** Dispositif de décodage caractérisé en ce qu'il est adapté à la mise en œuvre du procédé de décodage selon l'une quelconque des revendications 5 à 13, ledit dispositif comprenant $M+1$ modules de décodage, chaque module de décodage j étant apte à décoder des données encodées résultant de la $j^{\text{ème}}$ étape d'encodage (306_j) du procédé selon l'une quelconque des revendication 1 à 3, et chaque
- 25 module de décodage j fournissant des données dites extrinsèques aptes à être utilisées comme données a priori par un autre module de décodage, et au moins une donnée dite « soft » pour une estimation de la classe j .
- 30 **15.** Produit programme d'ordinateur comprenant les instructions pour réaliser les étapes du procédé d'encodage selon l'une quelconque des revendications 1 à 3 lorsqu'il est exécuté par un appareil informatique.

- 16.** Produit programme d'ordinateur comprenant les instructions pour réaliser les étapes du procédé de décodage selon l'une quelconque des revendications 5 à 13 lorsqu'il est exécuté par un appareil informatique.

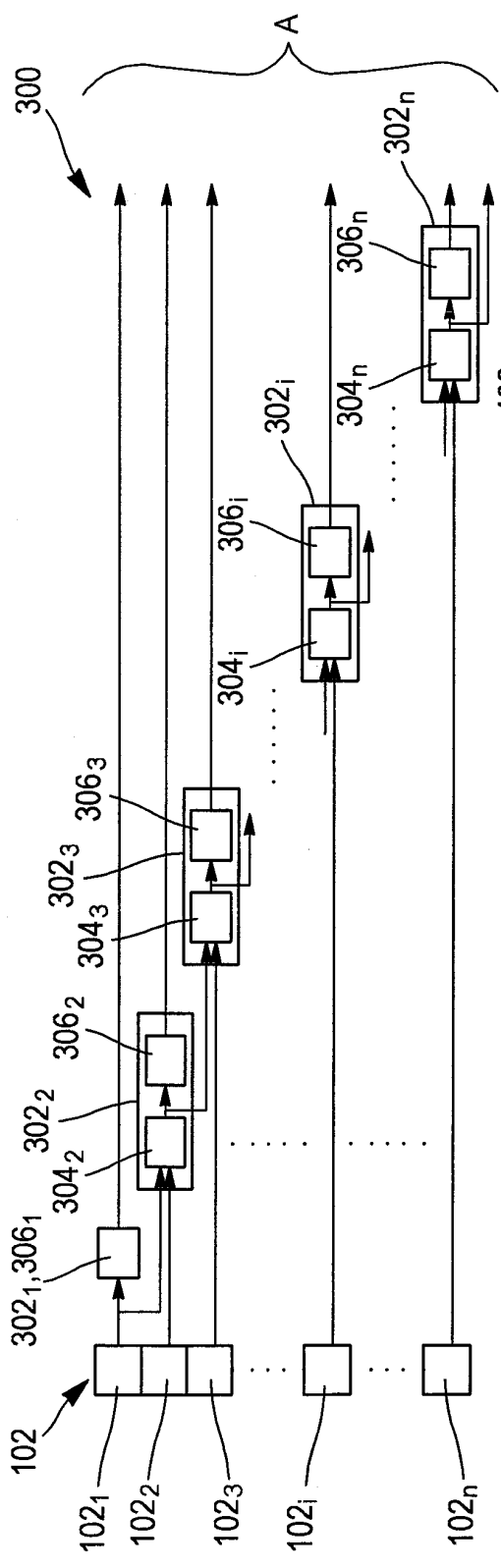


FIG. 1

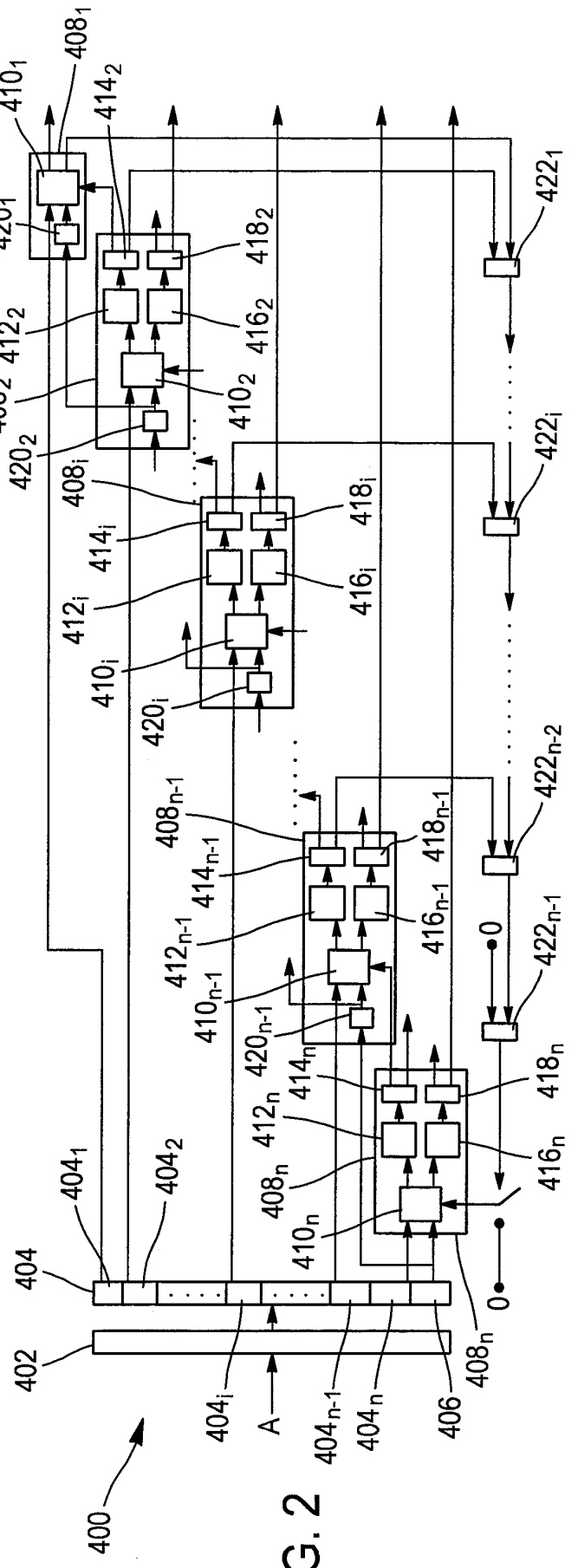


FIG. 2

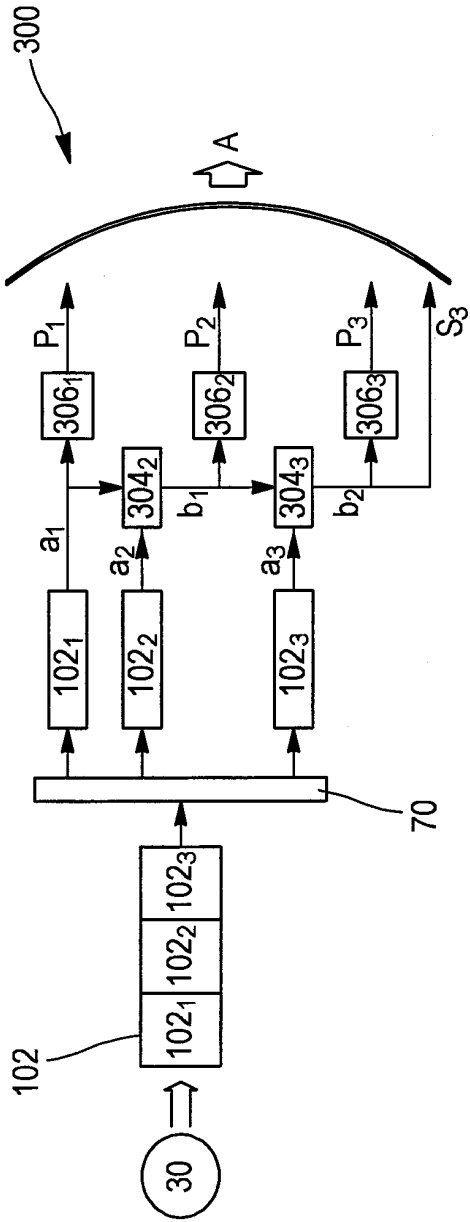


FIG. 3

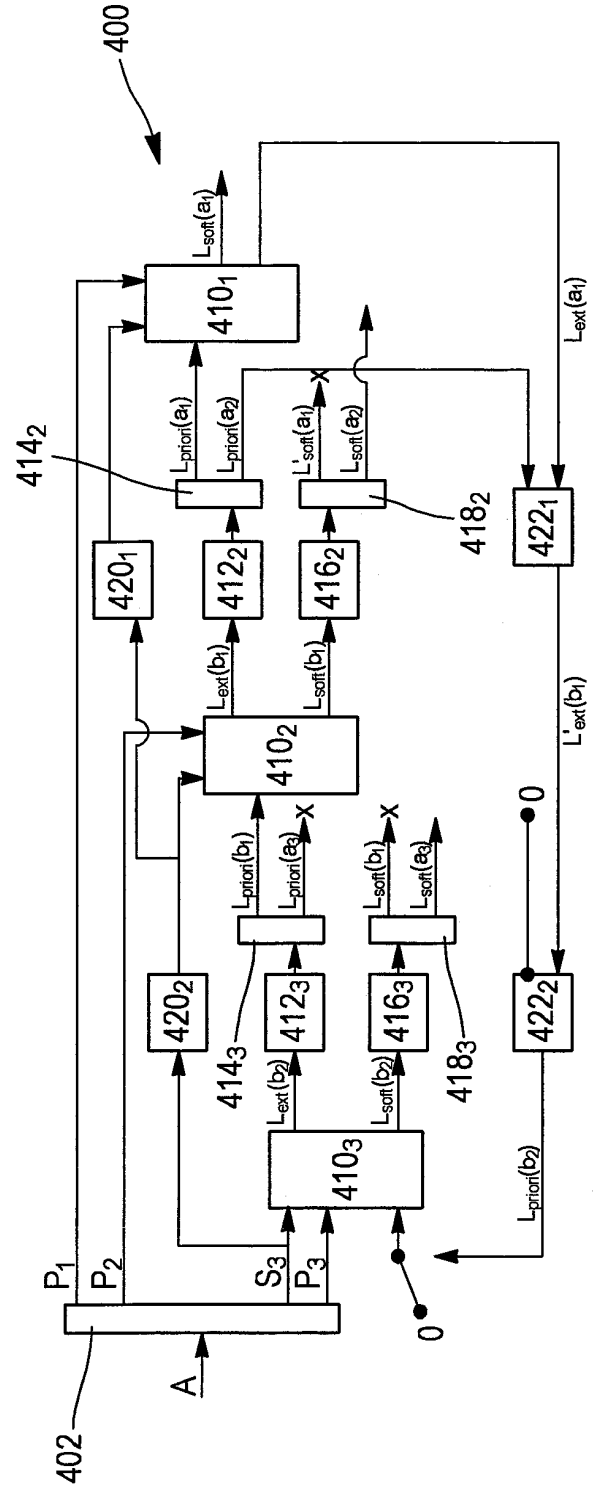


FIG. 4

3 / 3

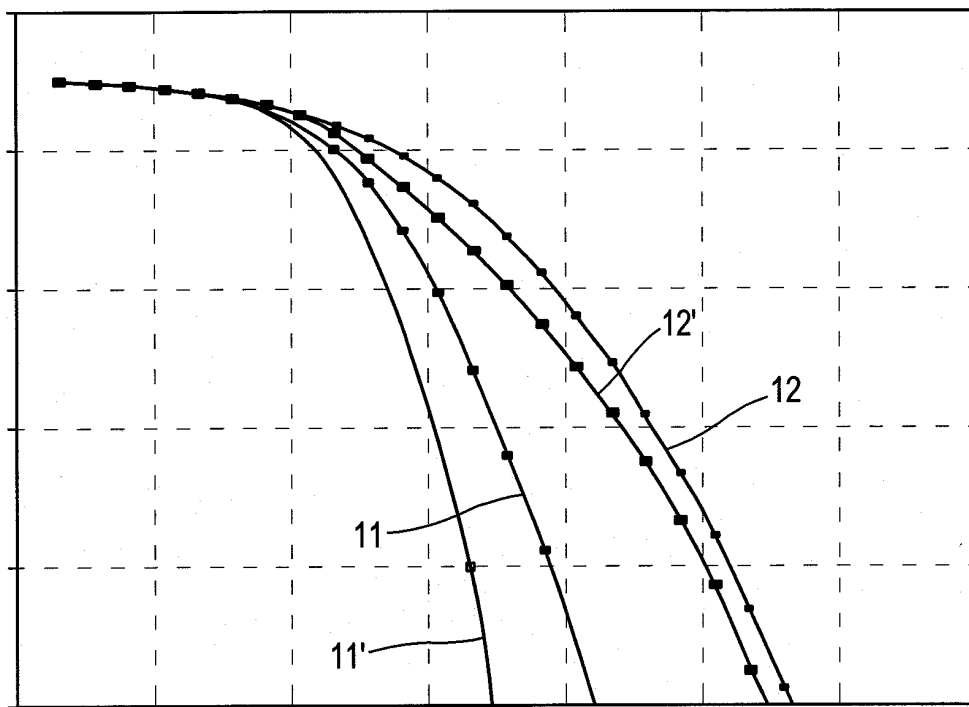


FIG. 5



**RAPPORT DE RECHERCHE
PRÉLIMINAIRE**
établi sur la base des dernières revendications
déposées avant le commencement de la recherche

N° d'enregistrement
national

FA 748347
FR 1152106

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
A	BURKERT F ET AL: "TURBO DECODING WITH UNEQUAL ERROR PROTECTION APPLIED TO GSM SPEECH CODING", GLOBAL TELECOMMUNICATIONS CONFERENCE 1996, GLOBECOM '96; [GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM)],, vol. 3, 18 novembre 1996 (1996-11-18), pages 2044-2048, XP002924961, ISBN: 978-0-7803-3337-6 * le document en entier * -----	1-16	H03M13/35 H03M13/29 H03M13/27 H03M13/23 H04L1/00 DOMAINES TECHNIQUES RECHERCHÉS (IPC) H03M
Date d'achèvement de la recherche		Examineur	
5 août 2011		Rydyger, Kay	
CATÉGORIE DES DOCUMENTS CITÉS X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure. D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	

1
EPO FORM 1503 12.99 (P04C14)