

Procédé de brouillage d'un système de communication par insertion de motifs factices dans un flux de données à émettre

L'invention concerne le domaine du brouillage dans lequel un objectif
5 consiste à neutraliser un système de communication en perturbant le signal émis par un émetteur du système et à destination d'un récepteur du système.

Les systèmes de communication utilisent, lors de la génération du signal à émettre, des séquences particulières, insérées dans le signal, qui sont utilisées pour synchroniser les équipements émetteur et récepteur entre
10 eux. Ces séquences de synchronisation sont fixées une fois pour toutes par le protocole de communication ou le standard de mise en œuvre du système et sont insérées dans le signal à émettre avec les séquences d'informations modulées.

L'invention porte sur un procédé de brouillage d'un système de communication qui vise à transmettre un signal respectant le même standard
15 que celui dudit système mais comportant plusieurs motifs de synchronisation factices insérés dans le but de rendre plus difficile, pour un récepteur, la synchronisation avec un émetteur.

L'invention vise également à transformer un émetteur standard du système de communication à brouiller en brouilleur sans modifier la chaîne
20 d'émission de cet équipement. Le ou les motif(s) de synchronisation factice(s) sont introduits dans le signal émis en générant directement dans la séquence de données binaires à transmettre, une séquence binaire adaptée. L'invention ne nécessite ainsi aucune modification de l'équipement émetteur
25 car elle intervient en amont de la chaîne d'émission.

Le problème technique visé par la présente invention consiste à neutraliser un système de communications à partir d'un ou plusieurs terminaux dudit système transformé(s) en brouilleur(s). En réutilisant un ou
30 plusieurs terminaux du système de communication que l'on souhaite neutraliser, on obtient un ou plusieurs brouilleurs à faibles coûts et qui

peuvent s'adapter à une grande diversité de réseaux de communication standardisés. L'invention vise ainsi à concevoir une solution de brouillage qui ne nécessite pas d'équipements brouilleurs dédiés à cette seule fin.

5 Les systèmes de brouillage connus sont en général basés sur des équipements brouilleurs spécifiques qui présentent les inconvénients suivants.

La consommation d'énergie dans un brouilleur est en général très élevée car la forme d'onde de brouillage utilisée n'est pas optimisée pour
10 neutraliser efficacement un système de communication.

Pour limiter la consommation énergétique et pour garantir l'efficacité de la neutralisation, la forme d'onde de brouillage doit au contraire être aussi cohérente que possible avec les signaux dont on souhaite perturber la réception.

15 Par ailleurs, les formes d'onde de brouillage usuelles connues sont en général soit de combinatoire et de complexité faibles et de ce fait aisément détectables (c'est le cas notamment des formes d'onde dites de brouillage de barrage, qui rendent ces équipements très indiscrets), soit très complexes, lorsque le brouilleur vise, par exemple, à implémenter la forme
20 d'onde exacte du système de télécommunication à neutraliser pour réaliser des effets de leurrage ou de saturation des accès.

En outre, de nombreux systèmes de brouillage connus souffrent le plus souvent d'un manque d'évolutivité, d'adaptabilité ou de souplesse principalement lié au choix des architectures matérielles et logicielles.

25 Enfin, les systèmes de brouillage connus qui présentent une portée significative sont aussi d'un coût, d'un encombrement et d'une consommation énergétique élevés.

L'invention vise à résoudre les problèmes précités et à supprimer les
30 limitations des solutions de l'art antérieur en proposant un procédé de brouillage d'un système de communications à partir d'une modification des

données utiles produites en entrée d'un terminal dudit système. L'invention consiste notamment à réaliser un codage spécifique des données utiles émises par un terminal compatible du système à neutraliser de sorte à générer indirectement dans le signal émis in fine, un ou plusieurs motifs de

5 synchronisation factices aux caractéristiques stationnaires, qui soient aisément interprétables par les récepteurs du système de communication visé, et dont la récurrence et le placement dans la trame émise sont choisis pour optimiser les effets de leurrage et de saturation de la chaîne de réception visée. De cette façon, un récepteur du système à neutraliser ne

10 peut plus se synchroniser correctement. La mise en œuvre du procédé selon l'invention au niveau des données utiles à émettre et non au niveau de la chaîne d'émission permet d'apporter une solution à faible coût qui ne nécessite pas de modifier l'équipement émetteur pour le transformer en brouilleur. En outre, l'utilisation de plusieurs terminaux modifiés permet de

15 constituer un réseau de brouilleurs coopératifs dont les performances de neutralisation sont accrues par rapport à l'utilisation d'un seul terminal modifié. En effet, la faible puissance d'émission d'un terminal est compensée par un maillage de l'espace à couvrir et par l'utilisation de plusieurs terminaux qui émettent des signaux de brouillage simultanés. Lorsque le

20 procédé selon l'invention est appliqué à une pluralité d'émetteurs, l'invention permet de reproduire, avec lesdits émetteurs, la topologie du système de communications à brouiller. L'invention permet également d'augmenter la couverture globale en environnement de propagation défavorable pour le réseau d'émetteurs mettant en œuvre l'invention. L'invention permet

25 également d'accroître les effets de leurrage mais aussi les effets de saturation des accès. Plus généralement, l'invention permet d'induire des effets neutralisant le système de communication à brouiller mais qui sont difficiles à diagnostiquer ou à interpréter, dans la mesure où ces effets reproduisent des cas rencontrés en ingénierie des réseaux de

30 radiocommunication, difficiles ou pathologiques, mais non exceptionnels.

L'invention a pour objet un procédé de brouillage d'un système de communication consistant à générer, dans un signal destiné à être émis par au moins un émetteur compatible dudit système de communication, au moins une séquence factice, ledit procédé étant caractérisé en ce qu'il comprend

5 les étapes suivantes :

- Définir ladite séquence factice et sa position temporelle et/ou fréquentielle au sein du signal à émettre,
- Estimer la valeur et la position des bits factices à insérer au sein de la séquence de données à émettre en entrée de la chaîne d'émission ou d'une sous-partie de la chaîne d'émission de sorte à obtenir, dans la séquence produite en sortie de la chaîne d'émission, la valeur et la position temporelle prédéfinies des symboles de ladite séquence factice,
- Insérer au sein de ladite séquence de données, les bits factices aux positions obtenues.

Selon un aspect particulier de l'invention, lorsque la fonction de transfert F de la chaîne d'émission est inversible, la valeur et la position des bits factices sont estimées en déterminant la fonction de transfert inverse F^{-1} de ladite fonction de transfert F et en appliquant ladite fonction de transfert inverse F^{-1} à ladite séquence factice.

Selon un aspect particulier de l'invention, la fonction de transfert inverse F^{-1} de ladite chaîne d'émission est déterminée en effectuant la composition, dans l'ordre inverse, des fonctions de transfert inverses des différents blocs composant ladite chaîne.

Selon un aspect particulier de l'invention, la chaîne d'émission comporte au moins un code correcteur de rendement k/n pour lequel l'inversion de sa fonction de transfert est réalisée en résolvant le système d'équations

$$[i_0 \quad \dots \quad i_{k-1}] = [c_{p_1} \quad \dots \quad c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \quad \text{où } [c_{p_1} \dots c_{p_d}] \text{ sont les}$$

symboles dont la valeur est fixée dans la séquence codée, $[i_0, \dots, i_{k-1}]$ est la séquence à produire en entrée dudit code, et m_{i,p_j} sont les coefficients de la matrice génératrice dudit code, $p_1, p_2, \dots, p_d$ étant l'ensemble des positions des symboles dont la valeur est fixée dans la séquence codée, avec d un entier
5 égal au plus au nombre n de symboles de la séquence codée.

Selon un aspect particulier de l'invention, ledit code correcteur est un code en bloc linéaire ou un code convolutif ou un turbo-code ou un code à faible densité LDPC

Selon un aspect particulier de l'invention, la chaîne d'émission
10 comporte en outre un brasseur et/ou un entrelaceur et/ou un module de mise en trame et/ou un codeur binaire à signal et/ou un modulateur.

Selon un aspect particulier de l'invention, lorsque la fonction de transfert F de la chaîne d'émission est non surjective, la valeur et la position des bits factices sont estimées en recherchant la séquence d'entrée T' de la chaîne d'émission qui minimise un critère de distance entre la séquence
15 $F(T')(t)$ obtenue en sortie de chaîne d'émission lorsque ladite séquence T' est effectivement produite à son entrée, et ladite séquence factice.

Selon un aspect particulier de l'invention, ledit critère de distance est pris égal à l'intégrale, sur une durée donnée, de la norme au carré de la
20 différence entre la séquence $F(T')(t)$ obtenue en sortie de chaîne d'émission ladite séquence factice.

Selon un aspect particulier de l'invention, la recherche de la séquence d'entrée T' de la chaîne d'émission qui minimise ledit critère de distance est effectué sur un sous-ensemble de l'ensemble des séquences binaires
25 possibles en entrée de la chaîne d'émission.

Selon un aspect particulier de l'invention, ladite séquence de données est produite par un applicatif, par exemple un codeur audio, image ou vidéo.

Dans un mode particulier de réalisation de l'invention, lorsque la fonction de transfert F de la chaîne d'émission est non surjective, les bits
30 factices sont insérés dans la séquence produite en entrée d'une sous-partie

de la chaîne d'émission dont la fonction de transfert est surjective et dont la sortie est commune à la sortie de la chaîne d'émission.

Selon un aspect particulier de cette variante de l'invention, ladite chaîne d'émission comporte un dispositif de chiffrement et ladite sous-partie
5 de la chaîne d'émission exclut ce dispositif.

Selon un aspect particulier de l'invention, les valeurs des symboles de ladite séquence factice sont identiques à celles des symboles d'une séquence de synchronisation que comporte ledit signal pour synchroniser entre eux un récepteur et un émetteur compatibles dudit système de
10 communication.

Selon un aspect particulier de l'invention, la position temporelle et/ou fréquentielle des symboles de ladite séquence factice sont différents de celles des symboles d'une séquence de synchronisation que comporte ledit signal pour synchroniser entre eux un récepteur et un émetteur compatibles
15 dudit système de communication.

L'invention a également pour objet un dispositif d'émission d'un signal comportant une chaîne d'émission pour transformer une séquence de données à émettre en un signal à émettre et un moyen adapté pour mettre en œuvre le procédé selon l'invention.

20 L'invention a également pour objet un système de brouillage coopératif comprenant une pluralité de dispositifs d'émission selon l'invention et pour lesquels les positions temporelles et/ou fréquentielles desdites séquences factices insérées dans le signal émis par chaque dispositif d'émission sont différentes entre elles.

25 L'invention a également pour objet un système de brouillage coopératif comprenant une pluralité de dispositifs d'émission selon l'invention et pour lesquels chacun desdits dispositifs d'émission émet un signal comprenant une pluralité de séquences factices dont les valeurs et/ou positions temporelles et/ou positions fréquentielles sont différentes entre
30 elles.

L'invention a également pour objet un programme d'ordinateur comportant des instructions pour l'exécution du procédé selon l'invention, lorsque le programme est exécuté par un processeur et un support d'enregistrement lisible par un processeur sur lequel est enregistré un
5 programme comportant des instructions pour l'exécution du procédé selon l'invention lorsque le programme est exécuté par un processeur.

D'autres caractéristiques et avantages de la présente invention apparaîtront mieux à la lecture de la description qui suit en relation aux
10 dessins annexés qui représentent :

- La figure 1a, un schéma illustrant la synchronisation entre un récepteur et un émetteur compatibles d'un même système de télécommunication,
- La figure 1b, un schéma illustrant, l'effet obtenu par application du
15 procédé de brouillage selon l'invention,
- La figure 2, un schéma illustrant la transformation opérée par la chaîne d'émission d'un terminal émetteur pour convertir les données utiles à émettre en symboles modulés prêt à être émis par voie radio,
- La figure 3, un schéma illustrant la génération, selon l'invention, de
20 bits factices au sein des données utiles en entrée de la chaîne d'émission d'un terminal émetteur,
- La figure 4, un schéma bloc des différentes fonctions successivement mises en œuvre par un terminal émetteur,
- La figure 5, un schéma des registres à décalage d'un code convolutif de rendement $\frac{1}{2}$,
25
- La figure 6, une représentation, pour l'exemple de code convolutif associé à la figure 5, de la matrice génératrice d'un tel code,
- La figure 7, une illustration de la condition nécessaire et suffisante
30 pour imposer, en sortie du code convolutif défini aux figures 5 et 6, les valeurs d'une séquence de bits consécutifs,

- La figure 8, une illustration des relations de parité d'un code poinçonné.

Dans la suite de la description, l'expression « données utiles », « bits utiles », « information utile » est employée pour désigner les données binaires à transmettre entre l'application exécutée par un émetteur et l'application correspondante exécutée par un récepteur par opposition aux données binaires présentes dans les trames transmises mais qui ne sont pas destinées à l'application exécutée par le récepteur mais sont employées à des fins de signalisation, synchronisation ou toute autre fonction nécessaire au bon fonctionnement du système de communication.

La figure 1a illustre, sur deux schémas, le principe de la synchronisation entre un récepteur et un émetteur compatibles d'un même système de télécommunication.

Sur le haut de la figure 1a est représenté un système de communication sans fils sous la forme d'un émetteur EM qui communique avec un récepteur REC par onde radio. La transformation des données binaires à émettre en signal radio S peut être spécifiée par un standard ou une norme de télécommunications. Cette spécification définit notamment l'insertion, au sein du signal à émettre, de séquences de synchronisation SYNC. De telles séquences sont constituées de symboles connus des équipements du système et positionnées périodiquement ou selon un motif temporel également connu à la fois de l'émetteur EM et du récepteur REC qui implémentent le même standard de télécommunications.

Sur le bas de la figure 1a est représenté, sur un diagramme amplitude/temps, le résultat 10 obtenu en effectuant une corrélation de la séquence de synchronisation connue avec le signal reçu. Le pic d'amplitude du résultat de corrélation donne une estimée de la position temporelle de la séquence de synchronisation dans le signal reçu. En effectuant une telle

opération de corrélation, le récepteur REC peut se synchroniser temporellement avec l'émetteur EM en détectant par exemple le début, le milieu ou la fin d'une trame, indiqué par la présence de ladite séquence, au sein du signal émis. Un but de l'invention est de neutraliser le système de communications EM, REC en perturbant la synchronisation des équipements émetteurs et récepteurs.

La figure 1b illustre, sur deux schémas, le procédé de brouillage selon l'invention.

10 Sur le haut de la figure 1b est représenté un ensemble d'émetteurs EM_B1, EM_B2, ..., EM_BN (avec $N=3$ dans l'exemple illustratif mais non limitatif de la figure 1b, les émetteurs étant notés EM_B1, EM_B2, EM_B3), compatibles du même standard de télécommunications que l'émetteur EM et le récepteur REC que l'on souhaite neutraliser. Ces émetteurs
15 EM_B1, EM_B2, ..., EM_BN (avec toujours $N=3$ dans l'exemple de la figure 1b) sont modifiés, selon l'invention, afin de générer un signal de brouillage S_B à destination du récepteur REC. Ils constituent un système de brouillage coopératif encore appelé réseau de brouilleurs coopératifs. Le signal de brouillage S_B est de même nature que le signal S transmis par un émetteur
20 EM du système de communications à neutraliser à la différence prêt qu'il comporte au moins une séquence de synchronisation factice $SYNC_F$. Sur la figure 1b est représenté le signal de brouillage S_B constitué de la superposition des signaux de brouillage respectivement émis par N émetteurs adaptés EM_B1, EM_B2, ..., EM_BN. Chacun des N signaux de
25 brouillage comporte au moins une séquence de synchronisation factice aux caractéristiques stationnaires $SYNC_{F1}$, $SYNC_{F2}$, ..., $SYNC_{FN}$, placées dans la trame selon des récurrences adaptées au fenêtrage temporel des récepteurs ciblés. Ces séquences sont donc à la fois plausibles pour le récepteur cible et suffisamment fréquentes pour être régulièrement présentes dans les
30 fenêtres de traitement desdits récepteurs. Avantageusement, les séquences factices peuvent comporter les mêmes symboles que la séquence de

synchronisation réelle SYNC mais être transmises à des instants temporels et/ou sur des fréquences différentes, et avec des récurrences trame multiples de celle du motif de synchronisation réel SYNC. Dans une variante de réalisation, chacun des N signaux de brouillage issus des émetteurs
5 EM_B1, EM_B2,..., EM_BN comporte les N séquences de synchronisation factices SYNC_{F1} , SYNC_{F2} ,..., SYNC_{FN} positionnées à des instants temporels différents dans la trame émise. Cette variante présente l'avantage d'augmenter la performance globale du signal de brouillage face à des réseaux de communications exploitant une combinatoire non triviale de
10 séquences de synchronisations SYNC.

Dans une autre variante de réalisation, chacun des N émetteurs de brouillage EM_B1, EM_B2,..., EM_BN émet respectivement un nombre significatif d'exemplaires de chacune des séquences de synchronisation factices (respectivement k_1 séquences SYNC_{F1} , k_2 séquences SYNC_{F2} ,..., k_N
15 séquences SYNC_{FN}) positionnées à des instants temporels différents dans chaque trame émise. Cette variante présente là encore l'avantage d'augmenter la performance globale du signal de brouillage face à des récepteurs cibles dont la synchronisation précise n'est pas connue des brouilleurs EM_B1, EM_B2,..., EM_BN.

20 La transmission de signaux de brouillage comprenant des séquences de synchronisation factices positionnées à des endroits aléatoires dans les trames émises a pour effet de rendre difficile voire impossible la synchronisation du récepteur REC qui ne saura pas discriminer le motif de synchronisation réel des motifs de synchronisation factices.

25 Cet effet est illustré sur le diagramme représenté au bas de la figure 1b qui donne le résultat de l'opération de corrélation, effectuée par le récepteur REC, en présence des signaux de brouillage émis. Le pic d'amplitude du résultat de corrélation 10 correspondant à la séquence de synchronisation SYNC réelle est fortement pollué par une pluralité de pics
30 d'amplitudes de résultats de corrélation 11,12,13,14 correspondant à la présence de séquences de synchronisation factices.

Un des objectifs de l'invention est de permettre l'insertion de motifs de synchronisation factices dans le signal émis par un terminal du système de communications à brouiller sans modifier les éléments de la chaîne d'émission de l'émetteur mais au contraire en agissant uniquement sur les données binaires utiles en entrée de la chaîne d'émission.

La figure 2 illustre schématiquement la transformation subie par une séquence de données binaires utiles D_U à transmettre pour obtenir une séquence de symboles S_T modulés, prêts à être émis par voie d'un signal radio. La transformation exécutée correspond à la fonction de transfert F de la chaîne de transmission de l'émetteur. La séquence de symboles S_T modulés est constituée d'une part de symboles utiles S_U issus de la transformation des données binaires utiles D_U et d'autre part d'au moins une séquence de synchronisation SYNC ou d'une séquence équivalente composée de symboles connus de tous les équipements du système de communication.

La figure 3 illustre la mise en œuvre du procédé selon l'invention, sur un terminal compatible du système de communication à brouiller ou sur tout autre type d'émetteur apte à produire des signaux compatibles du système de communication à brouiller.

Dans un premier temps, les séquences de synchronisation factices $SYNC_F$ générées sont positionnées dans une trame factice vide T_F de même taille qu'une séquence de symboles S_T modulés réelle, aux positions temporelles choisies, différentes de la position temporelle d'une séquence de synchronisation réelle.

Dans un second temps, on estime la valeur et la position des bits factices B_F à insérer au sein de la séquence de données à émettre en entrée de la chaîne d'émission de sorte à obtenir, en sortie de la chaîne d'émission, la valeur et la position temporelle prédéfinies des symboles desdites

séquences factices SYNC_F . Cette opération peut être réalisée en calculant la fonction de transfert inverse F^{-1} de la fonction de transfert F mise en œuvre par la chaîne de transmission puis en appliquant la fonction de transfert inverse F^{-1} à la trame factice T_F pour obtenir une trame modulée D_F 5 comprenant les bits factices B_F .

Les bits factices B_F sont ensuite insérés dans la séquence de données réelle à transmettre D_U en entrée de la chaîne de transmission en poinçonnant les positions relatives des bits factices de la trame factice D_F dans une séquence de données réelle. La séquence de symboles modulés 10 obtenus en sortie de la chaîne de transmission comprend à la fois les symboles de données utiles S_U , le motif de synchronisation réelle SYNC et les motifs de synchronisations factices SYNC_F .

Les bits factices peuvent être introduits directement dans le train de données utiles à transmettre et ce sans modifier ni effectuer d'intrusion dans 15 la chaîne de transmission de l'équipement émetteur.

Il est possible par exemple d'exploiter des modes de transmission de données au contenu utile complètement programmable (par exemple des services de messagerie) et de construire des messages complètement déterministes contenant les bits factices produisant en sortie les motifs 20 SYNC_F recherchés.

Il est aussi possible d'exploiter des modes de transmission correspondant à des applications, d'intervenir par exemple sur un flux utile provenant d'un codeur source audio ou vidéo. Pour cela, on intercepte par exemple les données binaires applicatives avant leur entrée dans la chaîne 25 de transmission au niveau de la couche physique d'un modem. Cette interception peut aussi se faire au niveau d'une couche intermédiaire, par exemple au niveau de la couche réseau. Dans les deux cas on doit tenir compte, selon les procédures décrites plus loin des bits d'informations utiles voisins qui sont aléatoires et variables.

La figure 4 représente un schéma bloc des différentes fonctions successivement mises en œuvre par un émetteur d'un système de communication pour émettre un signal contenant des données à transmettre. Les principales fonctions traditionnellement mises en œuvre sont représentées, étant entendu que le schéma de la figure 4 est donné à titre illustratif et non limitatif. En particulier, certaines fonctions peuvent être omises et l'ordre de certaines fonctions peut être modifié. La fonction de transfert F de la chaîne d'émission est égale à la composition des fonctions de transfert de chaque bloc fonctionnel indépendant de la chaîne, étant entendu que les blocs sont connectés en série. La fonction de transfert inverse F^{-1} est, lorsqu'elle existe, égale à la composition, dans l'ordre inverse, des fonctions de transfert inverses de chaque bloc. Autrement dit, si $f_1, f_2, \dots, f_N$ sont les fonctions de transfert de chaque bloc fonctionnel de la chaîne, alors la fonction de transfert F globale est égale à $F = f_1 \circ f_2 \circ \dots \circ f_N$ et la fonction de transfert inverse F^{-1} est égale à $F^{-1} = f_N^{-1} \circ f_{N-1}^{-1} \circ \dots \circ f_1^{-1}$.

Pour estimer la fonction de transfert inverse F^{-1} globale, il convient donc de déterminer la fonction de transfert inverse de chaque bloc unitaire.

La fonction de transfert F directe de la chaîne d'émission peut être connue lorsque l'invention est mise en œuvre par le concepteur du système de communications ou lorsque ledit système respecte un standard connu. Elle peut également être estimée en testant l'équipement émetteur, par exemple en injectant des signaux de test à son entrée et en analysant les signaux obtenus en sortie.

Les transformations appliquées dans la chaîne d'émission sur le train binaire sont généralement réversibles, c'est à dire qu'il est possible à partir du train binaire en sortie de retrouver le train binaire d'entrée.

Toutefois certaines fonctions mises en œuvre par la chaîne d'émission d'un système de communications peuvent ne pas être toujours surjectives. En d'autres termes, il peut arriver qu'un train binaire codé TBC, dont on voudrait forcer les valeurs, en sortie d'un module de la chaîne d'émission ne corresponde à aucune série de bits utiles TBU en entrée dudit module. Par

exemple, les opérations de codage canal ou de mise en trame transforment un train binaire utile de longueur L_u en un train binaire codé de longueur L_c . En raison des opérations de mise en trame, nécessaires pour assurer la synchronisation du récepteur, et de codage correcteur d'erreur, nécessaires
5 pour compenser l'effet du canal de propagation, on a en pratique toujours $L_c > L_u$. Cela signifie que parmi les 2^{L_c} séquences de L_c bits codés, seules 2^{L_u} séquences peuvent être obtenues par codage. Le codage n'est donc jamais surjectif.

Dans un tel cas, il n'est pas possible de déterminer la fonction de
10 transfert inverse F^{-1} de la chaîne d'émission globale, mais seulement l'inverse F^{-1} de la chaîne d'émission sur l'image restreinte $F(\{TBU\})$ de l'ensemble $\{TBU\}$ des trains de bits utiles en entrée de la chaîne d'émission.

On cherche donc à déterminer dans quelle mesure il est possible de forcer la valeur de certains des bits du signal codé. En particulier, on cherche
15 à déterminer le nombre de bits dont la valeur peut être imposée et dans quelle mesure il est possible de choisir non seulement la valeur mais aussi la position de ces bits. Dans le cas d'un codeur canal, on cherche à imposer la valeur d'une série de bits codés consécutifs de manière à obtenir des motifs codés qui s'apparentent à des motifs de synchronisation.

20 La mise en œuvre pratique consiste à analyser successivement les différentes transformations du train binaire en commençant par la transformation intervenant en dernier dans la chaîne d'émission. Pour chaque transformation, on détermine les entrées qui doivent être appliquées pour obtenir en sortie le train binaire codé voulu.

25

Dans cette optique, les transformations élémentaires du train binaire et leur inversibilité sont analysées au cas par cas dans la suite de la description.

30 La chaîne d'émission 400 représentée à la figure 4 comporte un applicatif 401 apte à générer ou transformer une séquence de données

binaires à émettre. Les données à émettre peuvent être des données textuelles, audio, vidéo ou toute autre information. L'applicatif 401 peut également comporter une fonction de codage source, par exemple un codeur audio, image ou vidéo apte à supprimer ou réduire la redondance d'information ou à réduire le bruit affectant la séquence. L'applicatif 401 génère en sortie une séquence binaire utile T à transmettre. L'invention est avantageusement mise en œuvre en sortie de l'applicatif 401 en modifiant la séquence binaire utile T pour y insérer des bits factices de sorte à obtenir en sortie de chaîne d'émission une séquence de symboles modulés $F(T)(t)$ à émettre comprenant au moins un motif de synchronisation factice.

La chaîne d'émission 400 peut également comporter un module de codage correcteur 402.

L'objectif d'une fonction de codage correcteur est de transformer la séquence binaire de données utiles reçue en sortie de l'applicatif 401 en une séquence binaire protégée de sorte que l'impact des erreurs dues au canal de transmission soit le plus réduit possible. Pour rendre la séquence binaire de données utiles plus robuste aux imperfections du canal de transmission, la fonction de codage correcteur ajoute de la redondance à cette séquence binaire.

La détermination de la fonction de transfert inverse d'un module de codage correcteur est équivalente à la recherche de la séquence binaire à produire en entrée du codeur correcteur pour, en sortie, obtenir une séquence codée dans laquelle la valeur et la position d'un nombre prédéterminé de bits sont imposées.

Il existe différents types de codes correcteurs parmi lesquels les codes linéaires en blocs, les codes convolutifs ou encore les turbo-codes et les codes à faible densité LDPC.

Par la suite, la détermination de la fonction de transfert inverse d'un codeur correcteur est décrite pour différents types de codes correcteurs, les

codes linéaires en blocs, les codes convolutifs, ainsi que les turbo-codes et les codes LDPC.

Codes linéaires en blocs

5

Un code correcteur linéaire en blocs de rendement k/n transforme une séquence binaire comprenant k symboles en une séquence binaire protégée comprenant n symboles avec n strictement supérieur à k . Un tel code introduit donc $n-k$ symboles de redondance. Les symboles peuvent être des bits ou être constitués de plusieurs bits concaténés. L'opération de codage en blocs est une transformation bi-univoque d'un mot du message $i = (i_0, \dots, i_{k-1})$ en un mot de code $c = (c_0, \dots, c_{n-1})$ définie par le système d'équations linéaires suivant (où « + » désigne l'addition modulo 2, « . » désigne la multiplication modulo 2) et g_{en} sont des coefficients à valeur dans le corps de Gallois $GF(2)$, rangés dans une matrice de taille $n \times k$:

$$c_n = \sum_{e=0}^k g_{en} i_e, \text{ pour } 0 \leq i \leq n-1$$

Parmi les 2^n séquences binaires comprenant n bits qui existent, seules 2^k peuvent être générées. L'opération de codage correcteur limite donc la possibilité de générer n'importe quelle séquence binaire souhaitée.

Le codage en bloc consiste à effectuer le produit d'un vecteur d'information en entrée de k bits par une matrice binaire, de rang plein, de taille $k \times n$, appelée matrice génératrice, pour obtenir un vecteur codé de n bits. Souvent le code est dit systématique à gauche, respectivement à droite, lorsque les k premiers, respectivement les k derniers, bits du vecteur codé de n bits correspondent aux k bits du vecteur d'information en entrée. L'opération de codage peut être illustrée par la relation suivante, où $i_0, \dots, i_{k-1}$ sont les bits de la séquence utile en entrée, $c_0, \dots, c_{n-1}$ sont les bits de la séquence codée et $m_{i,j}$ sont les coefficients de la matrice génératrice du code.

$$[c_0 \quad \dots \quad c_{n-1}] = [i_0 \quad \dots \quad i_{k-1}] \cdot \begin{bmatrix} m_{0,0} & m_{0,1} & m_{0,2} & \dots & m_{0,n-1} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,0} & m_{k-1,1} & m_{k-1,2} & \dots & m_{k-1,n-1} \end{bmatrix}$$

Cas de codes en bloc systématique

- 5 Dans le cas où le code est systématique à gauche, la séquence codée s'écrit $[i_0 \quad \dots \quad i_{k-1} c_k \dots c_{n-1}]$. La transformée inverse de l'opération de codage consiste à analyser le mot reçu pour déterminer s'il s'agit d'un mot de code possible. S'il ne s'agit pas d'un mot de code possible, il faut le remplacer par le mot de code se trouvant à distance minimale du mot de
- 10 code reçu. Ensuite, comme le code est systématique, l'information est obtenue en supprimant les n-k derniers bits du mot. Autrement dit, il est possible d'imposer, par le choix de la séquence d'entrée, la valeur en sortie des k premiers bits du mot codé. Les valeurs des n-k bits restants se déduisent alors des valeurs choisies pour les k bits qui ont été forcés. Il en
- 15 est exactement de même pour un code en bloc systématique à droite.

Cas de codage en bloc issus de codes cycliques

- 20 Les codes en bloc utilisés couramment sont des codes en bloc cycliques ou bien sont dérivés de codes en bloc cycliques par poinçonnage ou raccourcissement.

Dans le cas d'un code en bloc cyclique, si $[c_0 \quad \dots \quad c_{n-1}]$ est un mot de code, toute permutation circulaire du mot $[c_i \quad c_{i+1} \quad \dots \quad c_{n-1} \quad c_0 \quad \dots \quad c_{i-1}]$ est aussi un

25 mot de code.

En écrivant sous forme polynômiale les mots de code $c(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1}$, tous les mots de codes apparaissent comme des multiples d'un même polynôme $g(x) = g_0 + g_1 \cdot x + g_2 \cdot x^2 + \dots + g_{n-k} \cdot x^{n-k}$, de degré n-k, appelé polynôme générateur du code.

Un mot d'information $[i_0 \dots i_{k-1}]$ peut aussi s'écrire sous forme polynômiale $i(x) = i_0 + i_1 \cdot x + i_2 \cdot x^2 + \dots + i_{k-1} \cdot x^{k-1}$. Il est toujours possible d'écrire l'opération de codage sous forme systématique à droite. Cette opération de codage consiste à calculer la division de $i(x) \cdot x^{n-k}$ par $g(x)$. Le reste de la division est $\nu(x)$ (de degré inférieur ou égal à $n-k-1$) et le quotient de la division est $k(x)$. On a donc $i(x) \cdot x^{n-k} = k(x) \cdot g(x) + \nu(x)$, et $c(x) = i(x) \cdot x^{n-k} + \nu(x) = k(x) \cdot g(x)$. Comme $\nu(x)$ est de degré inférieur ou égal à $n-k-1$, les valeurs des coefficients de $c(x)$ pour les degrés supérieurs ou égaux à $n-k$ sont les coefficients de $i(x)$ décalés de $n-k$.

Il est donc toujours possible d'écrire les codes en bloc cycliques sous forme systématique à droite, et donc d'imposer la valeur des k derniers bits qui sont égaux aux bits d'information. Comme le code est cyclique, et que donc toute permutation circulaire d'un mot de code est aussi un mot de code, cela signifie qu'il est aussi possible, toujours pour les codes cycliques, d'imposer la valeur de tout groupe de k bits consécutifs d'un mot de code.

Par ailleurs, la dépendance entre les valeurs des bits en entrée du codeur et les bits en sortie du codeur est linéaire. Les valeurs des bits qui doivent être forcés en entrée du codeur dépendent de façon linéaire des valeurs des autres bits en entrée du codeur et des valeurs des bits forcés en sortie du codeur.

Cas général des codes en bloc

Dans le cas plus général où le code n'est pas systématique ni issu de codes cycliques, une condition suffisante pour pouvoir imposer la valeur d'un groupe de d bits en sortie du codeur, avec d inférieur ou égal à k , est que l'ensemble des positions $p_1, p_2, \dots, p_d$, des bits dans la séquence codée doit être tel que la sous-matrice de la matrice génératrice du code :

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}$$

est de rang plein, c'est-à-dire de rang égal à d.

En effet lorsque cette condition est remplie, il est possible de déterminer la séquence $[i_0 \dots i_{k-1}]$ en entrée du codeur qui permet de fixer

- 5 les valeurs de d bits ou symboles dans la séquence codée en résolvant le système d'équations suivant :

$$[i_0 \dots i_{k-1}] = [c_{p_1} \dots c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \text{ où } [c_{p_1} \dots c_{p_d}] \text{ sont les}$$

bits ou symboles dont la valeur est fixée dans la séquence codée, les indices $p_1, p_2, \dots, p_d$, désignant les positions des bits ou symboles dans la séquence de

- 10 n bits ou symboles.

Ainsi, pour tous les codes en bloc usuels, qui sont systématiques, ou cycliques, ou construits à partir de codes cycliques, c'est à dire pour la plupart des codes usuels, il est possible d'imposer, par le choix des entrées
15 du codeur, n'importe quel groupe de k bits consécutifs parmi les n bits du vecteur codé. Par ailleurs, dans le cas de deux mots de code successifs, en imposant les k derniers bits du 1^{er} mot de code et les k premiers bits du 2nd mot de code, il est possible d'imposer la valeur d'un groupe de 2k bits successifs sur une séquence binaire comprenant au moins deux mots de
20 codes.

- Dans le cas le plus général, la sous-matrice de la matrice génératrice du code en bloc correspondant aux positions des bits ou symboles à fixer est de rang plein. Cependant, dans certains cas, certaines sous-matrices de la
25 matrice génératrice peuvent ne pas être de rang plein. Un tel cas est illustré sur un exemple non limitatif d'un code de Hamming (7,4) dont la matrice génératrice $M(7,4)$ est donnée par

$$M(7,4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Le polynôme générateur de ce code est $g(x) = 1 + x + x^3$. Pour ce code il est possible d'imposer les 4 derniers bits (m_0 à m_3) du mot de code car le

5 codage est systématique à droite. Pour obtenir le mot de code suivant : $[c_0 \ c_1 \ c_2 \ m_0 \ m_1 \ m_2 \ m_3]$, il suffit de coder le vecteur d'information $i(x) = [m_0 \ m_1 \ m_2 \ m_3]$.

L'opération de codage est représentée par la relation suivante :

$c_{sys,droite}(x) = i(x) \cdot x^{n-k} + v(x) = k_{sys,droite}(x) \cdot g(x)$, où $k_{sys,droite}(x)$ est le vecteur à

10 coder et $c_{sys,droite}(x)$ est le mot de code obtenu.

Il est également possible d'imposer, par exemple, les 4 premiers bits du mot de code aux valeurs du vecteur d'information $[m_0 \ m_1 \ m_2 \ m_3]$. Pour cela il faut trouver le vecteur d'information qui, une fois codé, donne le mot de code

15 suivant :

$$k_{sys,gauche}(x) \cdot g(x) = m_0 + m_1 \cdot x + m_2 \cdot x^2 + m_3 \cdot x^3 + c'_4 \cdot x^4 + c'_5 \cdot x^5 + c'_6 \cdot x^6.$$

Pour calculer ce vecteur d'information $k_{sys,gauche}(x)$, on utilise la propriété que le code est invariant par permutation circulaire. On passe d'un codage systématique à droite à un codage systématique à gauche par 4

20 permutations circulaires vers la droite. Donc le mot de code $[c_0 \ c_1 \ c_2 \ m_0 \ m_1 \ m_2 \ m_3]$ devient le mot de code $[m_0 \ m_1 \ m_2 \ m_3 \ c_0 \ c_1 \ c_2]$ en effectuant ces 4 permutations circulaires. Le mot de code $[m_0 \ m_1 \ m_2 \ m_3 \ c_0 \ c_1 \ c_2]$ est obtenu en codant le vecteur d'information $i'(x) = [m_3 \ c_0 \ c_1 \ c_2]$ (car le code est systématique à

25 droite).

Par contre, on peut remarquer que si on considère la deuxième, la quatrième, la cinquième et la sixième colonne de la matrice génératrice du code $M(7,4)$, on obtient la sous-matrice suivante qui n'est pas de rang plein, les coefficients de sa dernière ligne étant égaux à 0 :

$$5 \quad \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

En effet, le rang d'une matrice correspond au nombre de colonnes indépendantes de la matrice ou de façon équivalente au nombre de lignes indépendantes de la matrice.

Il n'est donc pas possible de forcer les valeurs de ces 4 bits (le deuxième, le quatrième, le cinquième et le sixième) du mot codé : si on force la valeur de 3 de ces bits, la valeur du quatrième bit se déduit des valeurs imposées aux trois bits forcés.

On ne peut donc pas, pour tout code (n, k) de rendement k/n , forcer la valeur de n'importe quel groupe de k bits. Par contre, pour les codes utilisés le plus fréquemment, il est possible de forcer la valeur de tout groupe de $n-k$ bits consécutifs lorsque la sous-matrice de la matrice génératrice du code associée aux positions des bits à fixer est de rang plein.

On précise ici les dépendances entre les valeurs des bits en entrée du codeur et les valeurs des bits en sortie du codeur.

- Si on souhaite imposer la valeur d'un nombre limité l , inférieur à k , de bits en sortie du codeur, il suffit d'imposer la valeur de l bits en entrée du codeur.
- Par contre, la valeur de ces bits dépend non seulement de la valeur du motif généré en sortie du codeur mais aussi des valeurs des autres bits en entrée du codeur (que l'on peut le cas échéant forcer aussi dans le cadre de la mise en oeuvre de la présente invention).

En reprenant l'exemple de la matrice de codage $M(7,4) = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$,

- si l'on veut forcer la valeur des 2 premiers bits codés, c_0 et c_1 , on peut choisir un vecteur d'information $i(x) = [f_0 \ f_1 \ i_0 \ i_1]$, où i_0 et i_1 sont des bits d'information laissés libres et f_0 et f_1 des bits forcés pour obtenir le motif voulu. Les valeurs qu'il faut choisir pour f_0 et f_1 afin d'obtenir les valeurs voulues de c_0 et c_1 sont données par les relations :

$$f_0 = c_0 + i_1 + i_2$$

$$f_1 = c_0 + c_1 + i_2$$

- Cela illustre que les valeurs des bits forcés en entrée du codeur (f_0 et f_1) dépendent de façon linéaire des valeurs des bits du motif en sortie du codeur (c_0 et c_1) et des valeurs des autres bits en entrée du codeur.

Codes convolutifs

- Les codes convolutifs constituent la deuxième grande famille des codes correcteurs d'erreurs. Alors que les codes linéaires en blocs permettent de découper le message en blocs de k symboles, les codes convolutifs appliquent une fenêtre glissante de $k(m+1)$ symboles sur le message et produisent une séquence continue de symboles codés.
- En général, les symboles sont binaires (i.e. à valeur 0 ou 1 dans le corps de Gallois $GF(2)$, « + » désigne l'addition modulo 2 et « . » désigne la multiplication modulo 2). Le plus souvent les codes convolutifs ont pour paramètre $k=1$ et le rendement du code est donc de la forme $1/n$. Soit a_j un symbole d'information, les symboles de parité b_j associés à a_j sont définis par la relation de convolution suivante, où les $g_{e,j,i}$ sont les coefficients de n^*k polynômes de degré m à coefficients et valeurs dans le corps de Gallois $GF(2)$ (le code est entièrement défini par l'ensemble des coefficients $g_{e,j,i}$, $e = 0, \dots, m$, $j=0 \dots k-1$, $i=0 \dots n-1$) :

$$b_{p \cdot n + i} = \sum_{e=0}^m \sum_{j=0}^{k-1} g_{e,j,i} \cdot a_{p \cdot k + j - e \cdot k}, \text{ pour } i = 0 \dots n-1, \forall p.$$

Les n symboles en sortie du codeur dépendent linéairement des $k^*(m+1)$ derniers symboles en entrée du codeur.

5

A partir d'un code de rendement $1/n$, des codes dit « dérivés », correspondant à $k > 1$, peuvent être construits par poinçonnage (le plus souvent $k=n-1$ après poinçonnage).

10

Plus généralement, lorsque le rendement du code est égal à k/n , le codage convolutif est un codage périodique de période k bits sur le signal binaire en entrée. Pour chaque nouveau groupe de k bits, n bits codés sont calculés. Les n bits codés sont des combinaisons de bits portant sur les $(m+1)$ derniers groupes de k bits. m est la longueur de contrainte du code.

15

On illustre à présent sur un exemple non limitatif, l'étape du procédé selon l'invention qui consiste à inverser la fonction de transfert d'un code convolutif, autrement dit à déterminer la séquence de bits à produire en entrée pour obtenir en sortie une séquence codée dans laquelle la valeur et la position d'un nombre prédéterminé de bits sont fixées.

20

Exemple non limitatif d'un code convolutif de rendement $\frac{1}{2}$

On considère un code binaire convolutif usuel, dont les registres sont représentés sur la figure 5, de rendement $1/2$ et de longueur $m+1=7$ défini par deux polynômes de degrés 6 définis en notation octale par (171, 133). Ces deux polynômes s'écrivent $G_1(X)=1 + X + X^2 + X^3 + X^6$ et $G_2(X)=1 + X^2 + X^3 + X^5 + X^6$ et correspondent aux relations de récurrence $b_{2n}=a_n + a_{n-1} + a_{n-2} + a_{n-3} + a_{n-6}$ et $b_{2n+1} = a_n + a_{n-2} + a_{n-3} + a_{n-5} + a_{n-6}$ dans le corps de Galois GF2 (« + » désigne l'addition modulo 2). Les polynômes G_1 et G_2 sont

30

appliqués aux bits en entrée pour former respectivement les bits de sortie d'indice pair et les bits de sorties d'indice impair entrelacés ensuite deux à deux sous la forme $b_{2n}b_{2n+1}$ pour former un flux binaire de taille égale à un multiple de 2. On illustre ci dessous la possibilité de choisir le train binaire en

5 entrée de manière à générer des motifs voulus après codage.

Pour ce code, à chaque période, pour un bit produit en entrée du codeur, deux bits sont générés en sortie. Suivant l'état des registres du codeur, les deux bits (b_{2n} , b_{2n+1}) en sortie sont à choisir parmi soit (0,0) ou (1,1), soit (0,1) ou (1,0). En effet, le dernier bit qui entre dans le codeur est

10 utilisé pour le calcul de chacune des deux sorties du codeur : en changeant ce bit, on change les valeurs des deux sorties. Cela signifie que, pour un état du registre donné, les deux bits possibles en sortie sont à choisir parmi deux groupes complémentaires. Il est donc toujours possible de choisir le bit en entrée de manière à forcer la valeur d'un des deux bits de sortie. On peut

15 donc aisément avec ce code forcer un bit sur deux en sortie du codeur.

En pratique, pour les codes les plus performants utilisés dans un modem, les groupes de bits en sorties, pour un état du codeur, sont choisis de manière à être à distance maximale les uns des autres. Pour ces codes, le dernier bit qui entre dans le codeur est utilisé pour le calcul de chacune

20 des deux sorties du codeur. On peut donc, pour tous les codes de rendement 1/2 usuels, forcer la valeur d'un bit sur deux en sortie du codeur.

On démontre à présent qu'il est également possible de choisir une série de bits en entrée du codeur de façon à obtenir en sortie du codeur une séquence codée comprenant une série de bits consécutifs de valeur fixée.

25 Nous considérons à nouveau l'exemple précédent du codeur $r=k/n=1/2$, (171, 133) de longueur $m+1=7$. Pour ce codeur, la réponse impulsionnelle, c'est-à-dire la réponse du codeur à une séquence binaire d'entrée comprenant un bit de valeur 1 précédé et suivi de bits ayant tous la valeur 0, est donnée par la séquence 11101111000111, de longueur $14=2m+2$.

L'opération de codage peut s'écrire sous la forme matricielle, représentée en figure 6, où les lignes de la matrice génératrice du code correspondent à la réponse impulsionnelle du codeur, décalées de 2 bits d'une ligne à l'autre (car $n=2$), ou plus généralement décalées de n bits d'une ligne à l'autre lorsque le
5 code est de rendement $1/n$.

On obtient alors un formalisme identique à celui utilisé pour les codes en blocs linéaires, c'est-à-dire que la séquence codée est obtenue en effectuant le produit matriciel de la séquence d'information avec la matrice
10 génératrice définie ci-dessus.

La même règle précédemment édictée concernant les codes en blocs linéaires peut ainsi être appliquée aux codes convolutifs, c'est-à-dire qu'il est possible d'imposer la valeur et la position d'un ensemble de bits en sortie du codeur si et seulement si la sous-matrice M correspondant aux colonnes des
15 bits de sortie est de rang plein, ce qu'illustre la figure 7

On voit donc que pour ce code convolutif, il est possible de forcer la valeur de 14 bits consécutifs en sortie du codeur, à partir des entrées. On peut aussi interpréter ce résultat du point de vue des relations de parité entre bits codés. Ces relations –déterministes- caractérisent de manière
20 biunivoque les dépendances vérifiées par les groupes de bits codés. En d'autres termes, les valeurs des bits codés correspondant à une relation de parité sont inter-dépendantes, Les valeurs des bits codés ne correspondant pas à une relation de parité peuvent être fixées indépendamment les unes des autres. Le code précédemment considéré a des relations de parité de
25 longueur 14 et espacées de 2 bits. Il est donc possible de choisir 14 bits codés consécutifs ne correspondant à aucune relation de parité entière, ce qui signifie que leurs valeurs peuvent être fixées de manière indépendante.

Plus généralement il est possible d'imposer la valeur et la position
30 d'un ensemble de bits en sortie du codeur si et seulement si la sous-matrice

$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix} \text{ est de rang plein, avec } m_{i,j} \text{ les}$$

coefficients de la matrice génératrice et $p_1, p_2, \dots, p_d$, l'ensemble des positions des bits dont la valeur est fixée dans la séquence codée.

- 5 Finalement, dans le cas d'un code de rendement $\frac{1}{2}$, le nombre maximum de bits successifs dont il est possible de fixer la valeur est égal à la longueur de la réponse impulsionnelle du code, c'est-à-dire $2m+2$ où m est la longueur de contrainte du code.

10 Exemple non limitatif d'un code convolutif poinçonné construit à partir d'un code convolutif de rendement $\frac{1}{2}$

Pour les codes poinçonnés, qui présentent une redondance plus faible, il est possible de forcer un plus grand nombre de bits consécutifs en
 15 sortie du codeur. Par exemple, le code défini par les polynômes précédents (171, 133) peut être poinçonné pour obtenir un code de rendement $\frac{3}{4}$. Les relations de parité de ce code ont une longueur 26 (11111101011011001010011111) et sont espacées de 4 bits. Cet exemple
 20 est illustré à la figure 8 sur laquelle sont représentés les indices 800 des bits en sortie du codeur, une portion P_{28} de 28 bits consécutifs ne comportant aucune relation de parité entière et 3 relations de parités R_1, R_2, R_2 liées audit code, de longueur égale à 26 bits. Les relations de parités sont vérifiées pour toutes les séquences de 26 bits démarrant sur un indice décalé de 4 bits pour chaque nouvelle séquence.

25

Par analogie directe avec ce qui précède, il est donc possible de choisir les valeurs prises par un groupe de $4(6+1)=28$ bits.

Cas général des codes convolutifs usuels

Pour le cas plus général d'un code convolutif de rendement $(n-1)/n$, et de longueur de contrainte m , les relations de parité sont généralement de longueur $n \cdot m + 2$ (dans le cas de l'exemple précédent, on avait $m=6$ et $n=4$ après poinçonnage, les relations de parité étaient de longueur 26). Dans ce cas, et par analogie avec ce qui précède il est possible de choisir les bits en entrée du codeur de manière à forcer la valeur de $n \cdot (m+1)$ bits consécutifs en sortie du codeur.

On voit donc qu'avec les codes convolutifs, il est possible, comme pour les codes en bloc de forcer la valeur de certains bits codés, éventuellement consécutifs et parfois en grand nombre. La condition pour que cette opération soit possible est que la matrice formée par les colonnes de la matrice de codage correspondant aux bits forcés soit de rang plein.

Il est en particulier possible de forcer la valeur de séquences de bits consécutifs de longueur significative, particulièrement quand le rendement du code est proche de 1 et quand la mémoire m du code est importante.

De même que pour les codes en bloc, la dépendance entre les valeurs des bits en entrée du codeur et les bits en sortie du codeur est linéaire. Les valeurs des bits qui doivent être forcés en entrée du codeur dépendent de façon linéaire des valeurs des autres bits en entrée du codeur et des valeurs des bits forcés en sortie du codeur.

Turbo-Codes produit

L'invention s'applique également aux codes correcteurs de type turbo-codes produit.

Les turbo-codes sont des codes correcteurs qui combinent au moins deux codes simples en entrelaçant les entrées de manière à ce que chacun des codes simples voit une série d'informations différentes d'une part, et que l'information propre à chaque bit, bloc ou message soit répartie sur ses

voisins d'autre part. De ce fait, même si une partie des bits, des blocs ou des messages est altérée pendant la transmission, l'information correspondante existe encore peu ou prou sur des bits, blocs ou messages voisins. La procédure de décodage est itérative et collaborative entre chaque code

5 simple. Elle fait intervenir une notion de confiance sur chaque bit, bloc ou message décodé et diffère la décision finale sur leurs valeurs (« décision douce » ou « soft decision » en anglais). Chacun des décodeurs transmet aux autres l'information issue de son propre décodage (dite information extrinsèque) qui est multiplexée à l'information en entrée des autres codeurs.

10 Le bit, bloc ou message ainsi transmis est décodé une seconde fois par les autres codeurs simples, et l'information correspondante re-transmise vers les autres codeurs, etc. (d'où la dénomination « turbo » qui est liée à la procédure de décodage et non au code proprement dit).

Les codes simples employables sont multiples.

15 Il est possible d'utiliser des codes convolutifs. Des codes convolutifs récurrents et systématiques sont en pratique particulièrement adaptés. Les codes peuvent être placés en série ou en parallèle. La gestion astucieuse de l'entrelacement et de la détection/correction itérative des données par chaque code simple permet de démultiplier le pouvoir détecteur et correcteur

20 du procédé global tout en limitant le nombre d'itérations et la complexité.

Une autre structure de codage turbo correspond à des codes produits. Dans la version la plus simple correspondant à des codes produits à deux dimensions, le code produit de rendement $\frac{k_1 \cdot k_2}{n_1 \cdot n_2}$ est construit à partir de

25 deux codes élémentaires C_1 et C_2 de rendement $\frac{k_1}{n_1}$ et $\frac{k_2}{n_2}$. Les codes

élémentaires utilisés sont des codes en bloc très simples (typiquement des codes de parité, des codes de Hamming ou bien des codes de Hamming étendus). Les $n_1 \cdot n_2$ bits successifs apparaissent comme une suite de n_2

mots de code C_1 . En considérant le train binaire décimé d'un facteur n_1 , on obtient des mots du code C_2 .

Les turbo-codes produits, construits à partir de plusieurs codes en bloc s'apparentent à des codes en bloc quand il s'agit de déterminer s'il est possible de générer le motif voulu. En effet, l'opération de codage peut se décomposer en des opérations de codage et d'entrelacement.

Nous allons illustrer la décomposition d'un code produits en plusieurs opérations de codage et d'entrelacement à partir d'un exemple simple et non limitatif. Nous considérons un code produit construit à partir de deux codes

10 C_1 et C_2 de rendement $\frac{k_1}{n_1}$ et $\frac{k_2}{n_2}$.

L'opération de codage d'un bloc de $k_1 \cdot k_2$ bit peut se décomposer de la façon suivante :

- Un codage avec le code C_1 du train binaire : le codage transforme k_2 groupes de k_1 bits en k_2 groupes de n_1 bits.
- 15 - Un entrelacement ligne/colonne simple : les bits sont écrits ligne par ligne dans une matrice de taille k_2 lignes et n_1 colonnes. Les bits sont ensuite lus colonnes par colonne.
- Un codage avec le code C_2 du train binaire : le codage transforme n_1 groupes de k_2 bits en n_1 groupes de n_2 bits.
- 20 - Un entrelacement ligne/colonne simple : les bits sont écrits ligne par ligne dans une matrice de taille n_1 lignes et n_2 colonnes. Les bits sont ensuite lus colonnes par colonne.

En pratique, si on considère une portion limitée de bits consécutifs d'un bloc de $n_1 \cdot n_2$ bits codés (une portion de longueur $n \cdot k'$ sensiblement plus petite que $n_1 \cdot k_2$), les contraintes du code C_2 n'affectent pas cette portion car il n'existe aucune relation de parité liée au code C_2 qui soit contenue entièrement dans cette portion de bits. En conséquence, si on considère une portion de bits consécutifs de taille inférieure à $n_1 \cdot k_2$, tout se passe comme

s'il n'y avait que le code C_1 quand il s'agit de déterminer les motifs qu'il est possible de forcer. A l'instar des codes en bloc, il est donc possible, notamment, de générer des motifs bits consécutifs de taille $2 \cdot k_1$.

5 Codes LDPC (Low Density Parity Check)

Les codes LDPC sont des codes en bloc particuliers qui sont construits de manière à ce que les bits de parité soient calculés en faisant intervenir une relation de parité de poids faible. Il s'agit en pratique de codes
10 en bloc systématiques (mais non cycliques) et l'analyse faite sur les codes en bloc s'applique par analogie à la description faite précédemment.

Les codes LDPC sont généralement de très grande taille. A l'instar de ce qui a été établi pour les codes en blocs, les codes LPDC permettent donc, notamment, de générer des motifs choisis avec séries de bits consécutifs de
15 grande longueur.

En résumé, pour tout code correcteur pour lequel l'opération de codage peut être réalisée en multipliant la séquence d'information par une matrice génératrice pour obtenir la séquence codée, il est possible de fixer la
20 valeur et la position d'un ensemble de bits de la séquence codée en imposant une séquence binaire d'entrée particulière. Cette possibilité existe cependant uniquement si la sous-matrice, de la matrice génératrice, définie

par
$$\begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}$$
 est de rang plein, avec $m_{i,j}$ des

coefficients de la matrice génératrice et $p_1, p_2, \dots, p_d$, l'ensemble des positions
25 des bits dont la valeur est fixée dans la séquence codée. Une matrice de rang plein est une matrice dont toutes les colonnes sont indépendantes. Si cela est le cas, il suffit de résoudre le système d'équations

$$[i_0 \quad \dots \quad i_{k-1}] = [c_{p_1} \quad \dots \quad c_{p_d}] \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \quad \text{où } [c_{p_1} \dots c_{p_d}] \text{ sont les}$$

bits ou symboles dont la valeur est fixée dans la séquence codée et $i_1, \dots, i_k$ sont les inconnues du système à fournir en entrée du codeur.

- 5 La chaîne d'émission 400 peut également comporter un module d'embrouillage 403, également appelé brassage.

Le brassage, ou embrouillage, est utilisé pour rendre la séquence binaire à émettre la plus aléatoire possible, en vue d'améliorer la synchronisation symbole mais également de contribuer à la protection des
10 contenus des messages brassés. Son but est de supprimer les longues suites de bits égaux à 0 ou à 1 qui empêchent une récupération correcte du rythme symbole. Il existe plusieurs types de brasseurs parmi lesquels les brasseurs synchrones nécessitant une référence temporelle préalable ou les brasseurs auto-synchronisant.

- 15 Cette transformation est inversible, l'opération d'embrouillage consiste à transformer un groupe de L bits en un autre groupe de L bits et l'opération d'inversion de la transformation du train binaire est une opération effectuée par un récepteur. Ainsi, la fonction de transfert inverse d'un brasseur ou embrouilleur est aisément déductible de sa fonction de transfert directe.

20

Cas d'un embrouilleur synchrone

- Dans le cas d'un embrouilleur synchrone, une séquence pseudo aléatoire est ajoutée modulo 2 au signal binaire à embrouiller. La série
25 binaire $\{\dots, e_k, e_{k+1}, \dots\}$ représentant la séquence d'embrouillage est périodique, de période L longue. Les séquences les plus couramment utilisées pour l'embrouillage sont des séquences de longueur maximale, construites à partir d'un registre à décalage. Ces séquences sont générées par un registre à décalage rebouclé suivant un polynôme primitif $E(X) = 1 +$

$c_1.X + \dots + c_P.X^P$ de degré P . On obtient une séquence de période $L = 2^P - 1$. Les séquences utilisées pour l'embrouillage peuvent cependant être tronquées. La k -ième sortie e_k du registre à décalage vérifie la relation de récurrence $e_k = c_1.e_{k-1} + \dots + c_P.e_{k-P}$, avec $c_1, \dots, c_P$ des coefficients constants dans le corps de Galois $GF2$ indépendants des données d'entrées et des données embrouillées, « + » désignant toujours l'addition modulo 2 dans $GF2$. Pour une série binaire en entrée notée $\{\dots, b_k, b_{k+1}, \dots\}$, la transformation F opérée par un embrouilleur synchrone peut s'écrire $F(\{\dots, b_k, b_{k+1}, \dots\}) = \{\dots, b'_k, b'_{k+1}, \dots\}$ avec $b'_k = b_k + e_k$. La transformation F est parfaitement inversible. En effet, cette transformation F est égale à son inverse $F^{-1} = F$ puisque $b_k = b'_k + e_k$ et sachant que $e_k + e_k = 0$ dans le corps de Galois $GF2$. L'opération F consiste en une nouvelle addition modulo 2 de la valeur des bits embrouillés par les sorties du même registre à décalage qu'à l'émission pour produire une séquence désembrouillée de longueur identique. Afin de pouvoir forcer le train binaire en sortie d'un embrouilleur synchrone, il suffit d'être correctement synchronisé avec l'opération d'embrouillage. Autrement dit, les positions des indices k ou de début de période doivent coïncider avant d'appliquer la transformation $F^{-1} = F$ aux bits embrouillés. Dans le cas d'un embrouilleur synchrone, la dépendance entre la valeur i d'un bit en entrée de l'embrouilleur et la valeur c du bit en sortie de l'embrouilleur est affine. Suivant la position du bit considéré, la dépendance est, soit $c=i$, soit $c=i+1$.

Cas d'un embrouilleur auto-synchronisant

Dans le cas d'un embrouilleur auto-synchronisant, les états du registre sont remplis avec un nombre fini de données embrouillées. La sortie du registre est additionnée modulo 2 au bit de donnée en entrée pour former le nouveau bit embrouillé.

La transformation F , opérée par un embrouilleur auto-synchronisant est définie par un registre à décalage de polynôme primitif $E(X) = 1 + c_1.X + \dots +$

$c_P X^P$ de degré P de période $L = 2^P - 1$ et s'écrit pour une série binaire en entrée notée $\{\dots, b_k, b_{k+1}, \dots\}$ sous la forme : $F(\{\dots, b_k, b_{k+1}, \dots\}) = \{\dots, b'_k, b'_{k+1}, \dots\}$ avec $b'_k = b_k + c_1 \cdot b'_{k-1} + \dots + c_P \cdot b'_{k-P}$ « + » désignant toujours l'addition modulo 2 dans GF2). Ceci permet d'effectuer le désembrouillage

5 en réception de façon simple et sans avoir besoin de se synchroniser au préalable. En réception, on injecte les bits embrouillés $b'_{k-1}, \dots, b'_{k-P}$ dans le même registre qu'à l'émission pour reconstituer la sortie $b_k = b'_k + c_1 \cdot b'_{k-1} + \dots + c_P \cdot b'_{k-P}$.

La difficulté pour forcer les bits en sortie provient ici de ce que

10 l'expression des états du registre en fonction des données d'entrées b_k (données non embrouillées) fait intervenir un nombre illimité desdites données d'entrée. Autrement dit toutes les entrées b_k depuis la mise en route de l'embrouilleur interviennent dans la valeur des états du registre.

Pour pouvoir forcer la valeur des données embrouillées il est

15 indispensable d'avoir accès de façon synchrone aux données embrouillées pour pouvoir adapter les données d'entrée de façon dynamique. Cette condition est nécessaire à l'application de l'invention si la chaîne d'émission comporte un embrouilleur auto-synchronisant.

20 La chaîne d'émission 400 peut également comporter un module d'entrelacement 404.

L'entrelacement est très utilisé sur les canaux de transmission pour lesquels les occurrences des erreurs sont regroupées par paquets. Sa fonction est de répartir aussi uniformément que possible lesdites erreurs. En

25 réception, les erreurs sont, après désentrelacement, placées de telle façon qu'elles impactent des mots de code différents. Ces erreurs peuvent alors être considérées comme dé-corrélées, et le pouvoir correcteur des décodeurs permet d'en minimiser l'impact. L'entrelacement apparaît aussi comme un moyen d'introduire de la diversité temporelle dans la chaîne de

30 transmission et contribue de ce fait à protéger celle-ci des évanouissements, des interférences et des brouillages éventuels. De nombreux entrelaceurs

sont constitués par une table, les bits en entrée sont alors rangés par lignes dans la table, et les bits en sortie sont produits par lecture en colonne de la table.

La transformation opérée par un entrelaceur est également une opération
5 inversible. Sa fonction de transfert inverse est déductible de sa fonction de transfert directe. En effet, il s'agit d'une transformation qui transforme un bloc de L bits en un autre bloc de L bits, en permutant simplement l'ordre des bits. Afin d'obtenir un bloc de bits voulu après entrelacement $B' = b'_k, \dots, b'_{k+L-1}$, il suffit d'être synchronisé sur le bloc et d'appliquer sur ce train binaire B' la
10 permutation inverse, ce qui correspond à l'opération de dés-entrelacement effectuée classiquement par un récepteur. On obtient alors un bloc de bits $B = b_k, \dots, b_{k+L-1}$, qui une fois entrelacé, est strictement égal au train binaire B.

La chaîne d'émission 400 peut également comporter un module de
15 mise en trame 405. La mise en trame 405 permet au récepteur de se synchroniser sur les transformations du train binaire telles que les opérations d'entrelacement ou décodage, puis de récupérer des données structurées sous la forme d'un multiplexage de plusieurs flux ou de données sous forme de mots ou d'octets. Pour cela, une fois les données structurées sous forme
20 de trames correspondant à des motifs d'ordonnancement périodiques, la synchronisation du récepteur sur les trames s'effectue à l'aide de motifs périodiques synchrones des trames. Chaque trame est ainsi précédée, et/ou suivie, et/ou contient un mot de synchronisation spécifique utilisé pour effectuer la synchronisation du récepteur sur les trames reçues. Plusieurs
25 trames peuvent également être regroupées pour former une multi-trame ou une hyper-trame. Les motifs de synchronisation utilisés pour la mise en trame sont répétés dans la séquence émise et peuvent être détectés et brouillés comme cela a été expliqué précédemment.

30 La mise en trame limite la possibilité de forcer le signal modulé voulu puisque certains bits prennent des valeurs imposées à intervalle régulier, et

qu'il est nécessaire de garder ces bits pour le bon fonctionnement du récepteur en liaison avec l'émetteur. Cette transformation n'est donc pas surjective pour des blocs de données dont la taille en sortie dépasse celle d'un bloc de données utiles par trame.

- 5 Toutefois, la mise en trame n'occupe qu'une part très limitée (et temporellement bien délimitée) du débit total (exemple : motif de début de trame n'utilisant que quelques symboles, etc.) et n'empêche pas de générer le signal codé voulu au sein d'une trame.

Ce point n'est donc pas un écueil particulier pour la mise en œuvre de
10 l'invention.

La chaîne d'émission 400 peut également comporter un module de codage binaire à signal 406.

- Le codage binaire à signal est utilisé pour adapter le signal au canal
15 de transmission. Il transforme le message numérique en un signal électrique en bande de base ou en un signal à basse fréquence. On peut citer deux grandes classes de codes binaires à signal, les codes de transcodage NRZ (Non Remise à Zéro) et les codes alphabétiques.

- La transformation opérée par un codeur binaire à signal est une
20 opération inversible. Sa fonction de transfert inverse est donc déductible de sa fonction de transfert directe.

- La chaîne d'émission 400 comporte enfin un modulateur 407 qui transforme notamment la séquence binaire en une séquence de symboles
25 modulés. Les symboles sont pris dans un ensemble complexe appelé constellation. Un symbole peut regrouper plusieurs bits. On peut citer par exemple, les modulations de phase numérique ou PSK (Phase Shift Keying) ou les modulations d'amplitude numérique ou QAM (Quadrature Amplitude Modulation).

La transformation opérée par un modulateur est une opération inversible. Sa fonction de transfert inverse est donc déductible de sa fonction de transfert directe.

5 D'après ce qui précède, les transformations appliquées dans la chaîne d'émission sur le train binaire sont réversibles, c'est à dire qu'il est possible à partir du train binaire en sortie de retrouver le train binaire d'entrée. En pratique, de la redondance est même ajoutée de manière à pouvoir retrouver le signal binaire d'origine en présence d'erreurs sur le train binaire codé.

10

Toutefois, toujours d'après ce qui précède, certaines fonctions mises en œuvre par la chaîne d'émission d'un système de communications peuvent ne pas être toujours surjective ni inversibles. Il peut arriver qu'un train binaire codé TBC que l'on voudrait forcer en sortie de module ne corresponde à
15 aucune série de bits « utile » TBU en entrée de module. C'est le cas notamment d'une fonction de codage correcteur en bloc ou convolutif qui ne respecte pas les critères énoncés précédemment ou de toute autre opération de codage non bijective. Dans un tel cas, il n'est pas possible de déterminer la fonction de transfert inverse F^{-1} de la chaîne d'émission globale.

20

Pour contourner cette difficulté, deux variantes de réalisation de l'invention sont envisagées.

Une première variante consiste à rechercher, parmi l'ensemble $F(\{TBU\})$ des séquences modulées qu'il est possible d'obtenir en sortie de
25 chaîne d'émission à partir de l'ensemble $\{TBU\}$ des séquences binaires possibles en entrée de chaîne d'émission, la séquence binaire T' qui minimise la distance entre la transformation modulées en sortie $F(T')$ de la séquence T' et la séquence modulée D recherchée qui contient au moins un motif de synchronisation factice positionné à l'emplacement souhaité. La
30 distance considérée peut, par exemple, être une distance au sens des moindres carrés calculée en intégrant la différence entre une séquence

possible $F(T)$, T appartenant à l'ensemble $\{TBU\}$, et la séquence recherchée D sur un intervalle temporel fixé. Un critère possible peut être calculé à l'aide de la relation suivante :

$$\forall T \in \{TBU\}; C(T, D) = \|F(T) - D\|_{L^2} = \int |F(T)(t) - D(t)|^2 dt$$

- 5 On recherche ensuite la séquence d'entrée $T' \in \{TBU\}$ de la chaîne d'émission qui minimise le critère C : $T' = \underset{T \in \{TBU\}}{\operatorname{Argmin}} [C(T, D)]$.

Toutefois, la mise en œuvre pratique de cette variante doit tenir compte de deux contraintes principales.

- D'une part, pour des raisons de complexité, il n'est pas possible en
 10 pratique d'obtenir l'image $F\{TBU\}$ des signaux modulés en sortie de chaîne d'émission correspondant à l'ensemble de tous les trains binaires utiles possibles (autrement dit, correspondant à l'intégralité de l'ensemble $\{TBU\}$), mais seulement un sous-ensemble restreint de cette image.

- D'autre part, on peut simplifier la mise en œuvre de l'invention en se
 15 basant sur les supports temporels liés principalement aux opérations d'entrelacement, de codage et de mise en trame. En effet, il n'est pas utile en pratique pour la mise en œuvre de l'invention de considérer des trains binaires T utiles dans l'ensemble $\{TBU\}$ dont les signaux modulés $F(T)$ en sortie (après codage et entrelacement notamment) se trouveraient dispersés
 20 sur des intervalles de temps trop longs, ou sur un trop grand nombre de trames en sortie, ou dont la distribution des positions des symboles serait trop lacunaire.

- En pratique donc, cette variante de l'invention peut être mise en œuvre en restreignant l'ensemble $\{TBU\}$ à un sous-ensemble $\{TBU'\}$ qui ne
 25 comprend que des séquences de bits utiles T dont les longueurs et positions après passage dans la chaîne d'émission (et notamment après passage dans les modules de codage et d'entrelacement) correspondent à des motifs factices SYNC'_F , le cas échéant sous optimaux vis à vis du brouillage du réseau de communication, mais compatibles du standard de mise en œuvre

du système de télécommunication en terme de positions, de récurrence et de périodicité trame, et donc facile à générer et à insérer dans les trames de données utiles.

5 La structure temps/fréquence et/ou le choix des positions, longueurs et récurrences des signaux factices SYNC'_F approchant au sens du critère mentionné plus haut les motifs factices SYNC_F désirés en sortie, sont fixés en se basant sur les périodicités trames.

10 L'ensemble restreint $\{\text{TBU}'\}$ est ensuite pré-déterminé en inversant pour les motifs SYNC'_F , analytiquement ou par simulation, les modules de la chaîne d'émission et notamment les modules d'entrelacement et les modules de codage.

15 Ceci permet notamment de simplifier la mise en œuvre de l'invention en concentrant la recherche des séquences binaires T dans l'image inverse (notamment par les transformations de codage et d'entrelacement) d'un ensemble restreint de motifs factices SYNC'_F en sortie, approchant correctement la séquence désirée SYNC_F au sens du critère précédemment
20 mentionné, et correspondant au plus à la durée d'une trame en sortie de chaîne d'émission (et/ou à la durée d'une trame d'information utile en entrée de ladite chaîne d'émission). On périodise les séquences binaires T' ainsi obtenues par trame d'information utile pour générer en sortie des motifs factices eux aussi périodiques et de période indexée sur celle de la trame du
25 signal en sortie.

Le calcul des positions des bits dans la séquence binaire T' pour obtenir un motif factice en sortie de chaîne d'émission peut alors être réalisé une fois pour toutes.

30 Si l'on construit la séquence binaire T' de manière complètement artificielle en exploitant un mode de messagerie qui autorise une

programmation complète du contenu des messages, le calcul des valeurs des bits de la séquence binaire T' peut aussi être réalisé une fois pour toutes.

Si l'on construit la séquence binaire T' à partir d'un applicatif de type codeur audio ou vidéo produisant un contenu variable et aléatoire, le calcul
5 des relations (déterministes) donnant les valeurs des bits de la séquence binaire T' (en fonction du motif factice SYNC'_F recherché et des bits d'information voisins) peut aussi être réalisé une fois pour toutes.

En pratique, pour générer lesdites séquences binaires T , on doit donc
10 forcer la valeur de certains bits utiles, dits bits factices, (situés en entrée du codage canal) à des positions précises.. La transformation appliquée consiste à déterminer tout d'abord un partitionnement de la trame des données utiles en inversant les modules d'entrelacement codage : ceci détermine des positions bien précises, sur lesquelles on choisit
15 artificiellement les valeurs des bits factices de manière à générer les séquences factices après entrelacement et codage. Finalement, dans cette variante simplifiée de mise en œuvre de l'invention,

- les positions que prennent ces bits factices forcés sont déterminées en partant de la position et des valeurs des
20 séquences factices SYNC'_F et en remontant les transformations du train binaire une par une,
- les valeurs que prennent ces bits factices ou les relations donnant les valeurs des bits factices selon les bits d'information voisins sont déterminées elles-aussi en remontant les
25 transformations du train binaire une par une.

Si tous les étages de la chaîne d'émission sont concernés, les étapes clés portent sur l'inversion des opérations d'entrelacement et de codage correcteur.

Restant à l'échelle d'une trame de signal en sortie de chaîne d'émission, les positions, valeurs ou relations donnant les valeurs des bits factices dans le flux d'information utile sont fixes. Elles peuvent être calculées une fois pour toute puis être appliquées sur chaque trame successive.

Une seconde variante de réalisation de l'invention consiste à appliquer l'invention à un sous ensemble situé en aval de la chaîne d'émission constitué de blocs en série dont les fonctions de transfert sont toutes inversibles. En pratique, on identifie la portion de longueur maximale de la chaîne de transmission pour laquelle l'inversion du train binaire peut être réalisée de façon satisfaisante. Pour cela on remonte depuis le signal en sortie de modulateur vers la séquence de données utiles à émettre. Le train binaire destiné à produire des signaux modulés $D(t)$, contenant une séquence de synchronisation factice, n'est injecté qu'au niveau de l'entrée de ce sous-ensemble, autrement dit en sortie de la première fonction non inversible de la chaîne d'émission dans l'ordre séquentiel inverse des fonctions d'émission (allant du modulateur - fin de la chaîne- au codeur correcteur - début de la chaîne). Cette seconde variante de l'invention est par exemple appliquée si la chaîne d'émission comporte un dispositif de chiffrement ou cryptographie dont la fonction présente par nature un inverse non déterminable.

Le procédé selon l'invention peut s'implémenter à partir d'éléments logiciel. Un tel logiciel peut être exécuté par l'équipement émetteur de sorte à modifier la séquence binaire utile à émettre qui est par exemple produite par un applicatif. Il peut également être exécuté par un ordinateur connecté audit émetteur dans le but de le paramétrer.

Le procédé selon l'invention peut être disponible en tant que produit programme d'ordinateur sur un support lisible par ordinateur. Le support peut être électronique, magnétique, optique, électromagnétique ou être un support

de diffusion de type infrarouge. De tels supports sont par exemple, des mémoires à semi-conducteur (Random Access Memory RAM, Read-Only Memory ROM), des bandes, des disquettes ou disques magnétiques ou optiques (Compact Disk – Read Only Memory (CD-ROM), Compact Disk –
5 Read/Write (CD-R/W) and DVD).

REVENDICATIONS

1. Procédé de brouillage d'un système de communication (EM, REC)
5 consistant à générer, dans un signal (S) destiné à être émis par au moins un émetteur (EM_B1, EM_B2,..., EM_B3) compatible dudit système de communication (EM, REC), au moins une séquence factice (SYNC_F), ledit procédé étant caractérisé en ce qu'il comprend les étapes suivantes :
 - Définir ladite séquence factice (SYNC_F) et sa position temporelle
10 et/ou fréquentielle au sein du signal (S) à émettre,
 - Estimer la valeur et la position des bits factices (B_F) à insérer au sein de la séquence de données (D_{U,T}) à émettre en entrée de la chaîne d'émission ou d'une sous-partie de la chaîne d'émission de sorte à obtenir, dans la séquence produite en sortie de la chaîne
15 d'émission, la valeur et la position temporelle prédéfinies des symboles de ladite séquence factice (SYNC_F),
 - Insérer au sein de ladite séquence de données (D_{U,T}), les bits factices (B_F) aux positions obtenues.
- 20 2. Procédé de brouillage selon la revendication 1 dans lequel, lorsque la fonction de transfert F de la chaîne d'émission est inversible, la valeur et la position des bits factices (B_F) sont estimées en déterminant la fonction de transfert inverse F⁻¹ de ladite fonction de transfert F et en appliquant ladite fonction de transfert inverse F⁻¹ à ladite séquence factice (SYNC_F).
25
3. Procédé de brouillage selon la revendication 2 dans lequel la fonction de transfert inverse F⁻¹ de ladite chaîne d'émission est déterminée en effectuant la composition, dans l'ordre inverse, des fonctions de transfert inverses des différents blocs composant ladite chaîne.
- 30 4. Procédé de brouillage selon la revendication 3 dans lequel la chaîne d'émission comporte au moins un code correcteur (402) de rendement

k/n pour lequel l'inversion de sa fonction de transfert est réalisée en résolvant le système d'équations suivant

$$\begin{bmatrix} i_0 & \dots & i_{k-1} \end{bmatrix} = \begin{bmatrix} c_{p_1} & \dots & c_{p_d} \end{bmatrix} \begin{bmatrix} m_{0,p_1} & m_{0,p_2} & m_{0,p_3} & \dots & m_{0,p_d} \\ \dots & \dots & \dots & \dots & \dots \\ m_{k-1,p_1} & m_{k-1,p_2} & m_{k-1,p_3} & \dots & m_{k-1,p_d} \end{bmatrix}^{-1} \quad \text{où } [c_{p_1} \dots c_{p_d}] \text{ sont}$$

les symboles dont la valeur est fixée dans la séquence codée, $[i_0, \dots, i_{k-1}]$ est la séquence à produire en entrée dudit code, et m_{i,p_j} sont les coefficients de la matrice génératrice dudit code, $p_1, p_2, \dots, p_d$ étant l'ensemble des positions des symboles dont la valeur est fixée dans la séquence codée, avec d un entier égal au plus au nombre n de symboles de la séquence codée.

10

- 5 5. Procédé de brouillage selon la revendication 4 dans lequel ledit code correcteur est un code en bloc linéaire ou un code convolutif ou un turbo-code ou un code à faible densité LDPC
- 15 6. Procédé de brouillage selon l'une des revendications 3 à 5 dans lequel la chaîne d'émission comporte en outre un brasseur (403) et/ou un entrelaceur (404) et/ou un module de mise en trame (405) et/ou un codeur binaire à signal (406) et/ou un modulateur (407).
- 20 7. Procédé de brouillage selon la revendication 1 dans lequel, lorsque la fonction de transfert F de la chaîne d'émission est non surjective, la valeur et la position des bits factices sont estimées en recherchant la séquence d'entrée T' de la chaîne d'émission qui minimise un critère de distance entre la séquence $F(T')(t)$ obtenue en sortie de chaîne d'émission lorsque ladite séquence T' est effectivement produite à son
- 25 entrée, et ladite séquence factice ($SYNC_F$).
8. Procédé de brouillage selon la revendication 7 dans lequel, ledit critère de distance est pris égal à l'intégrale, sur une durée donnée, de la norme au

carré de la différence entre la séquence $F(T')(t)$ obtenue en sortie de chaîne d'émission ladite séquence factice ($SYNC_F$).

9. Procédé de brouillage selon l'une des revendications 7 ou 8 dans lequel,
5 la recherche de la séquence d'entrée T' de la chaîne d'émission qui minimise ledit critère de distance est effectué sur un sous-ensemble de l'ensemble des séquences binaires possibles en entrée de la chaîne d'émission.
10. Procédé de brouillage selon l'une des revendications précédentes dans lequel ladite séquence de données (D_U, T) est produite par un applicatif (401), par exemple un codeur audio, image ou vidéo.
11. Procédé de brouillage selon la revendication 1 dans lequel, lorsque la
15 fonction de transfert F de la chaîne d'émission est non surjective, les bits factices (B_F) sont insérés dans la séquence produite en entrée d'une sous-partie de la chaîne d'émission dont la fonction de transfert est surjective et dont la sortie est commune à la sortie de la chaîne d'émission.
- 20 12. Procédé de brouillage selon la revendication 11 dans lequel ladite chaîne d'émission comporte un dispositif de chiffrement et ladite sous-partie de la chaîne d'émission exclut ce dispositif.
- 25 13. Procédé de brouillage selon l'une des revendications précédentes dans lequel les valeurs des symboles de ladite séquence factice ($SYNC_F$) sont identiques à celles des symboles d'une séquence de synchronisation ($SYNC$) que comporte ledit signal pour synchroniser entre eux un récepteur et un émetteur compatibles dudit système de communication.
- 30 14. Procédé de brouillage selon l'une des revendications précédentes dans lequel la position temporelle et/ou fréquentielle des symboles de ladite séquence factice ($SYNC_F$) sont différents de celles des symboles d'une

séquence de synchronisation (SYNC) que comporte ledit signal pour synchroniser entre eux un récepteur et un émetteur compatibles dudit système de communication.

- 5 15. Dispositif d'émission (EM_B1, EM_B2, ..., EM_B3) d'un signal comportant une chaîne d'émission pour transformer une séquence de données (D_U, T) à émettre en un signal (S) à émettre et un moyen adapté pour mettre en œuvre le procédé selon l'une des revendications 1 à 14.
- 10 16. Système de brouillage coopératif comprenant une pluralité de dispositifs d'émission (EM_B1, EM_B2, ..., EM_B3) selon la revendication 15 et pour lesquels les positions temporelles et/ou fréquentielles desdites séquences factices ($SYNC_F$) insérées dans le signal émis par chaque dispositif d'émission sont différentes entre elles.
- 15 17. Système de brouillage coopératif comprenant une pluralité de dispositifs d'émission (EM_B1, EM_B2, ..., EM_B3) selon la revendication 15 et pour lesquels chacun desdits dispositifs d'émission (EM_B1, EM_B2, ..., EM_B3) émet un signal (S) comprenant une pluralité de
- 20 séquences factices ($SYNC_F$) dont les valeurs et/ou positions temporelles et/ou positions fréquentielles sont différentes entre elles.
18. Programme d'ordinateur comportant des instructions pour l'exécution du procédé selon l'une quelconque des revendications 1 à 14, lorsque le
- 25 programme est exécuté par un processeur.
19. Support d'enregistrement lisible par un processeur sur lequel est enregistré un programme comportant des instructions pour l'exécution du procédé selon l'une quelconque des revendications 1 à 14, lorsque le
- 30 programme est exécuté par un processeur.