

**PROCEDE DE LEURRAGE D' UN SYSTEME D' INTERCEPTION ET
DE BROUILLAGE PAR INSERTION DE MOTIFS DE SYNCHRONISATION
FACTICES DANS LE SIGNAL EMIS ETEMETTEUR METTANT
EN OEUVRE LE PROCEDE**

5 Le domaine technique de l'invention est celui de la lutte contre le brouillage de systèmes de télécommunications et l'intrusion au sein de tels systèmes. L'invention vise à fournir une solution permettant d'empêcher l'interception et/ou le brouillage d'un signal émis par un équipement émetteur, tel qu'un téléphone ou un terminal portable.

10 L'invention porte sur un procédé de leurrage d'un système d'interception et de brouillage par l'insertion de motifs de synchronisation factices dans le signal émis par l'équipement émetteur. Les motifs factices insérés sont destinés à perturber l'analyse du signal de l'émetteur par le système d'interception et de brouillage et à disperser à la fois la puissance
15 de calcul de sa fonction d'interception et la puissance du signal de brouillage. Le ou les motif(s) de synchronisation factice(s) sont introduits dans le signal émis en générant une séquence binaire adaptée directement dans la séquence de données binaires à transmettre.

L'invention ne nécessite aucune modification de l'équipement émetteur car
20 elle intervient en entrée de la chaîne d'émission.

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
25 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.

Dans le domaine du brouillage, un objectif est de neutraliser un
30 système de communication en émettant, à la même fréquence radio, un signal de forte puissance et ainsi entraîner une perte de synchronisation au

sein du système de communication. Un système d'interception et de brouillage dispose de fonctions d'interception et d'analyse grâce auxquelles il peut détecter une séquence de synchronisation au sein d'un signal émis par un émetteur, puis brouiller précisément cette séquence de manière
5 cohérente vis à vis d'un récepteur, de sorte que la synchronisation entre l'émetteur et un récepteur compatible de l'émetteur n'est plus possible.

Un problème à résoudre concerne la mise au point d'une solution de leurrage implémentée dans un émetteur de transmission permettant d'empêcher ou rendant plus complexe et/ou moins efficace le brouillage des
10 séquences de synchronisation au sein du signal émis. Un sous-problème associé consiste à concevoir une solution qui soit faiblement complexe, du point de vue de son implémentation, et qui nécessite peu ou pas de modification des équipements émetteurs et/ou récepteurs du système de communication. En particulier, la solution doit permettre de conserver la
15 compatibilité au standard de télécommunications mis en œuvre et de préserver la synchronisation entre émetteurs et récepteurs malgré la présence de brouilleurs.

Différentes méthodes sont connues pour protéger une communication
20 de perturbations extérieures dues à des interférences électromagnétiques ou à un brouillage intentionnel. On peut citer notamment les techniques d'étalement de spectre par séquence directe (en anglais Direct Sequence Spread Spectrum), d'étalement par saut de fréquence (en anglais Frequency Hopping) ou d'étalement par saut temporel (en anglais Time Hopping).

25 En outre, on connaît également des procédés d'anti-brouillage spatiaux par traitement d'antenne qui s'appliquent aux récepteurs et le cas échéant aux émetteurs dotés de réseaux antennaires.

Toutes ces méthodes sont implémentées au niveau des couches basses, notamment de la couche physique, d'un modem et nécessitent ainsi
30 de modifier la forme d'onde ou d'intégrer un traitement supplémentaire sur le signal émis ou reçu. Ainsi ces méthodes nécessitent une conception

spécifique impactant non seulement le logiciel embarqué dans les postes émetteurs et récepteurs mais aussi les composantes matérielles telles que la chaîne radio ou les antennes.

5 L'invention vise à résoudre les problèmes précités et à supprimer les limitations des solutions de l'art antérieur en proposant un procédé de leurrage de systèmes d'interception et de brouillage consistant à réaliser un codage spécifique des données utiles à émettre de sorte à générer indirectement dans le signal émis in fine, un ou plusieurs motifs de
10 synchronisation factices sous forme de motifs stationnaires. La mise en œuvre du procédé selon l'invention au niveau des données utiles à émettre et non en sortie de la chaîne d'émission permet de ne pas altérer les propriétés de l'émetteur vis-à-vis du respect du standard de télécommunications utilisé.

15

L'invention a ainsi pour objet un procédé de leurrage contre un système d'interception et/ou de brouillage, consistant à générer, dans un signal à émettre, au moins une séquence factice de sorte à leurrer ledit système tout en préservant la synchronisation avec un récepteur dudit signal
20 émis, ledit procédé étant caractérisé en ce qu'il comprend les étapes suivantes :

- Définir ladite séquence factice et sa position temporelle et/ou fréquentielle au sein du signal à émettre, les valeurs des symboles de ladite séquence factice et leurs positions temporelles et/ou
25 fréquentielles étant différentes de celles des symboles d'une séquence de synchronisation que comporte ledit signal,
- 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
30 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.

5 Au prix d'une légère perte de débit utile, la mise en œuvre du procédé de leurrage selon l'invention appliquée à un système de communication lui procure une résistance face à un système d'interception et de brouillage sélectif qui va alors s'accrocher et brouiller sélectivement les séquences de synchronisation factices au lieu de s'accrocher et de brouiller sélectivement
10 les motifs de synchronisation réellement utilisées par un récepteur compatible dudit système de communication.

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
15 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
20 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
25 d'équations suivant

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

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 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
5 faible densité LDPC

Selon un aspect particulier de l'invention, la chaîne d'émission 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
10 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 entrée, et ladite séquence factice.

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

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

25 Selon un aspect particulier de l'invention, lorsque la fonction de transfert F de la chaîne d'émission est non surjective, les bits 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 l'invention, ladite chaîne d'émission comporte un dispositif de chiffrement et ladite sous-partie de la chaîne d'émission exclut ce dispositif.

5 Selon un aspect particulier de l'invention, ladite séquence factice est positionnée dans une zone temporelle et/ou fréquentielle non vulnérable du signal à émettre.

Selon un aspect particulier de l'invention, ladite zone non vulnérable est une partie du signal protégée par un code correcteur ou est une zone de
10 bourrage ne contenant aucune information utile.

Selon un aspect particulier de l'invention, lorsque le signal comprenant ladite séquence factice est reçu par ledit récepteur les bits factices sont retirés de la séquence de données.

L'invention a également pour objet un dispositif d'émission d'un signal
15 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.

L'invention a également pour objet un programme d'ordinateur comportant des instructions pour l'exécution du procédé selon l'invention,
20 lorsque le programme est exécuté par un processeur.

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

25

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

- La figure 1, un schéma illustrant le problème du brouillage de
30 séquences de synchronisation dans un signal émis par un émetteur et à destination d'un récepteur,

- La figure 2, un schéma illustrant la solution proposée par l'invention pour résoudre le problème précité,
- Les figures 3a, 3b, deux schémas illustrant la génération, selon l'invention, de motifs de synchronisation factices,
- 5 - La figure 4, 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,
- La figure 5, un schéma des registres à décalage d'un code convolutif de rendement $\frac{1}{2}$,
- 10 - 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 pour imposer, en sortie du code convolutif défini aux figures 5 et 6, les valeurs d'une séquence de bits consécutifs,
- 15 - 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 » ou « symboles utiles » 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 1 illustre, sur un schéma, le problème du brouillage de séquences de synchronisation dans un signal émis par un émetteur et à destination d'un récepteur.

Sur le haut de la figure 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. En effectuant une corrélation de la séquence de synchronisation connue avec le signal reçu, le récepteur REC peut se synchroniser temporellement avec l'émetteur en détectant par exemple le début ou la fin d'une trame, indiqué par la présence de ladite séquence, au sein du signal émis.

Dans le domaine de l'écoute passive, il existe des équipements appelés intercepteurs INT qui sont aptes à intercepter le signal émis par un émetteur EM, à se synchroniser grâce aux séquences de synchronisation SYNC puis à démoduler et/ou analyser le cas échéant les symboles interceptés pour décoder l'information ou reproduire le signal correspondant.

Dans le domaine du brouillage, il existe en outre des brouilleurs BR, couplés aux intercepteurs INT, qui visent à neutraliser le système de communications EM,REC en émettant à la même fréquence radio un signal de forte puissance, compatible, fortement corrélé ou similaire avec les signaux attendus par le récepteur visé, entraînant alors la perte de synchronisation du système. Les systèmes d'interception et de brouillage BR,INT disposent ainsi à la fois de fonctions d'interception grâce auxquelles ils peuvent détecter, identifier et régénérer des séquences de synchronisation au sein d'un signal émis, et de fonctions de brouillage basées sur des émissions de signaux fortement corrélés ou identiques avec le signal attendu par le récepteur cible. En concentrant la puissance de brouillage uniquement sur une séquence de synchronisation SYNC, et, le

cas échéant, en reproduisant la séquence SYNC ou une séquence proche, comme cela est illustré au bas de la figure 1, l'efficacité du brouillage est renforcée car celui-ci concentre sa puissance de brouillage, sur les parties temporelles du signal émis qui sont les plus vulnérables, à savoir les
5 séquences de synchronisation, et pénètre ainsi profondément dans la chaîne de réception du récepteur REC.

L'exemple des séquences de synchronisation n'est pas limitatif et peut être étendu à toute séquence de symboles connus fixés à priori par le standard de télécommunications ou la technologie d'accès radio et qui,
10 lorsqu'elles sont brouillées, entraînent un dysfonctionnement du système. En particulier, de telles séquences englobent également les séquences d'égalisation et les motifs de synchronisation des trames au sein du signal. Ces séquences, selon les cas, précèdent ou sont multiplexées aux informations à transmettre. Elles ont le plus souvent une combinatoire limitée
15 en position et en valeur ce qui augmente leur vulnérabilité à l'interception et au brouillage.

La figure 2 illustre, sur un schéma similaire à celui de la figure 1, le principe à la base de l'invention permettant de leurrer un système
20 d'interception INT et/ou un brouilleur BR afin de préserver la synchronisation au sein du système de communication EM, REC.

Selon l'invention, un ou plusieurs motifs de synchronisation factices $SYNC_F$ sont introduits au sein du signal émis par l'émetteur EM. Ces séquences factices présentent des caractéristiques stationnaires qui sont
25 facilement détectables par un intercepteur. Elles sont insérées périodiquement ou selon un motif temporel prédéterminé au sein du signal à émettre. Par exemple, elles peuvent être insérées avec une période identique à celle des motifs de synchronisation réels SYNC présent dans le signal mais avec un décalage temporel et/ou un décalage fréquentiel de
30 sorte que les séquences de synchronisation factices ne remplacent pas, en tout ou partie, les séquences de synchronisation réelles qui doivent être

préservées pour assurer le bon fonctionnement du système de communication. La ou les séquences factices SYNC_F peuvent être avantageusement positionnées sur les parties les moins vulnérables ou les mieux protégées du signal, par exemple sur des parties protégées par un
5 code correcteur robuste ou sur des fréquences ou des intervalles de temps de bourrage vides de données utiles et non exploitées pour la prise de synchronisation ou l'accès au réseau, ou encore sur des fréquences ou des intervalles de temps vides de données utiles et dédiées à cet effet.

Les valeurs et positions des symboles des séquences de
10 synchronisation factices SYNC_F sont définies par des valeurs et/ou des positions différentes de celles utilisées par le récepteur légitime pour les besoins propres à l'établissement de la radiocommunication avec l'émetteur, afin d'éviter toute confusion avec les séquences utilisées par le récepteur légitime. L'expression récepteur légitime désigne ici un récepteur REC
15 compatible du système de communication et apte à communiquer avec un émetteur EM dudit système.

Les valeurs et caractéristiques de périodicité des séquences de synchronisation factices SYNC_F sont préférentiellement choisies parmi des motifs proches de ceux employés dans le domaine des télécommunications
20 et faciles à identifier pour un intercepteur. Il peut par exemple s'agir de motifs proches de ceux couramment destinés à la synchronisation fréquentielle constitués de série de symboles identiques « 0000... » ou « 1111... ». Il peut également s'agir de motifs proches de ceux couramment destinés à la synchronisation temporelle constitués de série de symboles alternés
25 « 010101... ». Il peut également s'agir de séquences de symboles proches de celles exploitées dans certains standards civils connus exploitant les mêmes gammes de fréquence que l'émetteur. Dans tous les cas, les valeurs et positions des symboles des séquences de synchronisation factices doivent cependant être différentes de celles employées par le système de
30 communication pour assurer la synchronisation entre un émetteur et un récepteur.

L'insertion de telles séquences de synchronisation factices ne perturbe pas la synchronisation entre les équipements émetteur EM et récepteur REC car les séquences factices choisies sont suffisamment
5 décorréliées des séquences de synchronisation réelles utilisées au sein du système de communication. Les valeurs et les positions, dans le domaine temps-fréquence, des symboles des séquences factices sont choisies de telles sorte qu'elles sont différentes des valeurs et des positions des séquences de synchronisation réelles. Ainsi, les séquences factices peuvent
10 être traitées par un récepteur du système de communication soit comme des données aléatoires comparables à des données transmises, soit comme des leurres dont les valeurs et positions sont connues à priori et qui sont donc aisément identifiable pour être éliminées au cours du processus de synchronisation.

15

L'insertion de séquences de synchronisation factices dans le signal a pour but de leurrer un système d'interception et de brouillage INT,BR. La capacité d'identification et d'analyse du signal par le brouilleur est altérée par l'existence de ces séquences factices que le brouilleur a tendance à détecter
20 et à identifier comme étant des séquences de synchronisation réelles. La capacité de calcul de la fonction interception du brouilleur est sollicitée sur des séquences non significatives du signal, le signal de brouillage aura tendance à être émis sur les zones temporelles du signal correspondant aux séquences factices alors que les séquences de synchronisation réelles sont
25 préservées. Les techniques d'identification orientée que la fonction d'interception pourrait mettre en œuvre sont perturbées par l'apparition des signaux factices, et plus encore si ceux-ci sont produits en grande quantité et sont variables d'une communication à une autre ou gérés avec des mécanismes de fluctuation à long terme.

30 Les performances d'un système d'interception et de brouillage sont d'autant plus diminuées qu'un nombre important de motifs de

synchronisation factices sont présents dans le signal émis. En effet, le système d'analyse du brouilleur est obligé de disperser ses efforts de calcul, jusqu'à saturation éventuelle de ses capacités de calcul, sur des motifs artificiels autres que le motif de synchronisation réel puisqu'il n'a pas de
5 moyen de le reconnaître à priori. Même si le système d'analyse du brouilleur travaille à l'aide d'une base de données de signaux à identifier (selon une approche d'analyse et d'identification orientées), la mise en œuvre de l'invention aura tendance à charger artificiellement ladite base de donnée, à disperser les efforts d'analyse et de recherches orientées et à pénaliser
10 l'identification, surtout si les motifs factices émis artificiellement sont rendus intentionnellement fluctuants selon le temps.

L'invention permet ainsi de leurrer un brouilleur intelligent mais également d'empêcher la synchronisation du signal émis par tout équipement récepteur ou intercepteur qui n'a pas la connaissance du motif
15 de synchronisation réel.

Insérer un motif de synchronisation factice au sein d'une trame de symboles modulés en sortie de la chaîne d'émission, conduit à modifier la chaîne d'émission elle-même et donc l'émetteur lui-même ce qui n'est pas
20 toujours techniquement possible d'une part, ou souhaitable du point de vue de la complexité de mise en œuvre.

L'invention vise à permettre l'introduction de motifs de synchronisation factices en intervenant en entrée de la chaîne d'émission c'est-à-dire directement sur les données binaires à émettre, sans modification de la
25 chaîne d'émission, et par le biais de moyens logiciels.

Les figures 3a, 3b illustrent le procédé de génération de séquences factices selon l'invention.

La figure 3a illustre schématiquement la transformation subie par une
30 séquence de données binaires utiles D_U à transmettre pour obtenir une séquence de symboles S_T modulés, prêts à être émis sous la forme 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
5 séquence de synchronisation SYNC ou d'une séquence équivalente composée de symboles connus de tous les équipements du système de communication.

La figure 3b illustre la mise en œuvre du procédé selon l'invention.

Dans un premier temps, l'emplacement et la constitution de la
10 séquence de synchronisation factice $SYNC_F$ sont choisis dans une trame vide T_F , de même taille qu'une trame réelle de symboles modulés S_T et sur un canal de fréquence donnée f . La position temporelle et/ou le canal en fréquence choisis, sont différents de la position temporelle et/ou du canal fréquentiel d'une séquence de synchronisation réelle. La taille et la nature
15 exacte de la séquence factice peuvent être variables.

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 de
20 ladite séquence factice $SYNC_F$. 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 comprenant les bits factices B_F .

25 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, en décalant ou en multiplexant les bits utiles correspondants à la séquence de données réelle. La séquence de symboles modulés obtenus en sortie de la chaîne de transmission comprend à la fois les symboles utiles
30 S_U , le motif de synchronisation réel SYNC et le motif de synchronisation factice $SYNC_F$.

Les bits factices peuvent ainsi être introduits directement dans la séquence de données à transmettre et ce sans modifier ni effectuer d'intrusion dans la chaîne de transmission de l'équipement émetteur. Si les données à transmettre proviennent d'une application, par exemple un codeur

5 source audio ou vidéo, il est possible d'intercepter les données binaires applicatives avant leur entrée dans la chaîne de transmission qui est mise en œuvre au niveau de la couche physique d'un modem. Cette interception peut se faire au niveau d'une couche intermédiaire, par exemple au niveau de la couche réseau.

10 A la réception du signal par le récepteur REC, les bits factices sont supprimés de la séquence de données démodulée et décodée.

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.

15 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

20 $f_1, f_2, \dots, f_N$ 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.

30 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.

- 5 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. C'est l'opération réalisée par le récepteur.

- 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
 10 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 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
 15 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 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 $F(\{\text{TBU}\})$ de
 25 l'ensemble $\{\text{TBU}\}$ 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 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
 30 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.

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
5 chaque transformation, on détermine les entrées qui doivent être appliquées pour obtenir en sortie le train binaire codé voulu.

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

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
15 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
20 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.

25

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
30 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
5 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
10 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
15 les codes LDPC.

Codes linéaires en blocs

Un code correcteur linéaire en blocs de rendement k/n transforme une
20 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 $i = (i_0$
25 $, \dots, i_{k-1})$ 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 existantes comprenant n bits, 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

Dans le cas où le code est systématique à gauche, la séquence codée s'écrit $[i_0 \quad \dots \quad i_{k-1} \quad c_k \quad \dots \quad c_{n-1}]$. 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 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 est exactement de même pour un code en bloc systématique à droite.

Cas de codage en bloc issus de codes cycliques

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 $[c_0 \dots c_{n-1}]$ est un mot de code, toute permutation circulaire du mot $[c_i \ c_{i+1} \dots c_{n-1} \ c_0 \dots c_{i-1}]$ est aussi un mot de code.

En écrivant sous forme polynômiale les mots de code

10 $c(x) = c_0 + c_1 \cdot x + c_2 \cdot x^2 + \dots + c_{n-1} \cdot x^{n-1}$, 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

15 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 $v(x)$ (de degré inférieur ou égal à $n-k-1$) et le quotient de la division est $k(x)$. On a donc $i(x) \cdot x^{n-k} = k(x) \cdot g(x) + v(x)$, et $c(x) = i(x) \cdot x^{n-k} + v(x) = k(x) \cdot g(x)$.

Comme $v(x)$ est de degré inférieur ou égal à $n-k-1$, les valeurs des

20 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, toute

25 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

30 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

5

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 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 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 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 codes.

- 5 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 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
- 10 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
- 15 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
- 20 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
- 25 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
 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.

- 5 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
 droite).

- Par contre, on peut remarquer que si on considère la deuxième, la
 10 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 :

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

- En effet, le rang d'une matrice correspond au nombre de colonnes
 15 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
 20 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 k bits
 25 consécutifs lorsque la sous-matrice de la matrice génératrice du code
 associée aux positions des bits à fixer est de rang plein.

On précise ci-dessous 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.
- 5 - 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 (qui sont a priori aléatoires car liées à l'information utile).

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
- 10 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 :

$$\begin{aligned} f_0 &= c_0 + i_1 + i_2 \\ 15 \quad f_1 &= c_0 + c_1 + i_2 \end{aligned}$$

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.

20 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
- 25 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 au symbole a_j sont définis par la relation de convolution suivante, où $g_{e,j,i}$ sont les coefficients de
- 5 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 $g_{e,j,i}$, $e = 0, \dots, m$, $j=0 \dots k-1$, $i=0 \dots n-1$) :

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

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

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

- 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
- 20 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.

- 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
- 25 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.

Exemple non limitatif d'un code convolutif de rendement $1/2$

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 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 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 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 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 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.

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
5 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
10 $n=2$), ou plus généralement décalées de n bits d'une ligne à l'autre lorsque le 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
15 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
20 codeur si et seulement si la sous-matrice M correspondant aux colonnes des 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
25 bits codés. Ces relations –déterministes- caractérisent de manière 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
30 des autres. Le code précédemment considéré a des relations de parité de 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
5 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.

10 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.

15 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
20 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 est illustré à la figure 8 sur laquelle sont représentés les indices 800 des bits
25 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.

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.

5 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.

15 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.

20 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

30

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
5 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 ces 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
10 procédure de décodage est itérative et collaborative entre chaque code 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
15 extrinsèque) qui est multiplexée à l'information en entrée des autres codeurs. 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 (d'où la dénomination « turbo » qui est liée à la procédure de décodage et non au code proprement dit).

20 Les codes simples employables sont multiples.

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
25 chaque code simple permet de démultiplier le pouvoir détecteur et correcteur 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
30 dimensions, le code produit de rendement $\frac{k_1 \cdot k_2}{n_1 \cdot n_2}$ est construit à partir de

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

5 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

10 décomposer en des opérations de codage et d'entrelacement.

Nous allons illustrer la décomposition d'un code produit 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

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

15 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.
- Un entrelacement ligne/colonne simple : les bits sont écrits

20 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.
- Un entrelacement ligne/colonne simple : les bits sont écrits

25 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
5 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 $2 \cdot k_1$.

10

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
15 intervenir une relation de parité de poids faible. Il s'agit en pratique de codes 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,
20 notamment, de générer des motifs choisis avec des séries de bits consécutifs de 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
25 matrice génératrice pour obtenir la séquence codée, il est possible de fixer la 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 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
5 cela est le cas, il suffit de résoudre le système d'équations

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

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.

10 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
15 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.

20 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.

25 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 synchrone

5

Dans le cas d'un embrouilleur synchrone, une séquence pseudo aléatoire est ajoutée modulo 2 au signal binaire à embrouiller. La série binaire $\{\dots, e_k, e_{k+1}, \dots\}$ représentant la séquence d'embrouillage est périodique, de période L longue. Les séquences les plus couramment
 10 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 $E(X) = 1 + c_1X + \dots + c_PX^P$ de degré P . On obtient une séquence de période $L = 2^P - 1$. Les séquences utilisées pour l'embrouillage peuvent cependant être
 15 tronquées. La k -ième sortie e_k du registre à décalage vérifie la relation de récurrence $e_k = c_1 \cdot e_{k-1} + \dots + c_P \cdot e_{k-P}$, 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
 20 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
 25 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
 30 période doivent coïncider avant d'appliquer la transformation $F^{-1} = F$ aux bits embrouillés.

Cas d'un embrouilleur auto-synchronisant

Dans le cas d'un embrouilleur auto-synchronisant, les états du registre sont
 5 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 $E(X) = 1 + c_1.X + \dots +$
 10 $c_P.X^P$ 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 $b'_k = b_k + c_1. b'_{k-1} + \dots + c_P. b'_{k-P}$ (« + » désignant toujours
 l'addition modulo 2 dans $GF(2)$). Ceci permet d'effectuer le désembrouillage
 en réception de façon simple et sans avoir besoin de se synchroniser au
 15 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 $b_k = b'_k + c_1. b'_{k-1} +$
 $\dots + c_P. b'_{k-P}$.

La difficulté pour forcer les bits en sortie provient ici de ce que
 l'expression des états du registre en fonction des données d'entrées b_k
 20 (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
 indispensable d'avoir accès de façon synchrone aux données embrouillées
 25 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.

La chaîne d'émission 400 peut également comporter un module
 30 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 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 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 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 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 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 de 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 de trames correspondant à des motifs d'ordonnement 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 trames peuvent également être regroupées pour former une multi-trame ou
5 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.

La mise en trame limite la possibilité de forcer le signal modulé voulu
10 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.

15 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
20 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
25 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
30 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 modulés. Les symboles sont pris dans un ensemble complexe appelé constellation. Un symbole peut regrouper plusieurs bits. On peut citer par
5 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
10 inversible. Sa fonction de transfert inverse est donc déductible de sa fonction de transfert directe.

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
15 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.

Toutefois, toujours d'après ce qui précède, certaines fonctions mises
20 en œuvre par la chaîne d'émission d'un système de communications peuvent ne pas être toujours surjectives ni inversibles. Il peut arriver qu'un train binaire codé TBC que l'on voudrait forcer en sortie de module ne corresponde à aucune série de bits « utile » TBU en entrée de module. C'est le cas notamment pour une fonction de codage correcteur en bloc ou
25 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.

30 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 chaîne d'émission à partir de l'ensemble $\{TBU\}$ des séquences binaires possibles en entrée de chaîne d'émission, la séquence binaire T' qui

5 minimise la distance entre la transformation modulée en sortie $F(T')$ de la séquence T' et la séquence modulée D recherchée qui contient au moins un motif de synchronisation factice $SYNC_F$ positionné à l'emplacement souhaité. La distance considérée peut, par exemple, être une distance au sens des moindres carrés calculée en intégrant la différence entre une séquence

10 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 :

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

On recherche ensuite la séquence d'entrée $T' \in \{TBU\}$ de la chaîne

15 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

20 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

25 basant sur les supports temporels principalement liés 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

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 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 leurrage d'un système d'interception et brouillage, 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.

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 signaux factices SYNC_F désirés en sortie, sont fixés en se basant sur les périodicités trames de la transmission et en éloignant autant que possible (dans le domaine temps/fréquence) les motifs factices des données véritablement utiles à la synchronisation et à la démodulation du récepteur. Par exemple, dans une transmission de type TDMA (Time Division Multiple Access) employant des signaux transmis sous forme de paquets ou bursts en anglais avec ou sans évansion de fréquence, on placera les motifs factices à des positions différentes de celles des motifs réels, en fin de paquet plutôt qu'en début de paquet, sur des fréquences balises fictives plutôt que sur celles véritablement employées. Dans une transmission multiplexant plusieurs flux utilisateur par des codes orthogonaux, on utilisera comme motifs factices des codes orthogonaux à l'ensemble des codes effectivement employés par les utilisateurs.

L'ensemble restreint {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.

5

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
10 correctement la séquence factice désirée SYNC_F au sens du critère précédemment 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
15 des motifs factices eux aussi périodiques et de période indexée sur celle de la trame du signal en sortie. Le calcul des positions des bits de la séquence binaire T' pour un motif factice en sortie peut alors être réalisé une fois pour toutes, le calcul de la dépendance des bits de la séquence binaire T' aux bits d'information voisins peut aussi être réalisé une fois pour toutes.

20

En pratique, pour générer lesdites séquences binaires T, on doit donc forcer la valeur de certains bits utiles (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
25 les modules d'entrelacement codage : ceci détermine des positions bien précises, sur lesquelles, au lieu de transmettre des données utiles, on remplace les bits d'information utiles par des bits utiles forcés servant à générer les séquences factices après entrelacement et codage. Par contre, ces positions étant déterminées, les valeurs de ces bits utiles factices
30 peuvent changer selon la valeur des bits d'information utiles voisins (variables et aléatoires) : la valeur de chacun de ces bits utiles forcés dépend

non seulement des valeurs des bits factices codés désirés mais aussi des valeurs de certains bits utiles voisins (selon le schéma d'embrouillage et de codage). Ces dépendances sont linéaires ou affines, ceci signifie que la valeur de chaque bit utile forcé s'exprime comme la somme de bits factices en sortie de codage et de bits d'information utiles voisins (i.e. dépendance
5 linéaire aux bits d'information utiles plus éventuellement une inversion des bits forcés codés).

Finalement, dans cette variante simplifiée de mise en œuvre de l'invention,

- 10 - les positions que prennent ces bits utiles forcés sont déterminées en partant de la position et des valeurs des séquences factices SYNC'_F et en remontant les transformations du train binaire une par une,
- les relations de dépendance linéaires entre les valeurs que
15 prennent ces bits utiles forcés et les bits d'information utiles voisins (variables et aléatoires) sont déterminées elles aussi en remontant les transformations du train binaire une par une.

Si tous les étages de la chaîne d'émission sont concernés, les étapes
20 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 et relations de dépendances des bits utiles factices dans le flux d'information utile sont fixes. Ces positions et relations de
25 dépendance peuvent être calculées une fois pour toute puis être appliquées pour le calcul et le positionnement des bits forcés sur chaque trame successive.

30 A la réception, on récupère trame par trame les données démodulées et décodées. Les bits utiles factices qui ont servi à générer les séquences

factices après codage sont de position connue a priori du récepteur qui les retire directement du train de bits d'informations démodulé/décodé.

Une seconde variante de réalisation de l'invention consiste à appliquer
5 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
10 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
15 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.

20

Le procédé selon l'invention peut se décliner avec des séquences
factice ($SYNC_F$) présentant des combinaisons significatives et des
périodicités ou des variations lentes, de manière à maintenir au cours du
temps la combinatoire d'analyse et à pénaliser la capacité d'identification et
25 de stockage de la fonction « interception » du système d'interception et de
brouillage leurré.

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
30 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
5 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 –
10 Read/Write (CD-R/W) and DVD).

L'invention trouve application pour tout système de communication que l'on souhaite rendre plus robuste à l'interception et au brouillage, en particulier, l'invention s'applique au rétrofit de systèmes de communication
15 de génération ancienne devenus de ce fait vulnérables aux intercepteurs et aux brouilleurs modernes.

REVENDICATIONS

1. Procédé de leurrage contre un système d'interception et/ou de brouillage,
5 consistant à générer, dans un signal à émettre, au moins une séquence factice (SYNC_F) de sorte à leurrer ledit système tout en préservant la synchronisation avec un récepteur dudit signal émis, 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 à émettre, les valeurs des symboles de ladite séquence factice et leurs positions temporelles et/ou fréquentielles étant différentes de celles des symboles d'une séquence de synchronisation (SYNC) que comporte ledit signal,
 - Estimer la valeur et la position des bits factices (B_F) à insérer au
15 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 d'émission, la valeur et la position temporelle prédéfinies des symboles de ladite séquence factice (SYNC_F),
 - 20 - Insérer, au sein de ladite séquence de données ($D_{U,T}$), les bits factices (B_F) aux positions obtenues.
2. Procédé de leurrage selon la revendication 1 dans lequel, lorsque la
fonction de transfert F de la chaîne d'émission est inversible, la valeur et
25 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).
3. Procédé de leurrage selon la revendication 2 dans lequel la fonction de
30 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.

4. Procédé de leurrage 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

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

- 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.

15

5. Procédé de leurrage 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
6. Procédé de leurrage 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).
7. Procédé de leurrage 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

25

d'émission lorsque ladite séquence T' est effectivement produite à son entrée, et ladite séquence factice ($SYNC_F$).

8. Procédé de leurrage 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 et ladite séquence factice ($SYNC_F$).
9. Procédé de leurrage selon l'une des revendications 7 ou 8 dans lequel, la recherche de la séquence d'entrée T' de la chaîne d'émission qui minimise ledit critère de distance est effectuée sur un sous-ensemble de l'ensemble des séquences binaires possibles en entrée de la chaîne d'émission.
10. Procédé de leurrage 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 leurrage selon la revendication 1 dans lequel, lorsque la 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.
12. Procédé de leurrage 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.
13. Procédé de leurrage selon l'une des revendications précédentes dans lequel ladite séquence factice ($SYNC_F$) est positionnée dans une zone temporelle et/ou fréquentielle non vulnérable du signal à émettre.

14. Procédé de leurrage selon la revendication 13 dans lequel ladite zone non vulnérable est une partie du signal protégée par un code correcteur ou est une zone de bourrage ne contenant aucune information utile.
- 5 15. Procédé de leurrage selon l'une des revendications précédentes dans lequel lorsque le signal comprenant ladite séquence factice ($SYNC_F$) est reçu par ledit récepteur les bits factices (B_F) sont retirés de la séquence de données (D_U, T).
- 10 16. Dispositif d'émission d'un signal comportant une chaîne d'émission pour transformer une séquence de données (D_U, T) à émettre en un signal à émettre et un moyen adapté pour mettre en œuvre le procédé selon l'une des revendications 1 à 15.
- 15 17. Programme d'ordinateur comportant des instructions pour l'exécution du procédé selon l'une quelconque des revendications 1 à 15, lorsque le programme est exécuté par un processeur.
- 20 18. 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 15, lorsque le programme est exécuté par un processeur.