N° d'ordre:1110

THESE

L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE

pour obtenir le titre de:

DOCTEUR DE L'UNIVERSITE

spécialité : Electronique

par

Abdelbasset JBARA

METHODE D'OBTENTION ET D'OPTIMISATION DES PRINCIPAUX PARAMETRES GARANTISSANT LA SECURITE D'UNE TRANSMISSION NUMERIQUE EN SE FIXANT L'OBJECTIF DE SECURITE

Applications aux systèmes de transports.

Soutenue le 30 Mars 1993 devant la Commission d'Examen

Membres du jury :	MM. R.GABILLARD,	Président, Directeur de Thèse
	J. FONTAINE,	Rapporteur
	Y. DAVID,	Rapporteur
	M. HEDDEBAUT,	Examinateur
	M. KLINGLER,	Examinateur
	Y. NGUYEN,	Examinateur

REMERCIEMENTS

Les travaux qui font l'objet de cette thèse ont été réalisés au Laboratoire de Radiopropagation et Electronique (LRPE) de l'Université des Sciences et Technologies de Lille.

Je remercie vivement Monsieur le Professeur R. GABILLARD pour l'opportunité qu'il m'a offerte en me proposant un sujet aussi passionnant que celui de la sécurité de la transmission numérique, qu'il trouve ici ma très profonde reconnaissance pour son soutien, ses encouragements, ses précieux conseils, sa confiance qu'il m'a toujours témoignée et pour l'honneur qu'il me fait en présidant le jury.

Je remercie également Monsieur le Professeur J. FONTAINE de l'Université BLAISE PASCAL (CLERMONT II) et ainsi que Monsieur Y. DAVID Directeur de l'INRETS-CRESTA pour avoir accepté de juger mon travail.

Je remercie Monsieur M. HEDDEBAUT Directeur de Recherche à l'INRETS-CRESTA et Monsieur M. KLINGLER Chargé de Recherches à l'INRETS-CRESTA qui ont bien voulu examiner ce travail et participer au jury.

Je remercie Monsieur Y. NGUYEN, Maître de conférences à l'USTL, pour avoir suivi les travaux d'essais et pour avoir accepté d'examiner mon travail et de participer au jury.

Je remercie Monsieur le Professeur F. LOUAGE, Directeur de l'Ecole Nationale Supérieure d'Arts et Métiers (ENSAM).

Je remercie Monsieur C. SEMET, Ingénieur de recherche au Laboratoire de Radiopropagation et Electronique à l'USTL qui a toujours porté une attention toute particulière à nos travaux depuis des années et pour ses précieux conseils.

Qu'il me soit permis d'adresser mes remerciements:

à Madame C. CHEROUTE pour les nombreux services administratifs qu'elle m'a rendus.

à Monsieur J.P. DEHORTER pour la reproduction du manuscrit.

à tous les membres du laboratoire qui par leur présence m'ont permis de mener à bien ce travail.

SOMMAIRE

Pages

INTRODUCTION

INTRODUCTION

Depuis plusieurs années notre laboratoire étudie les problèmes de mise en sécurité de systèmes automatisés de transport et a développé d'une part, des principes de fonctionnement des automatismes garantissant par eux-mêmes un haut niveau de sécurité et d'autre part des méthodes de contrôle des dispositifs de sécurité proprement dites.

Le travail que nous présentons porte sur le problème du codage des informations sécuritaires à transmettre entre le poste central de commande et le véhicule. Ces informations, très peu nombreuses, puisqu'elles ne portent que sur des ordres très spécifiques de mise en sécurité du système, doivent être reçues avec une certitude quasi-absolue. Cette certitude appelée objectif de sécurité (OS) est définie par le cahier des charges et sa valeur doit être suffisamment faible (10^{-8} à 10^{-12}) pour garantir la sécurité des passagers.

La réalisation de la transmission proprement dite doit être performante pour que les perturbations n'engendrent pas un taux moyen d'erreur supérieur à 10^{-3} ou 10^{-2}. Ces erreurs se manifestent d'ailleurs généralement par des paquets qui viennent perturber la transmission pendant de courts instants.

La méthode que nous avons mise au point permet dès l'instant où l'on a défini l'objectif de sécurité (OS) et la loi de probabilité d'erreurs sur la transmission (obtenue par l'étude de la voie de transmission) de calculer le nombre de bits de contrôle qu'il est nécessaire d'associer aux bits d'information. On obtient également la distance minimale à respecter entre chacun des mots codes. Connaissant cette distance minimale, on choisira le codage qui répond effectivement à cette condition de distance.

Le modèle numérique est établi à partir de deux conditions:

a) La probabilité de non détection d'erreur (PND) doit être inférieure à l'objectif de sécurité (OS):

$$PND \leq OS$$

b) La deuxième condition est la marge inférieure de Hamming.

Les deux conditions peuvent être développées et se mettre sous une forme analytique.

La résolution graphique de ces deux conditions permet d'obtenir les valeurs du nombre de bits de contrôle "k" et de la distance minimale à respecter "d".

On obtient donc, sans faire aucune hypothèse sur la nature du code, les paramètres caractéristiques "k" et "d" du code à réaliser.

Nous avons vérifié, dans le cas d'un code particulier, que les hypothèses majorantes et simplificatrices étaient valables. Nous avons d'ailleurs constaté que la marge entre l'objectif de sécurité et la valeur réelle de PND pouvait être importante. Cela est dû, bien sûr, d'une part aux majorations prises en compte et d'autre part à l'expression entière de d et k.

Nous proposons une méthode qui permet, dans le cas d'un code géneré par un polynôme, de réduire le nombre de bits de contrôle tout en conservant l'objectif de sécurité imposé.

L'ensemble du travail présenté aborde les deux hypothèses de forme d'erreurs: erreurs indépendantes et paquets d'erreurs.

Nous avons établi dans le chapitre deux une méthode générale de définition des caractéristiques du code et vérifié sa validité dans le cas particulier d'un code généré par un polynôme.

Nous avons défini systématiquement le polynôme générateur qui donnera la marge de sécurité maximale.

Le chapitre trois est consacré à l'application de cette étude aux systèmes de transport guidé (Trains ou Métro).

Nous présentons, dans le dernier chapitre, une méthode d'évaluation de la qualité d'une transmission numérique (taux et type d'erreurs) par calculateur dans le cas d'un débit faible et par automate programmable dans le cas d'un haut débit numérique.

Notre objectif dans ce travail est de montrer que la méthode présentée permet de définir systématiquement les caractéristiques que doit présenter un code quelconque pour garantir une transmission d'information en sécurité avec un niveau de certitude imposé.

CHAPITRE I

RAPPELS THEORIQUES ET GENERAUX SUR LES CODES DE DETECTION D'ERREURS.

I INTRODUCTION.

Les canaux de transmission, électromagnétiques ou électriques, sont affectés par des parasites et des perturbations de toutes natures susceptibles de modifier les signaux transmis et en particulier quand il s'agit de transmissions hertziennes dans un environnement industriel ou urbain. Les orages, les alimentations électriques, les moteurs.... créent par exemple des parasites impulsionnels, souvent très brefs et qui occupent un large spectre de fréquences. Les récepteurs sélectifs doivent être soigneusement calculés afin de ne pas être saturés par les impulsions parasites et ne pas engendrer de phénomènes d'oscillations qui rendraient impossible tout traitement ultérieur des signaux. Ce problème est bien connu des électroniciens qui sont amenés à concevoir et à réaliser des récepteurs à bas niveau.

Si nous supposons que ce problème est résolu, il convient néanmoins de tenir compte du fait que certains parasites impulsionnels ne sont pas éliminés par les dispositifs du filtrage et qu'ils vont se retrouver dans les étages suivants et être interprétés par les décodeurs comme s'il s'agissait d'informations proprement dites. Ces perturbations engendreront, suivant leur durée par rapport à celles des bits d'information deux types d'erreurs:

Erreurs indépendantes: ce sera le cas où l'on suppose que chaque bit transmis est affecté de manière indépendante par les perturbations. Nous dirons à ce moment là que la probabilité d'erreur est constante.

Paquets d'erreurs: si les perturbations ont une durée plus longue que la durée d'un bit, les erreurs apparaîtront groupées, on dit alors qu'il se présente "un paquet d'erreurs".

On définit un paquet d'erreurs de la façon suivante:

-deux séquences en erreurs correspondent à des paquets d'erreurs séparés s'il existe au moins entre ces deux paquets 10 bits binaires consécutifs non erronés.

-pour une séquence d'erreurs correspondant à un seul paquet la longueur du paquet "l" est le nombre binaire qui constitue cette séquence.

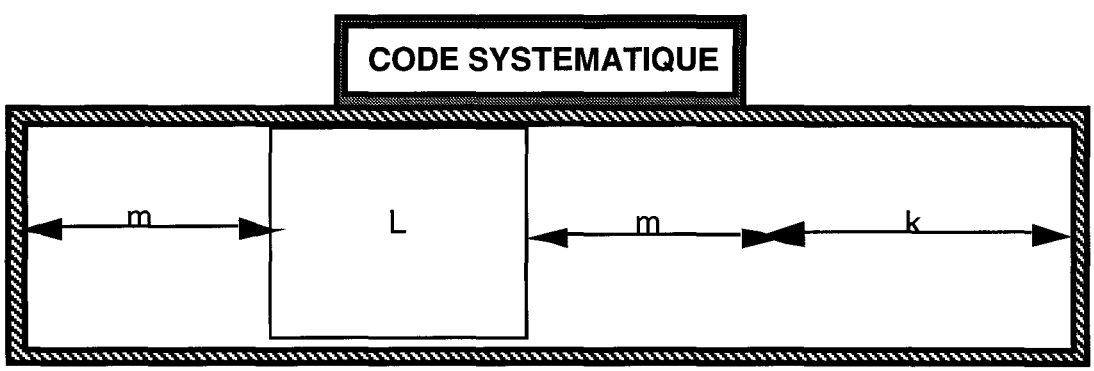
-à l'intérieur d'un paquet d'erreurs, il peut exister un ou plusieurs bits non erronés.

Dans tous les cas, ces erreurs sont inacceptables quand il s'agit de traiter des informations touchant à la sécurité de fonctionnement du système.

II PRINCIPE GENERAL D'UN CODE DE DETECTION D'ERREURS. (Réf: 5)

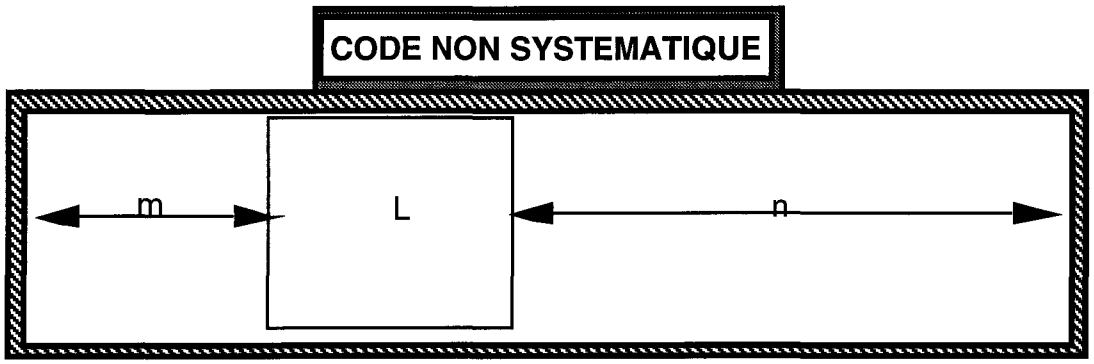
Le principe général de toute méthode de codage consiste à ajouter aux informations à transmettre des informations supplémentaires qui permettent de détecter et éventuellement de corriger les erreurs qui apparaîtront au cours de la transmission.

Cette information supplémentaire se présente sous forme de bits de contrôle associés aux bits d'information suivant une loi L connue du récepteur et de l'émetteur. Chaque message ou mot code sera formé de "m" bits d'information et de "k" bits de contrôle et aura donc une longueur n=m+k.



m: nombre de bits d'informations
k: nombre de bits de contrôle
n=m+k: nombre de bits du mot code
L: loi mathématique qui détermine les bits de contrôle en fonction des bits d'information.

Les bits d'informations sont groupés ensemble et les bits de contrôle également.



m: nombre de bits d'informations

n: nombre de bits du mot code

L: loi mathématique qui détermine le mot code en fonction des bits d'information.

Ce principe de détection d'erreurs n'est pas absolu. En effet un parasite peut très bien transformer un mot code en un autre mot code.

III MECANISME GENERAL DE DETECTION D'ERREURS. (Réf: 5)

A la réception de chaque mot code, le récepteur vérifie que celui-ci correspond à la loi L du codage. Si la vérification est réalisée cela veut dire que le mot reçu appartient à l'ensemble des mots susceptibles d'être émis. Mais nous n'avons pas la garantie absolue de la correspondance bi_univoque avec le mot émis puisqu'il est tout à fait possible que plusieurs perturbations transforment un mot code en un autre mot code.

Pour exprimer les capacités de détection ou de correction d'erreurs d'un code, nous utilisons la notion classique de distance entre les mots codes (nombre de bits par lesquels les mots diffèrent entre eux) et nous appellerons pour un code donné "**d**" la distance minimale. On sait que l'on peut alors:

-détecter **t** erreurs avec **t=d-1**

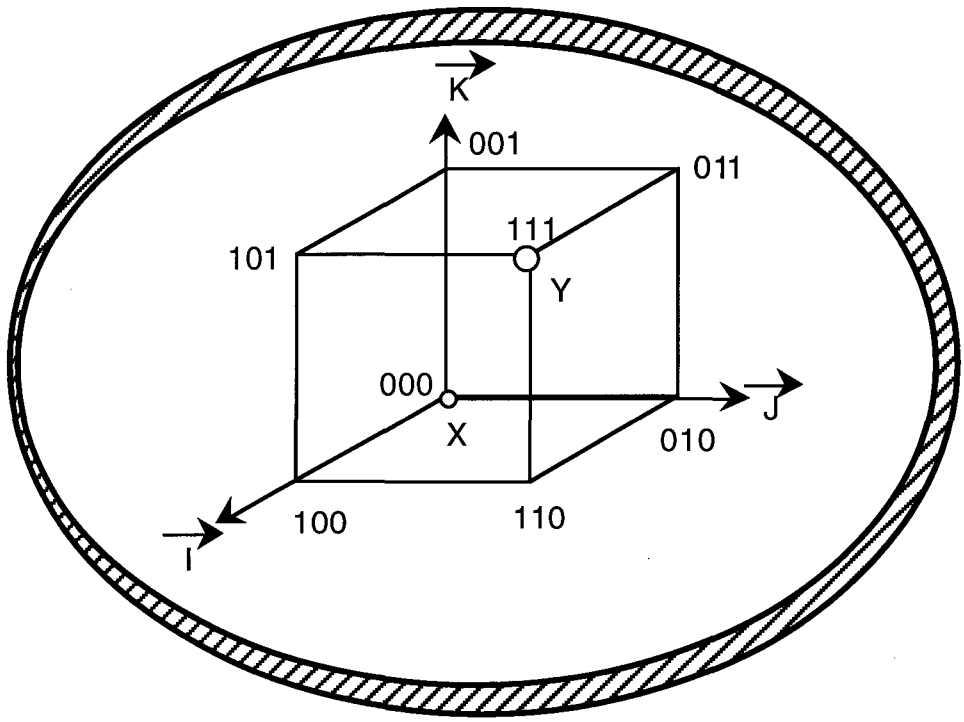
-corriger $\mathbf{c_o}$ erreurs avec $\mathbf{c_o}$=entier**(d/2 -1)**

IV DISTANCE ENTRE LES MOTS CODES. (Réf: 1)

La détermination de la distance "**d**" est un point fondamental dans le processus d'élaboration d'un code. Il est nécessaire d'en calculer la valeur optimale afin de ne pas allonger outre mesure la longueur des mots, ce qui entraine en particulier des problèmes de débit d'informations.

Si l'on appelle x_1, x_2, x_3 les coordonnées du point **X**, et y_1, y_2, y_3, celles du point **Y**, la distance géométrique entre les deux points codes sera donnée par la formule classique:

$$(XY)^2 = (x_1-y_1)^2 + (x_2-y_2)^2 + (x_3-y_3)^2$$



Comme les x_i et les y_i ne peuvent prendre que les valeurs **0** ou **1**, $(XY)^2$ sera un nombre entier égal au nombre de parenthèses dans lesquelles x_i sera différent de y_i.

Ce nombre est donc aussi le nombre de digits par lesquels les deux mots codes diffèrent. Ainsi s'explique la raison de l'expression

"distance" que l'on appelle aussi " distance de Hamming" du nom de son inventeur. En fait, il ne s'agit pas d'une distance mais du carré d'une distance.

Cette notion se généralise aisément à des codes formés de blocs de **n** digits en considérant un hypercube dans un espace à **n** dimensions.

V CODES LINEAIRES DE DETECTION D' ERREURS.(Réf: 2)

V-1 Définitions.

Si la loi **L** de formation du code se réduit à des combinaisons linéaires permettant de déterminer à partir des positions et des valeurs (0 ou 1) des bits d'information **m**, la valeur des bits à placer en position de contrôle **k**, on aura un code linéaire **L(n,m)**.

V-2 Représentation matricielle.

Un message binaire non codé "$\mathbf{M_i}$" composé de m symboles $\mathbf{a_i}$ pouvant prendre les valeurs (0 ou 1) est représenté sous la forme matricielle:

$$<M_i>=<a_0 \ a_1 \ \ldots . \ a_{m-1}>$$

$<M_i>$: est appelée la matrice des symboles d'information utile ou vecteur des symboles d'information.

De même, un mot code $\mathbf{N_i}$ formé de n symboles binaires a_i est représenté sous la forme matricielle:

$$<N_i>=<a_o \ldots . a_{m-1} \ldots . a_{n-1}>.$$

V-3 Matrice génératrice d'un code linéaire de détection d'erreurs.

D'après la définition d'un code linéaire, il existe une relation entre l'ensemble de tous les mots d'information utiles M_i et l'ensemble des mots du code N_i. Cette correspondance peut être établie en définissant un opérateur **g** tel que l'on ait:

$$g\ (M_i) = N_i\ .$$

Sa structure matricielle est la suivante:

$$<M_i>(G)_{m,n} = <N_i> \qquad (1)$$

Donc un code linéaire peut être représenté par une matrice de **m** lignes et **n** colonnes. Cette matrice est appelée "matrice génératrice" du code d'ordre **(m,n)**.

Remarque: Si le code est systématique, c'est à dire si les bits d'informations, sont groupés ensemble et les bits de contrôle également, alors les mots codes s'expriment par:

$$<N_i> = <M_i>(G)_{m,n}$$
$$= <\underbrace{a_0 \ldots\ldots a_{m-1}}_{\mathbf{m}\text{ bits d'information}} \quad \underbrace{c_0 \ldots\ldots\ldots c_{k-1}}_{\mathbf{k}\text{ bits de contrôle}}>$$

avec: $(G)_{m,n}=(I_{m,m}P_{m,k})$.

où $I_{m,n}$ est la matrice unité d'ordre **(m,m)**
et $P_{m,k}$ est une matrice d'ordre **(m,k)**.

V-4 Matrice de contrôle d'un code de détection d'erreurs.

Soit $<N_i>$ un mot code transmis sur un canal de transmission. A la réception, on reçoit un mot $<N_i'>$.

Ce mot peut être défini par: $<N_i'>=<N_i>+<E_i>$.

$<Ei>$ étant une matrice appelée "vecteur d'erreur" représentant les erreurs introduites par le canal de transmission.

Le rôle d'un code de détection d'erreur est de déterminer si $<E_i>=0$ ou si $<E_i>\neq 0$, c'est à dire si le mot reçu $<N_i'>$ est bien le mot émis $<N_i>$. Mais en fait, la seule certitude qu'il soit possible d'obtenir est celle que $<N_i'>$ est un mot du code. Ceci peut signifier soit que $<E_i>=0$ (il n'y a pas eu d'erreur de transmission), ou bien

que les erreurs ($<E_i>\neq 0$) qui se sont produites ont réussi à transformer un mot du code en un autre mot code.

Pour déceler si le mot reçu $<N_i'>$ est un mot du code, on calcule à partir des mots reçus et de la loi de codage **L** (connue du récepteur), un vecteur indicateur d'erreur $<S_i>$ appelé "syndrome". Ceci s'effectue à l'aide d'un opérateur H défini par:

$$H\ (N_i')=S_i$$

La structure matricielle de cet opérateur est:

$$<N_i'>\ (H)^t=<S_i> \qquad (2)$$

$<S_i>$ est le syndrome d'erreurs comportant **m** bits

$(H)^t$ est la matrice transposée de **(H)**.

La matrice **(H)** est appelée la matrice de contrôle ou matrice de vérification formée de **k** lignes et **n** colonnes.

Remarques:

-si le code est systématique alors:

$$(H)_{k,n} =((P_{m,k})^t\ I_{kk})$$

$I_{k,k}$ est la matrice unité

et

$(P_{m,k})^t$ est la matrice transposée de $P_{m,k}$.

-on montre facilement que:

$$(G).(H)^t=(0)_{m,k}$$

$(O)_{m,k}$ étant la matrice nulle.

V-5 Mécanisme de détection d'un code linéaire de détection d'erreurs.

Lors d'une transmission d'un mot $<N_i>$, le canal de transmission introduit un bruit caractérisé par:

$$<E_i>=<e_0 \cdots e_j \cdots e_n>$$

avec:

$e_j=0$ si le j^{iéme} bit est transmis correctement

$e_j=1$ si le j^{iéme} bit est changé par le bruit du canal

$<E_i>$ est le vecteur d'erreur

Le mot reçu est alors: $<N_i'>=<N_i>+<E_i>$.

A la réception, le décodeur peut calculer le "syndrome" qui est défini par l'équation:

$$<S_i>=<N_i'>(H)^t$$

Deux cas peuvent se présenter:

-Le syndrome $<S_i>=0$ alors le mot reçu a la configuration d'un mot du code, donc aucune erreur n'est décelée. Ceci peut signifier qu'effectivement aucune erreur ne s'est produite ou bien que les erreurs du canal de transmission ont transformé le mot code émis en un autre mot code.

-Le syndrome $<S_i>\neq 0$, $<N_i'>$ n'est pas un mot du code, donc le canal a introduit des erreurs $<E_i>\neq 0$. Dans ce cas, le décodeur signale une détection d'erreur.

V-6 Procédure de construction d'un code linéaire L(n,m).

L'expression (2) peut s'écrire:

$$<S_i>=<N_i'>(H)^t \Rightarrow <S_i>^t=(H)<N_i'>^t.$$

La matrice (H) peut s'écrire sous la forme:

$$(H)=(h_0 . . h_i . . h_{n-1})$$

où h_i représente une colonne de la matrice **(H).**

Théorème: La distance minimale d'un code linéaire est le poids minimal d'un mot du code non nul.

Remarque: Pour la définition des concepts de "distance" et de "poids, nous invitons le lecteur à se reporter à l'annexe III.

Soit $<N_i>$ le mot code de poids minimal d, alors:

$$(H)<N_i>^t = \begin{pmatrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix}$$

$$\sum_{i=0}^{i=n} a_i h_i = 0 \Rightarrow h_{i1}+h_{i2}+. . . +h_{id} = 0$$

$$h_{id} = h_{i1} + ... + h_{i(d-1)} \neq 0 .$$

C'est la relation d'indépendance linéaire de **(d-1)** colonnes de la matrice **(H).**

Théorème: pour détecter t erreurs, il faut que la distance minimale entre les mots du code satisfasse l'équation:

$$d=t+1.$$

Tenant compte de cette condition, les vecteurs colonnes de la matrice (H) devront satisfaire les relations suivantes:

$$h_{i0} \# 0 \qquad \text{pour } i_0 = 0, 1 \quad . . ., \quad (n-1)$$

$$h_{i0} + h_{i1} \neq 0 \qquad \text{pour } i_0, i_1 = 0,1,2,\ldots,(n-1) \text{ et } i_0 \neq i_1$$

$$\ldots\ldots\ldots$$

$$h_{i0} + h_{i1} + \ldots h_{i(d-2)} \neq 0.$$

pour $i_0, i_1 \ldots, i_{(d-2)} = 0, \ldots, (n-1)$

et $i_0, i_1 \ldots, i_{(d-2)}$ distincts

V-7 Codes linéaires particuliers.

Parmi les codes linéaires, on distingue:

- -les codes de Hamming
- -les codes de Reed-Muller
- -les codes de Mac Donald
- -les codes dérivés des matrices d'Hadamard

Des exemples de codes linéaires sont exposés en annexe I.

VI CODES CYCLIQUES DE DETECTION D'ERREURS (Réf: 4, 5).

VI-1 Définition.

Un code cyclique **C(n,m)** est un code linéaire possédant la propriété suivante:

Toute permutation cyclique d'un mot code est aussi un mot code.

Soit $<N_o>$ un mot code: $<N_o> = <a_o\ a_1\ a_2 \ldots . a_{n-1}>$.

Alors tous les mots de la forme:

$$<N_i> = <a_i\ a_{i+1} \quad a_{n-1}\ a_o \ldots a_{i-1}>$$

sont aussi des mots codes.

VI-2 Représentation polynomiale des codes cycliques.

Nous avons représenté les mots des codes linéaires par des vecteurs.

Une autre possibilité est de considérer les mots codes comme des éléments de l'ensemble des polynômes binaires de degré **(n-1).** Un mot code $<N_o>=<a_o\ a_1 \ldots a_{n-1}>$ est représenté par le polynôme binaire suivant:

$$N_o(x)= a_o + a_1 x + \ldots\ldots + a_{n-1} x^{n-1}$$
$$\textbf{modulo } x^n + 1$$

Cette représentation à l'aide des polynômes est très intéressante car les opérations à effectuer sur les polynômes sont simples.

Bien entendu, la multiplication des polynômes donnera en général un polynôme de degré supérieur à **(n-1)**, mais il est facile de le réduire modulo un polynôme binaire de degré **n.**

Ainsi la définition du code cyclique peut être la suivante:

Soit un mot du code représenté par le polynôme **N(x)**, si **xN(x)** appartient aussi au code, alors celui-ci est un code cyclique.

En effet, la multiplication du polynôme **N(x)** par x^i modulo (x^n+1) correspond à **i** permutations successives des bits de **N(x)** et par conséquent, $x^iN(x)$ modulo (x^n+1) représente un mot appartenant au code.

VI-3 Polynôme générateur.

Rappelons que la matrice génératrice **(G)** d'un code linéaire permet de déterminer tous les mots codes.

Pour un code cyclique, le calcul de la matrice génératrice **(G)** se ramène à la détermination d'un polynôme générateur **g(x)** de degré **k=n-m**, vérifiant les propriétés suivantes:

Propriété 1: Tout mot d'un code cyclique **C(n,m)** est un multiple du polynôme générateur **g(x)** de degré **k**.

Propriété 2: Le polynôme générateur d'un code cyclique **C(n,m)** est un polynôme qui divise (x^n+1).

Remarque: Comme les codes linéaires, les codes cycliques contiennent (2^m-1) mots codes et chaque mot code est formé de **n** symboles **(n bits)**. **m** symboles sont réservés à l'information utile et **k=n-m** sont réservés au contrôle.

Recherche des polynômes générateurs:

le polynôme générateur d'un code **C(n,m)** est un diviseur de (x^n+1). La recherche des polynômes générateurs susceptibles d'engendrer un code **C(n,m)** commence donc par la décomposition de (x^n+1) en produits de polynômes irréductibles de la forme:

$$(x^n+1) = m_1(x) \cdot m_2(x) \ldots\ldots m_t(x)$$

Tous les polynômes irréductibles $m_1(x)$, $m_2(x)$ . . . etc sont capables d'engendrer des codes cycliques **C(n,m)**.

On trouvera la définition d'un polynôme irréductible, en annexe II.

VI-4 Mise en oeuvre des codes cycliques.

VI-4-1 Codage.

Lorsque l'on se donne m bits qui constituent l'information utile à transmettre, on peut former un mot:

$$<M>=<a_0 \ldots\ldots a_{m-1}>$$

Ce mot est représentable par un polynôme:

$$M(x) = a_0 x^0 + a_1 x^1 + \ldots\ldots + a_{m-1} x^{m-1}$$ de degré **m-1.**

On obtient le polynôme du code en effectuant le produit

$$M(x).g(x)=N(x).$$

Cette procédure produit un code non systématique.

Pour obtenir un code cyclique systématique:

1) on multiplie x^k par **M(x)** ce qui donne $x^kM(x)$.
2) on divise $x^kM(x)$ par **g(x)** et on utilise le reste, changé de signe comme symbole de contrôle.

On peut toujours écrire: $x^kM(x)=Q(X)g(x)+C(X)$

C(x) est un polynôme de degré inférieur à **k.**

Par conséquent, le polynôme **N(x)** de degré **n=m+k** défini par:

$$N(x)=x^kM(x)-C(X)=Q(x)g(x)$$

est bien divisible par **g(x)** et, par suite, est un polynôme du code.

Les **m** premiers symboles sont des bits d'information utiles et les **k=n-m** symboles suivants sont des bits de contrôle.

VI-4-2 Décodage ou mécanisme de détection d'erreurs.

Comme dans le cas des codes linéaires, on calcule à la réception le syndrome d'erreurs du message reçu.

Soit **N'(x)** le message reçu:

$$N'(x)=N(x)+E(x)$$

E(x) étant le polynôme d'erreurs introduit par le canal de transmission.

A la réception, le décodeur calcule le "syndrome" défini par:

$$S(x)=\text{Reste de } (N'(x)/g(x))$$

Si le syndrome n'est pas nul, il y a eu erreur et on a $E(x)\neq 0$. Le décodeur signale alors une détection d'erreurs.

Si le syndrome est nul, on a la configuration d'un mot du code, donc aucune erreur n'est décelée. Il n'y a donc pas eu d'erreur, ou bien, les erreurs qui se sont produites ont transformé le mot code émis en un autre mot code.

VI-5 CODES CYCLIQUES PARTICULIERS.

-les codes B.C.H (Bose- Chaudhuri-Hocquenghem).
-les codes de Reed-Solomon à structure binaire.
-les codes de Fire.
-les codes d'Abranson.

Pour un exemple de code cyclique, voir annexe II.

CHAPITRE II

MISE EN EVIDENCE DES CONDITIONS A RESPECTER POUR OBTENIR UNE SECURITE DE TRANSMISSION PREDETERMINEE

I DEFINITION D'UN OBJECTIF DE SECURITE.

Dans le cas particulier de la transmission d'information en sécurité, il faut également définir un seuil limite supérieur de probabilité d'erreur non détectée, ce qui permet de trouver une solution acceptable. Le concept de sécurité probabiliste d'un système conduit donc à la définition d'un objectif de sécurité que nous appelons **(OS)**. La valeur **(OS)** est la probabilité d'erreur jugée suffisamment faible pour être considérée comme "acceptable". Il faut donc que le code que l'on va construire garantisse une probabilité de non détection d'erreur **(PND)** telle que:

$$(PND) \leq (OS) \qquad (1)$$

(PND) étant défini par la proportion de messages finalement faux après épuisement de la procédure de détection. La valeur de **PND** est reliée au taux d'erreur brut (τ) qui dépend de la probabilité d'erreur par bit **(P)** et du nombre de bits du message **(n)** ainsi que de l'efficacité **(E)** du code:

$$PND = \tau\,(1-E) + \tau^2\,E(1-E) + \tau^3\,E^2(1-E) \qquad (2)$$

Si la condition (1) est réalisée, la sécurité probabiliste du système sera assurée.

II HYPOTHESE DE TRAVAIL PRISE EN CONSIDERATION.

La construction d'un code particulier repondant au concept de sécurité probabiliste doit donc s'appuyer sur les hypothèses suivantes:

a) connaissance du nombre de messages **(N)** à transmettre permettant de définir le nombre de bits d'information:

$$N < 2^m - 1$$

b) connaissance de la valeur de l'objectif de sécurité **(OS)** qui satisfait l'exploitation du système.

c) connaissance des erreurs introduites dans la transmission en terme de probabilité.

Dans les applications que nous allons développer nous supposerons que le nombre de messages est relativement faible puisqu'il s'agit uniquement de transmettre des ordres de sécurité au système. En ce qui concerne les erreurs nous considérerons, d'une part que la probabilité d'erreur par bit **(P)** est relativement faible puisque toutes les précautions doivent être prises pour protéger la voie de transmission, et d'autre part que deux cas peuvent se présenter: ou bien il s'agit d'erreurs indépendantes, ou bien il s'agit de paquets d'erreurs. Nous aborderons ces deux cas séparément.

III CAS DES ERREURS INDEPENDANTES. (Réf: 1, 2)

III-1 Détermination des conditions à respecter.

Nous considérons que les erreurs introduites par la transmission se manifestent de façon indépendante sur chacun des bits et nous appelons **"P"** cette probabilité. Le calcul du nombre de bits de contrôle **"k"** (et par conséquent la longueur **"n"** des messages) et la détermination de la distance minimale **"d"** seront effectués en respectant les deux conditions nécessaires et suffisantes suivantes:

1ère condition: Un parasite unique, pouvant agir sur n'importe quel bit, engendrera un nombre moyen de messages faux égal à:

$$C_n^1 P^1 (1-P)^{n-1}$$

Si les parasites modifient **"i"** bits ($1 \leq i \leq n$) du message, le nombre moyen de messages faux sera:

$$C_n^i P^i (1-P)^{n-i}$$

Et donc le nombre moyen total de messages faux sera donné par l'expression:

$$\sum_{i=1}^{n} C_n^i \; P^i \; (1-P)^{n-i}$$

Parmi l'ensemble des messages faux un certain nombre sera détecté par le décodeur. Par contre les messages dont le nombre de bits erronés sera supérieur ou égal à la valeur de **"d"** ne seront pas forcement détectés. La proportion de ces messages s'exprime par:

$$\sum_{i=d}^{n} C_n^i P^i (1-P)^{n-i}$$

On a donc:

$$PND \leq \sum_{i=d}^{n} C_n^i P^i (1-P)^{n-i} = PND'$$

Pour satisfaire l'objectif de sécurité, il faut que:

$$\sum_{i=d}^{n} C_n^i P^i (1-P)^{n-i} \leq (OS) \qquad (3)$$

2ème condition: Pour que le code soit réalisable il faut respecter la marge inférieure de Hamming. La correction de $\mathbf{c_o}$ erreurs entraine la relation suivante:

$$\sum_{i=0}^{c_0} C_n^i \leq 2^{n-m} \qquad (4)$$

$\mathbf{c_o}$ étant la partie entière de:

$$\left(\frac{d}{2} - 1\right)$$

La condition de faisabilité est une relation à satisfaire entre les paramètres **n,m et d** pour qu'il soit possible de trouver un code répondant aux spécifications quelque soit le type de code que l'on envisage de mettre en oeuvre. Cette condition est nécessaire pour parvenir à trouver un code de détection d'erreur capable de corriger c_o erreurs.

Les relations **(3)** et **(4)** nous permettent de déterminer la valeur de la distance minimale **"d"** et le nombre de bits de contrôle **"k".**

III-2 Méthode de détermination des valeurs de "n" et "d".

Pour exprimer les conditions **(3)** et **(4)**, dans le but de conserver la généralité du calcul, il est intéressant d'introduire une variable indépendante:

$$x=\frac{d}{2n}$$

Il est évident que l'intervalle de variation de **x** est compris entre 0 et 0,5 puisque **"d"** ne peut être supérieur au nombre total de bits **"n".**

Une étude de la condition (3), **(ref. [1], [2])** nous a permis de la mettre sous la forme:

$$\frac{|\log(OS)|}{n} \leq E(x,P) \qquad (5)$$

Où **E(x,P)** représente une fonction de l'entropie **H** du canal de transmission:

E(x,P)=H(P)-H(2x)+(2x-P)(log(1-P)-log(P))
H(x) =-x.log(x)-(1-x).log(1-x)
(log: logarithme de base 10)

La relation (5) est obtenue en prenant en compte les approximations suivantes:

a) taux d'erreur par bit très faible: **P<<1**

b) développement de n! par la formule de Stirling:

$$n! = n^n \; e^{-n} \; \sqrt{2\pi n} \; .$$

Et en considérant la relation majorante :

$$n^n \; d^{-d} \; (n-d)^{-(n-d)} \; \rangle \; C_n^d \; .$$

Par ailleurs, la condition de faisabilité (4) est respectée si:

$$H_2 (x) \; \leq \; (1-\frac{m}{n}) \; . \qquad (6)$$

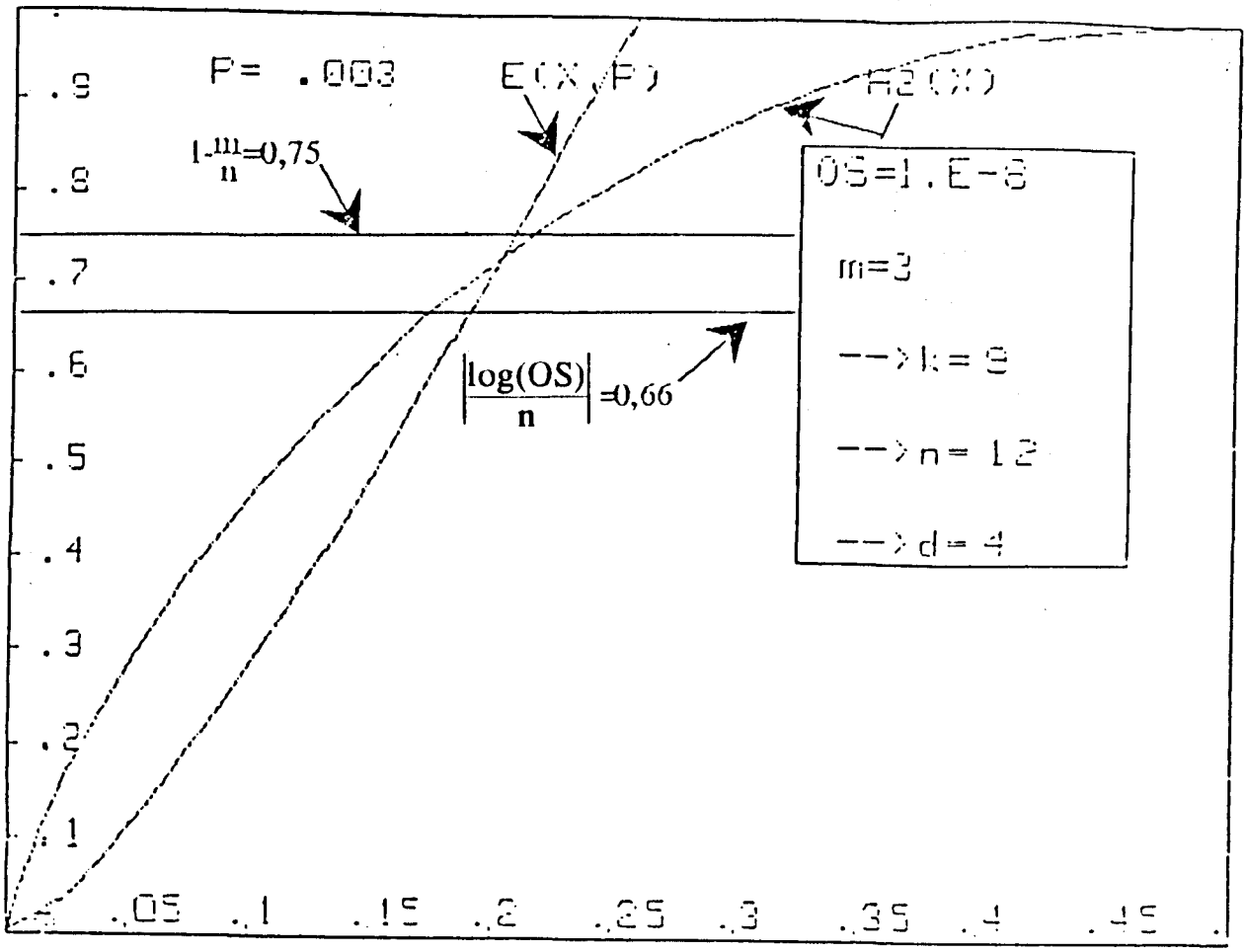
avec :

$$H_2 (x) = \frac{(-x\ln(x)-(1-x)\ln(1-x))}{\ln(2)}$$

ln: logarithme neperien

Pour résoudre le système formé par les deux relations (5) et (6), on peut tracer graphiquement (graphe1) les deux fonctions de **x**, **E(x,P)** et $\mathbf{H_2(x)}$ le point d'intersection donne la valeur de **x** et par conséquent la valeur de **"d"**:

$$d=Ent\,|2 \; . \; n \; . \; x \,|$$



graphe1

On obtient également la valeur de "**k**" puisque pour cette valeur particulière de **x**, **E(x,P)** et $H_2(x)$ sont égales et les relations (5) et (6) permettent d'écrire:

$$\left| \frac{\log(OS)}{n} \right| \leq 1-\frac{m}{n}$$

Soit encore

$$k \geq |\log(OS)|$$

III-3 Vérification de la méthode dans le cas d'un code obtenu à l'aide d'un polynôme générateur.

La méthode que nous venons de présenter est valable, dans le cadre des quelques approximations introduites, quelque soit la technique de codage mise en oeuvre. Nous allons vérifier que, dans le cas particulier des codes obtenus par un polynôme générateur, elle permet bien de définir celui qui engendrera un code répondant à l'objectif de sécurité exigé.

Les conditions que nous imposerons sont les suivantes:

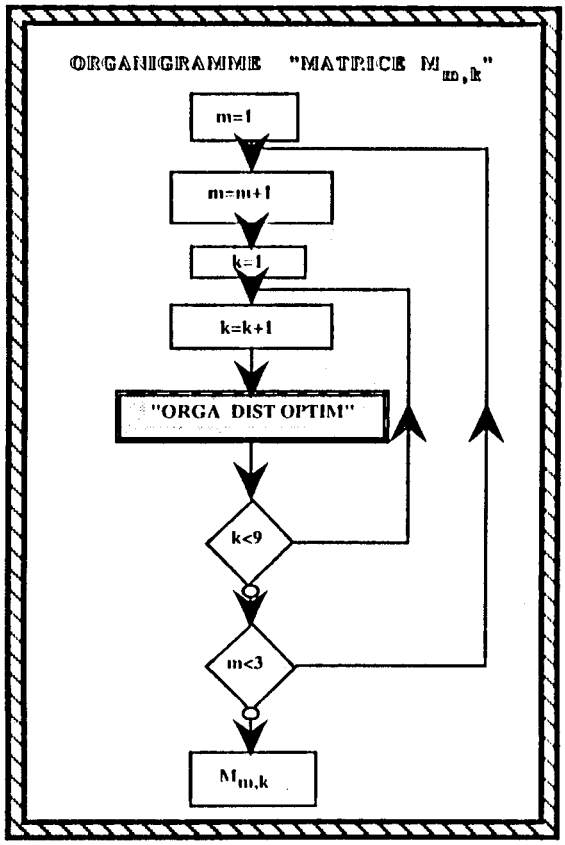
-nombre de messages à transmettre = 7 soit: **m= 3**
-probabilité d'erreur par bit transmis : **P=0,003**
-objectif de sécurité exigé : **$OS=10^{-8}$**

Nous pouvons tracer le graphique défini au **(III-2)** et nous obtenons les caractéristiques du code à réaliser:

distance minimale : **d=4**
nombre de bits de contrôle : **k=9**

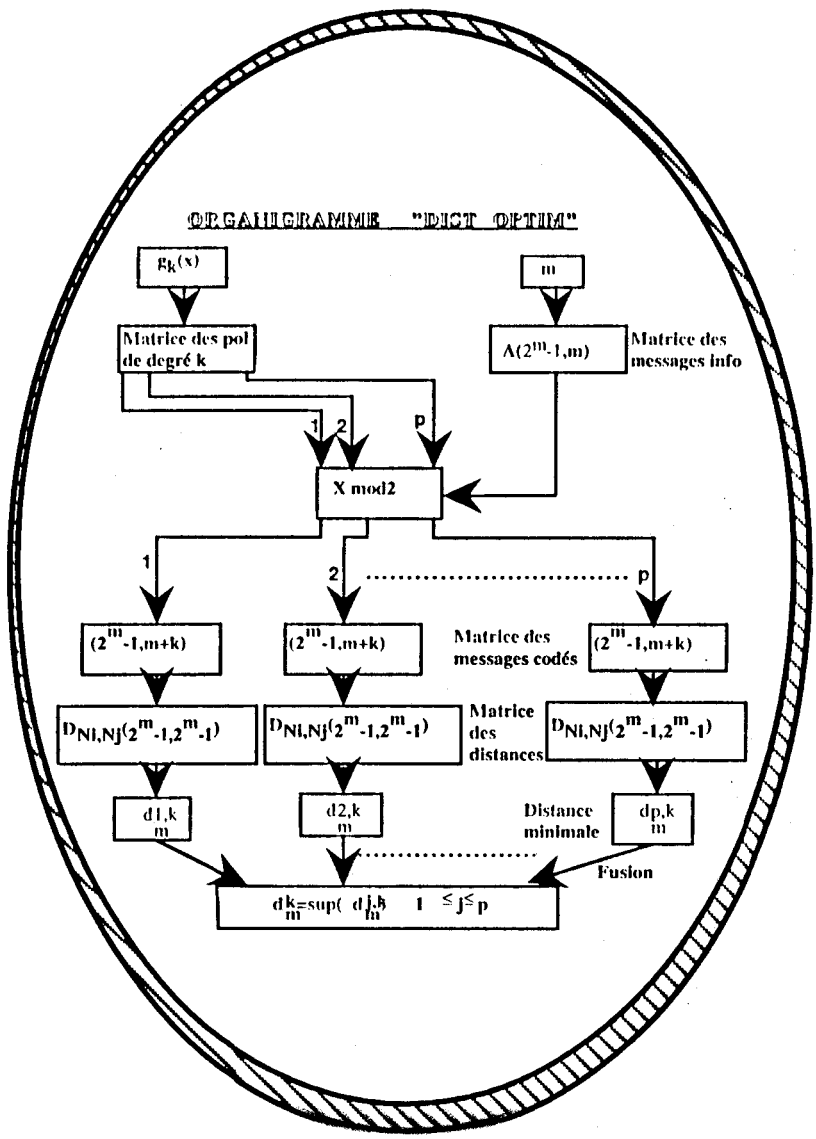
Les mots du code seront obtenus à partir d'un polynôme générateur de degré **k=9.** Ces polynômes sont nombreux **(réf. [5])** et nous pouvons rechercher, dans l'ensemble de ces polynômes, ceux qui nous donnent la plus faible probabilité de non détection d'erreur **(PND).**

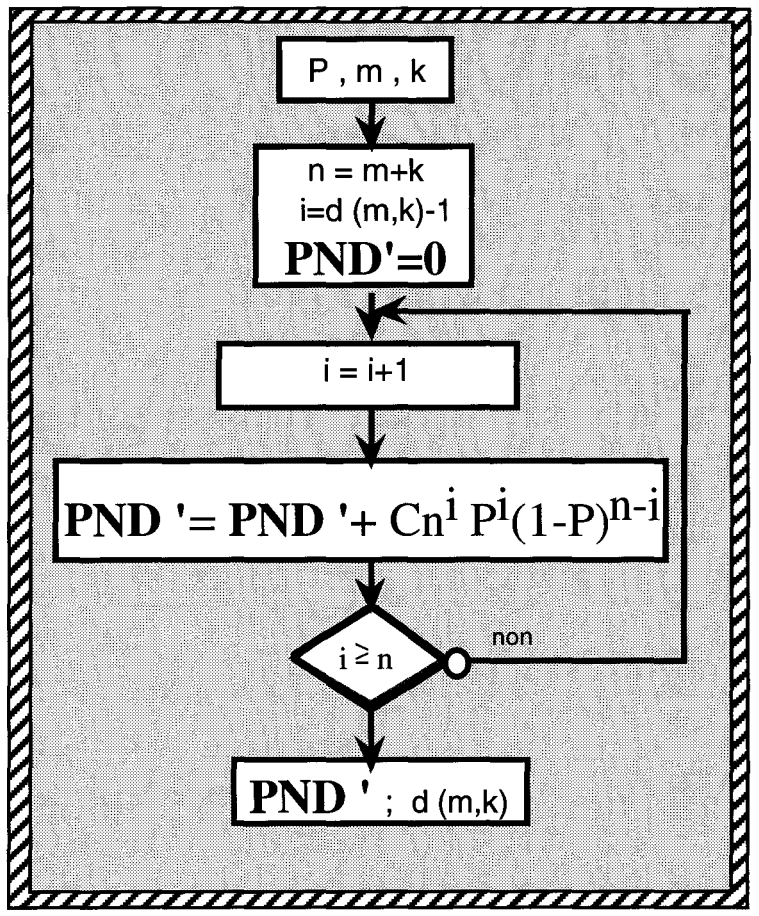
L'algorithme simple, schématisé ci-aprés, donne en introduisant les valeurs de **P, m, k,** la valeur de **PND'** ainsi que celle de la distance minimale entre les mots du code.



MATRICE $M_{m,k}$

k \ m	2	3
2	2	2
3	3	3
4	4	4
5	4	4
6	5	4
7	6	5
8	6	6
9	7	6





Cet algorithme fait intervenir le terme **d(m,k)** qui est un élément d'une matrice $\mathbf{M}_{m,k}$ (organigramme matrice $\mathbf{M}_{m,k)}$ définie de la façon suivante:

Soient $\mathbf{m_1}$ le nombre de bits d'information du message à transmettre et $\mathbf{k_1}$ le nombre de bits de contrôle nécessaires. On recherche pour chaque polynôme générateur de degré $\mathbf{k_1}$ la distance minimale réalisable entre les mots codes. Lorsque tous les polynômes auront été traités, on prendra la plus grande d'entre elles et elle sera appelée **d(m,k).** Ce calcul peut être répété pour différentes valeurs de **m** et de **k.** A titre d'exemple nous avons fait ces calculs pour des messages de longueur **2** et **3** et des valeurs de **k** de **2** à **9.**

MATRICE Mm,k

Mm,k =

k \ m	2	3
2	2	2
3	3	3
4	4	4
5	4	4
6	5	4
7	6	5
8	6	6
9	7	6

Dans l'exemple (graphe1) que nous avons choisi : **k=9 m=3** donc **d(m,k)=6.** Il existe au moins un polynôme générateur de degré **9** réalisant une distance minimale de **6** entre les mots codes.

L'algorithme de calcul du **PND'** donne: **PND' = $6{,}6.10^{-13}$**

Cette valeur est largement inférieure à l'objectif de sécurité exigé (**$OS=10^{-8}$**) et donc le code **C(12,3)** repond largement à ce qui est demandé.

Nous pouvons donc dire, et nous l'avons vérifié dans d'autres cas, que la méthode mise au point est valable et que les approximations utilisées sont correctes.

Cependant nous remarquons qu'il existe une différence très importante entre l'objectif de sécurité exigé et la probabilité de non détection d'erreurs **PND'**. Dans notre application la marge est de **$1{,}51.10^{4}$,** elle varie bien entendu suivant les valeurs de **P, OS et m.** Cela était prévisible puisque le calcul exposé en **III-3** donnait une distance **d=4** alors que la matrice $\mathbf{M_{m,k}}$ donnait pour le nombre de bits de contrôle imposé **k=9**, une distance d supérieure **(d=6).** Ceci provient d'une part des approximations introduites en **(III_2)** dans la détermination de **n et d,** et d'autre part dans le fait que le code

réalisé permet de garantir une correction d'erreurs (c_o = **2** dans notre cas).

Si on ne souhaite pas conserver une capacité de correction du code et cela est souvent le cas dans la transmission d'ordres en sécurité, on peut envisager d'optimiser le codage.

III-4 Recherche du code optimum.

Nous prenons en compte cette fois uniquement la valeur de "**d**" obtenue à l'aide de la méthode III-2. On détermine k à l'aide de la matrice $M_{m,k}$. Pour **m=3** et **d=4** on lit **k=4**.

Un algorithme pratiquement identique à celui donné en **(III-3)** donne la valeur de **PND'.**

On obtient cette fois: **PND'=2,81.10^{-9}**

Cette valeur est beaucoup plus proche de l'objectif de sécurité (marge de **3,5** environ) et l'on conserve toutefois la possibilité de corriger une erreur (c_o=**1**).

Il existe au moins un polynôme de degré **4** qui garantit une distance minimale entre les mots codes de **4** et il est intéressant de calculer pour un polynôme particulier la valeur du **PND** obtenue. Comme il peut exister plusieurs polynômes qui répondent à la condition de distance exprimée ci-dessus, nous allons les trier de la façon suivante:

On définit pour chaque polynôme une matrice **D(n,n),** donnant pour chaque couple de mot code N_i,N_j la distance d_{ij} entre ces mots. On dénombre ensuite le nombre de combinaisons de mots codes qui ont entre eux une distance **d, d+1, d+2...n.** Le meilleur polynôme sera celui qui présente le moins de combinaisons de mots qui ont une distance **d,** s'il y en a plusieurs on passe à **d+1** et ainsi de suite jusqu'à **n.** En effet la probabilité de non détection d'erreurs **(PND)** liée au code généré par un polynôme est proportionnelle au produit

du nombre de combinaisons des messages qui ont entre eux une distance **i** par la probabilité d'erreurs par bits (**P**) à la puissance **i** et par (**1-P**) à la puissance **n-i** pour **i** variant de **d** à **n**.

Dans l'exemple que nous traitons, avec les polynômes de degré **4,** deux d'entre eux seront optimum:

$$X^4 + X^2 + X + 1$$

et

$$X^4 + X^3 + X^2 + 1$$

Ces deux polynômes ont d'ailleurs une forme identique. Ils conduisent à une valeur de la probabilité de non détection d'erreur : (voir courbe 6 page 54).

$$PND = 1{,}38.10^{-11}$$

Conclusion.

Il est donc possible de trouver le polynôme générateur qui conduira au code le moins sensible aux perturbations.

CHAPITRE III

APPLICATION AUX SYSTEMES DE TRANSPORT

I PRESENTATION GENERALE.

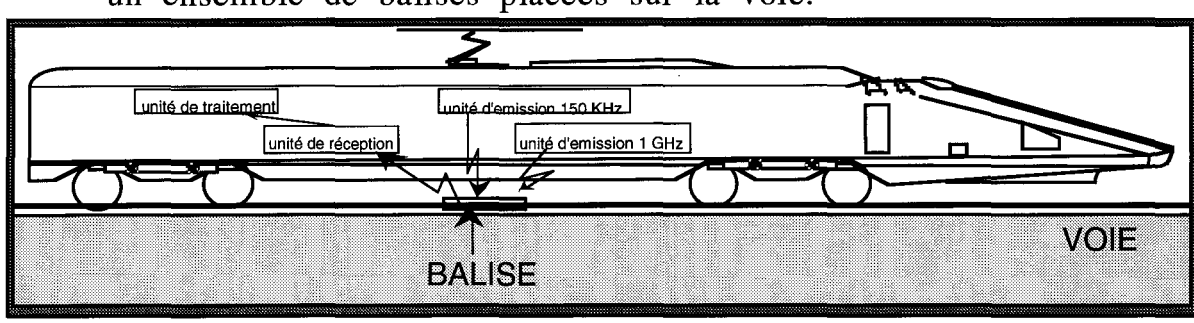
Dans le cadre des systèmes automatisés de transport, la localisation précise des véhicules est un point très important.

Pour connaître la position des mobiles, une solution consiste à effectuer un marquage du sol à l'aide de balises permettant à une motrice équipée d'un interrogateur de lire la voie et d'interpréter les messages reçus lorsque celle-ci passe au droit des balises.

Le dispositif de localisation des véhicules guidés comporte deux sous ensembles:

-un système embarqué composé d'un interrogateur comportant deux unités d'émission, une unité de réception et une unité de traitement.

-un ensemble de balises placées sur la voie.



Deux ondes sont émises simultanément aux fréquences F_{HF}=1GHz et F_{MF}=150 KHz. Au niveau de la balise, s'effectue le mélange de ces deux signaux et il y a réémission vers la voiture de l'onde UHF à la fréquence F=1GHz $\pm$150 KHz. Reçue par l'unité de réception cette onde est dirigée vers une centrale odomètrique pour y être traitée.

Notre étude porte sur la transmission des informations de sécurité de l'émetteur (interrogateur), supposé mobile avec une vitesse pouvant atteindre V_{MAX}=360 Km/h au récepteur (balise) supposé fixe. L'intervalle Δx sur lequel se fait la transmission est égal à 20 cm.

*: D'importants travaux ont été réalisés par l'équipe de Monsieur le Professeur R. GABILLARD (USTL/LRPE) pour des projets de transports public (SNCF).

Il est nécessaire d'étudier la sécurité des transferts d'informations entre l'émetteur et le récepteur et de définir les caractéristiques du code susceptible de répondre à l'objectif de sécurité fixé pour cette étude.

Pour mettre en oeuvre les caractéristiques d'un code de détection d'erreurs, nous avons besoin de connaître le taux moyen d'erreur de cette transmission ainsi que les modèles d'erreurs prédominants.

II ETUDE THEORIQUE.

II-1 Hypothèses prises en considération.

Les principales hypothèses à considérer sont:

-les messages envoyés de l'émetteur au récepteur sont au nombre de 7. Chaque message est formé de 8 bits.

-la vitesse de l'émetteur (V_{MAX}=360 Km/h=100 m/s)

-l'ntervalle sur lequel se fait la transmission (Δx=20 cm)

-l'bjectif de sécurité (OS=10^{-10})

II-2 Relation entre la vitesse de transmission et la vitesse de l'interrogateur.

Une relation simple existe entre la vitesse de l'interrogateur V (Exemple V_{MAX} = 360 Km/h), la vitesse de transmission V_t, le nombre de bits transmis (NBT) pendant le passage de l'interrogateur et l'intervalle Δx (Exemple Δx=20cm) sur lequel se fait la transmission:

En effet:

$$V = \frac{\Delta x}{\Delta T}$$

et

$$V_t = \frac{NBT}{\Delta T}$$

donc $$V_t = \frac{NBT}{\Delta x} \times V$$

II-3 Exemple.

La transmission de 8 messages à 8 bits (donc le nombre de bits à transmettre NBT=8×8=64 bits) à la vitesse V_{MAX}=100 m/s sur l'intervalle de 20 cm, nécessite une vitesse de transmission binaire supérieure ou égale à 32 Kbits/s.

En effet:

$$V_t = (64 \times 100) : 0,2 = 32\ 000 \text{ bits/s.}$$

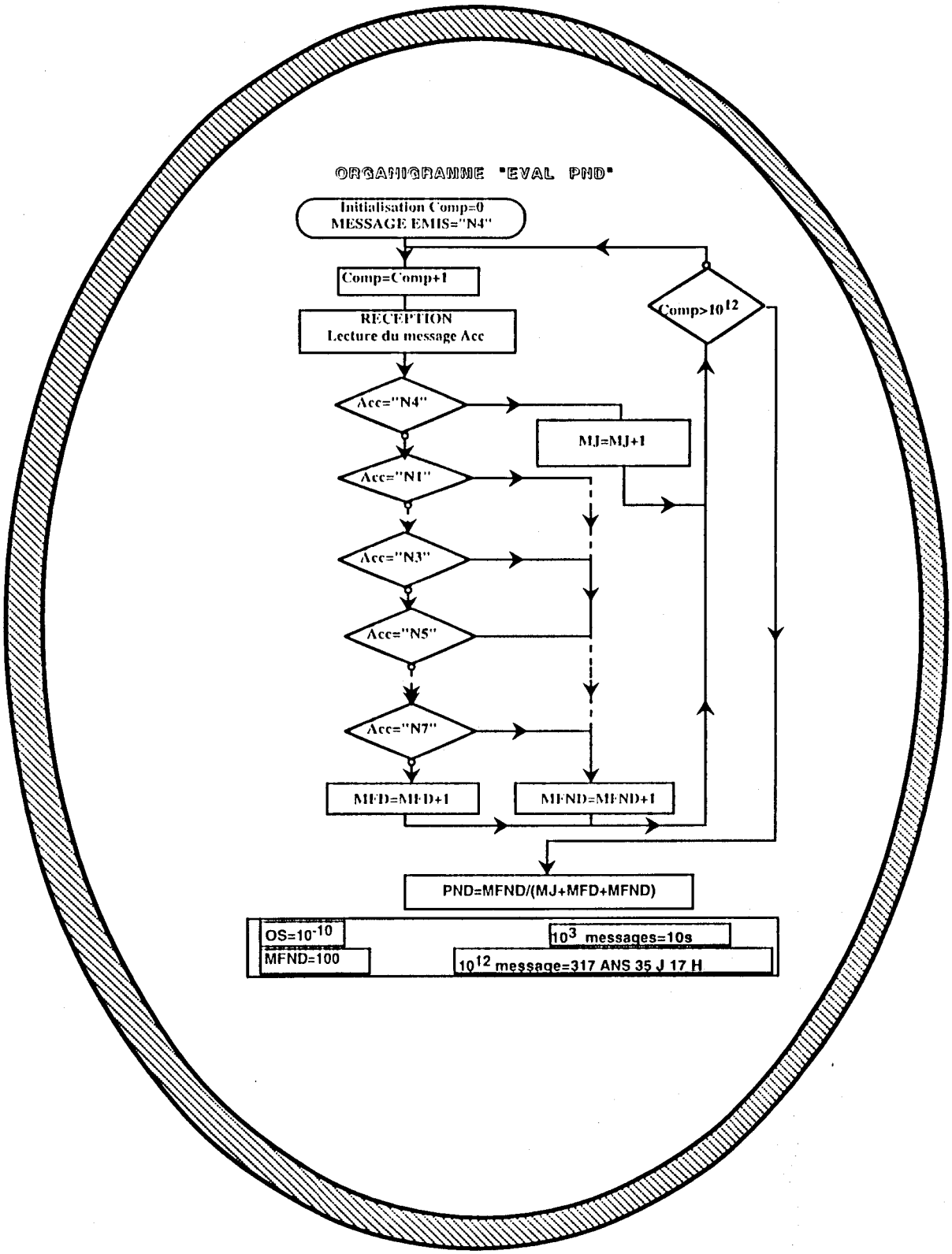
III Caractéristiques du code choisi.

Pour un nombre de bits d'information m=3, un objectif de sécurité OS=10^{-10} et une probabilité d'erreur moyenne P=0,003, la méthode définie au chapitre II donne pour un code optimum, une distance minimale d=4 et un nombre de bits de contrôle k=5. Le polynôme générateur optimum est $g(x)=1+x^2+x^3+x^4+x^5$.

Nombre de bits d'information m=3.
Nombre de bits de contrôles k=5.
Polynôme générateur $g(x)=1+x^2+x^3+x^4+x^5$.

Pour un code C(8,3) non systématique, les mots codes sont:

MESSAGES	BITS DES MESSAGES	MOTS CODES
N_1	0 0 1	0 0 1 1 1 1 0 1
N_2	0 1 0	0 1 1 1 1 0 1 0
N_3	0 1 1	0 1 0 0 0 1 1 1
N_4	1 0 0	1 1 1 1 0 1 0 0
N_5	1 0 1	1 1 0 0 1 0 0 1
N_6	1 1 0	1 0 0 0 1 1 1 0
N_7	1 1 1	1 0 1 1 0 0 1 1


ORGANIGRAMME "EVAL PND"
Initialisation Comp=0
MESSAGE EMIS="N4"
Comp=Comp+1
RECEPTION
Lecture du message Acc
Comp>10^12
Acc="N4"
MJ=MJ+1
Acc="N1"
Acc="N3"
Acc="N5"
Acc="N7"
MFD=MFD+1
MFND=MFND+1
PND=MFND/(MJ+MFD+MFND)
OS=10^-10
MFND=100
10^3 messages=10s
10^12 message=317 ANS 35 J 17 H

IV- RELEVES EXPERIMENTAUX

Nous avons relevé le taux, le type d'erreurs et le PND (Organigramme "EVAL PND") en statique (émetteur et récepteur fixes) et en dynamique (émetteur mobile, récepteur fixe) avec ou sans source de bruit.

Sur une série de 20 essais, nous présentons la moins favorisée c'est à dire celle ou il y a eu le maximum d'erreurs.

IV-1 ESSAIS EN STATIQUE:

IV-1-1: Sans source de bruit:

VITESSE DE LA BALISE = 0 m/s

		HEXA	BIN
MESSAGE A L'EMISSION	---->	F4 =	1 1 1 1 0 1 0 0
MESSAGE RECU PERTURBE	---->	E4 =	1 1 1 0 0 1 0 0
MESSAGE RECU PERTURBE	---->	F5 =	1 1 1 1 0 1 0 1
MESSAGE RECU PERTURBE	---->	74 =	0 1 1 1 0 1 0 0

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 3

TAUX D'ERREURS = .000468

NOMBRE DE MESSAGES TRANSMIS = 600
NOMBRE DE MESSAGES FAUX DETECTES = 3
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

Aucun des trois messages faux n'est un mot code, ce qui donne une probabilité de non détection d'erreur nulle.

Sur l'ensemble des trois messages faux, il y a cinq bits erronés d'une façon indépendante, ce qui donne un taux d'erreur brut de l'ordre de $4{,}7.10^{-4}$

IV-1-2: Avec source de bruit:

```
SOURCE DE BRUIT (EMISSION PAR UNE ANTENNE DIPOLE)
*************************************************

VITESSE DE LA BALISE                                          = 0 m/s
DISTANCE ENTRE LA SOURCE DE BRUIT ET LE SYSTEME INTER         = 1 m
LA FREQUENCE DE LA SOURCE DE BRUIT                            = 1 GHz
PUISSANCE DE SORTIE DE LA SOURCE DE BRUIT                     = 4 dbm
```

		HEXA		BIN							
MESSAGE A L'EMISSION	---->	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	---->	04	=	0	0	0	0	0	1	0	0
MESSAGE RECU PERTURBE	---->	00	=	0	0	0	0	0	0	0	0
MESSAGE RECU PERTURBE	---->	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	---->	CA	=	1	1	0	0	1	0	1	0
MESSAGE RECU PERTURBE	---->	15	=	0	0	0	1	0	1	0	1
MESSAGE RECU PERTURBE	---->	28	=	0	0	1	0	1	0	0	0
MESSAGE RECU PERTURBE	---->	B6	=	1	0	1	1	0	1	1	0
MESSAGE RECU PERTURBE	---->	45	=	0	1	0	0	0	1	0	1
MESSAGE RECU PERTURBE	---->	0F	=	0	0	0	0	1	1	1	1

```
NOMBRE DE BITS TRANSMIS  = 6400
NOMBRE DE BITS ERRONES   = 39

TAUX D'ERREURS = .006093
```

```
NOMBRE DE MESSAGES TRANSMIS              = 800
NOMBRE DE MESSAGES FAUX DETECTES         = 9
NOMBRE DE MESSAGES FAUX NON DETECTES     = 0

   PND = 0
   OS  = 1.E-10
```

```
SOURCE DE BRUIT (EMISSION PAR UNE ANTENNE DIPOLE)
*************************************************

VITESSE DE LA BALISE                                          = 0 m/s
DISTANCE ENTRE LA SOURCE DE BRUIT ET LE SYSTEME INTER = 1 m
LA FREQUENCE DE LA SOURCE DE BRUIT                            = 1 GHz
PUISSANCE DE SORTIE DE LA SOURCE DE BRUIT                     = 8 dbm
```

		HEXA		BIN							
MESSAGE A L'EMISSION	----)	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	----)	10	=	0	0	0	1	0	0	0	0
MESSAGE RECU PERTURBE	----)	05	=	0	0	0	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	0A	=	0	0	0	0	1	0	1	0
MESSAGE RECU PERTURBE	----)	5A	=	0	1	0	1	1	0	1	0
MESSAGE RECU PERTURBE	----)	88	=	1	0	0	0	1	0	0	0
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	CB	=	1	1	0	0	1	0	1	1
MESSAGE RECU PERTURBE	----)	46	=	0	1	0	0	0	1	1	0
MESSAGE RECU PERTURBE	----)	82	=	1	0	0	0	0	0	1	0
MESSAGE RECU PERTURBE	----)	00	=	0	0	0	0	0	0	0	0
MESSAGE RECU PERTURBE	----)	F5	=	1	1	1	1	0	1	0	1
MESSAGE RECU PERTURBE	----)	53	=	0	1	0	1	0	0	1	1
MESSAGE RECU PERTURBE	----)	11	=	0	0	0	1	0	0	0	1
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	8F	=	1	0	0	0	1	1	1	1
MESSAGE RECU PERTURBE	----)	99	=	1	0	0	1	1	0	0	1

```
NOMBRE DE BITS TRANSMIS  = 6400
NOMBRE DE BITS ERRONES   = 74

TAUX D'ERREURS = .011562



NOMBRE DE MESSAGES TRANSMIS          = 800
NOMBRE DE MESSAGES FAUX DETECTES     = 16
NOMBRE DE MESSAGES FAUX NON DETECTES = 0


   PND = 0
   OS  = 1.E-10
```

En présence d'une source de bruit on remarque l'existence de paquets d'erreurs.

Sur l'ensemble des neuf messages faux aucun n'est un mot code, ce qui donne un PND nul. On remarque que le taux d'erreur augmente lorsque la puissance de la source de bruit augmente.

IV-2 ESSAIS EN DYNAMIQUE:

IV-2-1: Sans source de bruit:

VITESSE DE LA BALISE = 10 m/s

		HEXA		BIN							
MESSAGE A L'EMISSION	---->	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	---->	0B	=	0	0	0	0	1	0	1	1
MESSAGE RECU PERTURBE	---->	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	---->	05	=	0	0	0	0	0	1	0	1
MESSAGE RECU PERTURBE	---->	0A	=	0	0	0	0	1	0	1	0
MESSAGE RECU PERTURBE	---->	11	=	0	0	0	1	0	0	0	1

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 28

TAUX D'ERREURS = .004375

NOMBRE DE MESSAGES TRANSMIS = 800
NOMBRE DE MESSAGES FAUX DETECTES = 5
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

En dynamique et sans source de bruit, il y a présence de paquets d'erreurs. Sur l'ensemble des 5 messages faux aucun n'est un mot code.

VITESSE DE LA BALISE = 15 m/s

		HEXA		BIN							
MESSAGE A L'EMISSION	----)	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	----)	36	=	0	0	1	1	0	1	1	0
MESSAGE RECU PERTURBE	----)	35	=	0	0	1	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	06	=	0	0	0	0	0	1	1	0
MESSAGE RECU PERTURBE	----)	F0	=	1	1	1	1	0	0	0	0
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	45	=	0	1	0	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	25	=	0	0	1	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	0B	=	0	0	0	0	1	0	1	1

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 32

TAUX D'ERREURS = .005

NOMBRE DE MESSAGES TRANSMIS = 800
NOMBRE DE MESSAGES FAUX DETECTES = 8
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

VITESSE DE LA BALISE = 15 m/s

		HEXA		BIN							
MESSAGE A L'EMISSION	----)	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	0F	=	0	0	0	0	1	1	1	1
MESSAGE RECU PERTURBE	----)	45	=	0	1	0	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	63	=	1	0	0	0	0	0	1	1
MESSAGE RECU PERTURBE	----)	06	=	0	0	0	0	0	1	1	0
MESSAGE RECU PERTURBE	----)	F0	=	1	1	1	1	0	0	0	0
MESSAGE RECU PERTURBE	----)	D6	=	1	1	0	1	0	1	1	0

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 28

TAUX D'ERREURS = .004375

NOMBRE DE MESSAGES TRANSMIS = 800
NOMBRE DE MESSAGES FAUX DETECTES = 7
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

IV-2-2: Avec source de bruit:

SOURCE DE BRUIT (EMISSION PAR UNE ANTENNE DIPOLE)
****

VITESSE DE LA BALISE = 10 m/s
DISTANCE ENTRE LA SOURCE DE BRUIT ET LE SYSTEME INTER = 1 m
LA FREQUENCE DE LA SOURCE DE BRUIT = 1 GHz
PUISSANCE DE SORTIE DE LA SOURCE DE BRUIT = 8 dbm

		HEXA		BIN							
MESSAGE A L'EMISSION	---->	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	---->	04	=	0	0	0	0	0	1	0	0
MESSAGE RECU PERTURBE	---->	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	---->	00	=	0	0	0	0	0	0	0	0
MESSAGE RECU PERTURBE	---->	15	=	0	0	0	1	0	1	0	1
MESSAGE RECU PERTURBE	---->	F6	=	1	1	1	1	0	1	1	0
MESSAGE RECU PERTURBE	---->	0F	=	0	0	0	0	1	1	1	1
MESSAGE RECU PERTURBE	---->	59	=	0	1	0	1	1	0	0	1
MESSAGE RECU PERTURBE	---->	B6	=	1	0	1	1	0	1	1	0
MESSAGE RECU PERTURBE	---->	83	=	1	0	0	0	0	0	1	1
MESSAGE RECU PERTURBE	---->	B5	=	1	0	1	1	0	1	0	1
MESSAGE RECU PERTURBE	---->	AA	=	1	0	1	0	1	0	1	0
MESSAGE RECU PERTURBE	---->	B6	=	1	0	1	1	0	1	1	0
MESSAGE RECU PERTURBE	---->	88	=	1	0	0	0	1	0	0	0

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 51

TAUX D'ERREURS = .007968

NOMBRE DE MESSAGES TRANSMIS = 800
NOMBRE DE MESSAGES FAUX DETECTES = 13
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

SOURCE DE BRUIT (EMISSION PAR UNE ANTENNE DIPOLE)
****

VITESSE DE LA BALISE = 15 m/s
DISTANCE ENTRE LA SOURCE DE BRUIT ET LE SYSTEME INTER = 1 m
LA FREQUENCE DE LA SOURCE DE BRUIT = 1 GHz
PUISSANCE DE SORTIE DE LA SOURCE DE BRUIT = 8 dbm

		HEXA		BIN							
MESSAGE A L'EMISSION	----)	F4	=	1	1	1	1	0	1	0	0
MESSAGE RECU PERTURBE	----)	58	=	0	1	0	1	1	0	0	0
MESSAGE RECU PERTURBE	----)	63	=	0	1	1	0	0	0	1	1
MESSAGE RECU PERTURBE	----)	CC	=	1	1	0	0	1	1	0	0
MESSAGE RECU PERTURBE	----)	CA	=	1	1	0	0	1	0	1	0
MESSAGE RECU PERTURBE	----)	F0	=	1	1	1	1	0	0	0	0
MESSAGE RECU PERTURBE	----)	0A	=	0	0	0	0	1	0	1	0
MESSAGE RECU PERTURBE	----)	05	=	0	0	0	0	0	1	0	1
MESSAGE RECU PERTURBE	----)	0F	=	0	0	0	0	1	1	1	1
MESSAGE RECU PERTURBE	----)	02	=	0	0	0	0	0	0	1	0
MESSAGE RECU PERTURBE	----)	00	=	0	0	0	0	0	0	0	0
MESSAGE RECU PERTURBE	----)	1F	=	0	0	0	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	56	=	0	1	0	1	0	1	1	0
MESSAGE RECU PERTURBE	----)	86	=	1	0	0	0	0	1	1	0
MESSAGE RECU PERTURBE	----)	53	=	0	1	0	1	0	0	1	1
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	52	=	0	1	0	1	0	0	1	0
MESSAGE RECU PERTURBE	----)	00	=	0	0	0	0	0	0	0	0
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1
MESSAGE RECU PERTURBE	----)	FF	=	1	1	1	1	1	1	1	1

NOMBRE DE BITS TRANSMIS = 6400
NOMBRE DE BITS ERRONES = 90

TAUX D'ERREURS = .014062

NOMBRE DE MESSAGES TRANSMIS = 800
NOMBRE DE MESSAGES FAUX DETECTES = 21
NOMBRE DE MESSAGES FAUX NON DETECTES = 0

PND = 0
OS = 1.E-10

En dynamique et avec source de bruit, le nombre de bits erronés augmente, ce qui donne un taux d'erreurs elevé mais aucun des messages faux ne s'est transformé en un autre mot code.

IV-3- Conclusion:

En statique, sans sources de bruits extérieures il y a très peu de bits erronés. Les quelques erreurs existantes sont individuelles et sont dues aux bruits du système.

En dynamique ou en présence de sources de bruits extérieures il y a présence de paquets d'erreurs dont la longueur dépend de la vitesse de la balise et de la puissance de la source de bruit.

Ces résultats ne sont pas représentatifs de la qualité de la liaison puisque le nombre de bits transmis n'est guère élevé, c'est pour cela que le nombre de messages faux non détectés est nul.

Pour un $OS=10^{-10}$ et pour obtenir un nombre moyen de messages faux non détectés de l'ordre de 100 (en supposant que 10^3 messages transitent pendant 10 secondes et que les essais s'éffectuent sans arrêt) il faudra 317 Ans 35 Jours 17 Heures pour faire transiter 10^{12} messages !!!. C'est pour cela qu'il nous a paru judicieux de calculer le PND par simulation.

V Calcul DU PND par simulation.

Si les parasites modifient un nombre de bits du message supérieur ou égal à la valeur de **d**, alors deux cas peuvent se présenter:

1er cas: Le message est faux et détecté par le décodeur, autrement dit les parasites ne réussissent pas à le transformer en un mot code. L'ensemble de ces messages est appelé: **MFD** (messages faux détectés).

2eme cas: Le mot code est transformé par les parasites en un autre mot code et échappe au système de détection d'erreurs. L'ensemble de ces messages est appelé: **MFND** (messages faux non détectés).

V-1 Caractéristiques du code choisi.

Pour un nombre de bits d'information m=3, un objectif de sécurité $OS=10^{-10}$ et une probabilité d'erreur moyenne P=0,003, la méthode définie au chapitre II donne pour un code optimum, une distance minimale d=4 et un nombre de bit de contrôle k=5. Le polynôme génerateur optimum est $g(x)=1+x^2+x^3+x^4+x^5$.

Nombre de bit d'informations m=3.
Nombre de bits de contrôles k=5.
Polynôme générateur $g(x)=1+x^2+x^3+x^4+x^5$.

Pour un code C(8,3) non systématique, les mots codes sont:

MESSAGES	BITS DES MESSAGES	MOTS CODES
N_1	0 0 1	0 0 1 1 1 1 0 1
N_2	0 1 0	0 1 1 1 1 0 1 0
N_3	0 1 1	0 1 0 0 0 1 1 1
N_4	1 0 0	1 1 1 1 0 1 0 0
N_5	1 0 1	1 1 0 0 1 0 0 1
N_6	1 1 0	1 0 0 0 1 1 1 0
N_7	1 1 1	1 0 1 1 0 0 1 1

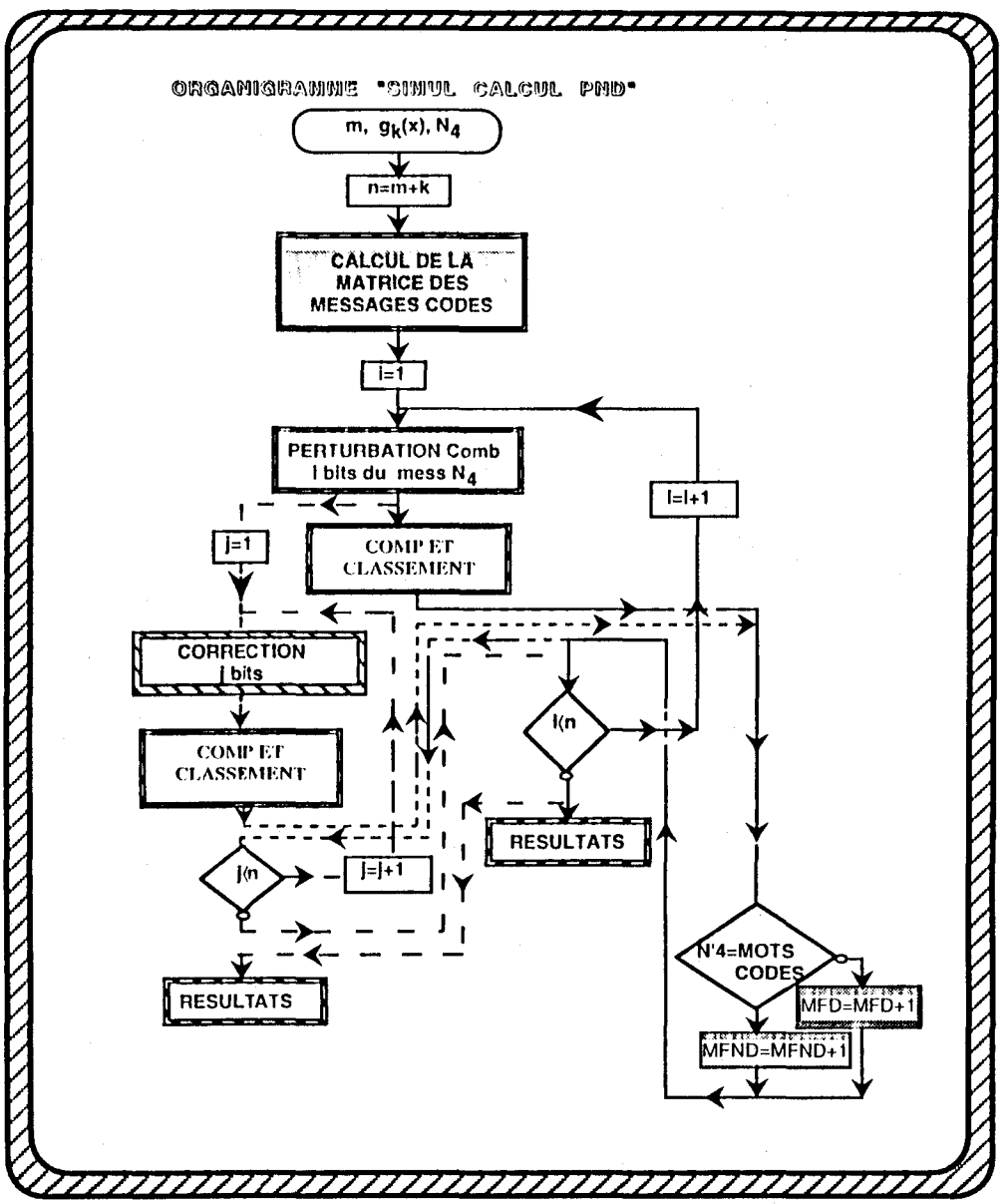
La matrice de distances $D_{Ni,Nj}$ entre les mots codes est:

$D_{Ni,Nj}$ =

	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_1	0	4	5	4	5	5	4
N_2	4	0	5	4	5	5	4
N_3	5	5	0	5	4	4	5
N_4	4	4	5	0	5	5	4
N_5	5	5	4	5	0	4	5
N_6	5	5	4	5	4	0	5
N_7	4	4	5	4	5	5	0

@

On constate une symétrie par rapport à la première diagonale de la matrice $D_{Ni,Nj}$ ($d(N_i,N_j)=d(N_j,N_i)$) et que les valeurs d(Ni,Ni) sont bien nulles.


ORGANIGRAMME "SIMUL CALCUL PND"
m, $g_k(x)$, N_4
n=m+k
CALCUL DE LA MATRICE DES MESSAGES CODES
I=1
PERTURBATION Comb I bits du mess N_4
I=I+1
J=1
COMP ET CLASSEMENT
CORRECTION I bits
COMP ET CLASSEMENT
I(n
RESULTATS
J(n
J=J+1
RESULTATS
N'4=MOTS CODES
MFD=MFD+1
MFND=MFND+1

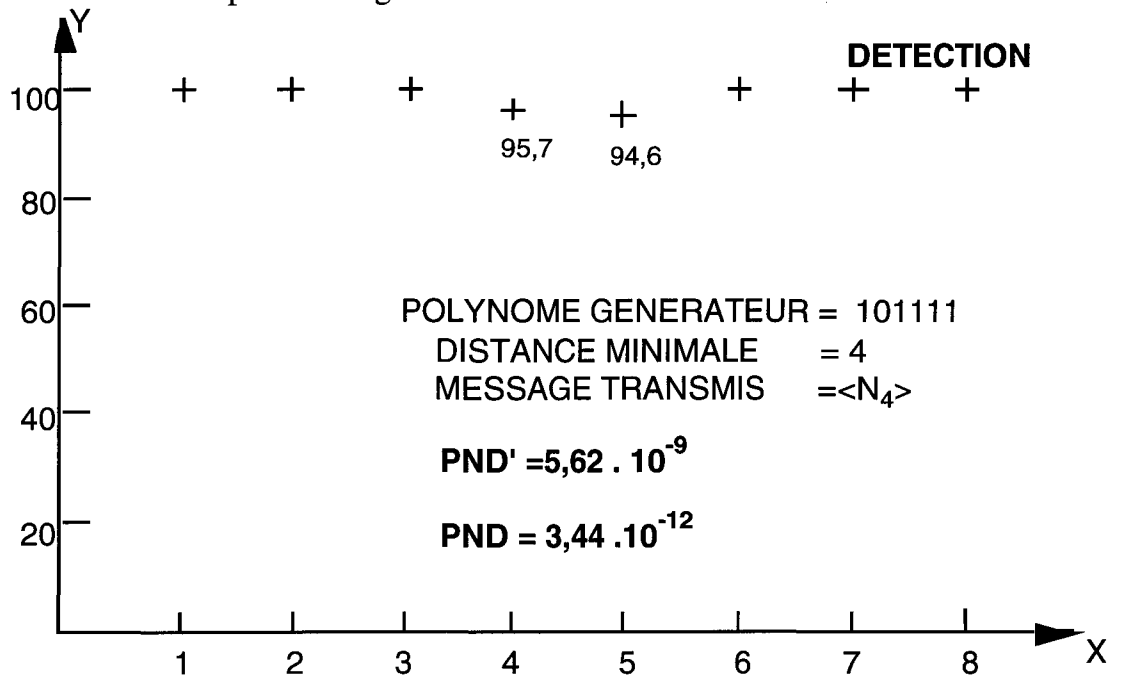
V-2 Courbe de pourcentage de détection d'erreurs.

cette courbe (courbe1) permet de donner le pourcentage de détection d'erreurs pour un polynôme générateur donné, c'est à dire le pourcentage de messages faux détectés et par conséquent celui des messages faux non détectés par le décodeur en fonction du nombre de bits perturbés.

courbe 1 : pourcentage de détection d'erreurs

axe des X : nombre de bits perturbés

axe des Y : pourcentage de détection d'erreurs



On obtient cette courbe (Organigramme "SIMUL CALCUL PND") en modifiant un bit à la fois du message puis toutes les combinaisons de deux bits, trois bits jusqu'à perturber tous les bits du message. Pour chaque modification, on regarde si le décodeur est capable de détecter le message faux afin de pouvoir le classer en messages faux détectés ou en messages faux non détectés. Si les mots codes ne sont pas équidistants, alors cette courbe change en fonction du message à transmettre.

Le code choisi permet de donner une distance minimale de 4. C'est alors avec certitude que l'on détectera une, deux ou trois erreurs

$D_{Ni,Nj}$ =

	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_1	0	4	5	4	5	5	4
N_2	4	0	5	4	5	5	4
N_3	5	5	0	5	4	4	5
N_4	4	4	5	0	5	5	4
N_5	5	5	4	5	0	4	5
N_6	5	5	4	5	4	0	5
N_7	4	4	5	4	5	5	0

pour tous les messages. Pour le message N_4 (courbe1) on a 95,7% de détection d'erreurs lorsqu'on perturbe 4 bits et 94,6% de détection d'erreurs lorsqu'on perturbe 5 bits, mais 100% de détection d'erreurs lorsqu'on perturbe 6, 7 ou 8 bits bien que ce soit supérieur à la distance minimale. Cela est dû au fait qu'il n'y a pas de couple de mots codes dont la distance est égale à 6, 7 ou 8 mais il y a 9 couples de mots codes dont la distance est égale à 4 et 12 couples dont la distance est égale à 5. Donc ce code permet de détecter les erreurs individuelles et les paquets d'erreurs de longueur 2,3,6,7 ou 8.

On peut vérifier ces valeurs par la formule proposé par Monsieur le Professeur R. GABILLARD. Pour le message N_4, lorsqu'on perturbe 4 bits, il y a 3 possibilités pour transformer le mot N_4 en un autre mot code (voir matrice de distance), donc un message faux non détecté MFND parmi C_8^4 cas possible. Le pourcentage de détection de 4 erreurs pour le message N_4 est:

$100\times(1-\frac{3}{C_8^4})=95,7\%$. (De même pour 5 erreurs $100\times(1-\frac{3}{C_8^5})=94,6\%$.)

V-3 Calcul du PND et du PND'.

Nous avons calculé la probabilité de non détection d'erreurs (PND) et son majorant (PND') (courbe1).

Définition du PND': On suppose que tous les messages dont le nombre de bits erronés est supérieur ou égal à d ne sont pas détectés par le décodeur(PND').

$$PND' = \sum_{i=d}^{n} C_n^i P^i (1-P)^{n-i}$$

Définition du PND: Le calcul du PND se fait de la même manière que le PND' en tenant compte du pourcentage du nombre de messages non détectés par le décodeur.

$$PND = \sum_{i=d}^{n} S(i) \times P^i (1-P)^{n-i}$$

S(i) étant le pourcentage de non détections d'erreurs par le décodeur pour "i" perturbation.

Si les mots codes ne sont pas équidistants, alors le PND change en fonction du message à transmettre. Donc on calcule le PND moyen qui est égal à la moyenne des PND de tous les messages du code.

Les messages du code choisi ne sont pas équidistants. Le PND moyen des sept messages du code est calculé et présenté ci-dessous.

MESSAGE	PND
N_1	$3{,}44.10^{-12}$
N_2	$3{,}44.10^{-12}$
N_3	$2{,}30.10^{-12}$
N_4	$3{,}44.10^{-12}$
N_5	$2{,}30.10^{-12}$
N_6	$2{,}30.10^{-12}$
N_7	$3{,}44.10^{-12}$

→ PNDmoyen=$2{,}95.10^{-12}$

V-4 Conclusion:

On peut utiliser cette méthode pour calculer le PND et aussi pour connaître la capacité de détection des paquets d'erreurs d'un code.

VI CAS DES PAQUETS D'ERREURS.

On appelle paquet d'erreurs une perturbation dont la longueur est supérieure ou égale à **2 bits.** L'une des méthodes pour lutter contre ces paquets d'erreurs consiste à regarder les capacités de détection et correction des paquets d'erreurs du code choisi.

VI-1 Codes permettant de détecter des paquets d'erreurs.

VI-1-1 Exemple1: mots codes non équidistants.

Nombre de bits d'informations **m=3**.

Nombre de bits de contrôles **k=4**, polynôme générateur$\mathbf{g(x)=1+x+x^4}$

Pour un code non systématique, les mots codes sont:

MESSAGES	BITS DES MESSAGES	MOTS CODES
N_1	0 0 1	0 0 1 0 0 1 1
N_2	0 1 0	0 1 0 0 1 1 0
N_3	0 1 1	0 1 1 0 1 0 1
N_4	1 0 0	1 0 0 1 1 0 0
N_5	1 0 1	1 0 1 1 1 1 1
N_6	1 1 0	1 1 0 1 0 1 0
N_7	1 1 1	1 1 1 1 0 0 1

Matrice de distances $D_{Ni,Nj}$ entre les mots codes.

$D_{Ni,Nj}$ =

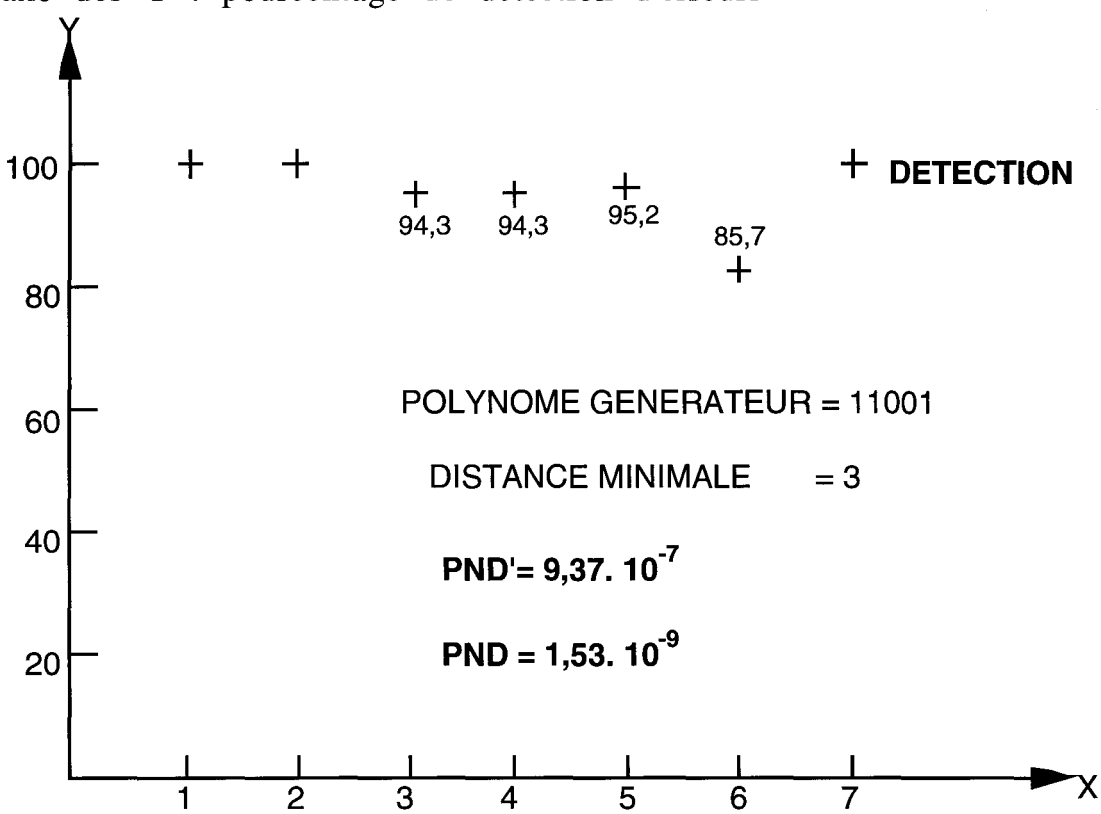
	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_1	0	4	3	6	3	5	4
N_2	4	0	3	4	5	3	6
N_3	3	3	0	5	4	6	3
N_4	6	4	5	0	3	3	4
N_5	3	5	4	3	0	4	3
N_6	5	3	6	3	4	0	3
N_7	4	6	3	4	3	3	0

Le code choisi permet de donner une distance minimale de 3. Donc on peut détecter à coup sûr une ou deux erreurs pour tous les messages du code. Pour le message N_4 (courbe2) on a 94,3% de détection d'erreurs lorsqu'on perturbe 3 bits, 94,3% de détection d'erreurs lorsqu'on perturbe 4 bits, 95,2% de détection d'erreurs lorsqu'on perturbe 5 bits, et 85,7% de détection d'erreurs lorsqu'on perturbe 6 bits, mais 100% de détection d'erreurs lorsqu'on perturbe 7 bits bien que ce soit supérieur à la distance minimale. Cela est dû au fait qu'il n'y a pas de couple de mots codes dont la distance est égale à 7, mais il y a 9 couples de mots codes dont la distance est égale à 3, 6 couples de mots codes dont la distance est égale à 4, 3 couples dont la distance est égale à 5 et 3 couples dont la distance est égale à 6.

courbe 2 : pourcentage de détection d'erreurs

Nombre de bits d'informations m=3
Nombre de bits de contrôles k=4, polynôme générateur $g(x)=1+x+x^4$
message transmis $<N_4> = <1001100>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de détection d'erreurs

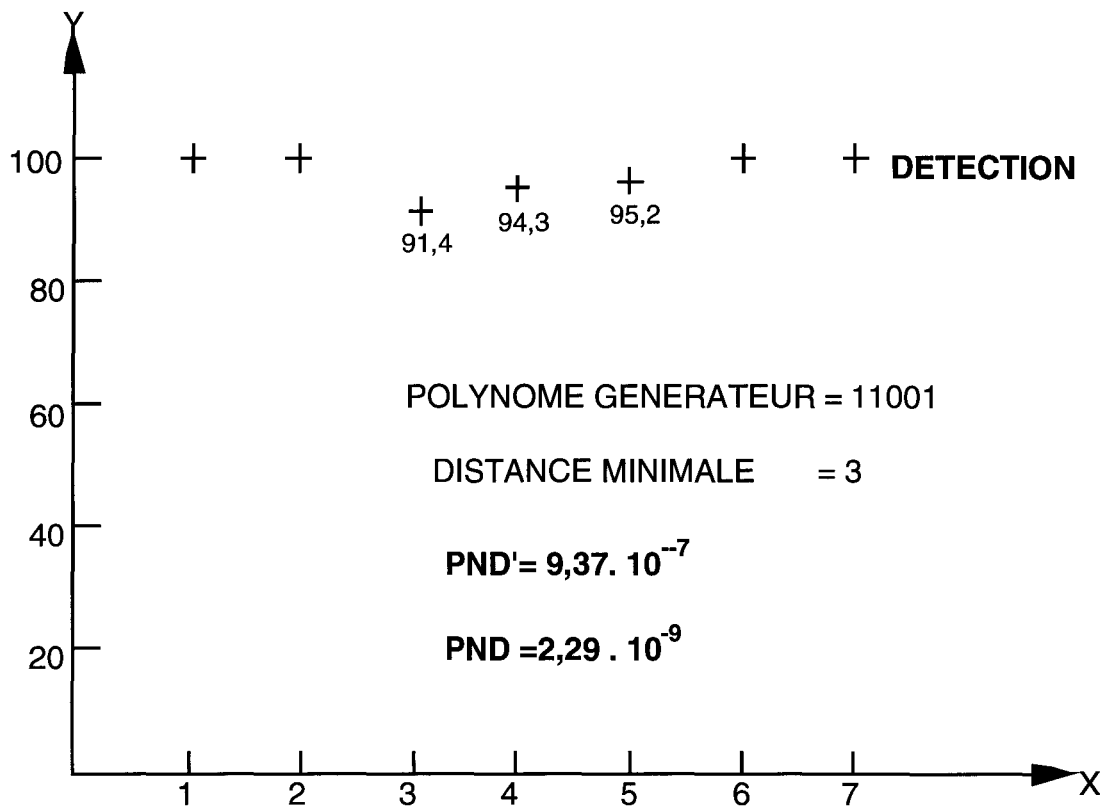


Ces valeurs changent en fonction du message à transmettre. En effet pour N_5, on obtient d'autres valeurs, (courbe3).

courbe 3: pourcentage de détection d'erreurs

nombre de bits d'informations m=3
nombre de bits de contrôles k=4, polynôme génerateur $g(x)=1+x+x^4$
message transmis $<N_5> = <1011111>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de détection d'erreurs



Remarque: Le PND change en fonction du message à transmettre (car les mots du code ne sont pas équidistants). On a calculé le PND moyen des sept messages du code.

$$\text{PND moyen} = 1{,}96 \cdot 10^{-9}$$

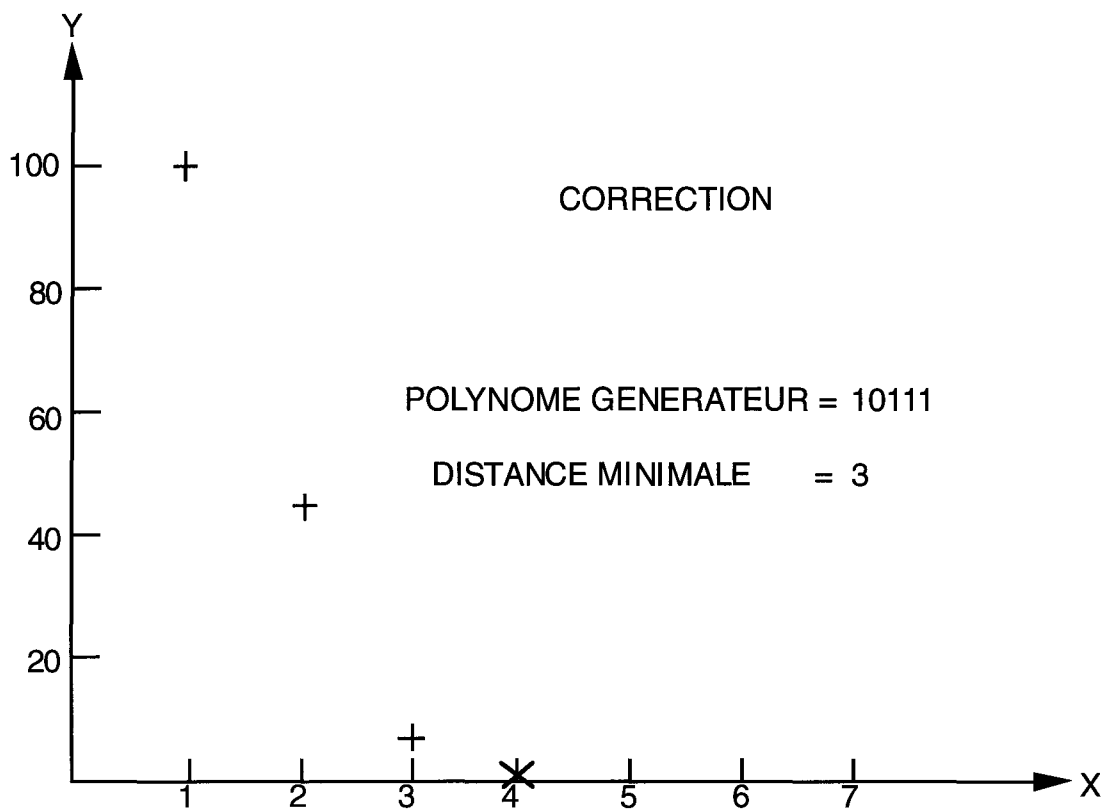
Ce code permet de détecter les erreurs individuelles et les paquets d'erreurs de longueur 2 ou 7.

De même pour la correction (courbe 4 et 5), un bit est corrigé à la fois du message ensuite toutes les combinaisons de deux bits, trois bits jusqu'à corriger tous les bits du message. Pour chaque correction on regarde si le décodeur est capable de détecter le message faux .

courbe 4: pourcentage de correction d'erreurs

nombre de bits d'informations m=3
nombre de bits de contrôles k=4, polynôme génerateur=$1+x+x^4$
message transmis $<N_4> = < 1001100>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de correction d'erreurs



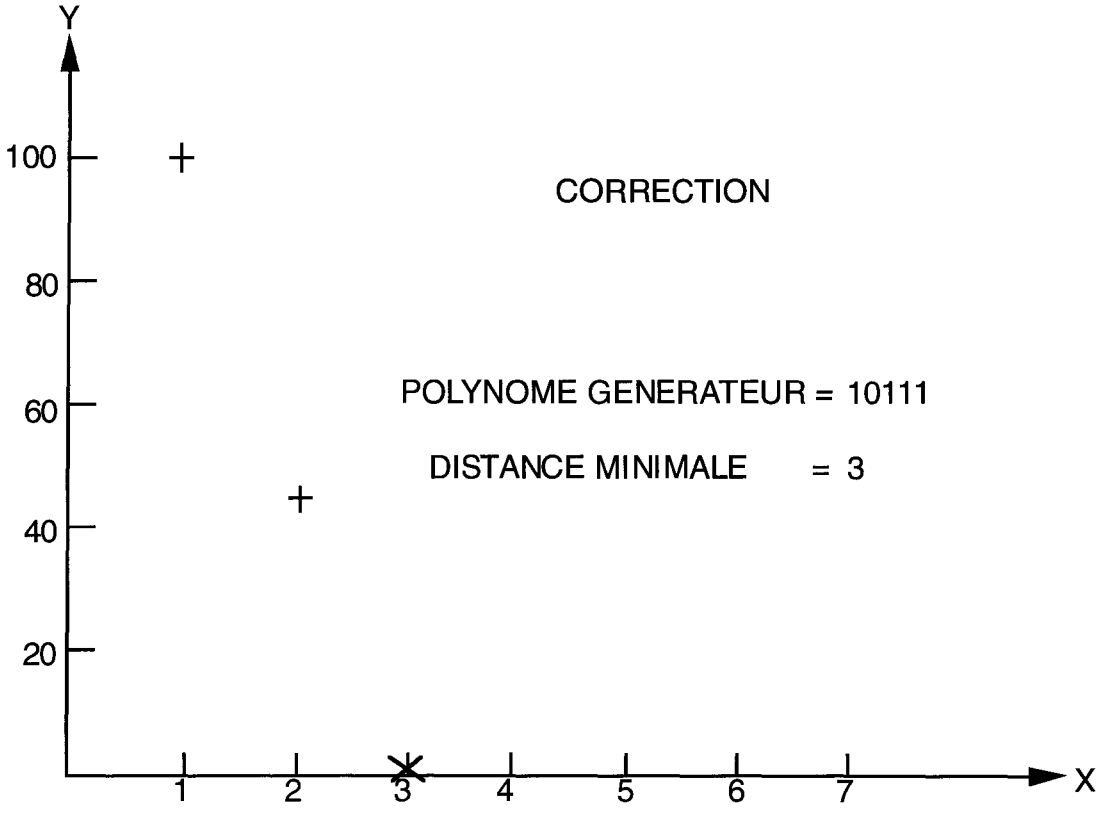
courbe 5: pourcentage de correction d'erreurs

polynôme générateur = $1 + x + x^4$

nombre de bits d'informations m=3
nombre de bits de contrôles k=4

message transmis $<N_5> = < 1011111>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de correction d'erreurs



IV-1-2 Exemple2: mots codes équidistants.

nombre de bits d'informations m = 3

nombre de bits de contrôles k=4, polynôme générateur $g(x)=x^4+x^3+x^2+1$

Pour un code non systématique, les mots codes sont:

MESSAGES	BITS DES MESSAGES	MOTS CODES
N_1	0 0 1	0 0 1 1 1 0 1
N_2	0 1 0	0 1 1 1 0 1 0
N_3	0 1 1	0 1 0 0 1 1 1
N_4	1 0 0	1 1 1 0 1 0 0
N_5	1 0 1	1 1 0 1 0 0 1
N_6	1 1 0	1 0 0 1 1 1 0
N_7	1 1 1	1 0 1 0 0 1 1

Matrice de distances $D_{Ni,Nj}$ entre les mots codes.

$D_{Ni,Nj}$ =

	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_1	0	4	4	4	4	4	4
N_2	4	0	4	4	4	4	4
N_3	4	4	0	4	4	4	4
N_4	4	4	4	0	4	4	4
N_5	4	4	4	4	0	4	4
N_6	4	4	4	4	4	0	4
N_7	4	4	4	4	4	4	0

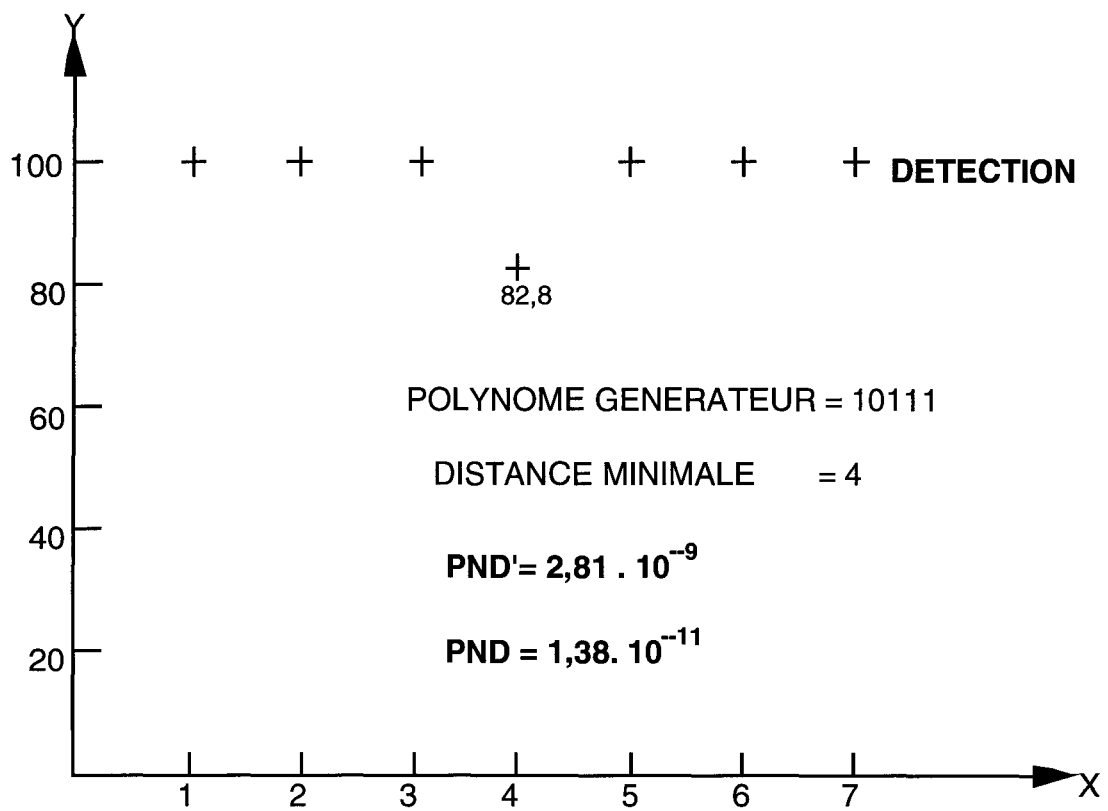
Le code choisi permet de donner une distance minimale équidistante égale à 4. Donc pour tous les messages on a 100% de détection d'erreurs lorsqu'on perturbe e bits ($e \neq 4$) et 82,8% de détection d'erreurs lorsqu'on perturbe 4 bits. (courbe 6)

courbe 6: pourcentage de détection d'erreurs

nombre de bits d'informations m=3
nombre de bits de contrôles k=4, polynôme générateur=$1+x^2+x^3+x^4$

message transmis $<N_4> = <1110100>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de détection d'erreurs

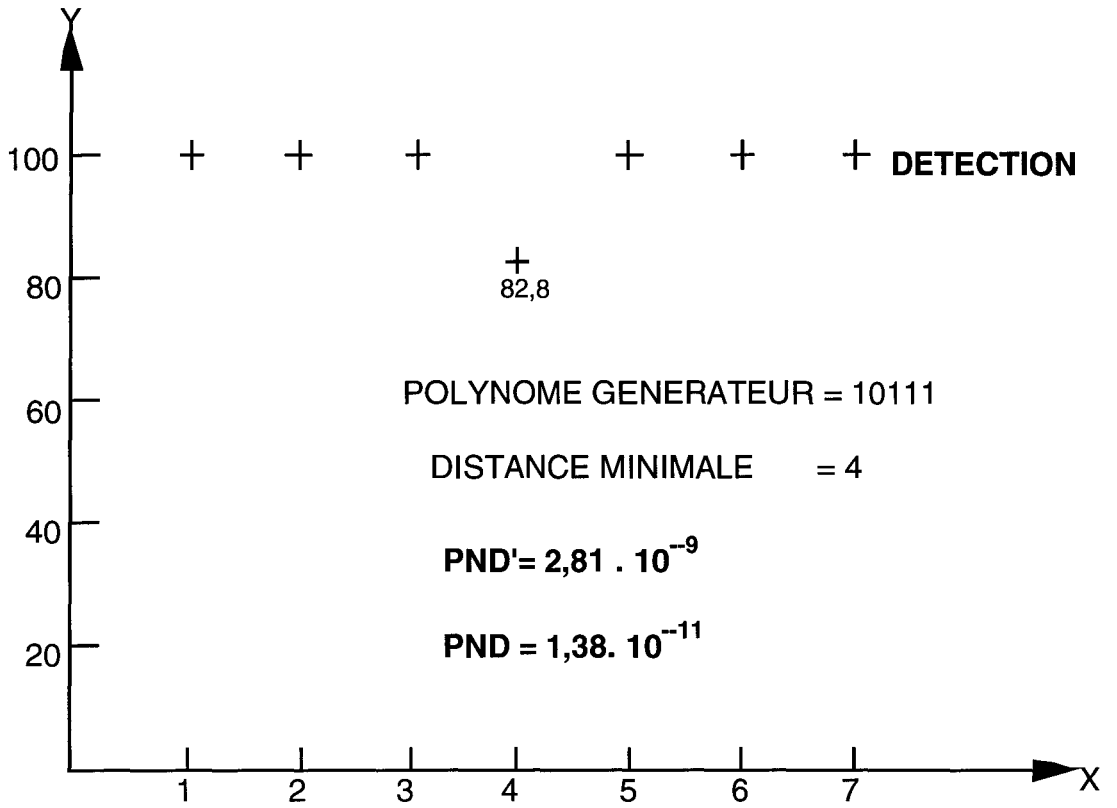


Les mots codes sont équidistants et on obtient la même courbe pour le message N6 (courbe 7).

courbe 7: pourcentage de détection d'erreurs.

nombre de bits d'informations m=3
nombre de bits de contrôles k=4, polynôme génerateur=$1+x^2+x^3+x^4$
message transmis $<N_6> = <1010011>$

axe des X : nombre de bits perturbés
axe des Y : pourcentage de détection d'erreurs



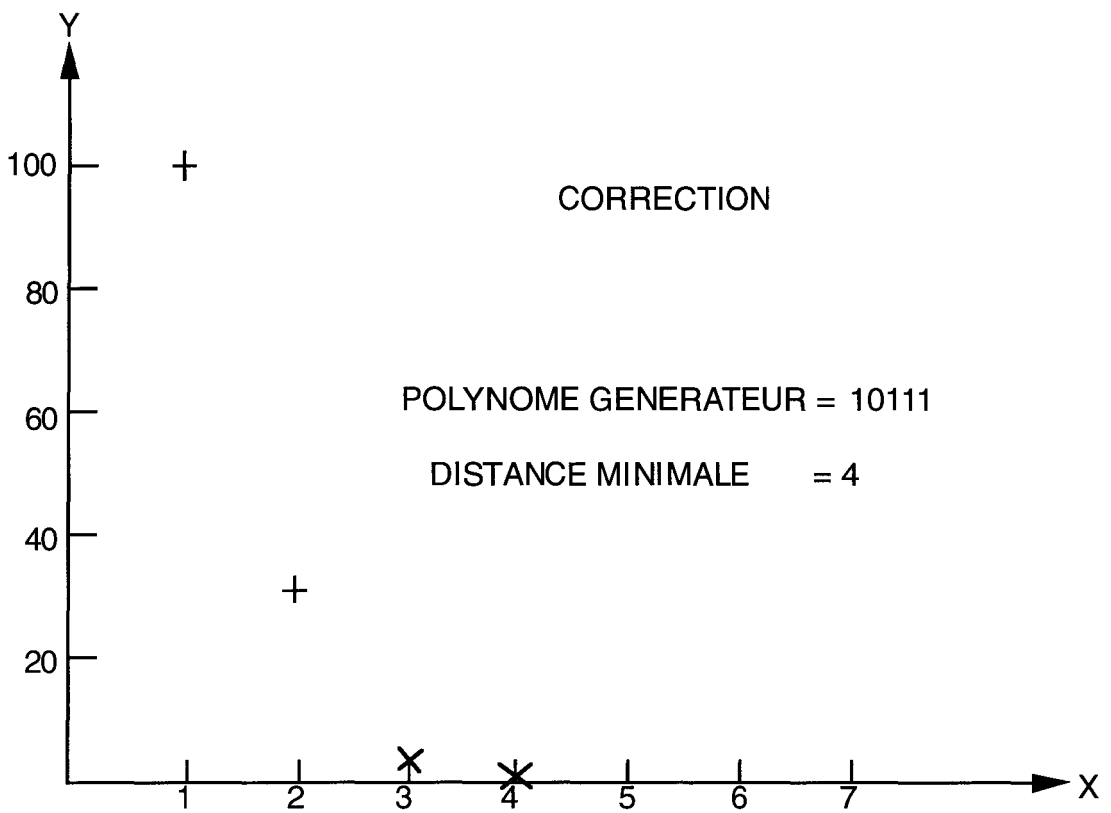
Ce code permet de détecter les erreurs individuelles et les paquets d'erreurs de longueur 2, 3, 5, 6 ou 7.

De même pour la correction on obtient les mêmes courbes pour tous les messages (courbe 8).

courbe 8: pourcentage de correction d'erreurs

nombre de bits d'informations m=3
nombre de bits de contrôles k=4, polynôme générateur=$1+x^2+x^3+x^4$
message transmis $<N_4>$ = <1110100>

axe des X : nombre de bits perturbés
axe des Y : pourcentage de correction d'erreurs



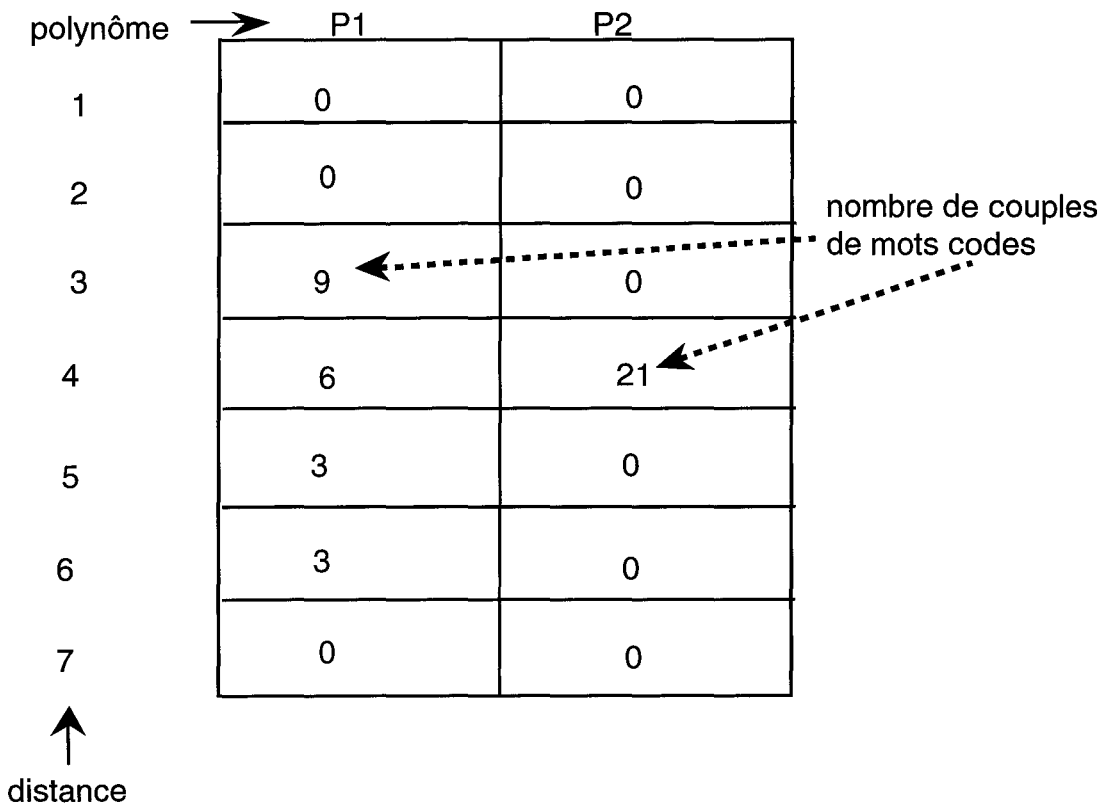
VI-1-3 Conclusion:

Pour le code généré par le polynôme P1, il y a 9 couples de mots codes dont la distance est égal à 3, 6 couples de mots codes dont la distance est égal à 4, 3 couples de mots codes dont la distance est égal à 5 et 6 couples de mots codes dont la distance est égal à 6.

Pour le code généré par le polynôme P2, il y a 21 couples de mots codes dont la distance est égal à 4.

Certes le meilleur polynôme sera celui qui présente le moins de combinaisons à des distances proches de la distance de Hamming, mais ne permettra pas forcément de détecter les paquets d'erreurs de longueur assez importante.

Le tableau ci-dessous, nous résume la capacité de détection des paquets d'erreurs des deux polynômes.



Pour des perturbations trop longues on peut utiliser des messages codés et entrelacés (code cyclique par exemple).

IV-2 Messages codés et entrelacés.

Pour éliminer ces paquets d'erreurs on peut aussi utiliser la méthode d'entrelacement simple. Cette méthode consiste à stocker dans une mémoire tous les mots codes à envoyer, puis on envoie le premier bit de chaque mot, le deuxième bit de chaque mot, et ainsi de suite jusqu'au dernier bit de chaque mot. Si on n'a qu' un seul mot à transmettre, on peut supposer une répétition du message. Le nombre de répétitions depend de la longueur du paquet d'erreur. On détermine statistiquement le temps maximum pendant lequel les parasites agissent sur les bits dans le canal de transmission. Ce temps va déterminer, en fonction de la vitesse de transmission numérique, la longueur des paquets d'erreurs. A la réception on rangera dans une autre mémoire les bits reçus de façon à regrouper chaque mot envoyé. Ceci va nous permettre d'éliminer les paquets d'erreurs, puisqu'un paquet d'erreurs va se transformer en erreurs individuelles. Ensuite on mesure la probabilité d'erreur par bit du canal de transmission et on détermine en fonction de l'objectif de sécurité le nombre de bits de contrôle et la distance minimale entre les mots codes.

On peut supposer les deux systèmes suivants:

1) Système de détection et correction d'erreur. (code cyclique par exemple).

2) Système permettant d'éliminer les paquets d'erreurs.

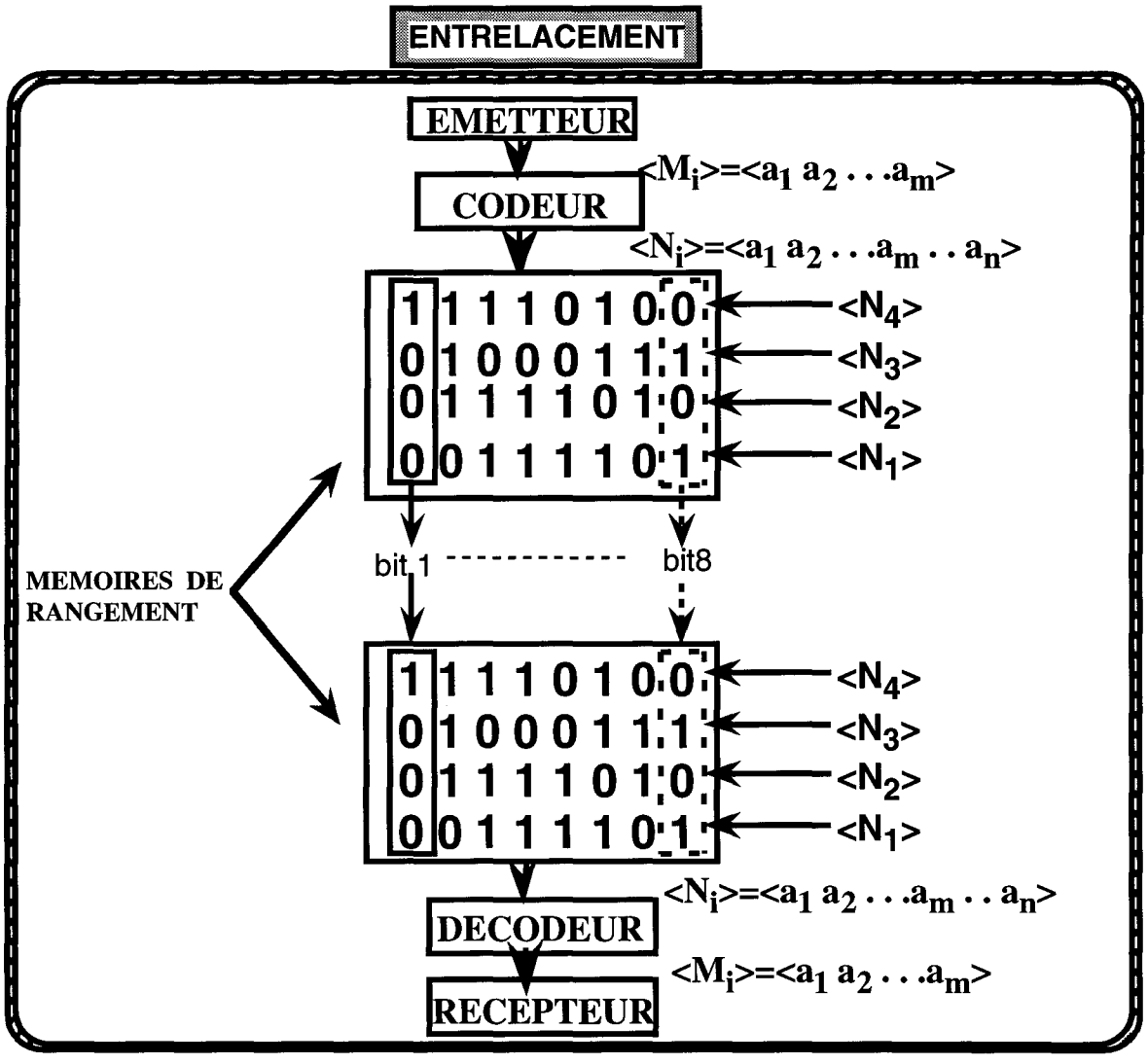
On peut schématiser ces deux systèmes selon le schéma page suivante.

Exemple:

Nombre de bit d'informations m=3.
Nombre de bits de contrôles k=5.
Polynôme générateur $g(x)=1+x^2+x^3+x^4+x^5$

Pour un code C(8,3) non systématique, les mots codes sont:

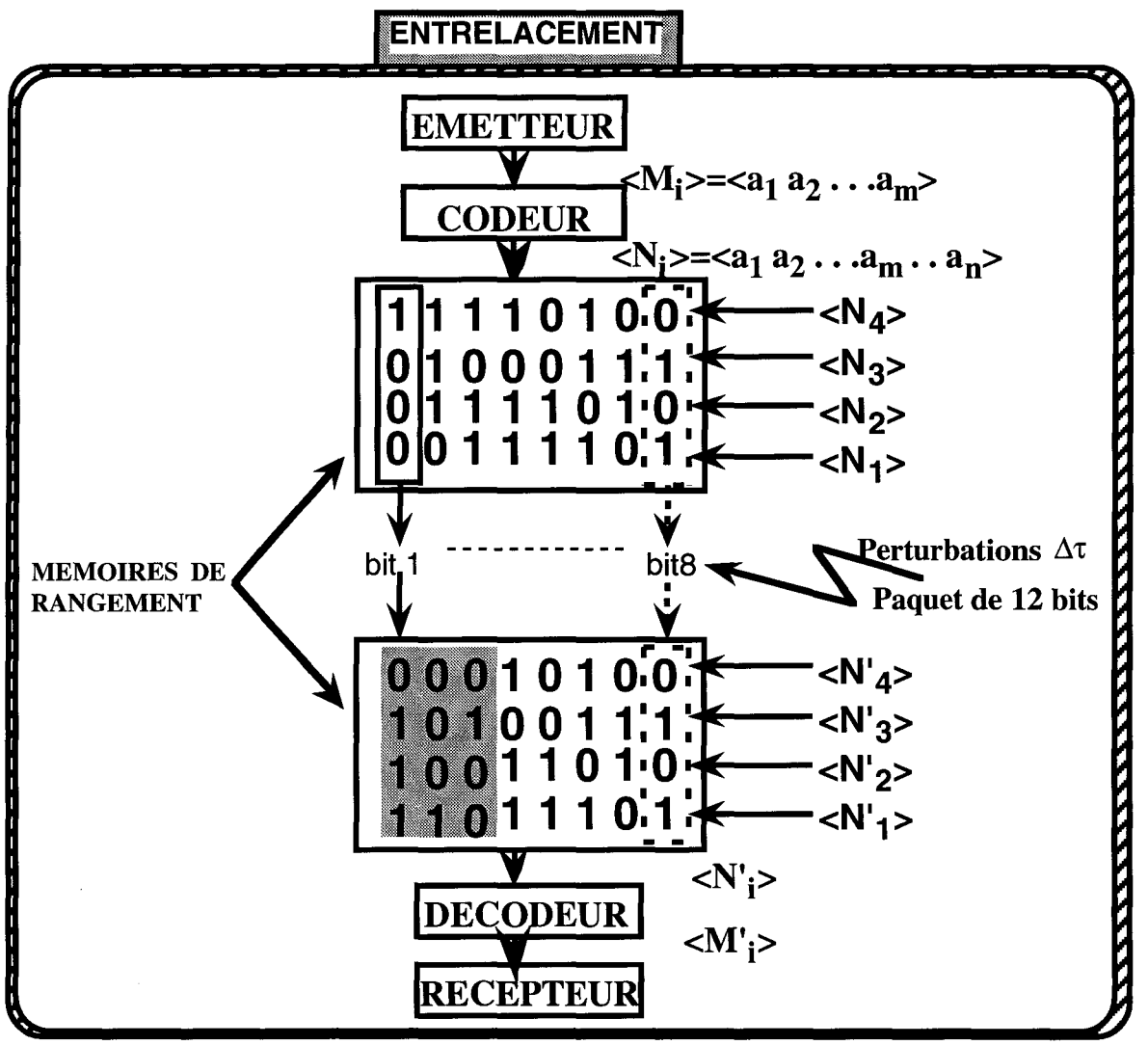
MESSAGES	BITS DES MESSAGES	MOTS CODES
N_1	0 0 1	0 0 1 1 1 1 0 1
N_2	0 1 0	0 1 1 1 1 0 1 0
N_3	0 1 1	0 1 0 0 0 1 1 1
N_4	1 0 0	1 1 1 1 0 1 0 0
N_5	1 0 1	1 1 0 0 1 0 0 1
N_6	1 1 0	1 0 0 0 1 1 1 0
N_7	1 1 1	1 0 1 1 0 0 1 1



Dans le cas d'entrelacement de 4 mots, un paquet d'erreur de longueur 12 bits se transforme en paquets d'erreurs de longueur 3 bits, comme la distance minimale entre les mots code est de 4, tous les mots faux seront détectés.

Matrice de distance $DNi,Nj =$

	N_1	N_2	N_3	N_4	N_5	N_6	N_7
N_1	0	4	5	4	5	5	4
N_2	4	0	5	4	5	5	4
N_3	5	5	0	5	4	4	5
N_4	4	4	5	0	5	5	4
N_5	5	5	4	5	0	4	5
N_6	5	5	4	5	4	0	5
N_7	4	4	5	4	5	5	0



On peut supposer que les données du problème dans le cas des paquets d'erreurs sont:

$\Delta\tau$:durée du parasite:

C'est le temps maximum pendant lequel les parasites agissent sur les bits dans le canal de transmission. Ce temps va déterminer la longueur du paquet d'erreur.

($\Delta\tau$: est une valeur déterminée statistiquement)

$\mathbf{t_d}$:temps de décision. C'est le temps maximum d'attente entre deux mots consécutifs.

n:nombre de bits total dans un mot **n=m+k**.
f:nombre de mots que peut contenir la mémoire. (taille de la mémoire)
d:distance minimale entre les différents mots du code.

$\mathbf{t_b}$:le temps pour envoyer un bit.
$\mathbf{t_n}$:le temps pour envoyer un mot.
T:le temps pour envoyer tous les mots de la mémoire=$\mathbf{n.f.t_b}$.

Donc si on transmet l'information avec une vitesse **V** de telle sorte que:

$$t_b \leq \frac{t_d}{n\,f}$$

En effet:

$$T = \sum_{i=1}^{f} t_{n_i}$$

Or

$$t_{n_i} = n\,t_b$$

Donc

$$T = \sum_{i=1}^{f} n\, t_b = t_b \sum_{i=1}^{f} n = n\, f\, t_b$$

Il faut que $\mathbf{T < t_d}$

Soit encore :

$$t_b \leq \frac{t_d}{n\, f}$$

On est sûr que la condition de décision est respectée, mais il ne faut pas accélérer la vitesse de transmission des bits de telle sorte que la longueur du paquet d'erreur devienne grande et échappe à notre système de détection d'erreurs. Donc on impose une deuxième condition.

$$f\,(d\text{-}1)\, t_b \geq \Delta\tau$$

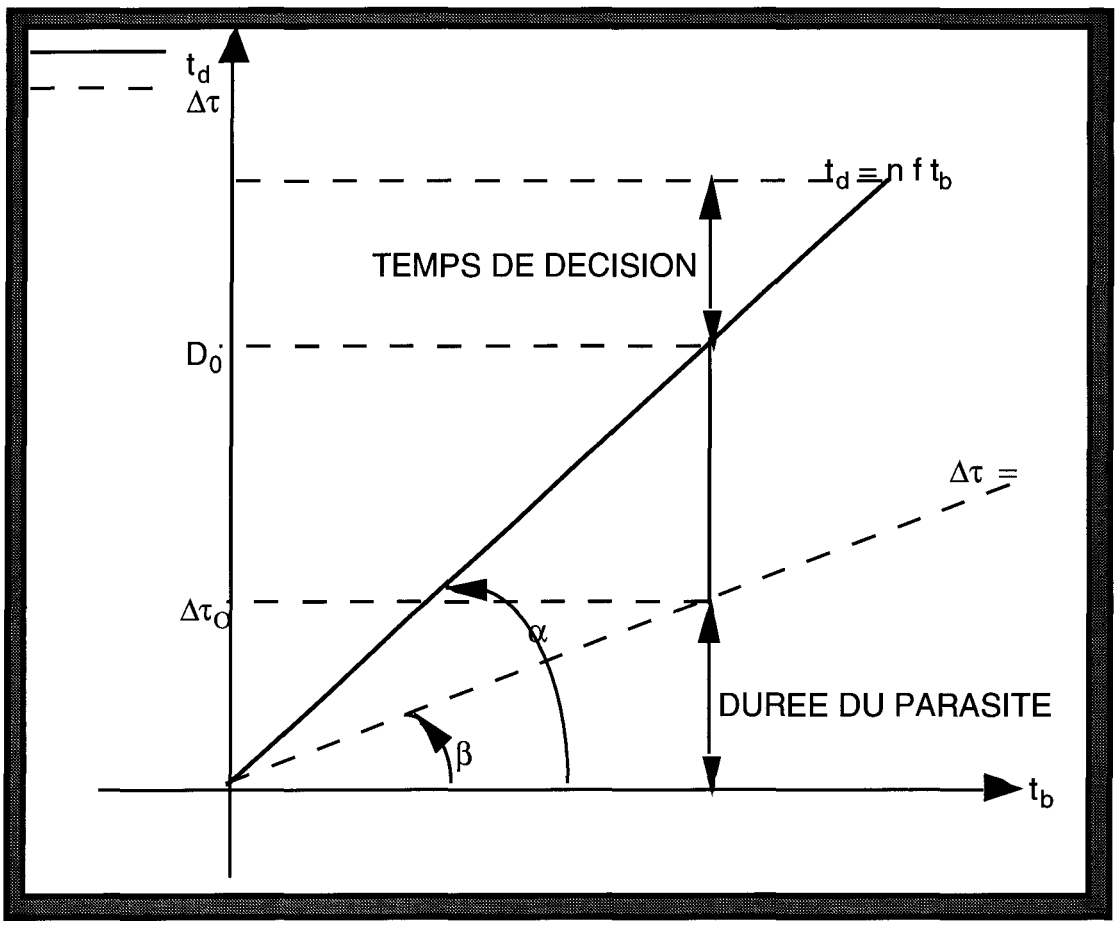
Soit encore :

$$t_b \geq \frac{\Delta\tau}{f(d\text{-}1)}$$

Donc si les deux conditions sont respectées on a:

$$\frac{\Delta\tau}{f(d\text{-}1)} \leq t_b \leq \frac{t_d}{n\, f}$$

On peut tracer graphiquement t_d et $\Delta\tau$ en fonction de t_b.



$tg\ \alpha = n\ f$

$tg\ \beta = f\ (d-1)$

Or $n > d-1$, donc $tg\alpha > tg\ \beta$

Soit encore $\alpha > \beta$

Conclusion:

Dans le cas des perturbations trop longues, on peut utiliser l'entrelacement simple des messages codés tout en respectant l'objectif de sécurité de notre système d'exploitation.

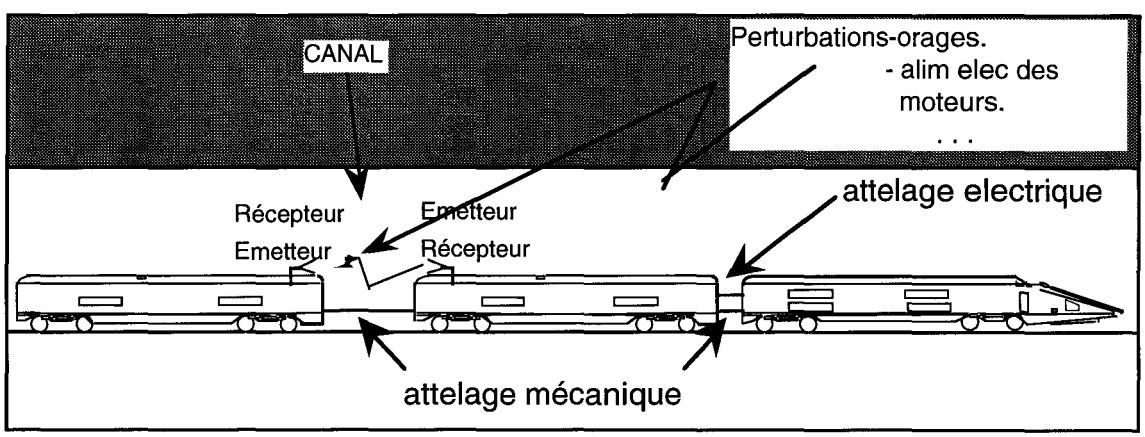
CHAPITRE IV

EVALUATION DE LA QUALITE D'UNE TRANSMISSION A HAUT DEBIT NUMERIQUE

I INTRODUCTION.

La conduite automatique dans les systèmes de transport urbain permet d'envisager des modes de fonctionnement nouveaux autorisant à la fois des économies pour l'exploitant (énergie, maintenance . . .) et une meilleure qualité de service à l'usager.

L'une des méthodes consiste à moduler la capacité offerte suivant la période de la journée en faisant varier non pas la fréquence, mais la dimension des rames en circulation. Un tel mode d'exploitation suppose que l'on puisse accoupler et désaccoupler facilement et de façon automatique les véhicules d'une rame. Pour réaliser une opération de ce type, il existe déja des coupleurs. Cependant étant donné leur technologie multibroche, ils se prêtent mal à des accouplements et désaccouplements fréquents. Cette étude, associée à une approche économique de l'exploitation d'un tel principe (non traitée ici), vise donc la possibilité de remplacer les prises multibroches par des liaisons sans contact matériel utilisant des ondes hyperfréquences comme support principalement.



D'importants travaux ont été réalisés par l'INRETS-CRESTA pour valider la qualité d'une transmission numérique à la fréquence de 23GHz pour des projets de transports public telque le projet ARAMIS. Le matériel hyperfréquence utilisé* est un dispositif d'émission-réception à modulation FSK.

*: Nous remercions l'INRETS-CRESTA et en particulier Monsieur M. HEDDEBAUT pour nous avoir prêté le matériel hyperfréquence.

La chaîne de transmission envisagée dans ce chapitre sert à l'envoi d'informations numériques de sécurité à haut débit de l'émetteur au récepteur, supposé fixe l'un par rapport à l'autre.

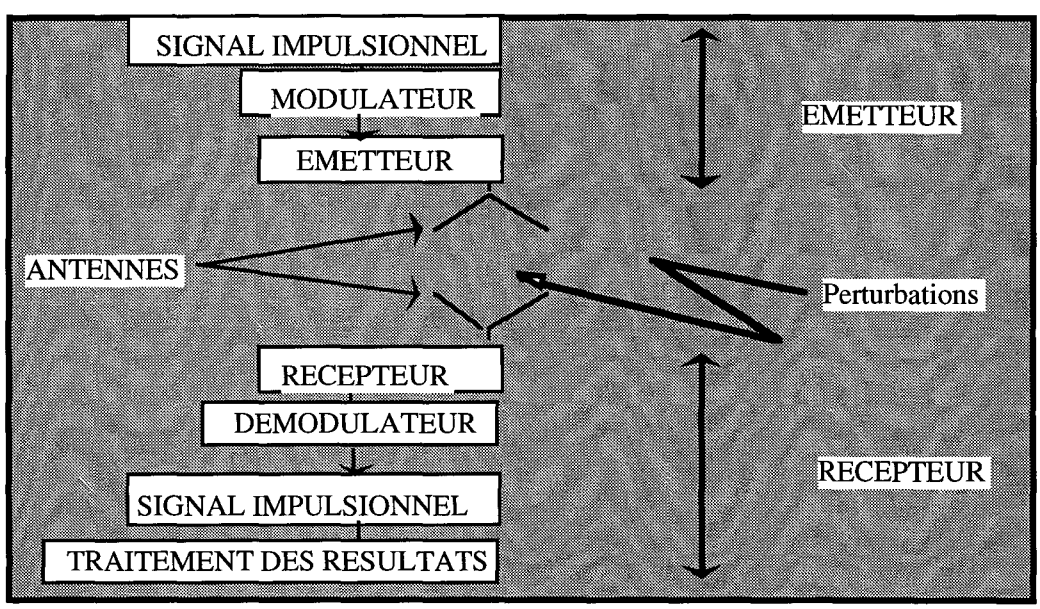
L'étude proposée consiste à rendre cette transmission en sécurité en utilisant des codes de détection d'erreur. Or, nous avons besoin de connaître le taux moyen d'erreur de cette transmission ainsi que les modèles d'erreurs prédominants afin de mettre en oeuvre un code de détection d'erreurs.

C'est pourquoi, il nous a paru judicieux de mettre en oeuvre une simulation de cette chaîne de transmission afin d'estimer le taux moyen d'erreurs dépendant des grandeurs caractéristiques de la liaison, telles que la modulation et la démodulation de l'onde porteuse.

II CHAINE DE TRANSMISSION.

Notre premier travail consiste à simuler le bruit dans la chaîne de transmission, ceci afin d'être proche de la réalité physique et d'utiliser la simulation comme aide à la conception.

Nous obtenons ainsi un schéma équivalent de la transmission simplifiée:

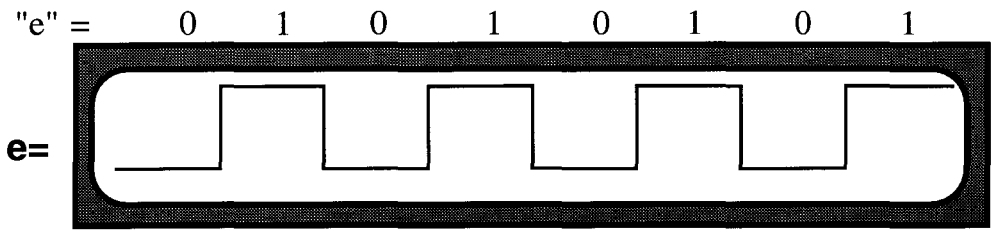


II-1 Caractéristiques du signal d'emission.

Le message à l'émission est du type binaire tel que chaque 1 est suivi d'un 0.

Le 1 logique est émis sous forme d'une impulsion et le 0 logique correspond à l'absence d'impulsion.

Informations à l'émission:



II-2 Type de modulation et démodulation.

La modulation utilisée est du type F.S.K (F.S.K: Fréquency shift keying).

A la succession d'états logiques d'entrées **(0)** et **(1)**, nous associons deux fréquences f_0 et f_1. A la sortie du modulateur, on obtient donc l'un des deux signaux suivants:

$$a(t)=(1+m\sin(2\Pi f_0 t))\sin(2\Pi F_p t)$$

Ou bien:

$$a(t)=(1+m\sin(2\Pi f_1 t))\sin(2\Pi F_p t)$$

Avec:

f_0 Fréquence associée à l'état logique **0**
f_1 Fréquence associée à l'état logique **1**
F_p Fréquence porteuse
m indice de modulation

A l'entrée du démodulateur, le signal s'écrit:

$$s(t)=(1+m\sin(2\Pi f t))\sin(2\Pi F_p t) + b(t)$$

Avec:

$f = f_0$ pour l'état logique "0"

$f = f_1$ pour l'état logique "1"

La démodulation **F.S.K** s'effectue à l'aide de deux filtres passe-bande du deuxième ordre accordés aux fréquences f_0 et f_1.

II-3 Le bruit.

Dans notre étude, nous ne considérons que le bruit de transmission, c'est à dire que le récepteur et l'émetteur ne font aucune erreur. Ce bruit sera simulé avant la modulation et vient donc se superposer au signal reçu au niveau de l'entrée du récepteur. Nous supposons que le bruit est défini comme un processus aléatoire stationnaire décrit par une fonction binaire aléatoire **b(t)**.

II-4 Traitement des résultats statistiques.

Avant de donner les résultats des erreurs de transmission, il est important de préciser la définition du rapport signal sur bruit car elle a une grande importance pour le taux d'erreur. Des résultats expérimentaux montrent que le nombre d'erreurs diminue quand le rapport signal sur bruit augmente.

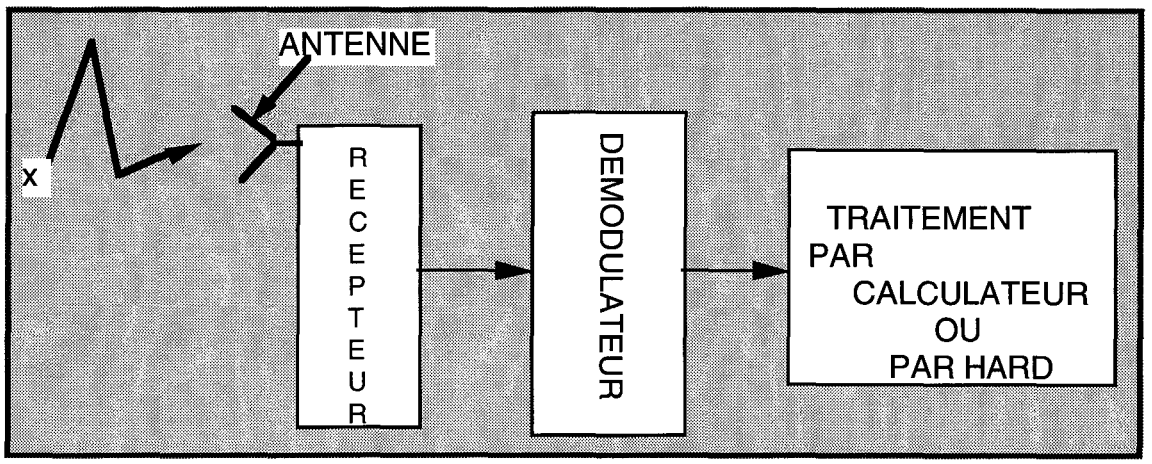
Le calcul de la probabilité d'erreur pour tout symbole (bit) émis est une propriété statistique.

Elle a pour expression:

$$P = \frac{\text{nombre d'erreurs comptées dans un intervalle de temps fixe}}{\text{nombre total de bits émis pendant le même intervalle de temps}}$$

Réception d'information:

Au niveau du récepteur on a une entrée séquentielle, $x=(x_1\ x_2\ x_3\ .\ .\ .\ .\ .\ .\ )$



III Détection des erreurs (faible débit numérique).

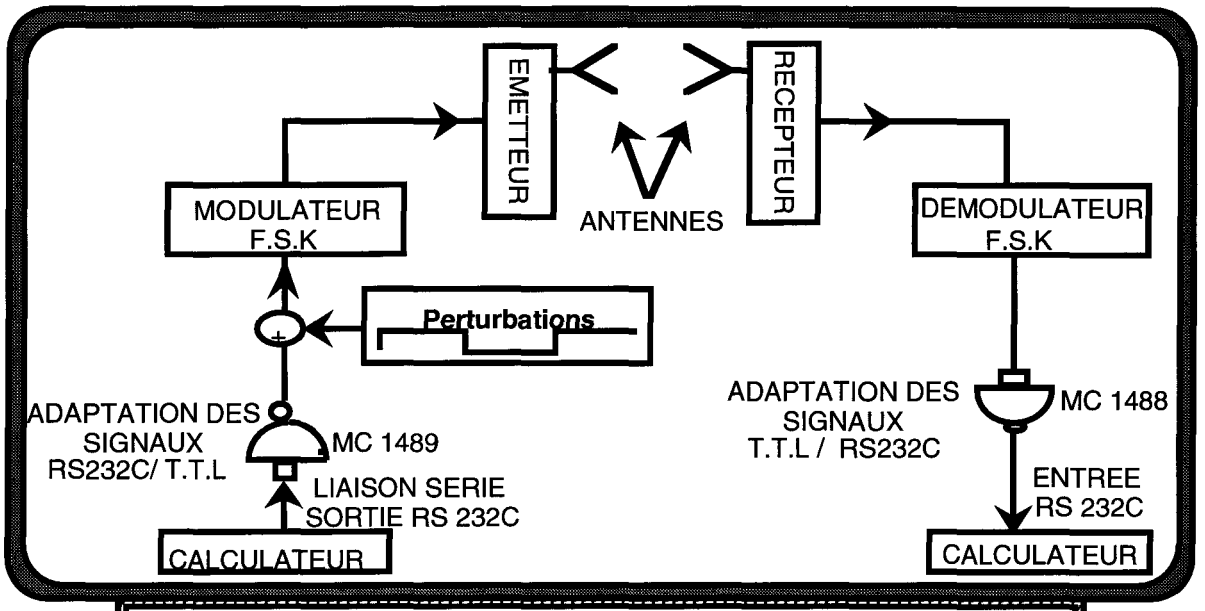
On compare le message reçu au message envoyé par calculateur. Si l'état du bit reçu est opposé à celui que nous aurions dû recevoir, nous savons alors qu'il y a erreur d'état de l'unité d'information.

(Forme du signal logique: la sortie série **RS232C** code le **1** logique par **-12 volts** et le **0** logique par **+12 volts.** La transmission des caractères en code **ASCII** se fait poids faible en tête).

III-1 Evaluation du taux d'erreurs d'une ligne de transmision à 23GHz.

L'evaluation du taux d'erreurs a été faite dans les conditions expérimentales suivantes:

-les essais sont effectués en mode statique, c'est à dire que l'émetteur est fixe par rapport au récepteur.
-distance entre l'émetteur et le récepteur=**1 m**
-vitesse de transmission=**19200 bits/s**
-durée de transmission=**10s**



CHEMA DU BANC DE MESURES DU TAUX D'ERREUR DE LA TRANSMISSION NUMERIQUE.

III-2 Taux d'erreurs en fonction de la fréquence des parasites.

DUREE DU PARASITE	NOMBRE DE BITS TRANS	NOMBRE DE BITS FAUX	TAUX D'ERREURS
104 µs	192000	3	$1,56\ 10^{-5}$
208 µs	192000	5	$2,60\ 10^{-5}$
416 µs	192000	9	$4,69\ 10^{-5}$
832 µs	192000	17	$8,85\ 10^{-5}$

Ces résultats sont intéressants car ils montrent la validité de l'utilisation de la méthode pour évaluer le taux d'erreurs d'une ligne de transmission à faible débit numérique.

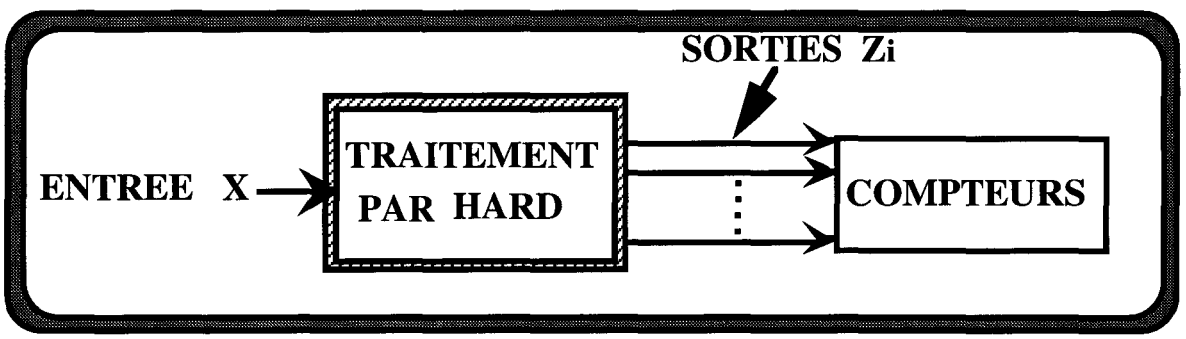
La vitesse de transmission doit être inférieure à la vitesse de réception maximale de la carte série **RS232C** (dans notre cas **19200 Bauds**) du calculateur.

C'est pour cela qu'il nous a paru judicieux d'utiliser des automates programmables à l'aide de circuits intégrés pour des transmissions numériques à haut débit.

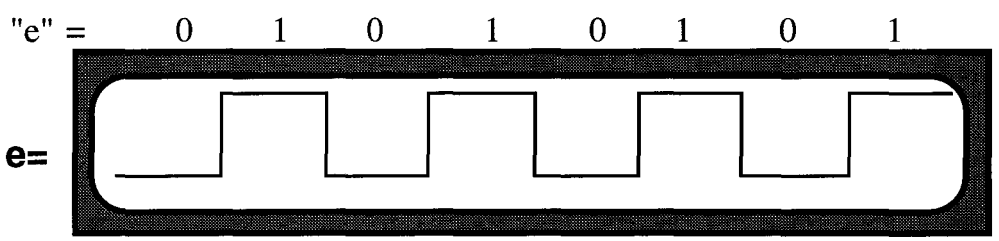
IV Détection des erreurs (haut débit numérique).

IV-1 Conditions expérimentales.

L'étude proposée consiste à connaître le taux moyen d'erreur d'une transmission numérique à haut débit ainsi que les modèles d'erreurs prédominants. C'est pourquoi, il nous a paru judicieux d'utiliser une information simple à l'émission (signal carré) pour simplifier le traitement par HARD.



e: Informations à l'émission:



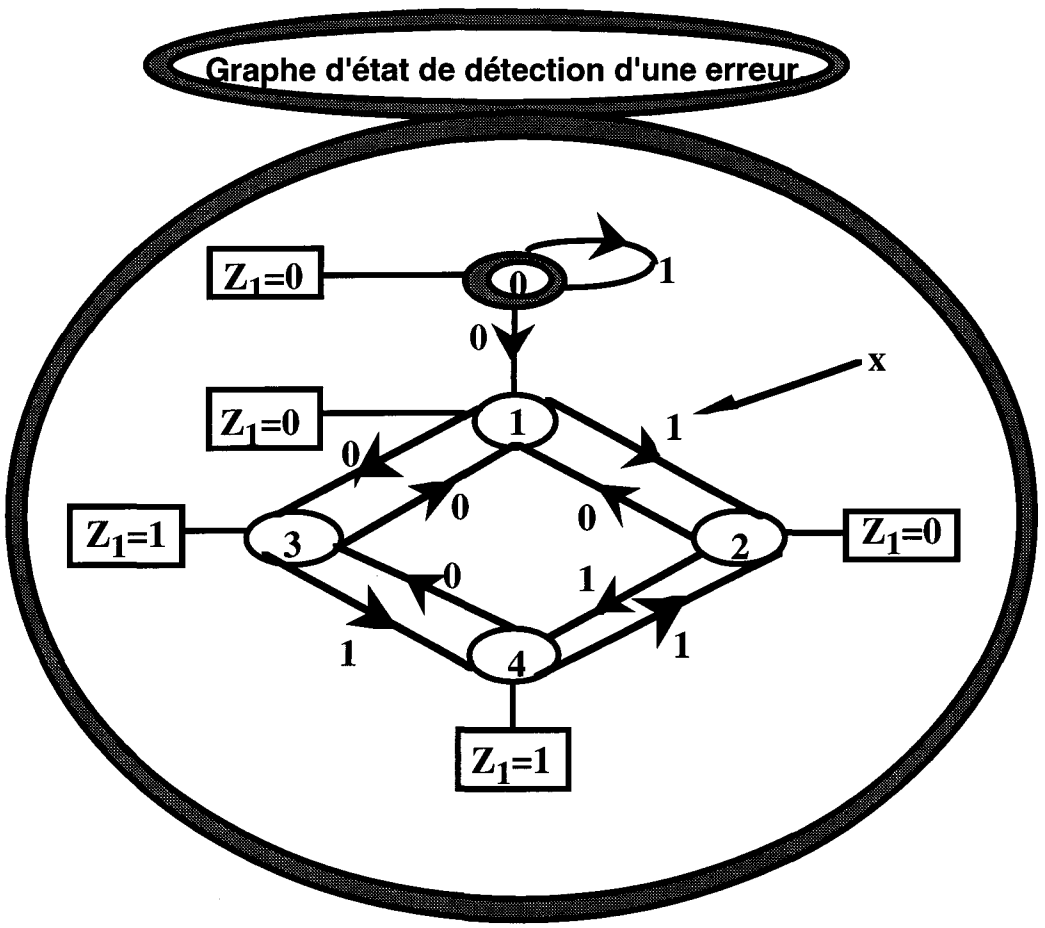
Le message à l'émission est tel que chaque **1** est suivi d'un **0.** (signal carré). A la réception ce message est entaché de bruit.

$$x = e + b$$

La détection d'erreur se fait de la manière suivante:

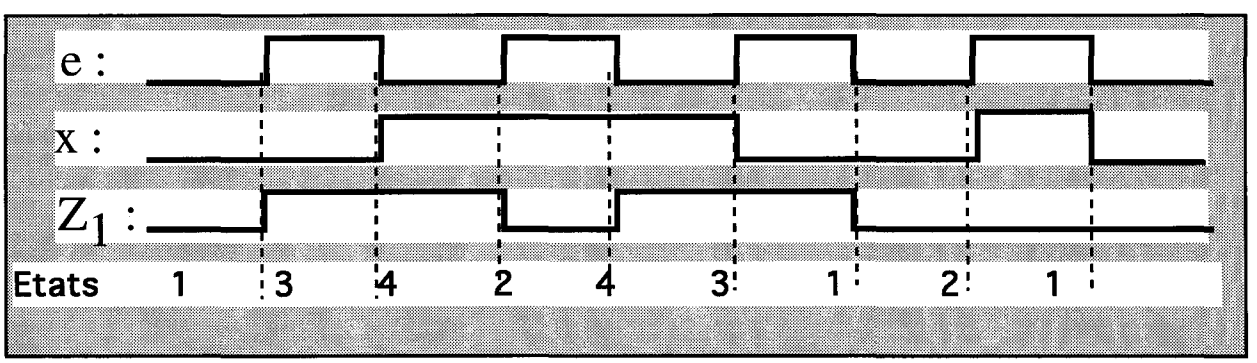
La sortie Z_1 de l'automate programmable se met à **1** à chaque fois que $e \neq x$, autrement dit qu'il y a une erreur sur l'information.

IV-2 Détection des erreurs individuelles.



Exemple d'un message reçu erroné.

On pose:
e : message à l'émission
x : message reçu **x=e+b** (**b**: bruit du canal de transmission).
$\mathbf{Z_1}$: sortie du détecteur d'erreurs



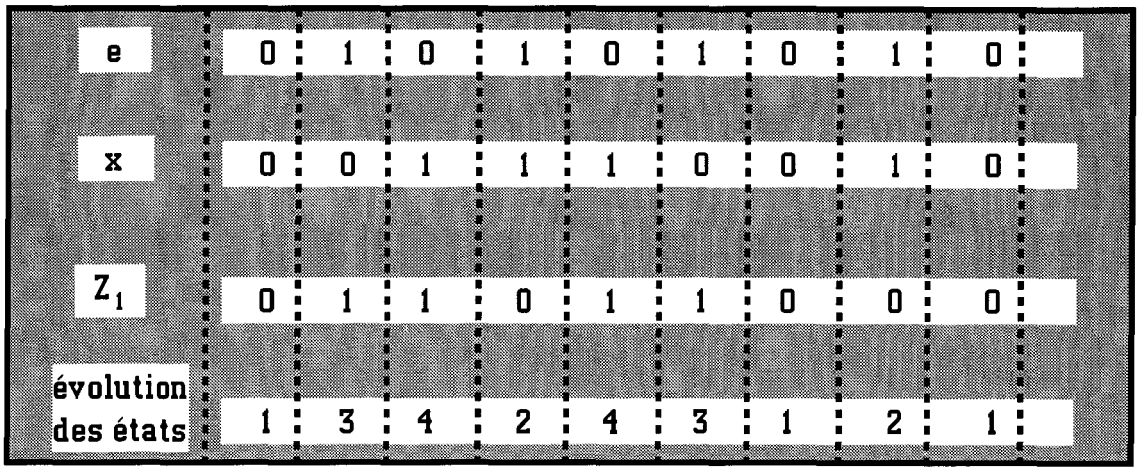


Table des états:

message reçu

x

Etats présents	0		1	
	Etats futurs	Sortie Z_1	Etats futurs	Sortie Z_1
0	1	0	0	0
1	3	1	2	0
2	1	0	4	1
3	1	0	4	1
4	3	1	2	0

Réalisation du détecteur d'une erreur à l'aide des bascules JK.

Code des états.

Le nombre d'états est égal à **5**, il faut donc **3** bascules ($\mathbf{Log_2(5)<3}$) pour coder les **5** états.

Code des états

états	Q_1	Q_2	Q_3
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0

Table des états codés

Q_1	Q_2	Q_3	0	1
0	0	0	0 0 1	0 0 0
0	0	1	0 1 1	0 1 0
0	1	0	0 0 1	1 0 0
0	1	1	0 0 1	1 0 0
1	0	0	0 1 1	0 1 0

Table logique des bascules J-K

x → 0 / 1

Q_3	Q_2	Q_1	J_3 K_3	J_2 K_2	J_1 K_1	J_3 K_3	J_2 K_2	J_1 K_1
0	0	0	0 --	0 --	1 --	0 --	-- 0	0 --
0	0	1	0 --	1 --	-- 0	0 --	1 --	-- 1
0	1	0	0 --	-- 1	1 --	1 --	-- 1	0 --
0	1	1	0 --	-- 1	-- 0	1 --	-- 1	-- 1
1	0	0	-- 1	1 --	1 --	-- 1	1 --	0 --

Equations logiques simplifiées (en utilisant le tableau de Karnaugh)

$K_1 = x$ $\qquad K_2 = 1$ $\qquad K_3 = 1$

$J_1 = \overline{x}$ $\qquad J_2 = Q_1 + Q_3$ $\qquad J_3 = x \cdot Q_2$

Equation de la sortie Z_1:

$Z_1 = \overline{Q_3} \cdot Q_2 \cdot Q_1 + Q_3 \cdot \overline{Q_2} \cdot Q_1$

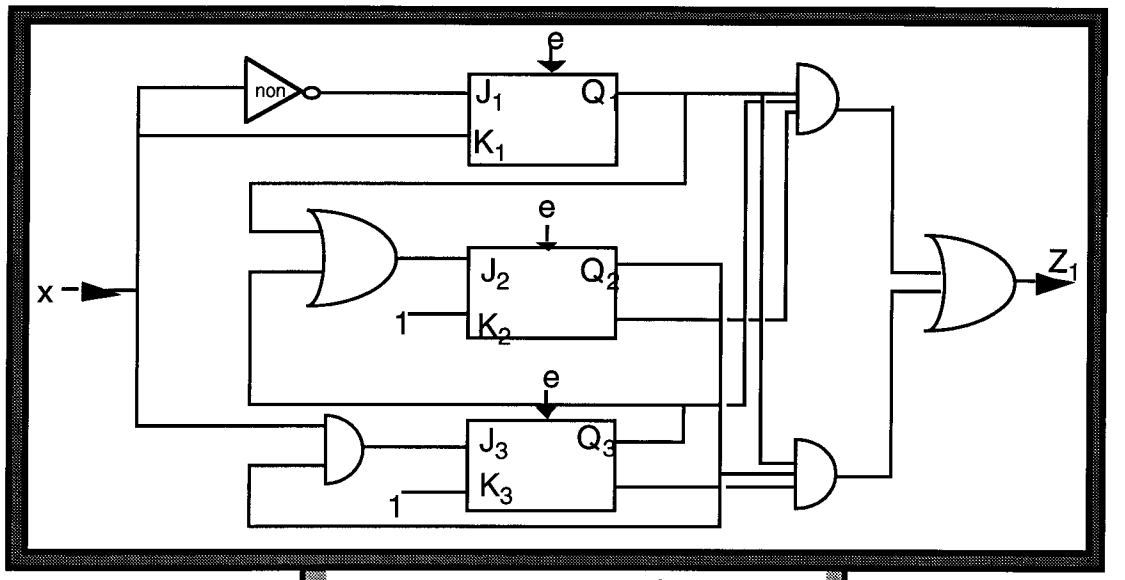


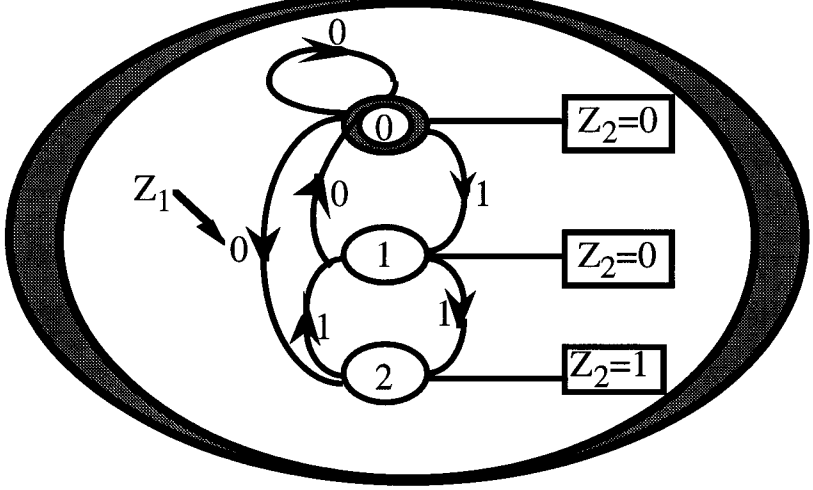
Schéma du détecteur d'une erreur

(x : signal à analyser e : signal réference)

IV-3 Paquets d'erreurs de longueur 2.

Détection des paquets d'erreurs de longueur 2: La sortie du détecteur des paquets d'erreurs de longueur 2 se met à 1 à chaque fois qu'il y a deux erreurs consécutives. Autrement dit à chaque fois que Z_1 se met à 1 deux fois de suite.

Graphe d'état du détecteur des paquets d'erreurs de longueur 2



Réalisation du "DP2" à l'aide des bascules JK: Le nombre d'etats est égal à 3, il faut donc 2 bascules pour réaliser le détecteur des paquets d'erreurs de longueur 2. En effet $Log_2(3)<2$.

Table des états:

entrée Z_1

Etats présents	0		1	
	Etats futurs	Sortie Z_2	Etats futurs	Sortie Z_2
0	0	0	1	0
1	0	0	2	1
2	0	0	1	0

Code des états

états	Q_1	Q_2
0	0	0
1	0	1
2	1	1

Table des états codés

Q1	Q2	0		1	
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	0	1

Table logique des bascules J-K

Z_1 : 0 | 1

Q1	Q2	J2	K2	J1	K1	J2	K2	J1	K1
0	0	0	--	0	--	0	--	1	--
0	1	0	--	--	1	1	--	--	1
1	0	--	1	0	--	--	1	1	--

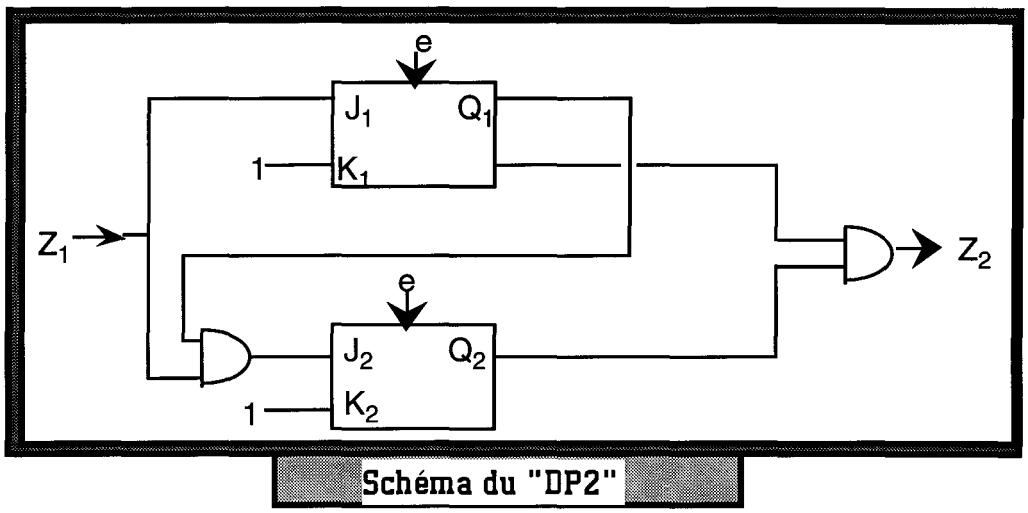
Equations logiques simplifiées (en utilisant le tableau de Karnaugh)

$K_1 = 1$ $\qquad$ $K_2 = 1$

$J_1 = Z_1$ $\qquad$ $J_2 = Z_1 \cdot Q_1$

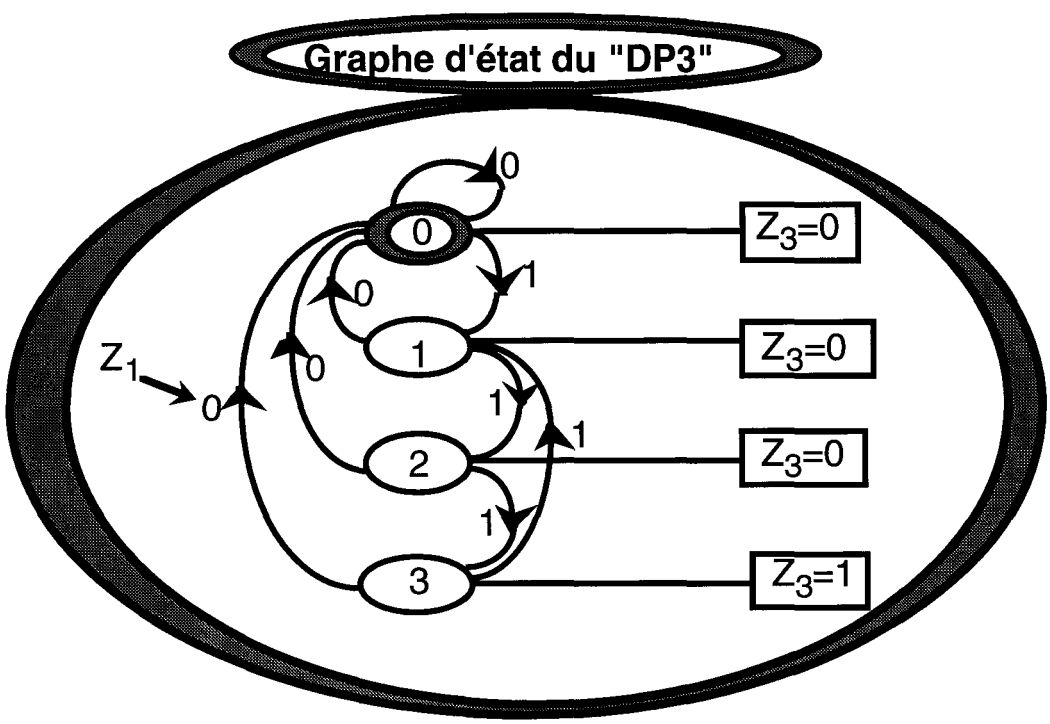
Equation de la sortie Z_2

$Z_2 = Q_2 \cdot \overline{Q_1}$



IV-4 Paquets d'erreurs de longueur 3.

Détection des paquets d'erreurs de longueur 3: la sortie Z_3 du détecteur des paquets d'erreurs de longueur 3 se met à 1 à chaque fois qu'il y a trois erreurs consécutives.



Réalisation du "DP3" à l'aide des bascules JK: Le nombre d'états est égal à 4, il faut donc 2 bascules pour réaliser le"DP3". En effet $Log_2(4)=2$.

Table des états:

entrée Z_1

Etats présents	Etats futurs (Z_1=0)	Sortie Z_3 (Z_1=0)	Etats futurs (Z_1=1)	Sortie Z_3 (Z_1=1)
0	0	0	1	0
1	0	0	2	0
2	0	0	3	1
3	0	0	1	0

Code des états

états	Q_1	Q_2
0	0	0
1	0	1
2	1	0
3	1	1

Tables des états codés

Q_1	Q_2	Z_1=0	Z_1=1
0	0	0 0	0 1
0	1	0 0	1 0
1	0	0 0	1 1
1	1	0 0	0 1

Table logique des bascules J-K

Q_1	Q_2	Z_1=0: J_2	K_2	J_1	K_1	Z_1=1: J_2	K_2	J_1	K_1
0	0	0	--	0	--	0	--	1	--
0	1	0	--	--	1	1	--	--	1
1	0	--	1	0	--	--	0	1	--
1	1	--	1	--	1	--	1	--	0

Equations logiques simplifiées (en utilisant le tableau de Karnaugh)

$J_1 = Z_1$ $\qquad$ $J_2 = Z_1 \cdot Q_1$

$K_1 = \overline{Z_1 \cdot Q_2}$ $\qquad$ $K_2 = \overline{Z_1 \cdot Q_1}$

Equation de la sortie Z_3

$Z_3 = Q_1 \cdot Q_2$

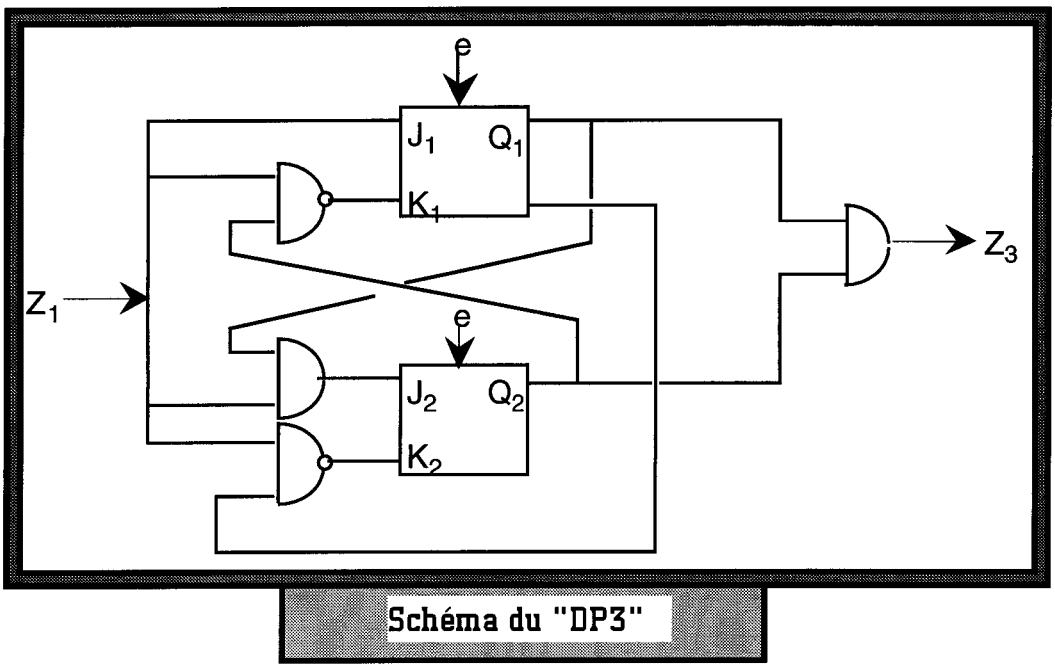


Schéma du "DP3"

On obtient de cette manière les détecteurs d'erreurs de longueur un nombre premier.

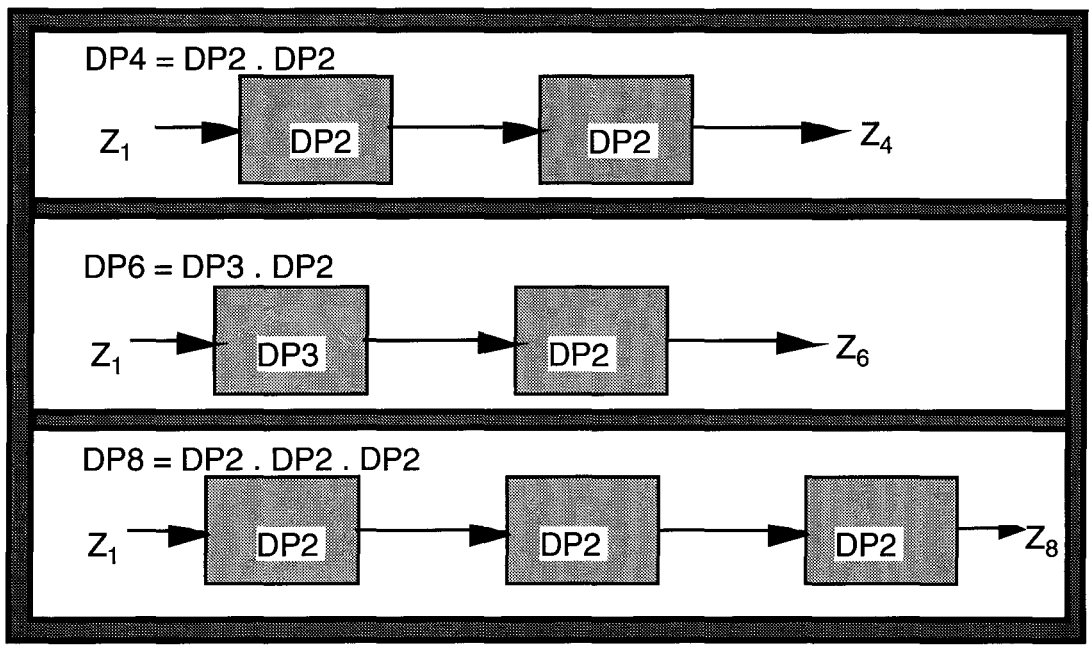
IV-5 Paquets d'erreurs de longueur n.

Détection des paquets d'erreurs de longueur n.

On peut réaliser les détecteurs des paquets d'erreurs de longueur un nombre non premier par combinaison de nombres premiers.

Exemples:

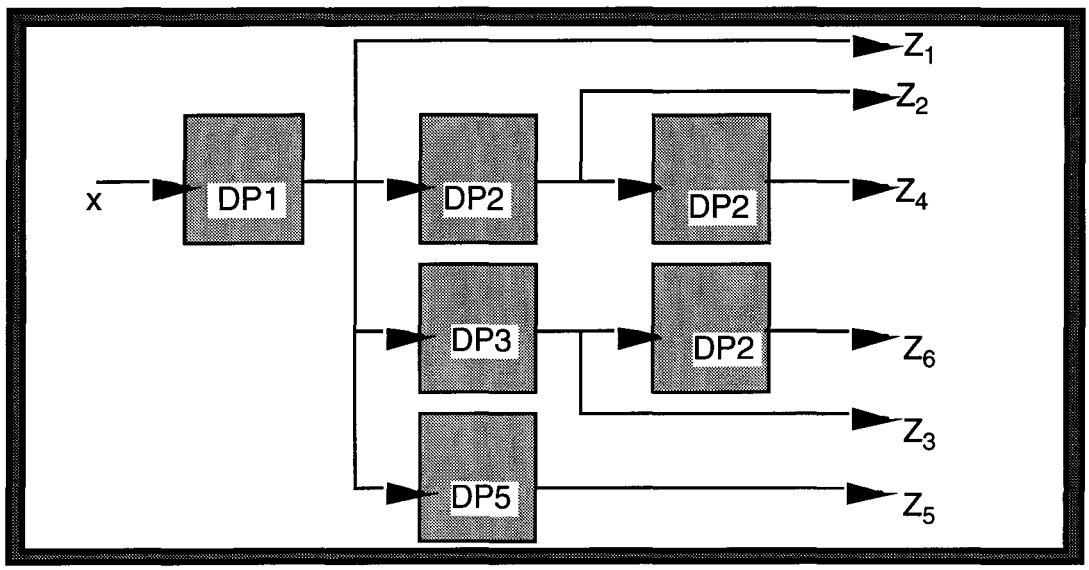
Détecteurs des paquets d'erreurs de longueur 4, 6 et 8 (voir schéma ci-après).



On peut ainsi obtenir un détecteur des paquets d'erreurs de longueur 121 par combinaison de deux détecteurs des paquets d'erreurs de longueur 11.

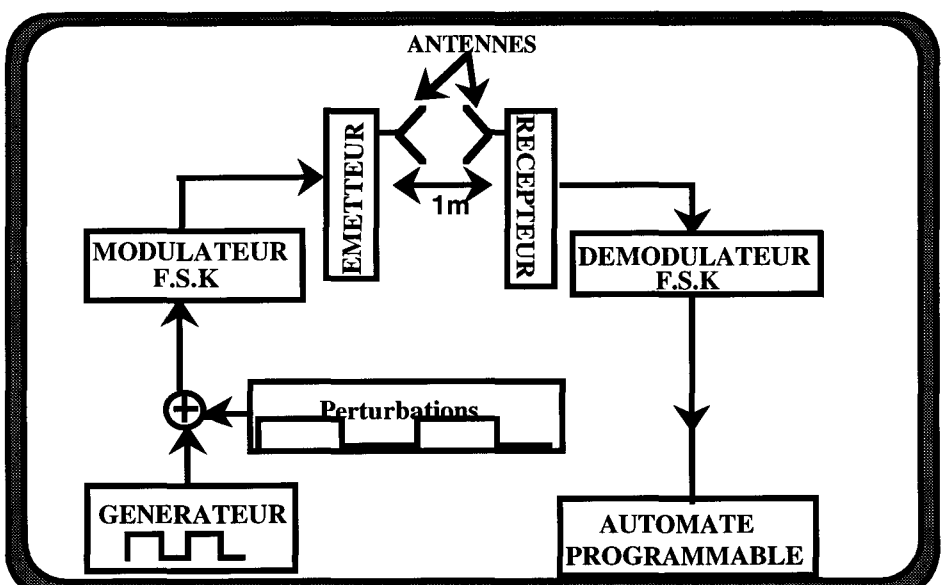
DPn	2	3	5	7	11
2	4	6	10	14	22
3	6	9	15	21	33
5	10	15	25	35	55
7	14	21	35	49	77
11	22	33	55	77	121

Exemple: détection des paquets d'erreurs de longueur inférieure ou égale à 6.



Si $Z_6=1$ alors $Z_5=1$, $Z_4=1$, $Z_3=2$, $Z_2=3$ et $Z_1=6$
Si $Z_5=1$ alors $Z_4=1$, $Z_3=1$, $Z_2=2$ et $Z_1=5$
Si $Z_4=1$ alors $Z_3=1$, $Z_2=2$ et$Z_1=4$
Si $Z_3=1$ alors $Z_2=1$ et $Z_1 = 3$
Si $Z_2=1$ alors $Z_1=2$

IV-6 Evaluation du taux et du types d'erreurs d'une ligne de transmision à 23 GHz



SCHEMA DU BANC DE MESURES DU TAUX D'ERREUR DE LA TRANSMISSION NUMERIQUE

L'evaluation du taux d'erreurs a été faite dans les conditions expérimentales suivantes:

-les essais sont effectués en mode statique, c'est à dire que l'émetteur est fixe par rapport au récepteur.

-distance entre l'émetteur et le récepteur = 1 m

-vitesse de transmission = 1 Mbits/s

DUREE DU PARASITE	COMPTEUR DE Z_1	COMPTEUR DE Z_2	COMPTEUR DE Z_3
104 µs	105	52	35
208 µs	209	104	69
416 µs	417	208	139

Ces résultats sont intéressants car il montrent la validité de l'utilisation des automates programmables pour evaluer le taux et le type d'erreurs d'une ligne de transmission à haut débit numérique.

CONCLUSION

L'étude que nous venons de présenter a mis en évidence la possibilité de définir les caractéristiques générales d'un code de détection d'erreurs (distance minimale et nombre de bits de contrôle) sans faire d'hypothèses sur la nature du code proprement dit, mais uniquement en imposant un objectif de sécurité et en connaissant les caractéristiques du canal de transmission.

Les résultats obtenus par la méthode générale mise au point ont été testés dans le cas particulier des codes générés par un polynôme. Nous avons également mis en évidence que dans le cas des codes générés par un polynôme il était possible d'optimiser le polynôme générateur en abaissant son degré et par la même raccourcir la longueur des mots codes.

Pour un nombre de bits de contrôles minimum, il existe un codage qui permet de trouver la plus faible probabilité de non détection d'erreurs.

Pour lutter contre les paquets d'erreurs nous avons présenté deux méthodes, la première consiste à regarder la capacité de détection des paquets d'erreurs du code choisi, la deuxième utilise l'entrelacement des messages codes avant la transmission.

Tous les résultats d'encodage étant basés sur la connaissance du type de perturbation des bits dans le canal de transmission, nous avons présenté deux méthodes d'analyse, l'une en faible vitesse basée sur une analyse logicielle, l'autre à haut débit numérique basée sur la réalisation matérielle d'un automate programmable.

Nous avons utilisé les conclusions de cette étude pour deux applications dans le domaine des transports et nous pensons qu'elles peuvent être utilisées pour toute transmission utilisant des informations de sécurité.

ANNEXE I

EXEMPLES DE CODES LINEAIRES: CODES DE HAMMING.

I DEFINITION.

Les codes de hamming sont des codes linéaires L(n,m), de distance minimale 3, définis par:

$n=(2^k-1)$, k étant le nombre de bits de contrôle.

$m=n-k$, m étant le nombre de bits d'information.

Ces codes sont capables de corriger une erreur simple et de détecter 2 erreurs.

Les colonnes de la matrice de contrôle (H) satisfont aux relations:

$h_{io} \neq 0$ pour $i_o = 0, ..., (n-1)$.

$h_{io} + h_{i1} \neq 0$ pour $i_o, i_1 = 0, ... , (n-1)$

$i_o = i_1$.

Pour cela, il suffit de donner aux colonnes de la matrice (H) les valeurs 1, 2, ..., jusqu'à n, en représentation binaire sur k positions.

II EXEMPLE.

Construction d'un code de Hamming L(7,4).

$n=2^3-1=7$

$m=7-3=4$

La matrice de contrôle est de la forme:

$$(H) = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Pour avoir un code de Hamming systématique, on remplace la matrice (H) par (Hs).

$$(H_S) = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

La matrice génératrice associée à (Hs) est:

$$(G_S) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

III CODAGES.

Un mot $<N_i>$ est de la forme :

$$<N_i> = <a_0\ a_1\ a_2\ a_3\ c_0\ c_1\ c_2>.$$

Pour obtenir les mots du code $<N_i>$, on multiplie les mots d'information $<M_i>$ par la matrice **(Gs)**.

Soit à coder le mot **<1000>**, on a:

$$<N_i>=<M_i>(Gs).$$

On trouve alors le mot suivant:

$$<1000110>.$$

IV DECODAGE.

A la réception, on obtient un mot $<N_i'>=<N_i>+<E_i>$.

On étudie les deux cas suivants:

1°) $<E_i>=0$ $<N_i'>=<1000110>=<N_i>$

On calcule le syndrome d' erreurs:

$$<S_i>=<N_i'>(H)^t.$$

Si on trouve $<S_i>=<0>$ alors le mot reçu est correct. .

2°) $<Ei>\neq 0$, $<E_i>=<0010000>$

$$<N_i'>=<1010110>.$$

On calcule le syndrome d'erreurs:

$$<S_i>=<N_i'>(H)^t=<011>.$$

Si on trouve $<S_i>\neq<O>$ alors le mot reçu est erroné.

ANNEXE II

I DEFINITION.

Un polynôme **f(x)** de degré **N** qui n'est divisible par aucun polynôme de degré inférieur à N dans le même corps est dit **irréductible.**

II NOMBRE DE POLYNOMES IRREDUCTIBLES.

Le nombre **Nb(r)** de polynômes irréductibles de degré r dans un corps de Galois 2, **(CG(2))** satisfait à la relation:

$$\sum_{r=1}^{r=N} r.Nb(r) = 2^N .$$

III DECOMPOSITION DE (x^n+1) EN POLYNOMES IRREDUCTIBLES.

Les racines de (x^n+1) sont des racines de l'unité. Si n est pair, (n=2p) alors :

$$x^n+1 = x^{2p}+1$$

$$= (x^p+1)(x^p+1) \text{ dans } CG(2).$$

Donc l'étude de (x^n+1) se ramène à celle pour laquelle n est impair.

Soit un polynôme **f(x)**, irréductible, de degré **m** et qui divise (x^n+1), **n** impair.

Ses m racines sont:

$$\beta^{2^0}, \beta^{2^1}, \ldots, \beta^{2^{m-1}}$$

et qui sont racines de (x^n+1).

Autrement dit, elles appartiennent à l'ensemble des racines de (x^n+1), soit:

$$S = \{ \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{n-1} \}$$

Donc

$$\beta = \alpha^j \Rightarrow \beta^2 = \alpha^{2j} \Rightarrow \beta^{2^i} = \alpha^{2^i j} = \alpha^r$$

r étant le reste de la division de $\mathbf{j.2^i}$ par **n**.

Pour la pratique de la décomposition de (x^n+1), on débute par **j=1**. On divise par **n** les puissances successives de $\mathbf{j.2^i}$. On arrête la division lorsque l'on obtient un reste déjà trouvé.

Les racines $\alpha^{r_1}, \alpha^{r_2} \cdots$ etc obtenues sont celles d'un polynôme partiel cherché.

On recommence l'opération pour une nouvelle valeur de **j** qui donne des restes différents de ceux déjà trouvés.

Exemple: Décomposition de (x^7+1) en polynôme irréductible.

j=1 $\quad 1.2^0 = 1 \quad$ **modulo 7**

$1.2^1 = 2 \quad$ **modulo 7**

$1.2^2 = 4 \quad$ **modulo 7**

$1.2^3 = 1 \quad$ **modulo 7**

Ce qui donne le polynôme cherché $\mathbf{f_1(x)}$:

$f_1(x)=(x+\alpha)(x+\alpha^2)(x+\alpha^4)$, de degré 3 .

Puis on passe à:

j=3 **$3.2^0 = 3$ modulo 7**

$3.2^1 = 6$ modulo 7

$3.2^2 = 5$ modulo 7

$3.2^3 = 3$ modulo 7

Ce qui donne le polynôme cherché $\mathbf{f_3(x)}$:

$f_3(x)=(x+\alpha^3)(x+\alpha^6)(x+\alpha^5)$, de degré 3

Donc : $(x^7+1)=f_1(x)f_3(x)(x+1)$.

$\mathbf{f_1(x)}$ s'écrit:

$$\mathbf{f_1(x)=x^3+Ax^2+Bx+1.}$$

Pour calculer les coefficients A et B, on cherche les restes de la division de $\mathbf{x^n}$, $\mathbf{(0<n<7)}$ par $\mathbf{x^3+x^2+1.}$

On trouve : **A = 1** et **B =0**, d'ou:

$$\mathbf{f_1(x) = x^3+x^2+1.}$$

De même, pour $\mathbf{f_3(x) = x^3+A'x^2+B'x+1}$,

On trouve:

A'=0 et **B'=1**, d'ou:

$$\mathbf{f_3(x)=x^3+x+1.}$$

$$\mathbf{(x^7+1)=(x+1)(x^3+x+1)(x^3+x^2+1)}$$
$$\mathbf{=(x+1)(x^6+x^5+x^4+x^3+x^2+x+1)}$$

$= A(x)B(x)$

$= (x^3+x^2+1)(x^4+x^3+x^2+1) = C(x)D(x).$

Les quatres polynômes **A(x), B(x), C(x)** et **D(x)** sont des polynômes générateurs de quatre codes cycliques différents:

A(x) pour le code cyclique **C(7,6)**,
B(x) pour le code cyclique **C(7,1)**,
C(x) pour le code cyclique **C(7,4)**,
D(x) pour le code cyclique **C(7,3)**,

IV EXEMPLE DU CODE CYCLIQUE C(7,3).

Il y a **3** bits d'information et **4** bits de contrôle.

$g(x) = x^4+x^3+x^2+1.$

Pour un code **C(7,3)** non systématique, les mots codes sont:

Bits de message	Polynôme associé N(x)	Polynôme du code N(x).g(x) =M(x)	Mots codes
0 0 1	1	$x^4+x^3+x^2+1$	0011101
0 1 0	x	$x^5+x^4+x^3+x$	0111010
0 1 1	x + 1	x^5+x^2+x+1	0100111
1 0 0	x^2	$x^6+x^5+x^4+x^2$	1110100
1 0 1	$x^2 + 1$	$x^6+x^5+x^3+1$	1101001
1 1 0	x2 + x	$x^6+x^3+x^2+x$	1001110
1 1 1	x2+x +1	x^6+x^4+x+1	1010011

Pour un code C(7,3) systématique, les mots codes sont:

Bits de message	$x^4.N(x)$	$C(x) = x^4.N(x)$ modulo g(x)	Polynôme du code	Mots codes
001	x^4	x^3+x^2+1	$x^4+x^3+x^2+1$	0011101
010	x^5	x^2+x+1	x^5+x^2+x+1	0100111
011	x^5+x^4	x^3+x	$x^5+x^4+x^3+x$	0111010
100	x^6	x^3+x^2+x	$x^6+x^3+x^2+x$	1001110
101	x^6+x^4	x+1	x^6+x^4+x+1	1010011
110	x^6+x^5	x3+1	$x^6+x^5+x^3+1$	1101001
111	$x^6+x^5+x^4$	x^2	$x^6+x^5+x^4+x^2$	1110100

Décodage:

Supposons que l'on ait émis un mot du code:

$$<N_i> = <0111010>.$$

A la réception on obtient un mot :

$$<N_i'>=<N_i>+<E_i>.$$

Deux cas peuvent se présenter:

a) $<E_i>=0$ $\quad <Ni'> = <0111010>.$

Le polynôme associé est : $x^5+x^4+x^3+x.$

On calcule le syndrome d'erreurs par:

$$S(x) = \text{Reste}\left\{\frac{x^5 + x^4 + x^3 + x}{x^4 + x^3 + x^2 + 1}\right\} = 0$$

Puisque **S(x)=0** alors le mot reçu est correct.

b) $<E_i> \neq <0>$ $<N_i'> = <N_i> + <E_i>$

Si on suppose que **$<E_i> = <0010000>$** alors:

$<N_i'> = <0101010>$, son polynôme associé est:

$N'(x) = x^5 + x^3 + x$.

On calcule le syndrome d'erreurs S(x) défini par:

$$S(x) = \text{Reste}\left\{\frac{x^5 + x^3 + x}{x^4 + x^3 + x^2 + 1}\right\}$$

$$= x^3 + x^2 + 1.$$

Puisque $S(x) \neq 0$, le mot reçu est erroné.

ANNEXE III

MESURE DES QUALITES D'UNE TRANSMISSION DE DONNEES UTILISANT UN CODE DE DETECTION

I NECESSITE DE LA DETECTION.

Quelle que soit la qualité de la transmission d'information, il est inévitable qu'à la réception, quelques symboles, (bits), binaires soient altérés.

D'ou la nécessité d'un code redondant pour la détection d'erreurs.

II RAPPELS SUR LA MESURE DES QUALITES DE TRANSMISSION.

Pour mesurer les qualités d'une transmission de données, il est nécessaire de définir les grandeurs suivantes, qui sont relatives au code et au canal.

II-1 Efficacité E d'un code de détection d'erreurs.

L'efficacité E d'un code est le rapport moyen du nombre de messages faux et détectés au nombre total de messages faux.

II-2 Taux d'erreur brut τ

C'est la proportion moyenne de message faux reçus qu'ils soient détectés ou non.

Le taux d'erreur brut mesure la qualité intrinsèque de la transmission et il est indépendant du code de détection d'erreurs choisi. Par contre, il dépend de la loi de probabilité de répartition des erreurs sur le canal.

Si ces erreurs sont indépendantes et ont pour probabilité d'erreur P par bit sur un message de longueur n bits, on a: $\tau = \mathbf{n.\ P}$

II-3 Taux d'erreur global par message.

Ce taux d'erreur global PND est défini comme étant la proportion de messages finalement faux après épuisement de la procédure de détection. Il mesure la qualité finale de la transmission, c'est à dire celle qui intéresse l'utilisateur.

Donc PND est la garantie que nous avons de ne pas laisser passer un message faux.

Il existe une relation liant l'efficacité E, le taux d'erreur brut τ et le taux d'erreur global PND:

$$PND=\tau(1-E)+\tau^2E.(1-E)+\tau^3E^2.(1-E)$$

$$PND\cong\tau(1-E), \text{ pour E voisine de 1.}$$

III CARACTERISTIQUES DES CODES DE DETECTION D'ERREURS.

III-1 Notion de poids et de distance.

III-1-1 Poids d'un message.

On appelle poids d'un message, le nombre de "1" qu'il contient.

III-1-2 Distance entre deux messages: Distance de Hamming.

On appelle distance $d(V_i, V_j)$ entre 2 messages V_i et V_j, la quantité:

$$d(v_i,v_j) = \sum_{m=1}^{m=n} (a_{im} \oplus a_{jm})$$

$\oplus$ étant l'addition modulo 2

avec $V_i = <a_{i1}...a_{im}...a_{in}>$

$V_j = <a_{j1}...a_{jm}...a_{jn}>$.

On note d la distance minimal entre les mots d'un code.

La distance minimale entre les mots d'un code est un paramètre qui conditionne la capacité de détection et de correction d'un code.

III-2 Relation entre la distance minimale entre les mots d'un code et le nombre de corrections ou de détection d'erreurs dans un mot code.

III-2-1 Détection des erreurs.

pour détecter t erreurs, il faut que la distance minimale entre les mots du code soit: $d=t+1$

III-2-2 Correction des erreurs.

Un code est capable de corriger e erreurs si la distance minimale entre les mots du code d est au moins égale à: $d=2e+1$.

III-3 Relation entre le nombre de bits de contrôle et le nombre de bits d'information.

III-3-1 Condition nécessaire.

Pour corriger e erreurs, il est nécessaire d'avoir:

$$2^{n-m} \geq \sum_{i=0}^{i=e} c_n^i$$

C'est la marge inférieure de hamming.

m est le nombre de bits d'information.

n est le nombre total des bits dans un mot du code.

III-3-2 Condition suffisante

Pour corriger e erreurs, il suffit d'avoir:

$$2^k \geq \sum_{i=o}^{i=2e-1} c_n$$

C'est la marge de warchanov -Gilbert.

Cette condition n'est pas nécessaire.

IV-LES DIFFERENTES SORTES D'ERREURS.

IV-1 Les erreurs individuelles:

Si on suppose que chaque symbole transmis est affecté de manière indépendante par les perturbations, les erreurs qui apparaissent seront indépendantes les unes des autres.

IV-2 Paquets d'erreurs:

Si les perturbations ont une durée plus longue que la durée d'un bit, les erreurs apparaîtrons groupées.
On dit alors qu'il se présente "des paquets d'erreurs ".

On définit un paquet d'erreurs de la façon suivante:

-Deux séquences en erreurs correspondant à des paquets d'erreurs séparés s'il existe au moins entre ces deux paquets 10 bits binaires consécutifs non erronés.

-Pour une séquence d'erreurs correspondant à un seul paquet, la longueur du paquet l, est le nombre binaire qui constitue cette séquence.

-A l'intérieur d'un paquet d'erreurs, il peut exister des bits non erronés.

ANNEXE IV

Nous établirons dans cette annexe l'inégalité

$$\frac{|\log (OS)|}{n} \leq E(x,P) \quad .$$

Il faut partir de la première condition:

$$\sum_{i=d}^{i=n} C_n^i P^i (1-P)^{n-i} \leq (OS) \qquad (A1)$$

Dont le premier membre peut s'écrire:

$$C_n^d P^d (1-P)^{n-d} \left[1+\frac{C_n^{d+1}}{C_n^d}\left(\frac{P}{1-P}\right)+\ldots \frac{C_n^{d+j}}{C_n^d}\left(\frac{P}{1-P}\right)^j+\ldots\frac{C_n^n}{C_n^d}\left(\frac{P}{1-P}\right)^{n-d} \right] \qquad (A2)$$

Nous allons remplacer le terme général de cette série par un majorant.

En effet :

$$\frac{C_n^{d+j}}{C_n^d} = \frac{n!}{(d+j)![n-(d+j)]!}\frac{d!(n-d)!}{n!}$$

$$\frac{C_n^{d+j}}{C_n^d} = \frac{n-(d+j)+1}{d+1} \cdot \frac{n-(d+j)+2}{d+2} \cdots\cdot \frac{n-(d+j)+j}{d+j}$$

$$\frac{c_n^{d+j}}{c_n^d} = \prod_{i=1}^{i=j} \frac{n-(d+j)+i}{d+i} \qquad (A3)$$

Nous pouvons d'une manière évidente écrire:

$$\frac{(n-d)-(j-i)}{d+i} \leq \frac{n-d}{d} \qquad (A4)$$

En effet, **i** étant un entier positif inférieur ou égal à **j**, nous avons:

$$d+i>d \qquad \text{et} \qquad (n-d)-(j-i)<n-d$$

D'ou l'inégalité **(A4).**

Considérons alors la série géométrique **s**:

$$s=1+x+\ldots x^{j}+\ldots x^{(n-d)}=\frac{1-x^{(n-d)+1}}{1-x} \qquad (A5)$$

avec:

$$x=\frac{n-d}{d}\cdot\frac{p}{1-p} \qquad (A6)$$

En fonction de l'inégalité **(A.4)** nous pouvons écrire:

$$C_n^d P^d (1-P)^{n-d}.s > \sum_{i=d}^{i=n} C_n^i P^i (1-P)^{n-i} \qquad (A7)$$

Donc, si nous nous imposons la condition:

$$C_n^d P^d (1-P)^{n-d}.s \leq (os) \qquad (A8)$$

La condition **(A1)** sera à fortiori satisfaite.

Dans l'expression **(A6)** de **x** nous avons:

n>d et **P<<1** (probabilité d'erreur par bits) et par suite **x<<1** et **s≅1** comme le montre l'expression **(A5)**

Nous pouvons donc remplacer **(A8)** par la condition:

$$C_n^d \, P^d(1-P)^{n-d} \leq (OS) \qquad (A9)$$

Nous avons :

$$C_n^d = \frac{n!}{d!(n-d)!} \qquad (A10)$$

Remplaçons dans cette formule les factorielles par leurs approximations données par la formule de Stirling:

$$n! = n^n e^{-n} \sqrt{2\pi n} \qquad (A11)$$

Ceci nous donne

$$C_n^d = \frac{\left[n^n e^{-n} \sqrt{2\pi n} \right]}{\left[d^d e^{-d} \sqrt{2\pi n} \,.(n-d)^{(n-d)} e^{-(n-d)} \sqrt{2\pi(n-d)} \right]}$$

Les exponentielles s'éliminent et il reste

$$C_n^d = \frac{1}{\sqrt{2\pi \frac{d}{n}(n-d)}} \frac{n^n}{\sqrt{d^d (n-d)^{(n-d)}}} \qquad (A12)$$

Puisque nous avons n> d nous avons aussi:

$$\frac{1}{\sqrt{2\Pi \frac{d}{n}(n-d)}} < 1 \qquad (A13)$$

Et par suite:

$$n^n d^{-d} (n-d)^{-(n-d)} > C_n^d \qquad (A14)$$

On voit facilement que l'on a:

$$n^n d^{-d} (n-d)^{-(n-d)} = (d/n)^{-d} (1-d/n)^{-(n-d)} \qquad (A15)$$

Nous pouvons donc remplacer la condition **(A9)** par la condition plus restrictive:

$$(d/n)^{-d} (1-d/n)^{-(n-d)} P^d (1-P)^{(n-d)} \leq OS \qquad (A16)$$

Calculons les logarithmes de base 10 des deux membres de cette inégalité. Il vient:

$$-d \log (d/n) - (n-d) \log(1-d/n) + d \log(P) + (n-d)\log(1-P) \leq \log(OS)$$

En divisant les deux membres de cette inégalité par n nous obtenons:

$$-(d/n) \log(d/n) - (1-d/n) \log(1-d/n) + (d/n) \log P + (1-d/n)\log(1-P) \leq \frac{\log(OS)}{n}$$

Si nous posons: **d/n=2x,** nous reconnaissons dans les deux premiers termes, la fonction "entropie" **H(2x)**.

Les deux termes restants peuvent alors s'écrire:

$$(d/n - P + P) \log P + (1 - P + P - d/n) \log(1-P)$$

Ou encore:

$$(d/n - P) [\log P - \log(1-P)] - H(P) = -(d/n - P) H'(P) - H(P)$$

En définitive nous pouvons écrire la condition **(A16)** sous la forme

$$-\mathbf{H(P)} + \mathbf{H(2x)} - (2\mathbf{x}-\mathbf{P})\ \mathbf{H'(P)} \leq \frac{\mathbf{\log(OS)}}{\mathbf{n}} \qquad (\mathbf{A}\,17)$$

Mais **(OS)** est plus petit que **1** et son logarithme est négatif. Pour éviter cette difficulté on peut multiplier par **(-1)** les deux membres de l'inégalité **(A17)** mais cette opération implique d'inverser son sens.

Nous obtenons ainsi la condition **(A18)**

$$\frac{|\mathbf{\log_{10}(OS)}|}{\mathbf{n}} \leq \mathbf{E(x,P)} \qquad \mathbf{(A18)}$$

Avec:

$$\mathbf{E(x,P)} = \mathbf{H(P)} - \mathbf{H(2x)} + (2\mathbf{X},\mathbf{P})\ \mathbf{H'(P)}$$

ANNEXE V

Etablissement de l'inégalité:

$$(1-\frac{\mathbf{m}}{\mathbf{n}}) \geq \mathbf{H}_2(\mathbf{x})$$

La condition **(A18)** impose une condition entre le rapport **d/n** et les quantités **P** et **(OS)** qui constituent les données du problème à résoudre.

Ce n'est donc pas suffisant pour déterminer séparément **d** et **n** de plus la donnée **m** (nombre de bits d'information utile) n'a pas encore été prise en compte. Il nous faut donc trouver une autre condition reliant **m** à **n** et **d**.

Cette condition nous est fournie par la "marge inférieure de Hamming".

En effet on démontre (voir référence 5) que pour pouvoir trouver un code capable de corriger c_0 erreurs il est nécessaire que le nombre de mots faux que l'on peut fabriquer en plaçant de toutes les manières possible sur les n bits composant les mots du code,1, puis 2, ..., puis C_0 erreurs, reste inférieur ou égal au nombre de combinaisons possibles des bits de redondance. Cette condition s'écrit :

$$\sum_{i=0}^{i=c_0} \mathbf{C}_\mathbf{n}^\mathbf{i} \leq 2^{(\mathbf{n}-\mathbf{m})} \qquad (\mathbf{A}19)$$

En fait nous nous intéressons pas aux codes correcteurs d'erreurs, mais puisque d'après la relation (14) du § 2.8, nous avons:

$$c_0 = (d/2) - 1$$

La condition **(A19)** va nous donner la relation que nous cherchons entre **n**, **m** et **d**.

Du fait de la symétrie des coefficients de la série du binôme de Newton qui s'écrit:

$$C_n^0 = C_n^n \qquad C_n^1 = C_n^{n-1} \ldots\ldots \quad C_n^i = C_n^{n-i}$$

Nous avons en posant **j= n-i**

$$\sum_{i=0}^{i=c_0} C_n^i = \sum_{j=n}^{j=n-c_0} C_n^j = \sum_{j=n-c_0}^{j=n} C_n^j$$

Or nous avons

$$\sum_{j=n-(c_0+1)}^{j=n} C_n^j = c_n^{n-(c_0+1)} + \sum_{j=n-c_0}^{j=n} C_n^j > \sum_{j=n-c_0}^{j=n} C_n^j$$

Donc si nous imposons la condition:

$$\sum_{j=n-(c_0+1)}^{j=n} C_n^j \leq 2^{(n-m)}$$

La condition **(A19)** sera à fortiori satisfaite.

Or d'après **(A20)**: $c_0+1= d/2.$

Nous sommes donc amenés à écrire la relation:

$$\sum_{j=n-d/2}^{j=n} C_n^j \leq 2^{(n-m)} \qquad (A21)$$

Si dans le premier membre de la condition **(A1)** on fait **p=1/2** nous avons:

$$\sum_{i=d}^{i=n} C_n^i \; P^i(1-P)^{n-i} = 2^{-n} \sum_{i=d}^{i=n} C_n^i \qquad \text{avec } P=\frac{1}{2}$$

Nous sommes donc amené à effectuer les mêmes calculs qu'au §A1 en faisant dans les résultats **P=1/2** et en remplaçant **d** par **n- d/2.**

L'inégalité **(A7)** pourra alors s'écrire:

$$C_n^{n-d/2} \cdot 2^{-n} \cdot S \; \rangle \sum_{j=n-d/2}^{j=n} C_n^j \cdot 2^{-n} \qquad (A22)$$

avec:

$$x = \frac{d/2}{n-d/2} \qquad \text{et} \qquad S = \frac{1-x^{(d/2+1)}}{1-x}$$

Si **n>d** on a **x<1** et **S** est un nombre positif plus grand que **1**. Nous admettrons que S est encore assez voisin de **1** pour que si l'on pose la condition

$$C_n^{n-d/2} \leq 2^{(n-n)} \qquad (A23)$$

La condition **(A21)** soit encore satisfaite.

Il suffit ensuite de remplacer **d** par **(n-d/2)** dans l'inégalité **(A14)** pour obtenir en tenant compte de **(A15)**:

$$(1-\mathbf{d/2n})^{-(n-d/2)} \ (\mathbf{d/2n})^{-d/2} \ \rangle \ C_n^{n-d/2}$$

Nous pouvons donc remplacer la condition **(A23)** par la condition

$$(1-\mathbf{d/2n})^{-(n-d/2)} \ (\mathbf{d/2n})^{-d/2} \ \leq \ 2^{(n-m)} \qquad (A24)$$

En calculant le logarithme de base **2** des deux membres de **(A24)** nous obtenons

$$\frac{-(n-d/2)\ln(1-d/2n) - (d/2)\ln(d/2n)}{\ln(2)} \leq \frac{\ln(2^{(n-m)})}{\ln(2)}$$

ou encore en divisant les deux membres de cette inégalité par **n** et en posant: $\mathbf{x=d/2n}$

$$\frac{-(1-x)\ln(1-x) - x\ln(x)}{\ln(2)} \leq (1-\frac{m}{n})$$

On reconnaît dans le premier membre la fonction "entropie" $\mathbf{H_2(x)}$ et on démontre ainsi l'inégalité de la deusième condition.

$$(1-\frac{m}{n}) \ \geq \ H_2(x) \qquad (A25)$$

BIBLIOGRAPHIE

1-Professeur R. GABILLARD
"Bruits et transmission de l'information"
Journée d'études SEE du jeudi 13 novembre 1986

2- Abdel Hadi Ouadghiri
"Sur les conditions d'utilisation des codes détecteurs d'erreurs dans les transmissions numériques nécessitant une sécurité quasi absolue"
Thèse présentée à l'université de LILLE I en 1986

3- Abdel Hadi Ouadghiri , J.F.Dhalluin
"Sécurité des transmissions d'information"
Rapport GRRT - 1984

4- G.Cullman
"Codes détecteurs et correcteurs d'erreurs"
Dunot 1967

5-A. Spataru
"Théorie de la transmission de l'information".
Masson et Cie 1973.

6- CH. Semet, M. El Koursi
"Etude des conditions de réalisation d'un accouplement sans contact matériel entre véhicules dans les transports urbains"
Convention CRESTA / LRPE - Janvier 1987

7- Leon J S, Pless V, Sloane NJA
"Codes ternaires"
1981 / 03

8-Robinson J P , Cohn M
"Counting sequences"
1981 / 01

9- Ludman J E
"Génération de codes de Gray"
1981 / 10

10- Marc Heddebaut
"Transmissions numériques en tunnel"
Note GRRT-IRT-CRESTA, Septembre 1985.

11- M. El Koursi, Ch. Magnier, J. F. Dhaluin
"Commande des portes VAL. Rapport d'avancement au 1er Mars 1984".
Note GRRT-LRPE.

12- Abdel Hadi Ouadghiri
"Application de la théorie des codages au traitement d'informations de sécurité analysées par microprocesseur".
D.E.A Electronique 1983, U.S.T.L .

13- Ch. Magnier
"Commande-contrôle de processus par traitement hierarchisé. Etude des aspects sécurité-disponibilité".
Thèse de 3ème cycle USTL Lille Décembre 1985.

14- J.F . Dhalluin
"Commande de contrôle de processus en sécurité.
Application à la commande d'un ensemble de portes véhicule VAL".
Thèse de Docteur Ingénieur. Décembre 1983.

15- E. KOURSI
"Une méthode sûre de développement des logiciels destinés à la commande de processus en sécurité".
Thése - U.S.T.L - Mai 1987

16- P. Huckett, G. Thow
"Analyse des performances en fonction des erreurs des systèmes (transmission numérique"
Electronique, Technique et industries (Paris).
1984 n° 5, pp 35-43

17- Jack. Wolf , Fellow
"On the probability of undected error for linear block codes"

Revue IEEE transactions on communication, Vol COM 30" n°2, Février 1982.

18- Tadao Kasami
"linear block codes for error detection".
Revue IEEE transactions on information theory,
Vol IT 29 n° 1, Janvier 1983.

19- Robero Padovani, Jack Keilwolf
"Poor error correction codes are poor error detection codes"
Revue IEEE transaction on information theory,
Vol IT 30 n° 1, Janvier 1984.

20- Mamadou Mbath
"Contribution à l'étude théorique et expérimentale de la propagat d'onde haute fréquence en tunnel"
Thèse présentée à l'université de Lille I en 1985.

21- Jacques Hervé
"Electronique appliquée à l'information"
-Tome 1 et 2. Masson 1981.

22- P . Fondanèdre, P. Gilbertas
"Filtres numériques. Principes et réalisation".
Masson 1981.

23- Cyril Leung
"Evaluation of the inducted error probability of single parity- check product codes".
Revue IEEE Transactions on communications,
Vol COM 31 n2, Fevrier 1983.

24- Marc Heddebaut
"Transports automatiques et transmissions électromagnétiques en tunnel"
Revue "Recherche, Transports, Sécurité" n° 4.

25- Abdellah Mahdi
"Codes auto-correcteurs d'erreurs adaptés à la protection d'informations numériques sur vidéodisque analogique"
Thèse présentée à l'université de Rennes I en 1982.

26- Peterson Wesly
"Error correcting code"
M.I.T Press 1961

Méthode d'obtention et d'optimisation des principaux paramètres garantissant l sécurité d'une transmission numérique en se fixant l'objectif de sécurité.

Résumé

Certaines erreurs de transmission numérique peuvent être à l'origine de catastrophes importantes. Les canaux de transmission électriques ou électromagnétiques sont de plus en plus affectés par des parasites et perturbations de toutes natures susceptibles de modifier la nature exacte de l'information véhiculée.

Pour augmenter l'efficacité de la transmission, une méthode classique consiste à coder les messages. Nous proposons une méthode permettant d'obtenir la distance minimale (**d**) entre les mots codes et le nombre de bit de contrôle (**k**) qu'il faut associer afin que le codage obtenu vérifie l'objectif de sécurité (**OS**). Cet objectif de sécurité (**OS**) est le point de départ de la méthode, paramètre que se fixe l'exploitant ainsi que le nombre de messages sécuritaires à transmettre.

Bien que cette méthode ne fasse pas intervenir un code particulier, nous la vérifions par un exemple dans le cas des codes générés par un polynôme. Nous proposons ensuite le polynôme générateur optimum qui donnera la marge maximale.

L'originalité de ce travail consiste à calculer la probabilité de non détection d'erreurs réelle du code optimum choisi ainsi que sa capacité de détection des paquets d'erreurs.

Nous avons utilisé ces codes de détection d'erreurs pour la sécurité des transmissions numériques dans le domaine des transports.

Enfin, nous présentons une méthode d'évaluation de la qualité d'une transmission numérique à haut débit.

Mots clés: Objectif de sécurité, Distance minimale, Pôlynome générateur, Probabilité de non détection d'erreurs, Erreurs indépendantes, Paquets d'erreurs.