

**Un logiciel de simulation
de transmission de données bruitées :
LOUSTICC**

A software program for the simulation of data transmission with noises: LOUSTICC



M. C. GENNERO

Équipe AAEC, Laboratoire LSI, Université Paul-Sabatier, 118, route de Narbonne, 31062 TOULOUSE CEDEX

Titulaire d'un diplôme d'ingénieur CNAM en informatique (1984). Occupe un poste d'ingénieur en informatique au CNES (Centre National d'Études Spatiales) depuis 1982.

M. C. Gennero prépare un doctorat d'ingénieur (Informatique), dans l'équipe AAEC dirigée par le Professeur A. Poli, dans le domaine des applications des codes correcteurs : transmission d'images, de sons, etc.

RÉSUMÉ

Nous présentons une version améliorée de notre logiciel de simulation de transmission de données, en présence de bruits : LOUSTICC. Dans cet article, après une brève description du logiciel, nous donnons une évaluation de sa complexité. Nous donnons également un exemple de simulation.

MOTS CLÉS

Codage, brouillage, décodage, codes tronqués, trame d'information, taux d'erreur, taux résiduel d'erreur.

SUMMARY

We present an improved version of our software program LOUSTICC. It simulates data transmissions through a noisy channel. We give an evaluation of its complexity and we also give an example of simulation.

KEY WORDS

Coding, addition of transmission error bits, decoding, shortened codes, binary information frame, binary error rate, binary residual error rate.

TABLE DES MATIÈRES

Introduction

1. Rappel des fonctions du logiciel

- 1.1. Codage et décodage
- 1.2. Brouillage de la trame
- 1.3. Techniques de correction d'erreurs détectées
- 1.4. Résultats fournis par le logiciel

2. Évaluation du temps d'exécution du logiciel

- 2.1. En fonction de la taille des trames
- 2.2. En fonction de la taille du fichier

3. Exemple de simulation

Conclusion

Bibliographie

Introduction

LOUSTICC est un logiciel de simulation de transmission de données à travers un canal bruité.

Nous présentons ici les améliorations que nous avons apportées au logiciel : possibilité de traiter des trames binaires de grande longueur. Nous donnerons une évaluation du temps de calcul nécessaire. Nous présentons un exemple de simulation pour une image.

D'abord nous rappelons les principales fonctions du logiciel. Pour alléger la présentation, nous ne donnerons pas de détails sur les méthodes employées. Celles-ci sont décrites dans le mémoire en référence [2].

1. Rappel des fonctions du logiciel

Dans LOUSTICC, nous simulons la transmission de trames binaires. Celles-ci sont obtenues à partir d'un fichier de données numérisées à transmettre.

La première fonction du logiciel est donc la constitution d'un fichier contenant les trames binaires. Ces dernières sont numérotées de manière absolue. Nous utilisons la numérotation des trames lors d'une éventuelle retransmission de celles-ci (cf. décodage).

Les liaisons simulées peuvent être de type simplex ou duplex intégral. Dans ce dernier cas, seule la voie principale (émetteur → récepteur) sera brouillée, et protégée par un code correcteur.

La technique de correction d'une erreur détectée (non corrigible par le code) sera différente selon le type de liaison simulée.

Le traitement des données binaires est réalisé trame par trame.

Pour chacune d'elles ont aura les fonctions suivantes :

1. codage bloc par bloc de toute la trame;
2. ajout d'erreurs sur la trame codée (brouillage);
3. répercussion des erreurs sur la trame non codée;
4. décodage bloc par bloc de la trame codée et brouillée.

1.1. CODAGE ET DÉCODAGE

Les codes correcteurs utilisés dans LOUSTICC, pour la protection de l'information, sont des codes en bloc. Le codage est systématique, sans entrelacement, ni concaténation de codes.

Actuellement quatre types de codes correcteurs sont utilisés dans le logiciel :

1. Des codes de Fire tronqués ou non. Ils permettent la correction et la détection d'erreurs de coupure (bursts). Ces codes sont largement employés pour la protection de l'information sur disques et mémoires magnétiques, par exemple.

Pour le décodage de ces codes, nous avons employé la technique de piégeage d'erreurs de Meggitt [10].

2. Des codes de Reed-Solomon sur des corps F_p (p premier). Dans ce cas, les modules de codage et de décodage assurent en entrée et en sortie, une conversion de l'information binaire en éléments du corps F_p , et réciproquement.

L'algorithme de décodage utilisé est celui proposé par Reed-Miller-Truong utilisant les fractions continues, et la transformée de Fourier inverse [11]. Ces codes de Reed-Solomon permettent également la correction d'erreurs de coupure.

3. Un code autodual (64, 32, 12). L'algorithme de décodage utilisé (13) pour ce code permet la correction d'au plus deux symboles binaires. Il est très puissant en détection. L'algorithme que nous avons implanté est une variante du décodage à seuil proposé par G. L. Feng [13].

4. Le code de Golay binaire étendu (24, 12, 8). Il permet la correction d'au plus trois symboles binaires. L'algorithme de décodage utilise l'ensemble de permutation du code.

1.2. BROUILLAGE DE LA TRAME

Par brouillage nous appelons l'ajout d'erreurs binaires sur une trame de symboles binaires. Ces erreurs peuvent être des erreurs de coupure ou bien des erreurs isolées. Dans ce qui suit nous considérons les erreurs isolées comme des erreurs de coupure de longueur 1.

Dans LOUSTICC les erreurs sont engendrées par le contenu d'un tableau simulant le fonctionnement d'un registre à décalage, dont les connexions de "feed

back" sont les coefficients d'un polynome primitif dans $\mathbb{F}_2[X]$.

Le brouillage, c'est-à-dire l'écart entre deux débuts d'erreur, est aléatoire. Plusieurs modèles sont simulés : une loi géométrique, normale, ou uniforme. Le taux binaire d'erreurs (nombre moyen de symboles binaires érronés) est alors donné par :

$$\tau_t = \frac{l/2}{(1 + aE(X^l) + t)}$$

où X est la v. a. suivant les lois précédentes (t est la valeur d'une v. a. certaine, et les paramètres a et l sont des entiers permettant d'ajuster ce taux à un taux d'erreur désiré pour la simulation).

Le modèle d'erreur est choisi une fois pour toute, pour la totalité du fichier à transmettre. La longueur des trames (choisie par l'utilisateur du logiciel) devra être choisie en conséquence surtout dans le cas d'une simulation de type duplex intégral.

1. 3. TECHNIQUES DE CORRECTION D'ERREURS DÉTECTÉES

Dans certains cas il est possible qu'une erreur non corrigible par le code correcteur soit détectée. Nous avons alors deux possibilités, selon que la liaison simulée est de type simplex ou duplex intégral.

(a) Cas d'une liaison de type simplex

Dans le cas où le fichier transmis constitue une image, nous utilisons la redondance interne de l'information transmise. Nous faisons donc un masquage du bloc contenant l'erreur détectée, par le bloc de même rang de la ligne précédente de l'image.

Dans le cas de données d'un autre type (textes, paroles, ...) il n'y a pas d'action possible.

(b) Cas d'une liaison de type duplex intégral

Les trames contenant une erreur (non corrigible) seront retransmises. La méthode de reprise utilisée dans LOUSTICC est la reprise « en fin », proposée dans le rapport du projet NADIR de l'INRIA [6].

A chaque trame décodée, il y a émission d'un accusé de réception positif ou négatif. L'émetteur gère les accusés de réception reçus et mémorise le numéro des trames qu'il devra retransmettre. Celles-ci seront ré-émises en fin de traitement.

1. 4. RÉSULTATS FOURNIS PAR LE LOGICIEL

En fin de transmission le logiciel fournit des résultats numériques. D'autre part nous avons les deux fichiers contenant les trames codées-brouillées-décodées et les trames non codées-brouillées. La visualisation de ces deux fichiers montre l'efficacité du codage, elle montre la qualité de la transmission.

Les résultats numériques sont :

– le rendement de la transmission; qui dans le cas d'une transmission de type simplex est exactement le taux de redondance du code (k/n);

– le taux binaire expérimental d'erreur, le taux résiduel binaire d'erreurs;

– des statistiques sur la répartition des erreurs (coupures ou isolées) :

- moyenne de la loi des écarts entre deux débuts d'erreur,
- variance et écart-type,
- nombre d'erreurs (coupures ou erreurs isolées),
- histogramme de l'écart entre deux débuts d'erreurs.

2. Évaluation du temps d'exécution du logiciel

Comme nous l'avons dit dans l'introduction, LOUSTICC a été amélioré de façon à accroître la longueur des trames traitées, et de façon à diminuer le temps de calcul nécessaire. L'amélioration est essentiellement informatique. En effet, nous utilisons la capacité maximale des mots de l'ordinateur pour coder les symboles binaires à transmettre (ex : 60 bits sur CDC, 37 sur Burroughs, 34 sur DPS).

De ce fait nous avons utilisé les fonctions logiques du FORTRAN, et gagné du temps à l'exécution.

Actuellement le logiciel permet de traiter des trames d'au plus 2,2 Mbits.

Nous donnons une évaluation du temps de calcul, en fonction de la taille des trames et de la taille du fichier à transmettre.

Les temps que nous donnons ont été déterminés par des relevés sur un calculateur DPS 8/70 du Centre Interuniversitaire de Calcul de Toulouse.

2. 1. EN FONCTION DE LA TAILLE DES TRAMES

Pour le premier module de la chaîne (constitution des trames binaires) :

$$t = 16,25 x^{-0,032}$$

Pour le module de simulation de la transmission :

$$t = 268,44 x^{-0,012}$$

Dans les deux modules nous voyons que nous avons intérêt à utiliser des trames les plus longues possible.

Les temps sont obtenus en secondes, x (la longueur des trames) est exprimé en nombre de symboles binaires.

2. 2. EN FONCTION DE LA TAILLE DU FICHIER

Nous donnons l'évaluation du temps de calcul nécessaire du module de transmission (codage + brouillage des deux trames + décodage), dans le cas d'une transmission simplex :

$$t = 2,57 x^{1,046}$$

(t en secondes, x en kilobits).

Remarque : Sur le calculateur utilisé (DPS 8/70 CII-HB) nous avons été handicapés par le fait que les opérations logiques (ET, OU, OUX, ...) sont simulées et non microprogrammées.

3. Exemple de simulation

L'exemple que nous donnons est celui qui figure dans la brochure d'utilisation du logiciel sur le Centre de Calcul de Toulouse (CICT).

Le fichier transmis est constitué d'une image à 5 niveaux de gris. Il contient 265 trames de 132 bits.

Le code correcteur utilisé est un code de Fire tronqué corrigeant des erreurs de coupure de longueur 4.

Le taux d'erreur désiré pour la simulation est de $1,18 \cdot 10^{-3}$, avec un espacement entre erreurs suivant le modèle d'une loi normale. La longueur maximale des erreurs injectées est de 5.

Cf. exemples.

Conclusion

Comme nous l'avons montré, LOUSTICC permet la simulation de transmission de données à travers un canal bruité.

C'est un outil qui permet l'étude de performance des codes correcteurs. Nous pouvons tester leur capacité de correction (et détection) dans un grand nombre de cas différents de répartition d'erreurs.

Tous les paramètres du logiciel sont laissés au choix de l'utilisateur (taux de brouillage, code utilisé, longueur des trames, etc.), ce qui lui confère une grande souplesse d'utilisation.

Tous les outils nécessaires sont en place pour l'utilisation de codes concaténés, et pour l'entrelacement de codes.

De plus les temps d'exécution ont été réduits et LOUSTICC est utilisable avec des temps relativement courts. Le temps de calcul pourrait être réduit encore par l'utilisation de certains sous-programmes en langage machine.

Exemple de simulation

MODULE 1

LOUSTICC

Logiciel de Simulation de Transmission
d'Information avec Codes Correcteurs

réalisé par l'équipe A.A.E.C.C.

NOMBRE DE PIXELS (CARACTERES) PAR LIGNE

?
60

NBRE DE BITS POUR LE CODAGE DES PIXELS (CARACTERES)

?
8

NBRE DE NIVEAUX DE GRIS DIFFERENTS (TEXTE : 0)

?
2

FORMAT DE LECTURE DU FICHIER INITIAL

?
(6011)

LONGUEUR D UN BLOC D INFORMATION

?
33

APPLICATIONS

NOMBRE DE BLOC D INFORMATION (PAR TRAME)

?
4

NOMBRE DE BITS POUR LE CODAGE DU NUMERO DE TRAME

?
12

ENTRER LE CODE DESIRE

- 1 : GOLAY BINAIRE (24,12,8)
- 2 : FIRE
- 3 : REED SOLOMON
- 4 : CODE AUTODUAL (64,32) 2-CORRECTEUR

?
2

ENTREE DU POLYNOME GENERATEUR :
ENTREE DU DEGRE DE g(X) :

?
13

ENTRER LES COEFFICIENTS DU POLYNOME GENERATEUR
(par degres croissants sans coef directeur) :

?
1001001100100 (1+x³+x⁶+x⁷+x¹⁰+x¹³)

ENTRER LE POLYNOME PREMULIPLICATEUR (MEGGITT) utilisé lors du décodage de Meggitt

?
1100110111011 (x^{n-n'} x^{n-k} modulo g(x), n la long du code, n' la long du code tronqué)

ENTRER LA LONGUEUR MAXIMALE DES ERREURS CORRIGIBLES

?
4

ENTREE DES DONNEES POUR LE BROUILLAGE

LOI DESIREE :

- 1 : GEOMETRIQUE
- 2 : NORMALE
- 3 : UNIFORME
- 4 : ORAGE

?
5

*** ERREUR ***

Valeur non conforme aux bonnes demandes : 5

2

VALEUR DE A (A et I permettent d'ajuster le taux d'erreur expérimental au taux désiré)

?
5

VALEUR DE I

?
2

VALEUR DE T

?
67

ENTRER LA LONGUEUR MAXIMALE DES COUPURES INJECTEES (erreurs isolées taper 0)

?
5

ENTRER LE POLYNOME GENERATEUR DE L ERREUR

?
11000 (coefficients du polynome de degré 5 utilisé pour le registre à décalage
permettant de générer les erreurs : 1+x+x⁵)

CHOIX DU TYPE DE LIAISON SIMULEE :

- 0 : SIMPLEX
- 1 : DUPLEX

?
0

UN LOGICIEL DE SIMULATION DE TRANSMISSION DE DONNÉES BRUITÉES : LOUSTICC

RESULTATS DE LA SIMULATION

LONGUEUR DU BLOC D INFORMATION : 33
NOMBRE DE BLOCS PAR TRAME : 4
NOMBRE DE TRAMES : 265
nombre de bits pour le codage du No de trame : 12
NOMBRE DE BITS POUR LE CODAGE DE CHAQUE PIXEL : 8

STOP (fin module 1)

MODULE 2

LONGUEUR DES BLOCS D INFORMATION : 33
LONGUEUR DES BLOCS CODES : 46
NOMBRE DE BLOCS PAR TRAMES : 4
NOMBRE DE BITS POUR CODAGE DU NO DE TRAME : 12
TYPE DE TRANSMISSION SIMULEE (0 : SIMPLEX, 1 : DUPLEX) : 0
NOMBRE DE TRAMES DU FICHIER INITIAL : 265
NOMBRE DE TRAMES RETRANSMISES : 0
RENDEMENT DE LA TRANSMISSION : 71.74
TAUX D'ERREURS EXPERIMENTAL : 0.127153E-02
(le taux d'erreurs théorique est : 0.118036E-02)
MOYENNE : 17.93
VARIANCE : 14.20
ECART TYPE : 3.7686
NRE DE TIRAGES : 29

0 I
1 I
2 I
3 I
4 I
5 I*
6 I
7 I
8 I
9 I
10 I
11 I
12 I
13 I**
14 I*****
15 I*****
16 I*****
17 I*****
18 I*****
19 I*****
20 I*****
21 I*****
22 I*****
23 I****
24 I**
25 I
26 I
27 I
28 I
29 I
30 I
31 I
32 I
33 I
34 I
35 I
36 I
37 I
38 I
39 I
40 I

FIN DU MODULE 2

BIBLIOGRAPHIE

- [1] M. C. GENNERO, A. POLI et J. A. THIONG LY, Codes correcteurs en transmission très bruitée : résultats de simulations, 9^e Colloque International GRETSI, mai 1983, Nice.
- [2] M. C. GENNERO, Un logiciel de simulation de transmission d'information, *Mémoire CNAM en Informatique*, Octobre 1983, Toulouse.
- [3] N. GLOVER, Accuracy issues for disk controller developers, *Computer design* (special report), octobre 1982.
- [4] D. L. GORDON, Minimal permutations sets for decoding the binary Golay code, *IEEE Transactions on Inf. theory*, IT 28, mai 1982, p. 541-543.
- [5] N. C. GORE, *Transmitting binary symbols with Reed-Solomon codes*, J. Hopkins University, EE report, n° 73-5, avril 1973.
- [6] J. L. GRANGE, C. HUITEMA et H. ZIMMERMANN, Utilisation informatique des satellites de télécommunication, identification des problèmes posés et éléments de solution, *Projet Pilote NADIR*, INRIA, réf. Gen 3.500, décembre 1980.
- [7] J. P. GRAY, Line control procedures, *IEEE*, 60, n° 11, novembre 1972, p. 1301-1312.
- [8] C. MACCHI et J. F. GUILBERT, *Téléinformatique, transport et traitement de l'information dans les réseaux et systèmes téléinformatiques*, Dunod/Informatique, 2^e éd., avril 1983.
- [9] F. J. MACWILLAMS et N. J. A. SLOANE, *The theory of error correcting codes*, North Holland Publishing Company, 1977.
- [10] W. W. PETERSON, *Error Correcting codes*, MIT Press, 1961.
- [11] I. S. REED, T. K. TRUONG et R. L. MILLER, The fast decoding of Reed-Solomon codes using Fermat theoretic transform and continued fractions, *IEEE Trans. on Inf. theory*, 24, n° 1, 1978, p. 100-106.
- [12] J. WOLFMANN, A permutation decoding of the (24, 12, 8) Golay code, *IEEE Trans. on Inf. theory*, les Arcs, 1983.
- [13] G. L. FENG, Generalized Threshold decoding of cyclic codes, *Colloque International : Algèbre et codes correcteurs: théorie et applications*, juin 1983 (à paraître).