

ECOLE NATIONALE POLYTECHNIQUE

DEPARTEMENT : ELECTRONIQUE

المكتبة - المكتبة  
BIBLIOTHEQUE - المكتبة  
Ecole Nationale Polytechnique

PROJET DE FIN D'ETUDES

SUJET

SIMULATION DU COMPORTEMENT  
D'UNE BOUCLE A VERROUILLAGE  
DE PHASE NUMERIQUE

Proposé par :  
J. GORALSKI

Etudié par :  
L. H. FEDAOUI

Dirigé par :  
J. GORALSKI

PROMOTION : JUIN 1990

SIMULATION DU COMPORTEMENT  
D'UNE BOUCLE A VERROUILLAGE DE PHASE  
NUMERIQUE

### REMERCIEMENTS

Ma reconnaissance va à Mr GORALSKI J., mon promoteur, qui m'a guidée dans mon travail avec beaucoup de disponibilité.

Mes remerciements s'adressent aussi à Mr HAMZA et à toutes les personnes qui m'ont aidée dans l'élaboration de ce logiciel.

CHAPITRE 1 INTRODUCTION

CHAPITRE 2 TRANSMISSION NUMERIQUE

2.1 Généralités

2.1.1 Introduction

2.1.2 Le signal numérique

2.2 Synoptique d'une transmission numérique

2.3 Types de transmission d'un signal numérique

2.4 Horloge Transmission synchrone

2.5 Synchronisation Critère de synchronisation

2.6 Les perturbations de critères

2.6.1 Sources d'imperfection

2.6.2 L'offset de fréquence

2.6.3 Influence de la structure des données

CHAPITRE 3 BOUCLE A VERROUILLAGE DE PHASE

3.1 Introduction

3.2 Généralités

3.3 Principe de fonctionnement Constitution générale

3.3.1 Schéma fonctionnel d'une PLL

3.3.2 Les éléments constitutifs de la PLL

3.3.3 Fonctionnement -cas général-

3.3.4 PLL numérique

CHAPITRE 4 SIMULATION

4.1 Introduction

4.2 Description

4.2.1 Partie émission

4.2.2 Partie réception

4.2.3 Visualisation Représentation graphique adoptée

4.3 Organigrammes

CHAPITRE 5 UTILISATION DU PROGRAMME

5.1 Mode d'utilisation

5.2 Exemples

CONCLUSION

ANNEXE Listing du programme en Turbo Pascal v5.0

BIBLIOGRAPHIE





## CHAPITRE 1

### INTRODUCTION



Dans les télécommunications modernes, l'importance de la transmission numérique est de plus en plus grande. Dans la plupart des cas, cette transmission est réalisée en mode synchrone, ce qui nécessite une récupération de la base de temps côté réception, récupération assurée par un dispositif (boucle) de synchronisation.

L'analyse théorique d'une boucle de synchronisation est très complexe et difficile. L'analyse par simulation pratique (réalisation) n'est ni facile, ni souple et exige un matériel important. Nous avons choisi l'analyse par simulation programmable parce que nous pensons qu'elle est la solution qui convient le mieux.

Le but de ce travail est de concevoir un logiciel qui permet de simuler une boucle de synchronisation numérique. Le programme conçu doit permettre de faire varier facilement les caractéristiques de cette boucle et d'en observer les résultats en fonction des paramètres choisis.

Dans les développements qui suivent, après le chapitre 2 qui traite de généralités sur la transmission numérique, vient une présentation de la boucle à verrouillage de phase. Le chapitre 4 est consacré à l'explication détaillée de notre simulation. Enfin, le dernier chapitre est réservé au mode d'utilisation du logiciel.

Les simulations faites nous ont montré que ce logiciel constitue un outil qui présente beaucoup d'intérêt par sa fiabilité et sa souplesse.

CHAPITRE 2

TRANSMISSION NUMERIQUE

## 2.1 GENERALITES

### 2.1.1 Introduction

Les premiers systèmes de télécommunication (téléphone, télégraphe) sont nés au XIX<sup>e</sup> siècle; après un développement considérable, on en arrive aujourd'hui à l'utilisation d'énormes banques de données, à l'introduction des terminaux téléinformatiques, à la distribution de nombreux programmes de télévision par câble, aux réseaux publics pour données à l'échelle du monde.

Toutes ces applications, et bien d'autres encore, actuelles et à venir, utilisent un transport de l'information.

La nature de l'information à échanger étant le plus souvent numérique, les équipements utilisent le système binaire comme logique de base pour représenter l'information.



### 2.1.2 Le signal numérique

#### Définitions:

Un signal numérique, d'une manière générale, est constitué d'une succession de grandeurs ne pouvant prendre que des valeurs discrètes bien définies pendant des durées déterminées.

Les valeurs de ces grandeurs sont les états ou les moments significatifs. Et l'on appelle **transition** le passage d'un état significatif au suivant.

Si 0 et 1 représentent les états significatifs, le passage de l'état 0 à l'état 1 (ou l'inverse) est une transition. [1]

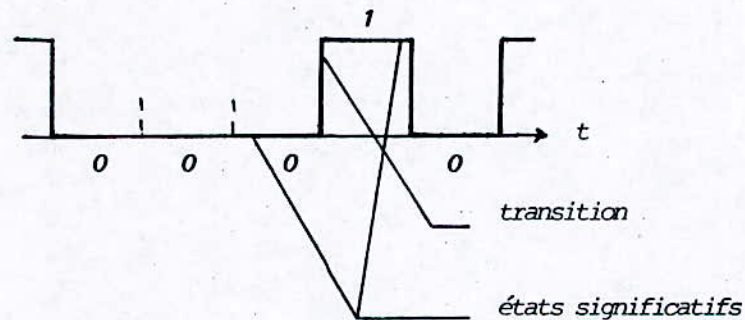


Fig. 1.



## 2.2 SYNOPTIQUE D'UNE TRANSMISSION NUMERIQUE

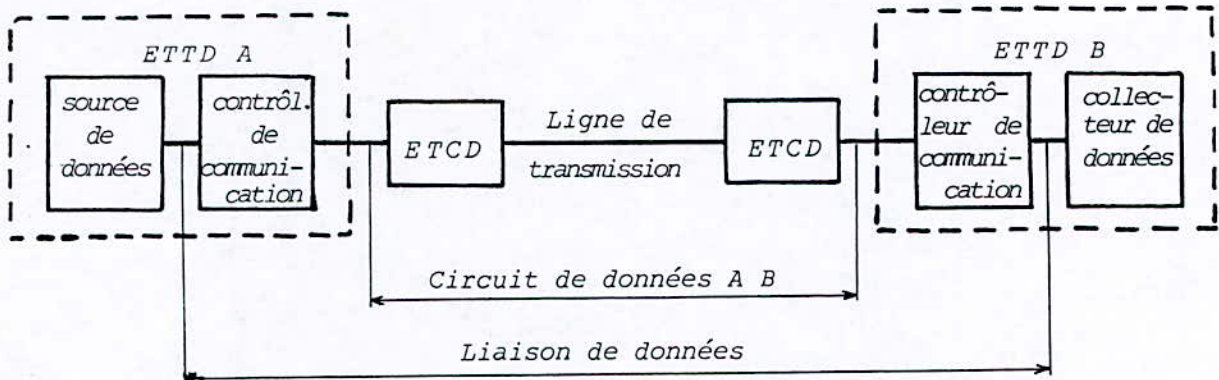


Fig.2 Transmission de données de A vers B

L'information est émise ou reçue par un équipement terminal de traitement de données ETTD.

Dans un ETTD, nous distinguons deux parties qui réalisent des fonctions différentes:

- la machine de traitement : qui peut être source ou collecteur de données;
- le contrôleur de communication : qui regroupe les organes chargés des fonctions de communication. ( il réalise en particulier la protection contre les erreurs ).

L'équipement de terminaison du circuit de données:ETCD est l'organe chargé d'adapter le signal électrique délivré par le terminal, au support de transmission. Cette fonction est réalisée le plus souvent par modulation-démodulation dans un équipement appelé MODEM. [2]

La ligne de transmission, support ou canal de transmission est utilisée pour transmettre le signal d'un émetteur au récepteur. [2]

### 2.3 TYPES DE TRANSMISSION DU SIGNAL NUMERIQUE

Il en existe 2 :

- en bande de base: qui consiste à transmettre directement sur le support les signaux numériques.

Le spectre des signaux numériques à valeurs 0 et 1 est illimité mais concentré au voisinage de la fréquence nulle. Tels quels, ces signaux ne sont pas adaptés à la transmission; mais de simples modifications, codage par exemple, permettent de modifier la forme du spectre, d'en supprimer la composante continue et de les adapter ainsi aux supports. [2](chap 3.2)

- en bande transposée: il s'agit d'une transmission par modulation d'une onde porteuse. Ce qui consiste à utiliser un signal de base  $y(t) = k \sin(2\pi f t + \varphi)$  dont les caractéristiques sont modifiées en fonction des signaux à transmettre. Le rôle de la porteuse est de transposer le signal à transmettre dans la bande passante du support utilisé.

**NB** Les opérations de modulation-démodulation sont réalisées par un MODEM. [2]

### 2.4 HORLOGE - TRANSMISSION SYNCHRONE

A l'émission, la transmission est synchrone d'un signal de fréquence issu d'une horloge très stable: c'est la base de temps de l'émission. Tous les états caractéristiques ont une durée  $T_b$  fixe et identique. La succession de ces moments est alors dictée par une horloge de fréquence  $F = 1/T_b$ . Toutes les transitions surviennent à des intervalles de temps  $T_m$  multiples entiers de l'unité de temps  $T_b$ . ( $T_m = n T_b$ )

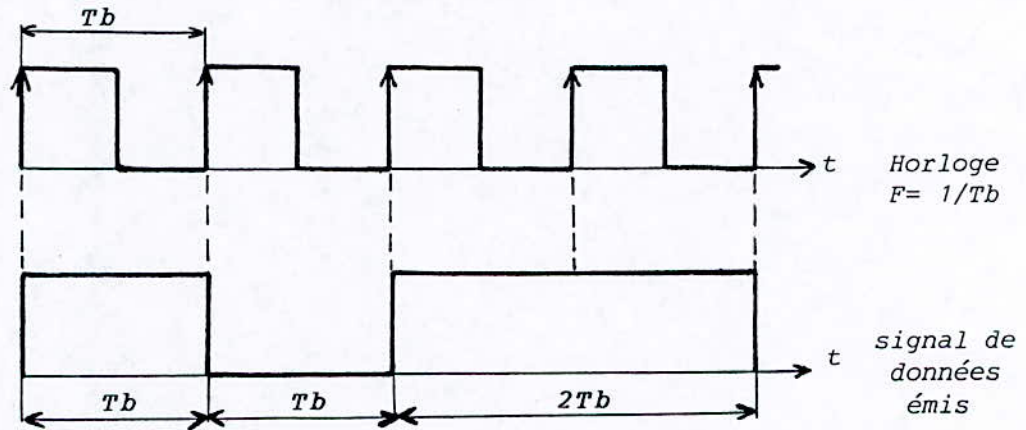


Fig. 3 Transmission synchrone

La fréquence de l'horloge à l'émission définit le débit binaire d'une voie en transmission synchrone, qui est le nombre maximal de symboles binaires transmis par seconde sur cette voie.



## 2.5 SYNCHRONISATION - CRITERE DE SYNCHRONISATION

La synchronisation est la récupération (la reconstitution locale) d'un signal de référence utilisé par le dispositif de réception.

Le signal de référence peut être, dans notre cas :

- la base de temps pour les éléments binaires;
- l'onde porteuse non modulée utilisée en démodulation synchrone (détection synchrone).

Il est nécessaire pour synchroniser de se baser sur ce qui est appelé : le **critère de synchronisation**. Il existe différentes manières de le choisir.

a) Lorsque le signal de référence est la base de temps, il existe 2 façons de choisir ce critère :

1°- Utiliser un signal pilote. Cette première méthode consiste en l'envoi d'un signal pilote : une sinusoïde ayant pour fréquence celle de la base de temps.

C'est la méthode la plus évidente mais, elle comporte certains désavantages; il faut, par exemple, prévoir un dispositif d'injection de ce signal pilote dans le signal utile et un autre dispositif d'extraction de ce même signal côté réception. Il faut protéger le signal de données du signal pilote et inversement.

Un autre désavantage réside dans le fait que, une partie de la puissance destinée au signal informatif étant utilisée par le signal pilote, il se produit une dégradation du rapport signal/bruit (S/N).

2°- Utiliser les moments caractéristiques du signal après la démodulation (s'il y a modulation), mais avant la régénération ( qui consiste à reconstituer la répartition dans le temps: à replacer aux bons endroits les moments caractéristiques. cf [12] ), ces moments caractéristiques sont pris comme étant les instants où se produisent les transitions.

Les critères de synchronisation sont donc les transitions.

b) Lorsque le problème de synchronisation se pose en détection synchrone pour la récupération de l'onde porteuse non modulée, 2 possibilités se présentent:

1°- Utiliser un signal pilote, de la même manière qu'en a)

2°- Utiliser les passages par 0 de l'onde porteuse modulée.



## 2.6 LES PERTURBATIONS DE CRITERES

2.6.1 Les supports de transmission n'étant pas parfaits, des perturbations s'ajoutent au signal utile. Ces perturbations se traduisent par des déphasages positifs ou négatifs qui faussent la position des critères de synchronisation.

Quelques sources d'imperfection de critère:

- le bruit thermique : dû à l'agitation thermique des électrons. C'est le bruit propre des composants.
- les perturbations atmosphériques;
- le bruit induit : à hautes fréquences, le courant circulant dans des dispositifs périphériques induit un courant par couplage capacitif;
- un mauvais choix de seuil de décision, ce qui conduit à des distorsions biaisées. cf fig.6.

### 2.6.2 L'offset de fréquence:

Il arrive que les fréquences des générateurs des signaux d'horloge de réception et d'émission ne soient pas exactement les mêmes. Il peut exister une différence de quelques hertz (en l'absence de critère de synchronisation). Cette différence est appelée: **OFFSET** de fréquence et peut être la cause d'erreurs. cf Fig.4.

On suppose toujours que c'est l'horloge d'émission qui impose son rythme, elle le fixe.

L'horloge d'émission peut, par exemple, imposer un débit de 2400 bit/s alors que pour la réception nous avons 2402 bit/s. Il y aura donc un offset de 2 Hz à la réception.

Le dispositif de synchronisation doit accorder l'horloge de réception avec celle d'émission de façon à éliminer entièrement l'effet d'offset. cf fig.5.

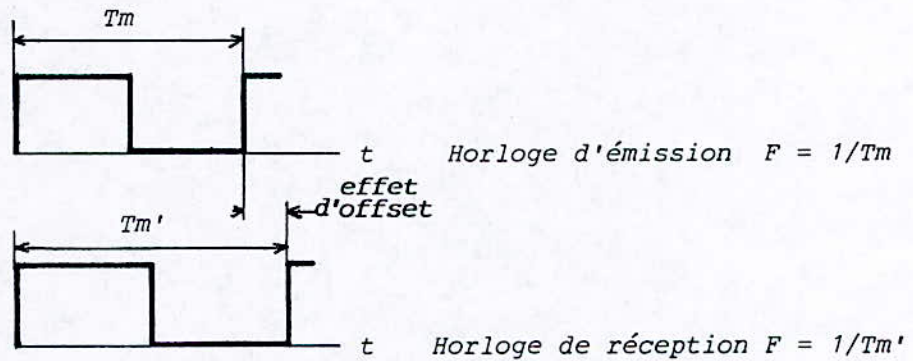
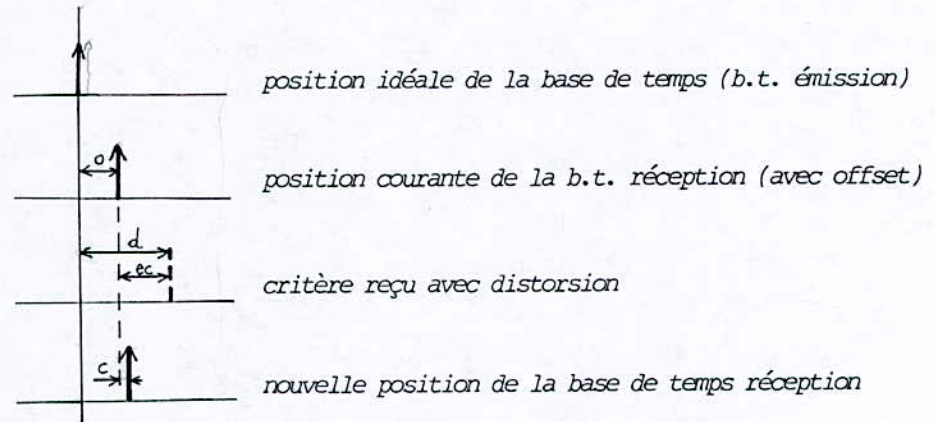


Fig. 4 Offset de fréquence et sa manifestation

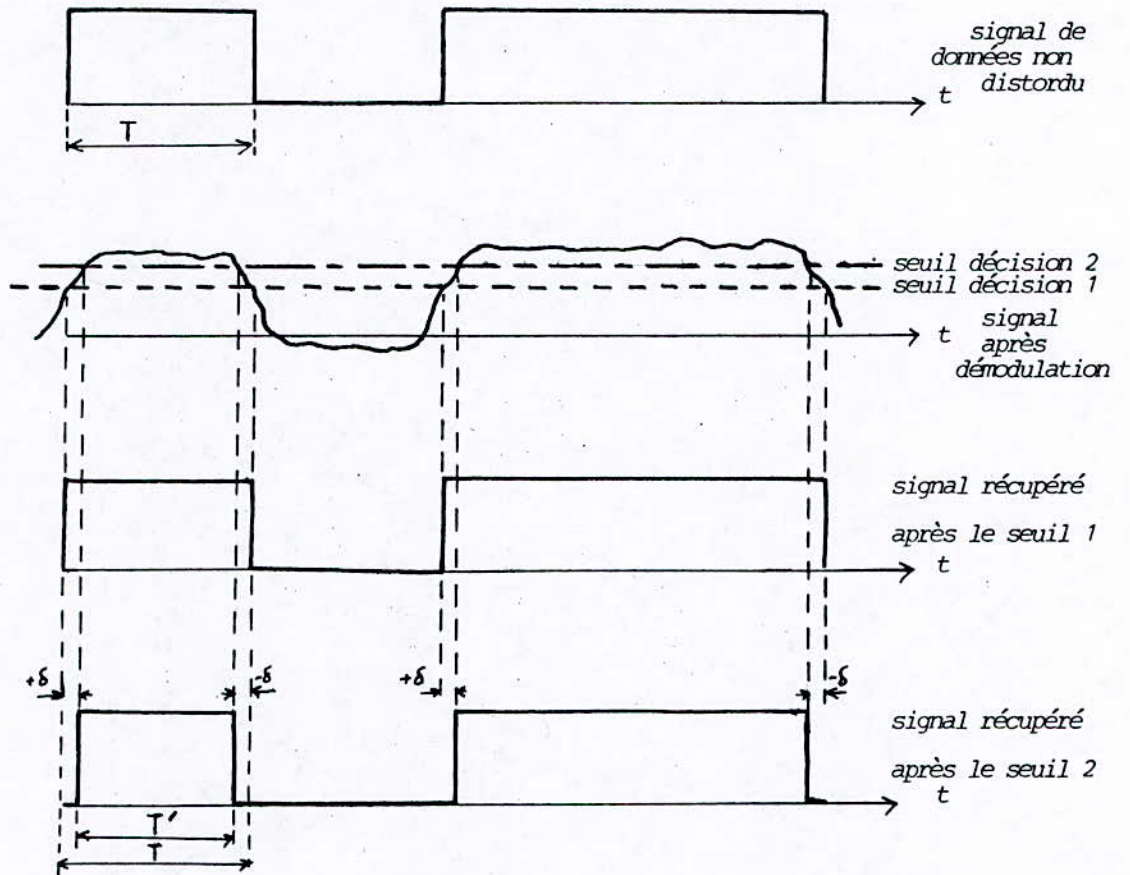
2.6.3 La synchronisation dépend aussi de la structure des données : en effet, s'il n'y a pas de transition, il n'existe pas de critère pour synchroniser.



*o* : offset  
*ec* : écart mesuré  
*d* : écart réel  
*c* : correction

Fig. 5 Influence de l'offset

La correction se fait par rapport à la base de temps de réception avec offset : par rapport à "ec" et non pas par rapport à "d".



$\delta$ : distorsion

Le seuil 1 est dit idéal.

Fig. 6 Distorsion biaisée

## CHAPITRE 3

### BOUCLE A VERROUILLAGE DE PHASE



### 3.1 INTRODUCTION

La synchronisation consiste à accorder un générateur local avec le critère (de synchronisation), le fonctionnement du dispositif de synchronisation est modélisé par une boucle à verrouillage de phase.

### 3.2 GENERALITES

La boucle à verrouillage de phase ou PLL (phase locked loop) est un système bouclé dans lequel la grandeur asservie est la phase.

Le principe a été étudié une première fois par de Bellescize en 1932.[3]

Grâce aux progrès de la technologie d'intégration, il est aujourd'hui possible de réaliser des montages simples et économiques avec des circuits intégrés PLL.

Quelques unes de ces applications:

- la démodulation synchrone;
- la démodulation de fréquence;
- la multiplication de fréquence;
- l'asservissement de vitesse;

et bien d'autres très nombreuses.[3]

### 3.3 PRINCIPE DE FONCTIONNEMENT

#### Constitution générale

#### 3.3.1 Schéma fonctionnel d'une PLL

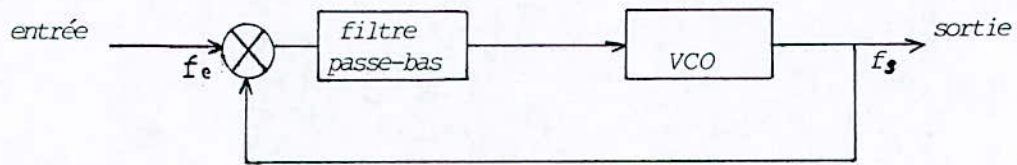


Fig. 7 Schéma fonctionnel d'une PLL

*N.B.* L'approche théorique est la même, que l'on raisonne en grandeur phase, pulsation ou fréquence.

Le passage de la fréquence à la pulsation se fait en utilisant le coefficient  $2\pi$ . Le schéma-bloc reste le même.

#### 3.3.2 Les éléments constitutifs d'une PLL (cas général):

Comme le montre le schéma fonctionnel, ils sont au nombre de trois:

- le comparateur de phase;
- le filtre passe-bas;
- l'oscillateur contrôlé en tension.

a) Le comparateur de phase

Ce circuit compare la phase de deux signaux et fournit une tension d'erreur  $u(t)$ .

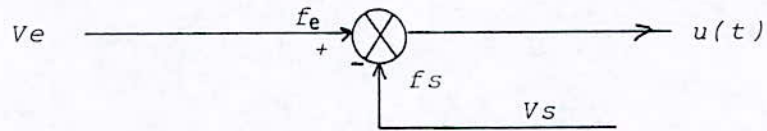


Fig. 8

La tension  $u(t)$  récupérée à la sortie du comparateur de phase comprend:

- des harmoniques de fréquence  $2 f_e$  et plus;
- une composante continue  $U_0$  proportionnelle, directement exploitable.

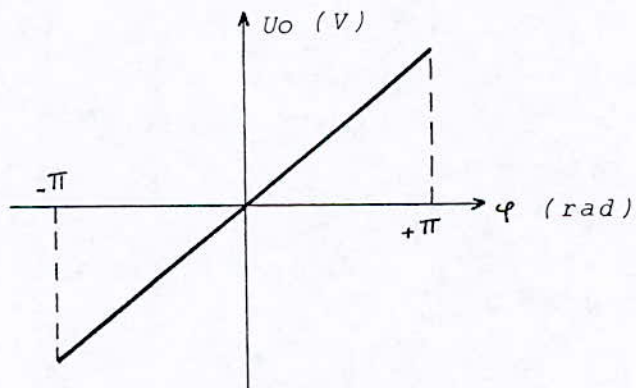


Fig.9 Caractéristique linéaire d'un comparateur

b) Le filtre passe-bas

Sa fonction est d'extraire la valeur moyenne  $U_0$  de tension d'erreur en éliminant tous les harmoniques de rang supérieur à la sortie du détecteur de phase.

c) L'oscillateur contrôlé en tension ou VCO

( VCO = Voltage Controlled Oscillator )

Sa fréquence varie à l'image d'une tension. Il est défini par une caractéristique  $f_s(U_0)$ .

Un VCO est un montage pouvant travailler dans une gamme de fréquence importante. Il possède une fréquence centrale  $f_0$  obtenue en choisissant les bonnes valeurs des éléments  $R, C$  associés à l'oscillateur. Ces valeurs sont en général indiquées par des abaques.

Le VCO d'une PLL bouclée, à entrée nulle, oscille à sa fréquence centrale  $f_0$ .

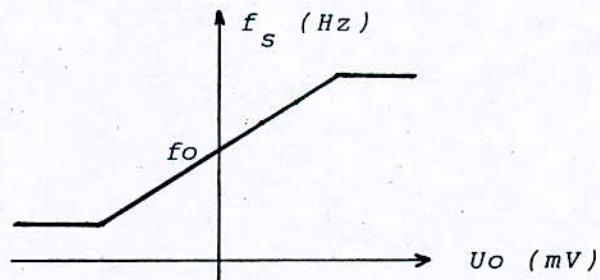


Fig. 10 Caractéristique  $f_s(U_0)$  d'un VCO



### 3.3.3 Fonctionnement d'une PLL - cas général - [4]

Il existe deux plages caractéristiques de fonctionnement correct de la PLL: la plage de verrouillage et la plage de capture.

a) La plage de verrouillage ou de maintien (lock range) :

C'est la bande de fréquence où la PLL reste verrouillée, ceci correspond à l'écart de fréquence par rapport à la fréquence centrale  $f$ , écart à partir duquel la PLL se déverrouille.

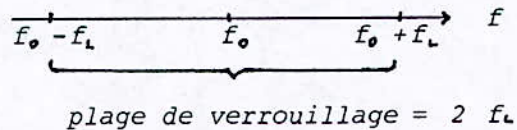


Fig. 11

Il y a verrouillage lorsque la fréquence des signaux d'entrée et de sortie est identique, il en résulte que  $f_s = f_e$ .

b) La plage de capture ou d'accrochage (pull in range):

C'est la bande de fréquence où la PLL se verrouille, elle correspond à l'écart de fréquence par rapport à la fréquence centrale, écart à partir duquel la PLL se verrouille.

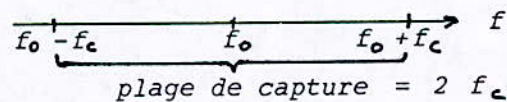


Fig. 12



c) Le verrouillage:

Soit un échelon de fréquence  $f_1$ , cf fig. 13.

A l'instant où la fréquence du signal d'entrée passe de la zone de non-verrouillage à la zone de capture ( $f_0 - f_c, f_0 + f_c$ ), le signal fourni par le détecteur de phase  $u(t)$  est de fréquence  $f_0 - f_1$ . Comme nous sommes dans la bande passante du filtre, la tension  $U(t)$  est aussi de fréquence  $f_0 - f_1$ . Mais très rapidement, cette situation évolue, car le VCO, recevant une tension variable, fournit une fréquence  $f_e$  variable. Il existe alors un écart "temporel" de fréquence; et, dû au fait de l'évolution de la fréquence du signal de sortie dans le sens de la diminution, cette fréquence  $f_s$  se déplace vers celle du signal incident  $f_e$ , jusqu'à ce qu'il y ait coïncidence; à ce moment-là, la boucle est verrouillée.

La durée du processus d'accrochage dépend en premier lieu de l'écart initial des fréquences  $f_0$  et  $f_1$ . Cette durée est appelée pull-in-time.

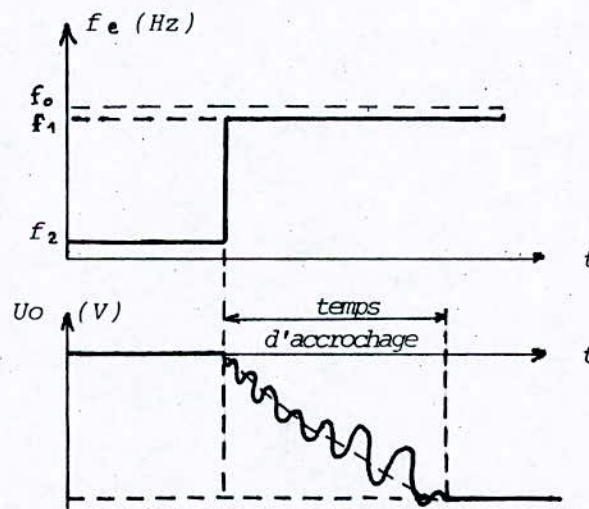


Fig.13 Temps d'accrochage dans le cas d'un échelon de fréquence

### 3.3.4 PLL numérique

La boucle à verrouillage de phase numérique modélisée dans notre simulation l'est selon la figure ci-dessous:

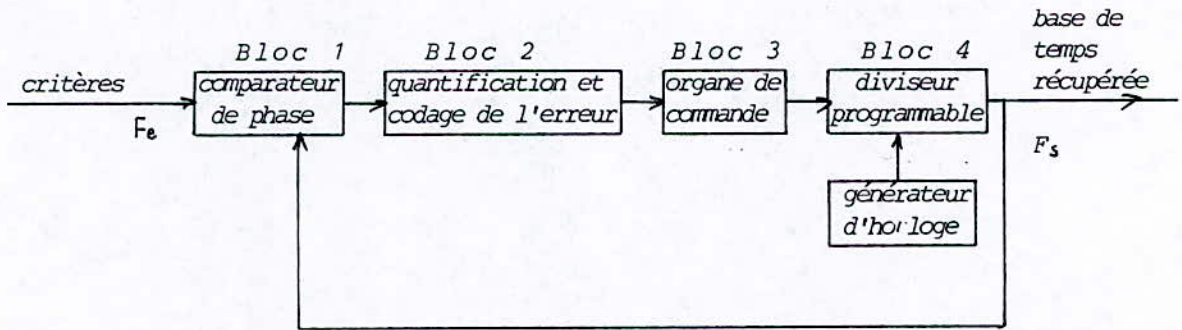


Fig. 14

L'erreur est quantifiée: à chaque valeur de la sortie du comparateur de phase correspond une valeur de correction discrète, la correction se fait pas à pas.

Bloc 1 : Le comparateur a le même fonctionnement que pour la PLL analogique.

Bloc 2 : Il correspond au filtre passe-bas de la PLL analogique; il permet de modéliser la caractéristique de la PLL.

Bloc 3 : Il traduit l'erreur en commande pour le diviseur programmable.

Bloc 4 : Il accélère ou retarde le signal d'horloge issu de l'horloge.

Les blocs 3 et 4 constituent le VCO.

CHAPITRE 4

**SIMULATION**

#### 4.1 INTRODUCTION

Le programme conçu permet d'analyser par simulation le comportement d'une boucle à verrouillage de phase numérique en fonction de plusieurs paramètres choisis.

L'analyse théorique d'une telle boucle étant très compliquée ( cf [4] chap. 4 ), il est parfois plus intéressant de faire une simulation, qui est un outil d'analyse plus simple, plus souple et plus rapide.

Il faudra donc simuler en plus de toutes les parties de la boucle, une structure de données qui sera la source des critères.

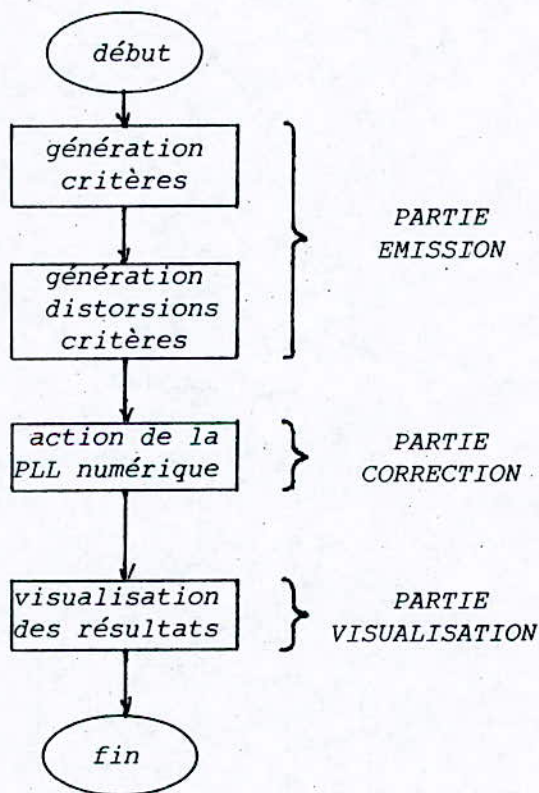


Fig. 15



#### 4.2 DESCRIPTION

La simulation comporte 3 parties principales:

- la partie émission;
- la partie réception;
- la visualisation.

##### 4.2.1 Partie EMISSION

a) La source de données: Il s'agit de simuler une source de données binaires, suite de 0 et de 1, pour pouvoir utiliser par la suite les transitions de ce signal de données. La source de données représente la source d'émission de critères utilisés par la boucle.

Le programme offre 2 possibilités de génération de critères:

- à partir d'une source binaire pseudo-aléatoire:

Il s'agit de la génération d'une séquence pseudo-aléatoire, les apparitions de 0 et de 1 sont indépendantes et équiprobables et ceci pour 511 éléments, puis la séquence se répète; elle est dite pseudo-aléatoire. Cette séquence est obtenue à l'aide d'un registre à décalage à 9 éléments bouclé selon la figure suivante: [5]

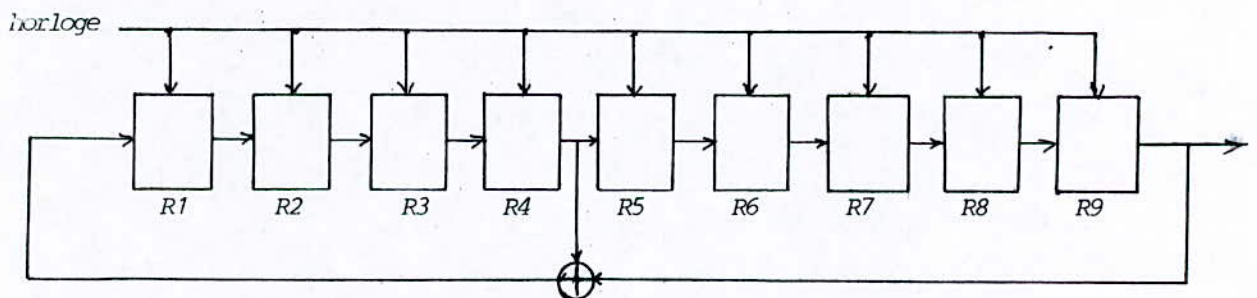


Fig. 16

La séquence comprend  $2^N - 1 = 512 - 1 = 511$  éléments, qui est la valeur recommandée par le CCITT [ 6 ] du nombre de bits à émettre pour tester un équipement de traitement numérique, car ainsi, on approche assez bien les propriétés d'une séquence aléatoire vraie et il est possible de répéter plusieurs fois le test dans les mêmes conditions (même séquence).

Dans le programme, le registre à décalage est un vecteur où les éléments sont décalés, en remplissant à chaque fois le 1<sup>o</sup> élément par le résultat d'un OU EXCLUSIF entre le 4<sup>o</sup> et le 9<sup>o</sup> éléments. L'état de la source est pris à chaque décalage comme étant le 9<sup>o</sup> élément du vecteur.

- à partir d'une source binaire périodique: Qui consiste en la répétition d'une séquence donnée de longueur fixée et définie par l'utilisateur du programme. La longueur maximale de cette séquence est de 9.

A la demande du programme, il faut introduire la longueur de la séquence désirée puis tous les éléments de la séquence, un à un . Le programme se charge de répéter celle-ci et de générer les transitions correspondantes ie. les critères.

Il faut maintenant simuler la source de distorsion:

- l'offset;
- les distorsions( gaussienne, uniforme ou biaisée).

b) *L'offset de fréquence:*

*Les horloges d'émission et de réception doivent avoir théoriquement la même fréquence. Or, dans la réalité, il existe généralement un écart de fréquence entre elles. (positif ou négatif).*

*Lorsqu'il n'y a pas de critère (absence de transition), il ne peut y avoir correction, et l'offset est présent sans être corrigé.*

*De même, s'il y a une transition : un critère, et donc correction, mais que celle-ci soit trop petite, l'effet de l'offset n'est pas rattrapé, et la base de temps de réception donnera l'impression de fuir, il est impossible de conserver la synchronisation, ce qui fait apparaître des erreurs. Ceci peut être visualisé en ne choisissant pas de distorsion et en prenant une correction inférieure à la valeur de l'offset.*

c) *Les distorsions:*

*Les critères reçus sont généralement distordus. Le programme peut générer 3 sortes de distorsion:*

- distorsion gaussienne: il faudra préciser la moyenne et la variance de la distribution que le programme générera. [5]*
  
- distorsion uniforme: le programme génère des distorsions appartenant à une répartition uniforme en utilisant le générateur de nombres aléatoires de l'ordinateur. Ces nombres étant équi-distribués sur  $[0,1]$ , après multiplication et translation, le programme génère des nombres distribués sur  $\pm$  une valeur limite donnée par l'utilisateur. (Pour mieux comprendre le fonctionnement de la boucle, nous avons introduit cette distorsion dans notre modèle.)*



- *distorsion biaisée*: soit  $B$  la valeur absolue de cette distorsion. A chaque nouvelle transition, la distorsion prendra alternativement comme valeur :  $+B$  ,  $-B$  ,  $+B$  ,  $-B$  ...  
A la demande du programme, il faudra introduire la valeur de la distorsion en pourcentage.

*La distorsion biaisée peut être ajoutée à l'une des deux autres.*

**NB** *La valeur maximale recommandée par le CCITT [6] pour l'offset de fréquence est de  $\pm 6\text{Hz}$ .*



#### 4.2.2 Partie RECEPTION

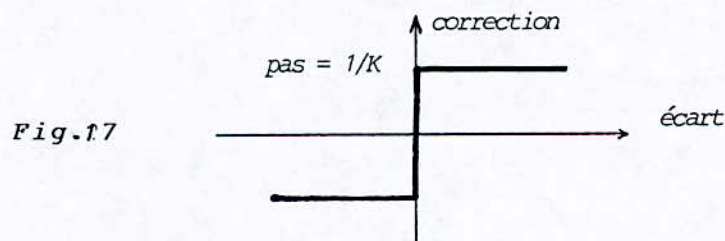
La correction: Son action s'exerce sur la position de la base de temps et est fonction de l'écart de celle-ci avec le critère reçu i.e. avec la distorsion. La distorsion n'est mesurable que lors des transitions qui constituent les critères de synchronisation. C'est pourquoi il n'y a correction qu'en la présence de transitions.

La correction est définie par sa caractéristique, qui donne la valeur de la correction à effectuer en fonction de l'écart observé entre la position de la base de temps et la distorsion.

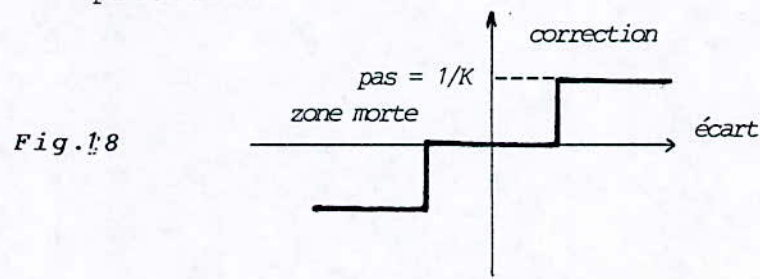
Il existe plusieurs sortes de caractéristiques:

a) A pas unique: Correction binaire

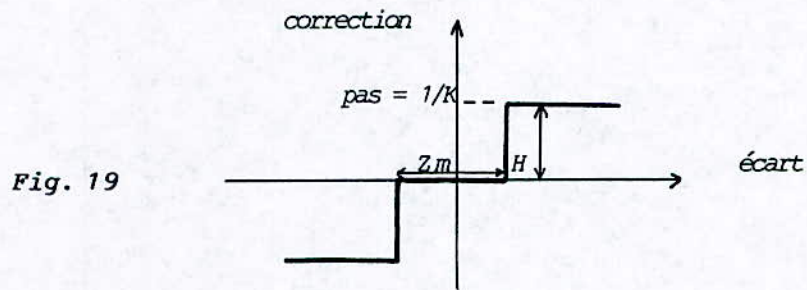
- sans zone morte: C , pas de correction est choisi par l'utilisateur.



- avec zone morte: Si l'écart est inférieur à une certaine valeur choisie par l'utilisateur, il tombe dans ce qui est appelé la zone morte, et aucune correction n'est effectuée. Autrement, la correction est retranchée comme exactement de la même manière que pour la caractéristique précédente.

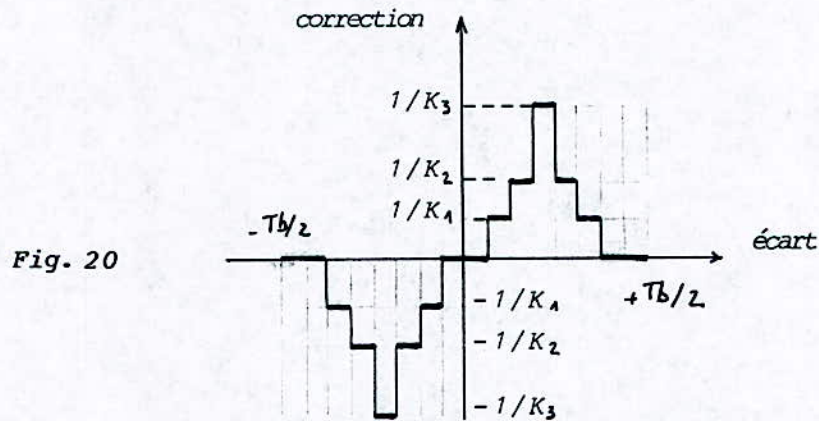


b) A pas multiples, avec ou sans zone morte



$Z_m$ ,  $L$  et  $H$  sont définis par l'utilisateur.

c) Définissables par l'utilisateur; en donnant la valeur du pas de correction pour chaque portion de caractéristique, après en avoir fixé le nombre.



Exemple de caractéristique définie par l'utilisateur

Plus grand est le nombre de segments, meilleure est la définition de la caractéristique.

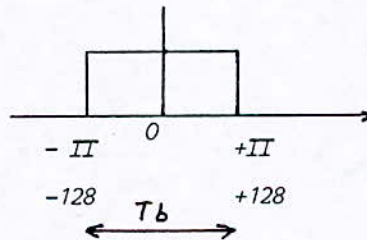
#### 4.2.3 Visualisation, représentation graphique adoptée

Prenons pour repère relatif le milieu d'un élément de donnée. L'écart peut donc varier de  $-T_b/2$  à  $+T_b/2$ : une  $1/2$  période de chaque côté, soit de  $-\pi$  à  $+\pi$  pour le déphasage.

L'écran graphique de l'ordinateur ayant une résolution typique (VGA, EGA) de 640 pixels en largeur, en supposant que l'on réserve une moitié d'écran pour la visualisation d'autres informations, il reste une largeur de 320. Une représentation sur 256 pixels a été adoptée: l'écart maximal  $+\pi$  correspondra à +128 pixels et  $-\pi$  à -128.

En résumé, sur l'écran, par rapport à une position idéale (celle de la base de temps d'émission), l'écart pourra varier jusqu'à 128 pixels à droite et autant à gauche à partir d'une position centrale représentant le zéro.

Fig. 21



Pour la correction:

La plage de variation de la base de temps est de 256 pixels, ainsi, un pas de correction de

- $1/256$  correspond à une correction de 1 pixel;
- $1/128$  : 2 pixels;
- $1/64$  : 4 pixels;
- $1/32$  : 8 pixels;
- $1/16$  : 16 pixels.



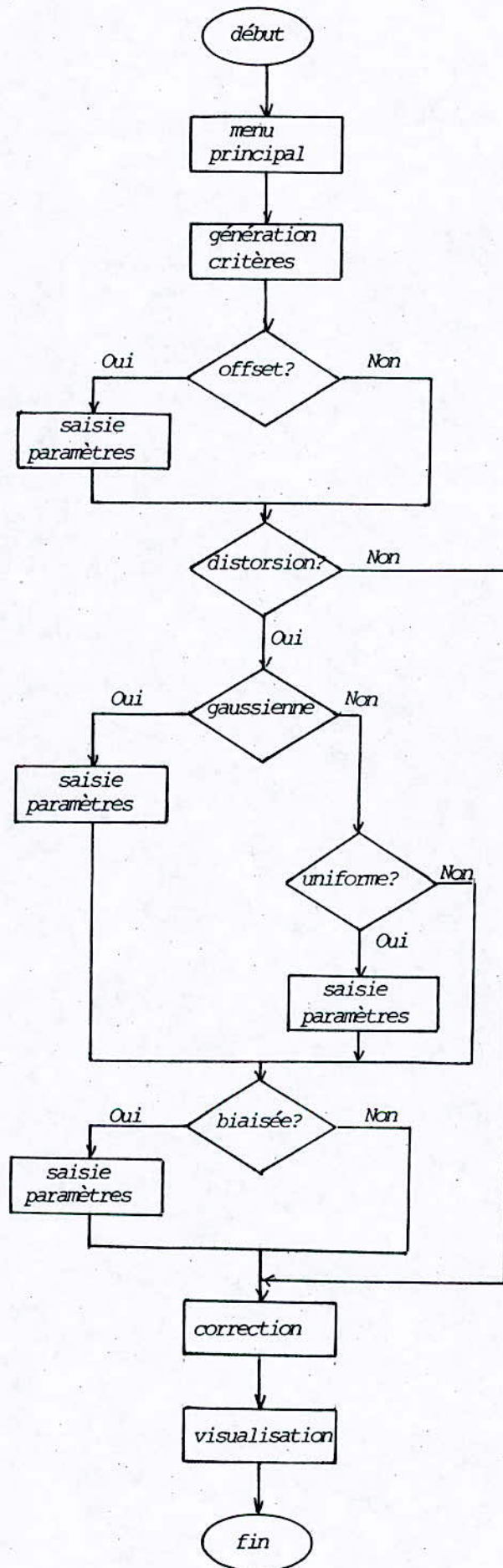


Fig.22 Organigramme général



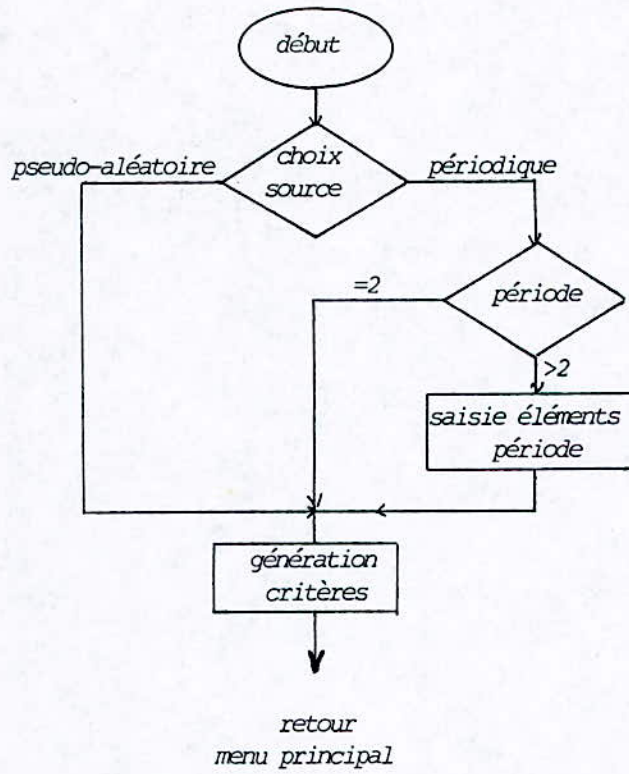


Fig.23 Génération des critères

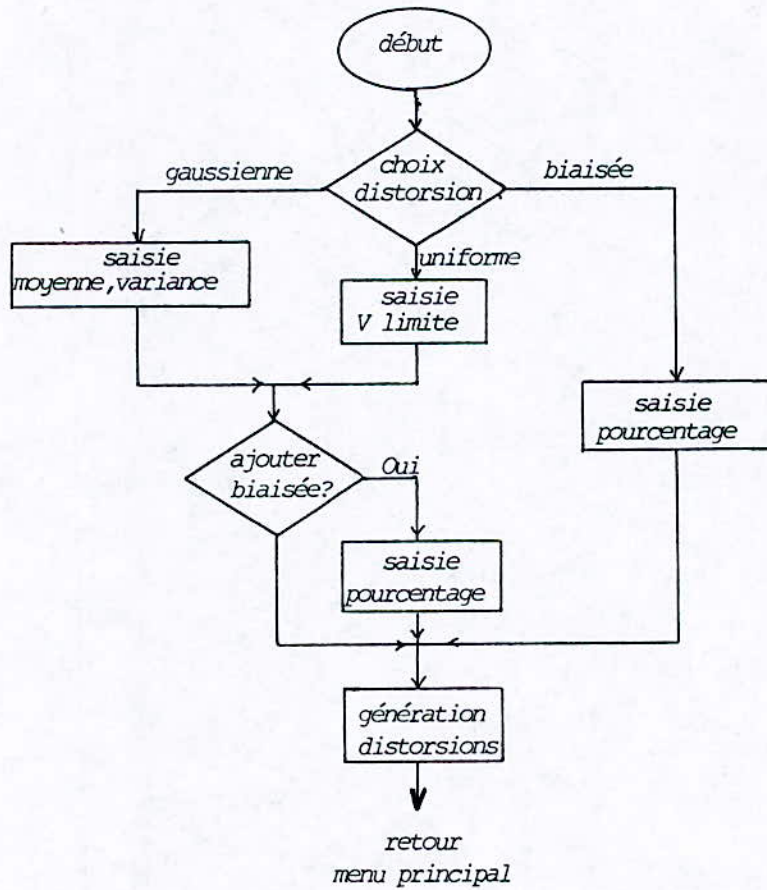


Fig.24 Génération des distorsions

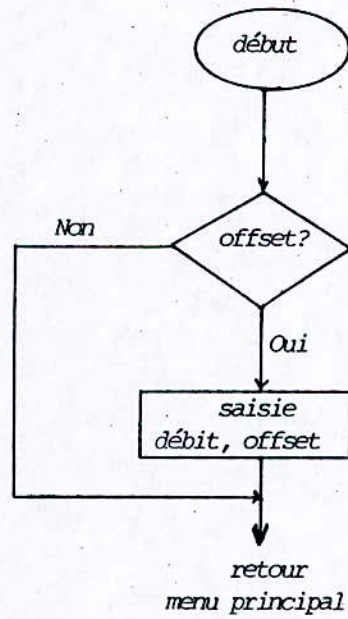


Fig.25 Génération de l'offset

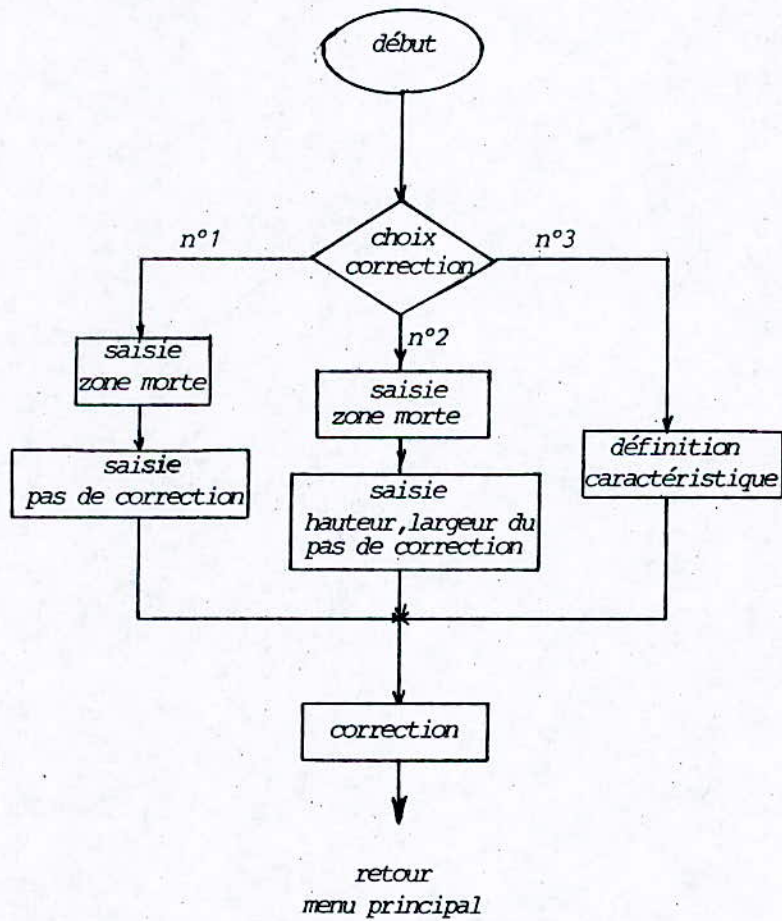


Fig.26 La correction

CHAPITRE 5

UTILISATION DU PROGRAMME

### 5.1 MODE D'UTILISATION

Vous avez lancé l'exécution du programme à partir du DOS avec: **SIMUL** <ENTER>.

1. Après lancement, au milieu de l'écran apparaît:

*SIMULATION D'UNE BOUCLE A VERROUILLAGE DE PHASE NUMERIQUE*

*Appuyer sur la barre d'espace pour poursuivre.*

2. Vous vous trouvez dans le **MENU PRINCIPAL**, l'écran affiche:

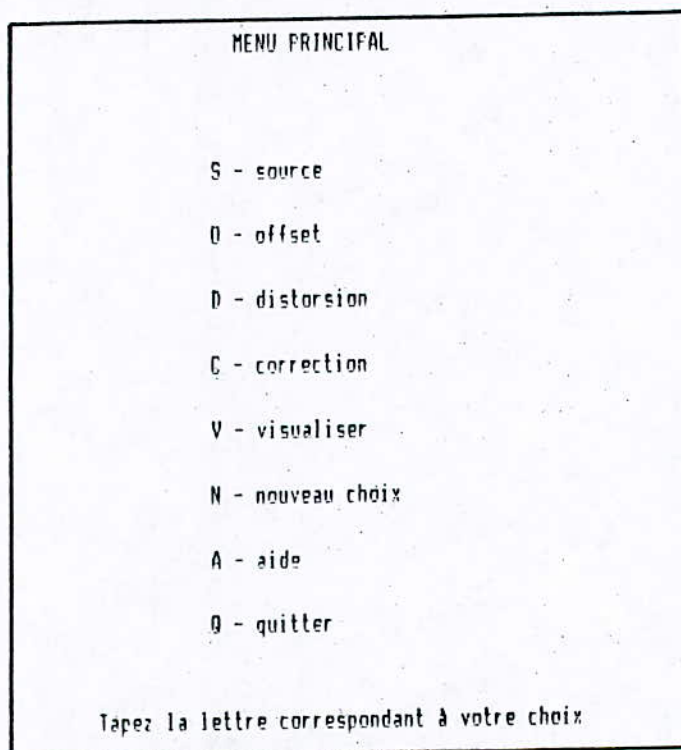


Fig. 27

et attend votre réponse .

3. Les choix possibles:

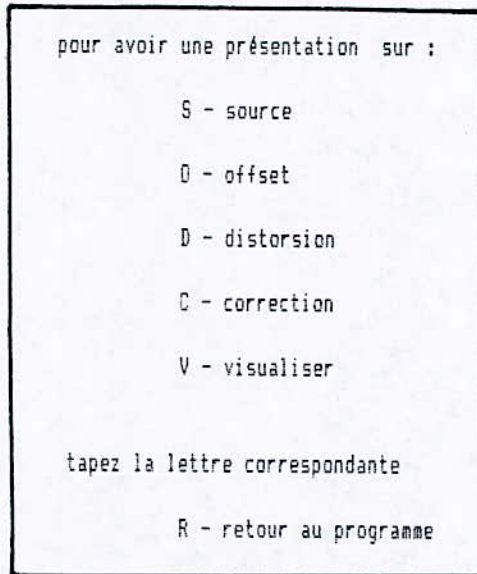
**Q** : pour quitter le programme lorsque vous désirez cesser de travailler l'écran affiche alors:

Etes vous sûr de vouloir quitter le programme? O/N

Appuyez sur **N** pour retourner au menu principal ou sur **O** : vous avez rejoint le DOS.



*A* : appelle un menu secondaire, celui de l'aide.



*Fig.28*

Appuyez sur la touche correspondant à l'aide voulue. Ensuite, pour retourner au MENU PRINCIPAL, appuyez sur la touche R.

*N* : correspond à l'initialisation de toutes les données. La source, la ou les distorsions et la correction sont désélectionnées.

4. Pour débuter la simulation, il faut tout d'abord choisir une structure pour la source binaire de données. En appuyant sur S. Puis si vous le souhaitez, introduire un offset avec O et ensuite une distorsion avec D . Et enfin une correction en tapant C.

Pour visualiser l'action de la boucle, c'est à dire la synchronisation, appuyez sur V.

Nous allons voir en détail ces étapes:

*S* : choix de la structure de la source de données. Le menu suivant apparaît:

```

CHOIX DE LA SOURCE

appuyer sur

P pour engendrer un signal Pseudo-aléatoire binaire
B pour engendrer un signal Binaire périodique

M pour revenir au Menu précédent

```

Fig. 29

- en appuyant sur P , vous choisissez la structure pseudo-aléatoire 511.
- en appuyant sur B, vous choisissez une source de données binaire périodique. Le programme vous demande la longueur de la période: introduisez un nombre entre 2 et 9. Si vous avez choisi 2, la source automatiquement sera 1-0-1-0-1-0 etc. Sinon, il vous sera demandé d'introduire l'un après l'autre, tous les éléments de la période. Vous devez entrer des 0 et des 1 selon votre choix.

Exemple: vous avez choisi d'émettre 1-0-0-1 1-0-0-1...  
la séquence 1-0-0-1 se répétant:

```

EMISSION D'UNE SOURCE BINAIRE PERIODIQUE

choisir la longueur de la période - entre 2 et 9 -

```

Fig. 30

Vous tapez 4 . Puis

```

entrer l'élément 1 de la séquence : 1
entrer l'élément 2 de la séquence : 1
entrer l'élément 3 de la séquence : 0
entrer l'élément 4 de la séquence : 0

```

Fig. 31

Et un message vous annonce que la source est choisie et activée. Vous devez appuyer sur une touche quelconque.

O : pour introduire un OFFSET.

Introduisez d'abord le débit en hertz.

Puis la valeur de l'offset de fréquence (positive ou négative).

Vous lisez ensuite le message:

```
Vous avez introduit un offset
appuyez sur
n'importe quelle touche
```

Fig.32

Rq: Pour introduire un offset, il faut avoir préalablement choisi et activé la source. (idem pour la distorsion)

Pour annuler l'offset, il suffit de taper O pour resélectionner l'offset et à la demande :

VOULEZ VOUS INTRODUIRE UN OFFSET?

tapez N(pour non) .

D : Pour introduire une distorsion.

Le menu suivant apparaît:

```
CHOIX DE LA DISTORSTION
appuyer sur
G pour distribution Gaussienne
U pour distribution Uniforme
B pour distribution Biaisée
M pour revenir MENU principal
```

Fig.33

Vous pouvez choisir une distorsion gaussienne, uniforme ou biaisée.

- Si vous choisissez une gaussienne, vous devez entrer la moyenne(positive et <2), et la variance (>0.3).

- Si vous choisissez une uniforme, vous devez entrer la limite de la répartition. La limite maximum étant de 128.



- Si vous choisissez une biaisée: vous devez introduire le pourcentage. Vous vous retrouvez ensuite dans le MENU PRINCIPAL.

Si vous avez opté pour l'une des 2 premières distorsions, une fois ses paramètres fixés, vous avez la possibilité de lui ajouter la distorsion biaisée, il vous suffit d'appuyer sur B et d'introduire sa valeur en %.

Vous ne voulez pas ajouter une distorsion biaisée? Appuyez alors sur M et vous êtes de nouveau dans le MENU PRINCIPAL.

Vous avez fixé et activé:

- la structure de la source de données
- l'offset ou/et la distorsion

Pour activer la boucle à verrouillage de phase, appuyez sur C.

C : Vous avez activé la correction. Il vous faut maintenant la choisir. 3 possibilités s'offrent à vous:

```
CHOIX DE LA CORRECTION

correction 1 : caractéristique binaire
correction 2 : caractéristique en escalier
correction 3 : construction de la caractéristique

choisir le numéro de correction 1, 2 ou 3 puis <ENTER>:
```

Fig. 34

**CORRECTION 1:** de caractéristique binaire.

Vous voulez une caractéristique avec zone morte? tapez 0 lors de la demande; puis introduisez sa valeur :  $\pm 32$ ,  $\pm 16$  ou  $\pm 8$

Entrez maintenant la valeur du pas de correction  $1/K$ , entrez K: 256, 128, 64, 32 ou 16.



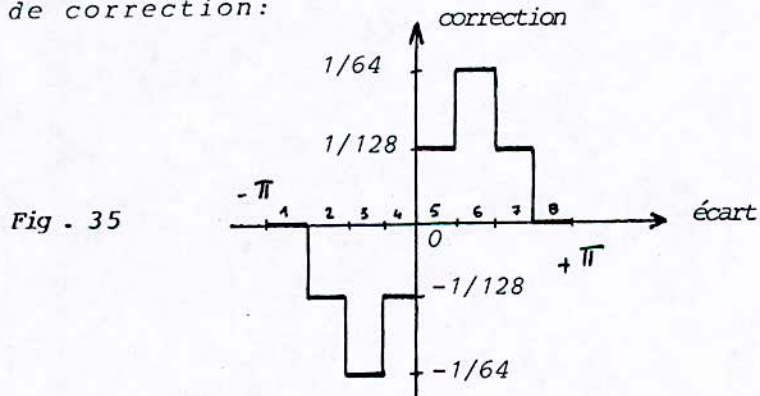
**CORRECTION 2: de caractéristique en escalier.**

Choisissez ou pas la zone morte. Il faut ensuite entrer le pas en largeur, puis le pas en hauteur: entrez K correspondant. Ne pas oublier que le pas en largeur doit être supérieur à celui en hauteur i.e.: K entré pour la hauteur doit être plus grand que celui introduit pour la largeur, sinon, le choix n'est pas accepté.

**CORRECTION 3: de caractéristique à construire par l'utilisateur.**

Introduire d'abord en combien de segments se partagera la caractéristique en abscisse. (nombre à sélectionner parmi 2, 4, 8, 16, 32, 64, 128 et 256). Ensuite pour chaque segment, il faut donner la valeur du pas de correction.

Exemple: vous choisissez cette caractéristique de correction:



Le nombre de segments est: 8

Correction 1°segment: 0

2° : -1/128

3° : -1/64

4° : -1/128

5° : 1/128

6° : 1/64

7° : 1/128

8° : 0

V : Pour visualiser la simulation.

Si vous n'avez pas fait de correction, ceci est impossible.  
La visualisation sera différente selon que vous ayez  
choisi un offset seul ou non.

a) Ceci permet, si l'on a choisi un offset mais pas de  
distorsion, d'observer le comportement de la base de  
temps en la présence d'offset.

Il est aussi possible d'observer le comportement de  
cette même base de temps après synchronisation ( ceci  
équivalent à observer la sortie de la boucle pour une  
entrée avec offset mais avec distorsion nulle).

Il y a possibilité de choisir l'option défilement  
lent ou à vitesse normale.

↑ : représente la position de la base de temps i.e. la  
sortie de la boucle de synchronisation.  
Les pointillés délimitent la zone où l'on considère  
bonne la position de la base de temps.

b) Vous avez activé une distorsion et éventuellement  
un offset.

Vous pouvez choisir un défilement lent ou pas.

Le cadre de gauche montre l'évolution de la base de  
temps. Les traits en pointillés délimitent la zone où  
la position de la base de temps est jugée correcte.

↑ : indique la position de la base de temps  
| : indique la position du critère (de la distorsion).

Le cadre supérieur droit donne la répartition des critères (des distorsions) avant correction.

Le cadre inférieur droit donne la répartition de la base de temps à la sortie de la boucle de correction.

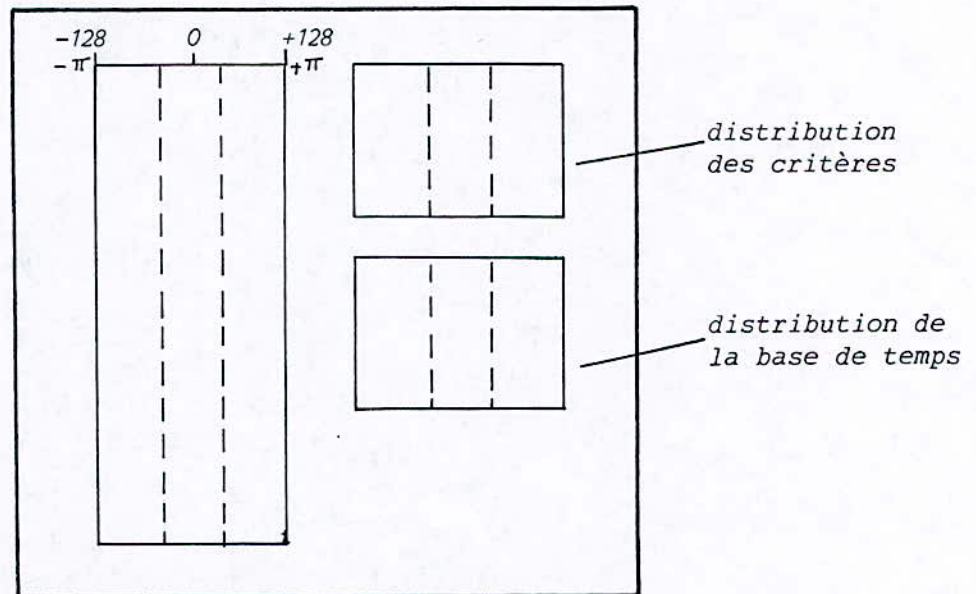


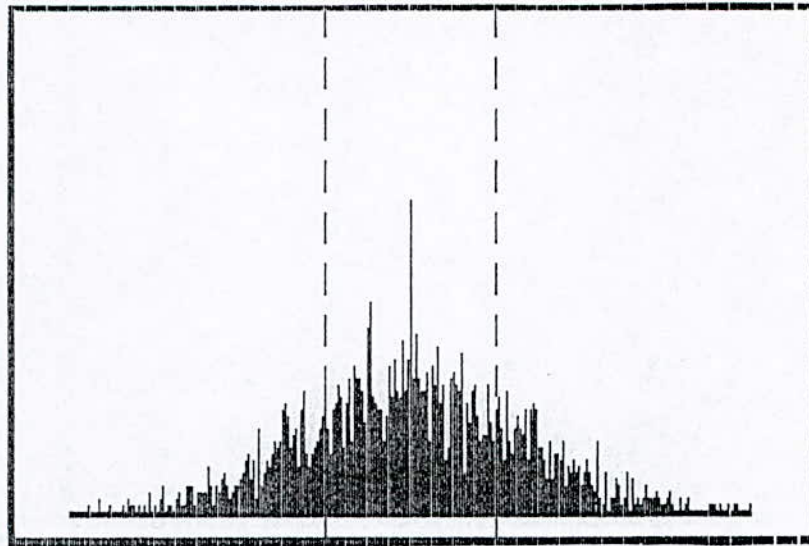
Fig.29 Ecran

Le nombre de périodes d'horloge écoulées est affiché:39  
En appuyant sur:

- la barre d'espacement, vous visualisez pour les 40 périodes suivantes, l'évolution de la base de temps.
- F ,vous obtenez la distribution après 2000 périodes d'horloge. Vous pouvez poursuivre pour les 2000 suivants avec P .(le maximum est de 20 000 périodes).
- T ,pour terminer et revenir au menu principal.
- I ,pour afficher les choix concernant la source, la valeur de l'offset, la distribution de la distorsion et la correction.

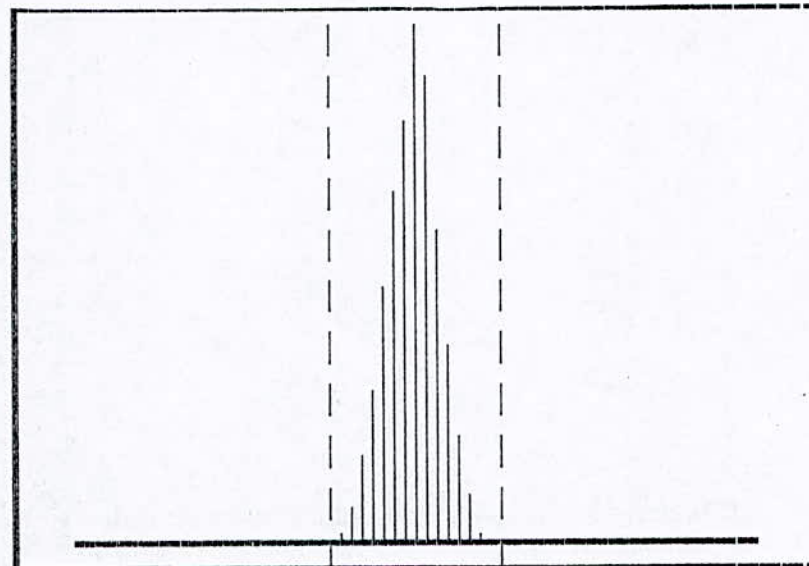


- A -



*Distribution des critères selon loi gaussienne*

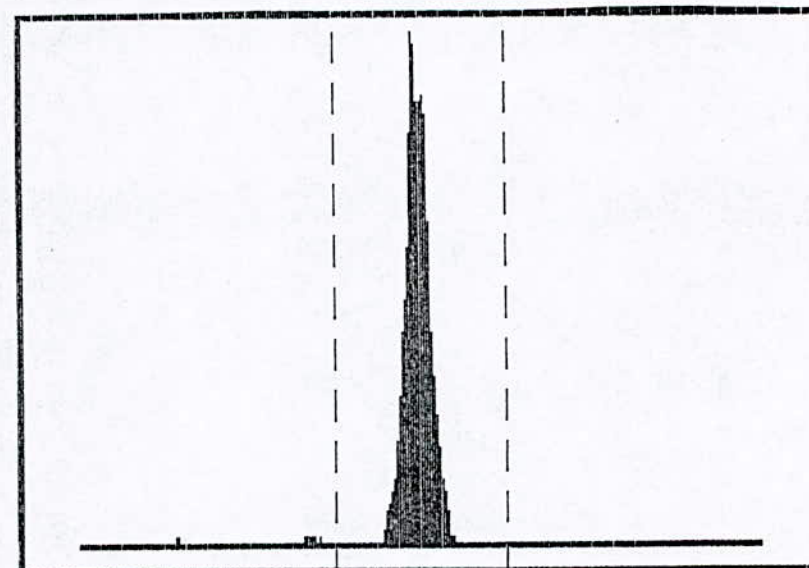
- B -



*Distribution de la b.t.*

*Correction binaire de pas = 1/ 64*

- C -

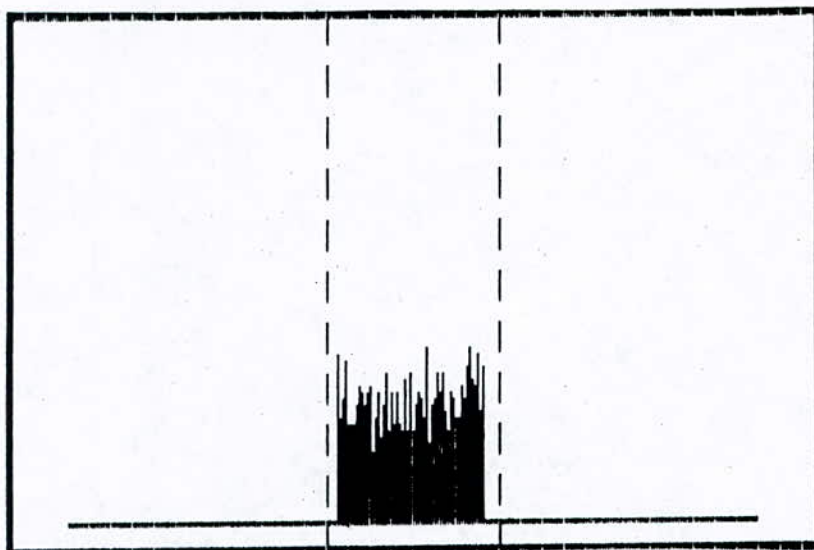


*Distribution b.t.*

*Correction binaire de pas = 1/ 256*

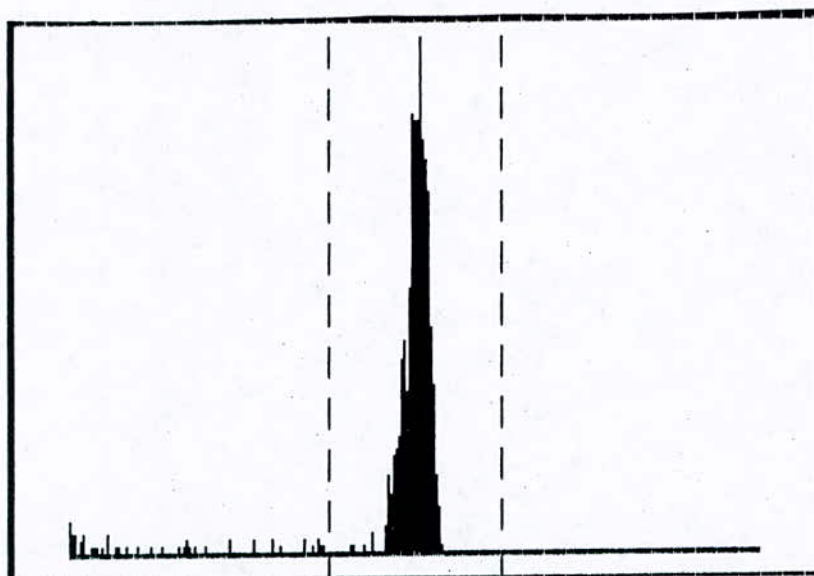


- D -



Distribution des  
critères selon  
uniforme

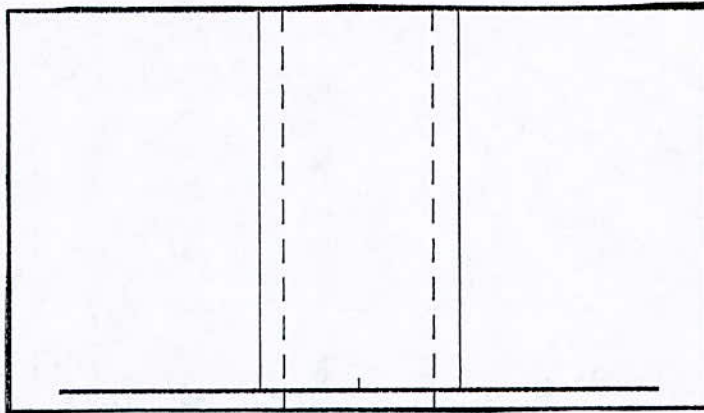
- E -



Distribution b.t.  
  
Correction binaire  
de pas = 1/ 256

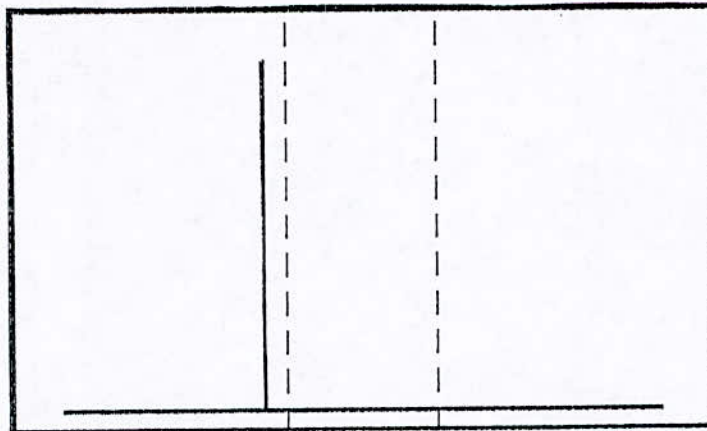
Bien que la distribution des critères se trouve  
entièrement dans la zone dite de synchronisation,  
la boucle permet d'en réduire la variance.

- F -



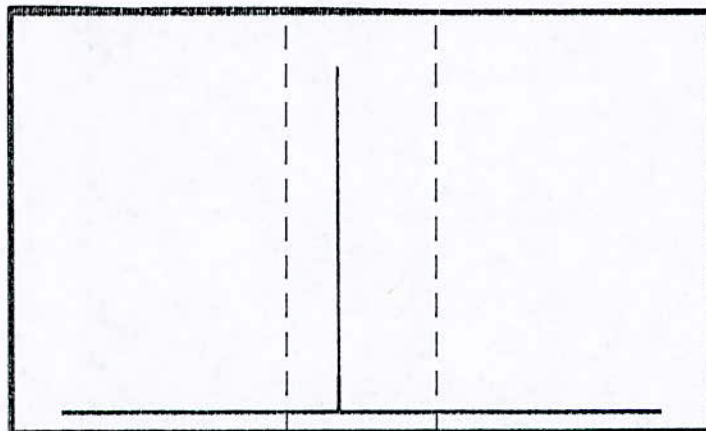
Distribution des  
critères selon  
loi biaisée

- G -



Distribution b.t.  
Correction binaire  
de pas = 1/ 256

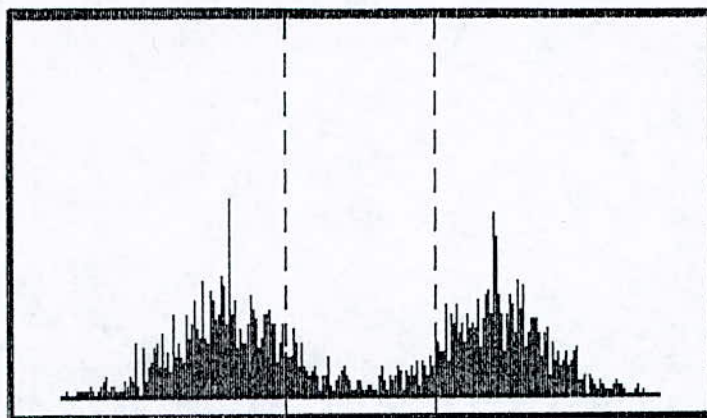
- H -



Distribution b.t.  
Correction binaire  
de pas = 1/ 256  
avec zone morte (= 32)

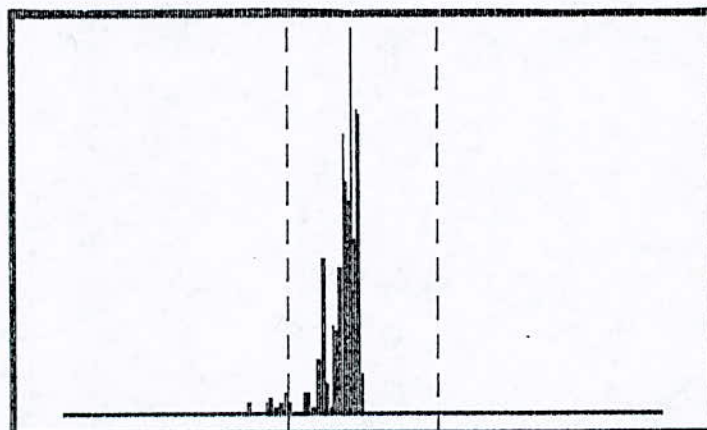
L'introduction d'une zone morte a permis à  
la base de temps de pouvoir entrer dans la  
zone de synchronisation.

- I -



*Distribution des  
critères selon  
loi gaussienne et  
biaisée.*

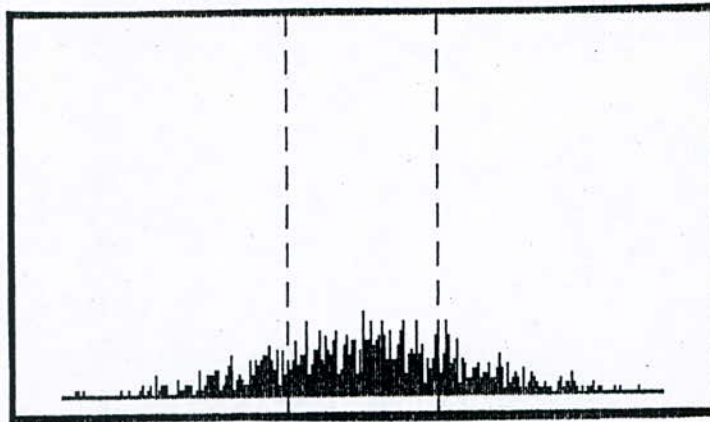
- J -



*Distribution de la b.t.*

*Correction binaire  
de pas = 1/ 256*

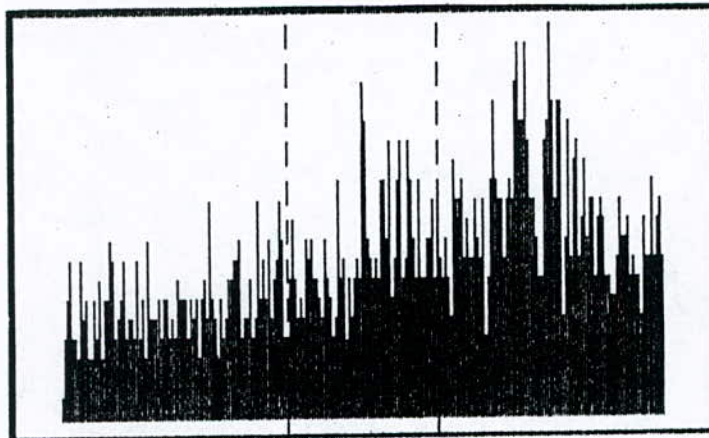
- K -



Distribution des  
critères selon  
loi gaussienne

+ OFFSET de fréquence

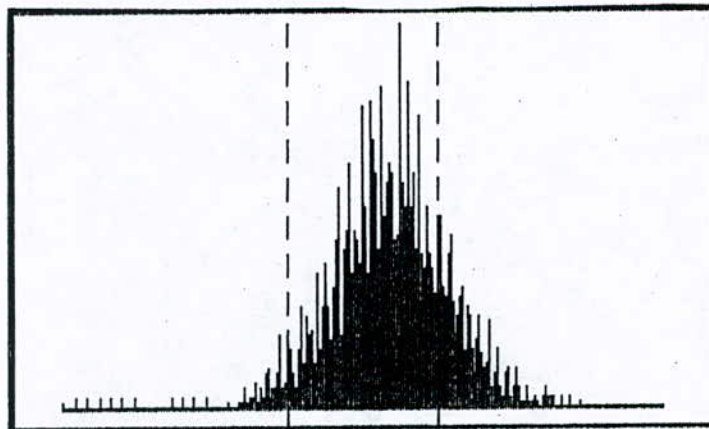
- L -



Distribution de la b.t.

Correction binaire  
de pas = 1/256

- M -



Distribution de la b.t.

Correction binaire  
de pas = 1/32

En -L- , l'effet d'offset n'est pas rattrapé,  
la base de temps ne peut pas être synchronisée,  
car la correction est trop petite. En -M-, le pas  
de correction est plus grand.



CONCLUSION

Le logiciel permet de simuler une boucle de synchronisation. Les multiples applications ont prouvé sa fiabilité et la souplesse de son utilisation.

Il permet à tout expérimentateur de simuler les caractéristiques qu'il a choisies avant de retenir celle qu'il considère comme optimale pour l'appliquer ensuite.

L'expérimentateur doit être avisé que ~~que~~ les irrégularités de la séquence RND (qui n'est pas très aléatoire au début) de l'ordinateur constituent un défaut pour le programme. L'effet de la concentration du début de la répartition disparaît si le temps d'observation est allongé.

Ce programme peut être enrichi, par exemple, en y ajoutant une procédure donnant une analyse statistique de l'entrée et de la sortie de la boucle (moyennes et écarts-types).

*BIBLIOGRAPHIE*

- [1] D. Thel - *Le multiplexage temporel. Polycopié de l'INT d'Evry.*
- [2] C. Macchi et J.F. Guilbert - *Téléinformatique. Ed. Dunod, 1987.*
- [3] W.C. Lindsey - *Phase Locked Loops and their Applications. IEEE Press, 1978.*
- [4] R.E. Best - *Phase Locked Loops, Theory, Design and Applications. McGraw-Hill Inc, 1984.*
- [5] Labarrère, Krief et Gimonet - *Le filtrage et ses applications. (Annexe IV). Ed. Cepadues, 1982.*
- [6] CCITT - *Red Book. Vol.8. Genève, Suisse, 1985.*
- [7] *Turbo Pascal 5.0, Owner's Handbook. Borland, 1988.*
- [8] M. Bellanger - *Traitement numérique du signal, théorie et pratique. Ed. Masson, 1984.*
- [9] G. Pujolle - *Réseaux et télématique. Tome 1. Ed. Eyrolles, 1985.*
- [10] Boukelloul et Engliz - *Thèse de fin d'études : conception et réalisation d'un dispositif de synchronisation pour la transmission numérique avec zone morte. ENP, 1987.*
- [11] J.C. Armici - *Le Turbo Pascal version 4. Editions LI, 1987.*

ANNEXE



```

( SIMULATION D'UNE BOUCLE A VERROUILLAGE DE PHASE NUMERIQUE )
( -- JUIN 1990 -- )

program simulation;
uses graph,crt,sec;
label debut,debut1;
type vecteur=array[0..2009]of integer;
   stri=string[33];
   tableau=array[0..9] of integer;
var giga,ps,t,result,xcorrige:vecteur;
   kchoix,mcar,mcar1:char;
   kl,i,j,a,ymax,xmax,nbiais,cor1,zmorte,position,large,haut,numero:integer;
   testgig,testdist,test:shortint;
   choix,corr:word;
   debit,gi3,xal,xrep,moy,ecart,dis,moyenne,gi9:real;
   testsource,defil,testbiais,encore,tesgaus,bin,tdist:boolean;
   seq:tableau;
   coor:array[0..256] of integer;

(*****)
procedure modegraphique; (initialisation du mode graphique)
var graphdriver:integer;
   graphmode:integer;
   errorcode:word;
begin
   graphdriver:=detect;
   initgraph(graphdriver,graphmode,'');
   errorcode:=graphresult;
   if errorcode<>gok then
   begin
      writeln('erreur graphique, appuyer sur <ENTER> ');
      readln;
      halt(1);
   end;
   setviewport(0,0,640,getmaxy,ClipOff);(dimensions EGA, VGA 640 x 348)
end;

(*****)
procedure bordure;( pour suivre évolution offset (sans distorsion))
begin
   setlinestyle(0,0,3);
   rectangle(29,10,291,340);
   setlinestyle(1,0,1);
   line(128,10,128,340);
   line(192,10,192,340);
end;

(*****)
procedure tracecadre;(pour suivre évolution du critère limite ± 128)
begin
   setlinestyle(0,0,3);
   rectangle(330,10,630,95); (tracé pour évolution distorsion initiale)
   rectangle(330,111,630,200);(tracé pour évolution distorsion corrigée)
   setlinestyle(3,0,1);      (limites tolérables)
   line(512,10,512,200);
   line(448,10,448,200);
end;

```

```
(*****)
(convertit un nombre entier en chaine : utilisé en mode graphique)
function convert(valeur:integer):string;
var st:string[6];
begin
    str(valeur,st);
    convert:=st;
end;
```

```
(*****)
(convertit un nombre réel en chaine)
function convreal(valeur:real):string;
var st:string[6];
begin
    str(valeur,st);
    convreal:=st;
end;
```

```
(*****)
procEDURE info;(donne informations sur le choix fait)
var w:integer;
    myr:char;
begin
    setactivepage(1);
    clearviewport;
    setlinestyle(0,0,3);
    rectangle(95,75,495,250);
    setvisualpage(1);
    settextstyle(0,0,1);
    outtextxy(110,90,'          INFORMATIONS');
    outtextxy(110,110,'SOURCE ----- Binaire');
    outtextxy(110,150,'OFFSET ----- Hz');
    outtextxy(110,170,'DISTORSION --');
    outtextxy(110,190,'CORRECTION --');
    outtextxy(240,190,' Caractéristique:');
    if bin then outtextxy(292,110,'periodique')
    else outtextxy(292,110,'pseudo-aleatoire 511');
    if test)=2 then
    begin
        w:=trunc(debit);
        outtextxy(222,130,' Debit='+convert(w));
    end;
    outtextxy(230,150,convreal(qi9));
    if tdist then
    begin
        if tesgaus then outtextxy(230,170,'gaussienne')
        else outtextxy(230,170,'uniforme');
        if testbiais then outtextxy(320,170,'+ biaisee');
    end
    else outtextxy(230,170,'biaisee');
```

```

( SIMULATION D'UNE BOUCLE A VERROUILLAGE DE PHASE NUMERIQUE )
( -- JUIN 1990 -- )

program simulation;
uses graph,crt,sec;
label debut,debut1;
type vecteur=array[0..2009]of integer;
   stri=string[33];
   tableau=array[0..9] of integer;
var giga,ps,t,result,xcorrige:vecteur;
   kchoix,mcarr,mcarr1:char;
   kl,i,j,a,ymax,xmax,nbiais,cor1,zmorte,position,large,haut,numero:integer;
   testgig,testdist,test:shortint;
   choix,corr:word;
   debit,gi3,xal,xrep,moy,ecart,dis,moyenne,gi9:real;
   testsource,defil,testbiais,encore,tesgaus,bin,tdist:boolean;
   seq:tableau;
   coor=array[0..256] of integer;

(*****)
procedure modegraphique; (initialisation du mode graphique)
var graphdriver:integer;
   graphmode:integer;
   errorcode:word;
begin
   graphdriver:=detect;
   initgraph(graphdriver,graphmode,'');
   errorcode:=graphresult;
   if errorcode<>gok then
   begin
      writeln('erreur graphique, appuyer sur <ENTER> ');
      readln;
      halt(1);
   end;
   setviewport(0,0,640,getmaxy,ClipOff);(dimensions EGA, VGA 640 x 348)
end;

(*****)
procedure bordure;( pour suivre évolution offset (sans distorsion))
begin
   setlinestyle(0,0,3);
   rectangle(29,10,291,340);
   setlinestyle(1,0,1);
   line(128,10,128,340);
   line(192,10,192,340);
end;

(*****)
procedure tracecadre;(pour suivre évolution du critère limite ± 128)
begin
   setlinestyle(0,0,3);
   rectangle(330,10,630,95); {tracé pour évolution distorsion initiale}
   rectangle(330,111,630,200);{tracé pour évolution distorsion corrigée}
   setlinestyle(3,0,1);      {limites tolérables}
   line(512,10,512,200);
   line(448,10,448,200);
end;

```



```

case numero of
1:outtextxy(230,210,'correction binaire de pas=1/'+convert(corr));
2:outtextxy(230,210,'correction en escalier');
3:outtextxy(230,210,'definie par l''utilisateur');
end;
outtextxy(124,300,'appuyez sur la barre d'espace pour revenir');
repeat
    myr:=readkey;
until myr=' '; (retour page d'écran précédente)
setactivepage(0);
setvisualpage(0);
end;

(****************************************************************************)
procedure choixsource(var s:vecteur);( choix entre aléatoire ou non )
label ad4;
var ch:char;

procedure transition(so:vecteur;var tr:vecteur);(génère tableau transition)
begin
    for i:=0 to 2000 do
        if so[i+1]=so[i] then tr[i]:=0 (pas de transition)
            else tr[i]:=1; (transition 1 à 0 ou 0 à 1 )
    end;
end;

(génère une séquence pseudo aléatoire de période 511)
procedure pseudo(var x:vecteur);
label ad5;
var e,f:tableau;
    xx:vecteur;
begin
    if encore then goto ad5;
    bin:=false;
    clrscr;
    gotoxy(12,10);
    write(' EMISSION D''UNE SOURCE PSEUDO-ALEATOIRE ');
ad5: (génération)
    for i:=0 to 9 do e[i]:=1;
    j:=0;
    repeat
        if e[4]=e[9] then e[0]:=0 else e[0]:=1;
        for i:=0 to 8 do f[i+1]:=e[i];
        for i:=1 to 9 do e[i]:=f[i];
        xx[j]:=e[9];
        j:=j+1;
    until j=2001; ( génération valeurs pseudo aléatoires)
    transition(xx,x);
end;

```



```

(génère une suite de 0 et de 1 )
procedure source(var x:vecteur);
label ad6;
var l:word;
    xx:vecteur;

procedure sequence(choix:word;var w:tableau);(génère la séquence de la période)
var kch:char;
begin
    clrscr;
    gotoxy(18,8);
    writeln('EMISSION D''UNE SOURCE BINAIRE');
    writeln;
    i:=0;
    repeat
        write('          entrer l''élément ',i+1,' de la séquence ');
        repeat
            kch:=readkey;
            until kch in ['0','1'];
        val(kch,w[i],k1);
        write(w[i]);
        writeln;
        i:=i+1;
    until i=choix;
    delay(300);
end;

begin (génération des 0 et des 1)
    if encore then goto ad6;
    bin:=true;
    clrscr;
    gotoxy(13,8);
    writeln('EMISSION D''UNE SOURCE BINAIRE PERIODIQUE');
    gotoxy(8,11);
    write('choisir la longueur de la période - entre 2 et 9 - ');
    repeat
        kchoix:=readkey;
    until kchoix in ['2'..'9'];
    val(kchoix,choix,k1);
    if choix <>2 then sequence(choix,seq) else
        begin seq[0]:=0;seq[1]:=1;end;
ad6: (génération)
    i:=0;
    repeat
        j:=0;
        repeat
            xx[i]:=seq[j];
            i:=i+1;
            j:=j+1;
        until j=choix;
    until i>2000;

    transition(xx,x);
end;(fin génération 0 et 1)

```

```

begin (corps procedure choix de la source)
  if encore then goto ad4;
  clrscr;
  testsource:=false;
  highvideo;
  gotoxy(12,7);writeln(' CHOIX DE LA SOURCE');
  gotoxy(11,11);write(' P');
  gotoxy(11,12);write(' B');
  gotoxy(11,14);write(' M');
  normvideo;
  gotoxy(11,9);writeln(' appuyer sur ');
  gotoxy(15,11);
  writeln('pour engendrer un signal Pseudo-aléatoire binaire');
  gotoxy(15,12);
  writeln('pour engendrer un signal Binaire périodique');
  gotoxy(15,14);writeln('pour revenir au Menu précédent');
  write;
  gotoxy(12,15);
  repeat
    ch:=readkey;ch:=upcase(ch);
  until ch in ['P','B','M'];
  case ch of
    'P':pseudo(s);
    'B':source(s);
    'M':exit;
  end;
  clrscr;
  gotoxy(15,9);writeln('Vous avez choisi et activé une source');
  if ch='P' then
  begin
    gotoxy(15,10);writeln(' binaire pseudo-aléatoire');
    testsource:=true;
  end;
  if ch='B' then
  begin
    gotoxy(22,10);writeln('binaire périodique');
    testsource:=true;
  end;
  gotoxy(20,14);
  write('appuyez sur une touche ');
  repeat until keypressed;
  exit;
ad4: (si encore)
  if bin=true then source(s);
  if bin=false then pseudo(s);
end;  {fin procedure choix et construction de la source}

```

```

(#####)
procedure entree(phrase:stri;var cha:char);
begin
  repeat
    clrscr;
    gotoxy(10,10);writeln('Souhaitez vous ',phrase);
    gotoxy(9,11);
    write('tapez      pour oui      et      pour non');
    highvideo;
    gotoxy(18,11);
    write('0');
    gotoxy(40,11);
    write('N');
    normvideo;
    gotoxy(52,11);
    cha:=readkey;cha:=upcase(cha);
  until cha in ['0','N'];
end;

```

```

(#####)
procedure gigue;
var chg:char;
    lin:stri;
begin
  lin:='entrer un offset?';
  testgig:=0;
  entree(lin,chg);
  if chg='0' then
  begin
    testgig:=2;
    repeat
      clrscr;
      gotoxy(10,9);
      writeln('Entrer la valeur du      d''émission en hertz');
      highvideo;
      gotoxy(30,9);write('débit');
      normvideo;
      gotoxy(10,10);
      writeln('(inférieure à 10 000 Hz) et appuyer sur <ENTER>');
      gotoxy(22,11);writeln('Hz');
      gotoxy(11,11);readln(debit);
      if debit>=10000 then
      begin
        gotoxy(10,12);
        writeln('débit trop grand !');
        delay(900);
      end;
    until debit<10000;
  end;
end;

```



```

repeat
  clrscr;
  gotoxy(10,9);
  write('Entrer la valeur de l''      en hertz');
  highvideo;
  gotoxy(33,9);write('offset');
  normvideo;
  gotoxy(10,10);
  writeln('(inférieure à 50 Hz) et appuyer sur <ENTER> ');
  gotoxy(22,11);writeln('Hz');
  gotoxy(11,11);readln(gi3);
  if gi3>=50 then
  begin
    gotoxy(10,12);
    writeln('offset trop important !');
    delay(900);
  end;
until gi3<50;
gi9:=gi3;
gi3:=gi3/debit; {transforme l'offset en pourcentage}
gi3:=gi3/2;      {gi3*5/10}
gi3:=gi3*256;{valeur de l'offset réel sans troncature}

  clrscr;
  gotoxy(18,10);writeln('Vous avez introduit un offset');
  gotoxy(26,12);writeln('appuyez sur ');
  gotoxy(20,14);write('n''importe quelle touche ');
  gotoxy(44,14);
  repeat
    chg:=readkey;
  until chg<>' ';
end
else gi9:=0;
end;

(%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%)
procedure visoff(var gi:vecteur);
var coof:char;
    lin:stri;
    gi2,gi6:real;
    corgig,gil:integer;

begin

  lin:='corriger l''offset?';
  entree(lin,coof);
  case coof of
  'N':begin {offset non corrigé}
    gi[0]:=0;
    gi6:=0;
    for i:=1 to 1999 do
    begin
      gi6:=gi6+gi3;
      if gi6>128 then gi6:=gi6-256;
      if gi6<-128 then gi6:=gi6+256;
      gi1:=trunc(gi6);
      gi[i]:=gi1;
    end;
  end;
end;

```



```

'0':begin (offset corrigé)
  repeat
    clrscr;
    gotoxy(8,8);
    writeln('L'offset est de ',gi3:7,' unités d''écran');
    gotoxy(3,10);
    writeln('Choix du pas de correction 1/K (K=256 représente la correction unité)');
    gotoxy(10,11);
    writeln('taper K choisi : 256 , 128 , 64 , 32 , 16');
    gotoxy(11,12);
    write(' et appuyer sur <ENTER>');
    gotoxy(40,12);
    readln(corgig);
    case corgig of
      256:corgig:=1;
      128:corgig:=2;
      64:corgig:=4;
      32:corgig:=8;
      16:corgig:=16;
    end;
  until corgig in [1,2,4,8,16];
  gi[0]:=0;
  gi6:=0;
  for i:=1 to 1999 do
  begin
    gi6:=gi6+gi3;
    if gi6>128 then gi6:=gi6-256;
    if gi6<-128 then gi6:=gi6+256;
    gi2:=gi6;
    if gi2>0 then
    begin
      gi6:=gi6-corgig;
      if gi6<-128 then gi6:=gi6+256;
    end;
    if gi2<0 then
    begin
      gi6:=gi6+corgig;
      if gi6>128 then gi6:=gi6-256;
    end;
    gi1:=trunc(gi6);
    gi[i]:=gi1;
  end;
end;
end;
end;
end;

```

```

(#####)
procedure visuoffset(off:vecteur);(visualisation de l'offset seul)
var yoff,ii,xoff:integer;
    cont:char;
const lettre=#24;

begin
ii:=0;
repeat
    bordure;
    settextstyle(0,0,1);
    outtextxy(400,278,' APPUYER SUR');
    outtextxy(340,288,'- la barre d'espacement - pour continuer');
    outtextxy(340,318,'- T - pour terminer et sortir');
    if defil then
    begin
        settextstyle(0,0,1);
        outtextxy(340,298,'- N - pour defilement vitesse normale');
    end;
    yoff:=10;
    for i:=ii to ii+39 do
    begin
        xoff:=off[i]+160;
        yoff:=yoff+8;
        moveto(xoff,yoff);
        if defil then delay(400);
        outtext(lettre);
    end;
    repeat
        cont:=readkey;cont:=upcase(cont);
    until cont in [' ','T','N'];
    if cont='N' then defil:=false;
    if cont=' ' then
    begin
        ii:=ii+40;
        clearviewport;
    end;
    if ii>1990 then cont:='T';
until cont='T';
end;

(#####)

procedure choixdistorsion(var y:vecteur);(choix et génération distorsion)
var ch:char;
    ligne:stri;

function norm(var no:real):integer;
begin
    norm:=trunc(no#256);
end;

```

```

{*****}
  (prg générant des nbres aléatoires suivant une gaussienne)
  procedure gauss(var x:vecteur);
  label ad1;
  var sum,g:real;
  const pi1=3.141592;
        pi2=0.159154;
        pi3=-6.283185;

  begin
    if encore=true then goto ad1;
    tesgaus:=true;
    randomize;
    clrscr;
    gotoxy(25,10);
    writeln('DISTORSION GAUSSIENNE');
    repeat
      gotoxy(59,13);
      writeln(' ');
      gotoxy(17,13);
      write('Entrer la moyenne - inférieure à 2 - ');
      gotoxy(17,14);
      write('et appuyer sur <ENTER>');
      gotoxy(60,13);
      readln(moyenne);
    until moyenne < 2 ;
    repeat
      gotoxy(59,16);
      writeln(' ');
      gotoxy(17,16);
      write('Donner la variance - supérieure à 0.3 - ');
      gotoxy(17,17);
      write('et appuyer sur <ENTER>');
      gotoxy(60,16);
      readln(ecart);
    until ecart >0.3;
    ecart:=sqrt(ecart);
    gotoxy(17,22);
    write;
  ad1: {génération}
    for i:=0 to 2000 do
      begin
        sum:=0;
        for j:=0 to 11 do sum:=sum+random;
        g:=(sum-6)*ecart+moyenne;

        repeat
          if g>pi1 then g:=g+pi3; {glissement de la base de temps}
          if g<-pi1 then g:=g-pi3;
          until (g<=pi1) and (g>=-pi1);

          g:=g*pi2; {normalisation pour avoir résultat entre ± 0.5}
          x[i]:=norm(g);
        end;
      end;
  end;

```

```

(*****)
procedure uniforme(var x:vecteur); (génération selon loi uniforme)
label ad2;
var maxi:real;
const pi1=3.1415926;
begin
  if encore=true then goto ad2;
  tesgaus:=false;
  randomize;
  repeat
    clrscr;
    gotoxy(20,7);
    writeln('DISTORSION UNIFORME');
    gotoxy(17,11);
    writeln('Donner la valeur maximale ');
    gotoxy(16,13);
    writeln(' - inférieure à 128 - ');
    gotoxy(18,15);
    writeln(' et appuyer sur <ENTER>');
    gotoxy(45,15);
    readln(maxi);
  until (maxi)>=2) and (maxi<=128);
  gotoxy(17,23);
  a:=trunc(maxi);
ad2: (génération)
  for i:=0 to 2000 do
  begin
    x[i]:=random(a#2-1)-trunc(a);
  end;

end;

(*****)
procedure biaisee(ys:vecteur;var x:vecteur); (distribution biaisée)
label ad3;
var biais:real;
begin
  if encore=true then goto ad3;
  testbiais:=true;
  repeat
    clrscr;
    gotoxy(26,7);
    writeln('DISTORSION BIAISEE');
    gotoxy(17,11);
    writeln(' Donner la distorsion en pourcentage ');
    gotoxy(26,13);
    writeln(' inférieure à 50% ');
    gotoxy(38,15);writeln('%');
    gotoxy(30,15);readln(biais);
  until biais <50;
  gotoxy(17,23);
  biais:=(biais/1000)#5;
  nbiais:=-norm(biais);

```



```

ad3: {remplissage}
  x[0]:=0;
  for i:=1 to 2000 do
  begin
    if not tdist then ys[i]:=0;
    if t[i-1]=0 then x[i]:=ys[i]
      else begin
        nbiais:=-nbiais;
        x[i]:=ys[i]+nbiais;
        if x[i]>128 then x[i]:=x[i]-256;
        if x[i]<-128 then x[i]:=x[i]+256;
      end;
    end;
  end;
end;

begin {corps procedure choix de la distorsion }
if not encore then testbiais:=false;
if encore=true then
begin
  if tdist then
  begin
    if tesgaus=true then gauss(y);
    if tesgaus=false then uniforme(y);
  end;
  if testbiais=true then biaisee(y,y);
  exit;
end;
repeat
  ligne:='introduire une distorsion?';
  entree(ligne,ch);
  if ch='N' then
  begin
    testdist:=0;
    ch:='M';
  end;
  if ch='O' then
  begin
    repeat
      clrscr;
      highvideo;
      gotoxy(10,7);writeln(' CHOIX DE LA DISTORSION');
      gotoxy(11,10);write('G');
      gotoxy(11,11);write('U');
      gotoxy(11,12);write('B');
      gotoxy(11,15);write('M');
      normvideo;
      gotoxy(14,9);writeln(' appuyer sur ');
      gotoxy(14,10);writeln('pour distribution Gaussienne');
      gotoxy(14,11);writeln('pour distribution Uniforme');
      gotoxy(14,12);writeln('pour distribution Biaisée');
      gotoxy(14,15);writeln('pour revenir MENU principal');
      writeln;
      gotoxy(11,16);
    repeat
      ch:=readkey;ch:=upcase(ch);
    until ch in ['G','U','B','M'];
  end;
end;

```

```

case ch of
  'G':begin
    testdist:=1;
    tdist:=true;
    gauss(y);
  end;
  'U':begin
    testdist:=1;
    tdist:=true;
    uniforme(y);
  end;
  'B':if tdist=true then
    begin
      ligne:='ajouter une distorsion biaisée?';
      entree(ligne,ch);
      if ch='O' then biaisee(y,y);
    end
  else begin
    biaisee(y,y);
    testdist:=1;
  end;
end;
until (ch='M')or(ch='O');
end;
if ch='M' then exit;
until ch('>')'M';
end;

```

```

(%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%)
procedure correction1(trans,x,offset:vecteur;var xcor:vecteur);(avec ou sans zone morte)
label ad7;
var vall,offnul,xnul,gi1:integer;
    err1,gi0,gi2:real;
    cha:char;
begin
  if encore then goto ad7;
  repeat
    clrscr;
    gotoxy(15,9);
    writeln('Correction avec zone morte?');
    gotoxy(15,10);
    writeln('Tapez      pour oui      et      pour non?');
    gotoxy(15,11);
    highvideo;
    gotoxy(22,10);write('O');
    gotoxy(47,10);write('N');
    normvideo;
    gotoxy(60,10);
    cha:=readkey;cha:=upcase(cha);
  until cha in ['O','N'];

```

```

if cha='0' then
begin
  repeat
    clrscr;
    gotoxy(17,9);
    writeln('Choix de la largeur de la zone morte ');
    gotoxy(23,10);
    writeln(' ±32   ±16   ±8 ');
    gotoxy(19,12);
    writeln('Entrer      32   16   ou   8 ');
    gotoxy(20,13);
    write(' et appuyer sur <ENTER>');
    gotoxy(46,13);
    readln(zmorte);
  until zmorte in [32,16,8];
end
else zmorte:=0;
clrscr;
gotoxy(18,10);
write('Choix du pas de correction 1/');
highvideo;
write('K');
normvideo;
repeat
  gotoxy(12,12);
  writeln('Tapez K choisi : 256 , 128 , 64 , 32 ou 16');
  gotoxy(13,13);
  write(' et appuyez sur <ENTER> ');
  gotoxy(42,13);
  readln(cor1);
  corr:=cor1;
  case cor1 of
    256:cor1:=1;
    128:cor1:=2;
    64:cor1:=4;
    32:cor1:=8;
    16:cor1:=16;
  end;
until cor1 in [1,2,4,8,16];

ad7: (correction1)

if testgig=0 then offset[i]:=0; ( il n'y a pas d'offset)
if testdist=0 then x[i]:=0; (il n'y a pas de distorsion)

if not encore then (si encore, pas d'initialisation de déphasage)
begin
  gi2:=-128;
  position:=-128;
  xcor[0]:=-128;
end;
for i:=1 to 1999 do
if trans[i]=1 then

```





```

        writeln('Choix de la largeur de la zone morte ');
        gotoxy(23,10);
        writeln(' ±32   ±16   ±8 ');
        gotoxy(19,12);
        writeln('Entrer      32   16   ou   8 ');
        gotoxy(20,13);
        write(' et appuyer sur <ENTER>');
        gotoxy(46,13);
        readln(zmorte);
    until zmorte in [32,16,8];
end
else zmorte:=0;    (zone morte définie)

repeat
repeat (choix de la largeur du pas de correction)
    clrscr;
    gotoxy(3,8);
    writeln('Choix de la largeur du pas de correction');
    gotoxy(3,9);
    writeln('la largeur du pas doit être supérieure à sa hauteur !');
    gotoxy(3,10);
    write('1/');
    highvideo;
    write('K');
    normvideo;
    gotoxy(12,11);
    writeln('Taper K choisi : 256 , 128 , 64 , 32 , 16 , 8');
    gotoxy(5,12);
    write(' et appuyer sur <ENTER> ');
    readln(large);
    case large of
        256:large:=1;
        128:large:=2;
        64:large:=4;
        32:large:=8;
        16:large:=16;
        8:large:=32;
    end;
until large in [1,2,4,8,16,32];(fin choix largeur pas)

repeat (choix de la hauteur du pas de correction)
    clrscr;
    gotoxy(3,8);
    writeln('Choix de la hauteur du pas de correction (hauteur d''une marche)');
    gotoxy(3,9);
    writeln(' ( la hauteur doit être inférieure à la largeur )');
    gotoxy(3,10);
    write('1/');
    highvideo;
    write('K');
    normvideo;
    gotoxy(12,11);
    writeln('Taper K choisi : 256 , 128 , 64 , 32 , 16');
    gotoxy(5,12);
    write(' et appuyer sur <ENTER> ');
    readln(haut);

```

```

    case haut of
      256:haut:=1;
      128:haut:=2;
      64:haut:=4;
      32:haut:=8;
      16:haut:=16;
    end;
    until haut in [1,2,4,8,16]; {fin du choix de la hauteur d'une marche}
  until large>haut;

  position:=-128;
  xcor[0]:=-128;
ad8: {correction2}
  for i:=1 to 1999 do
    if trans[i]=1 then
      begin
        if testgi3=0 then gi3:=0 ;
        if testdist=0 then xnul:=0 else xnul:=x[i];
        gi6:=position+gi3-xnul;
        val2:=trunc(gi6);{écart ppr B T}
        if val2>z morte then
          begin
            inter:=(val2-z morte)div(large);
            cor2:=inter#haut;
          end;
        if val2<-z morte then
          begin
            inter:=(val2+z morte)div(large);
            cor2:=inter#haut;
          end;
        position:=position-cor2;
        if position>128 then position:=position-256;
        if position<-128 then position:=position+256;
        xcor[i]:=position;
      end
    else
      begin
        gi2:=xcor[i-1]+gi3;
        if gi2>128 then gi2:=gi2-256;
        if gi2<-128 then gi2:=gi2+256;
        position:=trunc(gi2);
        xcor[i]:=position;
      end;
    end;
  end;
end;

```

```

{*****}
{procedure construite pas à pas}
procedure pasapas(trans,x,offset:vecteur;var xcor:vecteur);
label ad9;
var offnul,xnul,val3,cor3:integer;
    ordo,h,g,coord:integer;
    absi:word;
    gi6,gi2:real;

```

```

begin
  if encore then goto ad9;
  clrscr;
  gotoxy(7,8);
  writeln('Construction pas à pas de la caractéristique de correction');
  gotoxy(9,14);
  write('Nombre de segments en abscisses :');
  gotoxy(9,15);
  write('(Il peut être 2 , 4 , 8 , 16 , 32 , 64 , 128 ou 256)');
  repeat
    gotoxy(50,14);
    write(' ');
    gotoxy(50,14);
    readln(absi);
  until (absi in [1,2,4,8,16,32,64,128]) or (absi=256);
  j:=trunc(256/absi);
  g:=0;
  i:=0;
  clrscr;
  gotoxy(28,9);
  writeln('CORRECTION');
  gotoxy(7,10);
  writeln('La correction peut être de 0 ou ± 1/K ');
  gotoxy(7,11);
  writeln(' avec K = 256, 128, 64, 32 ou 16');
  writeln;
  repeat
    repeat
      write(' Entrez pour le segment ',g+1,' la correction: ');
      readln(coord);
      case coord of
        256 :coord:=1;
        -256:coord:=-1;
        128 :coord:=2;
        -128:coord:=-2;
        64 :coord:=4;
        -64 :coord:=-4;
        32 :coord:=8;
        -32 :coord:=-8;
        16 :coord:=16;
        -16 :coord:=-16;
      end;
      until abs(coord) in [0,1,2,4,8,16,32];
      for h:=i to i+j do coor[h]:=coord;
      i:=i+j;
      g:=g+1;
    until g=absi;

    position:=-128;
    xcor[0]:=-128;

  ad9:(correction 3)
  for i:=1 to 1999 do
    if trans[i]=1 then

```



```

begin
  if testgig=0 then gi3:=0 ;
  if testdist=0 then xnul:=0 else xnul:=x[i];
  gi6:=position+gi3-xnul;
  val3:=trunc(gi6);
  if val3>128 then val3:=128;
  if val3<-128 then val3:=-128;
  val3:=val3+128;
  cor3:=coor[val3];
  position:=position-cor3;
  if position>128 then position:=position-256;
  if position<-128 then position:=position+256;
  xcor[i]:=position;
end
else
begin
  gi2:=xcor[i-1]+gi3;
  if gi2>128 then gi2:=gi2-256;
  if gi2<-128 then gi2:=gi2+256;
  position:=trunc(gi2);
  xcor[i]:=xcor[i-1];
end;
end;

(#####)
procedure pr3(z,er,xcor:vecteur);(place les critères suivant un axe vertical)
label ad10;
const lettre=#24;
var cont:char;
  ii,rang1,rang2,top,mil:integer;
  rep1,rep2:array[0..255]of integer;
  icar:char;
  maxi:boolean;

procedure poursuite;
begin
  encore:=true;
  choixsource(t);
  choixdistorsion(ps);
  if numero=1 then correction1(t,ps,giga,xcorrige);
  if numero=2 then correction2(t,ps,giga,xcorrige);
  if numero=3 then pasapas(t,ps,giga,xcorrige);
end;

procedure reduction;
var redu:integer;
begin
  for redu:=0 to 255 do
    rep2[redu]:=trunc(rep2[redu]*80/top);
end;

```



```

procedure minim;
var cc:integer;
begin
  for cc:=0 to 255 do
    if repl[cc]<>0 then dec(repl[cc],1);
end;

procedure etat1;
begin
  settextstyle(0,1,1);
  outtextxy(8,90,#24+' position apres correction');
  settextstyle(0,0,1);
  outtextxy(400,278,' APPUYER SUR');
  outtextxy(340,288,'- I - pour informations');
  outtextxy(340,298,'- la barre d'espace - pour continuer');
  outtextxy(340,308,'- F - pour avoir distribution finale');
  outtextxy(388,318,'apres 2000 periodes ');
  outtextxy(340,328,'- T - pour terminer et sortir');
end;

procedure etat4;
begin
  outtextxy(356,99,'distribution criteres distordus');
  outtextxy(342,210,'distribution base de temps corrige');
end;

procedure etat2;
begin
  settextstyle(0,0,1);
  outtextxy(342,250,'periodes d''horloge ecoulees :'+convert(ii+39));
end;

procedure etat3;
begin
  settextstyle(0,0,1);
  outtextxy(340,338,'- N - pour defilement vitesse normale');
end;

begin
  top:=0;
  for i:=0 to 255 do begin repl[i]:=0;rep2[i]:=0;end;
  {initialisation répartitions initiales et finales}

  ii:=0;
  repeat
    bordure;
    tracecadre;
    setlinestyle(0,0,1);
    for j:=0 to 255 do
      begin
        line(j+352,90,j+352,90-repl[j]);
        line(j+352,195,j+352,195-rep2[j]);
      end;
  until top=255;
end;

```

```

ymax:=10;
etat1;
etat4;
for i:=ii to ii+39 do
begin
  if defil then
  begin
    etat3;
    delay(600);
  end;

  xmax:=er[i]+160;
  ymax:=ymax+8;
  if t[i]=1 then
  begin
    moveto(xmax,ymax);
    setlinestyle(0,0,1);
    linerel(0,5);
  end;

  xmax:=xcor[i]+160;
  moveto(xmax,ymax);
  outtext(lettre);

  {affichage répartition}
  setlinestyle(0,0,1);
  if t[i]=1 then
  begin
    rang1:=z[i]+128;
    inc(rep1[rang1],2);
  end;

  rang2:=xcor[i]+128;
  inc(rep2[rang2],1);
  if rep2[rang2]>top then top:=rep2[rang2];

  if rep1[rang1]<78 then
    line(rang1+352,90,rang1+352,90-rep1[rang1])
  else rep1[rang1]:=78;

  if rep2[rang2]>80 then reduction;
  {fin répartition}

end;
etat2;
for j:=0 to 255 do
line(j+352,195,j+352,195-rep2[j]);

repeat
  repeat
    cont:=readkey;cont:=upcase(cont);
    until cont in [' ','T','F','N','I'];
    if cont='I' then info;
  until cont<>'I';
  if cont='N' then begin cont:=' ';defil:=false;end;

```

```

case cont of
' ': begin
    ii:=ii+40;
    clearviewport;
    end;
'F':begin {distribution finale après 2000 périodes d'horloge}
    mil:=2000;
ad10:    maxi:=false;
    for i:=1 to 1999 do
    begin
        if t[i]=1 then
        begin
            rang1:=z[i]+128;
            inc(rep1[rang1],1);
            if rep1[rang1]>=78 then rep1[rang1]:=78;
        end;
        if rep1[rang1]>=78 then maxi:=true;

        rang2:=xcor[i]+128;
        inc(rep2[rang2],1);
        if rep2[rang2]>top then top:=rep2[rang2];

    end;
    reduction;
    clearviewport;
    tracecadre;
    setlinestyle(0,0,1);
    etat4;
    for j:=0 to 255 do
    begin
        line(j+352,90,j+352,90-rep1[j]);
        rep2[j]:=abs(rep2[j]);
        line(j+352,195,j+352,195-rep2[j]);
    end;
    outtextxy(350,250,' après '+convert(mil)+' périodes d''horloge');
    outtextxy(400,278,' APPUYER SUR');
    outtextxy(340,288,'- P - pour avoir la distribution');
    outtextxy(340,298,'    après 2000 autres périodes');
    outtextxy(340,308,'- T - pour terminer');
    repeat
        cont:=readkey;cont:=upcase(cont);
    until cont in['P','T'];
    mil:=mil+2000;
    if mil>=20000 then cont:='T';
    if cont='P' then
    begin
        outtextxy(340,320,' -- calcul en cours --');
        poursuite;
    end;
    if cont='P' then goto ad10;
    end;
    end;
    until cont='T';
    encore:=false;
end;

```



```

(%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%)
procedure menu(var carac:char);
var carac2:char;
begin
  repeat
    clrscr;
    highvideo;
    gotoxy(20,3);writeln('MENU PRINCIPAL');
    gotoxy(18,7);writeln('S');
    gotoxy(18,9);writeln('O');
    gotoxy(18,11);writeln('D');
    gotoxy(18,13);writeln('C');
    gotoxy(18,15);writeln('V');
    gotoxy(18,17);writeln('N');
    gotoxy(18,19);writeln('A');
    gotoxy(18,21);writeln('Q');
    normvideo;
    gotoxy(20,7);writeln('- source');
    gotoxy(20,9);writeln('- offset');
    gotoxy(20,11);writeln('- distorsion');
    gotoxy(20,13);writeln('- correction');
    gotoxy(20,15);writeln('- visualiser');
    gotoxy(20,17);writeln('- nouveau choix');
    gotoxy(20,19);writeln('- aide');
    gotoxy(20,21);writeln('- quitter');

    gotoxy(8,24);writeln('Tapez la lettre correspondant à votre choix');
    gotoxy(53,24);
    repeat
      carac:=readkey;
      carac:=upcase(carac);
    until carac in ['S','O','D','C','V','Q','N','A'];
    if carac='N' then carac:='W';
    if carac='Q' then
      begin
        clrscr;
        gotoxy(6,15);
        write('Etes vous sûr de vouloir quitter le programme? O/N ');
        gotoxy(6,70);
        carac2:=readkey;
        carac2:=upcase(carac2);
        writeln;
        if carac2='O' then begin clrscr;halt;end
          else carac:='M';
      end;
    until carac<>'M';
  end;
end;

(programme principal)
begin
  clrscr;
  gotoxy(9,10);
  highvideo;
  writeln('SIMULATION D'UNE BOUCLE A VEROUILLAGE DE PHASE NUMERIQUE');
  gotoxy(67,10);
  repeat until keypressed;
  normvideo;
  clrscr;

```



```

debut1:
    encore:=false;
    testsource:=false;
    testgig:=0;
    testdist:=0;
    testbiais:=false;
    tdist:=false;
    gi9:=0;
debut:
    clrscr;
    menu(mcar);
    case mcar of
        'S': choixsource(t);
        'O': if testsource then gigue;
        'D': if testsource then choixdistorsion(ps);
        'C': if testsource then
            begin
                test:=testgig+testdist;
                if test=0 then goto debut;(aucune distorsion)
                if test=2 then goto debut;(il n'y a que l'offset)
                {correction si distorsion ou distorsion+offset}
                {test=1 ou test=3}

                clrscr;
                highvideo;
                gotoxy(8,8);
                write('CHOIX DE LA CORRECTION');
                normvideo;
                gotoxy(5,10);
                write(' correction 1 : caractéristique binaire');
                gotoxy(5,11);
                write(' correction 2 : caractéristique en escalier');
                gotoxy(5,12);
                write(' correction 3 : construction de la caractéristique');
                gotoxy(6,14);
                write('choisir le numéro de correction 1, 2 ou 3 puis <ENTER> ');
                repeat
                    kchoix:=readkey;
                    kchoix:=upcase(kchoix);
                until kchoix in['1','2','3'];
                val(kchoix,numero,k1);
                write(numero);
                case numero of
                    1:correction1(t,ps,giga,xcorrige);
                    2:correction2(t,ps,giga,xcorrige);
                    3:pasapas(t,ps,giga,xcorrige);
                end;
            end;
        'V': begin
                test:=testgig+testdist;
                if test=0 then goto debut;
                if test=2 then visoff(giga);
                clrscr;
                gotoxy(10,10);
                write('défilement lent ? (O/N) ');
            end;
    end;

```

```
repeat
    mcar1:=readkey;
    mcar1:=upcase(mcar1);
until mcar1 in['O','N'];
write(mcar1);
if mcar1='O' then defil:=true else defil:=false;

modegraphique;
bordure;
if test=2 then visuoffset(giga);
if test<>2 then

begin
    tracecadre;
    pr3(ps,ps,xcorrige);
    repeat until keypressed;
end;
restorecrtmode;

end;
'A': intro;
'W': goto debut1;
end;
goto debut;

end.

(      fin du texte du programme de simulation      )
```

