

***APPRENEZ
L'ELECTRONIQUE
SUR ORIC ET ATMOS***

«La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part que «les copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective» et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants-droit ou ayants-cause, est illicite» (alinéa premier de l'article 40). Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 du Code Pénal.»

Pierre BEAUFILS

***APPRENEZ
L'ELECTRONIQUE
SUR ORIC ET ATMOS***

DIFFUSION:

SORACOM INFORMATIQUE

16 A, av . Gros Malhon - 35 000 RENNES
Tél : (99) 54. 22. 30. lignes groupées .

OUVRAGES PARUS AUX ÉDITIONS SORACOM

La Guerre des Ondes

de F. Mellet et S. Faurez

Alimentations de puissance

Collection Sélection de montages

Transat Terre-Lune

*Union pour la Promotion de la
Propulsion Photonique*

QSO en radiotéléphonie (français-anglais)

de L. Sigrand

Interférences TV (QRM TV)

2ème édition

de F. Mellet et K. Pierrat

A l'écoute des radiotélétypes

2ème édition

de J.L. Fis

Technique radio pour l'amateur

3ème édition

de F. Mellet et S. Faurez

Communiquez avec votre ZX81

2ème édition

de D. Bonomo et E. Dutertre

Télévisions du monde

de P. Godou

Le radioamateur et la QSL

de G. Lelarge

Technique de la B.L.U.

2ème édition

de G. Ricaud

La réception des satellites météo

de L. Kuhlmann

Les synthétiseurs de fréquences

de M. Levrel

Expédition

Pôle Nord Magnétique 1983

de M. Uguen

Visa pour ORIC

de F. Blanc et F. Normant

Forth pour ORIC

d'Oric France

Programmes pour votre ORIC

de E. Jacob et J. Portelli

Communiquez avec votre ORIC

de D. Bonomo et E. Dutertre

Concevoir un émetteur expérimental

de P. Loglisci

Apprendre l'électronique sur ORIC

de P. Beaufils

Introduction

Ce livre est un recueil de programmes relatifs aux fonctions et aux composants de l'électronique. Ces programmes sont rédigés en BASIC MICROSOFT et n'utilisent que les instructions les plus classiques du BASIC, de manière à les rendre « universels ». Ils ne font pas appel aux propriétés particulières du microordinateur employé pour les mettre au point (ORIC 1 en la circonstance), sauf pour les instructions graphiques dont nous reparlerons plus loin. Ils sont écrits sans aucune subtilité, de manière à être très clairs et sont évolutifs, c'est-à-dire qu'il est possible de les modifier facilement pour leur faire résoudre des problèmes similaires.

Ces programmes sont de trois types : ce pourront être, soit des idées d'algorithmes permettant de résoudre un certain type de problèmes, soit des « libres-services » apportant une solution générale, soit enfin des programmes résolvant une situation particulière donnée et que le lecteur pourra adapter à ses besoins en modifiant éventuellement pour cela l'algorithme de calcul, les valeurs numériques ou les approximations faites.

Quelques programmes ont un aspect pédagogique certain. Ils ne remplacent pas l'expérience de laboratoire, mais la complètent, en permettant la recherche du meilleur modèle mathématique possible. Ils ouvrent alors le champ à l'étude de l'influence de tel ou tel paramètre et offrent la possibilité d'étudier des grandeurs inaccessibles à l'expérience.

Nous faisons un grand usage de l'imprimante. Il s'agit là d'un outil indispensable pour tout travail scientifique sérieux. Notre modèle est une SEIKOSHA GP100. Il s'agit d'un « bas de gamme-premier prix » (sans sous-entendu désobligeant). Mais ses possibilités, comme pourra en juger le lecteur, sont étonnantes. Ce modèle est compatible CENTRONICS, ce qui fait que le dialogue à partir d'un autre microordinateur se fera sans difficulté avec les mêmes instructions.

Nous avons été guidés par l'expérience, c'est-à-dire que, partant de celle-ci, nous avons cherché à mathématiser le phénomène étudié, sans perdre de vue le réel. Cela nous a amenés à éliminer les quantités visiblement négligeables, ou physiquement inconséquentes.

Afin de ne pas alourdir les programmes, certaines améliorations qualitatives ne sont pas incluses dans les programmes. Par exemple, les tracés d'axes sont quelquefois omis. Suivant le microordinateur employé, il sera alors possible de les ajouter et de les graduer.

En général, les variables d'un programme sont incluses dans celui-ci, et n'ont pas à être fournies lors d'INPUT. Cela permet de conserver leur valeur lors d'un listing sur l'imprimante ; cela évite surtout de les perdre à la suite d'une erreur de manipulation... (cas de l'ORIC qui efface tous ses variables lors d'une modification de ligne de programme...).

Les seules instructions graphiques employées et particulières à l'ORIC sont CURSET, DRAW.

Rappelons qu'ORIC a un mode de haute résolution (HIRES), pour lequel l'origine est en haut à gauche. L'axe OY est vertical descendant et gradué de 0 à 199 ; l'axe OX est horizontal et gradué de 0 à 239. CURSET X,Y,1 place un point aux coordonnées X,Y. Draw $\Delta X, \Delta Y$ dessine une droite, à partir du point désigné en CURSET, correspondant à un déplacement $\Delta X, \Delta Y$.

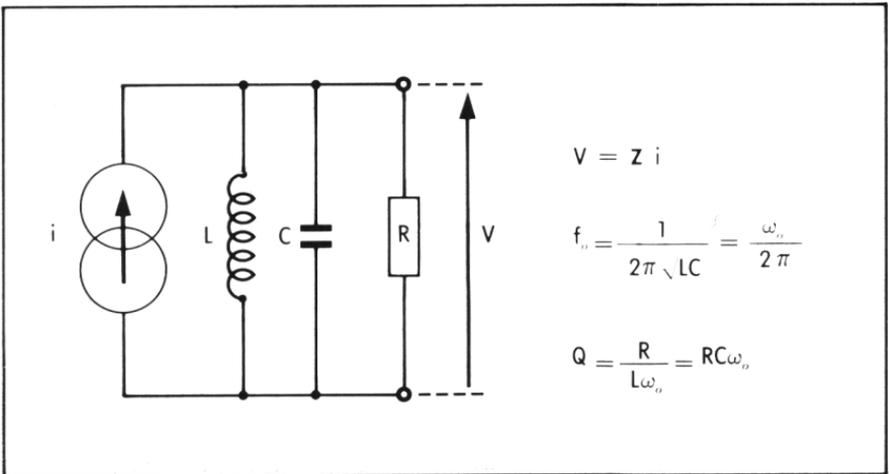
Enfin, nous nous sommes attachés à proposer des programmes courts. Cela se fait quelquefois au détriment de la présentation, mais jamais de la rigueur scientifique. L'expérience montre qu'il est très difficile de recopier un programme de quelques pages (en plus du fait que c'est lassant), de le vérifier et de le modifier. Cependant, ce sont souvent les programmes courts qui surprendront le plus le lecteur...

Chapitre 1

RESONANCE

CIRCUIT RÉSONNANT LC PARALLÈLE

Ce circuit est extrêmement répandu en électronique. Il est à la base de la sélectivité des récepteurs radio. Son étude mathématique est simple. Nous nous intéresserons aux courbes donnant le module et la phase de sa transmittance, ainsi qu'au temps de retard de groupe. Les définitions relatives à ce circuit sont les suivantes :



On a :
$$Z = \frac{R}{1 + jQ(x - 1/x)} \quad \text{avec } x = f/f_0$$

On a aussi :
$$\varphi = - \text{Arc tg } Q(x - 1/x).$$

Nous proposons de tracer les deux courbes relatives à ces grandeurs en fonction de x . Nous normaliserons Z (en le divisant par R). L'échelle sur l'axe des fréquences est linéaire : nous la graduerons de $x = 0,5$ à $1,5$. Il est possible d'étudier plus finement ces fonctions en réduisant cet intervalle. Nous étudierons ensuite le temps de retard de groupe, défini par : $T = d\varphi/d\omega$, qui, normalisé, devient $\omega_0 T = d\varphi/dx$. Ainsi, sa valeur maximum est $2Q$.

Programmes

§ Partie I : Lignes 10 à 130

- * Lignes 40 à 50 : Calcul du module et de la phase de la transmittance.
- * Ligne 70 : Inscription des points correspondants sur l'écran.
- * Lignes 90 à 120 : Dessin des axes.

§ Partie 2 : Lignes 190 à 310

- * Recopie rapide d'écran (voir annexe 2).

§ Partie 3 : Lignes 400 à 540

- * Ligne 470 : Calcul du module de la transmittance.
- * Ligne 480 : Calcul de la dérivée $d\varphi/dx$ autour du point courant d'abscisse x .
- * Lignes 510 à 530 : Axes.

§ Partie 4 : Lignes 700 à 2 080

- * Lignes 705 à 770 : Mise en page de l'impression.
- * Ligne 780 : Préparation de l'imprimante au mode graphique.
- * Lignes 810 à 830 : Intervalle de valeurs choisies pour x et définition de x .
- * Lignes 840 à 860 : Calcul de Z , P , T .
- * Lignes 870-880 : Envoi de Z sur l'imprimante.
- * Lignes 890-900 : Envoi de P sur l'imprimante.
- * Lignes 910-920 : Envoi de T sur l'imprimante.
- * Lignes 930 et 2 000 à 2 080 : Routine d'impression des axes.

Variables

§ Partie I

- * Q : coefficient de qualité du circuit étudié (ici, $Q = 20$).
- * x : variable réduite de la fréquence ; $x = f/f_0$. Dans cet exemple, x varie de 0,5 à 1,5.
- * Z : module de la transmittance.
- * P : phase de la transmittance.
- * U : variable de position horizontale.

§ Partie 2 : voir annexe 2.

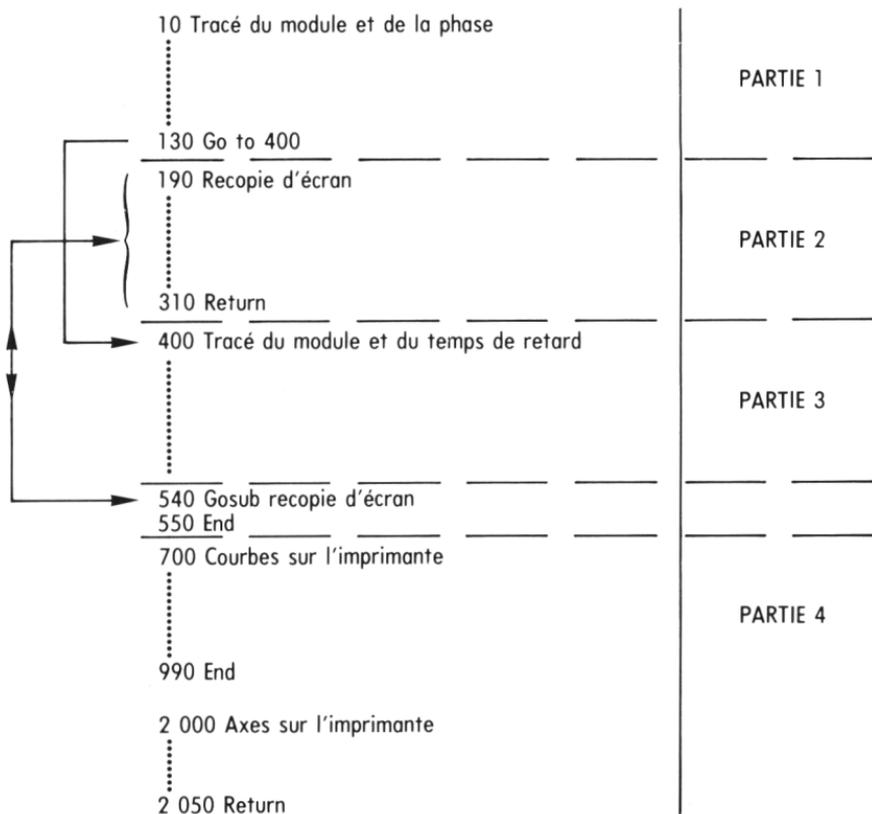
§ Partie 3

- * T : temps de retard de groupe réduit.

§ Partie 4

- * V , N : voir annexe 2.
- * x : variable réduite de la fréquence (ici x varie de 0,8 à 1,2).
- * B : variable de position horizontale sur l'imprimante.

Utilisation du programme



* Run 10 : 1 + 3 + 2

* 130, 310, 540 = Rem ; Run 10 : 1 + 2 + 3

* Run 700 : 4

```

1 REM*****
2 REM*
3 REM* CIRCUIT RESONANT *
4 REM*
5 REM*****
10 HIRES
20 Q=20
30 U=0
40 FOR X=0.5005 TO 1.5 STEP 0.005
50 Z=1/(1+((X-1/X)*Q)^2)^0.5
60 P=-ATN(Q*(X-1/X))
70 CURSETU,100-100*Z,1:CURSETU,150-30*P,1
80 U=U+1:NEXTX
90 REM AXES
100 CURSET0,100,1:DRAW 200,0,1
110 CURSET0,150,1:DRAW 200,0,1
120 CURSET100,199,1:DRAW 0,-199,1
130 GOTO400
190 REM Recopie rapide d'ecran
200 POKE 49,255
210 LPRINT CHR$(#08)
220 B=192:C=128:D=64:E=32
230 FORX=40999 TO 40960 STEP-1
240 FORY=X TO X+7960 STEP40
250 A=PEEK(Y)
255 IF A >= 64 THEN A=A-64
260 IF A >= 32 THEN A=(A+192) ELSE A=(A+128)
270 LPRINT CHR$(A);
280 NEXT Y:LPRINT CHR$(#0D)
290 NEXT X
300 LPRINT CHR$(#0F):POKE49,80
310 RETURN
400 REM Etude du temps de retard de groupe
410 POKE28,0
420 PRINT"Appuyer sur une touche Pour Passer
a l'etude du temps de retard"
430 GET A$
440 POKE28,203
450 U=0:HIRES
460 FOR X=0.98TO 1.02STEP 2E-4
470 Z=1/(1+((X-1/X)*Q)^2)^0.5
480 T=(-ATN(Q*((X-0.005)-1/(X-0.005)))+ATN(Q*
((X+0.005)-1/(X+0.005))))*100
490 CURSETU,100-100*Z,1:CURSETU,199-2*ABS(T),
1

```

```

500 U=U+1:NEXTX
510 CURSET0,100,1:DRAW200,0,1
520 CURSET0,199,1:DRAW200,0,1
530 CURSET100,199,1:DRAW0,-199,1
540 GOSUB190
550 END

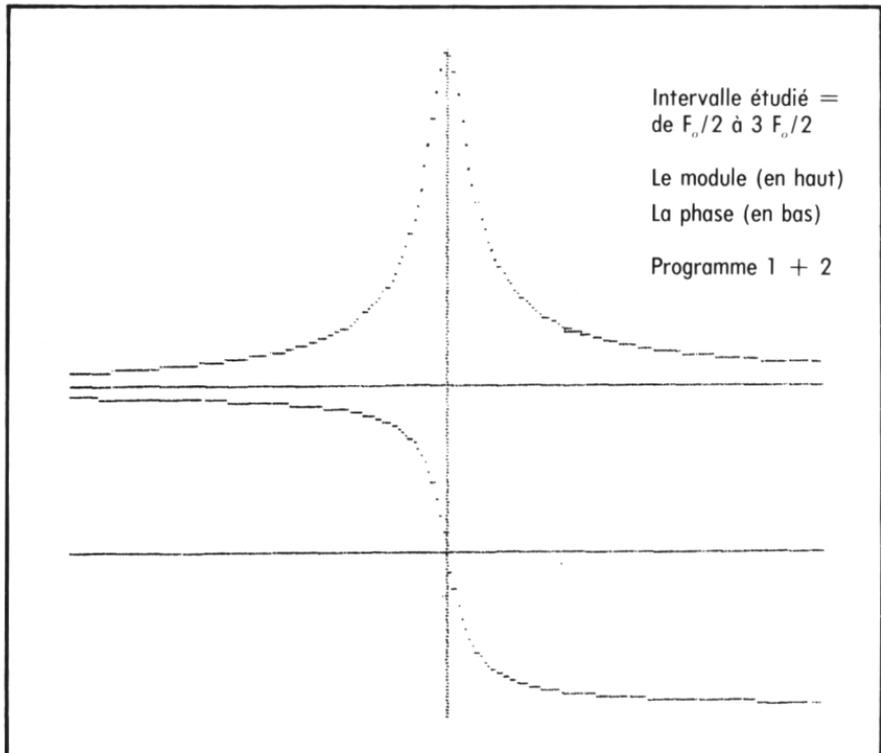
700 REM# Courbes sur l'imprimante #
705 LPRINT CHR$(#0F)
710 LPRINT CHR$(#10);"02Temps de retard";CHR$(#10);"27Phase";
720 LPRINTCHR$(#10);"50Module";CHR$(#0D)
730 LPRINT
740 LPRINTCHR$(#10);"000";CHR$(#10);"1920";CHR$(#10);"22-PI/4";
750 LPRINTCHR$(#10);"300";CHR$(#10);"33PI/4";CHR$(#10);"430";
760 LPRINTCHR$(#10);"601";CHR$(#0D)
780 LPRINTCHR$(#08):POKE49,255
800 Q=20
810 FOR V=0.8 TO 1.2 STEP0.007
820 FOR N=0 TO 6
830 X=V+N/1000
840 Z=1/(1+((X-1/X)*Q)^2)^0.5
850 P=-ATN(Q*(X-1/X))
860 T=(-ATN(Q*((X-0.005)-1/(X-0.005)))+ATN(Q*((X+0.005)-1/(X+0.005))))*100
870 B=5+Z*100
880 LPRINTCHR$(#1B);CHR$(#10);CHR$(1);CHR$(B);CHR$(128+2^N);
890 B=180+40*P
900 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(B);CHR$(128+2^N);
910 B=3*T
920 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(B);CHR$(128+2^N);
930 GOSUB2000
950 NEXTN:LPRINTCHR$(#0D)
960 NEXTV
970 LPRINT
980 LPRINTCHR$(#0F):POKE49,80
2000 REM # Axes sur l'imprimante #

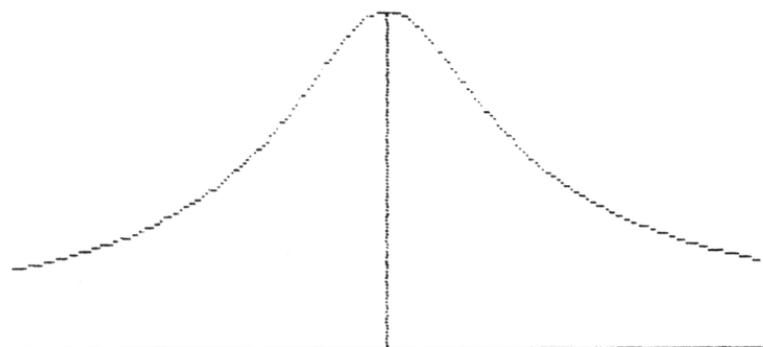
```

```

2010 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CH
R$(120);CHR$(136);
2020 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CH
R$(150);CHR$(136);
2030 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CH
R$(210);CHR$(136);
2040 LPRINT CHR$(#1B);CHR$(#10);CHR$(1);CH
R$(105);CHR$(136);
2050 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CH
R$(0);CHR$(#FF);
2060 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CH
R$(180);CHR$(#FF);
2070 LPRINT CHR$(#1B);CHR$(#10);CHR$(1);CH
R$(5);CHR$(#FF);
2080 RETURN

```



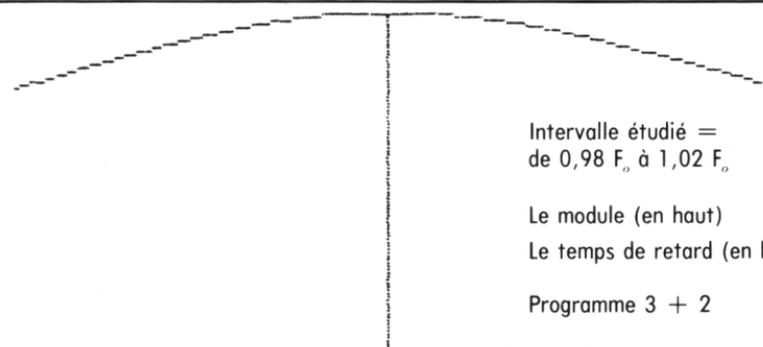
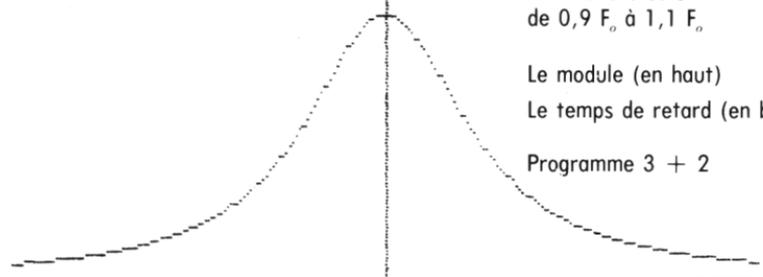


Intervalle étudié =
de $0,9 F_0$ à $1,1 F_0$

Le module (en haut)

Le temps de retard (en bas)

Programme 3 + 2

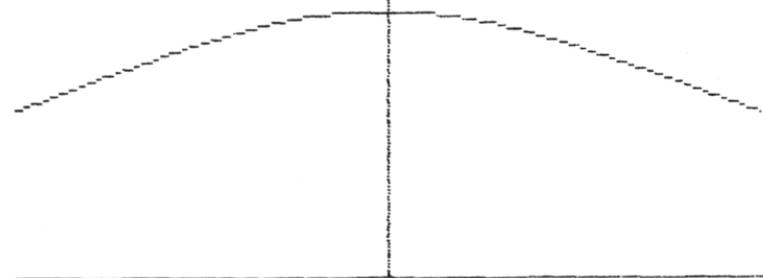


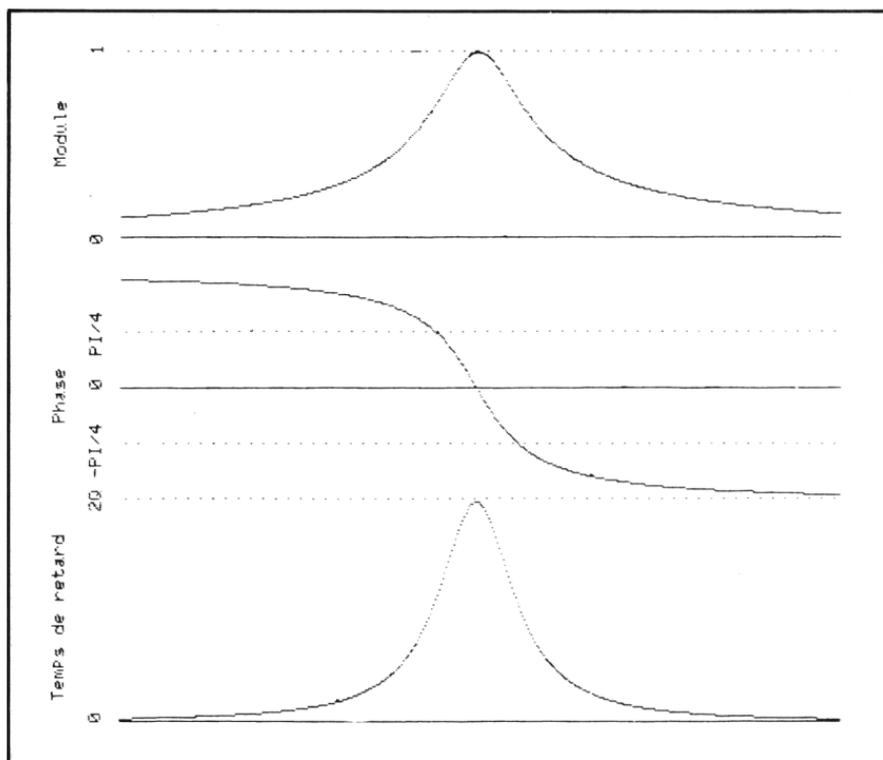
Intervalle étudié =
de $0,98 F_0$ à $1,02 F_0$

Le module (en haut)

Le temps de retard (en bas)

Programme 3 + 2



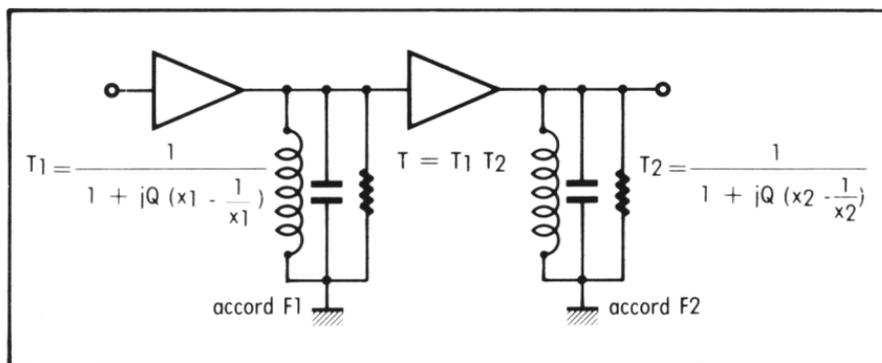


Courbes obtenues avec la partie 4

CIRCUITS DÉCALÉS

Le circuit résonant LC a un accord pointu. Il est intéressant pour sélectionner une fréquence parmi d'autres. Pour certaines applications, de type passe bande, il a une courbe de réponse trop aiguë et les flancs pas assez raides. Il est alors possible d'utiliser deux circuits résonants en cascade, à fréquences d'accord différentes, pour obtenir ce genre de gabarits.

Dans ce cas, la transmittance globale est égale au produit des transmittances correspondant à chacun des deux circuits (dont nous supprimerons les coefficients de qualité Q identiques).



en appelant $X_1 = f/f_1$ et $X_2 = f/f_2$ avec f_1 et f_2 pour fréquences de résonance.

Nous poserons pour simplifier :

$X_1 = X(1 + N)$ et $X_2 = X(1 - N)$; X reste donc la variable réduite de la fréquence. Nous décalerons les deux circuits concernés de part et d'autre de f_0 , leurs fréquences de résonance respectives étant alors : $F_1 = F_0 / (1 + N)$ et $F_2 = F_0 / (1 - N)$ car N est petit.

En faisant varier N , nous « écartons » les deux courbes de façon symétrique par rapport à f_0 .

Des programmes complémentaires montrent d'une part, comment se construit T à partir de T_1 et de T_2 et d'autre part, comment varie l'argument de T (de $+180^\circ$ à -180°) en fonction de la fréquence. On pourrait en déduire la variation du temps de retard de groupe du montage.

Programme

- * Lignes 10 à 100 : tracé des courbes correspondant à diverses valeurs de N.
- * Lignes 102 à 140 : tracé de la courbe correspondant à un circuit résonant seul (centré sur f_0).
- * Lignes 200 à 320 : recopie d'écran.

Variables

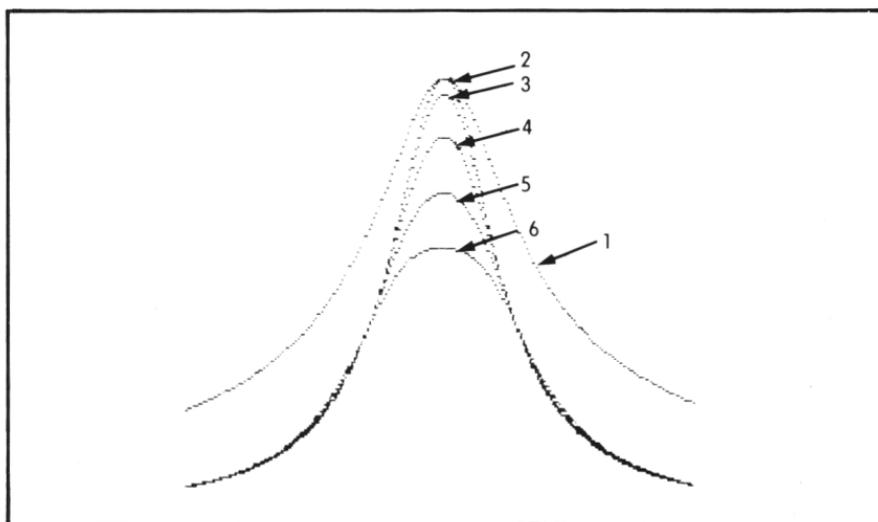
- * N : variable correspondant à l'écart entre les fréquences d'accord des deux circuits. Cet écart vaut $2Nf_0$.
- * Q : facteur de qualité des deux circuits.
- * X, X1, X2 : variables réduites de la fréquence ($X = f/f_0$; $X1 = f/f1$; $X2 = f/f2$.)
- * T1 et T2 : modules des transmittances de chacun des deux circuits.
- * T : module de la transmittance obtenue en mettant en cascade les deux circuits résonants décalés. On a évidemment $T = T1 T2$.
- * U : variable de position horizontale sur l'écran.
- * P, P1, P2 : déphasages introduits par chacun des circuits concernés.

```
5 HIRES:Q=20
10 FOR N=0 TO 0.025 STEP 0.005
20 U=0
30 FOR X=0.9 TO 1.1 STEP 0.001
35 X1=X*(1+N)
40 T1=1/(1+(Q*(X1-1/X1))^2)^0.5
45 X2=X*(1-N)
50 T2=1/(1+(Q*(X2-1/X2))^2)^0.5
60 T=T1*T2
70 CURSET20+U,199-190*T,1
80 U=U+1
90 NEXT X
100 NEXTN
102 U=0
105 FOR X=0.9 TO 1.1 STEP 0.001
110 T=1/(1+(Q*(X-1/X))^2)^0.5
120 CURSET20+U,199-190*T,1
130 U=U+1
140 NEXTX
200 REM Recopie d'écran
210 LPRINT CHR$(#08):POKE49,255
```

```

220 FOR X=40999 TO 40960 STEP -1
230 FOR Y=X TO X+7960 STEP40
240 A=PEEK(Y)
250 IF A>= 64 THEN A=A-64
260 IF A >= 32 THEN A=(A+192) ELSE A=(A+128)
270 LPRINT CHR$(A);
280 NEXT Y
290 LPRINT CHR$(#0D)
300 NEXT X
310 LPRINT CHR$(#0F):POKE49,80
320 END
1000 CALL #E60A
1010 LLIST

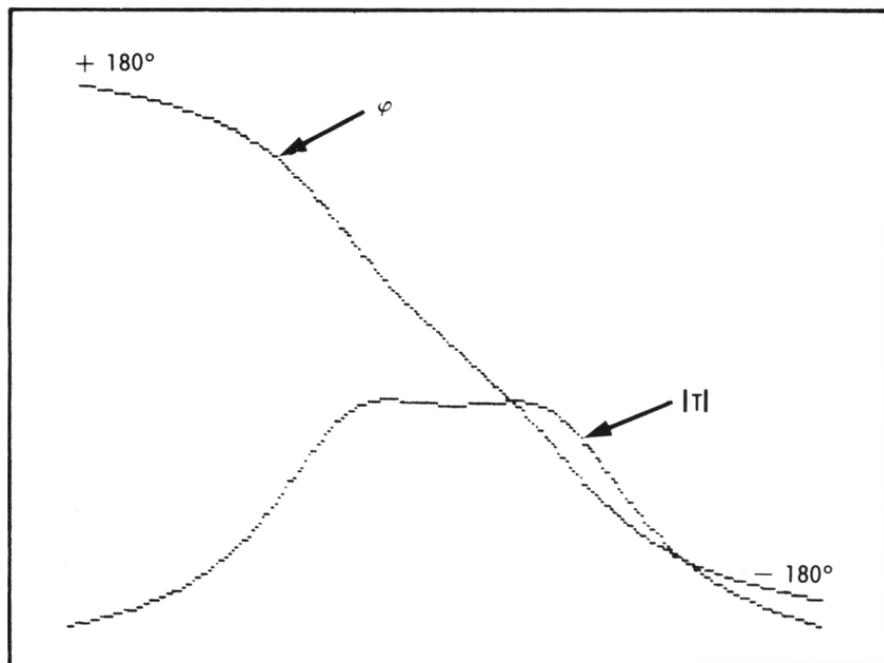
```



Intervalle étudié = $0,2 F_0$ autour de f_0

Courbes obtenues avec le programme :

- 1 : circuit LC simple (f_0)
- 2 : 2 circuits LC identiques en cascade (f_0)
- 3 : 2 circuits LC décalés de $0,01 f_0$
- 4, 5, 6 : 2 circuits LC décalés de $0,02 f_0$; $0,030 f_0$; $0,04 f_0$.



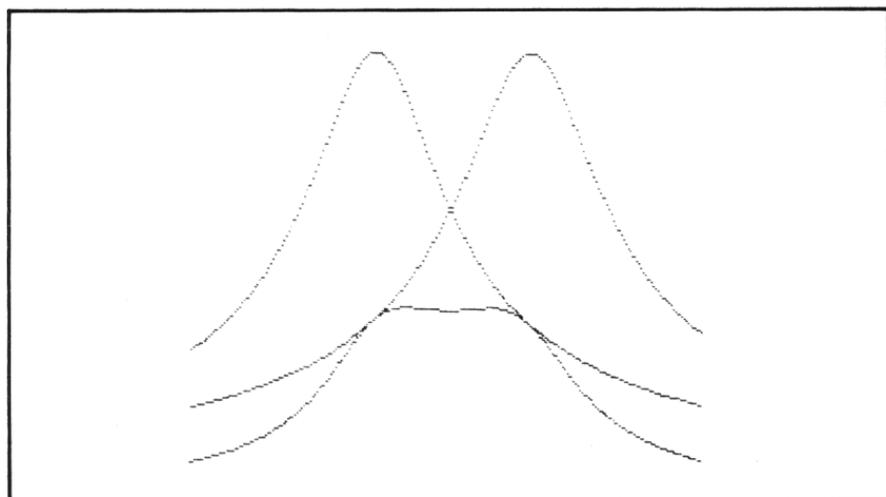
Intervalle étudié : $0,2 F_0$ autour de f_0

Programme modifié, montrant la variation de la phase pour $N : 0,030$

```

10 N=0.030
37 P1=-ATN(Q*(X1-1/X1))
47 P2=-ATN(Q*(X2-1/X2))
73 P=(P1+P2)*360/6.28
75 CURSET20+U,104-P/2,1
100 GOTO200

```



Intervalle étudié : $0,2 F_0$; écart = $0,06 F_0$.

Programme modifié, montrant la construction des courbes

```

10 N=0.030
72 CURSET20+U,199-190*T1,1
74 CURSET20+U,199-190*T2,1
100 GOTO210

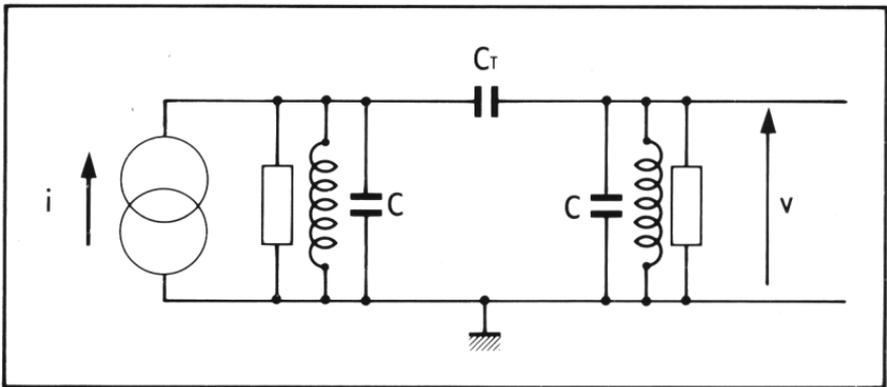
```

CIRCUITS COUPLÉS

La transmittance de circuits couplés est difficile à établir et délicate à interpréter. Il est cependant possible d'obtenir les courbes de réponses correspondantes par la méthode d'Evans. Cette méthode fait intervenir les zéros-valeurs de $j\omega$ qui annulent le numérateur de la transmittance (qui est une fonction complexe de $j\omega$) et les pôles valeurs de $j\omega$ qui annulent le dénominateur. Ces valeurs particulières de $j\omega$ étant trouvées tracer la courbe de réponse du système revient à exploiter des considérations géométriques dans le plan complexe. Dans cette étude, nous ne nous intéresserons pas à la phase de la transmittance.

La méthode proposée déborde largement du cadre de l'étude des circuits couplés. Elle s'applique à tout système dont on connaît les pôles et les zéros. Elle est très utilisée en servomécanisme.

Notre étude portera sur le circuit suivant :



Dans ce cas, on démontre que :

$$Z_T = \frac{v}{i} = \frac{jk\omega_0}{4(C + C_T)} \times \frac{1}{(p-p_1)(p-p_2)}$$

Relation dans laquelle :

$$* k \text{ est le facteur de couplage} = \frac{C_T}{C + C_T}$$

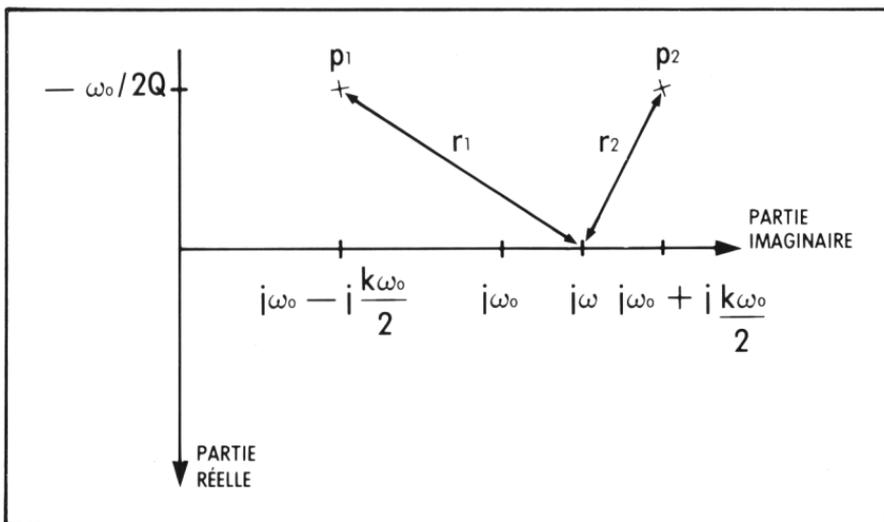
* ω_0 est la fréquence de résonance des circuits R, L, C pris séparément et supposés ici identiques. Q est leur facteur de qualité commun.

* p_1 et p_2 sont les pôles de la fonction ; on a :

$$p_1 = -\frac{\omega_0}{2Q} + j\omega_0 + j\frac{k\omega_0}{2} \text{ et } p_2 = -\frac{\omega_0}{2Q} + j\omega_0 - j\frac{k\omega_0}{2}$$

* $p = \text{variable complexe} = j\omega$

Dans le plan complexe, on a :



On constate que le module de Z_t peut s'écrire :

$$Z_t = \frac{k\omega_0}{4(C + C_T)} \times \frac{1}{r_1 r_2}$$

en appelant r_1 et r_2 les distances séparant les pôles du point $j\omega$. r_1 et r_2 sont les hypoténuses de triangles rectangles faciles à exploiter. $j\omega$ correspond à la pulsation pour laquelle on calcule Z_t .

Pour avoir l'allure des courbes en fonction de k , on peut se contenter d'évaluer $k/r_1 r_2$.

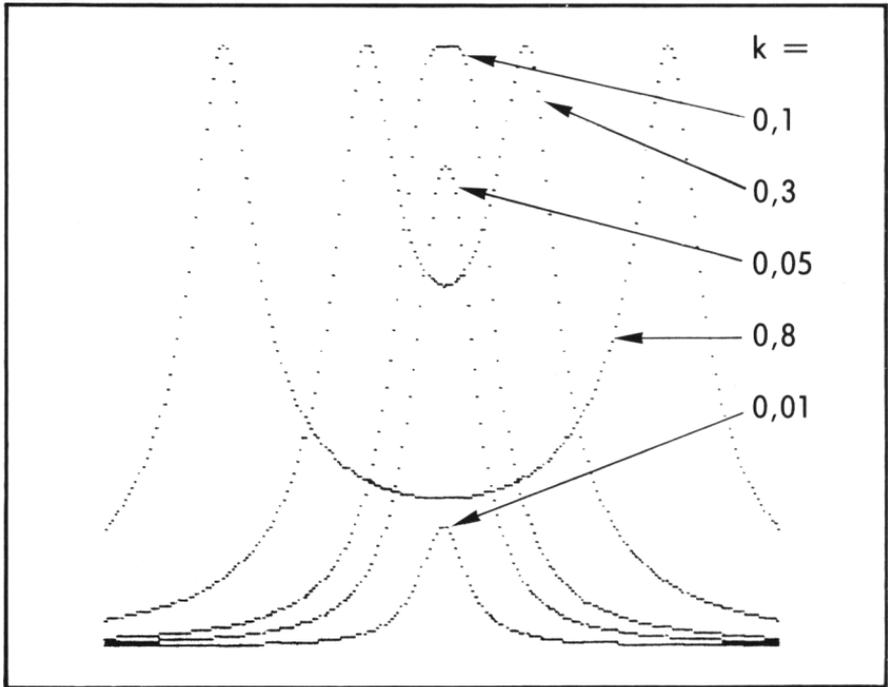
Nous étudierons un circuit pour lequel $Q = 10$, $f_0 = 1$ MHz. k prendra des valeurs comprises entre 0,01 et 0,8. La fréquence f variera de 0,4 MHz à 1,6 MHz.

LE PROGRAMME

- * Lignes 20, 30 et 50 : définition des constantes intervenant dans les calculs, permettant d'accélérer ceux-ci. A est un facteur d'échelle employé pour la disposition des courbes sur l'écran.
- * Lignes 70, 80 : calcul des distances r_1 et r_2 .
- * Lignes 40 et 110 : définition des différentes valeurs de k.
- * Ligne 90 : affichage du point calculé sur l'écran.
- * Lignes 200 à 340 : recopie d'écran sur imprimante.

VARIABLES

- * f_0 et Q : tels que définis plus haut.
- * f : fréquence considérée.
- * k : facteur de couplage (son nom est G en variable de boucle).
- * A et $Y\phi$: facteurs d'échelle.
- * R_1 et R_2 : distances correspondant aux deux pôles.
- * B, C, D, E, U, N : noms donnés à des constantes.



$f_0 = 1\text{MHz}$

$Q=10$

Largeur des courbes = 1,2 MHz

```

1 REM*****
2 REM* *
3 REM* CIRCUITS *
4 REM* COUPLES *
5 REM* *
6 REM*****
10 HIRES
20 F0=1E6
30 Q=10
40 FOR G=1 TO 5:READ K
50 A=(1E-13)/K:Y0=200:D=2:U=0.5:N=0
60 FOR F=0.4E6 TO 1.6E6 STEP 6E3
70 R1=((ABS(F-F0+F0*K/D))^D +(F0/(D*K))^D)^U
80 R2=((ABS(F-F0-F0*K/D))^D +(F0/(D*K))^D)^U
90 N=N+1:CURSET N,Y0-(1/(R1*R2))/A;.1
100 NEXT F
110 DATA 1E-2,5E-2,1E-1,0.3,0.8
120 NEXT G
200 REM copie d'ecran sur imprimante
210 B=192:C=128:D=64:E=32
220 POKE49,255
230 LPRINT CHR$(#08)
240 FOR X=40999 TO 40960 STEP-1
250 FOR Y= X TO X+7960 STEP 40
260 A=PEEK(Y)
270 A=A-D:IF A > E THEN A=(A+B) ELSE A=(A+C)
280 LPRINT CHR$(A);
290 NEXT Y:LPRINT CHR$(#0D)
300 NEXT X
310 LPRINT CHR$(#0F)
320 LPRINT "F0 = 1MHz"
330 LPRINT "Q=";Q
340 LPRINT"Largueur des courbes = 1,2 MHz"

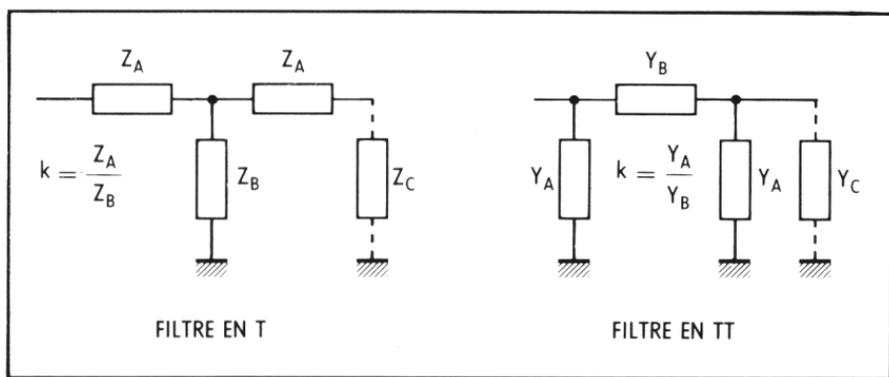
```


Chapitre 2

FILTRAGE

FILTRES PASSIFS

Nous nous limiterons au cas des filtres passifs symétriques non dissipatifs, fermés sur leur impédance caractéristique (Z_c). Leur schéma général est le suivant :



La transmittance d'un filtre passif est étudiée en deux étapes. D'abord, calcul d'une courbe universelle, valable pour tous ces filtres, ensuite l'application au cas particulier : ici, un filtre passe bas, dans lequel Z_o est une inductance et Z_b un condensateur.

On démontre que, pour ces deux circuits :

$$\bar{T} = \frac{V_2}{V_1} = (k + 1) \pm \sqrt{k(k + 2)} ; -\infty < k < +\infty$$

alors :

si $-2 \leq k \leq 0$: $|\bar{T}| = 1$ et $\text{Arg}(\bar{T}) = \text{Arc Cos}(1 + k)$

si $k < -2$: $|\bar{T}| = (1 + k) + \sqrt{k(k + 2)}$; $\text{Arg}(\bar{T}) = \pi$

si $k > 0$: $|\bar{T}| = (1 + k) - \sqrt{k(k + 2)}$; $\text{Arg}(\bar{T}) = 0$

Nous prendrons comme exemple le filtre en T, dans lequel $Z_o = jL\omega$ et $Z_b = 1/jC\omega$.

Alors $k = -LC\omega^2$ varie entre $-\infty$ et 0. (On peut écrire : $k = -(f/f_0)^2$).

Le programme n° 1 permet la construction de courbes universelles en fonction de k . Le programme n° 2 fait correspondre, pour un circuit donné, module et phase de la transmittance aux valeurs de f envisagées.

PROGRAMME N° 1

Les valeurs du module T et de la phase P de la transmittance sont calculées par zones de valeurs de k :

- * Lignes 10 à 40 : $-10 \leq k \leq -2$.
- * Lignes 50 à 80 : $-1,9 \leq k \leq -0,1$.
- * Lignes 90 à 130 : $0 \leq k \leq 10$.
- * Ligne 500 : affichage de la fonction sur l'écran.
- * Lignes 1000 à 1340 : dessin des axes et des graduations sur l'écran.
- * Lignes 2000 à 2030 : routine d'impression des mots « module » et « phase » sur l'écran.

PROGRAMME N° 2

Dans l'hypothèse où $k = -(f/f_0)^2$, nous faisons varier f de f_0 à $2 f_0$ par bond de $f_0/25$. Pour chaque valeur de k , il est fait la moyenne des valeurs de T et P correspondant aux valeurs entières de l'indice des matrices T (200) et P (200) encadrant la valeur trouvée N .

- * Lignes 10 à 160 : identiques au programme précédent.
- * Lignes 200 à 280 : évaluation et impression des valeurs de T et P pour l'intervalle de fréquence étudié.

VARIABLES

PROGRAMME N° 1

- * k : paramètre utilisé dans les fonctions T et P .
- * T : module de la transmittance.
- * P : phase de la transmittance.
- * X : variable d'incrémentatation, assurant la position horizontale des points sur l'écran.
- * U, N, L : variables de boucles locales (tracé d'axes...).

PROGRAMME N° 2

- * T (200) et P (200) : mémorisation des différentes valeurs de T et P .

* X : variable d'incrémentation.

* N : valeur de l'indice des matrices T (200) et P (200).

PROGRAMME N° 1

```
1 REM*****
2 REM* *
3 REM* FILTRES PASSIFS *
4 REM* *
5 REM*****
9 HIRES
10 FOR K=-10 TO -2 STEP 0.1
20 T=ABS(K+1+(K^2+2*K)^0.5):P=180
30 GOSUB500
35 X=X+1
40 NEXT K
50 FOR K=-1.9 TO -0.1 STEP 0.1
60 T=1:P=(180/PI)*ATN((K^2+2*K)/(K+1))
63 IF K < -1.05 THEN P=180-P
66 IF K > -1.05 THEN P=ABS(P)
70 X=X+1
75 GOSUB500
80 NEXTK
90 FOR K=0 TO 10 STEP 0.1
100 T=1+K-((K^2+2*K)^0.5):P=0
110 GOSUB500
120 X=X+1
130 NEXT K
140 GOTO 1000
500 CURSETX,194-90*T,1:CURSETX,90-P/3,1:RETURN
1000 REM trace de l'axe horizontal Pour le module
1010 CURSET0,194,0:DRAW 199,0,1:CURSET200,190,0
:CHAR62,0,1
1020 CURMOV6,0,0:CHAR75,0,1
1030 CURSET 0,90,1:DRAW200,0,1:CURSET200,86,0:C
HAR62,0,1
1040 CURMOV10,0,0:CHAR75,0,1
1050 FOR U=0 TO 105 STEP 105
1060 FOR N =0 TO 190 STEP10
1070 CURSET N,194-U,1:DRAW0,-5,1
1080 NEXTN
1090 CURSET 188,181,0:CHAR49,0,1
1100 NEXTU
1200 CURSET100,194,1:DRAW0,-92,1:CURSET98,97,0:
CHAR94,0,1
1210 CURSET110,94,1:A$="MODULE":GOSUB2000
```

```

1220 FOR L=194 TO 104 STEP-9
1230 CURSET100,L,1:DRAW5,0,1:NEXTL
1240 CURSET 90,100,0:CHAR 49,0,1
1300 CURSET 100,90,1:DRAW0,-63,1:CURSET 98,21,0
:CHAR94,0,1
1310 CURMOV 8,-9,0:A$="PHASE":GOSUB 2000
1320 FOR L=90 TO 30 STEP -10
1330 CURSE 100,L,1:DRAW 3,0,1:NEXTL
1340 CURSET 80,26,0:A$="100":GOSUB2000
1350 END

2000 FOR N=1 TO LEN(A$)
2010 CHAR ASC(MID$(A$,N,1)),0,1
2020 CURMOV 6,0,0:NEXT N
2030 RETURN

```

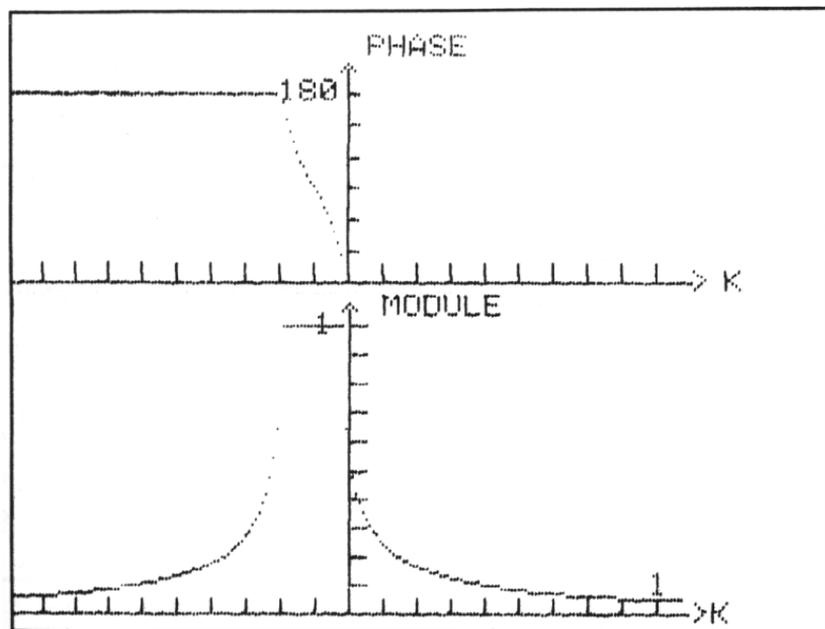
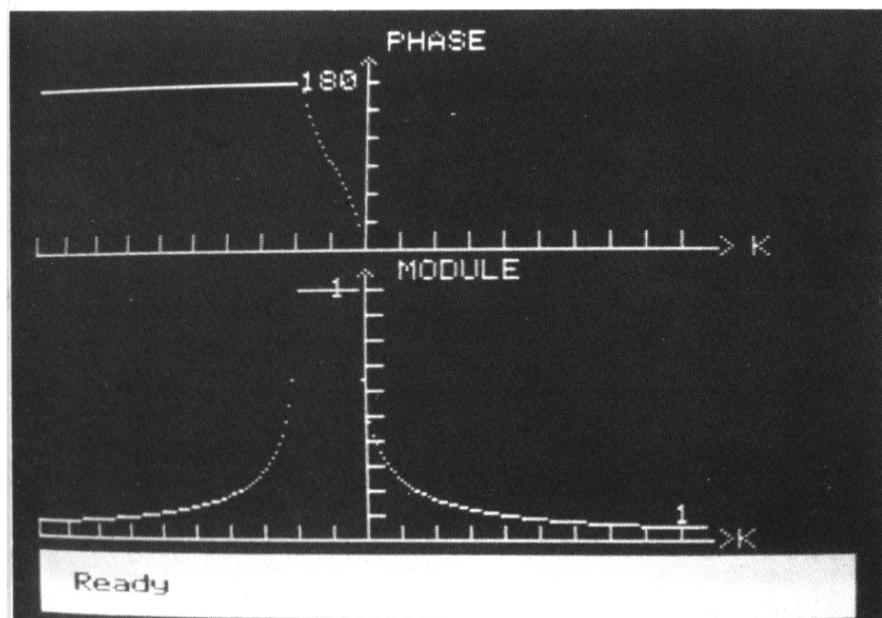
PROGRAMME N° 2

```

5 DIMT(200):DIMP(200)
7 X=-1
10 FOR K=-10 TO -2 STEP 0.1
20 T=ABS(K+1+(K^2+2*K)^0.5):P=180
30 X=X+1
35 T(X)=T:P(X)=P
40 NEXT K
50 FOR K=-1.9 TO -0.1 STEP 0.1
60 T=1:P=(180/PI)*ATN((K^2+2*K)/(K+1))
70 IF K < -1.05 THEN P=180-P
80 IF K > -1.05 THEN P=ABS(P)
90 X=X+1
100 T(X)=T:P(X)=P
110 NEXT K
120 FOR K=0 TO 10 STEP 0.1
130 T=1+K-((K^2+2*K)^0.5):P=0
140 X=X+1
150 T(X)=T:P(X)=P
160 NEXT K
200 REM EXEMPLE
205 F0=1
210 FOR F=F0 TO F0+F0 STEP F0/25
220 K=-(F/F0)^2:N=10*K+100
230 T=(T(INT(N))+T(INT(N+1)))*0.5
240 P=(P(INT(N))+P(INT(N+1)))*0.5
250 LPRINT(INT(100*F/F0)/100);"F0";
260 LPRINT TAB(25);INT(T*100)/100;

```

```
270 LPRINT TAB(45);INT(P#100)/100
280 NEXT F
290 END
```



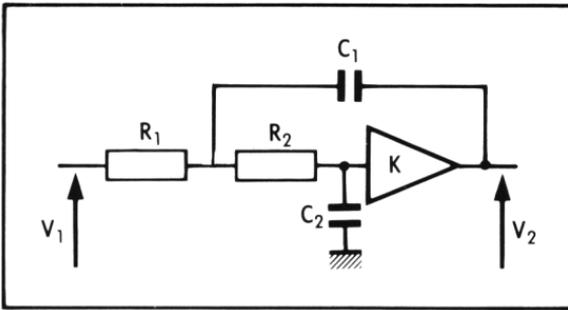
RECOPIE D'ECRAN (PROGRAMME N° 1)

F	MODULE	PHASE (°)
1 F0	1	81.23
1.04 F0	1	87.11
1.08 F0	1	92.88
1.12 F0	1	98.76
1.16 F0	1	105
1.2 F0	1	111.85
1.24 F0	1	119.57
1.28 F0	1	128.42
1.32 F0	1	138.53
1.36 F0	1	149.84
1.4 F0	1	161.92
1.44 F0	.82	174.03
1.48 F0	.58	180
1.52 F0	.44	180
1.56 F0	.4	180
1.6 F0	.36	180
1.64 F0	.33	180
1.68 F0	.29	180
1.72 F0	.27	180
1.76 F0	.26	180
1.8 F0	.23	180
1.84 F0	.22	180
1.88 F0	.2	180
1.92 F0	.19	180
1.96 F0	.18	180

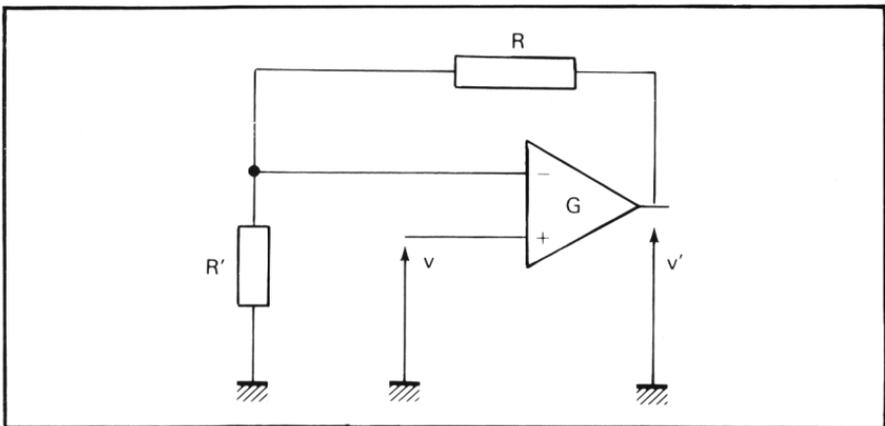
**MODULE ET PHASE DE LA TRANSMITTANCE
DU FILTRE PASSE BAS ETUDIE EN FONCTION
DE LA FREQUENCE (PROGRAMME N° 2)**

CONSTRUCTION D'UN FILTRE DU DEUXIEME ORDRE

Il existe différentes possibilités de construire un filtre passe bas du deuxième ordre. On sait l'importance de tels filtres ; il est en effet possible de réaliser un filtre du même ordre par mise en cascade de filtres du second ordre (et éventuellement du premier ordre). Nous avons choisi de réaliser un filtre actif contenant une source commandée de tension. Il s'agit là d'un amplificateur de gain positif, que l'on sait facilement réaliser à l'aide d'un amplificateur opérationnel. Le circuit proposé est le suivant :



La source commandée (K) peut être réalisée de la façon suivante :



G : amplificateur opérationnel. On a alors
$$\frac{V'}{V} = \frac{R + R'}{R'} = K \geq 1$$

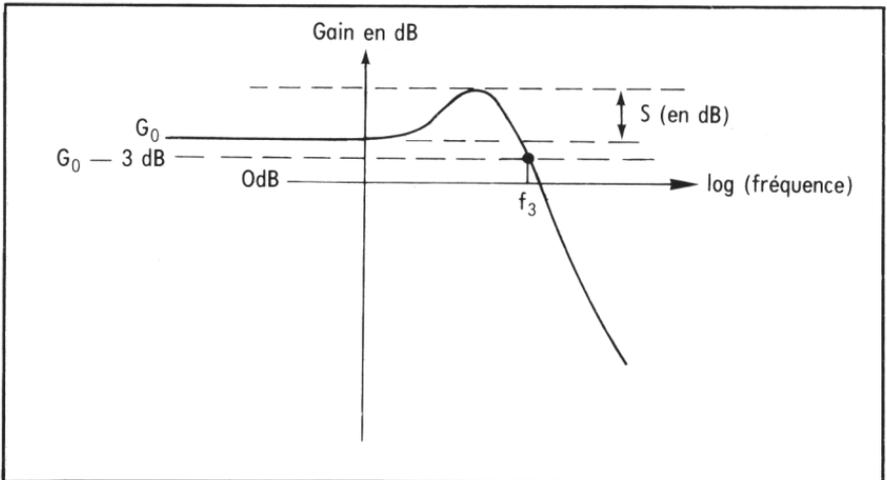
Pour le filtre proposé, on démontre que :

$$\frac{V_2}{V_1} = \frac{K}{R_1 R_2 C_1 C_2 (j\omega)^2 + [R_2 C_2 + R_1 C_2 + (1 - K) R_1 C_1] j\omega + 1}$$

un tel système est défini par :

- * son gain G_0 dans la bande passante (en dB) ($= 20 \log K$) ;
- * son coefficient de surtension S (en dB) ;
- * sa bande passante F_3 à -3 db.

Toutes ces quantités dépendent directement des valeurs de R_1, R_2, C_1, C_2 . L'allure des courbes obtenues est la suivante :



Le programme propose donc de réaliser un tel filtre.

Il est d'abord demandé à l'utilisateur les valeurs souhaitées de G_0, S, F_3 . A partir de ces données, le programme calcule des valeurs possibles pour C_1 et C_2 , en s'étant imposé $R_1 = R_2 = 10 \text{ k}\Omega$. Il y a en effet au départ 4 inconnues (R_1, R_2, C_1, C_2) pour 3 données. On peut donc s'imposer R_1 et R_2 . Il est ainsi proposé les valeurs de tous les éléments du circuit (K est imposé par G_0). Si ces valeurs ne conviennent pas, l'utilisateur peut alors proposer ses propres valeurs de R_1 et R_2 . Le programme en déduit de nouveau C_1 et C_2 .

Quand le choix des éléments est fait, le programme passe en haute résolution. Il dessine d'abord le gabarit correspondant au circuit désiré par l'utilisateur (G_0, S, f_3) et dessine la courbe théorique correspondante.

Puis, il demande les valeurs effectivement employées pour les composants, sachant que ceux-ci sont pris « dans le tiroir », c'est-à-dire correspondant à des valeurs normalisées. Il dessine alors la courbe correspondante, ce qui permet à l'utilisateur de juger si le filtre obtenu lui convient, dans la mesure où seuls les points des 2 courbes qui ne coïncident pas restent affichés.

Les axes représentés sur l'écran sont gradués en décades pour celui des fréquences, par bonds de 20 db pour celui du module de la transmittance. Ainsi, l'axe OX part de $F_0/100$ et va jusqu'à $100 F_0$ (2 décades de part et d'autre de F_0) ; OY est gradué de + 20 db à - 80 db.

Les relations mathématiques utilisées sont les suivantes : partant de S, on en déduit l'amplitude de la résonance .

$$TR = \exp \left(\frac{S \times 2,3}{20} \right)$$

et de là le coefficient d'amortissement m du circuit :

$$m = \sqrt{\frac{1 - \sqrt{1 - 1/TR^2}}{2}}$$

puis,

$$U = \sqrt{\frac{-(4m^2 - 2) + \sqrt{(4m^2 - 2)^2 + 4}}{2}} \quad \text{qui vaut } \frac{F_3}{F_0} ,$$

d'où la Valeur de la fréquence propre du circuit $F_0 = \frac{F_3}{U}$

$$\text{Enfin, posant de nouveau } U = \sqrt{\frac{C_2}{C_1}} = \frac{m + \sqrt{m^2 - 2(1-K)}}{2}$$

$$\text{et } V = C_1 C_2 = \frac{1}{R_1 R_2 \omega_0^2} :$$

on obtient $C_2 = \sqrt{U^2 V}$ et $C_1 = \frac{V}{C_2}$, sachant que :

$$m = \frac{1}{2} \left[2 \sqrt{\frac{C_2}{C_1}} + (1-K) \sqrt{\frac{C_1}{C_2}} \right]$$

On en déduit C_1 et C_2 , connaissant K, R_1, R_2 .

Un sous programme (630) recalcule les coefficients A et B de la fonction de transfert à partir de R_1, R_2, C_1, C_2 . Cette démarche est nécessaire puisque l'utilisateur peut faire intervenir ses propres valeurs de composant.

LE PROGRAMME

Lignes 10 à 60 : entrées des données numériques.

Lignes 70 à 150 : calcul des caractéristiques (m, F_0) du filtre.

Lignes 160 à 170 : affichage de ces caractéristiques.

Lignes 180 à 220 : calcul de C_1 et C_2 .

Lignes 230 à 300 : affichage des valeurs conseillées pour les composants.

Lignes 310 à 360 : modification éventuelle de ces valeurs.

Lignes 390 à 410 : tracé du gabarit de la fonction souhaitée par l'utilisateur.

Lignes 430 à 450 : définition de la fonction à tracer, (une décade occupant 50 points, l'incrément est de

$${}_{50}\sqrt{10} = 10^{0,02}$$

Lignes 460 à 490 : tracé de la fonction.

Lignes 510-520 : si la fonction convient, le programme s'arrête.

Lignes 530 à 600 : entrée des valeurs des composants possédés par l'utilisateur.

Ligne 620 : retour au tracé de la fonction avec ces nouvelles valeurs d'éléments. L'utilisateur du paramètre 2 dans le CURSET de la ligne 470 permet de n'afficher que les points qui ne sont pas identiques pour les 2 courbes

sous programme 630 : calcul des coefficients de la fonction étudiée

sous programme 63500 : tracé d'axes et des graduations.

LES VARIABLES

Elles ont toutes été explicitées dans les paragraphes précédents.

F_3 : fréquence de coupure à -3 db souhaitée.

S : surtension souhaitée en db.

G_0 : gain souhaité en db dans la bande passante.

TR et Z : correspondent à S et G_0 sur une échelle linéaire.

V, U : variable temporaire de calcul.

M : Coefficient d'amortissement du filtre.

W₀ et F₀ : pulsation et fréquence propre du filtre.

R₁, R₂, C₁, C₂, K = valeurs des éléments du filtre.

D : incrément pour les fréquences (= $\sqrt[50]{10}$)

FNT (W) : valeur du module de la transmittance à la pulsation W.

A et B : coefficients de la fonction de transfert

$$\left(\frac{V_2}{V_1} = \frac{1}{1 + Bj\omega + A(j\omega)^2} \right)$$

```
1 REM*****
2 REM# *
3 REM# CALCUL D'UN FILTRE *
4 REM# PASSE-BAS *
5 REM# DU SECOND ORDRE *
6 REM# *
7 REM*****
10 CLS:PAPER3:INK0
20 POKE616,6:PRINT:POKE617,5:INPUT "Frequence d
e coupure a -3dB =":F3
30 POKE616,12:PRINT:POKE617,5:INPUT"Surtension
souhaitee en dB (20)>=0)":S
40 POKE616,18:PRINT:POKE617,5:PRINT"Gain dans l
a bande Passante en dB ?"
50 POKE616,19:PRINT:POKE617,5:PRINT"Nous vous c
onseillons 0 dB)
60 POKE616,20:PRINT:POKE617,10:INPUT G0
70 LET TR=EXP(S*2.3/20)
80 LET U=(1-(1-(1/(TR^2)))^0.5)/2
90 LET M=U^0.5
100 LET H=4*(M^2)-2
110 LET U=(-H+((H^2)+4)^0.5)^0.5/(2^0.5)
120 W0=2*PI*F3/U :F0=W0/(2*PI)
130 LETK=G0
140 Z=EXP(G0*2.3/20)
150 CLS
160 PRINT"Le facteur d'amortissement du circuit
est ";INT(M*100)/100;
170 PRINT"et sa frequence naturelle est F0=";IN
T(100*W0/(2*PI))/100;" Hz."
180 LET R1=1E4:LETR2=1E4
190 LET U=(2*M+((4*M*M)+8*(Z-1))^0.5)/4
200 LET V=1/(R1*R2*(W0^2))
210 C2=(V*(U^2))^0.5
220 C1=V/C2
230 PRINT:PRINT:PRINT:PRINT:PRINT:
```

```

240 PRINT"Nous vous Proposons les valeurs suiiva
ntes Pour les composants : "
250 PRINT" R1= ";R1;" Ohm."
260 PRINT" R2= ";R2;" OHM."
270 PRINT" C2= ";INT(C2*1E11)/100;" nF."
280 PRINT" C1= ";INT(C1*1E11)/100;" nF."
290 PRINT"Il est obligatoire de Prendre K=")INT
(Z*100)/100
300 PRINT"Ces valeurs vous conviennent-elles ?"
310 GET A$
320 IF A$="0" THEN GOTO370
330 CLS:PRINT"Quelles valeurs Proposez vous Pou
r R1 et R2 ?"
340 INPUT"R1= ";R1
350 INPUT"R2= ";R2
360 GOTO 190
370 HIRES
380 GOSUB 630
390 PAPER0
400 CURSET0,0,0:FILL40-S-K,37,85
410 CURSET0,43-K,0:FILL150+K,INT((120+50*LOG(F3
/F0))/6),85
420 GOSUB63500
430 D=10^0.02
440 W=W0/(100*D)
450 DEF FNT(W)=Z/(((1-A*W*W)^2+(B*W)^2)^0.5)
460 FOR X=0 TO 199
470 CURSET20+X,40-20*LOG(FNT(W)),2
480 W=W*D
490 NEXT
500 POKE28,0
510 PRINT"Avez vous les elements conseilles ?":
GETA$
520 IF A$="0" THEN POKE28,203:STOP
530 PRINT"Indiquez les valeurs numeriques des e
lements que vous Possedez"
540 WAIT500
550 PRINT:PRINT:PRINT
560 INPUT "R1= ";R1
570 INPUT "R2= ";R2
580 INPUT "C1= ";C1
590 INPUT "C2= ";C2
600 PRINT:PRINT:PRINT
610 PRINT" Seuls les Points des 2 courbes ne co
incident Pas restent affiches"
620 GOTO 380
630 REM CALCUL DES CEFFICIENTS

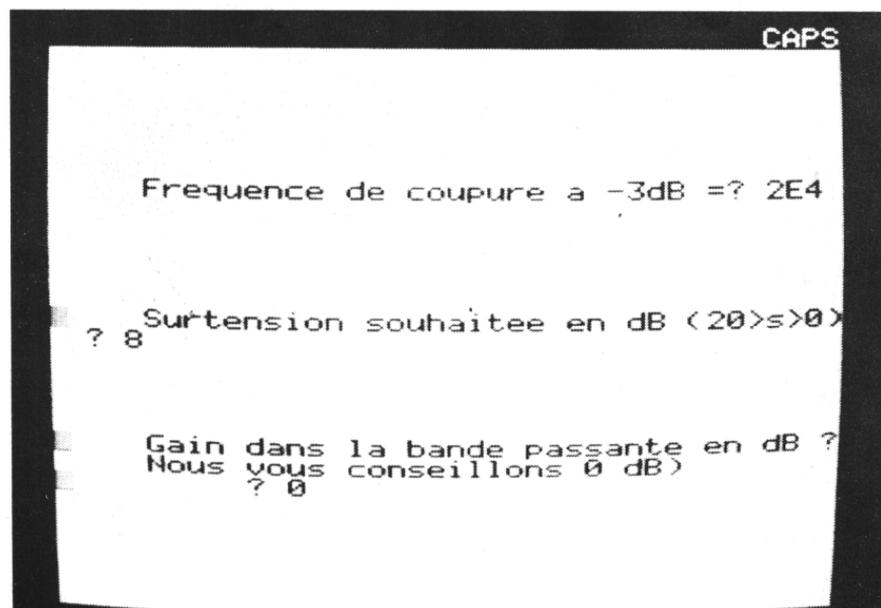
```

```

640 A=R1*R2*C1*C2
650 B=(R1*C2+R2*C2+(1-Z)*R1*C1)
660 RETURN
680 CALL #E6CA
690 LLIST
63500 HIRES
63510 CURSET20,40,0:DRAW210,0,1:DRAW-3,-3,0:CHA
R62,0,1
63520 CURSET20,199,0:DRAW0,-195,1:DRAW-2,-4,0:C
HAR94,0,1
63530 FOR N=1 TO 3
63540 CURSET20+50*N,42,0:DRAW 0,-4,1
63550 NEXT N
63560 FOR N=0 TO 5
63570 CURSET20,20+20*N,0:DRAW4,0,1
63580 NEXTN

```

EXEMPLE DE FONCTIONNEMENT



CAPS

Le facteur d'amortissement du circuit est .2 et sa fréquence naturelle est $F_0=13262.6$ Hz.

Nous vous proposons les valeurs suivantes pour les composants :

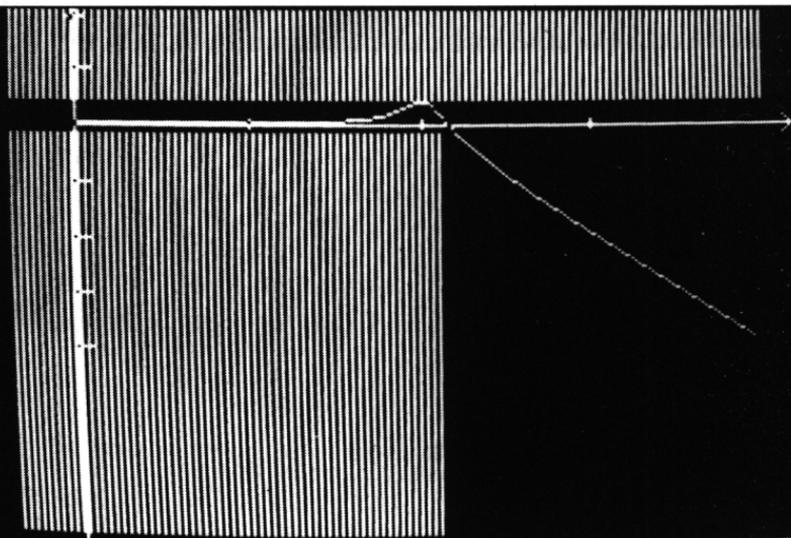
$R_1 = 10000$ Ohm :

$R_2 = 10000$ OHM :

$C_2 = .24$ nF :

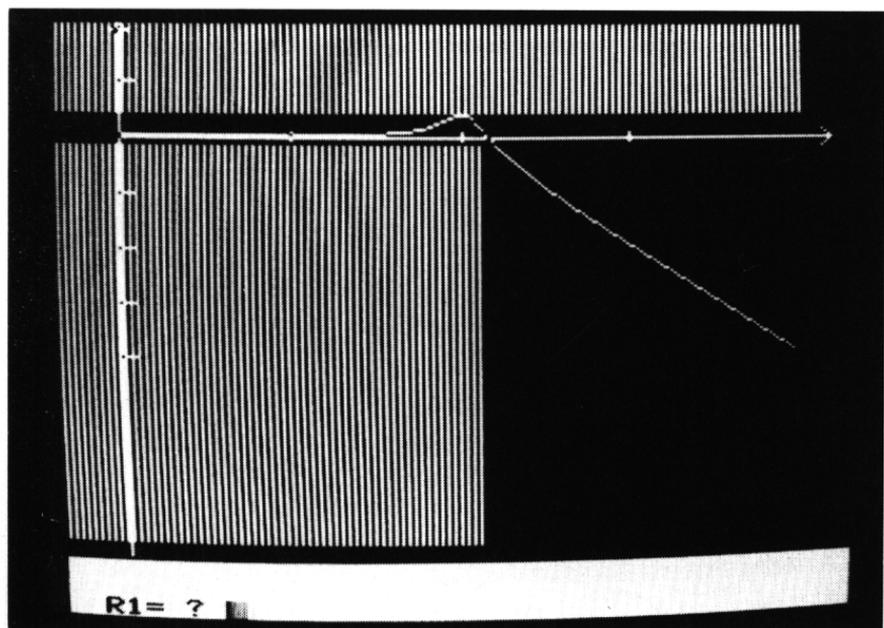
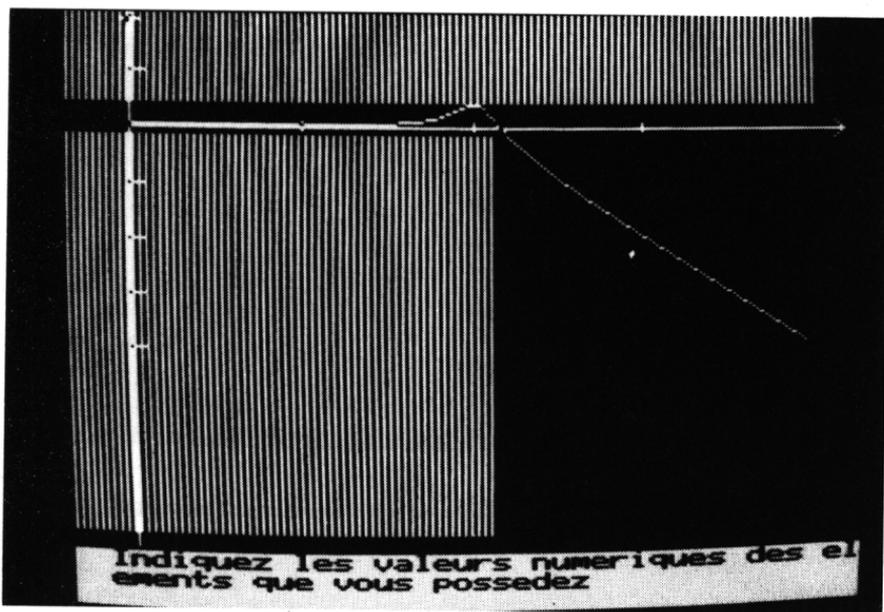
$C_1 = 5.89$ nF :

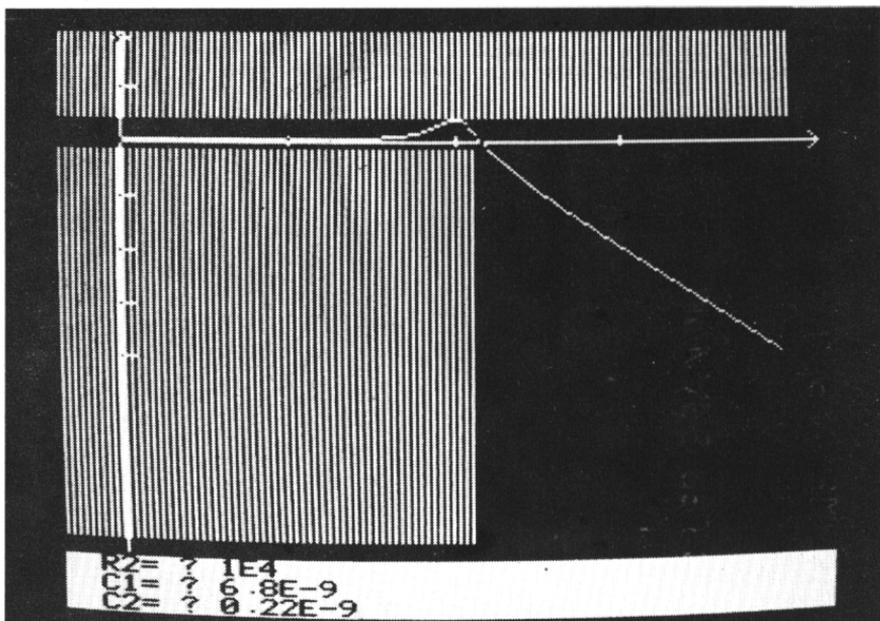
Il est obligatoire de prendre $K=1$
Ces valeurs vous conviennent-elles ?



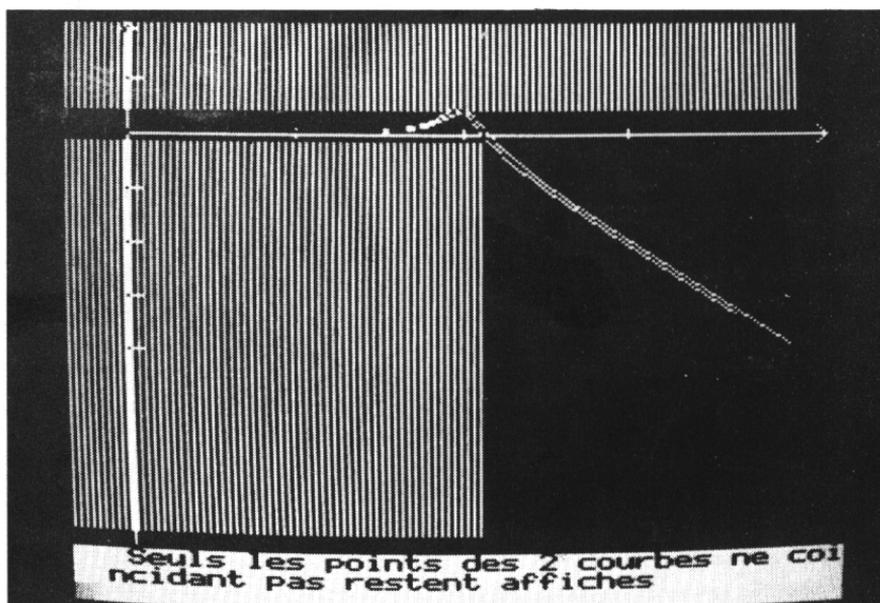
Avez vous les éléments conseillés ?

EXEMPLE DE FONCTIONNEMENT (suite)





EXEMPLE DE FONCTIONNEMENT (fin)



Chapitre 3

FOURIER

SERIES DE FOURIER

Le théorème de Fourier a un grand nombre d'applications. Il a surtout été employé comme base théorique de démonstration. S'il n'a guère été utilisé en pratique, c'est qu'il conduit à des calculs inextricables. Aujourd'hui, l'ordinateur permet de l'exploiter.

Ce théorème dit qu'à toute fonction périodique, peut se substituer une somme de fonctions sinusoïdales, dont les fréquences sont multiples de la fréquence fondamentale. Le nombre de termes de cette somme est infini ; voici trois exemples de développements :

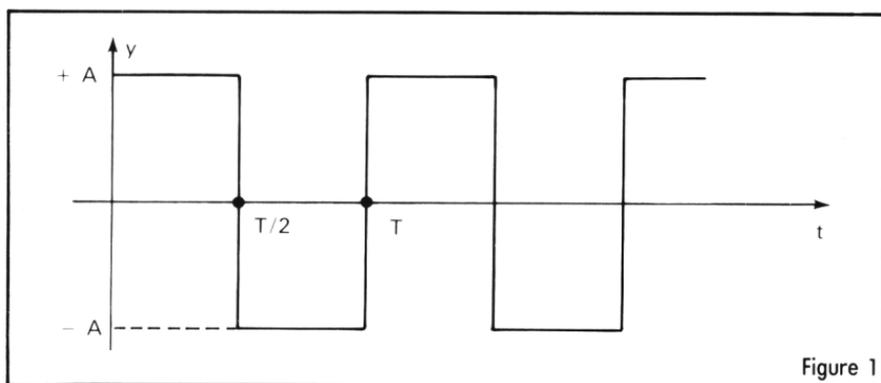


Figure 1

$$y = \frac{4 A}{\pi} \left[\sin \omega t + \frac{1}{3} \sin 3 \omega t + \frac{1}{5} \sin 5 \omega t + \dots \right]$$

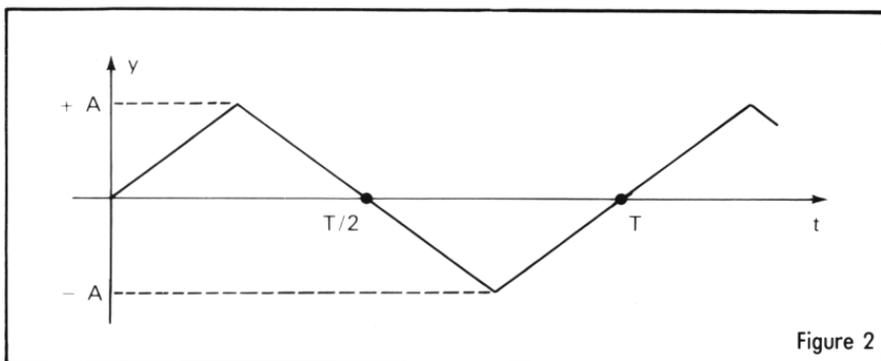
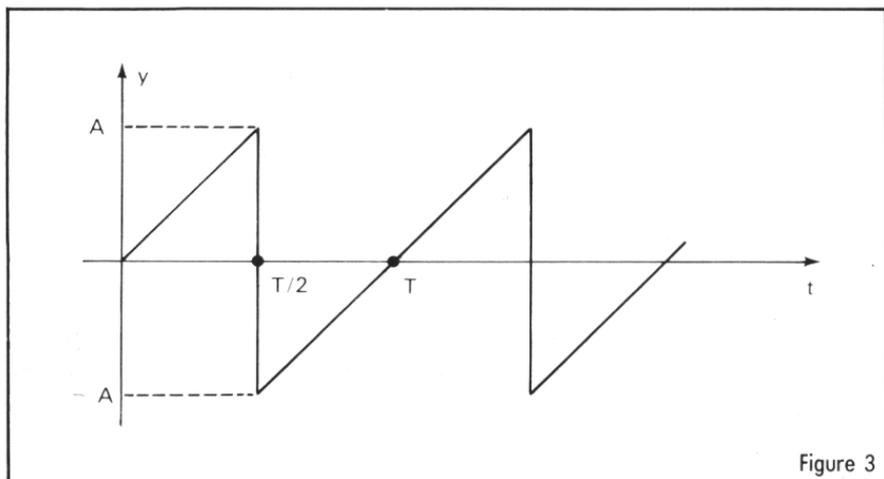


Figure 2

$$y = \frac{8 A}{\pi^2} \left[\sin \omega t - \frac{1}{9} \sin 3 \omega t + \frac{1}{25} \sin 5 \omega t - \dots \right]$$



$$y = \frac{2A}{\pi} \left[\sin \omega t - \frac{\sin 2\omega t}{2} + \frac{\sin 3\omega t}{3} - \dots \right]$$

Le programme n° 1 est démonstratif. Il propose de montrer sur l'écran la forme du signal obtenu quand on emploie K harmoniques. On visualise une période et demie du signal 3 pris en exemple. Après chaque balayage d'écran, un nouveau signal, augmenté d'un harmonique, vient remplacer le précédent.

Le programme n° 2 permet de recopier l'écran sur imprimante. Il est à utiliser avec le programme n° 1.

Le programme n° 3 permet d'obtenir sur l'imprimante la superposition de trois courbes différentes, correspondant à trois valeurs de K, dans les mêmes axes. B, C, D mémorisent les valeurs de K choisies.

* Lignes 30 à 57 : inscription sur l'écran du nombre d'harmoniques utilisés pour la courbe affichée.

* Lignes 60 à 80 : calcul de l'harmonique d'ordre K.

* Lignes 90 à 140 : affichage des points et effacement des précédents.

Variables :

* A (200) : contient la position de tous les points de la courbe sur l'écran.

* K : nombre d'harmoniques pris en compte.

* T : variable temps. La période du signal correspond à 120 points ; T varie de 0 à 179 ; une période et demie du signal est ainsi affichée.

* Y : valeur de l'harmonique K au temps T ; A (T) est la somme, au temps T, de tous les harmoniques de rang inférieur à K.

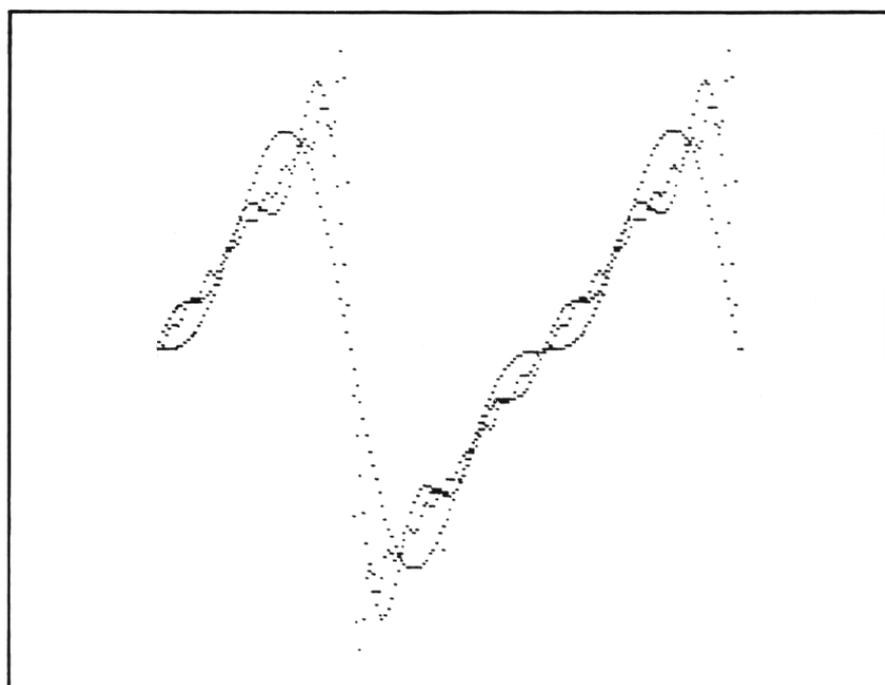
Programme n° 3

```
1 REM*****
2 REM* *
3 REM* SERIE DE FOURIER *
4 REM* *
5 REM*****
6 B=2:C=5:D=20
10 DIM A(200):DIMB(200):DIMC(200):DIMD(200)
20 HIRES
30 FOR K=1 TO 30:REM K est le nombre d'harmoniques pris en compte
40 CURSET0,0,0:CHAR75,0,1:CURMOV6,0,0:CHAR61,0,1
50 CURSET 12,0,0
53 FOR S=2 TO LEN(STR$(K))
55 CHAR ASC(MID$(STR$(K),S,1)),0,2:CURMOV6,0,0:
NEXTS:IFK = 1 THEN GOTO 60
56 CURSET12,0,0: FOR S=2 TO LEN(STR$(K-1))
57 CHAR ASC(MID$(STR$(K-1),S,1)),0,2:CURMOV6,0,0:
NEXTS
60 B=2:C=5:D=20
65 IF K = 1 THEN L = 1
70 FOR T =0 TO 179
80 Y = L*SIN(K*PI*T/60)/K
90 IF T = 179 THEN GOTO 110
100 CURSETT+1,100-55*A(T+1),0
110 A(T)=A(T)+Y
112 IFK = B THENB(T)=A(T)
114 IFK = C THENC(T)=A(T)
116 IFK = D THEND(T)=A(T)
120 CURSET T,100-55*A(T),1
130 NEXT T
140 CURSET0,100-55*A(0),0
150 NEXT K
200 LPRINT CHR$(#00)
210 LPRINT CHR$(#1C);CHR$(#FF);CHR$(192);CHR$(#1C);CHR$(#60);CHR$(192)
215 T=-1
220 FOR U= 0 TO 179 STEP 7
230 LPRINT CHR$(#FF);
240 FOR N= 0 TO 6
245 T=T+1
```

```

250 Y=150 + 50*B(T)
260 IF Y > 255 THEN B1=1:B2=Y-256 ELSE B1=0:B2=
Y
270 LPRINT CHR$(#1B);CHR$(#10);CHR$(B1);CHR$(B2
);CHR$(128+2^N);
280 Y=150 + 50*C(T)
290 IF Y > 255 THEN C1=1:C2=Y-256 ELSE C1=0:C2=
Y
295 LPRINT CHR$(#1B);CHR$(#10);CHR$(C1);CHR$(C2
);CHR$(128+2^N);
300 Y=150 + 50*D(T)
310 IF Y > 255 THEN D1=1:D2=Y-256 ELSE D1=0:D2=
Y
320 LPRINT CHR$(#1B);CHR$(#10);CHR$(D1);CHR$(D2
);CHR$(128+2^N);
330 NEXTN:LPRINT CHR$(#0D)
340 NEXT U
10000 CALL #E60A
10010 LLIST

```



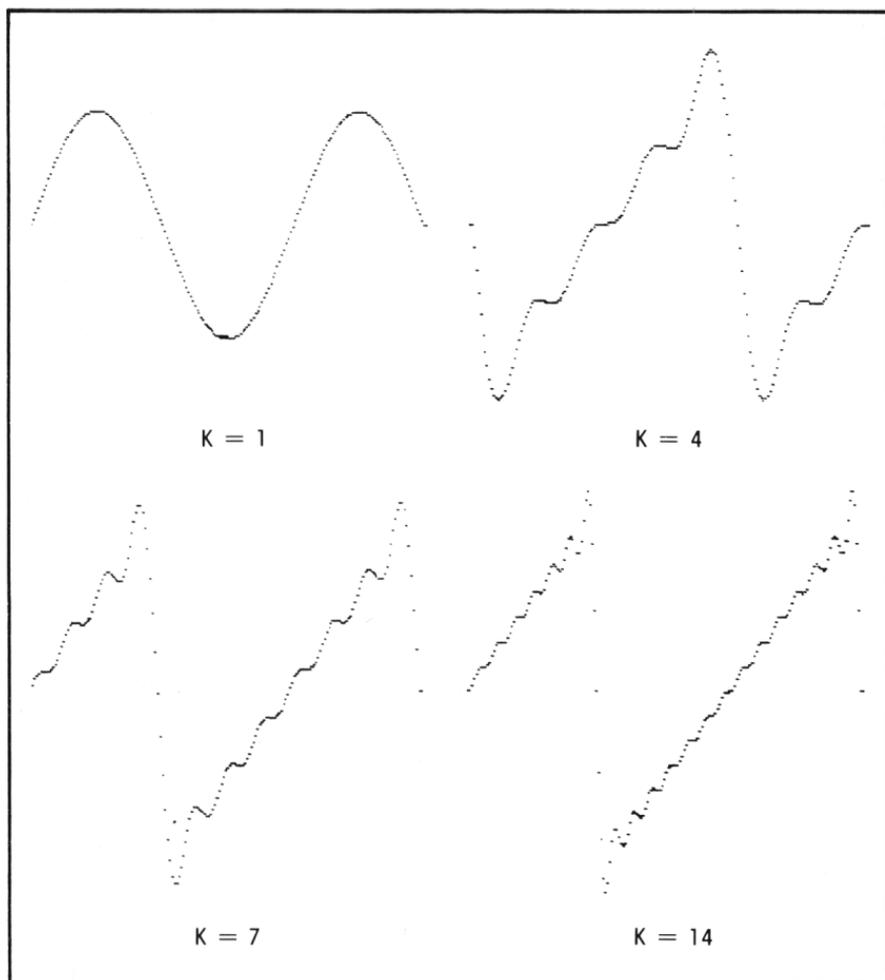
Courbes obtenues avec le programme n° 3

(Signal triangulaire reconstitué à l'aide de 2, 5, 20 harmoniques).

* **Programme n° 1** : rajouter 160 : END.

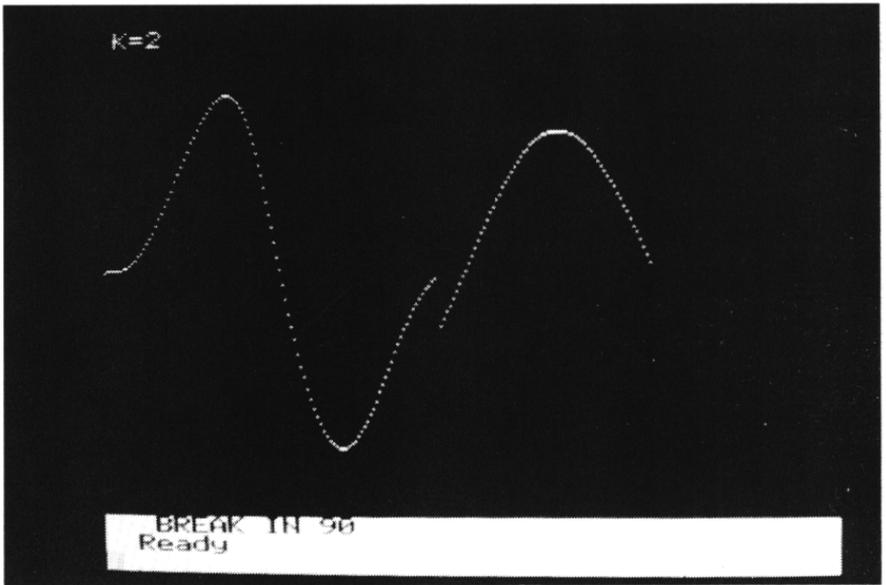
* **Programme n° 2** : mettre la valeur désirée de K en B (ligne 6) rajouter 275 GOTO 330.

* **Programme n° 3** : listing complet. Introduire les valeurs désirées de K en B, C, D, (ligne 6).

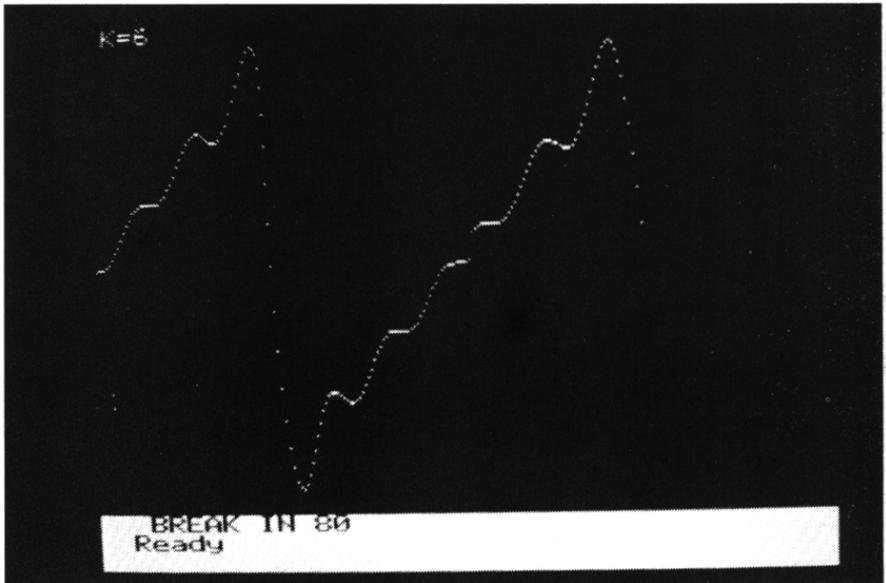


Courbes obtenues avec le programme n° 2

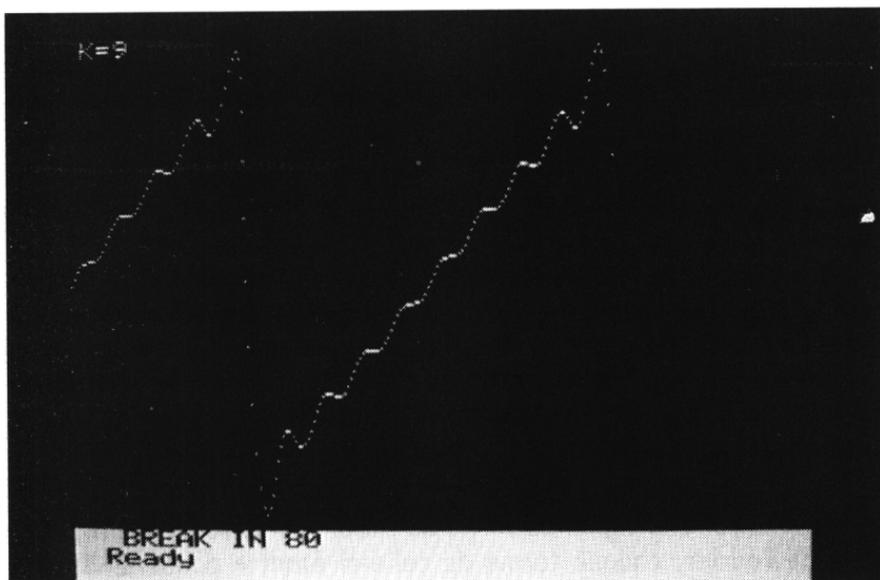
(Signal triangulaire reconstitué à l'aide de K harmoniques).



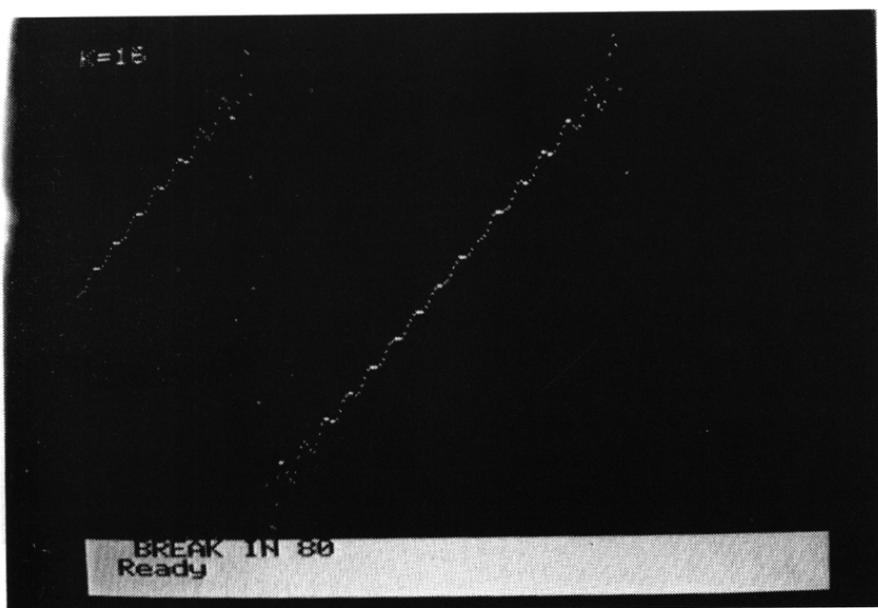
Passage de $K = 1$ à $K = 2$.



Passage de $K = 5$ à $K = 6$.



Passage de $K = 8$ à $K = 9$.



Passage de $K = 15$ à $K = 16$.

APPLICATION DE LA SERIE DE FOURIER

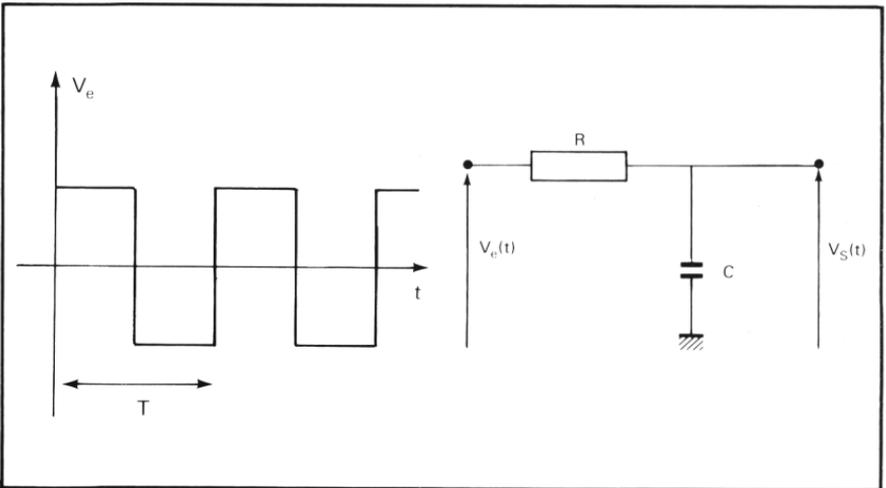
Lorsqu'un signal non sinusoïdal est appliqué à un circuit électrique, il peut être difficile de déterminer le régime permanent correspondant à la tension de sortie de ce circuit. Nous allons voir qu'un ordinateur, dont la capacité de calcul est infinie, permet de résoudre un tel problème.

Nous proposons donc :

— de relever (ou de calculer) le module et la phase de la transmittance du circuit étudié en régime sinusoïdal permanent, ce qui en général est aisé à faire.

— de construire le signal de sortie du circuit, en le remplaçant par sa série de Fourier, chaque terme de celle-ci étant « pondéré » par le module et la phase de la transmittance du circuit à la fréquence correspondante.

Le programme est très simple. Il prend comme exemple le cas d'un circuit intégrateur RC du premier ordre, soumis à une tension rectangulaire.



$$|\bar{T}| = \frac{1}{\sqrt{1 + \omega^2 R^2 C^2}}$$

$$\varphi = - \text{Arc tg } RC\omega$$

RC est la constante de temps du circuit, ω la pulsation du signal sinusoïdal incident.

Ce programme permet de comparer sur l'écran la réponse calculée par la série de Fourier du signal d'entrée et la réponse exacte, définie par une équation du type : $1 - \exp(-T/RC)$. (En toute rigueur, cette réponse n'est « exacte » que dans la mesure où RC est le dixième de la période du signal).

Remarquons toutefois que le cas pris en exemple est un peu particulier, dans la mesure où il s'agit d'un filtre de type passe bas et donc pour lequel les harmoniques de rang élevé ne jouent aucun rôle.

La période du signal T_0 est égale à 200 (points).

* Lignes 15-20 : définition de la réponse exacte.

* Lignes 30 à 60 : calcul de l'harmonique de rang K, modifié par la transmittance du circuit.

* Lignes 65 à 75 : affichage des deux fonctions, permettant de les comparer.

* Lignes 90 à 190 : recopie des courbes sur l'imprimante.

VARIABLES

* A et B sont les paramètres de calcul.

$A = 2 \pi RC/T_0$; $B = 2 \pi/T_0$: ici, $T_0 = 200$, $RC = T_0/10$.

* K : numéro de l'harmonique calculé.

* YK : harmonique de rang K.

* Y : fonction calculée avec K harmoniques.

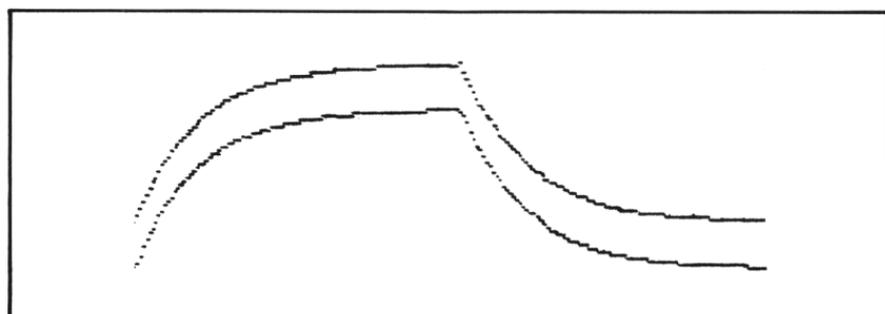
* A (200) et B (200) : mémorisation des points correspondant aux deux courbes.

```
1 REM*****
2 REM*                                     *
3 REM* Utilisation de Fourier             *
4 REM* en regime Permanent               *
5 REM*                                     *
6 REM*****
12 DIMA(200):DIMB(200)
15 DEF FNZ(T)=PI*(1-EXP(-T/20))/4
20 A=PI/5:B=PI/100
25 HIRES
30 FORT=0TO200
35 Y=0
40 FORK=1 TO 31 STEP2
```

```

50 YK=(1/(K*(1+(K*A)^2)^0.5))*SIN(K*B*T-AT
N(K*A))
55 Y=Y+YK
60 NEXTK
65 IF T <= 99 THEN Z=FNZ(T):CURSETT,110-60
*KZ,1
70 IF T > 99 THEN Z=FNZ(T-100):CURSETT,(11
0-60*PI/4)+60*KZ,1
75 CURSETT,50-30*Y,1
78 A(T)=Y:B(T)=Z
80 NEXTT
90' REM Trace des courbes sur l'imprimante
100 LPRINT CHR$(#08);T=0
110 FOR U= 0 TO 200 STEP7
120 FOR V= 0 TO 6
130 Y=100 +30*A(T)
140 IF T > 99 THEN Z= 90+PI*15-60*B(T):GOT
O150
145 Z=90+60*B(T)
150 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(
Y);CHR$(128+2*V);
160 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(
Z);CHR$(128+2*V);
170 T=T+1
180 NEXTV:LPRINTCHR$(#0D)
190 NEXTU

```



Programme

La réponse exacte et celle calculée à l'aide de la série de Fourier sont pratiquement identi-ques.

CALCUL DES COEFFICIENTS D'UNE SERIE DE FOURIER

La série de Fourier d'une fonction périodique s'écrit :

$$f(t) = A_0 + A_1 \sin \frac{2\pi t}{T} + A_2 \sin \left(2 \times \frac{2\pi t}{T} \right) + \dots$$

$$+ B_1 \cos \frac{2\pi t}{T} + B_2 \cos \left(2 \times \frac{2\pi t}{T} \right) + \dots$$

Les coefficients se calculent par les relations :

$$A_0 = \frac{1}{T} \int_0^T f(t) dt ; A_n = \frac{2}{T} \int_0^T f(t) \times \sin \left(n \times \frac{2\pi t}{T} \right) dt$$

$$\text{et } B_n = \frac{2}{T} \int_0^T f(t) \times \cos \left(n \times \frac{2\pi t}{T} \right) dt$$

Le calcul n'est aisé que pour des fonctions $f(t)$ simples.

Nous proposons de remplacer les intégrales précédentes par des sommes en nombre fini. Ainsi,

$$\frac{2}{T} \int_0^T f(t) \times \sin \left(n \times \frac{2\pi t}{T} \right) dt \text{ devient}$$

$$\frac{2}{T} \sum_0^T f(t) \times \sin \left(n \times \frac{2\pi t}{T} \right) \times \Delta t$$

t variant de 0 à T par bonds de Δt : $\Delta t = 1$ (point).

Nous définissons la période de $f(t)$ comme étant égale à 200 (points).

Un tableau contient les 200 valeurs prises par $f(t)$ sur cet intervalle.

Nous faisons le calcul dans le cas d'un signal rectangulaire ($f(t) = 1$

pour $1 \leq t \leq 100$ et -1 pour $101 \leq t \leq 200$). La série de

Fourier correspondante est alors connue et permet de chiffrer la précision

de la méthode (6 % sur le 21^o harmonique). Les termes trop

petits et non significatifs (inférieurs à 10^{-4}) sont éliminés. Une meilleure

précision pourrait être obtenue en définissant $f(t)$ sur un plus

grand nombre de points.

Le programme se déroule en deux parties. Il y a d'abord un affichage par ligne sur l'écran du numéro de l'harmonique et des valeurs de A_n et de B_n . Ensuite, l'ensemble de ces coefficients est transféré à

l'imprimante.

* Lignes 35 à 100 : calcul de la somme correspondant à l'harmonique de rang N.

* Ligne 102 : calcul de la valeur exacte de l'amplitude de l'harmonique de rang N.

* Ligne 105 : les valeurs trouvées sont arrondies.

* Ligne 118 : mise en mémoire des valeurs.

* Lignes 200 à 220 : définition de f(t).

* Lignes 1 000 à 2 100 : mise en page de l'impression du tableau des résultats.

* Lignes 2 110 à 2 150 : impression des coefficients dans le tableau.

Variables

* A (200) : valeurs de la fonction étudiée [f(t)].

* S (30) et C (30) : matrices contenant les valeurs calculées des coefficients A_n et B_n .

* E (30) : contient les coefficients (A_n) exacts de la série des sinus.

* N : numéro de l'harmonique.

* T : variable temps.

* ST : valeur d'un élément de la somme correspondant au calcul de A_n .

* CT : idem, mais pour B_n .

* SE : valeur exacte du coefficient A_n .

* U : variable de boucle locale.

```
1 REM*****
2 REM*                               *
3 REM*   COEFFICIENTS               *
4 REM* D'UNE SERIE DE              *
5 REM*   FOURIER                    *
6 REM*                               *
7 REM*****
10 DIM A(200):DIM S(30): DIM C(30):DIM E(30)
20 GOSUB 200
30 FOR N= 1 TO 30
35 S=0:C=0
40 FOR T=1 TO 200
50 ST=A(T)*SIN(N*2*PI*T/200)
60 S=S+ST
70 CT=A(T)*COS(N*2*PI*T/200)
80 C=C+CT
90 NEXT T
```

```

100 S=S/100:C=C/100
102 IF INT(N/2)-N/2 <> 0 THEN SE=4/(PI#N):SE=INT
T(SE#1E4)/1E4 ELSE SE=0
105 S=INT(S*10000)/10000:C=INT(C*10000)/10000
110 PRINTN,S:C
116 PRINT " ",SE,0
118 S(N)=S:C(N)=C:E(N)=SE
120 NEXT N
130 GOTO1900
200 REM EXEMPLE
210 FOR U=0 TO 100:A(U)=1:NEXT
220 FOR U=101 TO 200:A(U)=-1:NEXT
230 RETURN
1000 REM axe vertical continu
1010 LPRINT CHR$(#08);CHR$(#10);"15";CHR$(#10);
CHR$(1);CHR$(#FF);
1020 LPRINT CHR$(#10);"35";CHR$(#10);CHR$(1);CH
R$(#FF);
1030 LPRINT CHR$(#10);"55";CHR$(#10);CHR$(1);CH
R$(#FF);
1040 LPRINT CHR$(#0F);:RETURN
1100 REM axe vertical discontinu
1110 LPRINT CHR$(#08);CHR$(#10);"25";CHR$(#10);
CHR$(1);CHR$(179);
1120 LPRINT CHR$(#10);"45";CHR$(#10);CHR$(1);CH
R$(179);
1130 LPRINT CHR$(#0F);:RETURN
1200 REM axe horizontal continu
1210 LPRINT CHR$(#08);CHR$(#10);"00";CHR$(#10);
CHR$(#FF);CHR$(136);
1220 LPRINT CHR$(#10);CHR$(75);CHR$(136);
1230 LPRINT CHR$(#0F);:RETURN
1300 REM axe horizontal discontinu
1310 FOR N = 0 TO 28
1320 LPRINT CHR$(#2D);CHR$(#20);
1330 NEXT N
1340 RETURN
1900 POKE49,255
2000 LPRINT CHR$(#10);"18 TERMES EN SINUS";CHR$(
#10);"36 TERMES EN COSINUS";
2010 GOSUB1000:LPRINT CHR$(#0D)
2020 GOSUB 1200:LPRINT CHR$(#0D)
2030 LPRINT CHR$(#10);"05RANG DE";CHR$(#10);"17
VALEUR";CHR$(#10);"27VALEUR";
2040 LPRINT CHR$(#10);"37VALEUR";CHR$(#10);"47V
ALEUR";

```

```

2050 GOSUB 1000:GOSUB1100:LPRINT CHR$(#00)
2060 LPRINT CHR$(#10);"02L'HARMONIQUE";CHR$(#10)
);"16CALCULEE";
2070 LPRINT CHR$(#10);"27EXACTE";CHR$(#10);"36C
ALCULEE";
2080 LPRINT CHR$(#10);"36CALCULEE";CHR$(#10);"4
7EXACTE";
2090 GOSUB1000:GOSUB1100:LPRINT CHR$(#00)
2100 GOSUB1200:LPRINT CHR$(#00)
2110 FOR N=1 TO 30
2120 LPRINT CHR$(#10);"10";N;CHR$(#10);"18";S(N
);CHR$(#10);"28";E(N);
2130 LPRINT CHR$(#10);"38";C(N);CHR$(#10);"480"
);
2140 GOSUB 1000:GOSUB1100:LPRINT CHR$(#00):GOSU
B1000:GOSUB1100:LPRINT
2150 NEXTN
2160 END
10000 CALL #E6CA
10010 LLIST

```

TERMES EN SINUS | TERMES EN COSINUS||

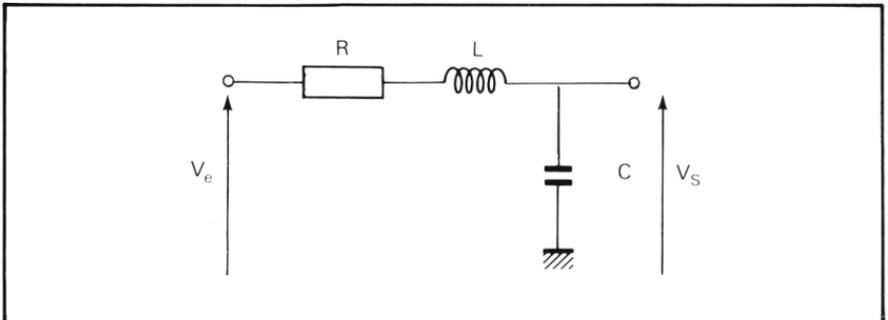
RANG DE L'HARMONIQUE	VALEUR CALCULEE	VALEUR EXACTE	VALEUR CALCULEE	VALEUR EXACTE
1	1.2731	1.2732	-.02	0
2	0	0	0	0
3	.424	.4244	-.02	0
4	0	0	0	0
5	.2541	.2546	-.02	0
6	-1E-04	0	-1E-04	0
7	.1811	.1818	-.02	0
8	0	0	0	0
9	.1405	.1414	-.02	0
10	0	0	-1E-04	0
11	.1145	.1157	-.02	0

12	0	0	0	0
13	.0965	.0979	-.02	0
14	0	0	-1E-04	0
15	.0833	.0848	-.02	0
16	0	0	-1E-04	0
17	.0731	.0748	-.02	0
18	-1E-04	0	0	0
19	.065	.067	-.02	0
20	-1E-04	0	0	0
21	.0584	.0606	-.02	0
22	-1E-04	0	-1E-04	0
23	.0529	.0553	-.02	0
24	-1E-04	0	0	0
25	.0482	.0509	-.02	0
26	0	0	-1E-04	0
27	.0442	.0471	-.02	0
28	0	0	0	0
29	.0408	.0439	-.02	0
30	-1E04	0	-1E-04	0

UTILISATION DE LA SERIE DE FOURIER

Le programme précédent a montré la méthode qui permet de trouver le régime permanent d'un circuit dans le cas d'une sollicitation quelconque, dans la mesure où l'on connaît à la fois le développement en série de Fourier de la sollicitation et la réponse en fréquence (module et phase) du circuit considéré. Nous allons utiliser cette méthode pour trouver la réponse d'un filtre passe bas du second ordre à un échelon (problème traité par ailleurs dans cet ouvrage, puisque dans ce cas particulier, on connaît l'équation différentielle qui régit le fonctionnement d'un tel circuit).

Le circuit étudié est donc le suivant :



$$\text{On a } \bar{T} = \frac{\bar{V}_s}{\bar{V}_e} = \frac{1}{1 + 2im \frac{\omega}{\omega_0} + \left(i \frac{\omega}{\omega_0}\right)^2}$$

$$\text{avec } Q = \frac{1}{2m} = \frac{L\omega_0}{R} \text{ et } \omega_0 = \frac{1}{\sqrt{LC}}$$

c'est-à-dire, en module et en phase :

$$|\bar{T}_\omega| = \frac{1}{\sqrt{\left(1 - \frac{\omega^2}{\omega_0^2}\right)^2 + \left(2m \frac{\omega}{\omega_0}\right)^2}}$$

$$; \varphi_{\omega} = \text{Arg} (T) = - \text{Arc tg} \frac{2 m \omega / \omega_0}{1 - \omega^2 / \omega_0^2}$$

Chaque terme du développement en série de Fourier de V_e , de la forme

$$\frac{1}{K} \sin K \omega t \text{ devient donc, en sortie : } \frac{1}{K} |T_{\omega}| \times \sin (K\omega t - \varphi_{\omega}).$$

La recomposition de la série de Fourier, après traitement de chacun de ses termes, donne la série de Fourier relative à la tension de sortie.

LE PROGRAMME

On a posé $\frac{\omega_F}{\omega_0} = \frac{1}{4}$, en appelant ω_F la pulsation du terme fondamental. Ainsi, pour chaque harmonique de rang K ,

$$\frac{\omega}{\omega_0} = \frac{K\omega_F}{\omega_0} = \frac{K}{4}.$$

Nous étudions ainsi un signal rectangulaire, dont la fréquence est 4 fois plus petite que la fréquence propre du circuit. D'autre part, pour

le fondamental ($K = 1$), on a $\omega_F t = \frac{\pi t}{200}$, ce qui fait que 400 points sur l'écran (on n'en peut dessiner bien sûr que 239 pour ORIC) correspondent à 1 période.

Ligne 10 : définition du coefficient d'amortissement m ($m = 0,4$).

Lignes 40 à 70 : évaluation de $|T_{k\omega}|$ et de $\varphi_{k\omega}$.

Ligne 90 : calcul de l'harmonique de rang K .

Ligne 100 : calcul de V_s , à l'instant t .

VARIABLES

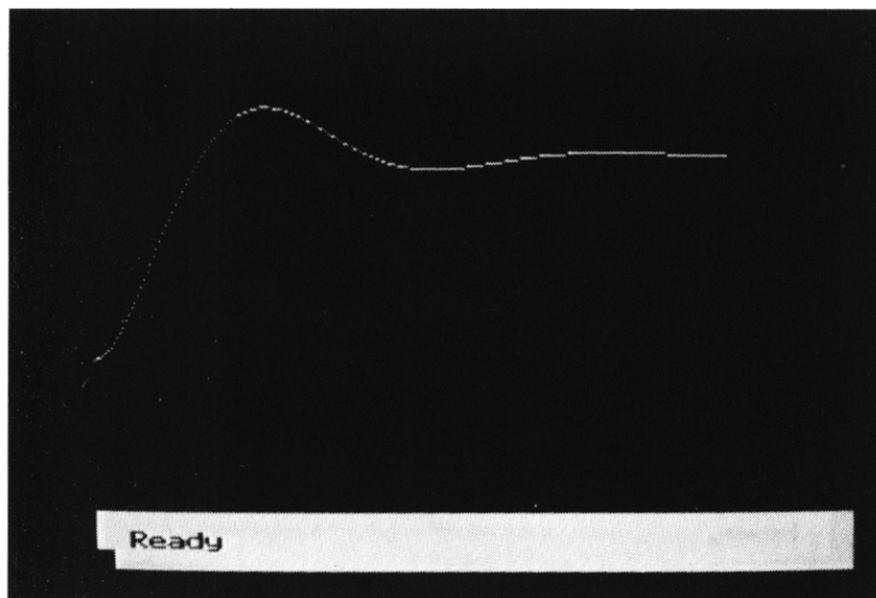
* m : coefficient d'amortissement du circuit

* T : variable temps

* K : rang de l'harmonique considéré

* A : valeur de $1 - \frac{\omega^2}{\omega_0^2}$

- * B : valeur de $2m \frac{\omega}{\omega_0}$, pour le rang K
- * AK : inverse de l'amplitude de l'harmonique de rang K
- * PH : valeur de $\varphi_{K\omega}$
- * YK : valeur de l'harmonique de rang K au temps t
- * Y : somme des harmoniques

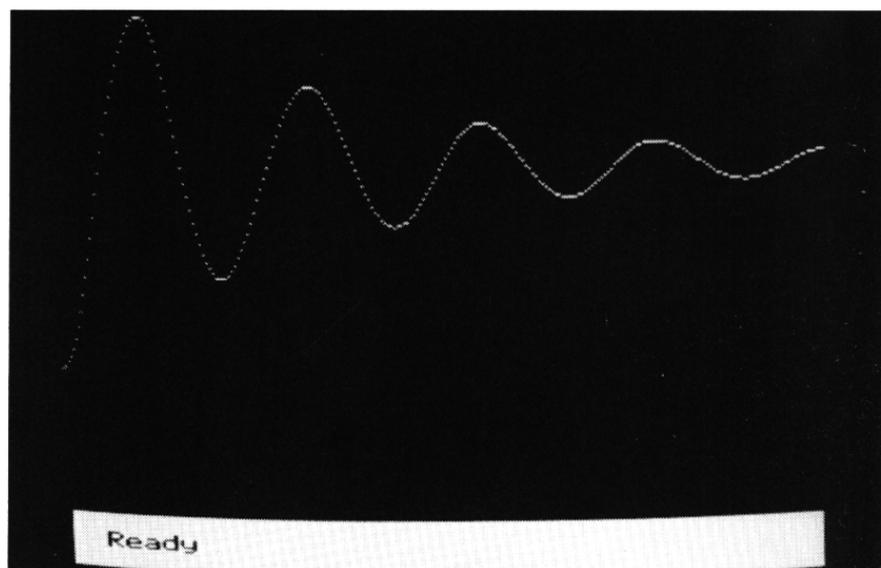


Pour $M = 0,4$ il y a un léger dépassement

```

1 REM*****
2 REM*                                     *
3 REM* UTILISATION *
4 REM* DE LA SERIE *
5 REM* DE FOURIER *
6 REM*                                     *
7 REM*****
10 HIRES:M=0.4
20 FORT=0T0200
30 FORK=1 TO 51 STEP2
40 A=(1-(K^2)/16)
50 B=M*K/2
60 AK=K*((A^2+B^2)^.5)
70 PH=ATN(B/A)
80 IF PH < 0 THEN PH=PI+PH
90 YK=(1/AK)*SIN((K*PI*T/200)-PH)
100 Y=Y+YK
110 NEXTK
120 CURSETT,150-50*Y,1
130 Y=0
140 NEXTT
150 END

```



On constate un dépassement de 0,75, ce qui est très proche de la valeur théorique 0,72

```

1 REM*****
2 REM*
3 REM* UTILISATION *
4 REM* DE LA SERIE *
5 REM* DE FOURIER *
6 REM*****
10 HIRES:M=0.1
20 FORT=0T0239
30 FORK=1T051STEP2
40 A=(1-(K^2)/225)
50 B=M*K/7.5
60 AK=K*((A^2+B^2)^0.5)
65 IF A = 0 THEN PH=PI/2:GOTO80
70 PH=ATN(B/A)
80 IF PH < 0 THEN PH=PI+PH
90 YK=(1/AK)*SIN((K*PI*T/400)-PH)
100 Y=Y+YK
110 NEXTK
120 CURSETT,100-50*Y,1
130 Y=0
140 NEXTT
150 END

```

Chapitre 4

REGIMES TRANSITOIRES

REGIMES TRANSITOIRES DU PREMIER ORDRE

Le programme propose une étude systématique des circuits RC du premier ordre, pouvant être soumis à différentes sollicitations (signaux triangulaires, rectangulaires et sinusoïdaux). Le régime étudié est donc le régime transitoire de ces circuits. En laissant le programme se dérouler un certain temps, on voit apparaître le régime permanent. La gamme des signaux proposés n'est pas limitative. On peut en utiliser d'autres qui seront définis, soit sous forme de fonction s'ils ont une expression analytique $FNA(T)$, soit sous forme d'éléments de tableau $A(T)$. Le problème est résolu par l'intégration numérique de l'équation différentielle. L'étude de tout circuit défini par le même genre d'équation se fera donc de manière identique.

Le programme se présente sous forme de menu. Il faut d'abord choisir le circuit dont on veut étudier le fonctionnement (tableau 1), puis la forme du signal (tableau 2), enfin leurs caractéristiques (tableau 3). Ensuite, l'écran montre le signal d'entrée choisi et le signal de sortie correspondant. On peut obtenir simultanément sur l'imprimante ces deux courbes en « continu », alors que sur l'écran, les périodes des signaux se superposent, montrant ainsi le passage du régime transitoire au régime permanent. Il faut arrêter le programme par un **BREAK** (control C). Afin de permettre le dessin des circuits sur l'écran **TEXT**, certains caractères (peu usités) sont redéfinis. Pour réaliser la copie sur l'imprimante de cet écran **TEXT**, le programme propose (lignes 2 000 à 2 110) de recopier cet écran en mode **HIRES** ; en effet, la communication entre l'ordinateur et l'imprimante se fait par le code **ASC II** des caractères, et non par leur définition. Une recopie directe de l'écran texte ne transmettrait donc que des caractères, et non les dessins correspondants.

LE PROGRAMME :

ORGANISATION DU PROGRAMME

10	}	Redéfinition caractères
70		
100	}	Premier tableau : choix du circuit
190		

LES VARIABLES (dans l'ordre du programme).

U,N : variables d'exploration des caractères redéfinis.

A\$: caractère redéfini.

A : l'une des 8 nouvelles définitions du caractère A\$.

X : positionnement des dessins sur l'écran.

N : numéro du circuit choisi.

K : numéro du signal choisi.

R, R₁, R₂, C : valeurs numériques des éléments du circuit choisi.

F : fréquence du signal d'entrée.

V ϕ : tension initiale (à t = 0) sur le condensateur.

V : tension d'entrée (en V) crête à crête (elle est supposée sans composante continue).

DT : intervalle de calcul, pris égal à 1/(200 F). L'écran complet contient donc une période du signal d'entrée.

T : variable temps (il y a 200 intervalles sur une période).

L : variable imposée par la discontinuité des signaux considérés : L = 0 pour la première demi-période, L = 1 pour la seconde demi-période.

A,D : variables de position relatives à V_e (tension d'entrée : valeur crête) et à V_s (tension de sortie) pour l'imprimante.

S : variable de septet pour l'imprimante (voir annexe).

DV : variation de la tension aux bornes du condensateur pendant l'intervalle de temps DT.

V_c : tension aux bornes du condensateur à l'instant T.

FNA (T) : fonction correspondant à la tension d'entrée.

Pour la partie du programme comprise entre 2 000 et 3 050, on se référera à l'annexe relative à l'imprimante.

```
1 REM*****
2 REM* *
3 REM* REGIMES TRANSITOIRES *
4 REM* DU PREMIER ORDRE *
5 REM* *
6 REM*****
8 CLS
9 PAPER0:INK3
10 REMDEFINITION CARACTERES
20 FOR U=0 TO 12
30 READ A$
40 FOR N=0 TO 7
50 READ A
```

```

60 POKE 46080+((8*ASC(A$))+N),A
70 NEXT N:NEXT U
80 POKE49,255
90 LPRINT CHR$(#08):S=0
100 CLS
110 X=48205:GOSUB1000
120 X=48214:GOSUB1010
130 X=48221:GOSUB1020
140 X=48232:GOSUB1030
150 X=48527:POKEX,177:POKEX+8 ,178:POKEX+16,179
:POKEX+26,180
160 PLOT1,20,0:PLOT2,20,22:PLOT11,20,"CHOISISSE
Z LE CIRCUIT"
170 PLOT1,21,0:PLOT2,21,22:PLOT10,21,"DONT VOUS
VOULEZ ETUDIER"
180 PLOT1,22,0:PLOT2,22,22:PLOT12,22,"LE FONCTI
ONNEMENT"
190 POKE616,25:POKE617,19:INPUTN:PLOT19,25,CHR$(
ASC(MID$(STR$(N),2))+128)
200 REM CHOIX DU SIGNAL
210 CLS
220 X=48141:ON N GOSUB 1000,1010,1020,1030
230 PLOT0,10,19:PLOT 1,10,4:PLOT8,10,"QUEL SIGN
AL VOULEZ VOUS "
240 PLOT 0,11,19:PLOT1,11,4:PLOT8,11,"APPLIQUER
A CE CIRCUIT ?"
245 A$(1)="SIGNAL RECTANGULAIRE"
250 PLOT 5,15,177:PLOT 7,15,A$(1)
255 A$(2)="SIGNAL TRIANGULAIRE"
260 PLOT5,17,178:PLOT 7,17,A$(2)
265 A$(3)="SIGNAL SINUSOIDAL"
270 PLOT5,19,179:PLOT 7,19,A$(3)
280 POKE617,18:POKE616,21:INPUT A$:K=VAL(A$)
290 PLOT 5,15+((K-1)*2),1
300 CLS
310 X=48109:ON N GOSUB 1000,1010,1020,1030
320 PLOT 30,6,LEFT$(A$(K),6):PLOT25,7,MID$(A$(K
),7)
325 PLOT0,13,12
330 PLOT 1,13,"CARACTERISTIQUES DU CIRCUIT ?":
340 IF N=1 OR N=2 THEN INPUT "R":GOTO360
350 INPUT "R1 =":R1:INPUT"R2 =":R2
355 R=R1+R2
360 INPUT"C=":C
370 PLOT 0,13,8
380 PLOT0,17,12:PLOT 2,17,"FREQUENCE DU SIGNAL
?"

```

```

390 INPUT "F =";F
395 PLOT0,17,8
400 PLOT 0,21,12:PLOT 2,21,"TENSION INITIALE SU
R C ?"
410 INPUT "V0=";V0
420 PLOT0,21,8
430 PLOT0,25,12:PLOT2,25,"TENSION D ENTREE V?"
440 INPUT "V=";V :PLOT0,25,8
500 DT=1/(200*F):HIRES: REM CALCULS
505 VC=V0
506 L=0
510 IF K=1 THEN DEF FNA(T)=1
520 IF K=3 THEN DEF FNA(T)=SIN(2*PI*T/100)
530 IF K=2 THEN DEF FNA(T)=-1+T/25
560 FOR T=0 TO 49
570 GOSUB 800:ON N GOSUB900,910,920,930
575 GOSUB 3000
580 NEXT
590 IF K=1 THEN DEF FNA(T)=-1
600 IF K=2 THEN DEF FNA(T)=3-T/25
610 FOR T=50 TO 99
620 GOSUB 800:ON N GOSUB900,910,920,930
625 GOSUB 3000
630 NEXT
640 L=L+1
650 IFL=1 THEN GOTO510
660 IFL=2 THEN CLS: GOTO506
800 VE=V*FNA(T)/2
805 D=FNA(T)*50
810 CURSETT+L*100, 50-40*FNA(T),1
820 DV=DT*(VE-VC)/(R*C)
830 VC=VC+DV
840 I=(VE-VC)/R
850 RETURN
900 CURSET T+L*100,50-40*VC/V,1:A=100*VC/V:RE TU
RN
910 CURSET T+L*100,50-40*R*I/V,1:A=100*R*I/V:RE
TURN
920 CURSET T+L*100,50-40*R2*I/V,1:A=100*R2*I/V:
RETURN
930 CURSET T+L*100,50-40*(R2*I+VC)/V,1:A=100*(R
2*I+VC)/V:RETURN
1000 REM CIRCUITS
1002 POKEX-39,82:POKEX,64:POKEX+1,37:POKEX+2,38
:POKEX+3,33:POKEX+43,93
1004 POKEX+44,94:POKEX+82,62:POKEX+83,60:POKEX+
84,123:POKEX+42,67

```

```

1006 RETURN
1010 POKE X-40,67:POKE X+43,82:POKE X,95:POKE X+1,3
3:POKE X+41,35:POKE X+81,91
1011 POKE X-1,125
1012 POKE X+120,62:POKE X+121,60:POKE X+122,123
1014 RETURN
1020 POKE X,64:POKE X+1,37:POKE X+2,38:POKE X+3,95:
POKE X+4,33:POKE X+44,35
1022 POKE X+84,91:POKE X+124,60:POKE X+123,62:POK
EX+125,123:POKE X-40,82
1024 POKE X-39,49:POKE X-37,67:POKE X+46,82:POKE X+
47,50
1026 RETURN
1030 POKE X,64:POKE X+1,37:POKE X+2,38:POKE X+3,33:
POKE X+43,35:POKE X+83,91
1032 POKE X+123,93:POKE X+124,94:POKE X+163,60:POK
EX+162,62:POKE X+164,123
1034 POKE X-40,82:POKE X-39,49:POKE X+45,82:POKE X+
46,50:POKE X+122,67
1036 RETURN
1100 DATA @,0,7,4,60,4,7,0,0,%,0,63,0,0,0,63,0,
0,&,0,56,0,15,0,56,0,0
1110 DATA _,18,18,18,51,18,18,18,0,!,0,0,0,60,4
,4,4,4,#,4,4,31,17,17,17,17,17
1120 DATA [,17,17,17,17,17,31,4,4,],4,4,63,0,0,
63,4,4,^,0,0,32,0,0,32,0,0
1130 DATA >,0,0,0,7,4,9,18,0,<,4,4,4,63,36,9,18
,0,(,0,0,0,56,32,0,0,0
1140 DATA ),0,0,0,63,0,0,0,0
2000 F=0:DIMA(1120)
2010 FORE=48040 TO 49119:A(F)=PEEK(E)
2020 IF PEEK(E) > 128 THEN A(F)=PEEK(E)-128
2025 IF A(F) < 32 THEN A(F)=32
2030 F=F+1:NEXT
2040 HIRES:RESTORE
2050 FORU=0 TO 12:READ A#
2060 FOR N=0 TO 7:READ A
2070 POKE 38912+((8*ASC(A#))+N),A
2080 NEXT N:NEXT U
2085 F=0
2090 FOR N=0 TO 199 STEP 8:FOR M=0 TO 239 STEP
6
2100 CURSETM,N,0
2101 IF M < 234 THEN CHARA(F),0,1
2110 F=F+1:NEXT M:NEXT N
2120 FORN=0 TO 6:B(N)=2^N:NEXT

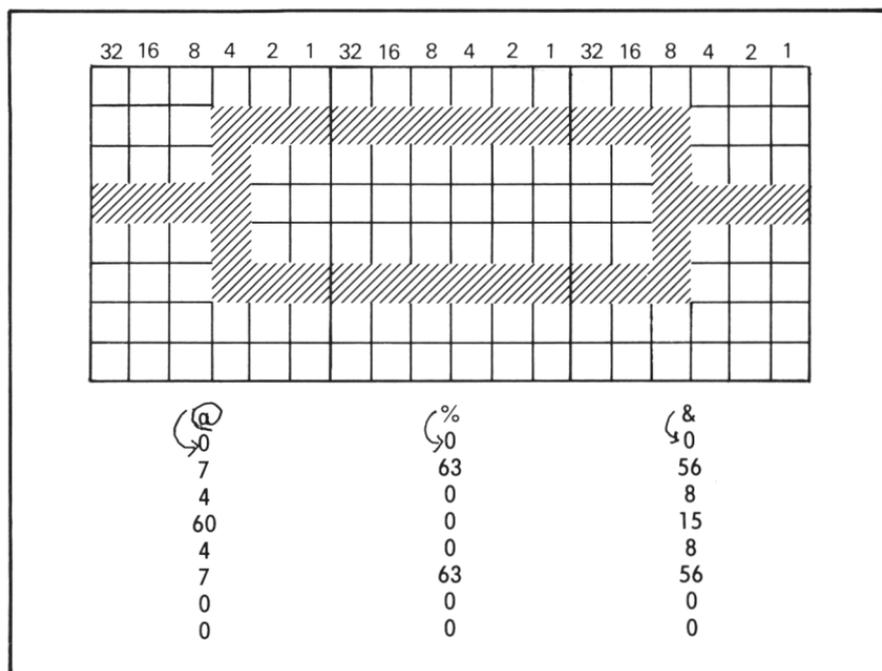
```

```

2130 POKE49,255:LPRINT CHR$(#00)
2135 FORN=1TO239:LPRINT CHR$(129);:NEXT:LPRINT
CHR$(#00)
2140 FOR Y=7 TO 199 STEP7:LPRINT CHR$(#FF);
2150 FOR X=3 TO 239
2160 A1=0:FOR A=0 TO 6
2170 Y1=Y+A:K=POINT(X,Y1)
2180 IFK=0 THEN 2200
2190 A1=A1+K*B(A)
2200 NEXTA
2210 A1=128+ABS(A1)
2220 LPRINT CHR$(A1);
2230 NEXT X:LPRINT CHR$(#FF);
2240 LPRINT CHR$(#00)
2250 NEXT Y
2260 END
3000 REM Impression des courbes sur l'imprimant
e
3010 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(A+
120);CHR$(128+2^S);
3020 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(D+
120);CHR$(128+2^S);
3030 S=S+1
3040 IF S = 6 THEN LPRINT CHR$(#1B);CHR$(#10);C
HR$(0);CHR$(120);
3050 IF S = 6 THEN LPRINT CHR$(201);S=0
3060 RETURN
10000 CALL #E6CA
10010 LLIST

```

Exemple de redéfinition de caractère : dessin d'une résistance horizontale.



Le code ASCII de @ est 64.

Sa définition mémoire est située de $46\ 080 + (8 \times 64)$ à $46\ 080 + (8 \times 64) + 7$, mémoires dans lesquelles il faut « paker » successivement 0, 7, 4, 60, ...

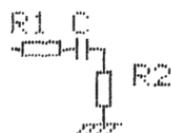
En mode HIRIS, 46 080 doit être remplacé par 38 912.

RECOPIES D'ECRAN

1 2 3 4

CHOISISSEZ LE CIRCUIT
DONT VOUS VOULEZ ETUDIER
LE FONCTIONNEMENT

Tableau 1 : Choix du circuit (1), (2), (3) ou (4)



QUEL SIGNAL VOULEZ VOUS
APPLIQUER À CE CIRCUIT ?

- 1 SIGNAL RECTANGULAIRE
- 2 SIGNAL TRIANGULAIRE
- 3 SIGNAL SINUSOIDAL

Tableau 2 : Choix du signal

R2 =? 2E4
 C=? 3E-8
 F=? 2E4
 U0=? 0
 U=? 5



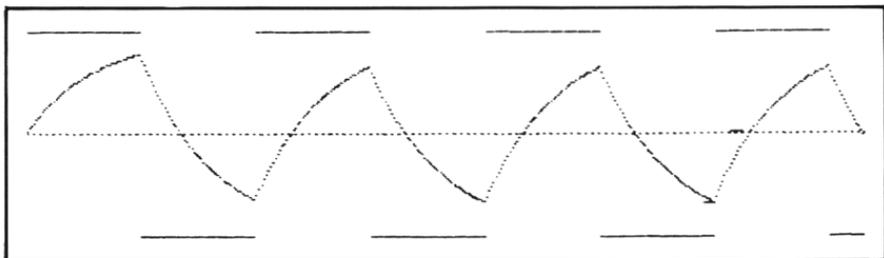
SIGNAL
TRIANGULAIRE

CARACTERISTIQUES DU CIRCUIT ?

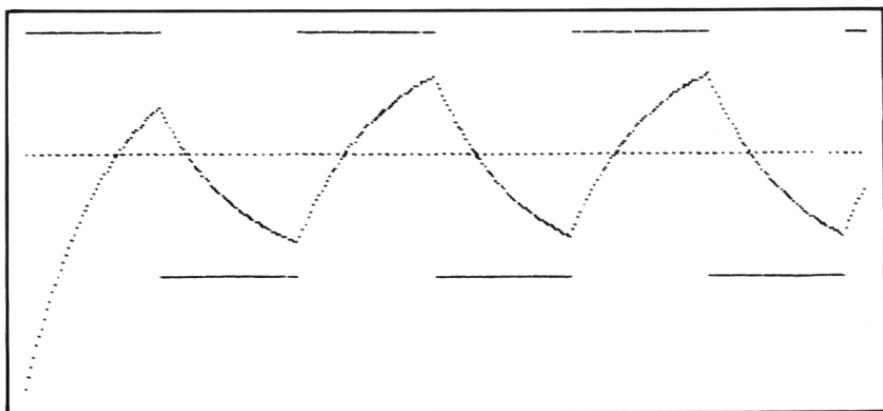
FREQUENCE DU SIGNAL ?

TENSION INITIALE SUR C ?

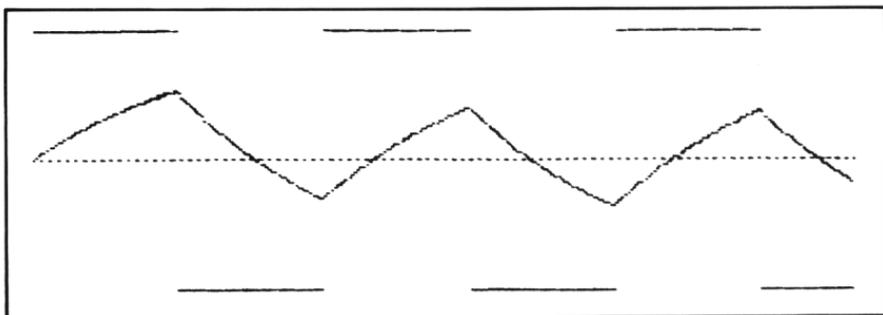
Tableau 3 : Introduction des valeurs numériques



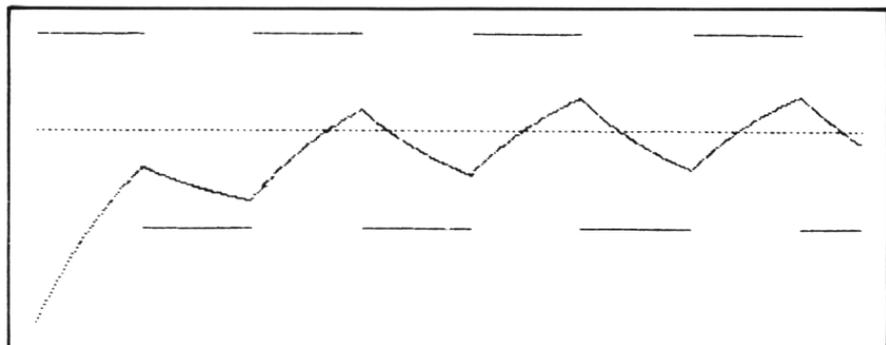
1 $R = 10 \text{ k}\Omega$ $V_o = 0 \text{ V}$
 $C = 10 \text{ nF}$ $V = 10 \text{ V}$ (crête à crête)
 $f = 1,6 \text{ kHz}$ Signal d'entrée rectangulaire



2 idem que 1 , mais $V_o = -10 \text{ V}$

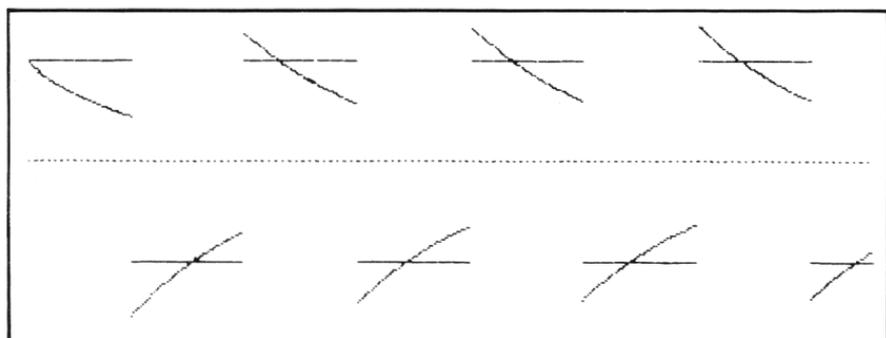


3 $R = 10 \text{ k}\Omega$
 $C = 10 \text{ nF}$
 $f = 3,2 \text{ kHz}$
 $V_o = 0 \text{ V}$
 $V = 10 \text{ V}$ (c. à c.)



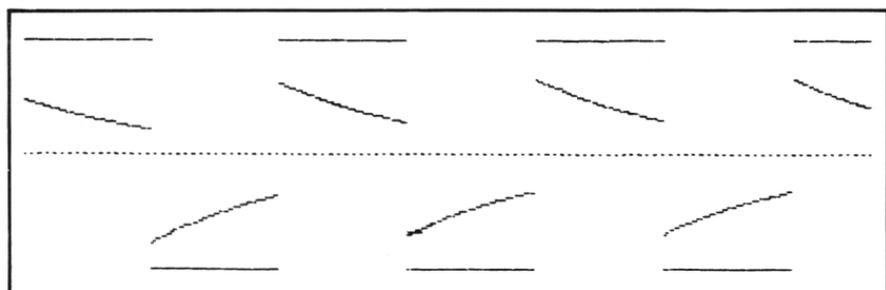
4 $R = 10 \text{ k}\Omega$
 $C = 10 \text{ nF}$
 $f = 6 \text{ kHz}$

$V_o = -10 \text{ V}$
 $V = 10 \text{ V (c. à c.)}$



5 Circuit (2)
 Signal rectangulaire

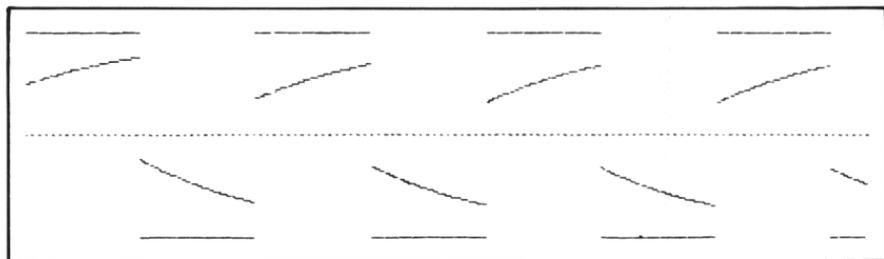
$R = 10 \text{ k}\Omega$
 $C = 10 \text{ nF}$
 $f = 3,2 \text{ kHz}$
 $V_o = 0 \text{ V et } V = 10 \text{ V}$



6 Circuit (3)
 Signal rectangulaire

$R_1 = 5 \text{ k}\Omega$
 $R_2 = 5 \text{ k}\Omega$
 $C = 10 \text{ nF}$

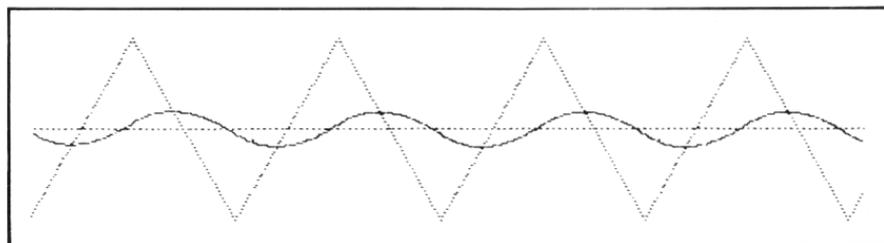
$f = 3,2 \text{ kHz}$
 $V_o = 0 \text{ V et } V = 10 \text{ V}$



7 Circuit (4)
Signal rectangulaire

$R_1 = 5 \text{ k}\Omega$
 $R_2 = 5 \text{ k}\Omega$
 $C = 10 \text{ nF}$

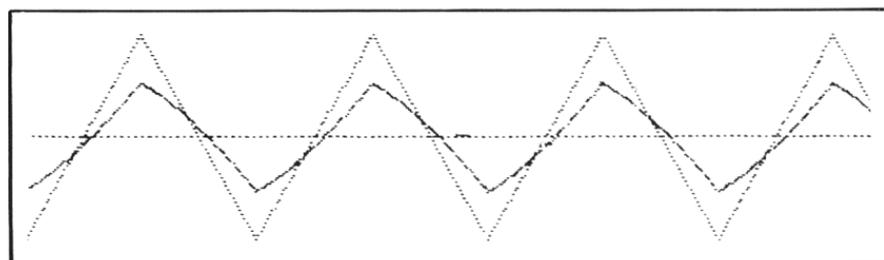
$f = 3,2 \text{ kHz}$
 $V_o = 0 \text{ V}$ et $V = 10 \text{ V}$



8 Circuit (1)
Signal triangulaire

$R = 10 \text{ k}\Omega$
 $C = 10 \text{ nF}$
 $f = 3,2 \text{ kHz}$

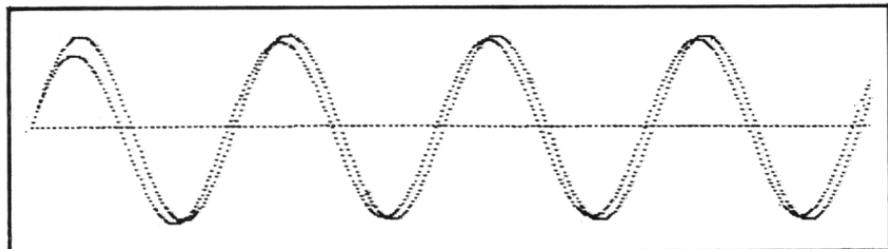
$V_o = 0 \text{ V}$
 $V = 10 \text{ V}$



9 Circuit (4)
Signal triangulaire

$R_1 = 5 \text{ k}\Omega$
 $R_2 = 5 \text{ k}\Omega$
 $C = 10 \text{ nF}$

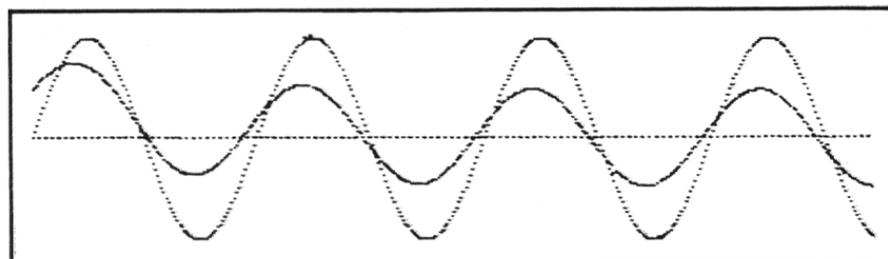
$f = 3 \text{ kHz}, 2$
 $V_o = 0 \text{ V}$
 $V = 10 \text{ V}$



10 Circuit (2)
Signal sinusoïdal

$R = 10 \text{ k}\Omega$
 $C = 10 \text{ nF}$
 $f = 3,2 \text{ kHz}$

$V_o = 0$
 $V = 10 \text{ V}$



11 Circuit (3)
Signal sinusoïdal

$R_1 = 5 \text{ k}\Omega$
 $R_2 = 5 \text{ k}\Omega$
 $C = 10 \text{ nF}$

$V_o = 5 \text{ V}$
 $V = 10 \text{ V}$

Exemple n° 1

CAPS

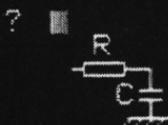
1

2

3

4

CHOISISSEZ LE CIRCUIT
DONT VOUS VOULEZ ETUDIER
LE FONCTIONNEMENT



QUEL SIGNAL VOULEZ VOUS
APPLIQUER A CE CIRCUIT ?

- SIGNAL RECTANGULAIRE
- SIGNAL TRIANGULAIRE
- SIGNAL SINUSOIDAL

R? 1E4
C=? 1E-8
F=? 1E3
UG=? 10
V=? 10 ■



SIGNAL
RECTANGULAIRE

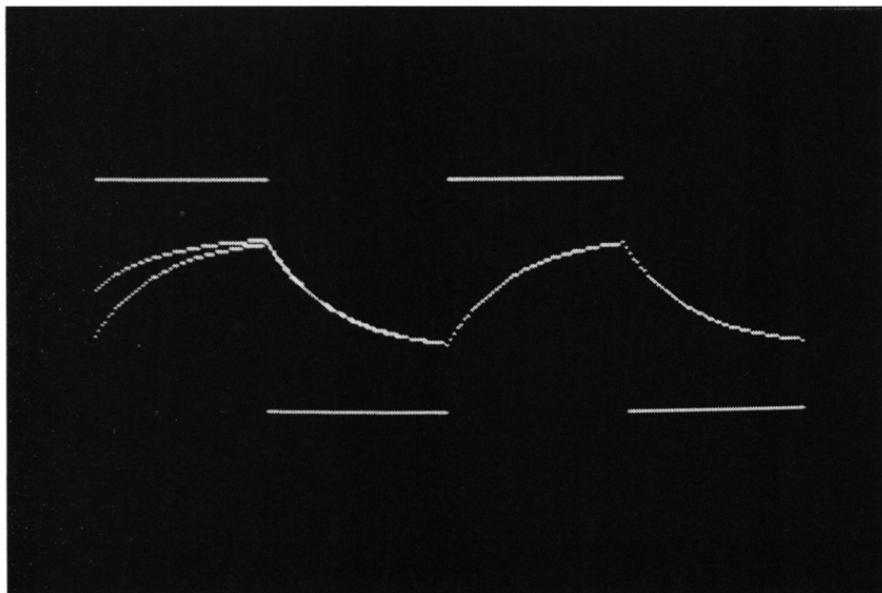
CARACTERISTIQUES DU CIRCUIT ?

FREQUENCE DU SIGNAL ?

TENSION INITIALE SUR C ?

TENSION D ENTREE U?

Exemple n° 1 (fin)



Exemple n° 2

7 4 ■

CAPS

■ ■ ■ ■

CHOISISSEZ LE CIRCUIT
DONT VOUS VOULEZ ETUDIER
LE FONCTIONNEMENT

CAPS



QUEL SIGNAL VOULEZ VOUS
APPLIQUER A CE CIRCUIT ?

- SIGNAL RECTANGULAIRE
- SIGNAL TRIANGULAIRE
- SIGNAL SINUSOIDAL

```
R1 =? 1E4
R2 =? 1000
C =? 1E-5
UG =? -5
U =? 10
```

CAPS

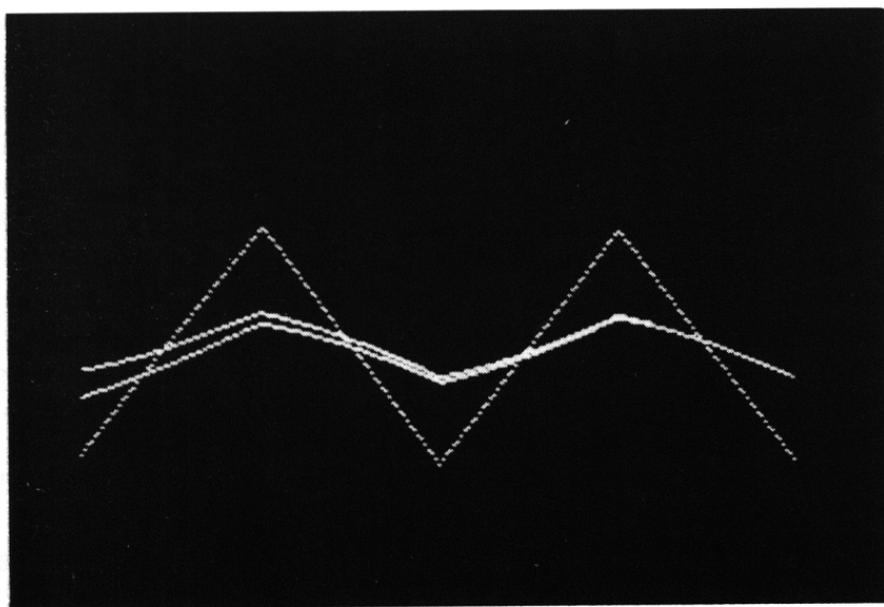
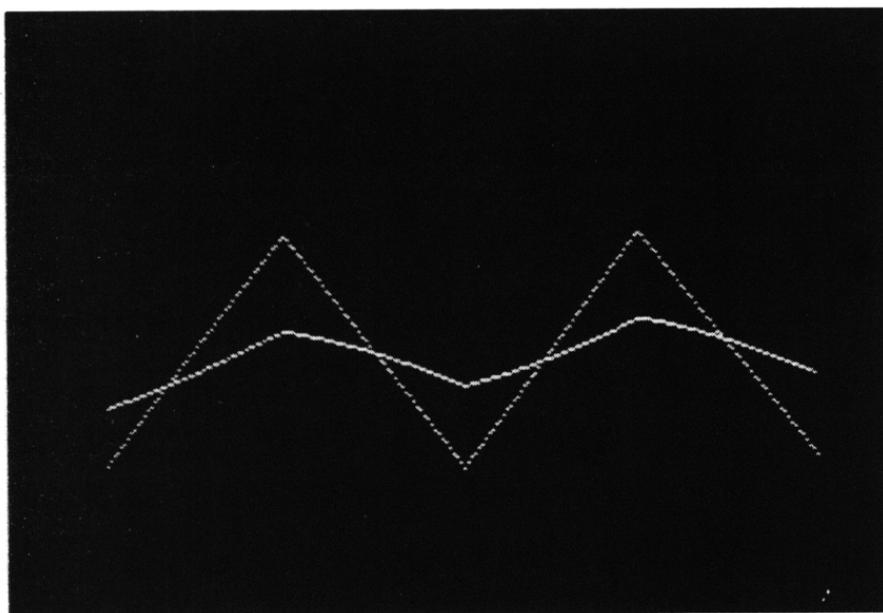


CARACTERISTIQUES DU CIRCUIT ?

FREQUENCE DU SIGNAL ?

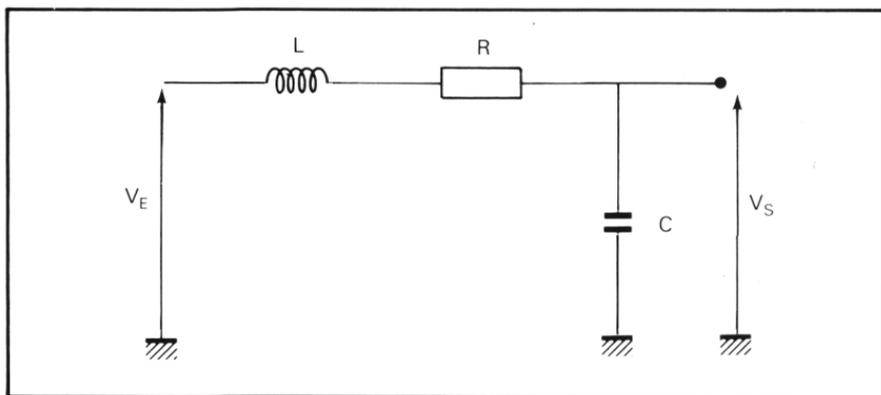
TENSION INITIALE SUR C ?

Exemple n° 2 (fin)



RÉGIMES TRANSITOIRES DU SECOND ORDRE

Ces régimes sont importants en électronique. Ils ont lieu dans tous les circuits du type filtre passe-bas, dont le schéma de principe est le suivant :



$$\text{On a } \bar{T} = \frac{V_S}{V_E} = \frac{1}{1 + 2jm \omega / \omega_0 + (j\omega / \omega_0)^2}$$

$$\text{avec ici } 2m = RC \omega_0 \quad \omega_0 = \frac{1}{\sqrt{LC}}$$

m et ω_0 sont des constantes « universelles » dans la mesure où tous les circuits ayant mêmes valeurs de m et de ω_0 présenteront, pour une sollicitation donnée, la même forme de réponse. L'étude de ces régimes transitoires, si elle connaît une solution analytique, est assez délicate. On se contente en général, de tracer quelques points donnés par des courbes universelles.

L'équation différentielle qui régit le fonctionnement de tels circuits est la suivante :

$$V_E = V_S + \frac{2m}{\omega_0} \cdot \frac{dV_S}{dt} + \frac{1}{\omega_0^2} \cdot \frac{d^2V_S}{dt^2}, \text{ dans laquelle } m \text{ est le coefficient}$$

d'amortissement et ω_0 la pulsation propre. Le problème est résolu de la façon suivante :

(1) à l'instant t , introduction de V_E , et des conditions initiales $V_S, \frac{dV_S}{dt}$

(2) on en déduit : $\frac{d^2V_S}{dt^2} = \omega_0^2 (V_E - V_S - \frac{2m}{\omega_0} \times \frac{dV_S}{dt})$

(3) soit $d \left(\frac{dV_S}{dt} \right) = \frac{d^2V_S}{dt^2} \times dt$

(4) puis $\frac{dV_S}{dt} = (\text{ancienne valeur de } \frac{dV_S}{dt}) + d \left(\frac{dV_S}{dt} \right)$

(5) enfin $V_S = (\text{ancienne valeur de } V_S) + \left(\frac{dV_S}{dt} \times dt \right)$

(6) Ces 2 nouvelles valeurs de V_S et de $\frac{dV_S}{dt}$ sont alors introduites dans

(2) pour permettre une nouvelle étape de calcul.

Remarquons qu'au départ, il faut introduire des conditions initiales (V_S et $\frac{dV_S}{dt}$ à l'instant $t = 0$) pour permettre le démarrage du calcul.

La précision des résultats obtenus est liée au choix de l'intervalle de calcul dt . Il faut que dt soit « petit » devant $2\pi/\omega_0$ (période propre du circuit) et que d'autre part V_E varie assez peu pendant le temps dt . Nous proposons l'étude de ces circuits pour des tensions d'entrée du type : échelon, impulsion, signal rectangulaire. Celles-ci pourront être remplacées par d'autres fonctions.

LE PROGRAMME N° 1

Il permet d'observer la forme des courbes correspondant à la tension de sortie du circuit, ceci pour différentes valeurs de m qu'il est facile de modifier à la ligne 40. Il est conçu pour un signal d'entrée ayant la forme d'un échelon (ligne 70). Quelques petites modifications indiquées sous les figures permettent d'accéder à l'impulsion et au signal rectangulaire.

* Ligne 10 : initialisation de l'imprimante.

* Lignes 20 à 100 : définition des constantes.

* Lignes 110 à 170 : calculs.

* Lignes 180 à 230 : tracé des courbes correspondant à V_E et V_S (l'usage du paramètre $FB = 2$ lignes 180 et 190 permet d'effacer

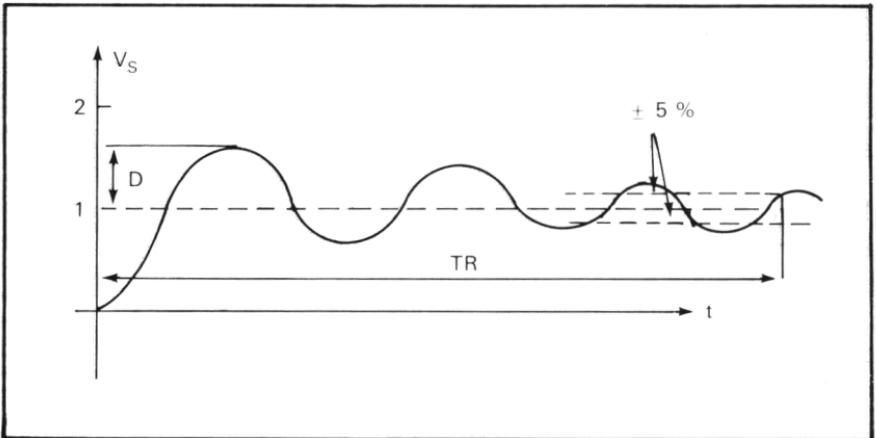
successivement tous les points de l'ancienne courbe au profit de la nouvelle).

- * SUB 260 : tracé des points sur l'imprimante.
- * SUB 330 : envoi de l'ordre d'impression à l'imprimante.
- * SUB 370 : graduations de l'axe OX pour l'imprimante.

Le programme se déroule de façon continue ; il permet ainsi de visualiser le passage progressif du régime transitoire au régime permanent. Arrivée dans la partie droite de l'écran, la courbe repart sur sa partie gauche en effaçant progressivement l'ancien tracé.

LE PROGRAMME N° 2

Il reprend les bases du précédent, mais fournit cette fois-ci des résultats chiffrés. Il y a 2 grandeurs qui caractérisent un régime de second ordre : ce sont le dépassement D et le temps de réponse à 5 % : T_R .



Définition du dépassement D et du temps de réponse à 5 % d'un système du second ordre.

S'il existe une relation donnant D en fonction de m (on a $D =$

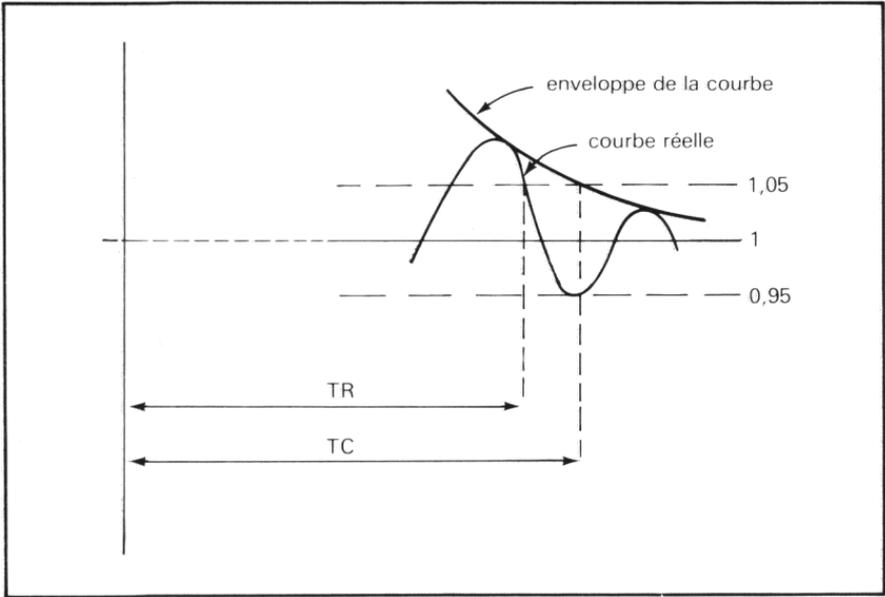
$\exp\left(\frac{-m\pi}{\sqrt{1-m^2}}\right)$ pour $m \leq 1$; il n'y a pas de dépassement pour

$m > 1$); il n'en existe pas pour T_R . En effet, à cause de l'aspect sinusoïdal de V_S , il peut exister des discontinuités pour T_R quand m varie.

Une formule approchée donnerait $T_R = \frac{3}{m\omega_0}$

(relation valable pour $m < \frac{\sqrt{2}}{2} = 0,707$). Ici, l'étude numé-

rique trouve tout son intérêt. Remarquons que le temps calculé TR est toujours plus petit que celui, TC, donné par la relation précédente. (TC ne s'intéresse qu'à l'enveloppe de la courbe).



Le temps de réponse TR calculé par programme est toujours plus petit que TC, donné par la relation mathématique.

- * Ligne 12 : mise en page des résultats sur l'imprimante.
- * Lignes 15 à 80 : initialisation des constantes et des variables.
- * Lignes 90 à 140 : intégration de l'équation différentielle.
- * Lignes 200 à 280 : calcul du dépassement (cela revient à chercher la plus grande des valeurs de V_s). Le résultat est arrondi à 2 chiffres.
- * Lignes 300 à 380 : recherche du temps de réponse (il faut chercher, en partant des plus grandes valeurs de V_s , le temps qui correspond à 2 valeurs successives de V_s , dont l'une est supérieure à 1,05 et l'autre inférieure — ou bien le contraire pour 0,95).
- * Ligne 385 : impression des résultats sur l'imprimante.
- * Lignes 390-400 : calcul de la nouvelle valeur de m (augmentée de 0,05) et retour en début de programme.
- * SUB 500 : mise en page sur l'imprimante.
- * SUB 600 : impression des données.

LES VARIABLES

PROGRAMME 1

- * V_0 : valeur de la tension aux bornes de C.
- * D_0 : valeur de $\left(\frac{dV_s}{dt} \right)$ (au temps t).
- * V_S : variation de tension aux bornes du condensateur à l'instant t.
- * D_1 : variation de $\left(\frac{dV_s}{dt} \right)$ à l'instant t.
- * V_E : amplitude de la tension d'entrée.
- * M : coefficient d'amortissement du système étudié.
- * F_0 : fréquence propre du circuit étudié.
- * W_2 : pulsation propre (au carré).
- * W_0 : pulsation propre.
- * $A(200)$: matrice contenant la fonction d'entrée.
- * $B(200)$: matrice contenant la fonction de sortie.
- * N : variable d'incrément, pour la définition de la fonction d'entrée.
- * K : constante de valeur $2 M/W_0$.
- * DT : intervalle de calcul, pris ici égal au 50^e de la période propre du circuit.
- * N : variable d'incrément pour l'imprimante.
- * D_2 : valeur de $\left(\frac{d^2V_s}{dt^2} \right)$ au temps t.
- * A : variable d'incrément pour les graduations de l'axe Ot sur l'imprimante (ces graduations seront espacées de $T_0 =$ période propre du circuit).
- * M : mémoire tampon temporaire pour A.

PROGRAMME 2

Les mêmes, sauf :

$M\%$, $M,N\%$: variables inélégantes, permettant seulement d'éviter au microordinateur employé (ORIC) d'écrire (ce qui est gênant pour la mise en page sur l'imprimante) : 0,80000001 au lieu de 0,8. Le programme initial prévoyait $M = M + 0,05$.

D : mémoire temporaire durant la recherche du maximum de V_s .

DP : valeur du dépassement, arrondi à 2 chiffres.

DR : valeur exacte du dépassement, arrondi à 2 chiffres.

TR : temps de réponse à 5 % calculé.

TC : valeur approchée du temps de réponse, donnée par la formule.

```
1 REM*****
2 REM*
3 REM* REponse D'UN CIRCUIT *
4 REM* DU SECOND ORDRE *
5 REM*
6 REM*****
10 LPRINTCHR$(#08):POKE49,255
20 REM CONSTANTES DU PROGRAMME
30 V0=0:D0=0:VS=V0:D1=D0:VE=1
40 M=0.1:F0=1E4
50 W2=(2*PI*F0)^2:W0=2*PI*F0
60 DIMA(200):DIMB(200)
70 FORN=0TO200:AC(N)=1:NEXT
80 K=2*M/W0
90 DT=1/(50*F0)
100 N=0
110 HIRES
120 REM CALCULS
130 FORU=0TO200
140 VE=AC(U)
150 D2=W2*(VE-V0-K*D0)
160 D1=D2*DT:D0=D0+D1
170 VS=D0*DT:V0=V0+VS
180 CURSETU,99-49*B(U),2
190 B(U)=V0:CURSETU,99-49*V0,2
200 A=A+1:IF (INT(A/50)-A/50) = 0 THEN GOSUB
370
210 IF VE > 2 THEN 240
220 GOSUB 260
230 CURSETU,99-49*VE,1
240 NEXT U
250 GOTO 130
260 REM IMPRIMANTE
270 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(1
20*V0);CHR$(128+2^N);
280 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(1
20*VE);CHR$(128+2^N);
290 IF N = 6 THEN N=-1
300 IF N = -1 THEN GOSUB330
```

```

310 N=N+1
320 RETURN
330 REM FIN DE LIGNE
340 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(0)
;CHR$(#FF);
350 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(2
40);CHR$(201);
360 LPRINT CHR$(#0D):RETURN
370 REM GRADUATIONS
380 M=A
390 REPEAT
400 A=A-7
410 UNTIL (A < 0)
420 A=A+7
440 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(1
15);
450 LPRINT CHR$(#1C);CHR$(#0A);CHR$(128+2^A);
460 A=M
470 RETURN
500 CALL#E6CA
510 LLIST

```

PROGRAMME 2

```

1 REM*****
2 REM* CALCUL DU DEPASSEMENT *
3 REM* ET DU TEMPS DE REPONSE*
4 REM*      A 5 %      *
5 REM*****
10 REM Calcul des valeurs de la fonction
12 GOSUB500
15 M%=5:M=1:N%=100
20 DIMA(400)
30 F0=1E4
40 W0=2*PI*F0
50 W2=W0^2
60 V0=0:D0=0:VS=V0:D1=D0:VE=1
70 K=2*M/W0
80 DT=1/(50*F0)
90 FORU=0 TO 400
100 D2=W2*(VE-V0-K*D0)
110 D1=D2*DT:D0=D0+D1
120 VS=D0*DT:V0=V0+VS
130 A(U)=V0

```

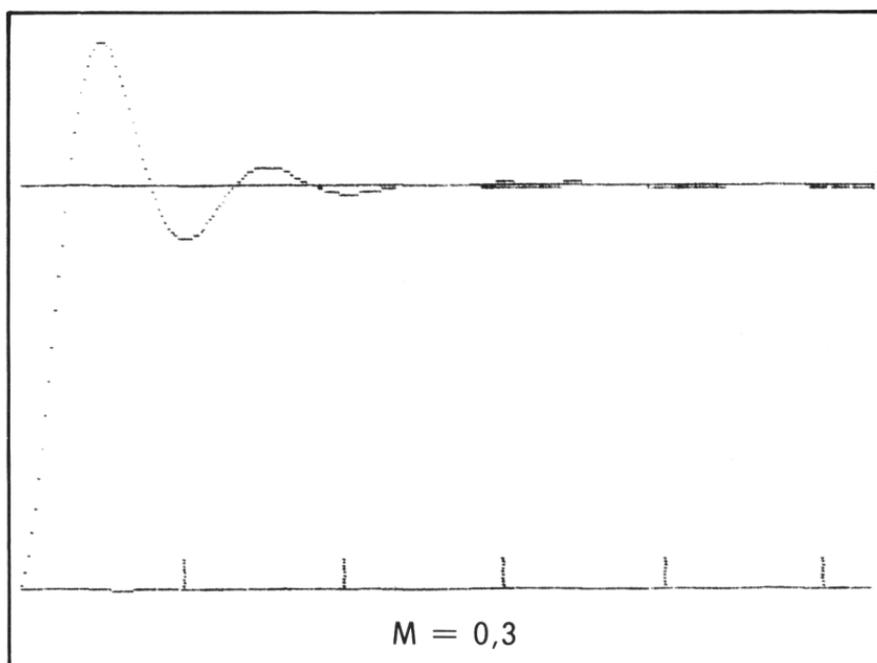
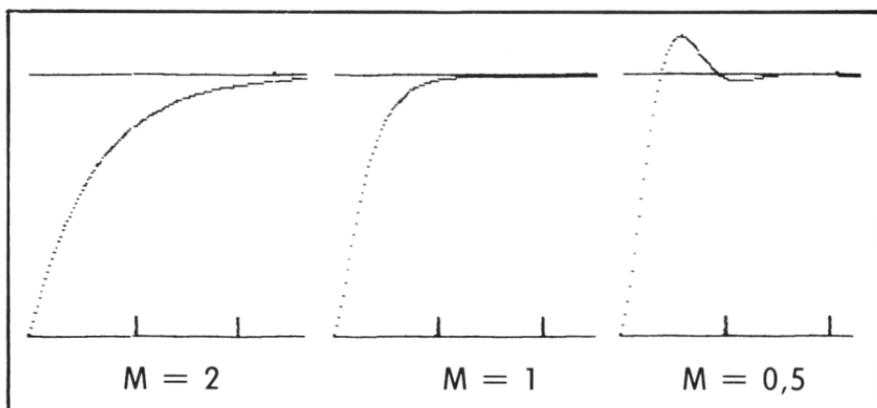
```

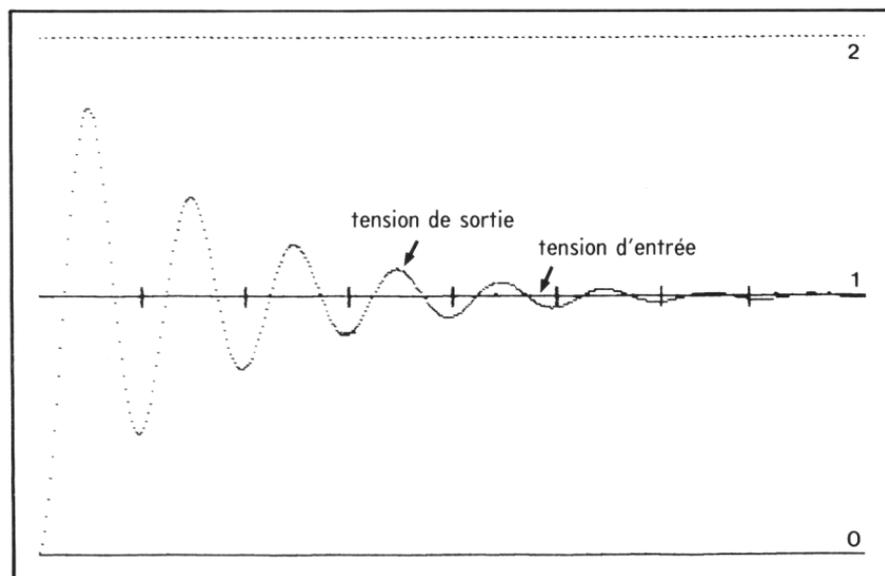
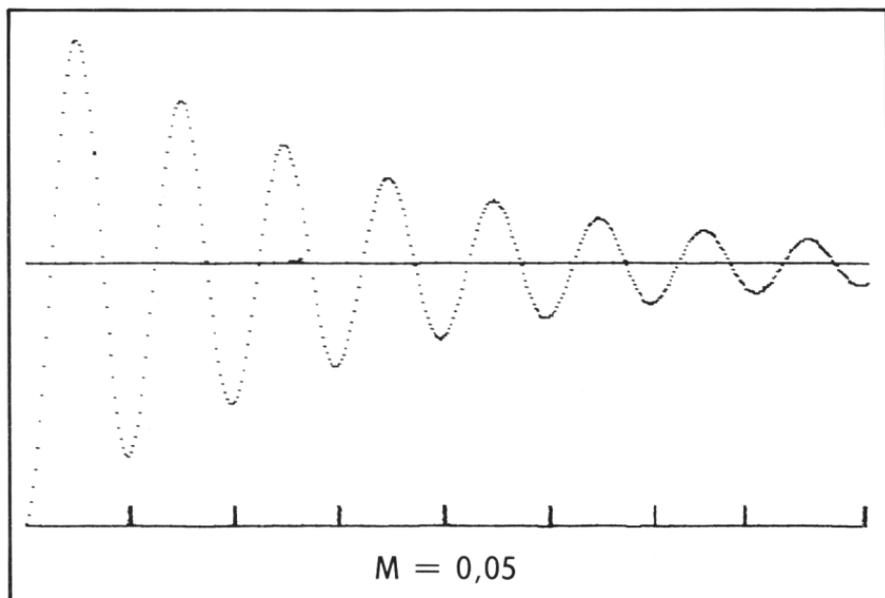
140 NEXTU
200 REM Determination du dePassement
210 U=1:D=0
220 REPEAT
230 IF A(U) > D THEN D=A(U)
240 U=U+1
250 UNTIL A(U) < D OR U=400
260 DP=INT((D-1)*100)/100
270 DR=EXP((-M*PI)/((1-M^2)^0.5))
280 DR=INT(DR*100)/100
300 REM Temps de rePonse
310 U=400
320 REPEAT
330 U=U-1
340 UNTIL (ABS(A(U-2)-1) > 0.05 AND ABS(A(U-1)-
1) < 0.05)
350 TR=U/(50*F0)
360 TR=INT(TR*1E6)
370 TC=INT(3E6/(M*W0))
380 PRINTM,DP,DR,TR,TC
385 GOSUB600
390 NX=N%+M%:M=N%/100
400 GOTO 30
500 REM Imprimante MISE EN PAGE
510 LPRINT CHR$(#10);"10*";CHR$(#10);"18DEPASSE
MENT";CHR$(#10);"38*";
520 LPRINT CHR$(#10);"44TEMPS DE REPONSE (";CHR
$(#AE);"s)":LPRINT
530 LPRINT CHR$(#10);"05M";CHR$(#10);"10*";CHR$
(#10);"13CALCULE";
540 LPRINT CHR$(#10);"24*";CHR$(#10);"26THEORIQ
UE";CHR$(#10);"38*";
550 LPRINT CHR$(#10);"41CALCULE";CHR$(#10);"52*
";CHR$(#10);"54THEORIQUE"
555 LPRINT
560 RETURN
600 REM Impression des donnees
610 LPRINT CHR$(#10);"05";M;CHR$(#10);"10*";CHR
$(#10);"16";DP;CHR$(#10);"24*";
620 LPRINT CHR$(#10);"28";DR;CHR$(#10);"38*";CH
R$(#10);"43";TR;
630 LPRINT CHR$(#10);"52*"
640 RETURN

```

Echelon de tension d'amplitude 1.

$F_0 = 10 \text{ kHz}$; graduations espacées de $1/F_0$.



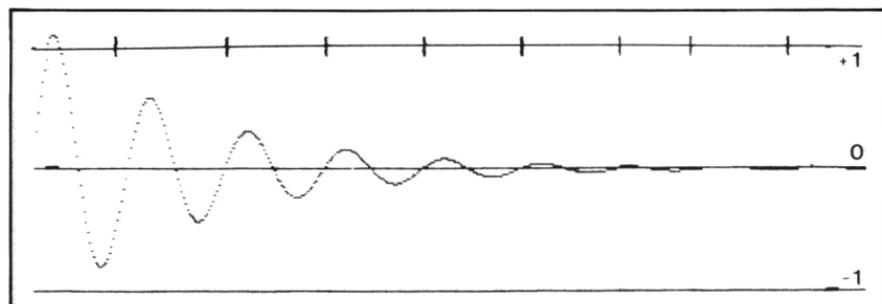


Programme 1 : échelon de tension d'amplitude 1.

$M = 0,1$; $F_0 = 10$ kHz.

(Graduations espacées de $1/F_0 = 100 \mu\text{s}$).

(Ligne 70 : FOR N = 0 TO 200 : A (N) = 1 : NEXT).



Programme 1 : impulsion de tension d'amplitude 10.

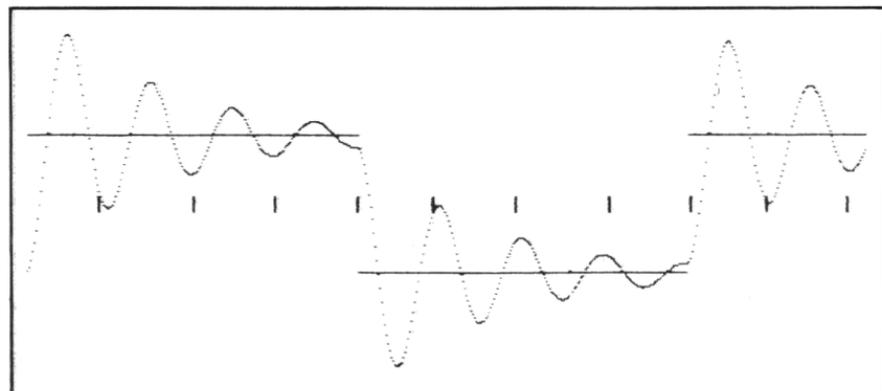
$M = 0,1 : F_0 = 10 \text{ kHz.}$

Modification.

```

70 FORN=0 TO200:A(N)=0:NEXT:A(1)=10
250 A(1)=0:GOTO130
260 REM IMPRIMANTE
270 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(
(60+60*V0);CHR$(128+2^N))
280 LPRIN CHR$(#1B);CHR$(#10);CHR$(0);CHR$(
(60);CHR$(128+2^N));

```



Programme 1 : signal rectangulaire.

$M = 0,1 F_0 = 10 \text{ kHz.}$

Modification.

```

95 I=0
125 IF I = 2 THEN RESTORE:I=0
127 I=I+1:READA:FORU=0 TO 200:A(U)=A:NEXT
129 DATA1,0

```

M	DEPASSEMENT		TEMPS DE REPONSE (µs)	
	* CALCULE	* THEORIQUE	* CALCULE	* THEORIQUE
.05	* .85	* .85	* 763	* 954
.1	* .72	* .72	* 457	* 477
.15	* .61	* .62	* 308	* 318
.2	* .52	* .52	* 216	* 238
.25	* .43	* .44	* 167	* 190
.3	* .36	* .37	* 124	* 159
.35	* .3	* .3	* 122	* 136
.4	* .24	* .25	* 115	* 119
.45	* .19	* .2	* 81	* 106
.5	* .15	* .16	* 81	* 95
.55	* .11	* .12	* 79	* 86
.6	* .08	* .09	* 78	* 79
.65	* .05	* .06	* 71	* 73
.7	* .03	* .04	* 45	* 68
.75	* .01	* .02	* 50	* 63
.8	* 0	* .01	* 54	* 59
.85	* 0	* 0	* 60	* 56
.9	* 0	* 0	* 66	* 53
.95	* -.01	* 0	* 71	* 50
1	* -.01	* 0	* 78	* 50
1.05	* -.01	* 0	* 83	* 50
1.1	* -.01	* 0	* 90	* 50
1.15	* -.01	* 0	* 96	* 50
1.2	* -.01	* 0	* 101	* 50
1.25	* -.01	* 0	* 108	* 50
1.3	* -.01	* 0	* 112	* 50
1.35	* -.01	* 0	* 117	* 50
1.4	* -.01	* 0	* 124	* 50
1.45	* -.01	* 0	* 128	* 50
1.5	* -.01	* 0	* 134	* 50
1.55	* -.01	* 0	* 137	* 50
1.6	* -.01	* 0	* 143	* 50
1.65	* -.01	* 0	* 150	* 50
1.7	* -.01	* 0	* 154	* 50
1.75	* -.01	* 0	* 159	* 50
1.8	* -.01	* 0	* 163	* 50
1.85	* -.01	* 0	* 170	* 50
1.9	* -.01	* 0	* 174	* 50

NON SIGNIFICATIF

NON SIGNIFICATIF

Programme 2 : comparaison des valeurs théoriques et calculées du dépassement et du temps de réponse à 5 % pour un circuit du second ordre en fonction de M. (On a $F_0 = 10$ kHz).

REGIMES TRANSITOIRES DANS LES BOUCLES A VERROUILLAGE DE PHASE

L'annexe III rappelle le principe de fonctionnement d'une boucle à verrouillage de phase. Ce composant joue un rôle important dans les systèmes de communication moderne. En particulier, il est aujourd'hui universellement employé pour la démodulation de fréquence dans les récepteurs radio. Les notations employées dans le texte sont les mêmes que dans l'annexe III.

Le programme sert principalement à étudier le régime transitoire pour un signal d'entrée FSK. Ce type de modulation est utilisé pour transmettre des informations binaires. Un « 0 » est représenté par une certaine fréquence F_0 (qui sera justement la fréquence centrale — ou de repos — de la boucle étudiée) et un « 1 » par une autre fréquence F_1 . Nous proposons un exemple numérique, en liaison avec une manipulation qui a été effectuée au laboratoire. Cela nous permettra de confronter résultats numériques et expérimentaux.

Considérons une boucle intégrée de Motorola MC 14046, pour laquelle le choix des valeurs des composants ($R_1 = 27 \text{ k}\Omega$, $R_2 = 68 \text{ k}\Omega$, $C_1 = 1250 \text{ pF}$) donne les résultats suivants : $F_0 = 53 \text{ kHz}$; $F_1 = 25 \text{ kHz}$. La fréquence de capture mesurée est d'environ $F_c \# 7 \text{ kHz}$ (le filtre passe bas utilisé est du type RC, avec $\frac{1}{2 \pi RC} = 1 \text{ kHz}$).

Le programme, au départ, « tourne » avec ces valeurs numériques. Le phénomène transitoire ayant à $t = 0$ une fréquence $F_1 - F_0$, il faut choisir DT , intervalle de calcul, petit devant la période d'un tel signal. Nous proposons à la fin de ce paragraphe 3 photos réalisées au laboratoire montrant la réponse d'une telle boucle pour 3 valeurs différentes de F_1 .

Le programme proposé permet de tracer la réponse de la boucle $V_2 = f(t)$, ainsi que la phase réellement appliquée au comparateur $\varphi = \varphi_i - \varphi_r$, qui est inaccessible à la mesure.

Le programme de base, tel qu'il est donné, montre la première valeur de F_1 pour laquelle il n'y a pas verrouillage ($F_1 = 61,95 \text{ kHz}$) ainsi que la variation de la phase φ (elle croît indéfiniment) (photo 1) (faire un GOTO 500 dès l'apparition de ILLEGAL QUANTITY ERROR). Il y a ensuite recopie de l'écran sur imprimante. On peut modifier ce programme de plusieurs façons.

— Observer dans les mêmes conditions V_2 et φ lorsqu'il y a verrouillage. Prendre alors (ligne 230) une valeur adéquate de F_i ($F_i < 61,95$ kHz). Le programme se déroule de la même manière ($F_i = 55$ kHz ; photo 2).

— Observer l'influence d'un paramètre. Par exemple, on peut s'interroger sur les conséquences d'une valeur non nulle de φ_i à $t = 0$. On entre pour cela la valeur désirée dans la variable P, ligne 220 (voir figure 1).

— On peut superposer les figures correspondant à diverses valeurs de P (par exemple, avec :

```
85 FOR P2=0 TO 2*PI STEP P1/4
(620 NEXT P) (voir figure 2).
```

— On peut enfin, si l'on ne dispose pas d'une imprimante, ne travailler que sur l'écran. On supprimera alors le GOTO 500 (ligne 370). L'écran apparaîtra comme sur la photo 3, en indiquant le dépassement mesuré et le dépassement théorique. On placera un GETA\$ ligne 485 et un STOP ligne 495, solution qui permettra, si le tracé des courbes n'est pas entièrement contenu sur l'écran, de réaliser un GOTO 280 qui permettra de superposer la suite des courbes (photo 4).

— Si l'on ne s'intéresse pas à la phase, on peut supprimer les lignes 120 à 160, ainsi que 360

Remarque : le calcul de la valeur finale VF (ligne 270) est approché. On a en fait $VF = \text{ARC Sin} ((F_i - F_0) * 6.28) / \text{kV}$.

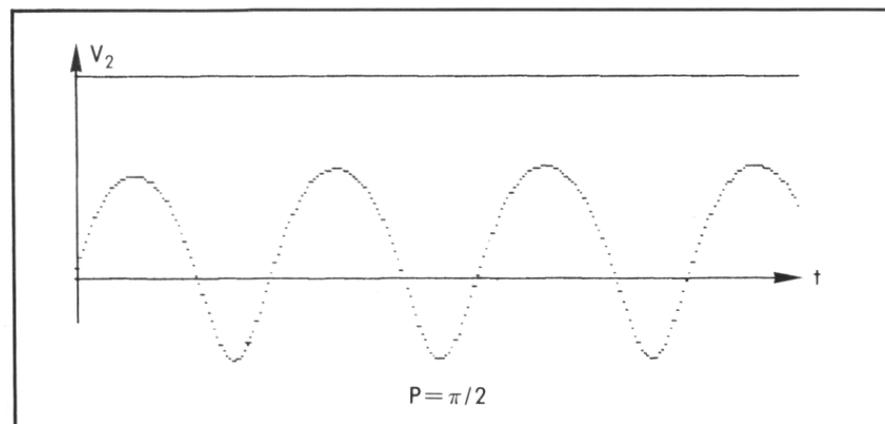
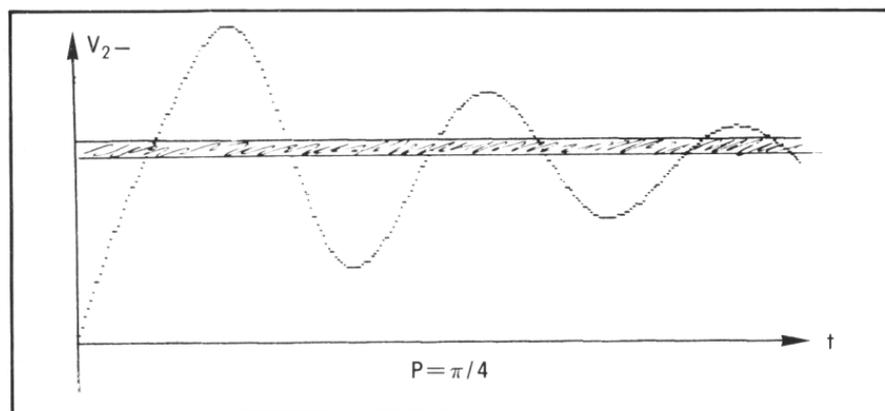
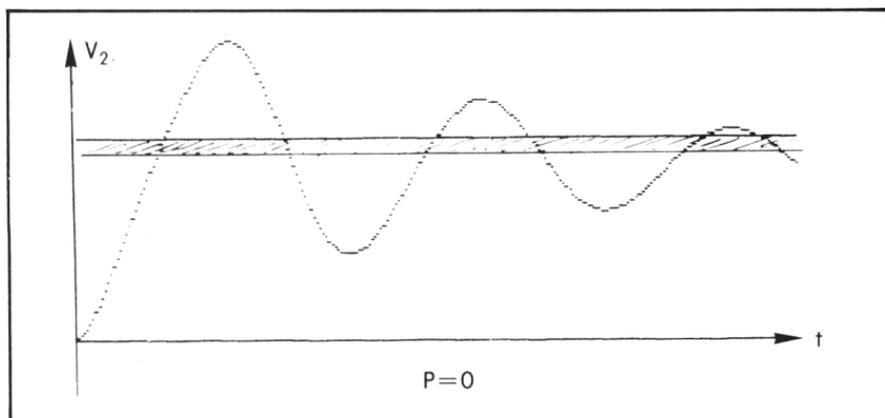
```
10 REM*****
20 REM* *
30 REM* P.L.L *
40 REM* FILTRE DU PREMIER *
50 REM* ORDRE *
60 REM* *
70 REM*****
80 DIMA(240)
90 HIRES
100 CURSET0,140,1:DRAW239,0,1
110 CURSET0,75,1:DRAW239,0,1
120 PATTERN99
130 FORN=0 TO 6
140 CURSET0,79+20*N,1
150 DRAW200,0,1
160 NEXT
170 LET F0=53E3
180 LET FL=25E3
190 LET F1=1E3:RC=1/(2*PI*F1)
```

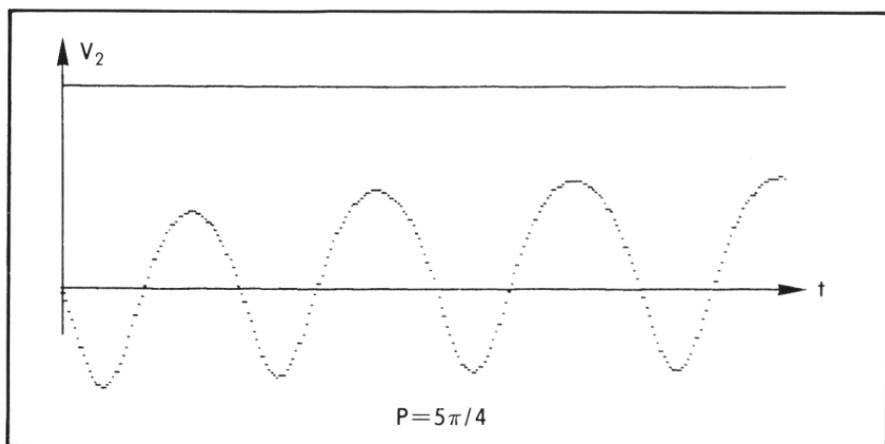
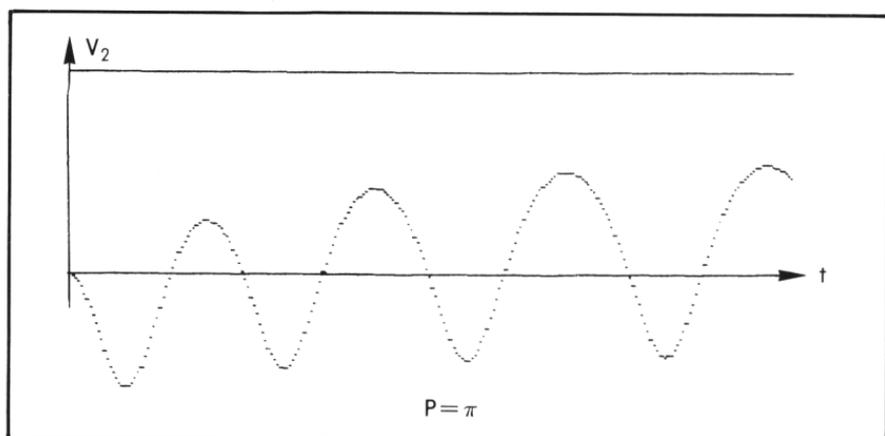
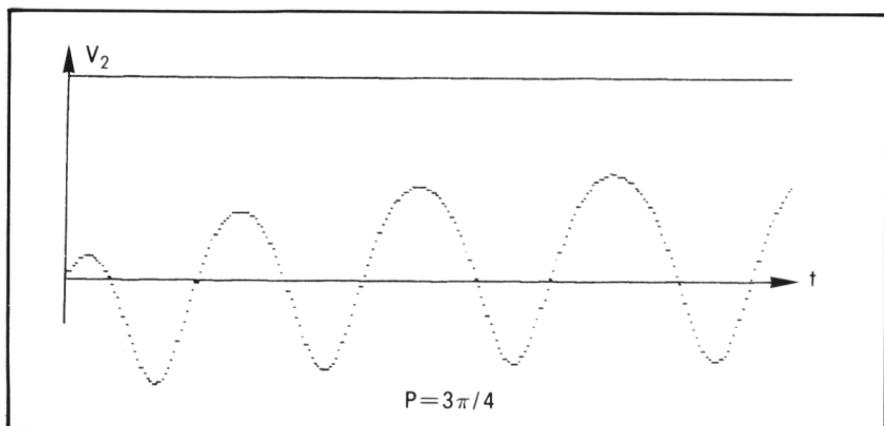
```

200 LET VDD=10
210 LET FX0=F0
220 PH=P
230 LET FI=61.95E3
240 LET KV=4*PI*FL/VDD
250 LET V1=0:V2=0:DV2=0
260 LET DT=2.5E-6
270 LET VF=((FI-F0)*6.28)/KV
280 FOR I=0 TO 239
290 LET PH=(FI-FX0)*2*PI*DT+PH
300 LET V1=(VDD/2)*SIN(PH)
310 LET DV2=((V1-V2)*DT)/RC
320 LET FX0=((KV/6.28)*(V2+DV2))+F0
330 LET V2=V2+DV2
340 A(I)=V2
350 CURSETI,140-(65*V2/VF),1
360 CURSETI,139-40*PH/PI,1
370 NEXTI:GOTO500
380 I=0:M=0
390 REPEAT :I=I+1
400 IF A(I) > M THEN M=A(I)
410 UNTIL A(I) < M
420 LET M=(M-VF)/VF
430 POKE28,0
440 PRINT" Le dePassement est : ";INT(M*100)/10
0;
450 PRINT"alors que la valeur theorique est";
460 M=(1/(16*RC*FL))^0.5:D=EXP((-M*PI)/((1-M^2)
^0.5))
470 D=INT(D*100)/100
480 PRINTD
490 POKE28,203
500 POKE49,255
510 LPRINT CHR$(#08)
520 FOR X=40999 TO 40960 STEP-1
530 FOR Y=X TO X+7960 STEP 40
540 A=PEEK(Y)
550 IF A >= 64 THEN A=A-64
560 IF A >= 32 THEN A=(A+192) ELSE A=(A+128)
570 LPRINT CHR$(A);
580 NEXTY
590 LPRINT CHR$(#0D)
600 NEXTX
610 LPRINT CHR$(#0F):POKE 49,80
620 STOP

```

Figure 1. $V_2(t)$ pour différentes valeurs du déphasage initial.





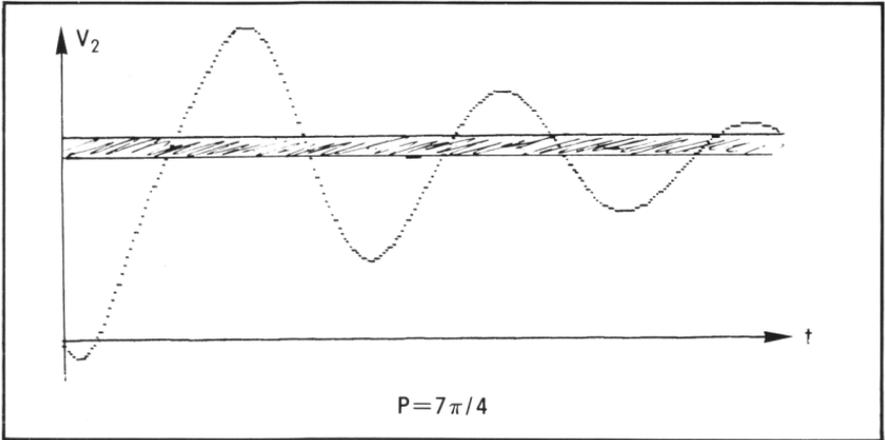
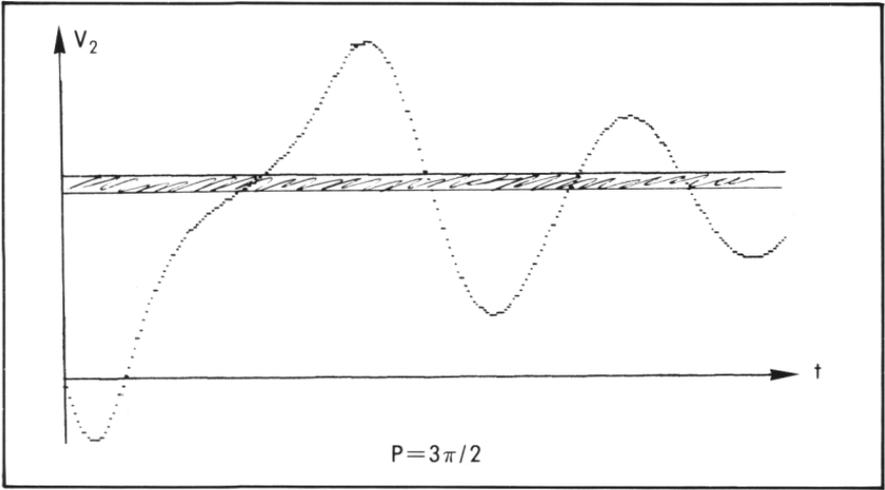
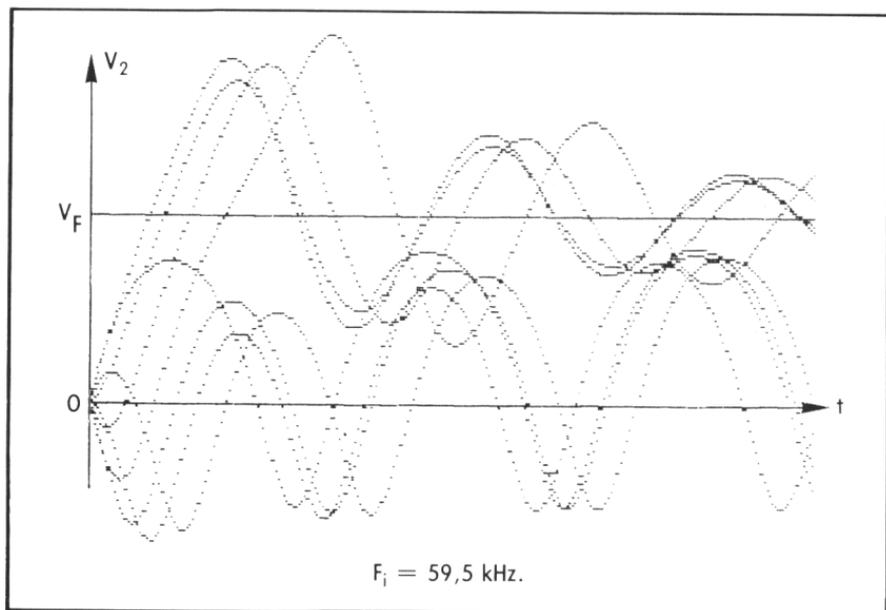
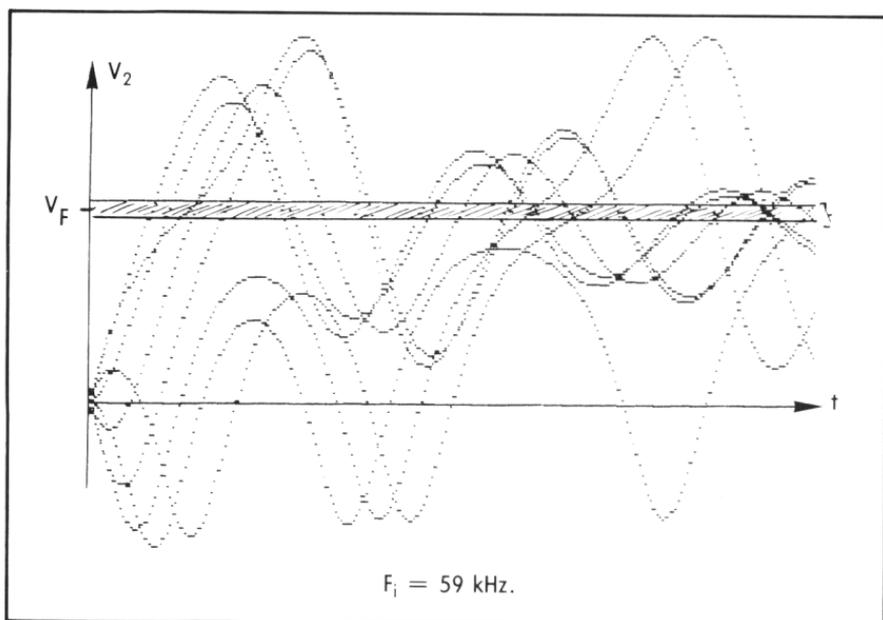


Figure 2. Superposition des courbes relatives à plusieurs valeurs du déphasage initial.



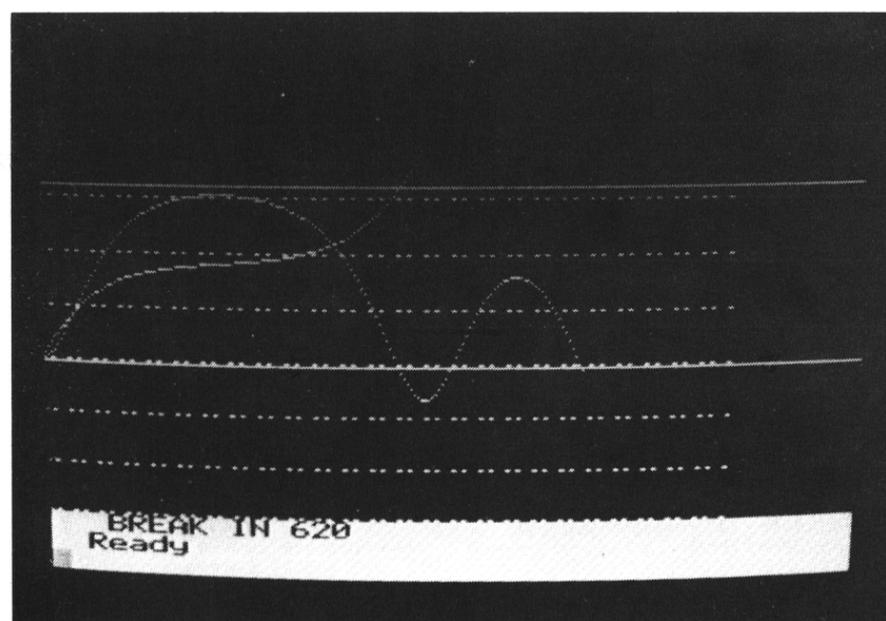
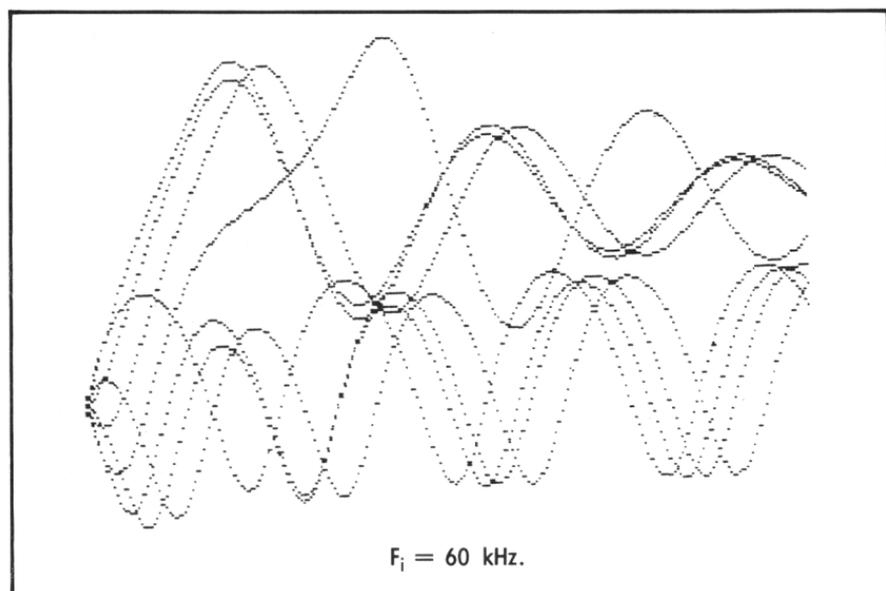


Photo 1 : $F_i = 61,95 \text{ kHz}$ (il n'y a pas verrouillage).

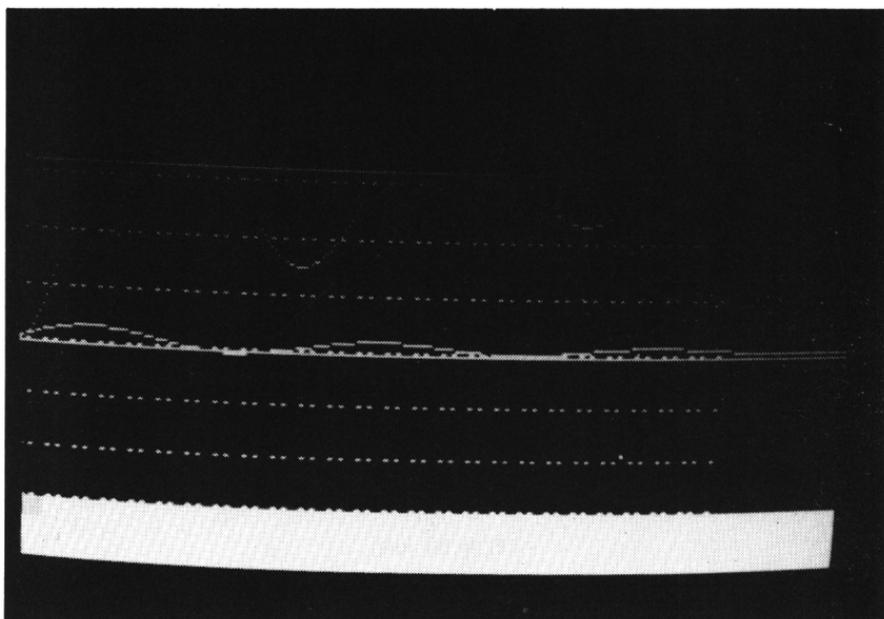


Photo 2 : $F_i = 55$ kHz (le régime peut être considéré comme linéaire).

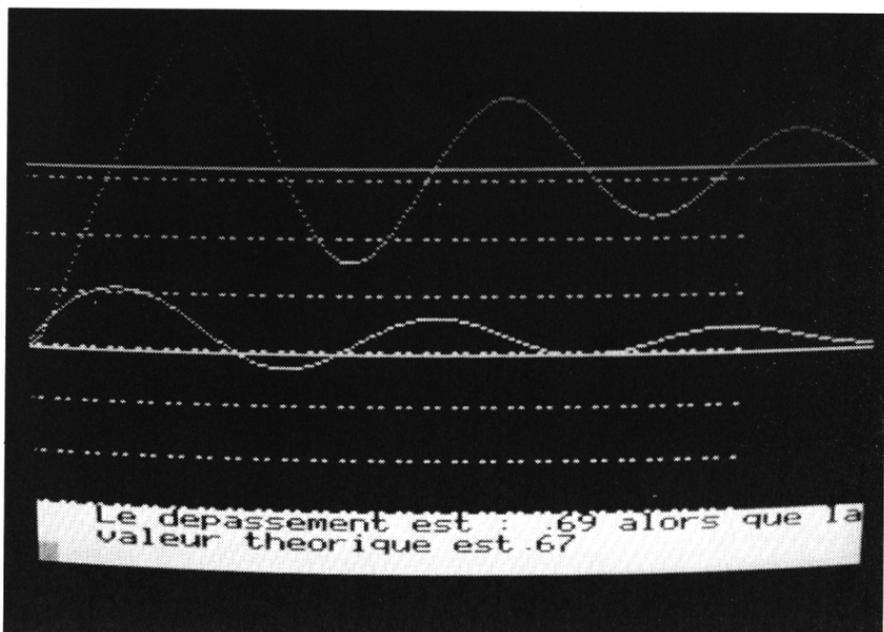


Photo 3 : $F_i = 60$ kHz (premier passage).

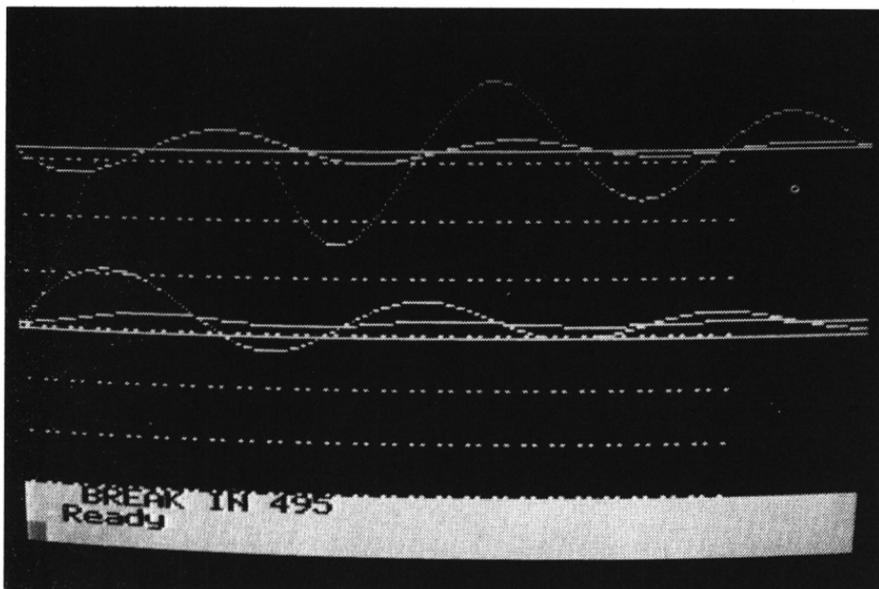
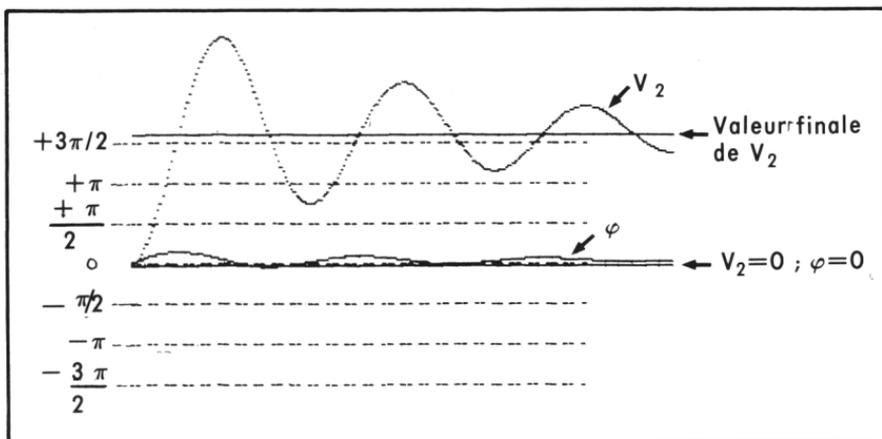
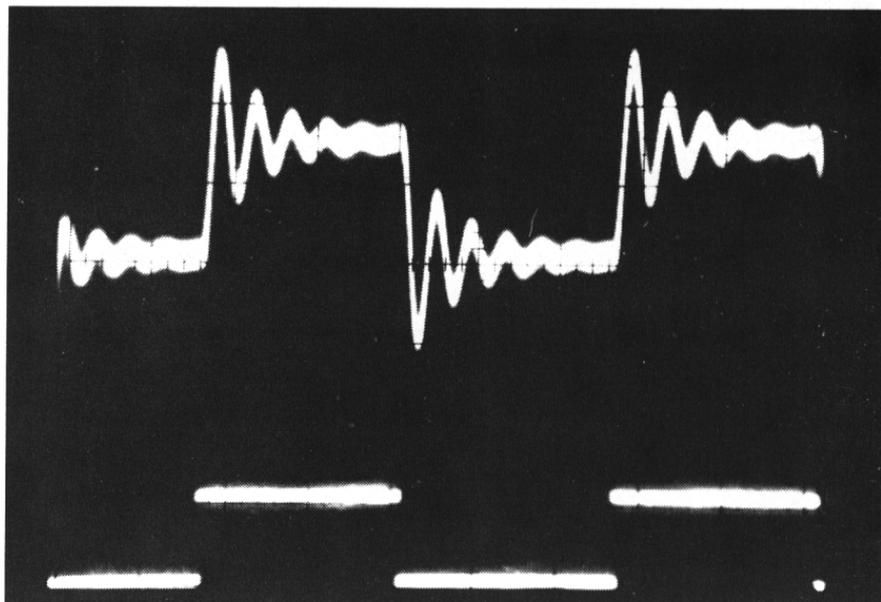


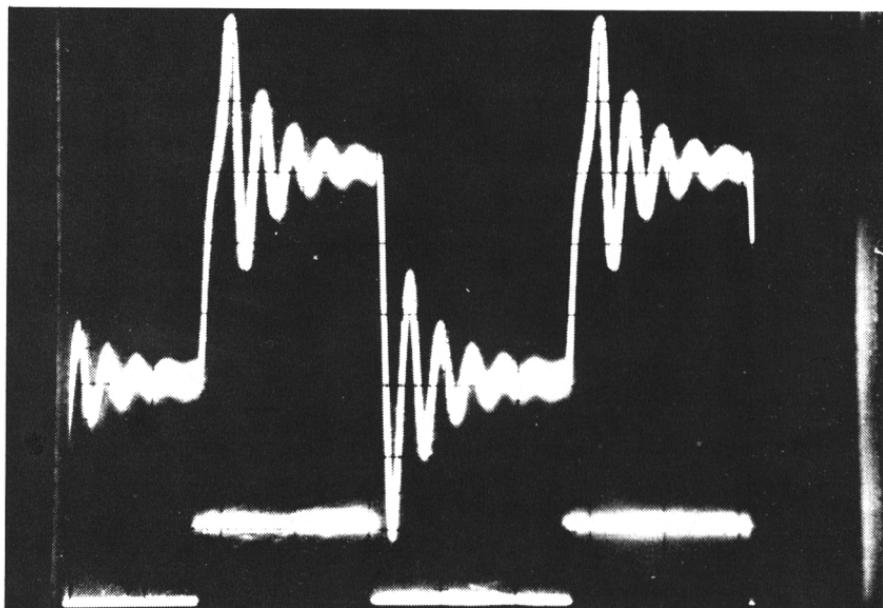
Photo 4 : $F_1 = 60 \text{ kHz}$ (deuxième passage).



« RECOPIE SUR L'IMPRIMANTE

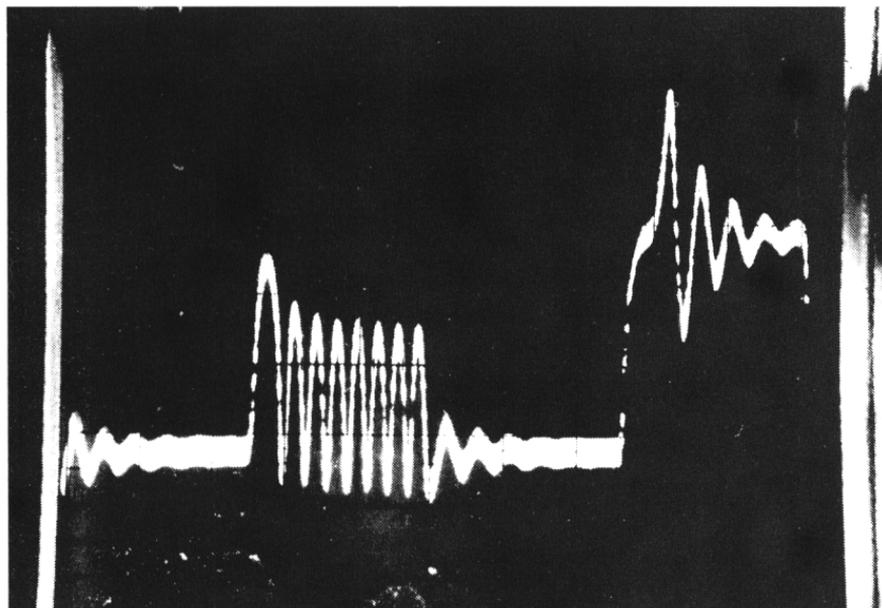


Oscillogramme N° 1 (comparer à photo 2).

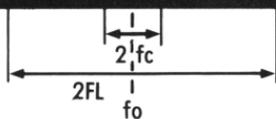
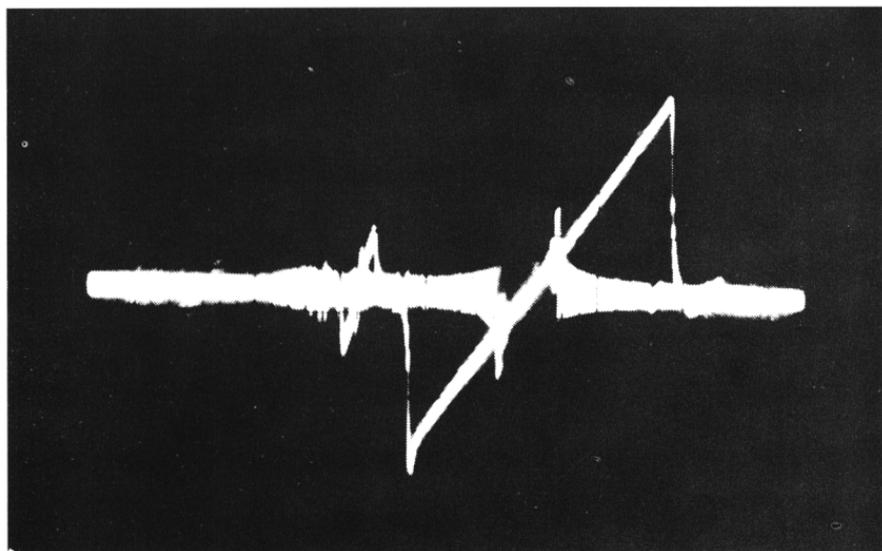


Oscillogramme N° 2 (comparer à figure 2) (Remarquer la « distorsion » sur la première demi-période).

Oscillogrammes relevés au laboratoire (boucle de même caractéristique que celle proposée dans le programme).



Oscillogramme N° 3 (instabilité de fonctionnement autour de la fréquence de capture).



Courbe de transfert relevée de la boucle ($V_2 = f(f_i)$)

Oscillogramme N° 4.

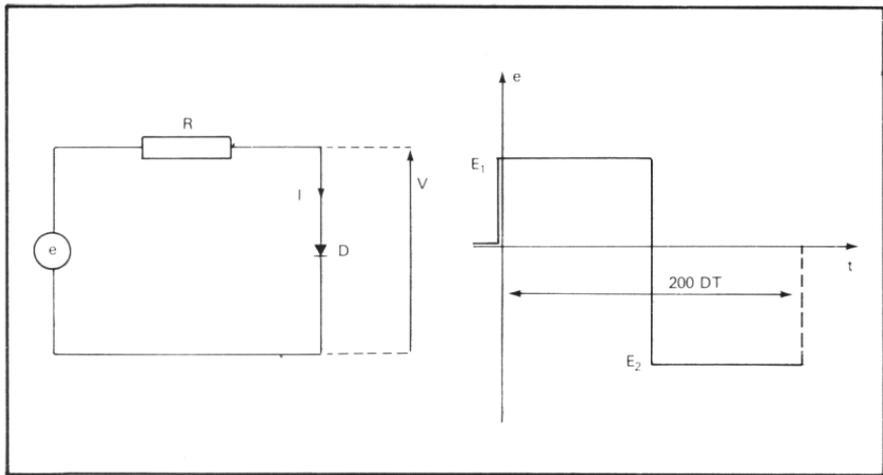
Chapitre 5

CIRCUITS NON LINEAIRES

DIODE EN COMMUTATION

La diode semi-conductrice est un composant essentiel en électronique. En commutation, elle présente des caractéristiques qui dépendent de sa technologie. Le choix d'un modèle particulier ne peut alors être quelconque.

Nous allons étudier les formes du courant et de la tension aux bornes d'une diode lorsque celle-ci est soumise à une tension rectangulaire.



Nous allons introduire pour cela un « modèle » par parties de la diode, faisant intervenir les phénomènes de charge stockée et de capacité de transition.

Nous distinguerons ainsi plusieurs types de fonctionnement :

* $I \geq 0$: $V = 0,6 \text{ V}$: $dQ/dt + Q/T = I$, en appelant Q la charge stockée dans la diode et T la durée de vie des porteurs minoritaires de celle-ci.

* $I < 0$ et $Q > 0$: la diode conduit, $V = 0 \text{ V}$.

$dQ/dt + Q/T = I$ (il y a déstockage).

* $I < 0$ et $Q = 0$: la diode se comporte comme une capacité (de diffusion) C_t . On a $I = C_t \cdot dV/dt$.

Ce modèle est justifié par l'étude expérimentale de la commutation d'une diode. Le signal d'entrée rectangulaire est défini dans un

tableau A (200), sur 200 points. L'intervalle de calcul DT fournit la largeur temporelle d'un point. Horizontalement, la fonction appliquée au circuit dure ainsi un temps 200 DT.

Il est possible de tenir compte de la variation de l'épaisseur de la zone de transition en fonction de la tension inverse aux bornes de la diode par la relation $C_1 = C_m \cdot (0,6 + V)^{-0,5}$. Ici, nous avons une capacité qui varie de 10 pF (pour $V = 0V$) à environ 2 pF (pour $V = -12 V$).

LE PROGRAMME EST SIMPLE :

- * Ligne 15 : définition des constantes employées dans le programme.
- * Lignes 20 à 40 : définition du signal d'entrée.
- * Lignes 60 à 120 : calcul de I et de V dans la diode.
- * Lignes 130, 150 à 180 : dessin des axes et inscription des notations sur l'écran.
- * Lignes 300 à 405 : recopie rapide d'écran.

VARIABLES

- * R : résistance du circuit (= 10 k Ω).
- * T : durée de vie des porteurs minoritaires (= 10⁻⁶s).
- * DT intervalle de calcul (10 ns).
- * C capacité de transition (C_1) de la diode en inverse.
- * A (200) contient le signal appliqué du circuit (en V).
- * E valeur instantanée de la tension appliquée au circuit (en V).
- * V tension aux bornes de la diode.
- * I courant dans la diode.
- * Q charge stockée.
- * DV variation de V pendant l'intervalle de temps DT.
- * DQ variation de Q pendant l'intervalle de temps DT.
- * N position horizontale du curseur ($0 \leq N \leq 200$).

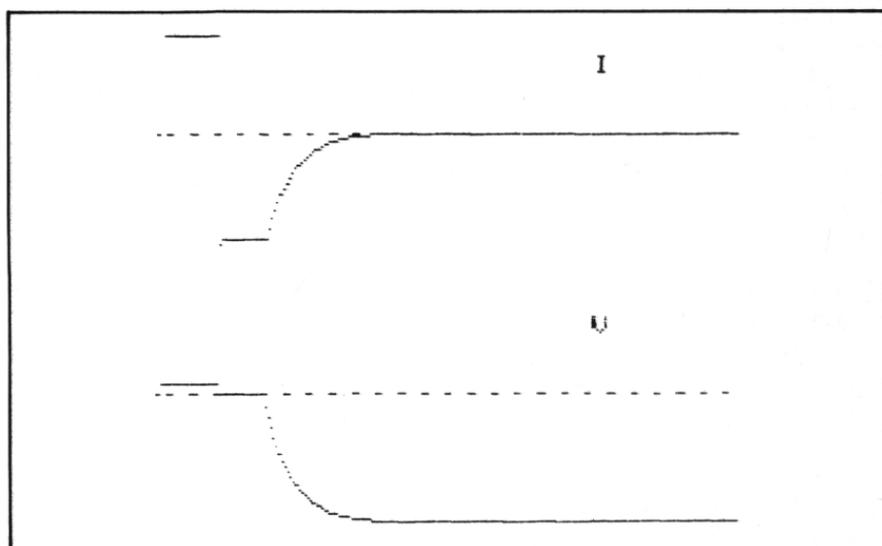
```
1 REM*****
2 REM*
3 REM* DIODE EN COMMUTATION *
4 REM*
5 REM*****
10 REM DIODE EN COMMUTATION
15 R=1E4:T=1E-6:DT=1E-8:C=1E-11
20 REM Exemple de signal d'entree
30 DIMA(200)
40 FORN=1 TO 200:A(N)=10*SIN(2*PI*N/200):NEXT
```

```

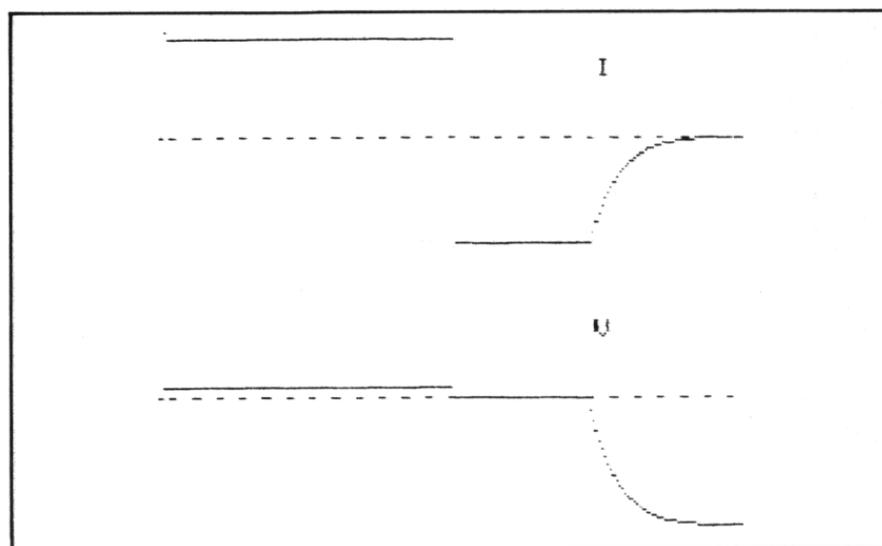
55 HIRES
60 FORN=1 TO 200
70 E=A(N)
80 I=(E-V)/R
85 IF I > 0 AND V < 0 THEN Q=0:DQ=0:DV=I*DT/C:
GOTO120
90 IF I > 0 THEN V=0.6:DV=0:DQ=(I-0/T)*DT
100 IF I < 0 AND Q > 0 THEN V=0:DV=0:DQ=(I-0/T)
*DT
110 IF I < 0 AND Q < 0 THEN Q=0:DQ=0:DV=I*DT/C
120 Q=Q+DQ:V=V+DV
130 CURSET N,50 - 4E4*I,1:CURSET N,150-5*V,1
140 NEXTN
150 CURSET150,20,0:CHAR73,0,1
160 CURSET150,120,0:CHAR86,0,1
170 CURSET0,50,1:PATTERN56:DRAW 200,0,1
180 CURSET 0,150,1:DRAW200,0,1
300 REM Recopie d'echan
310 LPRINT CHR$(#08):POKE49,255
320 FORX=40999 TO 40960 STEP -1
330 FOR Y=X TO X+7960 STEP 40
340 A=PEEK(Y)
350 IF A>= 64 THEN A=A-64
360 IF A >= 32 THEN A=(A+192) ELSE A=(A+128)
370 LPRINT CHR$(A);
380 NEXT Y
390 LPRINT CHR$(#0D)
400 NEXTX
405 LPRINT CHR$(#0F):POKE49,80
410 END
10000 CALL #E6CA
10010 LLIST

```

Influence de la durée du temps de conduction directe

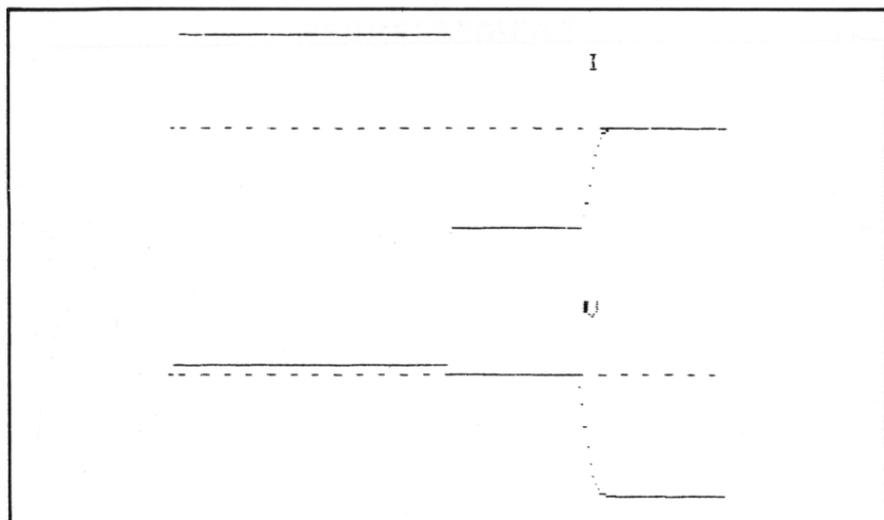


```
30 DIM A(200)
40 FOR N=1 TO 20: A(N)=10: NEXT
50 FOR N=21 TO 200: A(N)=-10: NEXT
```

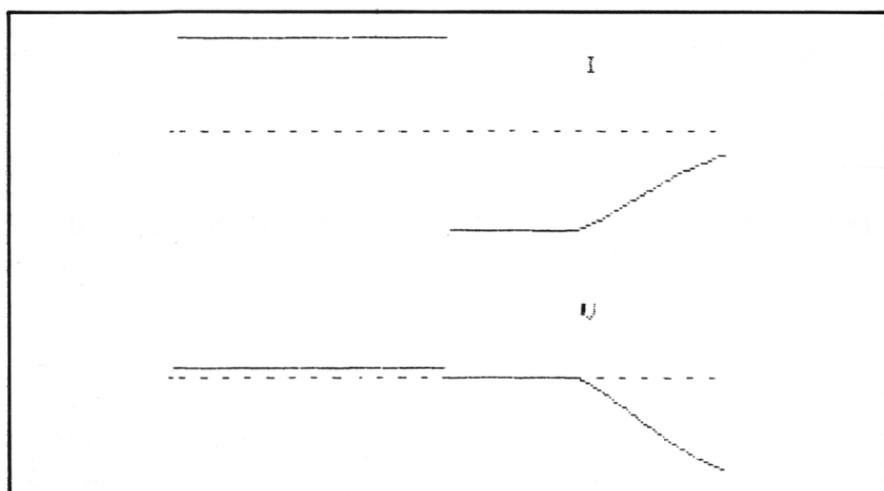


```
20 REM: Exemple de signal d'entree
30 DIM A(200)
40 FOR N=1 TO 100: A(N)=10: NEXT
50 FOR N=101 TO 200: A(N)=-10: NEXT
```

Variante : la capacité de transition dépend de la tension inverse



```
17 DEF FNC(V)=7.7E-12/((0.6-V)^0.5)
```



```
17 DEF FNC(V)=7.7E-11/((0.6-V)^0.5)
```

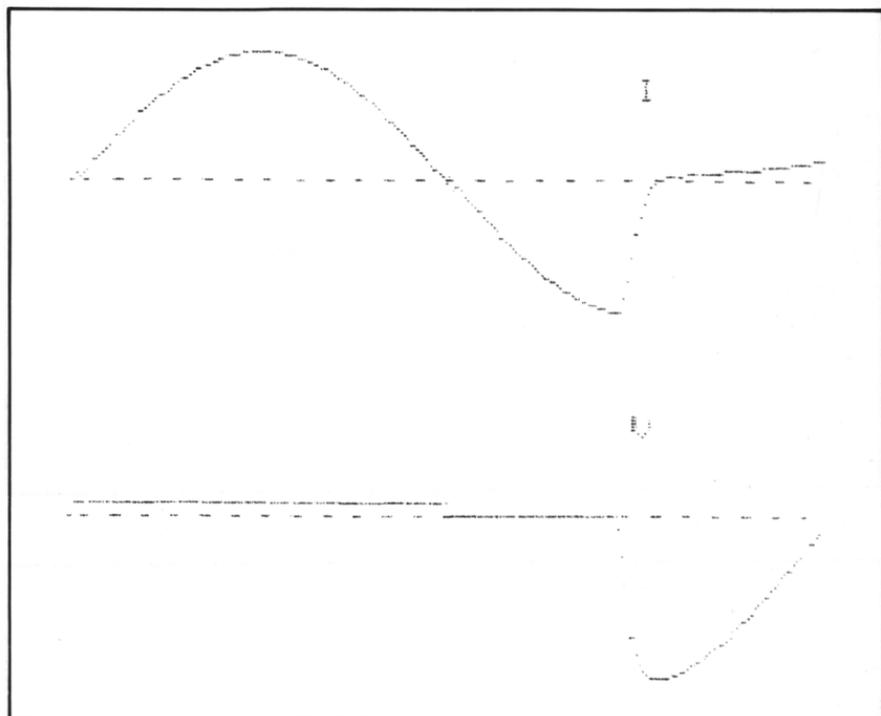
```
85 IF I > 0 AND V < 0 THEN C=FNC(V):Q=0:DQ=0:DV=I*DT/C:GOTO120
```

```
90 IF I > 0 THEN V=0.6:DV=0:DQ=(I-Q/T)*DT
```

```
100 IF I < 0 AND Q > 0 THEN V=0:DV=0:DQ=(I-Q/T)*DT
```

```
110 IF I < 0 AND Q <= 0 THEN C=FNC(V):Q=0:DQ=0:DV=I*DT/C
```

Application du programme (+ variante) au cas d'un signal sinusoïdal



```
(40 FOR N = 1 TO 200 : A(N) = 10 * SIN(2 * Pi * N / 200) : NEXT)
```

REDRESSEMENT

Ce programme propose l'étude de circuits classiques destinés au redressement. Pour des alimentations conçues pour de faibles débits, on utilise un montage simple alternance (fig. 1) ; sinon, on effectue un redressement double alternance (fig. 2). On s'intéresse en général à la qualité de la tension continue obtenue en sortie et au courant traversant la diode.

Le programme de base est relatif au montage de la figure 1. La tension d'entrée V_E est définie dans un tableau A(200) ; sa fréquence est celle du secteur : 50 Hz. La présence de la diode D impose 2 types de fonctionnement.

- $V_E > V_S$: la diode est conductrice ; ou a $V_E = V_S$

$$\text{et } I = \frac{V_S}{R} + C \frac{dV_S}{dt}$$

- $V_E < V_S$: la diode est bloquée ; le condensateur C se décharge dans la résistance R ; ou a $I = 0$

$$\text{et donc } \frac{V_S}{R} = -C \frac{dV_S}{dt}$$

Les différentes valeurs de V_S (il y en a 200 au total) sont mises en mémoire dans un tableau B(200). Celui-ci permet, après l'exécution des courbes par le programme, de calculer, puis d'afficher, les grandeurs caractéristiques de V_S : valeur moyenne et ondulation crête à crête. Le programme se modifie aisément pour permettre l'étude du courant I et du redressement double alternance. Tous les résultats obtenus sont « classiques », bien qu'il n'existe pas de solution analytique pour ce problème. Cependant, l'intérêt d'une telle étude est qu'elle permettrait, par exemple, de tenir facilement compte d'une charge R variable avec la tension appliquée V_S . On pourrait aussi étudier l'influence des imperfections de la diode (résistance directe, charge stockée).

LE PROGRAMME

Lignes 10 à 30 : définition des paramètres de fonctionnement du circuit : choix de R et C (DT = intervalle de calcul est fixe et vaut le centième de la période secteur). Définition de la tension V_E , dont

l'amplitude est prise ici égale à 90 V.

Lignes 40 à 100 : calcul de V_S et affichage des fonctions.

Lignes 110 à 160 : recherche des valeurs maximum, minimum et moyenne de V_S .

Lignes 170 à 210 : affichage des résultats.

LES VARIABLES

DT intervalle de calcul ($1/100^e$ de 20 ms = période secteur).

R résistance du circuit (en Ω).

C capacité du condensateur (en F).

T = RC constante de temps du circuit.

A(200) tableau contenant les 201 valeurs de V_E .

B(200) tableau contenant les 201 valeurs de V_S .

U variable de boucle locale.

DV variation de V_S pendant le temps DT.

MA, MI variables contenant, à la fin de la boucle 120 — 160, les valeurs maximum et minimum de V_S .

S somme des 100 dernières valeurs de V_S permettant, ligne 180, le calcul de la valeur moyenne de V_S .

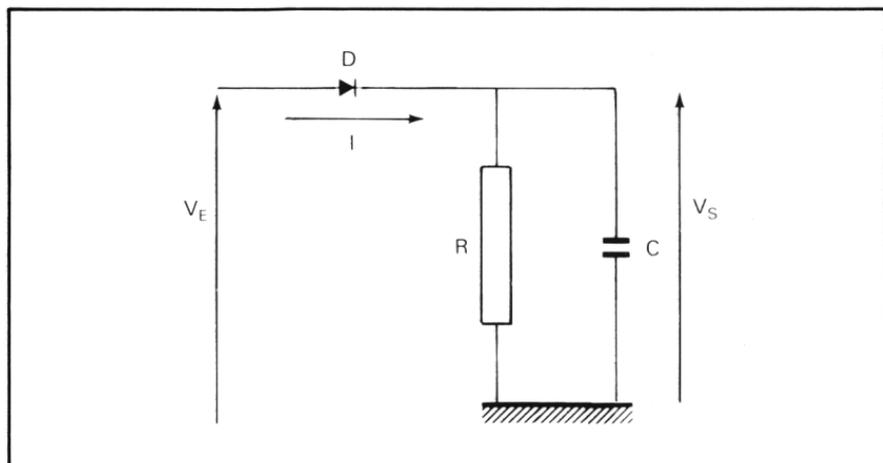


Figure 1

Redressement simple alternance.

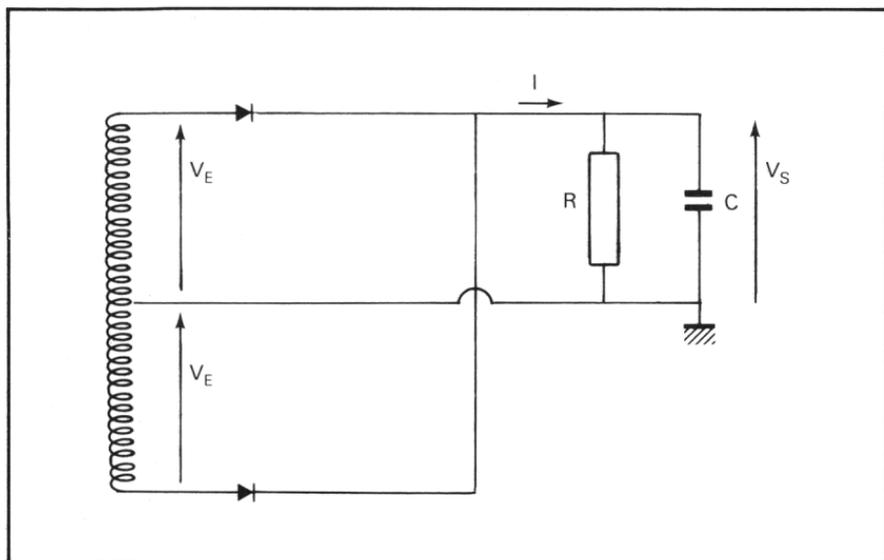


Figure 2
Redressement double alternance.

```

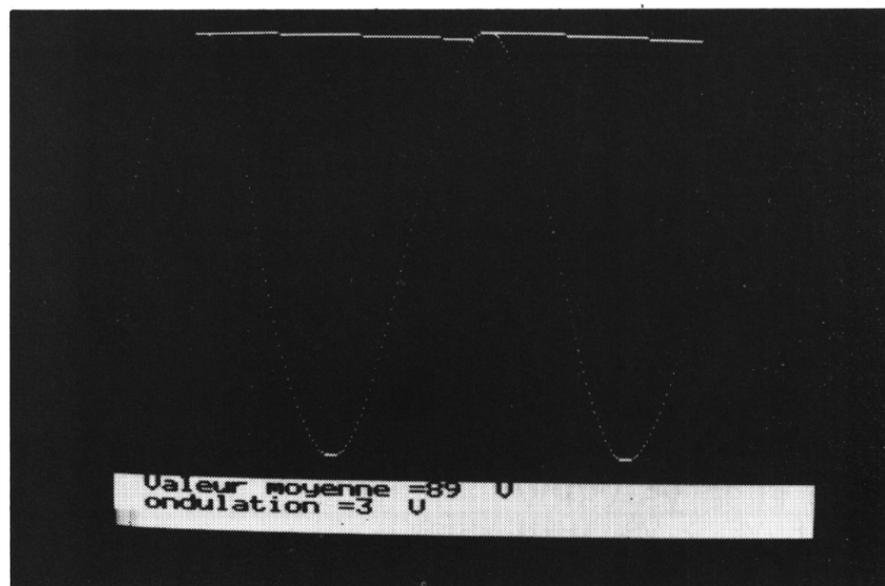
1 REM*****
2 REM*                                     *
3 REM*   ETUDE DU                         *
4 REM*   REDRESSEMENT                     *
5 REM*                                     *
6 REM*****
10 DT=1/(100*50):R=1E2:C=5E-4:T=R*C
20 DIMA(240):DIMB(240):HIRES:N=0
30 FOR U=0 TO 240:A(U)=90*SIN(2*PI*U/200):
NEXT
40 FOR U=0 TO 200
50 D=A(U+1)-A(U):DV=-VS*DT/T:VE=A(U):B(U)=
VS
60 IF ABS(D) >= ABS(DV) AND I > 0 THEN VS=
A(U) ELSE VS=VS+DV
70 IF VE > VS THEN VS=A(U)
80 I=(VS/R) + (D*C/DT)
90 IF I < 0 OR VS > VE THEN I=0
100 CURSETU,100-VE,1:CURSET U,100-VS,1
110 CURSETU,199-5*I,1

```

```

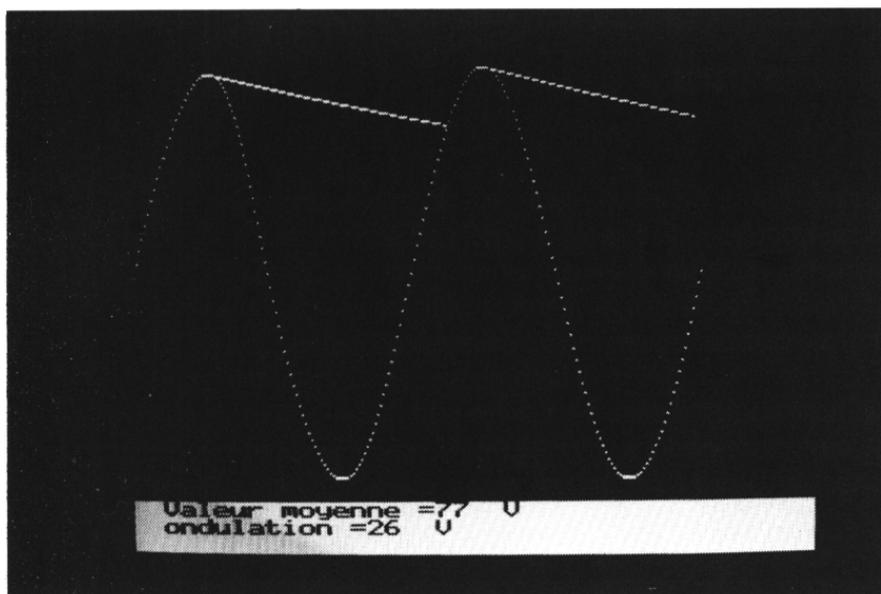
120 NEXTU:N=N+1:IF N < 2 THEN HIRES:GOTO40
130 REM CALCULS
140 MA=90:MI=90:S=0
150 FOR U = 0 TO 200
160 IF B(U) < MI THEN MI=B(U)
170 S=S+B(U)
180 NEXTU
190 POKE28,0
200 PRINT"Valeur moyenne =";INT(S/200);"V."
"
210 PRINT"Ondulation=";INT(MA-MI);"V."
220 GETA$
230 POKE28,203
240 END
250 CALL#E6CA
260 LLIST

```



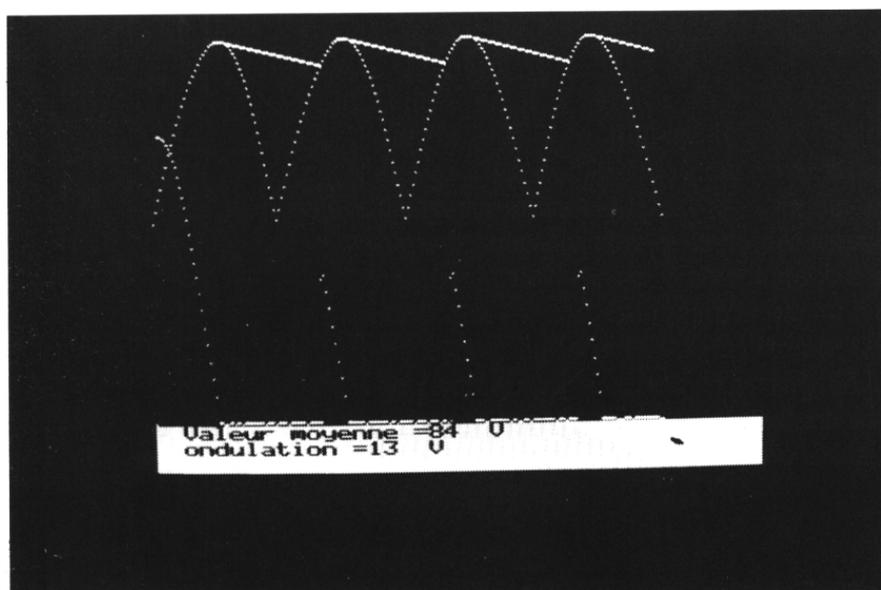
Redressement simple alternance

$R = 1000 \Omega ; C = 500 \mu F$



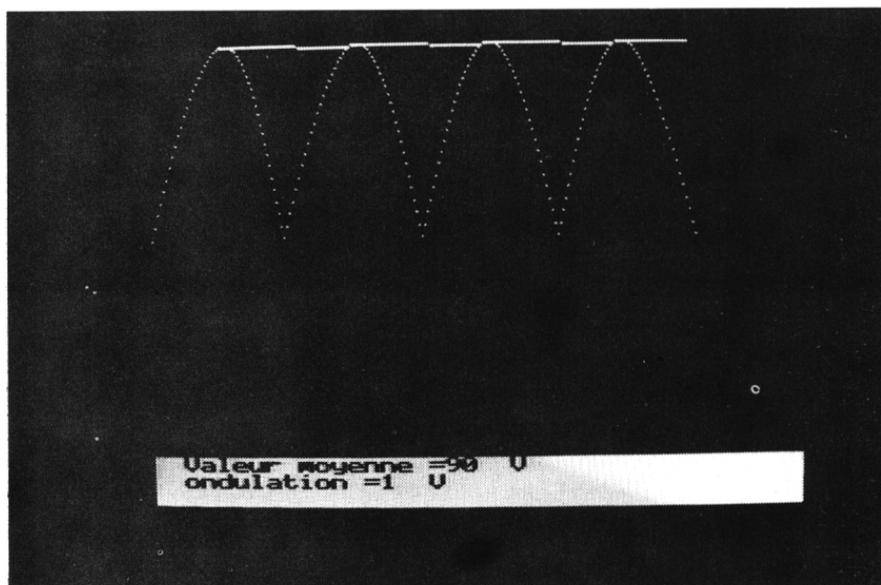
Redressement simple alternance

$R = 1000 \Omega ; C = 500 \mu\text{F}$



Redressement double alternance

$[V_E = \text{ABS}(A(U))]$ $R = 1000 \Omega ; C = 500 \mu\text{F}$



Redressement double alternance

(En bas : aspect du courant dans la diode)

[Rajouter : 97 CURSET U, 190 — 10*1,1]

$R = 1000 \Omega$; $C = 500 \mu\text{F}$

DETECTION

Ce programme, presque identique au précédent, s'attaque à un problème pourtant différent : la détection d'une onde modulée en amplitude, c'est-à-dire la démodulation. Il permet d'obtenir et d'étudier la forme du signal recueilli en sortie du détecteur. Il met en évidence le fait que la démodulation ne peut pas être parfaite quand le taux de modulation est de 1.

Le circuit détecteur étudié est identique à celui de la figure 1 du paragraphe « redressement ». L'expression d'une onde modulée en amplitude est donnée par :

$V_E : V_M \text{ Sin } \omega_H t \times (1 + m \text{ Sin } \omega_B t)$, dans laquelle :

- V_E est la valeur instantanée de l'onde modulée.
- V_M l'amplitude de l'onde (sinusoïdale) non modulée, c'est-à-dire quand $m = 0$.
- ω_H est la pulsation de la porteuse (HF).
- ω_B est la pulsation de la tension de modulation (BF).
- m est le taux de modulation ($0 \leq m \leq 1$).

Le condensateur C se charge à la valeur crête de V_E pendant ses maximums, puis se décharge dans la résistance R pendant les « creux » de la modulation. La tension V_S est donc une image de l'enveloppe de l'onde modulée ; on recueille ainsi en sortie du détecteur une tension basse fréquence (BF). Celle-ci n'est pas parfaite pour $m = 1$ (cas du programme). (On démontre en effet qu'il faudrait avoir

$$\frac{1}{RC\omega_B} \geq \frac{m}{\sqrt{1-m^2}}, \text{ ce qui est impossible pour } m = 1).$$

Le programme est difficilement exploitable sur un moniteur vidéo, quelle que soit la qualité de la haute résolution du micro-ordinateur employé. En effet, pour que le phénomène étudié soit clairement visible, il faudrait que l'axe OX ait une définition supérieure à 1000 points. L'imprimante convient ici particulièrement, puisque son axe OX est infini.

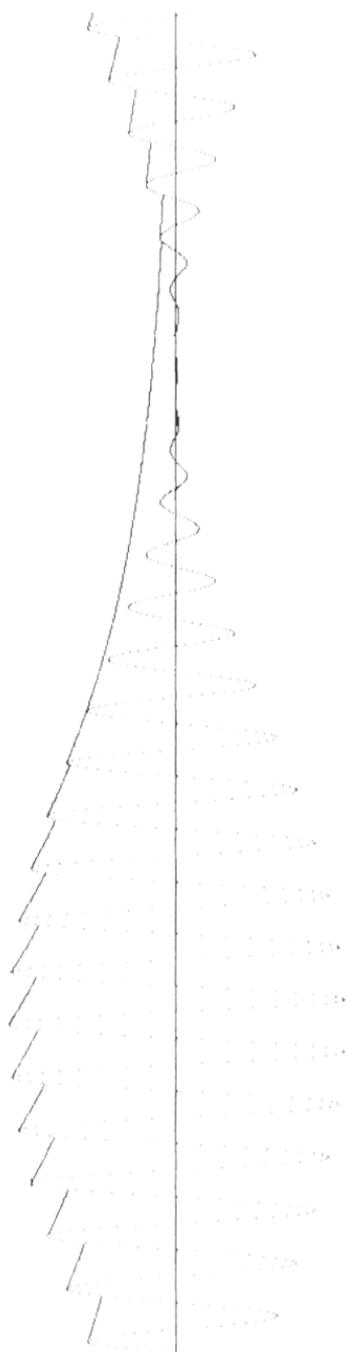
Ce programme est très court.

- Lignes 60 à 90 : définition des constantes.
- Lignes 100 à 180 : boucle de calcul. Le système est en équation lignes 130-140-150.
- Lignes 190-200 : impression d'une ligne de points sur l'imprimante.

LES VARIABLES

- BF : fréquence de l'onde basse fréquence (en Hz) ; ici BF = 2kHz.
- WB : pulsation de l'onde basse fréquence (en rad/s).
- HF : fréquence de l'onde haute fréquence (en Hz) ; ici HF = 50 kHz.
- WH : pulsation de l'onde haute fréquence (en rad/s).
- M : taux de modulation ; ici M = 1.
- RC : constante de temps du circuit détecteur (en s) ; ici RC = 100 μ s.
- DT : intervalle de calcul en s ; ici DT = 0,5 μ s.
- V_M : amplitude de l'onde HF non modulée.
- T : variable temps.
- V_E : valeur instantanée de l'onde modulée.
- V_S : tension de sortie du circuit détecteur.
- U : variable de travail pour l'imprimante, qui gère simultanément 7 points successifs (0 ≤ U ≤ 6).

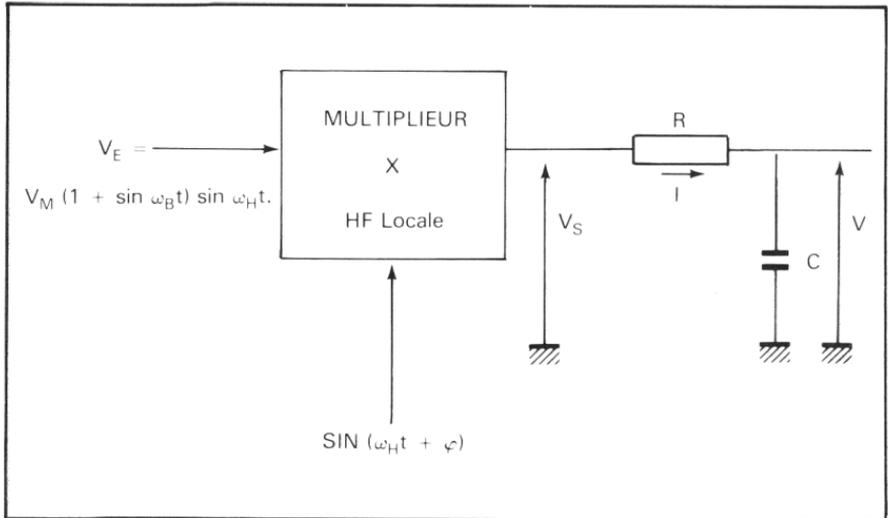
```
10 REM*****
20 REM* *
30 REM* DETECTION *
40 REM* *
50 REM*****
60 BF=2E3:WB=2*PI*BF
70 HF=5E4:WH=2*PI*HF
80 M=1:RC=1E-4:DT=5E-7:VM=1
90 T=0
100 LPRINT CHR$(#08):POKE49,255
110 FORU=0TO6
120 T=T+DT
130 VE=VM*SIN(WH*T)*(1+M*SIN(WB*T))
140 IF VE >= VS THEN DV=(VE-VS) ELSE DV=-VS*DT/
RC
150 VS=VS+DV
160 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(120
+60*VE);CHR$(128+2*U);
170 LPRINT CHR$(#1B);CHR$(#10);CHR$(0);CHR$(120
+60*VS);CHR$(128+2*U);
180 NEXTU
190 LPRINTCHR$(#1B);CHR$(#10);CHR$(0);CHR$(120
);CHR$(#FF);
200 LPRINTCHR$(#0D)
210 GOTO110
```



Pour $m = 1$, la démodulation d'amplitude ne peut être parfaite.

DETECTION SYNCHRONES

Ce programme est à comparer au précédent. Il permet de démoduler une onde modulée en amplitude. Le circuit de principe est le suivant :



On démontre que

$$V_S = \frac{1}{2} V_M \cos \varphi + \frac{1}{2} V_M \cos \varphi \sin \omega_B t - \frac{1}{2} V_M (\sin \omega_B t + 1) \cos (2\omega_H t + \varphi)$$

Le détecteur se compose donc simplement d'un multiplieur (que l'on appelle mélangeur équilibré en HF) ; sa sortie fournit une composante continue, l'onde BF démodulée et une nouvelle onde modulée en amplitude, de pulsation $2\omega_H$ que l'on élimine à l'aide du filtre RC (passe bas). On comprendra aisément que ce type de démodulation est beaucoup plus linéaire que le précédent. Rappelons que le rapport signal/bruit est aussi amélioré de 6 db. L'onde $\sin (\omega_H t + \varphi)$ doit être absolument synchrone de la porteuse. En effet, leur déphasage φ intervient directement dans l'expression de l'onde démodulée et il est hors de question qu'un tel déphasage varie en fonction du temps. A titre d'exemple, on peut voir sur les photos l'influence de la valeur de

φ sur l'aspect de l'onde démodulée.

Le programme est simple. L'expression de l'onde modulée V_E est établie, ainsi que celle de V_S . Le circuit RC est mis en équation de façon classique.

$$V_S = RI + V$$

$$I = C \frac{dV}{dt} \quad \text{d'où } dV = \frac{(V_S - V) dt}{RC}$$

Il y a donc deux « passages » sur l'écran : seul le second a été photographié, le premier en effet est affecté du régime transitoire du système.

LE PROGRAMME :

- * lignes 70 à 120 : définition des constantes
- * lignes 130 à 190 : calcul des différentes tensions
- * ligne 200 : mise en équation du circuit RC.
- * lignes 210 et 220 : affichage de l'enveloppe de l'onde modulée et de l'onde démodulée.

LES VARIABLES

- * BF : fréquence de l'onde basse fréquence (en Hz).
- * WB : sa pulsation.
- * HF : fréquence de l'onde porteuse (en Hz).
- * WH : sa pulsation.
- * PH : déphasage (en radian) entre la porteuse et la HF locale.
- * M : taux de modulation ($0 < M < 1$).
- * DT : intervalle de calcul (en s).
- * V_M : amplitude de l'onde HF.
- * FC : fréquence de coupure adoptée pour le filtre passe bas.
- * RC : valeur correspondante de la constante de temps.
- * T : variable temps.
- * L : compteur du nombre de « passages » sur l'écran.
- * DV : variation de V pendant DT.
- * VB : expression de l'enveloppe de l'onde modulée.
- * VE : expression de l'onde modulée.
- * VS : tension de sortie du multiplieur.
- * V : tension de sortie du filtre passe bas.
- * U : variable de boucle pour l'imprimante.

```

10 REM*****
20 REM*
30 REM* DETECTION *
40 REM* SYNCHROME *
50 REM*
60 REM*****
70 BF=4E3:WB=2*PI*BF
80 HF=1E5:WH=2*PI*HF
90 PH=4*PI/8
100 M=1:DT=1E-6:VM=1
110 FC=1E4:RC=1/(2*PI*FC)
120 T=0
130 HIRIS
140 L=L+1
150 FORU=@T0239
160 T=T+DT
170 VB=VM*(1+M*SIN(WB*T))
180 VE=VM*SIN(WH*T)*(1+M*SIN(WB*T))
190 VS=VE*SIN(WH*T+PH)
200 DV=(VS-V)*DT/RC:V=V+DV
210 CURSETU,99-49*VB,1
220 CURSETU,190-99*V,1
230 NEXTU
240 IF L = 2 THEN STOP
250 GOTO 130

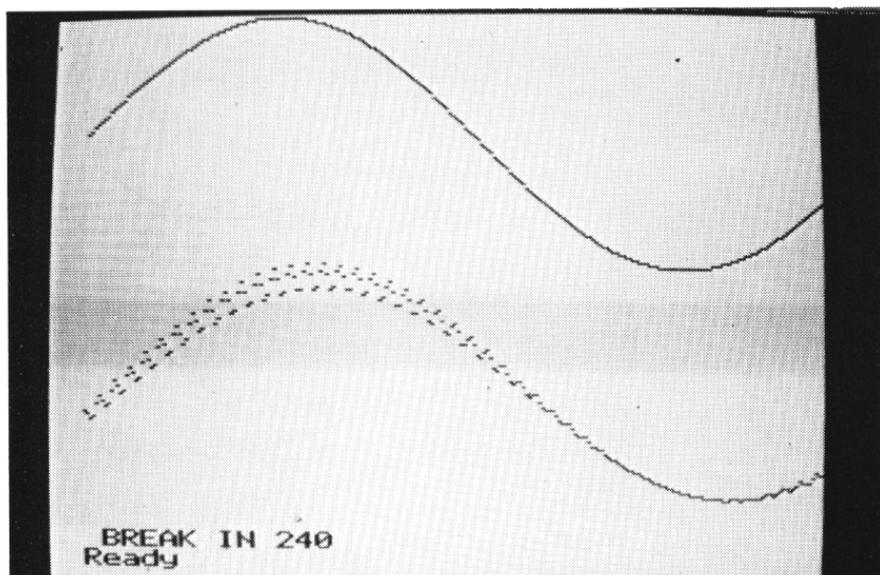
```

Légende

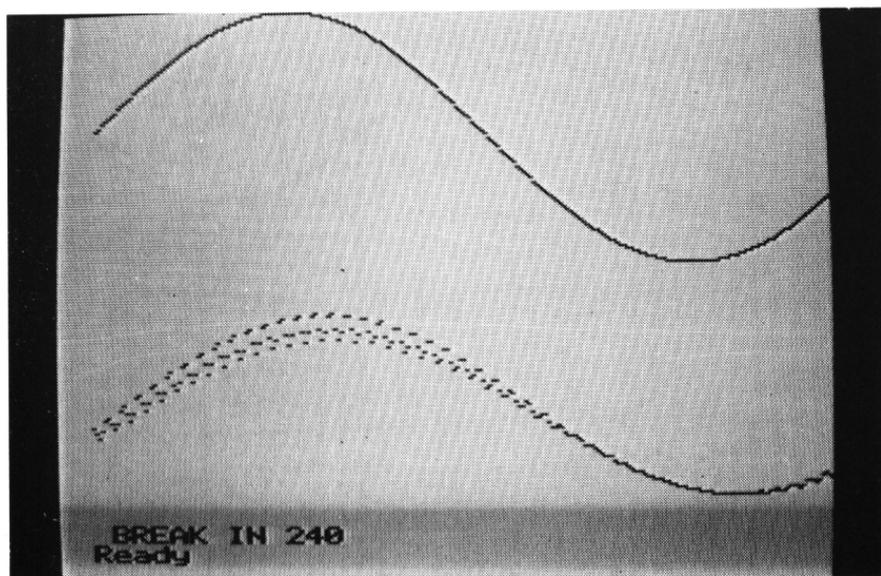
Les courbes présentées montrent l'aspect de la tension BF (en haut) et de la tension démodulée (en bas) suivant la valeur de PH (déphasage entre porteuse et l'oscillateur local). On constate que la tension de sortie est bien la somme :

- d'une composante continue ;
- de la BF démodulée ;
- d'une onde modulée en amplitude de pulsateur $2\omega_H$.

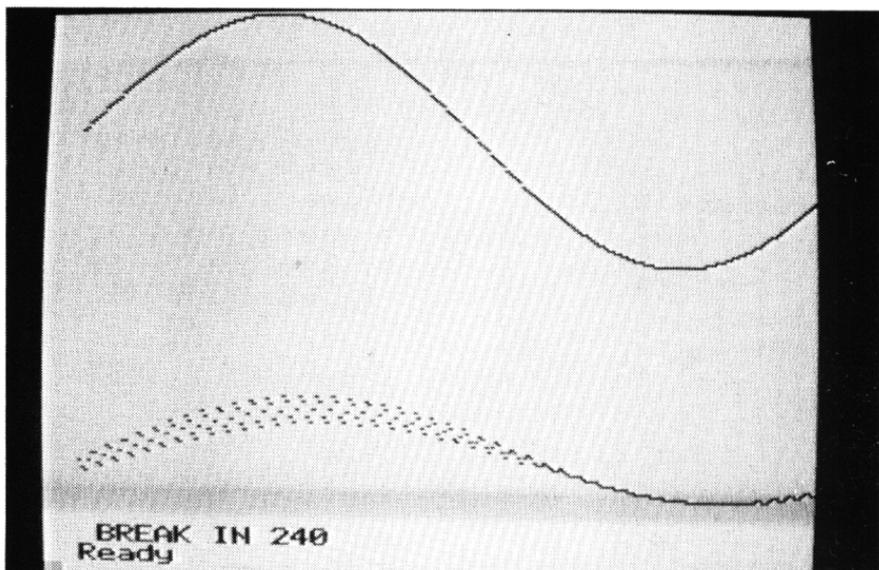
Il n'y a pas de détection pour $PH = \pi/2$.



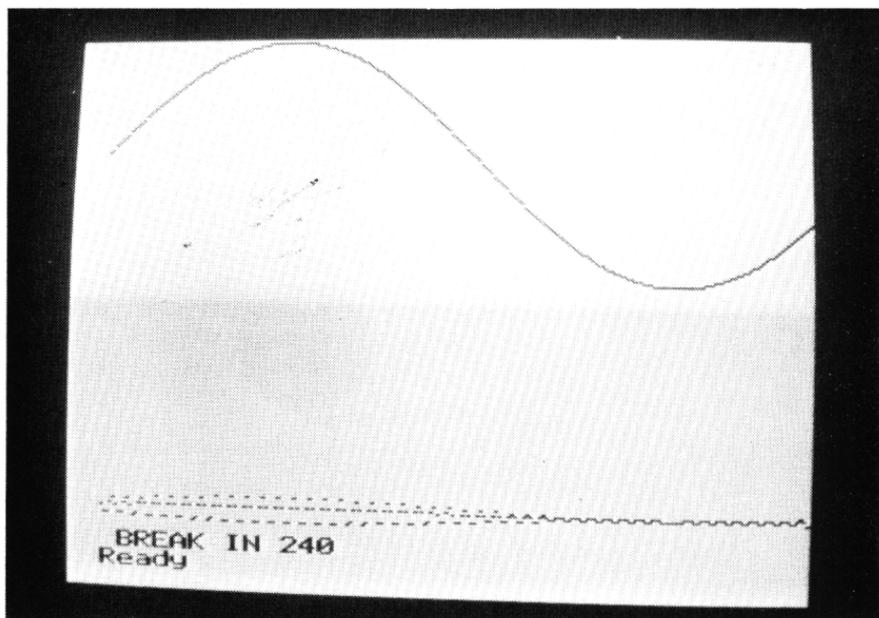
$PH = 0$



$PH = \pi/8$



$$PH = 3\pi/8$$



$$PH = \pi/2$$

Chapitre 6

ABAQUE DE SMITH

ABaque DE SMITH

Cette abaque, imaginée par l'ingénieur Smith, permet de résoudre un grand nombre de problèmes relatifs à la transmission de signaux HF. Il s'agit d'un cercle, à l'intérieur duquel on peut représenter n'importe quelle impédance, un déplacement le long d'un câble de transmission se ramenant à une rotation dans ce plan. Le programme que nous proposons utilise la méthode classique d'exploitation de l'abaque, mais par l'emploi de ses propriétés analytiques et non graphiquement.

Cette abaque est un cercle, dont tout point intérieur permet de représenter une impédance normalisée. Soit Z l'impédance étudiée et Z_c l'impédance caractéristique de la ligne employée ;

alors $\frac{Z}{Z_c}$ est un nombre sans dimensions, s'exprimant par :

$$\frac{Z}{Z_c} = R + jX, \quad R \text{ représentant la partie résistive,}$$

$X (\geq 0)$ la partie réactive.

— un R donné correspond à un cercle dont le centre est sur l'axe des x d'abscisse

$$\frac{R}{R + 1} \text{ et d'ordonnée } 0.$$

Le rayon de ce cercle est $\frac{1}{R + 1}$.

— un X donné correspond à un cercle, dont le centre a pour abscisse 1, pour ordonnée

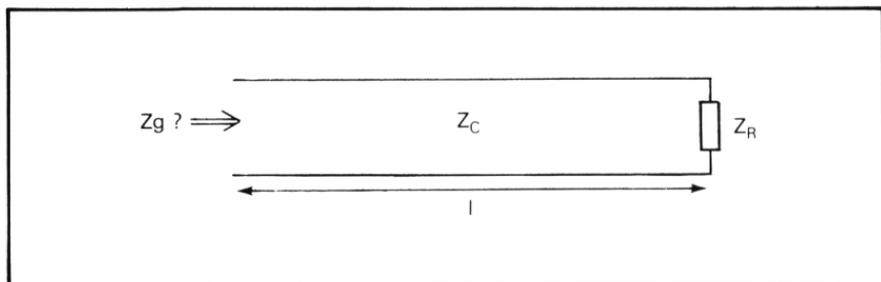
$$\frac{1}{X}. \text{ Le rayon de ce cercle est } 1/X.$$

Ceci permet de tracer la figure 1.

D'autre part, le grand cercle est gradué en longueur d'onde, un tour complet correspond à $\lambda/2$.

Rappelons la manière dont on utilise une telle abaque.

Soit un générateur attaquant une ligne d'impédance caractéristique Z_c , fermée au bout d'une longueur l sur une impédance Z_r . Quelle est l'impédance Z_g que « voit » le générateur ?



La méthode consiste à :

— placer sur l'abaque le point représentatif de $\frac{Z_R}{Z_C} = R + jX$. (Point A)

(sur le dessin, nous avons ici $R = 0,33$; $X = + 0,5$).

— faire tourner ce point autour du centre O de la figure, en lui faisant décrire un arc α

tel que $\alpha = \frac{4 \pi l}{\lambda}$.

— placer le point B ainsi obtenu et « lire » ses caractéristiques (ici, on a environ : $R' = 2$, $X' = + 1,5$)

— En déduire $Z_g = (R' + jX')Z_C$.

Cette méthode graphique est extrêmement pratique et permet d'ailleurs de résoudre une infinité d'autres problèmes. En fait, mise au point à une époque où les calculatrices n'existaient pas, c'était le seul moyen rapide d'obtenir la solution.

Maintenant, nous pouvons reprendre les équations des cercles et traiter le problème de façon purement numérique. Nous en faisons grâce au lecteur et lui épargnons la démonstration. Les équations que nous fournissons apparaissent dans tous les traités un peu âgés sur l'abaque de Smith.

— Connaissant R et X, on peut en déduire la position du point correspondant dans le plan. On a :

$$\text{Position du point A} \quad \left\{ \begin{array}{l} y_0 = \frac{2X}{X^2 + (R + 1)^2} \\ x_0 = \frac{1 - Y(R + 1)}{X} \end{array} \right.$$

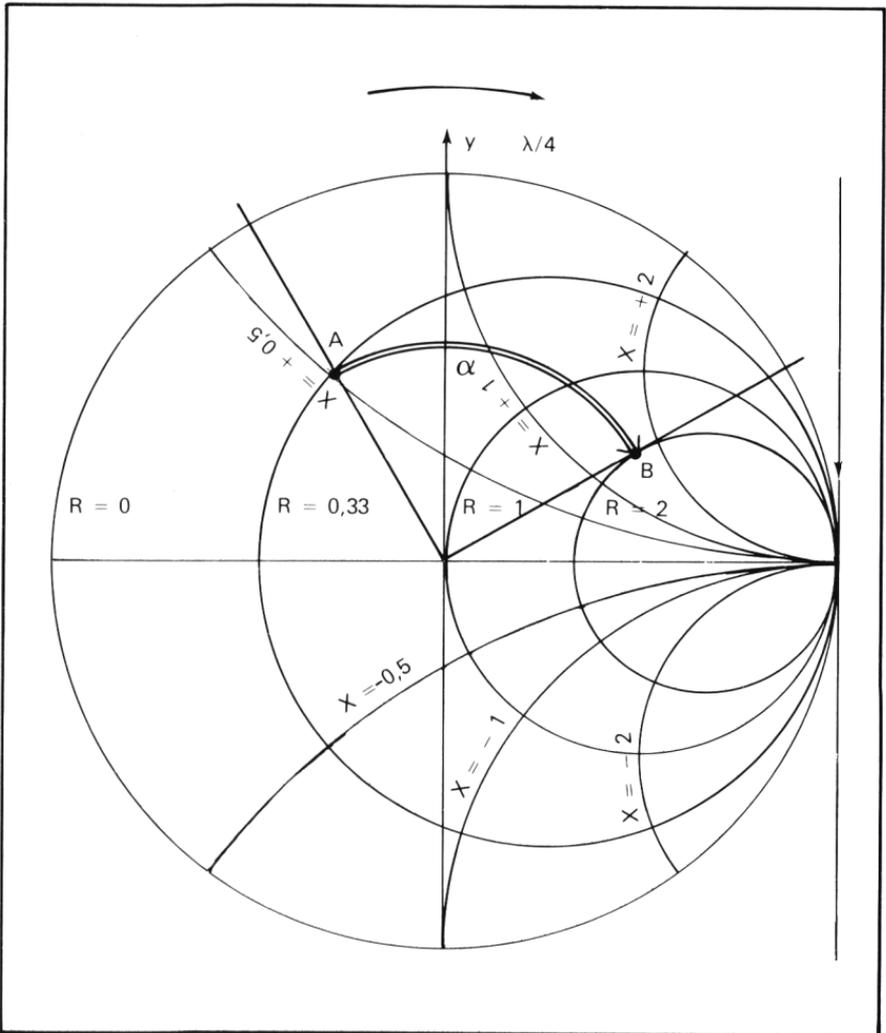
— une rotation d'angle θ dans ce plan amène aux nouvelles coordonnées :

$$\begin{cases} x_1 = x_0 \cos \theta - y_0 \sin \theta \\ y_1 = x_0 \sin \theta + y_0 \cos \theta \end{cases}$$

— une transformation inverse conduit aux résultats :

$$X' = \frac{2 y_1}{x_1^2 + y_1^2 - 2 x_1 + 1}$$

$$R' = \frac{x' (1 - x_1)}{y_1} - 1$$



LE PROGRAMME propose une partie « graphique » et une partie « numérique ». La partie graphique n'est pas indispensable, mais permet d'une part, de rendre moins austère l'utilisation de ce logiciel, d'autre part de vérifier la position des points correspondant aux impédances sur l'abaque. Le point de départ (A) et le point d'arrivée (B) sont indiqués sous forme d'astérisque. Pour rendre plus lisible l'abaque, il n'y a que 5 cercles de résistance, ainsi que 6 arcs de cercle de réactance. Il faut fournir au programme, lorsqu'il le demande, l'impédance caractéristique du câble employé, la résistance et la réactance du récepteur ainsi que la longueur du câble exprimé en longueur d'onde λ

(rappelons que $\lambda = \frac{C}{f}$ avec C = vitesse des ondes

électromagnétiques dans le câble et f = fréquence de travail).

Le programme fournit alors les valeurs (en ohms) de la résistance et de la réactance de l'impédance vue par le générateur.

- Lignes 10 à 300 : partie graphique.
 - 20 à 70 : dessin des cercles correspondant aux résistances.
 - 80 à 190 : dessin des arcs de cercle correspondant aux réactances négatives.
 - 200 à 300 : idem pour les réactances positives.
 - 310 à 470 : partie numérique.
 - 320 à 345 : entrée des données.
 - 350 à 390 : calcul des coordonnées correspondant à l'impédance du récepteur ; disposition du point sur l'écran.
 - 400 à 410 : rotation.
 - 420 à 430 : calcul de l'impédance réduite vue du générateur.
 - 440 - 450 : disposition du point correspondant sur l'écran.
 - 460 - 470 : affichage des résultats sur l'écran.

Variables

- N : variable de boucle non significative.
- R : résistance réduite (tracé des cercles).
- X : réactance réduite (tracé des courbes).
- T : valeur de l'angle de rotation correspondant aux arcs de cercles relatifs aux réactances.

- R : résistance de la charge en ohms
- X : réactance de la charge en ohms
- ZC : impédance caractéristique de la ligne en ohms
- T : longueur du câble exprimé en longueurs d'onde.
- X, Y (lignes 360 à 370) : position du point de départ sur l'abaque.
- XN, YN : positions du point d'arrivée sur l'abaque.
- NR, NX : résistance et réactance réduite de l'impédance vue par le générateur.

```

1 REM*****
2 REM* *
3 REM* ABAQUE DE SMITH *
4 REM* *
5 REM*****
10 HIRES
20 CURSET100,90,1:CIRCLE90,1
30 FORN=1 TO 5
40 READ R
50 CURSET100+90*(R/(R+1)),90,1:CIRCLE90/(R+1),1
60 NEXTN
70 DATA 0.33,0.66,1,2,4
80 FORN=1 TO 3
90 READ X:READT0
110 T=T0
120 REPEAT
130 CURSET190-90*(COS(T)/X),90+(90/X)-(90*(SIN(T)/
X),1
150 T=T+PI/128
160 UNTIL T >= PI/2
170 DATA 2,-0.638136,1,0.024543,0.5,0.613592
190 NEXTN
200 FOR N=1 TO 3
210 READ X:READT0
220 T=-PI/2
230 REPEAT
240 CURSET 190-90*(COS(T)/X),90-(90/X)-(90*(SIN(T)/
X),1
250 T=T+PI/128
270 UNTIL T >= T0
290 NEXT N
300 DATA 2,0.662679,1,0,0.5,-0.613592
310 POKE28,0
320 INPUT"Impedance caracteristique du cable ";
ZC

```

```

330 INPUT "resistance de la charge ";R
340 INPUT "reactance de la charge ";X
345 INPUT "Longueur du cable en longueurs d'onde ";T:T=-4*PI*T
350 R=R/ZC:X=X/ZC
360 Y=2*X/((X^2)+(R+1)^2)
370 X=1-Y*(R+1)/X
380 CURSET90+(90*X)-3,100-(90*Y)-3,1
390 CHAR42,0,1
400 XN=X*COS(T)-Y*SIN(T)
410 YN=X*SIN(T)+Y*COS(T)
420 NX=2*YN/(XN^2+YN^2-(2*XN)+1)
430 NR=(NX*(1-XN)/YN)-1
440 CURSET90+(90*XN)-3,100-(90*YN)-3,1
450 CHAR42,0,1
460 PRINT"La resistance vue du generateur est";
NR*ZC;" ohms ";
470 PRINT"et la reactance est ";NX*ZC;" ohms .";
475 GETA#
480 STOP
1000 CALL #E6CA
1010 LLIST

```

Impédance du récepteur	Impédance vue par le générateur
.5 + j .	1.14 + j 1.68
.6 + j .	1.34 + j 1.57
.7 + j .	1.52 + j 1.44
.8 + j .	1.69 + j 1.31
.9 + j .	1.83 + j 1.16
1 + j .	1.96 + j 1.01
1.1 + j .	2.08 + j .85
1.2 + j .	2.18 + j .7
1.3 + j .	2.26 + j .54
1.4 + j .	2.33 + j .38
1.5 + j .	2.39 + j .23
1.6 + j .	2.43 + j .08
1.7 + j .	2.47 + j -.06
1.8 + j .	2.49 + j -.2
1.9 + j .	2.51 + j -.34

Longueur de la ligne = 0,05 λ .

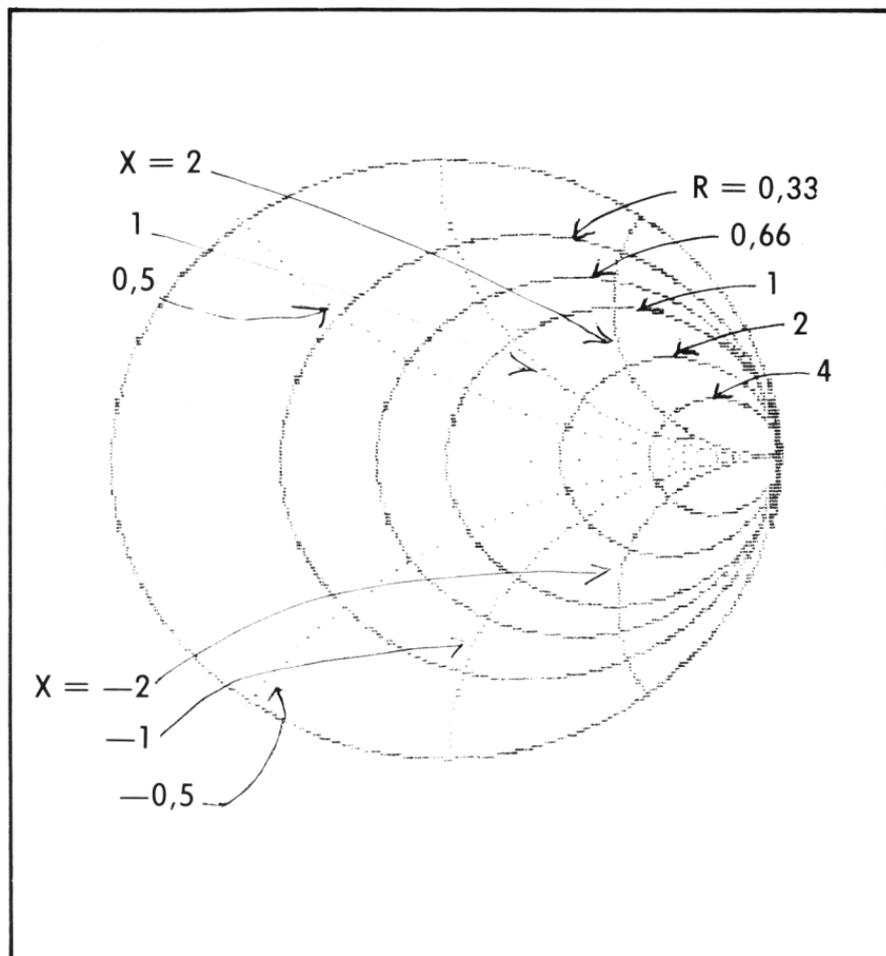
Exemple de modification de programme (Pour l'imprimante).

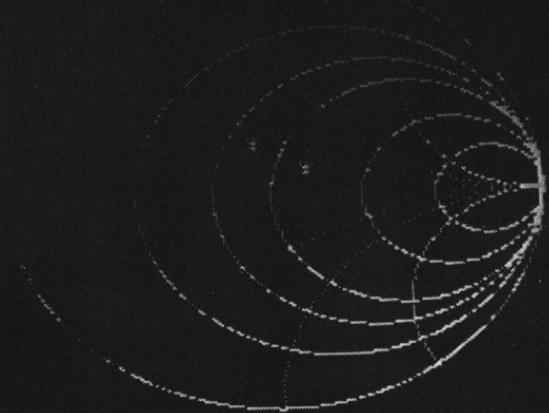
```
2000 X=1:T=-0.05*4*PI
2010 FOR R=0.5 TO 2 STEP 0.1
2020 GOSUB 360
2030 LPRINT R;" + j .",INT(NR*100)/100;" + j ";
INT(NX*100)/100
2035 X=1
2040 NEXT R
```

Programme modifié.

(supprimer 380 et 390 et rajouter 435 return). Faire RUN 2000.

Recopie d'écran.





La resistance vue du generateur est 145
.82 ohms et la reactance est 77.47 o
hms .

Annexe 1

PLL : PRINCIPE

CONSTITUTION D'UNE BOUCLE A VERROUILLAGE DE PHASE

Une boucle se compose de quatre parties :

(x) un comparateur de phase (CP), chargé d'élaborer la différence entre les phases de deux tensions (figure 1).

Suivant les cas, la caractéristique de transfert peut être triangulaire (figure 2) ou sinusoïdale (figure 3).

En régime linéaire, nous écrivons dans les deux cas :

$$K_p = V_1 / \varphi = \frac{V_{dd}/2}{\pi/2} = V_{dd}/\pi$$

En régime non linéaire, la caractéristique figure 2, discontinue, ne se prête que difficilement à une exploitation mathématique ; nous adopterons dans ce cas l'équation de la caractéristique figure 3 soit :

$$V_1 = V_{dd}/2 \sin(\phi_i - \phi_r) = V_{dd}/2 \sin \varphi .$$

(x) un filtre passe-bas, chargé de :

- définir la zone de capture F_c (voir plus loin)
- assurer la stabilité du système
- filtrer la composante à $\omega_i + \omega_r$, générée en même temps que le signal utile dans le CP.

(x) un VCO, oscillateur commandé en tension, délivrant une tension de sortie, de fréquence proportionnelle à sa tension d'entrée. Sa transmittance est appelée K_o .

La définition des différents termes est donnée figure 4.

(x) un intégrateur, « immatériel », dans la mesure où il n'a aucune présence physique, signalé \int dans le schéma figure 4. En effet, la grandeur de sortie du VCO est la pulsation d'un signal dont on utilise à l'entrée de la boucle la phase, c'est-à-dire l'intégrale par rapport au temps de la pulsation : $\phi_r = \int \omega_r dt$.

FONCTIONNEMENT DE LA BOUCLE

1° Régime transitoire : cas d'un échelon de fréquence.

En l'absence de signal d'entrée, la boucle délivre sa pulsation centrale, soit ω_0 . A l'apparition d'un signal, tel que sa pulsation soit comprise entre $\omega_0 - \omega_L$ et $\omega_0 + \omega_L$, il naît, à la sortie du comparateur

une tension, variable, fonction de la différence des phases des signaux d'entrée et de réaction, de la forme : $v_1 = K_p (\phi_i - \phi_r)$. Cette fonction croît dans le temps si $\omega_i > \omega_0$ (en négligeant bien sûr toute non-linéarité pour le moment). En l'absence de filtre passe-bas, on a : $v_2 = v_1 (> 0)$ et donc le VCO voit sa pulsation de sortie augmenter et passer à ω_{NO} . Ceci se traduit par une moindre augmentation de $v_1 (= K_p [\omega_i t - \omega_{NO} t])$, mais une nouvelle croissance de ω_{NO} . Ce phénomène se poursuit jusqu'à la capture, moment précis où $\omega_{NO} = \omega_i$; à cet instant là, v_1 est devenu constant, et correspond à un certain déphasage (constant) entre v_i et v_r . La boucle est verrouillée.

En pratique, ce régime transitoire n'est pas aussi simple que cela, pour plusieurs raisons.

2° Les imperfections d'une boucle réelle.

(x) Il y a d'abord un terme parasite à la sortie du comparateur de phase dû à son principe de fonctionnement même. En effet, que celui-ci soit constitué d'un OU exclusif ou bien d'un multiplieur différentiel, le signal d'erreur est constitué du produit des 2 tensions v_i et v_r :

$$v_i = V_i \sin \omega_i t$$

$$v_r = V_r \sin \omega_r t$$

$$v_i v_r = \frac{V_i V_r}{2} \left[\sin (\omega_i t - \omega_r t + \frac{\pi}{2}) - \cos (\omega_i t + \omega_r t) \right]$$

nous voyons donc apparaître un terme en $\omega_r + \omega_i$, en plus du terme utile en $\omega_i - \omega_r$. Il se superpose donc au signal d'erreur v_1 . Cependant, en régime permanent, il ne modifie pas la pureté spectrale de v_r , dans la mesure où sa valeur moyenne est nulle sur une demi-période de v_r .

Ensuite, le comparateur de phase n'a pas la caractéristique rectiligne indiquée figure 4. Au-delà de $\pi/2$, il y a en effet « retournement » de la caractéristique $v_1 (\varphi)$, que celle-ci soit linéaire (OU exclusif) ou sinusoïdale : la réaction devient alors positive jusqu'à $3\pi/2$, négative jusqu'à $5\pi/2$, etc. Le système n'a cependant pas de point de fonctionnement stable dans les zones à réaction positive : il s'en éjecte de lui-même, pour se retrouver dans une zone à réaction négative contiguë.

(x) Enfin, la dernière difficulté provient du fait que, lors d'un échelon de fréquence par exemple, on ne maîtrise que rarement la phase ini-

tiale avec laquelle se présente le signal à $t = 0$. Dans ce cas, seule une étude statistique permet de prévoir les limites des lieux de $v_1(t)$ et de connaître ainsi les valeurs extrêmes que peuvent prendre les paramètres de fonctionnement, tels que temps de réponse, etc.

3° Le filtre passe-bas.

Bien que la réponse de la boucle soit satisfaisante en l'absence de filtre, il est presque nécessaire d'en inclure un dans la boucle et ceci pour plusieurs raisons.

Remarquons d'abord que, contrairement à beaucoup de systèmes asservis, on ne s'intéresse ici que peu au signal de sortie v_r . On utilise en fait les tensions V_1 ou V_2 qui sont les images, en régime établi, de la fréquence du signal d'entrée v_i . Or, sans filtre, on a pu constater précédemment que, au signal utile, se superpose une ondulation de pulsation $\omega_i + \omega_r$. Celle-ci perturbe le régime transitoire et doit de toutes façons être éliminée de V_1 . Elle constituerait un « bruit » inadmissible en démodulation de fréquence. L'utilisation d'un filtre a donc pour but de faire disparaître ce signal gênant et, par là même, tout signal interférant parasite. Cependant, ce filtre ne doit pas pour autant faire disparaître l'erreur « utile » en $\omega_i - \omega_r$, responsable du verrouillage de la boucle : d'ailleurs, son atténuation est responsable de l'existence d'un nouveau paramètre, appelé fréquence de capture (f_c). Pour qu'il y ait verrouillage, il faut que la fréquence incidente (f_i) soit comprise dans un domaine de largeur $2f_c$ entourant f_o . Cette contrainte disparaît dès qu'il y a eu verrouillage. Le choix de la constante du filtre agit donc directement sur f_c .

Enfin, le fait d'introduire un tel circuit dans la boucle la transforme en un système du second ordre : il faudra alors prendre garde à la stabilité.

BOUCLE SANS FILTRE PASSE-BAS

Le schéma est alors très simple (voir figure 1). En boucle fermée, le système est du premier ordre et donc toujours stable.

1° Régime linéaire.

Soit V_{dd}/π la transmittance du comparateur de phase.

Soit $2\omega_l/V_{dd}$ la transmittance du VCO.

Le gain de boucle s'écrit :

$$\phi_r / \phi_i - \phi_r = \frac{V_{dd}}{\pi} \times \frac{2\omega_l}{V_{dd}} \times \frac{1}{p} = \frac{2\omega_l}{\pi} \times \frac{1}{p}$$

Le gain en boucle fermée est donc :

$$\phi_r / \phi_i = \frac{2\omega_l / \pi}{1 + p\pi / 2\omega_l}$$

La constante de temps de ce système est $\tau = \pi / 2\omega_l$ et le temps de réponse à 5 % vaut $3\tau = \frac{3\pi}{2\omega_l}$. On peut éliminer la fonction SIN de la

ligne 220 pour étudier la boucle dans cette hypothèse.

2° Régime non linéaire.

C'est là naturellement le fonctionnement normal de la boucle : la résolution numérique présente tout son intérêt. Le programme permet en effet de tracer la réponse à un échelon de fréquence, en tenant compte de la phase initiale aléatoire avec laquelle se présente l'entrée v_i à $t = 0$. Il y a plusieurs possibilités d'étude :

(x) Tracer la réponse pour plusieurs valeurs fixées de cette phase (PHI en degrés) : $0^\circ, 30^\circ, \dots$. Comme prévu, tant que PHI est compris entre -90° et $+90^\circ$, le système démarre d'une zone à réaction négative et à un comportement linéaire ; en dehors de cet intervalle, la réaction est d'abord positive ; cela se traduit au départ par un écart à la valeur finale grandissant dans le temps, puis passage de V_i à la valeur -1 , enfin retour dans une zone à réaction négative. Le phénomène est spectaculaire pour $PHI = 100^\circ$!

(x) Tracer la réponse pour des valeurs aléatoires de PHI et superposer sur l'écran toutes les trajectoires correspondantes. Cela permet ainsi de mesurer, dans les conditions les plus défavorables, le temps de réponse de la boucle. Il faut alors remplacer la ligne 193 par :

```
193 LET PHI = - 360 * RND + 180
```

(x) Il est possible enfin de calculer par programme le temps de réponse de la boucle pour des valeurs aléatoires de PHI, puis de tracer un diagramme à barres permettant de connaître le temps de réponse le plus probable.

PROGRAMME

```
5 REM "BOUCLES A VERROUILLAGE DE PHASE"  
10 LET PHI = 0  
20 LET V1 = 0  
30 LET F0 = 50 E 3  
40 LET FI = 69 E 3
```

```

50LET FL = 20 E 3
60 LET FN0 = F0
70 LET DT = 1/(8 FI)
90 REM "TRACE DE L'AXE OY"
100 FOR I = 0 TO 43
110 PLOT 0, I
120 IF I = 40 THEN PRINT "+I"
130 IF I = 0 THEN PRINT "-I"
135 NEXT I
138 REM "TRACE DE L'AXE OX"
140 FOR I = 0 TO 63
150 PLOT I, 20
160 FOR N = 0 TO 7
170 IF I = N * 8 THEN PLOT I, 21
180 NEXT N
190 NEXT I
193 LET PHI =
195 REM "PHI EST LA PHASE INITIALE EN DEGRES"
198 LET PHIRAD = PHI * PI/180
200 FOR I = 0 TO 63
210 LET PHIRAD = 2 * DT * PI * (FI - FN0) + PHIRAD
220 LET V1 = SIN (PHIRAD)
230 LET FN0 = FL * V1 + F0
240 PLOT I, 20 * V1 + 20
250 NEXT I
252 LET V1 = 0
254 LET FN0 = F0
255 STOP ou 255 GO TO 193

```

REMARQUES SUR LE PROGRAMME

1° Comme on a pu le constater, nous avons éliminé, par souci de simplification, l'état de repos de la boucle. En pratique, un comparateur de phase est constitué par un système qui délivre au repos (fréquence d'entrée $F_i =$ fréquence centrale du VCO $= F_o$, ou bien pas de signal d'entrée) une tension de valeur moyenne égale à la moitié de la tension d'alimentation V_{dd} . Ceci correspond à un déphasage de $\pi/2$ entre les 2 entrées du CP. Nous avons donc posé, à l'équilibre, $V_1 = V_2 = 0$ (au lieu de $V_{dd}/2$).

$\varphi = 0$ (au lieu de $\pi/2$). Les notations utilisées sont les mêmes que dans le texte, sauf :

PHI = phase initiale en degrés (devient PHIRAD en radians)

FNØ = fréquence instantanée du VCO

DT = intervalle de calcul, pris égal à $1/8$ de la période $1/F_i$ du signal d'entrée.

2° L'exemple proposé est le suivant :

$$F_0 = 50 \text{ kHz}$$

$$F_L = 20 \text{ kHz}$$

$F_i = 69 \text{ kHz}$ (valeur quelconque entre $50 - 20 = 30 \text{ kHz}$ et $50 + 20 = 70 \text{ kHz}$ pour qu'il y ait verrouillage).

3° On constate que les résultats « linéaires » sont à peu près applicables lorsque la phase initiale est comprise entre -90° et $+90^\circ$. Dans ces cas, en effet le système part d'une zone où il y a effectivement contre réaction ($\Delta V_i / \Delta \varphi > 0$) et donc n'a aucune raison d'en sortir. Par contre, si $-270^\circ < \varphi < -90^\circ$ et $+90^\circ < \varphi < +270^\circ$, le système voit son gain dynamique changer de signe ($\Delta V_i / \Delta \varphi < 0$) : il y a réaction positive. Sur l'écran, on constate que le point de fonctionnement s'éloigne d'abord de l'équilibre final (c'est-à-dire $V_i = 0,95$ dans l'exemple numérique choisi), passe par $V_i = -1$ (soit : $\varphi = 270^\circ$), puis retourne au régime linéaire, avec une phase initiale à ce moment-là de $+270^\circ$. Dans de tels cas, le temps de verrouillage peut atteindre 8 cycles de F_i .

4° Nous pouvons également remarquer que, même en régime de réaction négative, le temps de verrouillage de la boucle est plus important que ne le prévoit la théorie de son fonctionnement linéaire. En effet, dans ce cas, et à cause de la caractéristique sinusoïdale du CP, le gain dynamique de la boucle diminue et tend vers 0 lorsque F_i se rapproche de $F_0 + F_L$ (ou $F_0 - F_L$) car, alors la limite finale de V_i est $+1$ (ou -1), valeur pour laquelle la pente de la fonction sinus tend vers 0.

LEGENDES DES COURBES, FIGURES 6 ET 7

FIGURE 6 : Réponse de la boucle sans filtre passe-bas à un échelon de fréquence : $F_i = 69 \text{ kHz}$, correspondant à une valeur finale de V_i égale à 0,95. L'axe des temps est gradué en périodes de la tension d'entrée, soit $1/F_i$; l'axe V_i de -1 à $+1$. La courbe (notée $\varphi' = 0$) correspond à une boucle dont le comparateur de phase est à caractéristique triangulaire (système linéaire).

FIGURE 7 : Même légende que figure 6, mais courbes correspondant cette fois-ci à une phase initiale de la tension d'entrée comprise entre $+90^\circ$ et $+270^\circ$. Il y a réaction positive au départ dans tous les cas. La figure 5 résume les conclusions relatives à ces deux courbes.

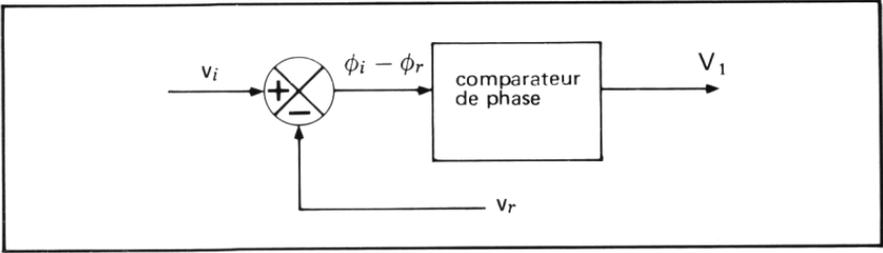


Figure 1 : En posant
 $v_i = V_i \sin (\omega_i t + \varphi_i)$
 $v_i = V_i \sin \phi_i$
 et $v_r = V_r \sin \omega_r t$
 $v_r = V_r \sin \phi_r$
 on a $V_1 = K_p \phi_i - \phi_r$.

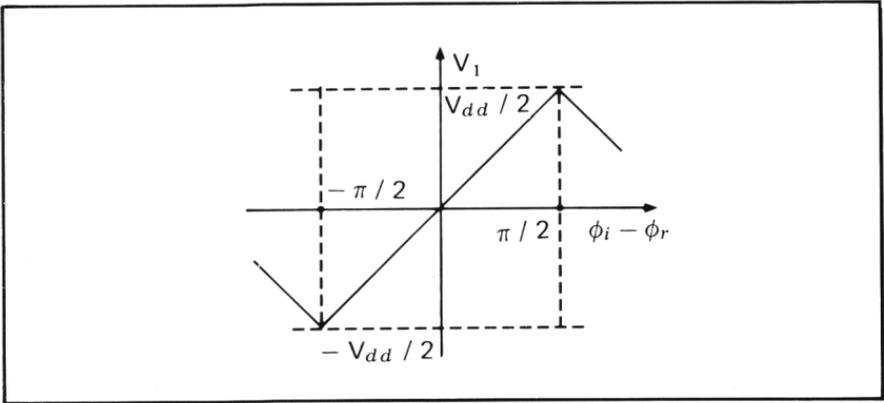


Figure 2 : Comparateur de phase à caractéristique triangulaire.

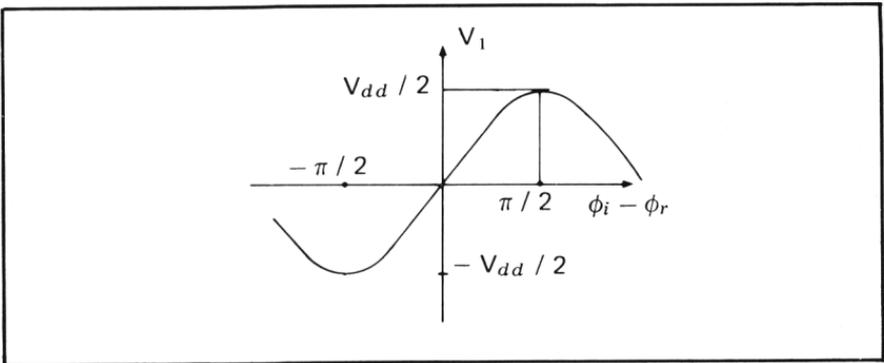


Figure 3 : Comparateur de phase à caractéristique sinusoïdale.

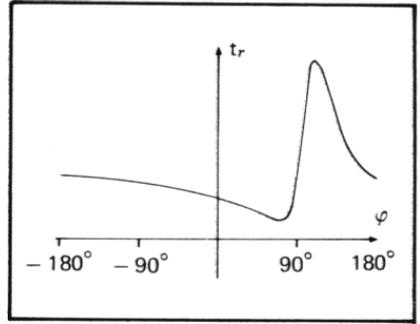
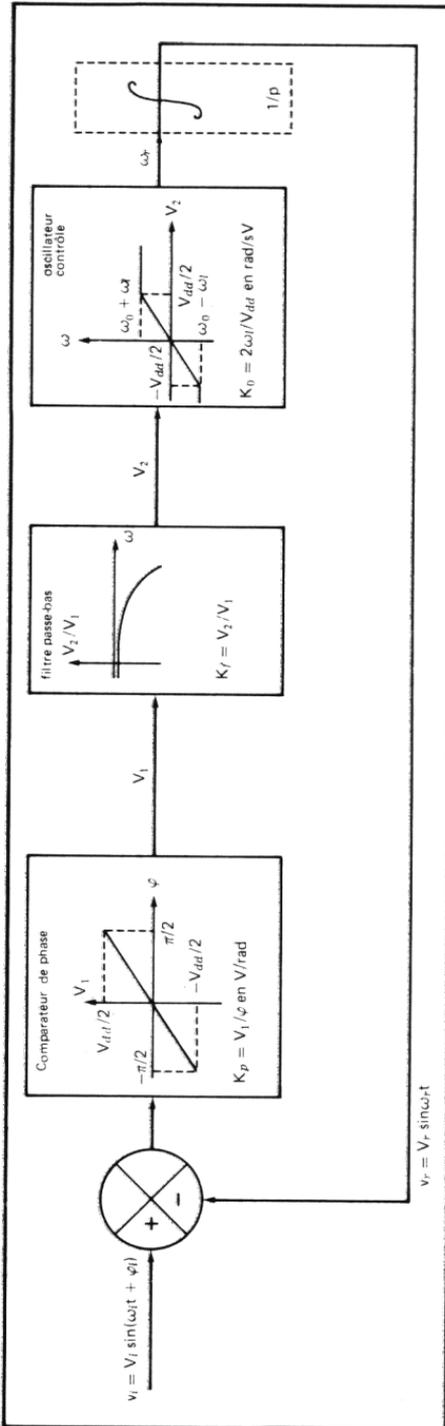
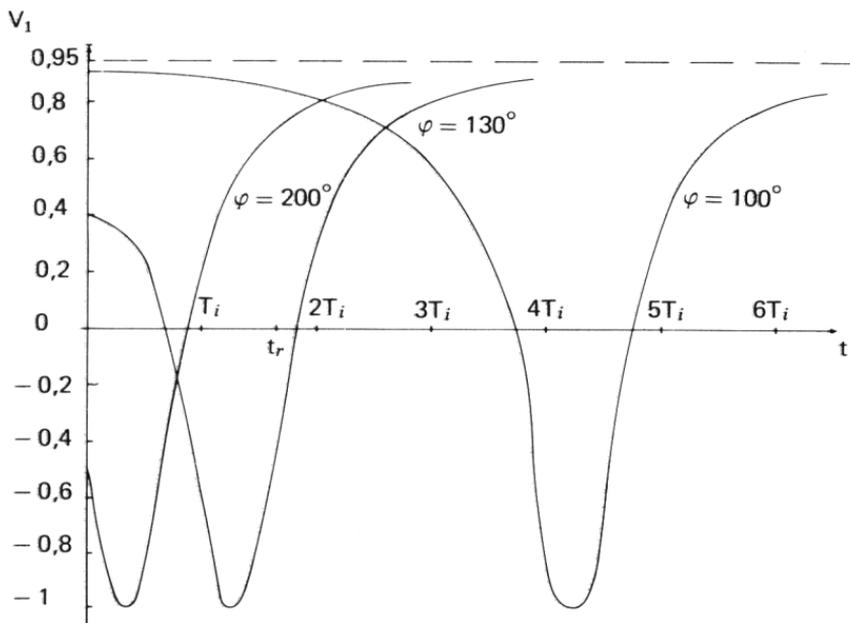
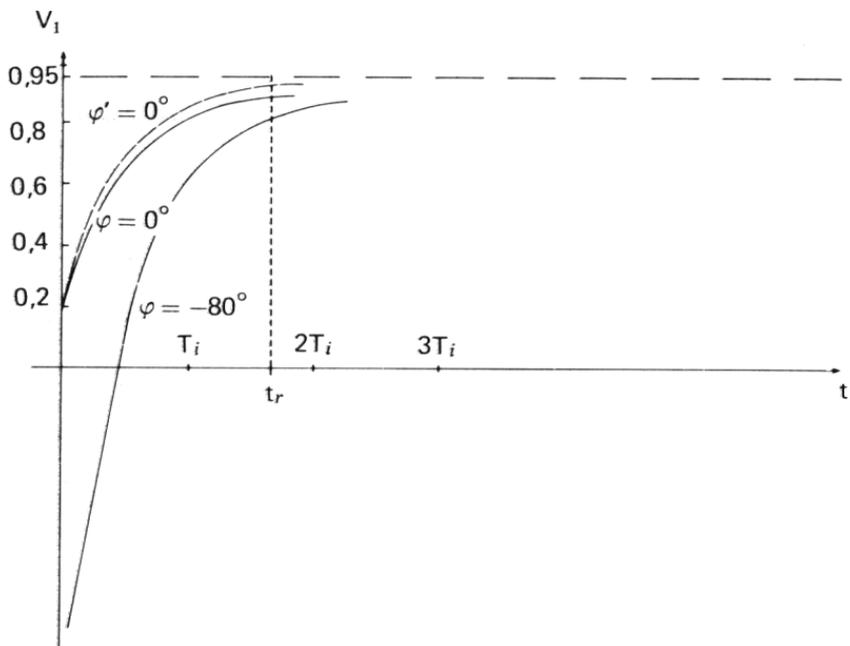


Figure 5 : Temps de réponse de la boucle en fonction de la phase initiale.

Figure 4 : Schéma synoptique d'une boucle à verrouillage de phase.



CONCLUSION

Cette introduction à l'étude des régimes transitoires dans les boucles à verrouillage de phase nous a permis de comprendre le rôle des différents éléments qui les composent. En particulier, nous avons vu l'influence du comparateur de phase dont la caractéristique sinusoïdale transforme le système en une succession de sous-ensembles, les uns à réaction positive, les autres à réaction négative.

Cependant, ce mode d'utilisation d'une boucle (c'est-à-dire sans filtre passe-bas) est assez peu répandu : n'oublions pas que, pour un tel système, la tension de sortie v_2 est constituée d'une composante continue, image de la fréquence instantanée du VCO, et du signal en $f_i + f_r$ non éliminé. En régime permanent, le signal issu du VCO a, à un déphasage près, la fréquence du signal d'entrée. Un tel fonctionnement peut cependant être utile pour piloter une horloge, par exemple, le VCO prenant alors le relais lors des absences du « pilote » V_i .

Annexe 2

UTILISATION DE L'IMPRIMANTE

L'INSTRUCTION LPRINT D'ORIC

Oric ne dispose que de deux commandes destinées à l'imprimante : LPRINT ET LLIST. Ce sont les seuls moyens « d'action » (à défaut du langage machine qu'il est prématuré d'envisager). En pratique, toutes nos communications se feront à l'aide de la première. Dans le mode normal, il est facile de constater que l'on peut effectivement imprimer des caractères ou des variables mais, première surprise, on ne peut en mettre que 67 par ligne (la GP 100 en permet 80). Ce 67 est-il lié au TAB(13)* ? C'est sans doute le cas. On peut tourner la difficulté en pokant un nombre en 49. En effet, POKE 49,93 permet maintenant d'imprimer 80 caractères par ligne. A la mise sous tension, il est facile de constater que 49 contient 80, qui est le nombre maximum de caractères par instruction. Charger 93 ne modifiera pas en fait ce nombre, mais simplement la longueur des lignes imprimées. On peut d'ailleurs laisser les choses en l'état, dans la mesure où il existe une largeur de papier correspondant à 67 car/l. Il sera par contre indispensable de modifier la valeur de la mémoire 49 pour travailler en mode graphique. Signalons tout de suite que 255 permet d'obtenir 242 impressions par ligne. Quand nous disons ceci, il est sous entendu que cela signifie qu'il est possible de LPRINTER 242 caractères avant qu'ORIC n'envoie un signal de retour de chariot à l'imprimante. Il nous est possible d'envoyer nous-mêmes un tel signal, afin de réaliser un retour anticipé à la ligne.

L'IMPRIMANTE SEIKOSHA GP 100

Il s'agit d'une imprimante parallèle type centronics. Elle génère les caractères sous forme normale ou en double largeur. Elle dispose de possibilités graphiques très intéressantes. Le choix du mode d'impression se fait par les caractères de contrôle, que nous allons examiner. Ceux-ci sont fournis à l'imprimante par l'instruction LPRINT CHR\$(#XX), dans laquelle XX est le code du caractère de contrôle correspondant exprimé en hexadécimal. Il est curieux de constater à cet effet que #0D, par exemple, est assimilé à un nombre par ORIC, alors que HEX\$(13), qui vaut # 0D, est une chaîne ! Ne pas tenter d'utiliser HEX\$!

* TAB (I) est inopérant pour I compris entre 1 et 13.

MODES D'IMPRESSION

Il y en a 3 qui sont mixables sur une même ligne.

§ # 0 : Mode caractère. C'est celui que fournit sans doute, naturellement, ORIC. Il est ainsi possible d'imprimer des caractères définis par leur code ASCII. ORIC connaît celui des lettres courantes et l'on n'a pas à s'en soucier. Cependant, pourquoi se priver des lettres accentuées (par exemple) ? Il suffit de consulter la notice de l'imprimante pour trouver le code correspondant à ces lettres et utiliser alors à leur place l'expression CHR\$(N), dans laquelle N est justement ce code. Dans ce type d'opération, le marteau se déplace de gauche à droite et imprime les différents caractères sous la forme de barres verticales, constituées de 7 points chacune. Les espacements verticaux et horizontaux sont automatiques.

§ # 0E : Mode double largeur. Dans ce cas, les lettres sont deux fois plus larges qu'en impression normale. On ne peut donc en imprimer que deux fois moins par ligne. POKE 49,53 résout le problème du retour à la ligne.

§ # 08 : Mode graphique. Dans ce cas, le type d'impression reste évidemment le même (le chariot va de gauche à droite, entraînant son marteau), mais il peut imprimer 480 colonnes de 7 points de hauteur (que nous appellerons septets pour simplifier) sur une ligne. Il n'y a pas d'espacement entre ces différents septets, ni horizontalement, ni verticalement. Le code à fournir est alors le suivant : chaque point d'un septet est pondéré, le point supérieur vaut 1, le point inférieur vaut 64. La somme de ces nombres, pour un septet donné, augmentée de 128, est le code à fournir dans l'instruction LPRINT CHR\$(). Il y a 480 septets par ligne. Au prix d'un certain logiciel, il serait ainsi possible de redéfinir les caractères de l'imprimante. Pourquoi ne pas lister vos programmes en japonais ?

L'ADRESSAGE D'IMPRESSION

Il existe deux possibilités. Cependant, quelle que soit celle choisie, il faut bien voir la puissance des caractères de contrôle correspondant. En effet, sur une ligne, on peut spécifier une abscisse pour chaque motif (caractère ou septet), mais on n'est pas obligé de les fournir à l'imprimante dans l'ordre d'impression (de gauche à droite). Ils peuvent être transmis dans un ordre quelconque, l'imprimante se chargeant de réorganiser la ligne au moment de l'impression. Bien sûr, si cette possibilité n'est pas utilisée, l'impression se fait normalement,

les motifs s'affichant les uns à la suite des autres.

§ #10 : Ce code doit être suivi de 2 octets, chacun étant le code ascii des 2 chiffres précisant la position d'impression d'un caractère. Ainsi, pour imprimer un A à la 37^e position (rappelons qu'il y a 80 positions par ligne), il faudra écrire : LPRINT CHR\$ (#10) ; CHR\$ (51) ; CHR\$ (55) ; «A» ; ou, plus simplement : LPRINT CHR\$ (#10) ; «37A».

§ #1B;10 : Adressage par point, utilisé en mode graphique. Il y a 480 septets par ligne et l'on peut adresser chacun d'eux. Dans ce cas, les caractères de contrôle sont suivis de la position choisie, exprimée en hexadécimal, sur 2 octets (puisque 480 est supérieur à 255). Par exemple, le septet d'abscisse 330 est adressable par : LPRINT CHR\$ (#1B) ; CHR\$ (#10) ; CHR\$ (1) ; CHR\$ (74) ; remarquons que la position peut être exprimée en décimal. (on a : $330 = 1 \times 256 + 74$).

LES RETOURS A LA LIGNE

Il y a 3 codes possibles, mais deux sont identiques. Nous verrons qu'il existe en fait 4 possibilités. Nous supposons dans ce § que 49 contient 80.

§ #0A : Il ne présente aucun intérêt. En effet, s'il provoque bien un retour à la ligne (par LPRINT CHR\$ (#0A) ;), ORIC ne perd pas pour autant le compte des caractères expédiés à l'imprimante. Arrivé le 67^e, il déclenche à son tour le retour à la ligne. Ainsi, si l'on a expédié ce code de contrôle après 50 caractères, la ligne suivante n'en comportera plus que 17 !

Quant à LPRINT CHR\$ (#0A), (sans ;), il provoque un espacement entre 2 lignes consécutives. Le phénomène précédent n'existe plus, dans la mesure où l'absence de ; a obligé ORIC à fournir son propre signal de retour à la ligne.

§ #0D ou #14 : ces 2 codes sont équivalents.

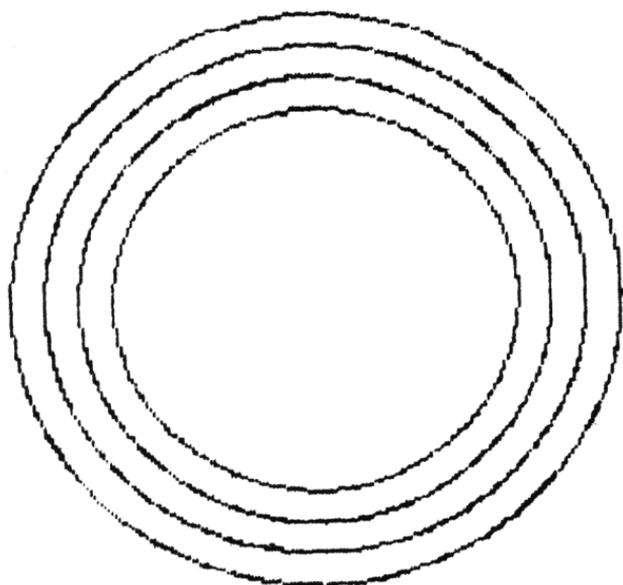
LPRINT CHR\$ (#0D); permet d'obtenir un curieux phénomène. Il entraîne dans un premier temps un retour de chariot sans mise à la ligne suivante. Au 67^e caractère, il y a retour à la ligne. Ceci est évidemment inexploitable.

Enfin (et heureusement), on s'aperçoit que LPRINT CHR\$ (#0D) a tous les avantages :

- Il impose un retour à la ligne.
- Il oblige ORIC à fournir le sien.
- L'imprimante confond ces 2 ordres et va simplement à la ligne suivante.

Le dessin obtenu est parfait : il a la même définition que celui de l'écran. Nous verrons par la suite comment obtenir un résultat semblable, au prix d'une légère perte de qualité cependant, mais beaucoup plus rapidement.

Programme n° 1.



```
1 REM*****
2 REM*
3 REM* RECOPIE LENTE
4 REM* D'ECRAN
5 REM* HAUTE RESOLUTION
6 REM* POUR ORIC
7 REM*
8 REM*****
10 REM RECOPIE D'ECRAN
20 REMCOPYRIGHT PM BEAUFILS
30 REM EXEMPLE DE DESSIN
40 HIRES
50 CURSET110,100,1
60 FORN=0 TO 4:CIRCLE 50+10*N,1:NEXT
70 REM COPIE LENTE
80 POKE49,255
```

```

90 FOR N=0 TO 6:BC(N)=2^N:NEXT
100 LPRINT CHR$(#08)
110 FOR Y=1 TO 199 STEP 7
120 FOR X=1 TO 239
130 A1=0
140 FOR N=0 TO 6
150 Y1=Y+N
160 IF Y1 > 199 THEN 200
170 K=POINT(X,Y1)
180 IF K = 0 THEN 200
190 A1=A1+K*BC(N)
200 NEXT N
210 A1=128+ABS(A1)
220 LPRINT CHR$(A1);
230 NEXT X
240 LPRINT CHR$(#0D)
250 NEXT Y
260 LPRINT CHR$(#0F):POKE49,80
270 REM FIN COPIE LENTE
280 STOP

```

TRACE DIRECT DES COURBES $Y = F(X)$

Nous allons utiliser ici les possibilités d'adressage en mode graphique. (figure 2).

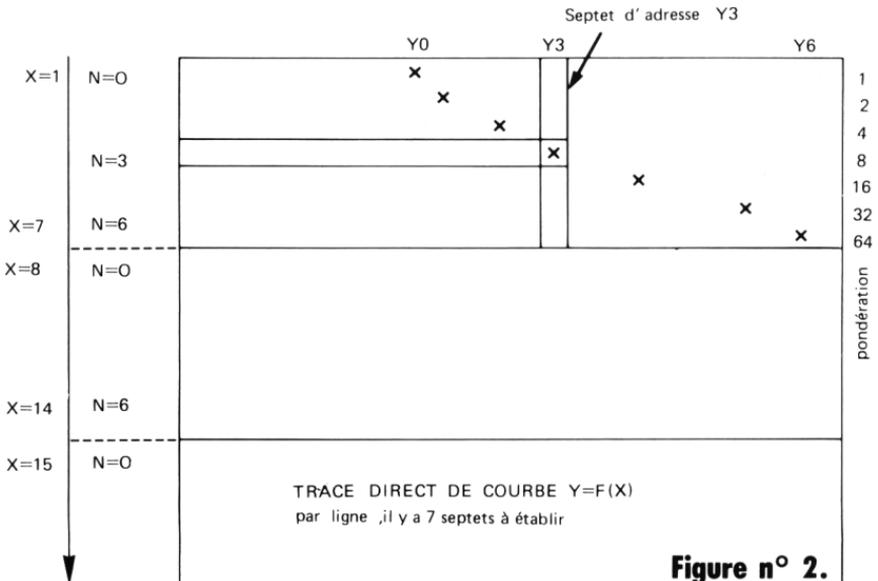
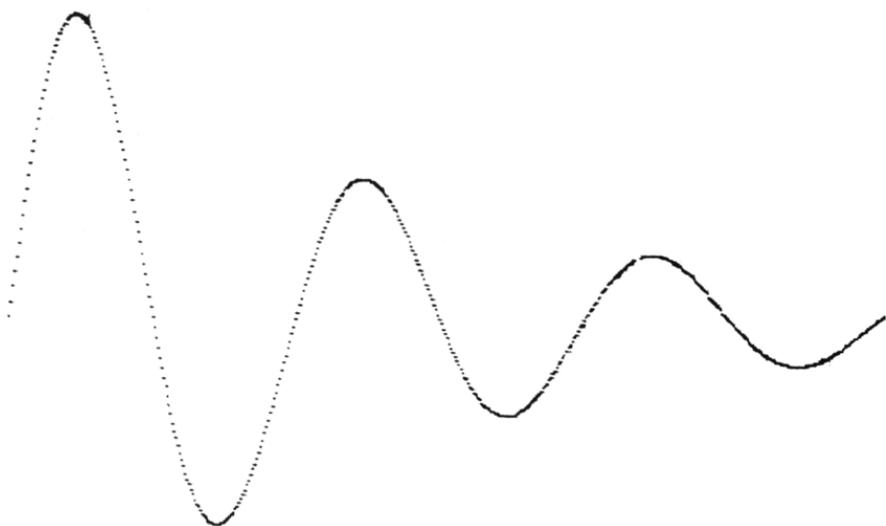


Figure n° 2.

En effet, pour chaque ligne, il n'y a que 7 septets à réaliser, un pour chaque valeur de X. Y est alors la position d'impression. Pour $N = 3$, il faudra par exemple envoyer : LPRINT CHR\$(#1B); CHR\$(#10); CHR\$(A); CHR\$(B); CHR\$(136);

A et B représentant la position Y exprimée en hexadécimal. X varie ainsi de 0 à 6, par l'intermédiaire de N, sur chaque ligne de septets.

Programme n° 2.



```

1 REM*****
2 REM*                                     *
3 REM* TRACE DE COURBES                   *
4 REM* EN HAUTE RESOLUTION                *
5 REM*                                     *
6 REM*****
10 LPRINT CHR$(#08)
20 FOR X=0 TO 450 STEP 7
30 FOR N=0 TO 6
40 REM FONCTION
50 K=0.0418879
60 Y=INT((200+180*10^(-(X+N)/450))*SIN(K*(X+N)))+0.5)
70 IF Y > 255 THEN A=1:B=Y-255 ELSE A=0:B=Y
80 LPRINT CHR$(#1B);CHR$(#10);CHR$(A);CHR$(B);CHR$(128+2^N);

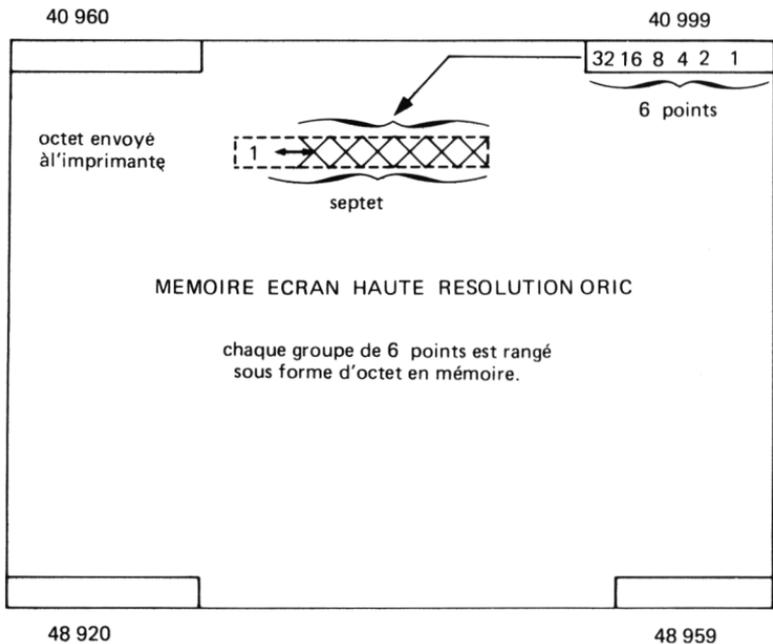
```

```

90 NEXTN:LPRINTCHR$(#0D)
100 NEXTX
200 REN LISTING
210 LPRINTCHR$(#0E)
220 LPRINT CHR$(#10);"16TRACE DE COURBES";
CHR$(#0D)
230 LPRINT CHR$(#10);"13EN HAUTE RESOLUTIO
N"
240 LPRINT CHR$(#08)
250 LPRINT CHR$(#1B);CHR$(#10);CHR$(#00);C
HR$(#08);
260 FOR N =1 TO 162
270 LPRINT CHR$(#AA);
280 NEXT
290 LPRINT CHR$(#0F)
300 END
400 CALL#E6CA
410 LLIST

```

Figure n° 3.



RECOPIE RAPIDE D'ECRAN HAUTE RESOLUTION

Le programme proposé précédemment était long car chaque point de l'écran était testé. Ce serait beaucoup plus rapide si l'on pouvait travailler directement par groupe de points. Or, malchance, l'imprimante génère des septets, alors que la mémoire écran d'ORIC contient les points par groupes de 6. Nous pouvons donc essayer de tricher. Nous allons extraire de cette mémoire un groupe de 6 points, tester le dernier d'entre eux pour savoir s'il est « allumé » ou pas, rajouter alors un septième point artificiel identique au sixième et envoyer le tout sous forme de septet à l'imprimante, ceci afin de masquer la tricherie. (figure 3).

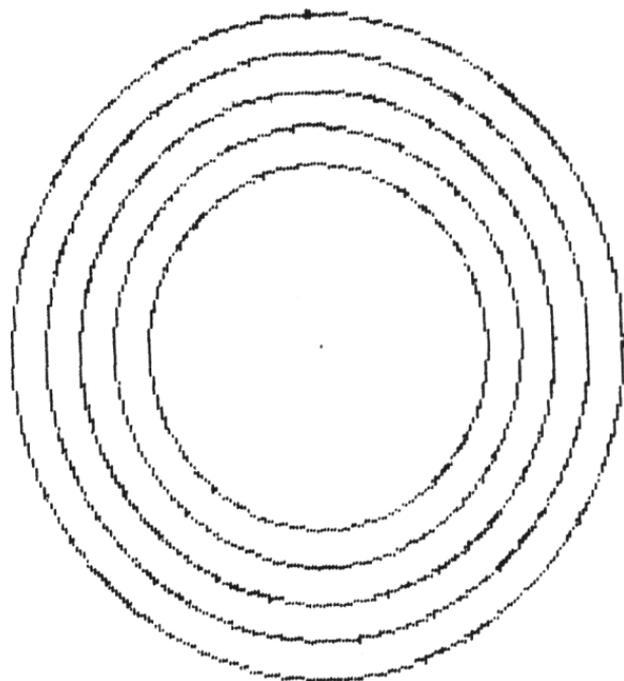
Le programme est beaucoup plus rapide, mais il y a un inconvénient : 6 points ayant été répartis sur 7, cela entraîne une dilatation d'axe, de rapport 7/6. Un carré n'est plus un carré, mais un rectangle. Ce n'est pas gênant dans la plupart des applications. De fait, nous avons bien sur l'écran des cercles qui sont ovales... alors pourquoi pas sur l'imprimante ?

Quelques remarques à propos de ce programme. POKE 49,255 et LPRINT CHR\$(0D) (lignes 105 à 170) pourraient être remplacées par POKE 49,213 puisque nous n'imprimons que 200 septets par ligne. Le signal de retour de chariot serait alors fourni naturellement par ORIC. La ligne 130 teste l'octet d'adresse Y. Le bit b_6 d'un tel octet vaut 0 pour de l'encre et 1 pour le fond. La ligne 135 l'élimine. La ligne 140 teste le bit b_5 : si celui-ci vaut 1, on rajoute un bit b_6 égal à 1, puis le bit b_7 (= 1) que réclame l'imprimante en mode graphique. Si b_5 vaut 0, on laisse b_6 à 0. Nous sommes ainsi passés de l'octet en mémoire au septet destiné à l'imprimante.

Programme n° 3.

```
1 REM*****
2 REM*
3 REM* RECOPIE RAPIDE
4 REM* D'ECRAN
5 REM* HAUTE RESOLUTION
6 REM* POUR ORIC
7 REM*
8 REM*****
10 REM RECOPIE D'ECRAN
20 REMCOPYRIGHT PM BEAUFILS
30 REM EXEMPLE DE DESSIN
```

```
40 HIRES
50 CURSET110,100,1
60 FORN=0 TO 4:CIRCLE 50+10*N,1:NEXT
310 POKE 49,255:LPRINT CHR$(#08)
320 FOR X=40999 TO 40960 STEP -1
330 FOR Y=X TO X+7960 STEP 40
340 A=PEEK(Y)
350 IF A >= 64 THEN A=A-64
360 IF A >= 32 THEN A=(A+192) ELSE A=(A+128)
370 LPRINT CHR$(A);
380 NEXT Y
390 LPRINT CHR$(#0D)
400 NEXT X
410 LPRINT CHR$(#0F):POKE49,80
420 REM FIN COPIE RAPIDE
430 CALL#E60A
440 LLIST
```



IMPRESSION DE PLUS DE 242 SEPTETS PAR LIGNE : MODE GRAPHIQUE.

Nous savons que, d'origine, ORIC peut imprimer 67 caractères par ligne. Cela veut dire qu'il enverra un signal de retour à la ligne à la suite de 67 instructions du type LPRINT CHR\$(X) ; POKE 49,93 permet d'atteindre 80 caractères par ligne ; c'est suffisant dans la mesure où le papier ne peut pas supporter un nombre de caractères supérieur. Cela devient très gênant en mode graphique. En effet, le nombre maximum de septets imprimables est 480 par ligne pour la GPIOO. POKE 49,255 ne permet d'en imprimer que 242, c'est-à-dire qu'on utilise alors seulement la moitié de la largeur du papier (et donc de la définition horizontale). Quel dommage ! Nous allons voir les différentes méthodes permettant d'éliminer cette contrainte.

1°) Répétition de motifs.

Si un motif graphique doit être répété un certain nombre de fois, il est possible alors d'employer le code $\neq \neq 1 C$ qui s'utilise ainsi :

```
LPRINT CHR$( $\neq \neq 10$ ) ; « XX » ; CHR$( $\neq \neq 1 C$ ) ; CHR$(Y) ;  
CHR$(N) ;
```

— XX est la position du début d'impression ($00 \leq XX \leq 79$) (adressage par caractère).

— Y est le nombre désiré de répétition du motif.

($0 \leq Y \leq 255$).

— N est le motif du septet, tel que défini dans l'article précédent.

Une telle ligne de programme est équivalente à :

```
LPRINT CHR$( $\neq \neq 10$ ) ; « XX » ;
```

```
FOR J = 1 to Y
```

```
LPRINT CHR$(N) ;
```

```
NEXT J
```

Mais elle utilise seulement 6 fois l'instruction LPRINT ; au lieu de $Y + 3$, ce qui est un gain appréciable (voir programme 5).

Remarquons que l'utilisation en fin de ligne de LPRINT CHR\$($\neq \neq \emptyset d$), impose un retour à la ligne sans espacement. L'utilisation de : LPRINT CHR\$($\neq \neq \emptyset F$) ; CHR\$($\neq \neq \emptyset D$) puis CHR\$($\neq \neq \emptyset 8$) en début de ligne suivante impose un espacement (programme 4). En effet, lors d'un retour de ligne, un espacement est fourni (si l'on est en mode caractère $\neq \neq \emptyset F$) ou ne l'est pas (en mode graphique $\neq \neq \emptyset 8$). Ceci accroît encore la souplesse d'emploi de la GPIOO.

2°) On peut également utiliser la totalité de la largeur du papier si l'on n'a qu'un petit nombre de septets à imprimer par ligne. Expliquons-nous. Il est possible d'adresser chaque septet par les codes $\neq \neq 1B$, $\neq \neq 10$. Donc, pour un septet de code graphique N donné, d'adresse XY, on peut l'imprimer par :

LPRINT CHR\$($\neq \neq 1B$) ; CHR\$($\neq \neq 10$) ; CHR\$(X) ; CHR\$(Y) ; CHR\$(N) sachant que, si A est l'adresse en question : ($0 \leq A \leq 479$)

* si $A < 256$: X = 0 ; Y = A.

* si $A \geq 256$: X = 1 ; Y = A - 256.

On utilise ainsi 5 fois LPRINT par septet ; on peut donc sans problèmes expédier à l'imprimante 242/5 $\neq \neq 28$ septets par ligne, ORIC envoyant son retour de chariot au 242° LPRINT (Il vaut mieux l'imposer par CHR\$($\neq \neq 0D$), puisqu'il nous reste une marge de 2 LPRINT, soit 242 - 240).

3°) Enfin (le meilleur est toujours gardé pour la fin), n'existe-t-il pas un « truc » permettant de s'affranchir de cette contrainte de 242 LPRINT par ligne ?

Ce truc devrait :

* obliger ORIC à envoyer son retour de chariot (nous serions alors tranquille pour les 242 septets suivants).

* être tel que l'imprimante ne comprenne pas cet ordre et qu'en désespoir de cause, elle poursuive la ligne entamée.

Nous aurions ainsi éliminé ce retour à la ligne intempestif et pourrions alors imprimer 480 motifs différents (et jointifs) par ligne. Ce truc existe : c'est la ligne 80 du programme 6. C'est elle qui, détectant le 220° LPRINT (220 est pris comme exemple), envoie le début du code d'adressage par septet : $\neq \neq 1B$; $\neq \neq 10$ (remarquer l'absence de ; et de, imposant donc à ORIC l'envoi de son signal de fin de ligne).

Mais l'adresse, qui devrait suivre ces deux codes n'est pas fournie. L'imprimante n'y comprend rien : elle doit « penser » qu'il y a erreur de transmission ; elle poursuit alors imperturbablement son impression sur la même ligne. Dans le programme 2, il y a 19 résistances dessinées, chacune occupant 20 septets. Nous avons ainsi 380 septets sur chaque ligne. Nous pouvons ne pas espacer celles-ci (programme 7) par une nouvelle instruction 100.

Vous pouvez ainsi dessiner (par exemple) une carte de France (que de DATA en perspective) avec une résolution de 480 points.

RECOPIE D'ECRAN TEXTE DONT LES CARACTERES ONT ETE REDEFINIS.

Il est impossible d'effectuer une telle opération directement. En effet, les caractères ORIC sont définis sur 8 points, alors que, pour la GPIOO ils sont sur 7. D'autre part, ORIC transmet le code ASCII de chaque caractère à l'imprimante, et non sa définition.

1°) Une méthode directe consisterait à :

- * redéfinir les caractères TEXTE
- * recopier l'écran TEXT dans une matrice A (1120)
- * passer en HIRES
- * redéfinir les caractères HIRES (dont l'emplacement mémoire n'est pas le même)
- * imprimer (CURSET et CHAR) chaque caractère redéfini à sa place sur l'écran HIRES (à ce propos, il semblerait que l'on ne puisse pas utiliser la 40^e colonne)
- * recopier l'écran HIRES

C'est long, mais ça marche...

Il faut dans ce cas éviter les attributs (de code ASCII compris entre 1 et 31) et se rappeler qu'un espace a pour code 32.

2°) Pour un travail soigné (et plus précis : la définition de la GPIOO est de 480 points, à opposer aux 240 de l'ORIC), il est préférable d'attaquer directement l'imprimante avec le dessin. Ici, programme 8, une miniature représentant un amplificateur à transistor.

Signalons pour terminer que les impressions sont superposables : on peut ainsi réaliser un dessin, en plusieurs « passes » par ligne. Il suffit d'envoyer des motifs différents aux mêmes emplacements, par adressage, par exemple (programme 9).

Enfin, on peut tenter d'améliorer la présentation des graphiques (il y a des septets parasites sur le papier sous forme de barres - voir programme 8 la liaison entre les deux résistances —) en débranchant le clavier pendant l'impression par CALL ≠ ≠ E6CA (le reconnecter ensuite par CALL ≠ ≠ E804). On pourra consulter à ce sujet MICRO-RIC n° 2 page 48 ou bien VISA POUR ORIC, chapitre clavier. Mais cela pose quelques problèmes en général...

CONCLUSION

Il semblerait que l'imprimante SEIKOSHA GP 100 (pas de publicité pour la marque) en liaison avec ORIC constitue une solution de choix. La dénomination « graphique » n'est pas un trompe l'œil publicitaire. A certains points de vue, cet ensemble est plus performant que

d'autres modèles nettement plus onéreux. Nous ne saurions trop conseiller au lecteur de se constituer un petit programme spécifique convenant à ses besoins, qu'il chargera dans l'ORIC avant de commencer un nouveau travail.

Programme 4 (avec espacement entre lignes).

```

*****
REM*****
REM*
REM* ESPACEMENT ENTRE LIGNES *
REM*
REM*****
10 LPRINTCHR$(#08);:LPRINTCHR$(#10);:LPRINTCHR$(#10);:LPRINTCHR$(#10);
20 LPRINTCHR$(#FF);:LPRINTCHR$(#213);:LPRINTCHR$(#0F);:LPRINTCHR$(#0D)
30 GOTO10

```

```

1 REM*****
2 REM*
3 REM* ESPACEMENT ENTRE LIGNES *
4 REM*
5 REM*****
10 LPRINTCHR$(#08);:LPRINTCHR$(#10);:LPRINTCHR$(#10);:LPRINTCHR$(#10);
20 LPRINTCHR$(#FF);:LPRINTCHR$(#213);:LPRINTCHR$(#0F);:LPRINTCHR$(#0D)
30 GOTO10

```

Programme 5 (sans espacement entre lignes).

```

*****
REM*****
REM*
REM* ESPACEMENT ENTRE LIGNES *
REM* SUPPRIME
REM*
REM*****
10 LPRINTCHR$(#08);:LPRINTCHR$(#10);:LPRINTCHR$(#10);:LPRINTCHR$(#10);

```

```

1 REM*****
2 REM*
3 REM* ESPACEMENT ENTRE LIGNES *
4 REM* SUPPRIME
5 REM*
6 REM*****
10 LPRINTCHR$(#08);:LPRINTCHR$(#10);:LPRINTCHR$(#10);:LPRINTCHR$(#10);

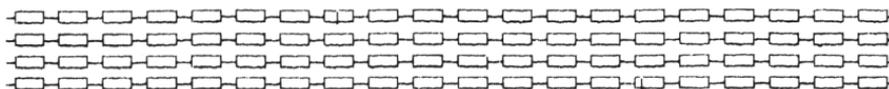
```

```

T"10") :LPRINTCHR$(#10);
20 LPRINTCHR$(#FF); :LPRINTCHR$(#213); :LPRI
NTCHR$(#0D)
30 GOTO10

```

Programme 6 (avec espacement).

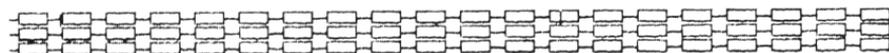


```

1 REM*****
2 REM*                                     *
3 REM*  REALISATION D' UN                 *
4 REM*  MOTIF GRAPHIQUE                  *
5 REM*                                     *
6 REM*****
10 LPRINTCHR$(#08); :POKE49,255
20 FORU=0 TO 19
30 FOR N=1 TO 20
40 READX
50 M=M+1
60 LPRINT CHR$(X);
70 NEXTN
80 IF M = 220 THEN LPRINT CHR$(#1B);CHR$(#
10)
90 RESTORE :NEXTU
100 LPRINT CHR$(#0F);CHR$(#0D);LPRINT CHR$
(#08); :M=0
110 GOTO 20
120 DATA 136,136,136,136,191,161,161,161,1
61,161,161,161,161,161,161,161,161
130 DATA 191,136,136
140 END

```

Programme 7 (sans espacement).

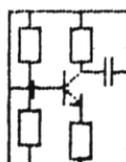


```

100 LPRNT CHR$(#0D) :M=0

```

Programme 8.



```
1 REM*****
2 REM*                                     *
3 REM* REALISATION DIRECTE *
4 REM*   D'UN DESSIN *
5 REM* SUR L'IMPRIMANTE *
6 REM*                                     *
7 REM*****
10 LPRINTCHR$(#08)
20 FORN=1TO7
30 LPRINTCHR$(#10);:LPRINT"10";:LPRINTCHR$(
(255);
40 FORK=1TO34
50 READA:LPRINT CHR$(A+128);
60 NEXTK:LPRINTCHR$(255);CHR$(#0D);NEXTN
70 LPRINTCHR$(#10);:LPRINT"10";:LPRINTCHR$(
(#1C):LPRINTCHR$(#22);
80 LPRINTCHR$(129);:LPRINTCHR$(#0D)
90 END
100 DATA 1,1,97,33,33,63,33,33,97,1,1,1,1,
1,1,1,1,97,33,33,63,33,33,97
110 DATA 1,1,1,1,1,1,1,1,1,1
120 DATA 0,0,127,0,0,0,0,0,127,0,0,0,0,0,
0,0,127,0,0,0,0,0,127
130 DATA 0,0,0,0,0,0,0,0,0,0
140 DATA 0,0,7,4,4,124,4,4,7,0,0,0,0,0,0,9
6,0,7,4,68,60,36,36,39,32,32,32
150 DATA 126,0,126,32,32,32,32
160 DATA 8,8,8,8,8,127,8,8,8,8,8,8,8,8,8,1
27,20,34,65,0,96,
170 DATA 0,0,0,0,0,0,7,0,7,0,0,0,0
180 DATA 0,0,120,8,8,15,8,8,120,0,0,0,0,0,
0,3,0,2,2,3,127
190 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0
```

```

200 DATA 0,0,127,0,0,0,0,0,127,0,0,0,0,0,0,0
,0,0
210 DATA 127,1,1,1,1,1,1,127,0,0,0,0,0,0,0,0
,0,0
220 DATA 64,64,65,65,65,127,65,65,65,0,0,0
,0,0,0,0,0
230 DATA 95,80,80,112,80,80,95,64,64,64,64
,64,64,64,64,64,64
240 LPRINT CHR$(#0F)
250 END

```

Programme 9.

```

1 REM*****
2 REM*                                     *
3 REM*   SUPERPOSITON                     *
4 REM*   DE                                 *
5 REM*   MOTIFS GRAPHIQUES                *
6 REM*                                     *
7 REM*****
10 LPRINTCHR$(#08)
20 LPRINTCHR$(#1B);CHR$(#10);CHR$(#00);CHR
$(#64);
30 FORN=1TO7
40 READA:LPRINTCHR$(A+128);
50 NEXTN
60 LPRINT CHR$(#0D)
70 LPRINTCHR$(#1B);CHR$(#10);CHR$(#00);CHR
$(#64);
80 FORN=1TO7
90 READA:LPRINTCHR$(A+128);
100 NEXTN
110 LPRINT CHR$(#0D)
120 DATA 65,34,20,8,20,34,65
130 DATA 127,65,65,65,65,65,127
140 LPRINT CHR$(#0F)
150 END

```



Annexe 3

CHOIX DE VALEURS NORMALISÉES

Recherche de valeurs normalisées

Lorsque l'on a à trouver la valeur de composants électroniques obéissant à une certaine valeur, cela est très facile. Dans le cas d'une formule du type

$$f = \frac{1}{RC}$$

(cas du décodeur 567), connaissant f , il est aisé, en se fixant R , d'en déduire une valeur de C . En général (loi dite de Murphy), cette valeur de C n'est pas normalisée. On se fixe alors une nouvelle valeur de R , d'où l'on déduit une nouvelle valeur de C , qui, elle aussi, n'est pas normalisée... Le problème est donc, ayant choisi une valeur normalisée pour R , en trouver une pour C , ou éventuellement la totalité des couples R, C normalisés obéissant à la relation.

La capacité de calcul d'un micro-ordinateur est ici la bienvenue. Il est possible d'examiner par programme toutes les valeurs normalisées de R , puis, pour chacune d'entre elles, s'il existe, avec une certaine précision, une valeur normalisée correspondante pour C .

Nous supposerons que R peut prendre toutes les valeurs de la série E 12, comprise entre 100Ω et $1 M\Omega$. C sera également prise dans la série E 12, et pourra prendre des valeurs comprises entre 1 nF et $1\ 000 \mu\text{F}$. Ces contraintes sont facilement modifiables.

Les programmes. Ils ont tous la même architecture.

* Programme n° 1 :

— lignes 30 et 40 : initialisation des valeurs normalisées.

— lignes 70 à 100 : calcul de toutes les valeurs possibles de résistances.

— lignes 110 à 120 : pour chaque valeur de résistance, on cherche la valeur de C correspondant à la relation

$$f = \frac{1}{RC}$$

Les valeurs trop petites ($< 1 \text{ nF}$) ou trop grandes ($> 1\ 000 \text{ nF}$) sont éliminées.

— lignes 130 à 150 : pour une valeur de C convenable, on cherche la puissance de 10 par laquelle il faut multiplier C pour trouver un nombre supérieur à 1 (donc compris entre 1 et 10).

— lignes 160 à 180 : ce nouveau nombre est comparé à $\pm 5 \%$ aux différents nombres composant la série dans laquelle est pris le condensateur C.

— lignes 190-195 : s'il existe une valeur convenable, celle-ci est affichée.

— ligne 210 : data contenant les 13 nombres de la série E 12, dans laquelle sont supposés pris condensateurs et résistances.

* Le programme n° 2 distingue la série E 12 choisie pour les résistances (data en ligne 210) de la série E 6 choisie pour les capacités (data en ligne 220).

* Le programme n° 3 permet de calculer les éléments d'un multivibrateur astable réalisé à l'aide d'un Timer type 555. La formule utilisée est alors la suivante :

$$f = \frac{1,44}{(R_A + 2 R_B) C}$$

Il est demandé à l'utilisateur la valeur de f choisie et une valeur possible pour R_A . Le programme en déduit R_B et C.

Les variables :

* V (13) : tableau contenant la série (E 12) des résistances.

* W (13) : tableau contenant la série (E 12 ou E 6) des condensateurs.

* R (52) : tableau contenant toutes les valeurs possibles normalisées de résistances (de 100Ω à $1 M\Omega$).

* N : valeur de la puissance de 10 que contient le nombre représentant la capacité du condensateur.

* K : variable de défilement de toutes les valeurs de résistances.

* A : variable de défilement des valeurs normalisées de condensateur.

Programme n° 1

(R et C choisis dans la série E 12)

```
1 REM*****
2 REM*                                     *
3 REM*      CALCUL                         *
4 REM*                                     *
5 REM* DE VALEURS NORMALISEES *
6 REM*                                     *
7 REM*****
10 PRINT"Ce programme calcule les valeurs normalisees des composants")
```

```

20 PRINT"R et C correspondants a la formule : f
= 1/RC"
30 DIMV(13):DIMR(52):DIMW(13)
40 FOR I=1 TO 13:READU:V(I)=U:W(I)=U:NEXT
50 INPUT"F =":F
55 LPRINT"      ";CHR$(#0E); "F=";F;"Herz";CHR$(
#0F)
60 K=1:P=1
70 FOR N=2TO5:FORI=1 TO 13
80 R(K)=V(I)*10^N
90 K=K+1
100 NEXTI:NEXTN
110 FOR K=1 TO 52
120 C=1/(F*R(K)):D=C:IF C < 1E-9 OR C > 1E-3 TH
EN200
130 N=0
140 REPEAT:D=D*10:N=N+1
150 UNTIL D >= 1
160 FORA=1 TO 13
170 IF D < 1.05*W(A) AND D > 0.95*W(A) THEN190
180 NEXTA:GOTO200
190 PRINT"R =" ;R(K); "Ohm", "C=";W(A); "#10^";6-N;
"Microf"
195 LPRINT"R =" ;R(K); "Ohm", "C=";W(A); "#10^";6-N
;"Microf"
200 NEXTK
210 DATA 1,1.2,1.5,1.8,2.2,2.7,3.3,3.9,4.7,5.6,
6.8,8.2,10
220 END
1000 CALL#E6CA
1010 LLIST

```

Exemple :

```

      F=11500 Herz
R =150 Ohm C=5.6 #10^-1 Microf
R =180 Ohm C=4.7 #10^-1 Microf
R =220 Ohm C=3.9 #10^-1 Microf
R =270 Ohm C=3.3 #10^-1 Microf
R =330 Ohm C=2.7 #10^-1 Microf
R =390 Ohm C=2.2 #10^-1 Microf
R =470 Ohm C=1.8 #10^-1 Microf
R =560 Ohm C=1.5 #10^-1 Microf
R =1500 Ohm      C=5.6 #10^-2 Microf
R =1000 Ohm     C=4.7 #10^-2 Microf

```

R =2200 Ohm	C=3.9 *10 ⁻² Microf
R =2700 Ohm	C=3.3 *10 ⁻² Microf
R =3300 Ohm	C=2.7 *10 ⁻² Microf
R =3900 Ohm	C=2.2 *10 ⁻² Microf
R =4700 Ohm	C=1.8 *10 ⁻² Microf
R =5600 Ohm	C=1.5 *10 ⁻² Microf
R =15000 Ohm	C=5.6 *10 ⁻³ Microf
R =18000 Ohm	C=4.7 *10 ⁻³ Microf
R =22000 Ohm	C=3.9 *10 ⁻³ Microf
R =27000 Ohm	C=3.3 *10 ⁻³ Microf
R =33000 Ohm	C=2.7 *10 ⁻³ Microf
R =39000 Ohm	C=2.2 *10 ⁻³ Microf
R =47000 Ohm	C=1.8 *10 ⁻³ Microf
R =56000 Ohm	C=1.5 *10 ⁻³ Microf

F=200000 Herz

R =150 Ohm	C=3.3 *10 ⁻² Microf
R =180 Ohm	C=2.7 *10 ⁻² Microf
R =220 Ohm	C=2.2 *10 ⁻² Microf
R =270 Ohm	C=1.8 *10 ⁻² Microf
R =330 Ohm	C=1.5 *10 ⁻² Microf
R =620 Ohm	C=8.2 *10 ⁻³ Microf
R =820 Ohm	C=6.2 *10 ⁻³ Microf
R =1500 Ohm	C=3.3 *10 ⁻³ Microf
R =1800 Ohm	C=2.7 *10 ⁻³ Microf
R =2200 Ohm	C=2.2 *10 ⁻³ Microf
R =2700 Ohm	C=1.8 *10 ⁻³ Microf
R =3300 Ohm	C=1.5 *10 ⁻³ Microf

F=540000 Herz

R =100 Ohm	C=1.8 *10 ⁻² Microf
R =120 Ohm	C=1.5 *10 ⁻² Microf
R =150 Ohm	C=1.2 *10 ⁻² Microf
R =180 Ohm	C=1 *10 ⁻² Microf
R =220 Ohm	C=8.2 *10 ⁻³ Microf
R =270 Ohm	C=6.8 *10 ⁻³ Microf
R =330 Ohm	C=5.6 *10 ⁻³ Microf
R =390 Ohm	C=4.7 *10 ⁻³ Microf
R =470 Ohm	C=3.9 *10 ⁻³ Microf
R =560 Ohm	C=3.3 *10 ⁻³ Microf
R =680 Ohm	C=2.7 *10 ⁻³ Microf
R =820 Ohm	C=2.2 *10 ⁻³ Microf

R =1000 Ohm	C=1.8 *10 ⁻³ Microf
R =1200 Ohm	C=1.5 *10 ⁻³ Microf
R =1500 Ohm	C=1.2 *10 ⁻³ Microf
R =1800 Ohm	C=1 *10 ⁻³ Microf

Programme n° 2

(R dans la série E 12, C dans la série E 6)

```

1 REM*****
2 REM# #
3 REM# CALCUL #
4 REM# #
5 REM# DE VALEURS NORMALISEES #
6 REM# #
7 REM*****
10 PRINT"Ce Programme calcule les valeurs norma
lisees des composants";
20 PRINT"R et C correspondants a la formule : f
= 1/RC"
30 DIMV(13):DIMR(52):DIMW(13)
40 FOR I=1 TO 13:READU:V(I)=U:NEXT
45 FOR I=1 TO 7:READU:W(I)=U:NEXT
50 INPUT"F =";F
55 LPRINT" ";CHR$(#0E); "F=";F;"Herz";CHR$(
#0F)
60 K=1:P=1
70 FOR N=2TO5:FORI=1 TO 13
80 R(K)=V(I)*10^N
90 K=K+1
100 NEXTI:NEXTN
110 FOR K=1 TO 52
120 C=1/(F*R(K)):D=C:IF C < 1E-9 OR C > 1E-3 THE
N200
130 N=0
140 REPEAT:D=D*10:N=N+1
150 UNTIL D >= 1
160 FORA=1 TO 7
170 IF D < 1.05*W(A) AND D > 0.95*W(A) THEN190
180 NEXTA:GOTO200
190 PRINT"R =";R(K); "Ohm", "C=";W(A); "#10^";6-N;
"Microf"
195 LPRINT"R =";R(K); "Ohm", "C=";W(A); "#10^";6-N
;"Microf"
200 NEXTK
210 DATA 1,1.2,1.5,1.8,2.2,2.7,3.3,3.9,4.7,5.6,
6.8,8.2,10

```

```

215 DATA 1,1.5,2.2,3.3,4.7,6.8,10
220 END
1000 CALL#E6CA
1010 LLIST

```

Exemple :

```

          F=11500 Herz
R =180 Ohm C=4.7 *10^-1 Microf
R =270 Ohm C=3.3 *10^-1 Microf
R =390 Ohm C=2.2 *10^-1 Microf
R =560 Ohm C=1.5 *10^-1 Microf
R =1800 Ohm          C=4.7 *10^-2 Microf
R =2700 Ohm          C=3.3 *10^-2 Microf
R =3900 Ohm          C=2.2 *10^-2 Microf
R =5600 Ohm          C=1.5 *10^-2 Microf
R =18000 Ohm         C=4.7 *10^-3 Microf
R =27000 Ohm         C=3.3 *10^-3 Microf
R =39000 Ohm         C=2.2 *10^-3 Microf
R =56000 Ohm         C=1.5 *10^-3 Microf

```

Programme n° 3

```

1 REM*****
2 REM#                                     *
3 REM#          CALCUL                    *
4 REM#                                     *
5 REM# DE VALEURS NORMALISEES *
6 REM#                                     *
7 REM*****
10 PRINT"Ce programme calcule les valeurs norma
lisees des composants";
20 PRINT"RA et C correspondants a la formule :f
=1.44/(RA+2RB)C.
30 DIMV(13):DIMR(52):DIMW(13)
40 FOR I=1 TO 13:READU:V(I)=U:NEXT
45 FOR I=1 TO 7:READU:W(I)=U:NEXT
50 INPUT"F = ";F
52 INPUT "RB = ?";RB
53 LPRINT "          ";CHR$(#0E); "F=";F; "Herz";CHR$(
#0F):LPRINT:LPRINT
55 LPRINT "          RB = ";RB:LPRINT

```

```

60 K=1:P=1
70 FOR N=2TO5:FORI=1 TO 13
80 R(K)=V(I)*10^N
90 K=K+1
100 NEXTI:NEXTN
110 FOR K=1 TO 52
120 C=1.44/((R(K)+2*RB)*F):D=C:IF C < 1E-9 OR C
> 1E-3 THEN200
130 N=0
140 REPEAT:D=D*10:N=N+1
150 UNTIL D >= 1
160 FORA=1 TO 7
170 IF D < 1.05*W(A) AND D > 0.95*W(A) THEN190
180 NEXTA:GOTO200
190 PRINT"R =";R(K);"Ohm","C=";W(A);"*10^";6-N;
"Microf"
195 LPRINT"R =";R(K);"Ohm","C=";W(A);"*10^";6-N;
;"Microf"
200 NEXTK
210 DATA 1,1.2,1.5,1.8,2.2,2.7,3.3,3.9,4.7,5.6,
6.8,8.2,10
215 DATA 1,1.5,2.2,3.3,4.7,6.8,10
220 END
1000 CALL#E6CA
1010 LLIST

```

Exemple :

F=11500 Herz

RB = 10000

R =5600 Ohm	C=4.7 *10 ⁻³ Microf
R =6800 Ohm	C=4.7 *10 ⁻³ Microf
R =18000 Ohm	C=3.3 *10 ⁻³ Microf
R =39000 Ohm	C=2.2 *10 ⁻³ Microf
R =100000 Ohm	C=1 *10 ⁻³ Microf
R =100000 Ohm	C=1 *10 ⁻³ Microf

Annexe 4

APPENDICES UTILITAIRES

Comment accélérer l'exécution d'un programme

Indépendamment des particularités de chaque micro-ordinateur, il existe des règles universelles permettant de rendre plus rapide l'exécution d'un programme écrit en Basic. Citons en quelques-unes.

* Il est judicieux de placer les sous-programmes (SUB) parmi les premières lignes d'un programme. En effet, lors d'un GOSUB, le moniteur examine toutes les lignes depuis la première.

* Il n'est pas indispensable, lors d'un NEXT, de faire suivre cette instruction par la variable de boucle considérée. Mais il faut alors qu'il n'y ait pas ambiguïté sur celle-ci. Dans le cas où cette variable est omise, l'ordinateur saute à la ligne FOR sans chercher à vérifier sa valeur.

* Enfin, et ceci est particulièrement vrai pour les boucles, il est toujours préférable d'utiliser des variables plutôt que des nombres. Une

expression du type $\frac{2\pi t}{50}$ (soit $2 * \pi * t / 50$) sera remplacé par $A * t$

on définira alors en début de programme la variable A par $LET A = 2 * \pi / 50$.

Nous n'avons jamais employé ces méthodes dans les programmes proposés dans le texte. Cela se serait fait au détriment de la clarté et de la compréhension des listings. Nous invitons cependant le lecteur à employer ces méthodes chaque fois que le besoin s'en fera sentir (calculs avec des séries de Fourier par exemple...).

* Enfin, dernière remarque, il faut éviter le plus possible l'emploi de tests (IF... THEN) dans les boucles (FOR... NEXT).

PHOTOGRAPHIES D'ECRAN

Les valeurs : 1/8^e de seconde, F/8 conviennent pour un film 400 ASA et une luminosité moyenne de l'écran. Eviter alors l'emploi de couleurs.

Comment économiser la mémoire

* Dans les cas vraiment critiques (programmes très longs), on a intérêt à remplacer les constantes souvent employées (2 par exemple) par des variables ($LET DE = 2$). Ainsi DE n'utilise en général qu'un octet (nom de variable), plus 6 pour sa valeur numérique.

Inversement écrire 2 en deux endroits différents consomme 12 octets. Remarquons que ce procédé augmente la vitesse d'exécution des programmes.

* Il faut éviter toutes les choses inutiles au fonctionnement du programme : les espaces dans les instructions, les REM, les STOP, les indices de variables de boucle (après NEXT),... et si possible mettre plusieurs instructions par ligne.

ORIC 1 ET ATMOS

L'ATMOS est le successeur de l'ORIC 1. Il semble promis au même succès. N'utilisant que rarement la possibilité de POKER des cases mémoire utilisées par la ROM, les programmes proposés sont directement compatibles pour l'ATMOS. Celui-ci n'interfère pas avec l'imprimante par sa scrutature clavier, contrairement à l'ORIC 1. Il n'est donc plus nécessaire d'utiliser CALL #E6 CA (qui est d'ailleurs devenu CALL #E76A) pour éliminer ce phénomène. La mémoire 49 de l'ORIC 1, qui contient le nombre de caractères par ligne de l'imprimante, est devenue la mémoire # 256 et contient 80 à la mise sous tension. On trouvera la liste des anciennes et nouvelles adresses des variables système dans « VISA POUR ORIC », Editions SORACOM.

BIBLIOGRAPHIE

- Filtres actifs (Paul Bildstein). Edition Radio 1972.
- Analyse et calcul des amplificateurs haute fréquence (A. Bensasson). Edition Radio 1967.
- Electronique Tome 1 (R. Guillien). Puf 1963.
- Asservissements linéaires (F. Milsant). Eyralles 1971.
- Cours élémentaires de mathématiques supérieures (J. Quinet). Dunod 1964.
- Théorie et Pratique des circuits de l'Electronique et des amplificateurs (Tome 3 : Abaque de Smith). (J. Quinet, Dunod 1963).
- La revue MEGAHERTZ (Editions Soracom).
- Visa pour Oric (Editions SORACOM).

SOMMAIRE

Introduction	7
CHAPITRE 1 : RÉSONANCE	9
Circuit résonnant LC parallèle	11
Circuits décalés	19
Circuits couplés.	24
CHAPITRE 2 : FILTRAGE	29
Filtres passifs	31
Construction d'un filtre du deuxième ordre	37
CHAPITRE 3 : FOURIER	47
Séries de Fourier	49
Utilisation de la série de Fourier	56
Calcul des coefficients d'une série de Fourier	59
Application de la série de Fourier	64
CHAPITRE 4 : RÉGIMES TRANSITOIRES	69
Régimes transitoires du premier ordre	71
Régimes transitoires du second ordre.	88
Régimes transitoires dans les boucles à verrouillage de phase	100
CHAPITRE 5 : CIRCUITS NON LINÉAIRES	113
Diode en commutation	115
Redressement	121
Détection.	127
Détection synchrone	130
CHAPITRE 6 : ABAQUE DE SMITH	135
Abaque de Smith.	137
ANNEXE 1 : PLL - Principe	145
Constitution d'une boucle à verrouillage de phase	147
ANNEXE 2 : Utilisation de l'imprimante	157
L'instruction LPRINT d'ORIC	159
ANNEXE 3 : Choix de valeurs normalisées	177
Recherche de valeurs normalisées	179
ANNEXE 4 : Appendices utilitaires.	187
Comment accélérer l'exécution d'un programme	189
BIBLIOGRAPHIE	191

CHAQUE MOIS N'OUBLIEZ PAS*

MEGAHERTZ



*Pour ne pas l'oublier, ABONNEZ VOUS !

MEGAHERTZ : SORACOM - 16, av. Gros Malhon
35000 RENNES - Tél. : (99) 54.22.30

DEMANDEZ NOUS UN EXEMPLAIRE DE PRESSE GRATUIT.

Photocomposition : Presse de Bretagne
Maquette : Soracom
Impression : Jouve - Mayenne
N° d'Editeur : 27
N° d'Imprimeur : 12710
Dépôt légal : Avril 1984

L'étude des phénomènes électroniques est souvent rendue difficile par le côté abstrait des formules présentées dans les manuels scolaires.

Pour la première fois, les circuits complexes deviennent vivants, les courbes se tracent automatiquement sous vos yeux en fonction des paramètres que vous introduisez dans votre micro ordinateur ORIC et ATMOS.

Des circuits élémentaires jusqu'à l'étude de l'Abaque de SMITH, Pierre BEAUFILS présente une approche entièrement nouvelle des circuits électroniques.

LE PRINCE DE MONTECRISTO