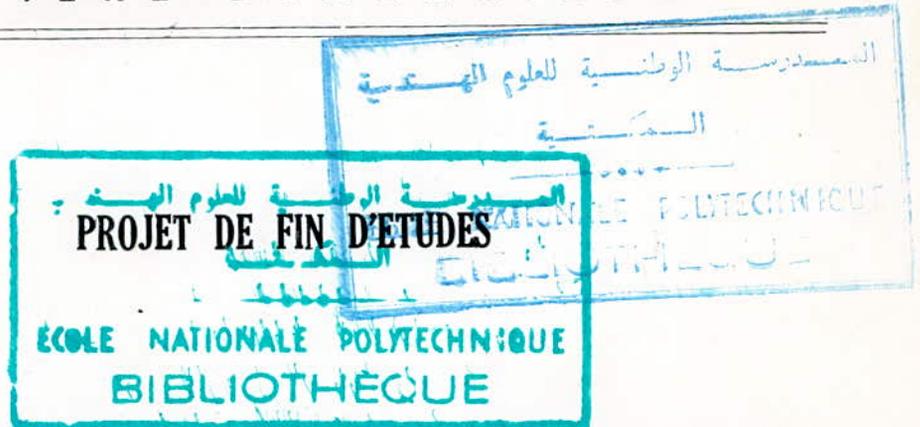


6/82
2er

ÉCOLE NATIONALE POLYTECHNIQUE D'ALGER
DÉPARTEMENT D'ÉLECTRONIQUE
FILIERE D'INGÉNIEUR



– Étude de l'Apple pascal –
Application au tracé automatique
du lieu de Nyquist de la fonction

$$F(p) = \frac{K}{(Bo + B1 p) (Ao + A1 p + A2 p^2)}$$

Proposé par :

H. FARAH

Réalisé par :

Noureddine AMARDJIA

Mustapha MAKLOUFI

ECOLE NATIONALE POLYTECHNIQUE D'ALGER

DEPARTEMENT D'ELECTRONIQUE

FILIERE D'INGENIEUR

المدرسة الوطنية للعلوم الهندسية
المكتبة

PROJET DE FIN D'ETUDES

ECOLE NATIONALE POLYTECHNIQUE
BIBLIOTHEQUE

– Etude de l'Apple pascal –
Application au tracé automatique
du lieu de Nyquist de la fonction

$$F(p) = \frac{K}{(Bo + B1 p) (Ao + A1 p + A2 p^2)}$$

Proposé par :

H. FARAH

Réalisé par :

Noureddine AMARDJIA

Mustapha MAKLOUFI

JUIN 1982

REMERCIEMENTS

Nous tenons à remercier :

- Monsieur H. FARAH, Enseignant à l'E.N.P.A. pour nous avoir proposé ce sujet et aidé dans notre travail.
- Monsieur B. BOUDRAA, enseignant à l'E.N.P.A. pour tous les conseils qu'il nous a prodigués.

Nos remerciements vont également à tous les enseignants qui ont contribué à notre formation, ainsi que tous ceux qui ont participé à l'élaboration de ce mémoire.

DEDICACES

- A la mémoire de ma grand-mère,
- A mes parents, pour leur soutien constant tout au long de mes études,
- A toute ma famille, ainsi qu'à tous mes amis.

Mustapha MAHLOUFI

- A la mémoire de mon père,
- A ma mère,
- A ma famille,
- A tous mes amis.

Nourreddine AMARDJIA

S O M M A I R E

PAGE

Introduction

P1

1ere partie

Rappels sur l'étude des systèmes linéaires

P2

2° partie

Présentation de l'APPLE II PLUS

12

3° Partie

Langage Utilisé (Pascal)

19

4° Partie

Programme Nyquist

26

5° Partie

Exemples et conclusions

35

I) INTRODUCTION GENERALE :

(1)

La stabilité d'un système est, avec la précision, l'un des problèmes fondamentaux posés par les asservissements. Les critères algébriques de stabilité des systèmes linéaires donnent lieu à des calculs compliqués et d'interprétation difficile dès que l'ordre du système est supérieur à deux aussi emploie-t-on le plus souvent les critères graphiques. Cependant, aboutir à un tracé précis et déterminer la marge de stabilité nécessitent un temps assez long. L'économie de ce temps est le but recherché par notre présent travail qui consiste à écrire un programme (en langage pascal) qui permet de tracer le lieu de Nyquist et déterminer éventuellement la marge de stabilité du type de fonctions de transfert de la forme

$$F(p) = K / (b_0 + b_1 p) (a_0 + a_1 p + a_2 p^2) \quad p = j\omega$$

Rappels sur l'étude des systèmes linéaires

A) Définition d'un système physique linéaire.....	3
B) Notion de fonction de transfert	3
C) Stabilité d'un système	4
C 1) Définition 1.....	4
C 2) Définition 2	4
C 3) Critères de stabilité	4
C 3 1) Critère algébrique	4
a) Condition de stabilité	4
b) Condition de limite de stabilité	4
c) Etude de la stabilité.....	4
C 3 2) Critère Graphique	4
a) Diagramme de Nyquist	4
b) Critère de Nyquist	5
c) Critère du revers dans le plan de Nyquist	5
D) Marge de gain et de phase	5
a) Marge de gain	5
b) Marge de phase	5
c) Exemples de calculs	5
d) Remarques.....	6
E) Application aux systèmes à F.T en BO	6
$F(p) = K/(b_0 + b_1 p) (a_0 + a_1 p + a_2 p^2)$	6
a) F.T des systèmes linéaires fondamentaux.....	6
b) (Choix de $F(p)$).....	6
c) Etude de $F(p)$	7
C 1) Système de 1 ^{er} degré ;.....	7
C 2) Système du 2 ^{ème} degré	7
C 3) Système du 3 ^{ème} degré.....	8

1ere PARTIE.Rappels sur l'étude des systèmes lineaires.A) Definition d'un système physique lineaire.

Un système physique est lineaire si la relation entre ^{l'entrée} la (ou les) grandeur(s) de sortie est un système d'équation differentielles lineaires à coefficients constants.

Si le système est à entrée et sortie uniques l'équation differentielle lineaire le regissant s'écrit.

$$b_n \frac{d^n s(t)}{dt^n} + \dots + b_1 \frac{ds(t)}{dt} + b_0 s(t) = \quad (1)$$

$$a_m \frac{d^m e(t)}{dt^m} + \dots + a_1 \frac{de(t)}{dt} + a_0 e(t).$$

avec m inferieur ou egal à n.

B) Nation de fonction de transfert :

En utilisant la transformation de laplace à l'équation precedente et en considerant les conditions initiales nulles on aura :

$$E(p) = L(e(t)) ; S(p) = L(s(t))$$

$$p^m E(p) = L\left(\frac{d^m e(t)}{dt^m}\right) ; p^n S(p) = L\left(\frac{d^n s(t)}{dt^n}\right)$$

l'équation (1) s'écrit

$$S(p) = \frac{a_m p^m + \dots + a_1 p + a_0}{b_n p^n + \dots + b_1 p + b_0} \cdot E(p)$$

La fonction de transfert du système est definie par

$$F(p) = \frac{K(1+T_1 p)(1+T_2 p) \dots (1+T_m p)}{p^q (1+T'_1 p) \dots (1+T'_n p)}$$

Ou K represente le gain statique du système et T1, T'1, T2, T'2, les constantes de temps.

F(p) peut s'écrire :

$$F(p) = k \frac{(p-z_1)(p-z_2) \dots}{p^q (p-p_1)(p-p_2) \dots} = Re^{j\theta}$$

k : constante sans signification physique.

z1, z2, : zeros de F(p)

p1, p2, : poles de F(p)

R module de F(p) et θ son argument.

les poles et les zeros de F(p) peuvent étre reels ou complexes conjugués.

ν caractérise le type du système

pour $\nu=0$ (pas de pôles à l'origine): le système est dit de type zero.

pour $\nu=n$ (n pôles à l'origine) : le système est dit de type n.

C) Stabilité d'un système asservi linéaire.

C1) Def1: Un système linéaire est stable lorsque écarté de sa position d'équilibre, tend à y revenir. Il est instable lorsqu'il tend à s'en écarter.

C2) Def2: un système est stable si sa réponse est nulle lorsqu'on ne lui applique aucune entrée.

C3) Critères de stabilité.

les critères de stabilité étant nombreux, nous nous limitons dans cette étude à enoncer seulement deux critères l'un algébrique et l'autre graphique :

C31) critère algébrique :

soit le système à retour unitaire (tout système peut se ramener à un système à retour unitaire) : voir fig 1.

, $F(p)$: fonction de transfert en boucle ouverte

$T(p)$: " " " " " fermée

$$T(p) = \frac{F(p)}{1+F(p)}$$

a) condition de stabilité

pour qu'un système linéaire soit stable en chaîne fermée, il faut et il suffit que tous les pôles de sa fonction de transfert $T(p)$ aient leurs parties réelles négatives.

b) condition de limite de stabilité :

Un système est dit à la limite de stabilité si sa fonction de transfert $T(p)$ possède un ou plusieurs pôles purement imaginaires.

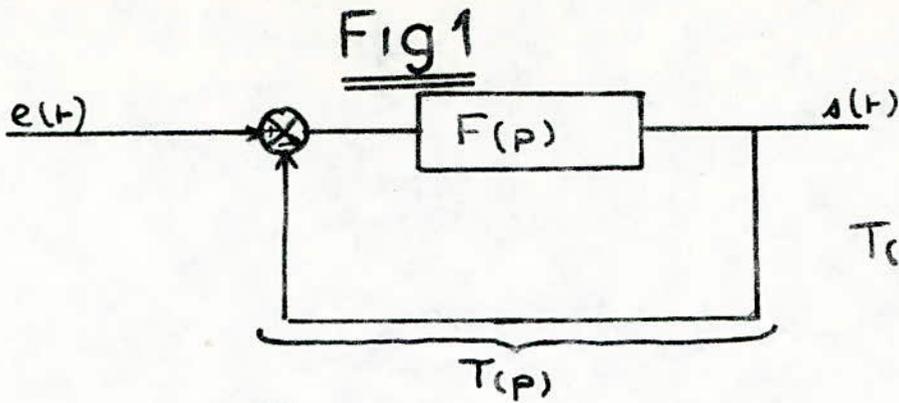
c) étude de la stabilité : des deux conditions précitées, étudier la stabilité d'un système revient à la résolution de l'équation caractéristique : $1+F(p)=0$

La limite de stabilité se traduit donc par l'existence d'une pulsation ω_1 telle que :

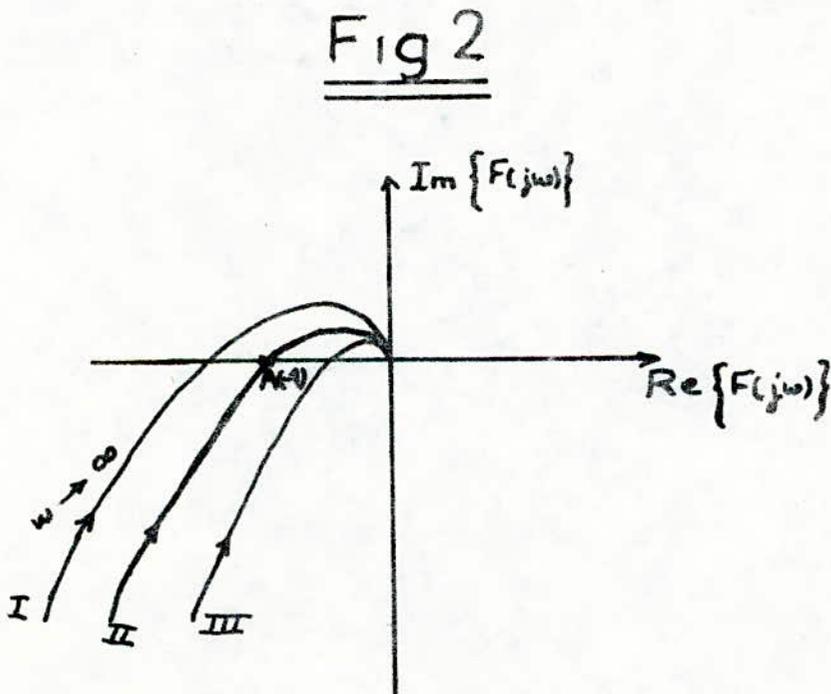
$$\begin{array}{l} |F(j\omega_1)| = 1 \\ \text{Arg}(F(j\omega_1)) = \pm \pi (2k+1) \end{array}$$

C32) critère graphique :

a) Diagramme de Nyquist :



- $F(p)$: Fonction de transfert en B.O.
- $T(p)$: Fonction de transfert en B.F.



- I : système instable.
- III : système stable.
- II : Limite de stabilité.

Ce diagramme consiste à représenter en coordonnées polaires à partir d'un axe réel Ox , le module et la phase d'une fonction de transfert.

b) critère de Nyquist : un système asservi est instable si la courbe de Nyquist en boucle ouverte $F(j\omega)$ enveloppe le point critique $A(-1)$ il est stable si la courbe ne l'enveloppe pas .

c) critère du revers dans le plan de Nyquist :

Un système asservi est stable si en décrivant la courbe de Nyquist dans le sens des fréquences croissantes, on laisse le point critique à sa gauche , il est instable dans le cas contraire (voir Fig 1)

Remarques :

- Ce critère n'est utilisable que pour les systèmes à déphasage minimum qui sont les seuls utilisés pratiquement dans l'industrie.

- le point critique $A(-1)$ est traduit par ∞

$$\begin{array}{l} | \\ | \quad / \overline{\theta A} = 1 \\ | \\ | \quad (Ox, OA) = \pm \pi(2k+1) \\ | \\ | \end{array}$$

D) Marges de gain et de phase :

Il ne suffit pas qu'un système soit stable . Il doit encore posséder un certain degré de stabilité .

Dans le plan de Laplace , cela nécessite une certaine zone de sécurité entre les racines de $(1+F(p))=0$ et l'axe imaginaire . A des racines $r_1, r_2, r_0 = j\omega_0$ de $(1+F(p))=0$ correspondent des diagrammes de Nyquist de gain statique K_1, K_2, K_0 (voir Fig 3)

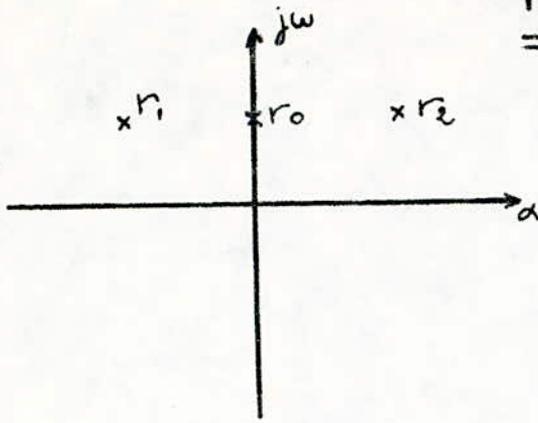
La courbe de Nyquist est d'autant plus proche du point critique que la racine est plus proche de l'axe imaginaire . Ainsi , pour juger du degré de stabilité d'un asservissement est on amené à apprécier la distance du diagramme de Nyquist au point critique en faisant varier séparément le module et la phase de $F(j\omega)$. dans ces conditions on définit une marge de gain et une marge de phase .

a) marge de gain : la marge de gain est le nombre de décibels dont le gain statique peut augmenter en boucle ouverte sans provoquer l'instabilité.

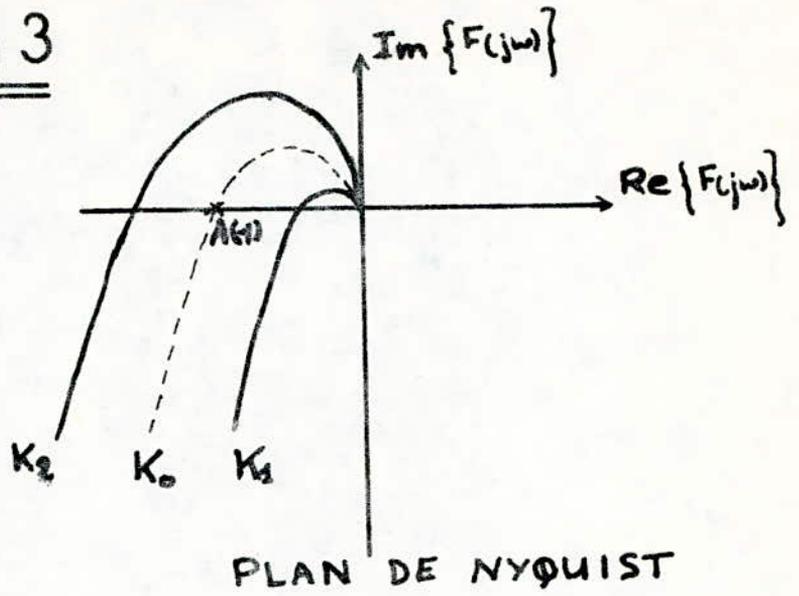
b) marge de phase : la marge de phase est le déphasage supplémentaire qui ferait passer la courbe de Nyquist de l'autre côté du point critique.

c) Exemple de calcul : Soit un système asservi dont la fonction de transfert en boucle ouverte $F(p)$ a pour lieu de Nyquist celui de la

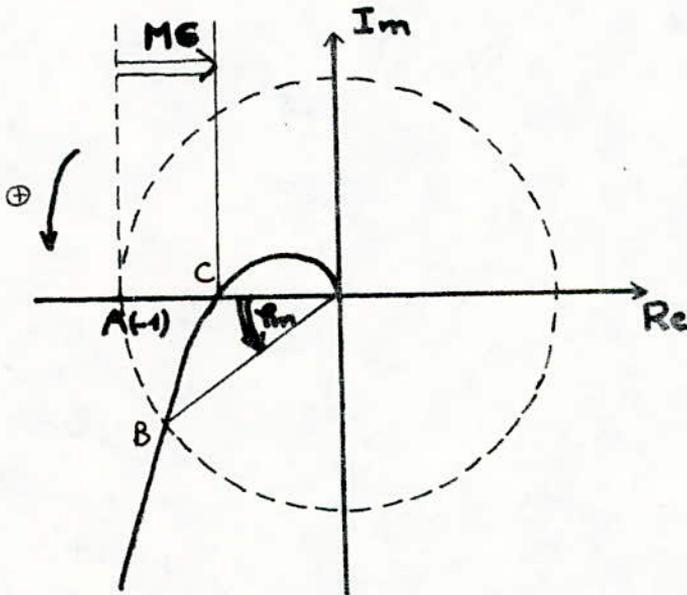
Fig 3



Plan de Laplace



PLAN DE NYQUIST



MG: marge de gain.
 P_m : marge de Phase.

Fig 4

Fig 4.

- pour rendre instable le système, sans modifier sa phase, il faut faire une homotétie de rapport K_m telle que :

$$K_m = \frac{OA}{OC} = \frac{1}{OC}$$

$$K_m(\text{dB}) = -20 \cdot \log_{10} / OC / \text{marge de gain}$$

pour un système stable (OC inférieur à 1) : donc $K_m(\text{dB})$ supérieur à 0
 " " " Instable (OC supérieur à 1) : donc $K_m(\text{dB})$ inférieur à 0
 - Si on trace un cercle de centre O et de RAYON égal à 1 (fig 4) celui-ci coupe le lieu de Nyquist en un point B. La marge de phase est alors égale \widehat{BOC} . en effet cet angle représente la rotation que doit effectuer le diagramme pour obtenir, à gain constant, l'instabilité.

d) Remarques :

- l'utilité d'un diagramme de Nyquist (ou de Black) est de permettre la lecture directe sur le diagramme des marges de gain et de phase. Néanmoins si l'on connaît ω_c (Pulsation relative au point C) et ω_b (pulsation relative au point B) on peut calculer :

$$\text{la marge de gain : } K_m(\text{dB}) = -20 \log_{10} |F(j\omega_c)|$$

$$\text{la marge de phase } \phi_m = \phi_b + 180^\circ \quad (\phi_b = \text{Arg}(F(j\omega_b)))$$

- pour que la stabilité d'un système asservi persiste malgré des variations imprévues du gain en boucle ouverte et malgré certains retards parasites, on admet généralement les valeurs suivantes :

$$10 \text{ dB} < K_m < 15 \text{ dB} \text{ et } 40^\circ < \phi_m < 50^\circ$$

E) Application aux systèmes à fonctions de transfert en boucle ouverte

$$F(p) = K / ((b_0 + b_1 p)(a_0 + a_1 p + a_2 p^2))$$

a) Fonction de transfert des systèmes linéaires fondamentaux :

- Intégrateur pur $H(p) = K/p$

- Système du 1^{er} ordre $H(p) = \frac{K}{1 + T_p p}$

- Système du 2^{em} ordre $H(p) = \frac{K}{1 + 2m T_o p + T_o^2 p^2}$

$\omega_o = 1/T_o$ pulsation propre ; m : coefficient d'amortissement

b) Choix de $F(p)$:

Un système physique linéaire quel qu'il soit est généralement formé par la mise en cascade de systèmes linéaires fondamentaux. en pratique le degré du système dépasse rarement 3. Le choix de

$F(p) = K / (b_0 + b_1 p)(a_0 + a_1 p + a_2 p^2)$ est donc tout indiqué pour une utilisation plus fréquente du présent travail.

c) Etude de $F(p)$.

$$F(p) = K / (b_0 + b_1 p)(a_0 + a_1 p + a_2 p^2) = \text{Re}^{j\theta}$$

C1 systèmes du 1er degré :

- sans pôle à l'origine : ($a_1 = a_2 = 0$)

$$F(p) = \frac{K}{a_0 b_0 (1 + b_1 p)} \quad p = j\omega \quad u = b_1 \omega / b_0 : \text{ pulsation reduite.}$$

$K_0 = K / a_0 \cdot b_0 =$ gain statique.

$$F(j\omega) = \frac{K_0}{1 + u^2} + j \left(\frac{-u \cdot K_0}{1 + u^2} \right) = \text{Re}^{j\theta}$$

$$R = \frac{K_0}{(1 + u^2)^{1/2}} \quad \theta = \theta_0 - \text{Arctg}(u)$$

tracé de Nyquist : c'est un demi cercle de centre $(K_0/2, 0)$: et de rayon $K_0/2$. (fig 5)

Marge de phase

$$R(u) = 1 \quad u_1 = (K_0^2 - 1)^{1/2}$$

MP = $180^\circ - \text{Artg}(u_1)$: marge de phase.

- avec pôle à l'origine ($a_1 = a_2 = b_0 = 0$)

$$F(j\omega) = \frac{K/b_1}{j\omega} \quad K_0 = K/b_1 : \text{ gain statique}$$

$$F(j\omega) = 0 + j(-K_0/\omega) = \text{Re}^{j\theta}$$

$$R = K_0 / \omega \quad \theta = \theta_0 = -\pi/2$$

tracé de Nyquist (fig6)

marge de phase MP = 90°

C2 Systèmes du 2eme ordre :

- sans pôle à l'origine : ($b_1 = 0$)

$$F(j\omega) = \frac{K_0}{1 + 2j\mu u - u^2}$$

$K_0 = K/b_0 \cdot a_0$: gain statique

$$u = \frac{(a_2)^{1/2}}{a_0} \cdot \omega \quad \text{pulsation reduite}$$

$$m = \frac{a_1}{2 \sqrt{a_0 \cdot a_2}} : \text{ coef d'amortissement .}$$

$$F(j\omega) = \frac{K_0(1 - u^2)}{(1 - u^2)^2 + 4m^2 u^2} + j \frac{-K_0 \cdot 2mu}{(1 - u^2)^2 + 4m^2 u^2} = \text{Re}^{j\theta}$$

$$R = \frac{K_0}{((1 - u^2)^2 + 4m^2 u^2)^{1/2}} \quad \theta = \theta_0 - \text{Artg} \frac{2mu}{1 - u^2}$$

Trace de Nyquist (fig⁷)

(8)

marge de phase

$$R(u^2) = 1$$

$$u^2 = \left((1-2m^2) + ((1-2m^2)^2 + Ko^2 - 1)^{\frac{1}{2}} \right)^{\frac{1}{2}}$$

$$MP = 180 + \phi(u^2)$$

- avec un pôle à l'origine : (b1=ao=0)

$$F(jw) = \frac{K}{jw \cdot bo \cdot a1 (1 + \frac{a2}{a1} jw)}$$

Ko = $\frac{K}{bo \cdot a1}$: gain statique . u = $\frac{a2}{a1} w$: pulsation reduite

$$F(ju) = \frac{Ko \cdot a2}{a1 \cdot ju(1+ju)} = - \frac{Ko \cdot a2}{a1(1+u^2)} + j \frac{-Ko \cdot a2}{a1 \cdot u(1+u^2)} = Re^{j\phi}$$

$$R = \frac{Ko \cdot a2}{a1 \cdot u(1+u^2)^{\frac{1}{2}}} \quad \phi = \phi = \frac{PI - \text{Artg}(u)}{2}$$

lieu de Nyquist (fig⁸)

la courbe presente une asymptote verticale le : $X = -\frac{Ko \cdot a2}{a1}$

En effet $\lim_{u \rightarrow 0} \text{Re}(F(ju)) = -\frac{Ko \cdot a2}{a1}$ et $\lim_{u \rightarrow 0} \text{Im}(F(ju)) = \text{infini}$

Marge de phase : R(u'2) = 1

$$u'^2 = \left((-1 + \frac{Ko \cdot a2}{a1})^2 \right)^{\frac{1}{2}}^{\frac{1}{2}}$$

- avec 2 poles à l'origine : (b1=ao=a1=0)

$$F(jw) = \frac{-K}{bo \cdot a2 \cdot w^2} = -\frac{Ko}{w^2} = Re^{j\phi}$$

Ko = $\frac{K}{bo \cdot a2}$ gain statique

$$R = Ko \quad \phi = \phi = -PI$$

MP = 0 système toujours à la limite d'instabilité

C3 Systèmes du 3eme ordre :

- Sans pôle à l'origine :

$$F(jw) = \frac{K}{(bo+b1;jw)(ao+a1 \cdot jw - a2 \cdot w^2)}$$

S = max $\left(\frac{bo}{b1}, \left(\frac{ao}{a2} \right)^{\frac{1}{2}} \right)$ u = $\frac{w}{s}$ pulsation reduite

$$F(ju) = \frac{K(bo \cdot ao - bo \cdot a2 \cdot s^2 \cdot u^2 - b1 \cdot a1 \cdot s^2 \cdot u^2)}{(bo^2 + b1^2 s^2 u^2) \left((ao - a2 s^2 u^2)^2 + a1^2 s^2 u^2 \right)} +$$

$$j - \frac{K(b_0 \cdot a_1; s u + b_1 s u (a_0 - a_2 s^2 u^2))}{(b_0^2 + b_1^2 s^2 u^2)((a_0 \cdot a_2 s^2 u^2)^2 + a_1^2 s^2 u^2)} = \text{Re}^{j\theta} \quad (9)$$

$$R = \frac{K}{((b_0^2 + b_1^2 s^2 u^2)((a_0 \cdot a_2 s^2 u^2)^2 + a_1^2 s^2 u^2))^{\frac{1}{2}}}$$

$$\theta = f = -\text{Artg} \left(\frac{b_1 s u}{b_0} \right) - \text{Artg} \left(\frac{a_1 s u}{a_0 - a_2 s^2 u^2} \right) = -\text{Artg} \left(\frac{S U (a_0 \cdot b_1 + a_1 \cdot b_0 - a_2 b_1 s^2 u^2)}{b_0 \cdot a_0 - b_0 \cdot a_2 s^2 u^2 - b_1 \cdot a_1 s^2 u^2} \right)$$

$K_0 = \frac{K}{b_0 \cdot a_0}$: gain statique .

Tracé de Nyquist (fig)9

Marge de phase determination graphique (sur le microordinateur par recherche)

Marge de gain

$$\text{Im} (F(j\omega)) = 0 \quad U_3 = \left(\frac{b_0 a_1 + b_1 a_0}{a_2 S^2} \right)^{\frac{1}{2}} \quad \text{MG} = -20 \log(R(U_3))$$

= Avec un pôle à l'origine : (10)

$$b_0 a_0 = 0 \quad (b_0 \neq a_0)$$

Prenons le cas où $b_0 = 0$ (pour $a_0 = 0$ le calcul est le même)

$$F(j\omega) = \frac{K}{a_0 b_1 j\omega (1 + j \frac{a_1}{a_0} \omega - \frac{a_2}{a_0} \omega^2)}$$

$$K_0 = \frac{K}{a_0 b_1} : \text{gain statique}$$

$$U = \left\{ \frac{a_2}{a_0} \right\}^{1/2} \omega : \text{pulsation réduite}$$

$$m = \frac{a_1}{2 \sqrt{a_0 a_2}} : \text{coef d'amortissement}$$

$$S = \left\{ \frac{a_0}{a_2} \right\}^{1/2}$$

$$F(j\omega) = \frac{-2m K_0}{S((1-u^2)^2 + 4m^2 u^2)} + j \frac{-K_0(1-u^2)}{Su((1-u^2)^2 + 4m^2 u^2)} = \text{Re}^{j\phi}$$

$$R = \frac{K_0}{su((1-U^2)^2 + 4m^2 u^2)^{1/2}}$$

$$\phi = f = -\frac{\text{PI}}{2} - \text{Artg} \frac{2mu}{1-u^2}$$

lieu de Nyquist

Il présente une asymptote verticale $X = -\frac{K_0 a_1}{a_0}$

$$\text{En effet } \lim_{u \rightarrow 0} \text{Re}(F(j\omega)) = -\frac{2m K_0}{S} = -K_0 \frac{a_1}{a_0}$$

$$\text{et } \lim_{u \rightarrow 0} \text{Im}(F(j\omega)) = \text{Infini}$$

Marge de Phase :

déterminé graphiquement (se fait par recherche sur le microordinateur)

(Fig 10)

Marge de gain

$$\text{Im}(F(j\omega)) = 0 \iff U = 1$$
$$\text{MG} = -20 \text{Log}_{10}(R(1))$$

- Avec 2 poles à l'origine (Fig 11)

$$b_0 = a_0 = 0$$

$$F(p) = \frac{K}{a_1^2} \frac{1}{p^2} \left(1 + \frac{a_2}{a_1} p \right)$$

$$F(j\omega) = \frac{K_0}{-\omega^2 \left(1 + j \frac{a_2}{a_1} \omega \right)}$$

$$U = \frac{a_2}{a_1} \omega \quad K_0 = \frac{K}{b_1 a_1} \quad \text{gain statique}$$

$$F(j\omega) = - \frac{K_0 a_2^2}{a_1^2 \omega^2 (1 + u^2)} + j \frac{K_0 a_2}{a_1^2 \omega (1 + u^2)} = R e^{j\phi}$$

$$R = \frac{K_0 a_2^2}{a_1^2 \omega^2 (1 + u^2)^{1/2}} \quad \phi = \pi - \text{Arctg } u$$

$$\text{II} : \text{PI}$$

Tracé de Nyquist(Fig 11)

Le système n'est jamais stable. Il nécessite donc un régulateur

C4) Remarques :

Les systèmes du 1er et 2ème degré sont toujours stables. Ils nécessitent seulement une marge de phase de 45°. Par contre les systèmes de 3^{ème} ordre ont une stabilité qui dépend fortement du gain statique.

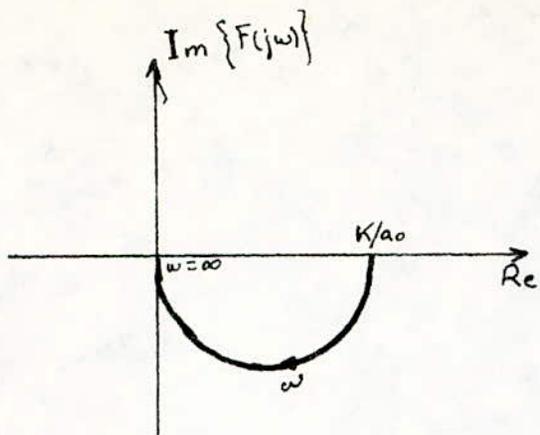


Fig 5

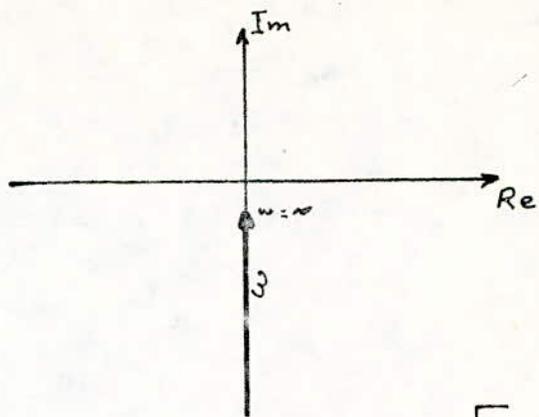


Fig 6

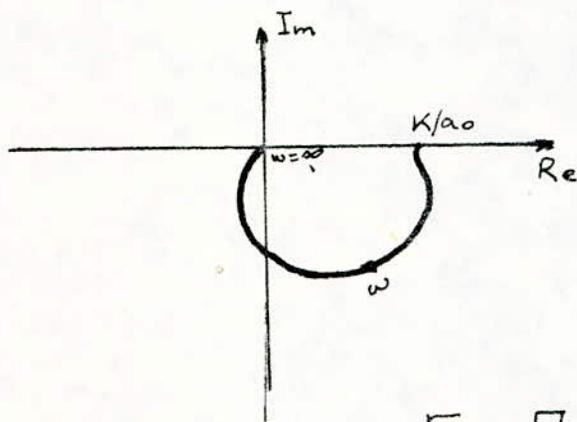


Fig 7

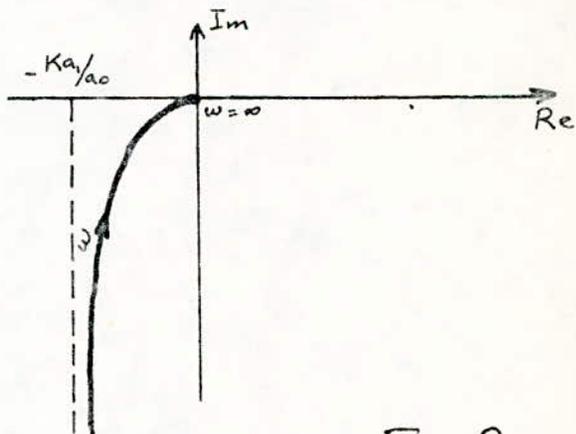


Fig 8

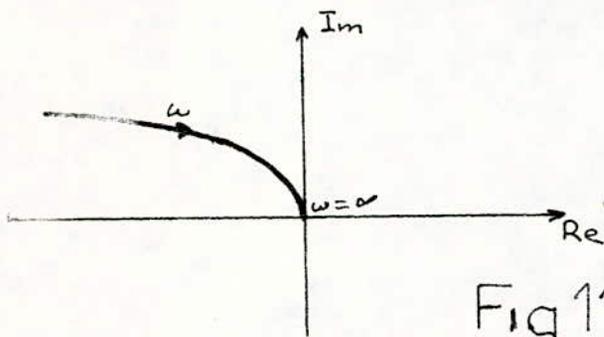


Fig 11

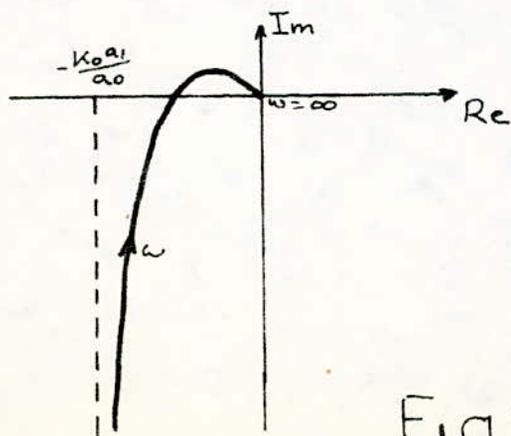


Fig 10

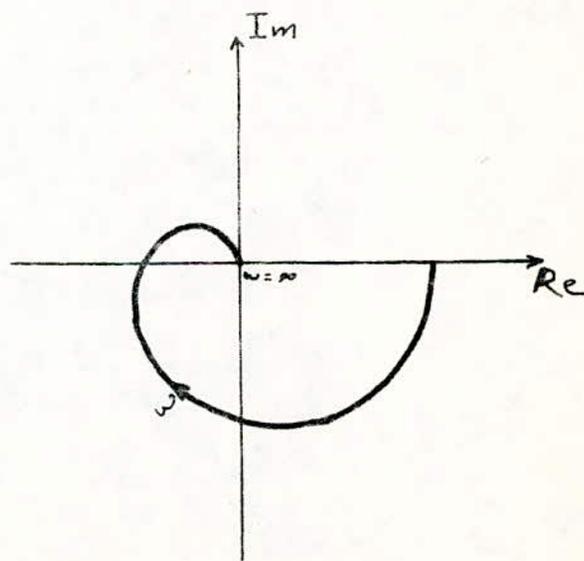


Fig 9

2^e PartiePresentation de l'APPLE II PLUS

	Page
A) Introduction	13
B) Le Hardware	13
B1) Unité centrale.....	13
B2) La mémoire centrale.....	15
B3) Les unités Periferiques'	17
C) Le SOFTWARE /... ..	17
C1) Le Software de base	17
C2) Le software d'application	18
C3) Les langages disponibles sur L'APPLE PLUS....	18

- PRESENTATION DE L'APPLE II PLUS :

(13)

A) INTRODUCTION :

L' Apple II plus est un microordinateur transportable de fabrication américaine. Il a été conçu en 1979 par STEVEN JOBS et STEPHEN WORNIAK. Ses domaines d'utilisation sont nombreux (scientifiques, Gestion, Application domestiques...).

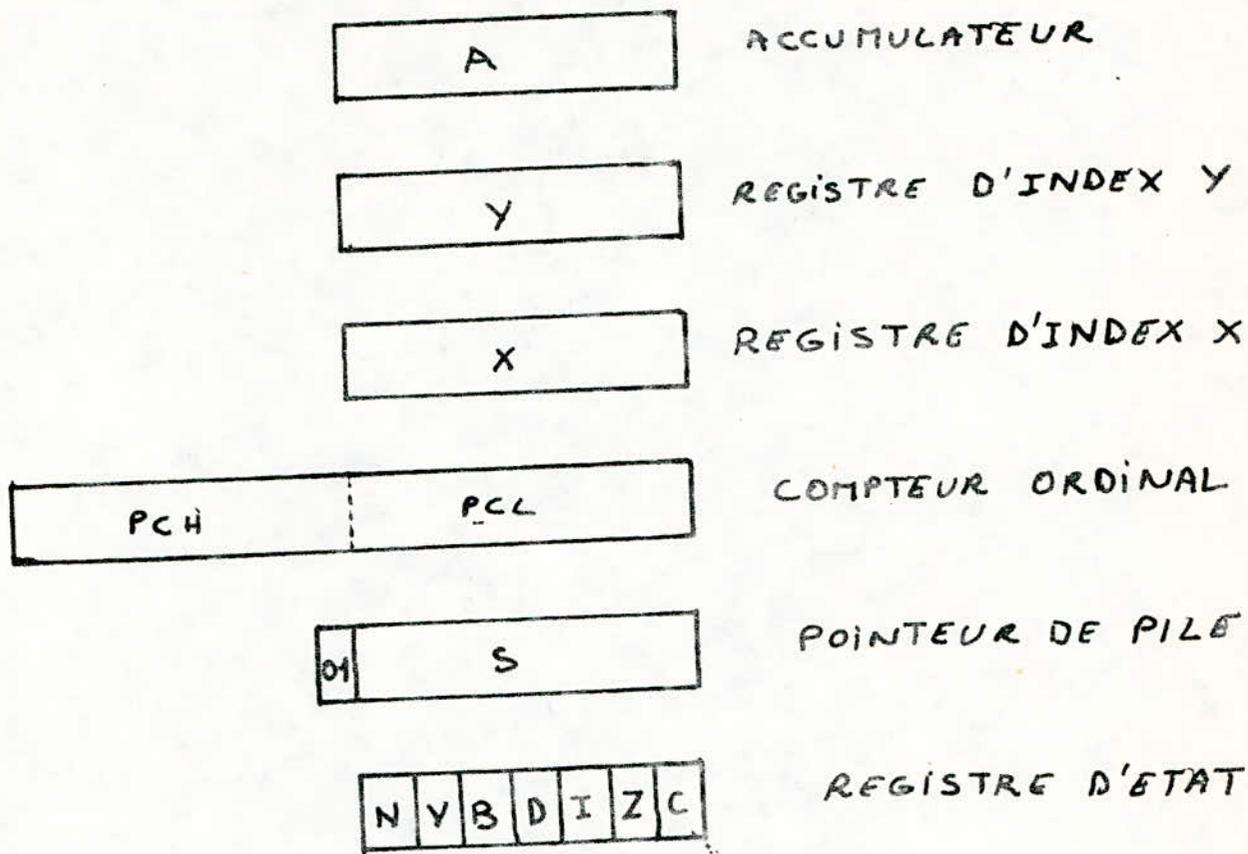
B) Le HARDWARE

B. 1 - L'Unité Centrale.

L'Unité Centrale de l'Apple II est composée autour du microprocesseur MCS 6502/S* 6502.

W B. 1.1 Fiche Technique du MCS 6502.

- Modèle MCS 6502/SY 6502.
- Fabriqué par MOS Technologie, Inc. Synerlek, Rockwell.
- Nombre d'Instruction 56.
- Modes d'adressage 13.
- Voltage : **5 Volts.**
- Puissance : 0.25 Watt .
- Horloge : 1.023 MHz.



* INDICATEURS:

N: signe

V: Depassement

B: Arrêt

D: Decimal arithmetic

I: Interruption

Z: zero

C: retenue.

* Les BUS:

- BUS d'Adresse 16 Bits (paralleles).

- BUS de données 8 bits (paralleles, bidirectionnels).

ORGANISATION MEMOIRE	
N° Page	
0 1 2 ⋮ 190 191	RAM (48K)
192 193 ⋮ 199 200	I/O (2K)
207	I/O ROM (2K)
208 209 ⋮ 254 255	ROM (12K)

La M.C. est constituée de 256 'pages memoires' reparties en:

RAM → 192 pages

I/O → 16 pages

ROM → 48 pages

1 page → 256 'locations memoires'

B 3) - Les Unités Perifériques :

Elles comprennent les dispositifs d'entrée, de sortie et de stockage.

B 3 1)- Le Clavier :

Comprend 52 Touches à caractère alphanumerique.

B 3 2)- La Visualisation : Elle est assurée par 1 TV N/B " Sonelec " qui reçoit directement la sortie video du ordinateur sur son ampli video.

B 3 3)- Les connecteurs d'interfaces : Ils sont au nombre de 8 et sont reliés au BUS de l'Apple II. Ils peuvent recevoir différentes cartes d'interfaces pour adapter des unités de disques, une imprimante.....

B 3 4)- Les drivers de disques : sont au nombre de 2 et sont pilotés par un seul contrôleur.

La mémoire disponible n'étant que de 48 K, Celle-ci ne répond pas aux besoins de l'utilisateur. Aussi, faut-il recourir à d'autres espaces mémoires (mémoires auxiliaires ou de masse). D'après l'utilité des disquettes. La capacité d'une disquette est de 140 K bytes.

C)- Le SOFTWARE :

C 1)- Le Software de base (ou support logiciel).

La tâche du programmeur est de réaliser et charger des programmes que le microprocesseur exécute.

Ces programmes sont appelés " Programmes d'Application ".

Pour faciliter la réalisation de ces programmes un ensemble de programmes d'aide est nécessaire. Cet ensemble constitue le support logiciel.

Il comprend :

C 1 1)- Le programme d'exploitation :

C 1 1 1)- Le programme moniteur :

Il gère l'enchaînement des tâches et détecte les commandes de l'opérateur tout en y répondant correctement. Il permet ainsi l'accès aux disques, à organiser la mémoire etc.....

C 1 1 2)- Le programme éditeur :

C'est un programme permettant de composer, modifier et stocker sur une mémoire de masse, un texte, à partir d'un terminal conversationnel (clavier et écran). Le texte ainsi composé peut être un programme, un fichier de données, un message, etc.....

C 12) - Les Programmes de traduction :

Ils assurent la transcription d'un programme source en un programme objet. Suivant le langage utilisé (assembleur ou évolué), le programme est appelé assembleur compilateur ou interpréteur. Pour l'Apple II on dispose d'un mini assembleur, d'un interpréteur pour l'utilisation de l'Applesoft Basic et d'un compilateur pour le pascal.

C 13)- Les Programmes de diagnostic (DEBUG)

Ils permettent de detecter les erreurs eventuelles de logique. Dans l'Apple II cette fonction est remplie par les programmes :

- SYSTEM SYNTAX pour les langages évolués.
- 6560 ERRORS pour le langage assembleur.

C 14 Autres Programmes d'assistances :

En plus des programmes précités, l'apple II plus presente d'autres programmes d'aide comme les programmes LINKER (sert à lier différents programmes), FILER (sert à defiler le contenu d'une memoire par exemple), etc.....

C 2 Le Software d'application :

Les programmes élaborés par l'utilisateur font appel à des fonctions trigonometriques, logiques nemuriques, graphiques. Ce qui nécessite des programmes d'aide.

Ces programmes sont fournis par le fabricant.

Pour le Pascal ces programmes sont sur disquettes sous les noms :

- SYSTEM LIBRARY (pour les fonctions trig, logiques,.....)
- SYSTEM CHARSET (programmes permettant l'écriture sur la page graphique)

C 3 Langage disponibles dans l'Apple II :

- Le langage machine.
- Le langage assembleur.
- le Basic Applesoft (Interprété resident
- le Basic interpreter (interpreté sur disque)
- le FORTRAN (absent au niveau de l'Ecole)
- le Pascal (compilé sur disque).

La chaine de microprogrammation incorporée est l'applesoft.

Les autres langages évolués nécessitent une carte de programmation.

Le langage utilisé (Pascal)	
A1) Choix du langage	20
B) Le langage Pascal?.....	20
B1) Introduction.....	20
B2) Constitution d'un programme en Pascal.....	20
C) Graphisme à l'aide de 14APPLE II PLUS.....	23

LE LANGAGE UTILISE :A -) CHOIX DU LANGAGE :

Comme nous l'avons signalé au début, le but de ce travail est de réaliser un programme en langage évolué qui permet de tracer le lieu de Nyquist et de donner la marge de stabilité pour un certain type de fonction de transfert. Mais il faudrait que l'exécution de ce programme donne le maximum de résultats souhaités (graduation de la courbe en pulsation, phase et rayon) tout en mettant le moins de temps possible.

Tout d'abord nous avons opté pour le Basic Applesoft pour les raisons suivantes :

- Simplicité du langage.
- Sa présence directe dans la mémoire de l'APPLE II. plus. Cela permet d'avoir un espace plus grand dans la disquette.
- Son large éventail d'utilisation. En effet il permet des traitements sur fichier et présente également des possibilités étendues en mathématique, logique et graphisme (GR et HGR).

Cependant nous avons été confronté à 2 problèmes essentiels :

- Le temps d'exécution était très long (le tracé d'une courbe dépassait facilement les 6 minutes). En effet, la transcription du programme source en programme objet se faisant par un programme interpréteur (traduction du programme instruction par instruction)

- Impossibilité d'écriture sur la partie graphisme à moins de créer un fichier où l'on doit reproduire les formes des caractères à afficher ce qui d'un côté n'était pas facile à réaliser et d'un autre côté alourdissait le programme. Cela ne permettait donc pas de graduer la courbe.

Nous avons donc été amené à utiliser le langage Pascal (Plutôt le Applepascal) qui lui utilise le compilateur comme transcritteur du programme source en programme objet, et possède des procédures d'écriture sur la partie graphique.

B - LE LANGAGE PASCAL :B1)- INTRODUCTION:

C'est un langage qui a fait son apparition en 1968. Il a été élaboré par KATHLEEN JENSEN ET NIKLAUS WIRTH.

Le premier compilateur fut opérationnel en 1970. Ainsi après trois années d'expériences et de retouches, c'est en 1973 qu'il fut publié. L'APPLE-PASCAL est une des variantes du Pascal particulière à l'APPLE II.

Les différences existant entre le PASCAL et L'APPLE PASCAL seront signalées au fur et à mesure.

B2)- CONSTITUTION D'UN PROGRAMME EN PASCAL :

Un programme Pascal (ou APPLE-PASCAL) se compose d'un en-tête et d'un corps appelé Bloc- (Voir 3ème Partie).

Dans l'en-tête on donne un nom au programme et la liste de ses paramètres. Ces paramètres sont des variables qui contiennent les arguments et les résultats de calcul :

.../...

Le Bloc lui est constitué de 6 parties :

- Partie Déclaration d'étiquette.
- Partie Définition des constants
- Partie Définition de Types
- Partie Déclaration de variable
- Partie Déclaration de fonction et procédures.
- Partie ~~Introductions.~~ **Instructions.**

Excepté la dernière partie, toutes les autres peuvent être absentes.

a) - EN-TÊTE D'UN PROGRAMME :

L'en-tête sert à donner un nom au programme et une liste de paramètres à travers lesquels le programme communique avec son environnement.

b) - PARTIE DECLARATION D'ETIQUETTE :

Afin de pouvoir accéder à n'importe quel niveau du programme dans le cas par exemple, d'une introduction de branchement (GOTO), toute instruction peut être précédée d'une étiquette suivie d'un signe (=).

Toutes les étiquettes utilisées doivent être définies dans cette rubrique avant leur utilisation.

Cette rubrique, introduite par le mot clé LABEL a pour forme
LABEL étiquette initiale, étiquette finale.

c) - PARTIE DEFINITION DES CONSTANTES :

Celle-ci permet l'introduction d'un identificateur comme synonyme d'une constante.

Cette rubrique commençant par le mot clé CONST a pour forme :

CONST identificateur 1 = constante \$, identificateur 2 =
constante 2 ; ou constante 1, constante 2, peuvent être soit 1
nombre, soit une chaîne de caractères.

d) - PARTIE DEFINITION DE TYPE :

En Pascal un type de donnée (Reels, entiers, caractères,) peut être soit directement décrit dans la partie variable (lorsque c'est 1 type standard) soit reperé par un identificateur de type qui permet d'en créer de nouveau. Chaque définition introduit un ensemble de valeurs et l'identificateur qui lui est associé. La forme générale de cette rubrique est de la forme :

TYPE identificateur = (Type), identificateur = (type);

Exp : TYPE couleur = (jaune, rouge, vert, bleu) ;
Sexe = (Masculin , Féminin) ;

PARTIE DECLARATION VARIABLE:

Toute variable apparaissant dans une introduction doit avoir été déclarée au préalable dans une rubrique déclaration de variables.

Celle-ci associe un identificateur et un type de donnée à chaque nouvelle variable.

Le mot réservé VAR introduit cette rubrique - elle est de la forme :

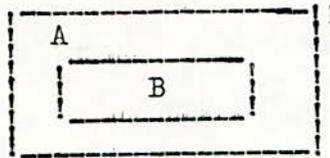
VAR identificateur 1, identificateur 2, : (Type 1) ;
identificateur 1, identificateur 2, : (Type 2) ;

Exp: VAR A,B,C,D,E : REAL (réels)
N,M,L : INTEGER (entiers)
H,I : CHAR (caractères)

.../...

Cette association identificateur / type est variable dans tout le Bloc contenant la déclaration, à moins que l'identificateur ne soit redéfini dans un bloc **imbriqué** (voir partie procédure et fonctions)?.

Supposons un bloc B imbriqué dans le bloc A .



Toute variable déclarée dans B est invisible par A, c'est à dire ne peut être utilisée en dehors du **sous** programme B.

Par contre toute variable déclarée dans A peut être utilisées en B. Enfin un même identificateur ne peut être déclaré 2 fois au même niveau avec un type différent ex: VAR A: REAL,
A: INTEGER ;

f)-PARTIE PROCEDURES ET FONCTIONS :

Les procédures sont des **sous** programmes activés par des instructions " PROCEDURE "

Les fonctions sont des procédures qui retournent une valeur et qui peuvent être utilisées dans des expressions, elles sont activées par des instruction : " FUNCTION "

Toute fonction ou procédure doit être définie avant son utilisation

f-1 - Les procédures -

f.1-1- Constitution :

Une déclaration de procédure consiste à définir un sous programme et à lui associé un identificateur de façon à ce que l'on puisse l'activer par une instruction procédure :

Une procédure a la même constitution qu'un programme à ceci près, elle commence par 1 en-tête procédure au lieu de l'en-tête de programme.

Cependant, et c'est là une des différences essentielles entre le Pascal et APPLE-PASCAL, on ne peut déclarer une procédure ou une fonction à l'intérieur d'une procédure (ou d'une fonction).

La notion de champ de visibilité que l'on a une dans la rubrique déclaration de variable est valable ici c'est à dire que toute variable déclarées dans une procédure, ne peut être utiliser à l'extérieur de celle-ci.

f.1-2- Les procédures standards:

Ce sont les procédures pré-déclarées contenues dans le LIBRARY-SYSTEM et qui sont supposées être définies dans un bloc qui engloberait tout le programme (celles-ci sont données en Annexe *Sur Langage reference Manual*)

f-2 - Les Fonctions -

Ce sont des sous programmes au même titre que les procédures qui calculent une valeur utilisable dans l'évaluation d'une expression. Ce qui augmente la souplesse de cette dernière.

Un appel de fonction se fait en donnant l'identificateur et la liste de parametres. Ces parametres sont des variables, expressions, procédures en fonction que l'on substitue aux parametres formels correspondant.

Une déclaration de Fonction à la même forme qu'un programme (en une procédure) à la difference près de l'en-tête de fonction qui s'écrit :

FUNCTION identificateur (groupe de parametres formels : Type du parametre) : TYPE du resultat.

Exp: FUNCTION RAYON (U : REAL) R REAL
qui veut dire qu'on est en présence d'une fonction Rayon à 1 variable réelle U et à valeur dans R.

f-2-1- Les fonctions Standard:

Chaque compilateur Pascal peut offrir des fonctions pré-déclarées. Celles-pré-définies dans l'APPLE II sont énumérées en Annexe :

Tout comme les procédures standard, elles sont supposées être définies dans un bloc qui engloberait tout le programme.

B -) GRAPHISME A L'AIDE DE L'APPLE II.

Le graphisme nécessite des programmes d'aide prédéfinis contenus dans une section appelé " TURTLEGRAPHICS" appartenant au SYSTEM-LYBRARY (LIBRAIRIE).

Avant tout tracé, il est impératif de déclarer immédiatement après l'en-tête du programme l'utilisation de cette section.

Cette déclaration prend la forme :

USES TURTLEGRAPHICS

Une telle déclaration nous permet d'englober notre programme par toutes les procédures et fonctions utilisables par le graphisme.

L'ECRAN :

La mémoire met à notre disposition 2 pages d'utilisations. L'une d'elle se trouve réservée au texte est appelée TEXMODE, la seconde elle est réservée au tracé on l'appelle GRAFMODE.

Cette dernière est un plan de coordonnées avec $X = 279$ et $Y = 191$ points, l'origine (ϕ, ϕ) de ce système d'axe est situé en bas et à gauche de l'Ecran.

Les coordonnées X et Y de ce plan ne peuvent être qu'entiers. On dispose de 3 procédures pour commander l'un ou l'autre des 2 modes, ce sont les procédures INITTURTLE, GRAFMODE ET TEXMODE.

PROCEDURE INITTURTLE :

C'est une procédure qui nous place dans la page graphique. Elle est utilisée au début de tout tracé.

En plus cette procédure permet d'initialiser le spot sur l'écran en le plaçant au centre de celui-ci et en le dirigeant vers la droite. Le Spot ici est sans couleur, c'est à dire qu'il ne laisse aucune trace en se déplaçant.

PROCEDURE GRAFMODE :

Elle permet le passage de la partie texte à la page graphe, dans le cas où l'on voudrait visualiser le contenu de celle-ci, sans pour autant perdre le texte.

PROCEDURE TEXTMODE :

C'est la procédure inverse de la précédente, elle permet le passage de la page réservée au graphisme vers la page réservée au texte sans perdre le tracé.

Remarque : une fois un tracé terminé, le programme passe automatiquement sur la partie "graphique"

L'UTILISATION DE COULEURS : PROCEDURE PENCOLOR :

Cette procédure nous permet de choisir la couleur avec laquelle le tracé pourrait être effectué.

Elle se présente sous la forme :

PENCOLOR (COULEUR).

Les différentes couleurs dont on dispose sont :

- 3 sortes de blanc : WHITE, WHITEA, WHITER .
- 3 sortes de Noir Black , Black 1, Black 2,

4 1 vert : GREEN	} necessitent un T.V. Couleur
- 1 violet : VIOLET	
4 1 orange : ORANGE	
- 1 Bleu : BLUE	

PROCEDURES DE DEPLACEMENT DU SPOT :

Pour un tracé, on dispose de 4 procédures de déplacement du spot :

TURNT0 - TURN -- MOVETTO -- MOVE --

PROCEDURE TURENTO :

Initialement le spot est au centre de l'écran et dirigé vers la droite (après l'initialisation par la procédure INITTURTLE). On utilise la procédure TURNT0 lorsqu'on veut orienter le spot vers une direction bien définie. Cette instruction se présente sous la forme :

TURNT0 (N)

Où N est un entier représentant en degrés la direction vers laquelle on pointe l'axe, le sens d'orientation est le sens trigonométrique.

Exm. TURNT0 (30) dirigera le spot dans la direction + 30 degrés.

PROCEDURE TURN :

Cette procédure permet la déviation du spot par rapport à sa position initiale.

Exm. TURN (30) dévie le spot de 30 degrés dans le sens trigonométrique par rapport à sa direction initiale.

PROCEDURE MOVETTO :

Permet de déplacer le spot en 1 point dont on désignera les coordonnées. Cette procédure se présente sous la forme :

MOVETTO (X, Y)

où X et Y sont les coordonnées du point auquel on veut déplacer le spot :

Remarque : Cette procédure ne fait pas changer de direction au spot.

PROCEDURE MOVE :

Elle se présente sous la forme :

MOVE (D)

où D est un entier représentant la distance de déplacement du spot, déplacement se fait dans la direction vers laquelle celui-ci est pointé.

ECRITURE SUR LA PAGE A GRAPHIQUE :

Nous disposons de 2 procédures nous permettant de positionner des caractères sur la page réservée au graphisme. Ces procédures appartiennent au programme d'aide nommé SYSTEM-CHARSET.

PROCEDURE WCHAR :

Elle se présente sous la forme :

WCHAR (CH)

où CH est un des 128 premiers caractères de l'ASC II. Chaque caractère est inscrit dans un carré de 7 points de côté.

Ex: Supposons qu'on veuille placer le caractère A au milieu du rectangle 279 x 191, on aura les instructions suivantes :

MOVETO (140,96) :

WCHAR ('A')

Remarque : C'est le coin inférieur gauche qui sera placé en (140,96)

PROCEDURE WSTRING :

Elle est présentée dans la forme :

WSTRING (S)

où S est une chaîne de caractère (type STRING)

Remarque : Cette procédure fait appel à la procédure WCHAR pour le positionnement de chaque caractère, par conséquent se sera le coin inférieur gauche du premier caractère qui sera situé au point considéré.

Remarques:

Il existe aussi une 3^e procédure appelée CHARTYPE que l'on utilise pour positionner les caractères en les encadrant par exemple par un carré blanc; ou en écrivant le caractère en noir inscrit dans un carré blanc, etc... Il existe 10 modes pour cette procédure. (voir APPLE SYSTEM LANGUAGE)

4° Partie

Page

A) Conception du programme Nyquist	27
1) Performances souhaitées.....	27
2) Problèmes rencontrés.....	27
3) Utilisation des programmes utilitaires.....	27
B) <u>FONCTIONS ET PROCEDURES</u> :	27
C) ORGANIGRAMME	28
D) Programme.....	29

1) Performances souhaitées.

le programme a été conçu de manière à ce que :

- Son temps d'exécution soit le plus court possible
- Le tracé soit continu et d'assez bonne qualité
- Le nombre d'informations fourni satisfasse l'utilisateur (marges de stabilité et graduation de la courbe en pulsation, rayon et phase)

2) Problemes rencontrés :

21) Les tracés de Nyquist different par leur forme suivant :

- Le degré du système
- pour un même degré suivant le nombre de poles à l'origine et éventuellement l'amortissement du Système

Cela nous a amené à segmenter le programme en plusieurs parties (ou procédures), de ne faire appel qu'à un nombre restreint de procédures dans le programme general et d'éviter ainsi des tests qui augmenteraient le temps d'exécution, cela a été facilité par la conception même du langage Pascal.

22) Pour les systèmes très selectifs (amortissement petit), ^{avec} un pas grand la courbe obtenue est cassée. Mais un choix d'un pas très petit augmenterait le temps d'exécution, d'ou le ^{compro}mi pas temps d'execution.

23) La graduation :

de la courbe en pulsation rayon et phase surchargeant le graphe, nous a amène à la transcrire sur la partie texte tout ^{en} reperant sur la partie graphique par des lettres. ^{la}

3) Utilisation des programmes utilitaires :

Le programme utilise des fonctions arithmetiques, logiques, trigonométriques et graphiques se trouvant sous forme de programme d'applications dans le système LIBRARY et des fonctions ou procédures permettant l'écriture sur la partie graphique se trouvant elles aussi, ^{sous forme} de programme d'application dans le SYSTEM, CHARSET

B) Fonction et Procedures :

Afin de permettre une compréhension meilleure du programme, nous avons exposé par de petits paragraphes, l'utilité de chaque fonction et procédure .

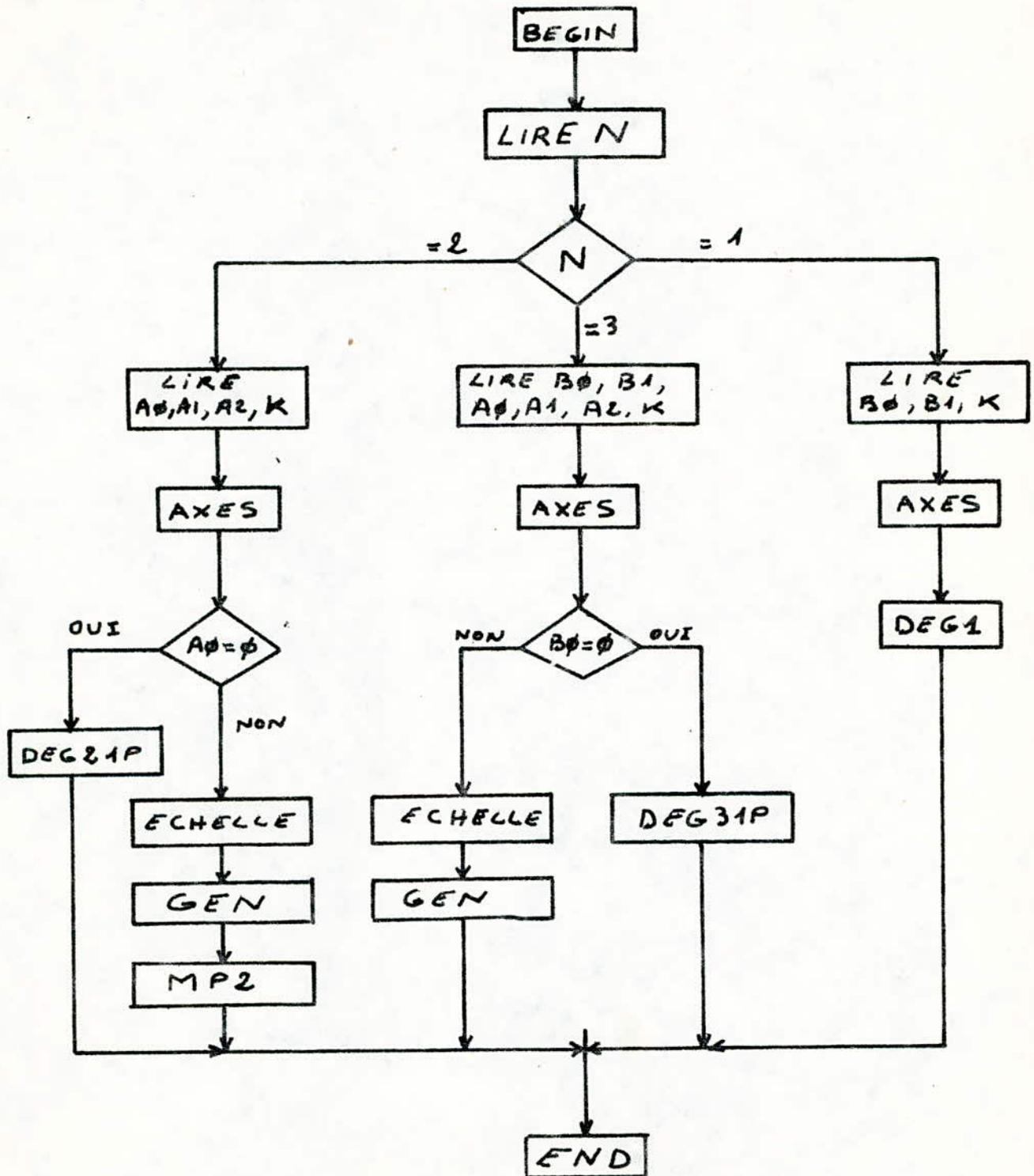
Fonction (RAY et Phase : permettent le calcul du rayon et de la phase de $F(p)$ ces deux fonctions sont rencontrées dans la plupart des procédures. Afin de ne pas surcharger l'écriture du programme

on fait appel à elles chaque fois qu'on en a besoin après les avoir déclarées au debut

- 2) Procédure Axes : Permet de tracer les axes (reel et imaginaire) et des droites de repère qui permettent la graduation de la courbe (tous les 30°).
- 3) Procédure échelle : La page graphique comporte 280×192 points. Suivant K et les constantes de temps de $F(p)$, le trace peut dans les cas extrêmes se réduire à un point, centre de l'écran (K faible) ou sortir totalement de ce dernier (K grand). Il est donc impératif de normaliser toutes les courbes de façon à les visualiser correctement. c'est l'utilité de cette procédure, Elle n'est utilisée que pour les systèmes de degré 2 et 3 sans pôles à l'origine.
- 4) Procédure M PHASE : permet de déterminer par la méthode de dichotomie la marge de phase pour les systèmes de degré 3.
- 5) Procédure GEN : permet le tracé du lieu de Nyquist avec graduation pour les systèmes de degré 2 et 3 sans pôles à l'origine. Elle fait appel à la procédure M PHASE
- 6) Procédure MP2 : permet le calcul direct de la marge de phase des systèmes de degré 2 sans pôles à l'origine !
- 7) Procédure DEG 3IP : permet le tracé du lieu nyquist avec graduation des systèmes de degré 3 avec pôle à l'origine, elle fait appel à M PHASE.
- 8) Procédure DEG 2 IP : permet le tracé du lieu de Nyquist avec graduation des systèmes de degré 2 avec pôle à l'origine, elle permet aussi le calcul direct de la marge de phase.
- 9) Procédure DEG 1 : permet le tracé du lieu de Nyquist et la détermination de la marge de phase des systèmes de degré 1 sans pôles à l'origine.
- 10) Procédure MGAIN : permet la détermination de la marge de gain des systèmes du 3^o ordre.

Organigramme.

(29)



PROGRAM .NYQUIST :

USES APPLESTUFF, TURTLEGRAPHICS, TRANSCEND ;

VAR B0, B1, A0, A1, A2, K, S, U, X, Y, R, MP, RMAX, U2 : REAL; ~~U2~~

~~INTEGER~~ I, N : INTEGER ;

FUNCTION RAY(U: REAL) : REAL ;

VAR D: REAL ;

BEGIN

D := SQR(S* U); SQR(A0 - A2*D) ;

D := SQR(A1*S*U) + D ;

D := D*(BQR(B0) +SQR(B1*S*U)) ;

RAY := -K/ SQRT(D) END ;

FUNCTION PHASE (U: REAL): REAL ;

VAR D: REAL ;

BEGIN

D := SQR(S*U) ;

D := U*S*(A0*B1 +A1*B0 -B1*A2*D)/(A0*B0 -A1*B1*D -B0*A2*D) ;

PHASE := - ATAN(D)

E N D ;

PROCEDURE AXES ;

VAR C : CHAR ;

L : INTEGER ;

BEGIN

INITTURTLE ;

PENCOLOR(NONE); MOVETO(0,96) ;

PENCOLOR(WHITE); MOVE(264); WSTRING('RE');

PENCOLOR(NONE); MOVETO (140,32) ;

WSTRING ('IM');

PENCOLOR(NONE); MOVETO (140,182) ;

PENCOLOR (WHITE); MOVETO(140,0) ; MOVETO (140,96) ;

IF B0*A0 = 0 THEN I:=3 ELSE I:=0 ;

REPEAT

L := 360-30*I ; I:=I+1 ; C:=CHR(64+I) ;

PENCOLOR (WHITE) ;

TURNT0 (L) ; MOVE(100) ; WCHAR(C) ;

PENCOLOR (NONE); MOVETO (140,96) ;

UNTIL I=3*N ;

PENCOLOR WHITE

E N D ;

PROCEDURE ECHELLE ;

(3A)

```
BEGIN
  IF N= 3 THEN
  (BEGIN
  (IF B0/B1 <= SQRT(A0/A2) THEN S:=SQRT(A0/A2) ELSE S:=B0/B1
  E N D ;
  IF N=2 THEN
  (BEGIN
  (B0:=1; B1:=0; S:= SQRT(A0/A2)
  ( E N D ;
```

```
RMAX : = 0; U:=0 ;
REPEAT
R:= RAY(U) ;
IF R > RMAX THEN RMAX: = R ;
U := U + 0.009 ;
UNTIL U > 1
E N D ;
```

PROCEDURE MPHASE ;

```
VAR A,B,P : REAL ;
L : INTEGER ;
BEGIN
MP := 180 ; A:=U-P; B:=(U+A)/2
WHILE ABS(R-1) > 0.01 DO
BEGIN
R := RAY(B) ;
IF R > 1 THEN
BEGIN
B : = (B+U)/2 ;
A : = B
E N D ;
ELSE
BEGIN
B := (B+A)/2 ;
U : = B
E N D ;
```

```

E N D ;
MP: = PHASE(B)*180/3.14 ;
  IF B(0) 0 THEN
BEGIN
  IF B(U2 THEN MP:=MP+180
E N D ;
  WRITELN ('MARGE DE PHASE = ', TRUNC (MP), 'DEG')
E N D ;

```

PROCEDURE GEN ;

```

VAR L : INTEGER; P; M: REAL ;
BEGIN
  U : = 0 ; I : = 0; L : = 1; MP:=0 ;
  U2 : = B0*A0/(B0*A2+B1*A1); U2 : = SQRT(U2)/S ;
  M = A1/SQRT(A0*A2)/2
  REPEAT
  R:= RAY(U) ;
  IF U>U2 THEN MP:=PHASE(U)-3.14 ELSE MP:=PHASE(U) ;
  X : = 140+90*R*COS(MP)/RMAX ;
  Y : = 96+90*R*SIN(MP)/RMAX ;
  MOVETO (TRUNC(X), TRUNC(Y) ;
  MP : = MP*180/3,14
  IF MP<=-30.I THEN
  BEGIN
  WRITELN(CHR(I+65), '(PULS W=;S*U, ' ;RAY=, 'PHASE=TRUNC(MP), ')) ;
  I = I + 1
  E N D ;
  IF (R<L)AND(N=3) THEN
  BEGIN
  MPHASE ;
  L : = 0
  E N D ;
  U : = U + P;
  UNTIL (I>2.N) AND (X> 137)
  E N D ;

```

PROCEDURE DEG 31P ;

```

VAR L : INTEGER ; P : REAL ;
BEGIN
  RMAX : = K*A1/SQR(A0); B0 : = 0; B1:=1 ;
  S = SQRT (A0/A2) ; L : = 1 ;

```

```
U2:=-A1*SQRT(3/A2/A0); U=SQR(U2)+4;U:=(SQRT(U)+U2)/2 ;
```

```
REPEAT
```

```
R := RA(U); MP:=PHASE(U)-3.14 ;
```

```
X := 140+R*YCOS (MP)*70/RMAX ;
```

```
Y := 96+R*SIN(MP)*70/RMAX ;
```

```
MOVETO (TRUNC(X), TRUNC(Y));
```

```
IF R<L THEN
```

```
  BEGIN
```

```
  MPHASE ;
```

```
  L := 0
```

```
  E N D ;
```

```
P := 0,009 ;
```

```
U := U+p;
```

```
UNTIL x) 137 ;
```

```
  I := 4 ;
```

```
REPEAT
```

```
U:=A1.SIN(I*0.523)/COS(I*0.523)/SQRT(A0*A2)/2
```

```
U2:=SQR(U)+1;U2:=SQRT(U2);U:=(U2-U) ;
```

```
R = RAY(U) ;
```

```
WRITELN(CHR(I+65), '(PULSW=', S.U, ';RAY=', PHASE=', -30.I, ')');
```

```
  I := I+1 ;
```

```
UNTIL I)8 ;
```

```
E N D ;
```

```
PROCEDURE DEG 21P ;
```

```
  BEGIN
```

```
  B1:=0;B0:=1; A0=0 S:=A1/A2 ;
```

```
  RMAX:=K/A1/S; U:=SQRT(1/3);
```

```
  REPEAT
```

```
  R := RAY(U); MP:=-ATAN(U) ;
```

```
  X := 140+90*R*SIN(MP)/RMAX ;
```

```
  Y := 96-90*R*COS(MP)/RMAX ;
```

```
  MOVETO (TRUNC(X), TRUN(Y));
```

```
  U := U+0,009 ;
```

```
  UNTIL X)137 .
```

```
  U := SQR(RMAX)+1; U:=SQRT(U)-1 ;
```

```
  U := SQRT(U);
```

```
  MP := 1.57-ATAN(U); MP:=MP.180/3.14 ;
```

```
  I := 4 ;
```

```
  REPEAT
```

(34)

```
U:=COS(I*.523)/SIN(-I*.523);
R:=1+SQR(U); R:=RMAX/SQR(R)/U;
WRITELN (CHR(I+64);(PLUS W='S',S*U,';RAY,'PHASE=,-30.I;')));
I := I+1;
UNTIL I>5;
WRITELN ('MARGE DE PHASE='MP'DEG');
TEXTMODE
E N D ;
```

PROCEDURE DEG 1

```
BEGIN
U :=0 ;
X :=196+50COS(U);
Y :=96-50*SIN(U);
MOVETO (TRUNC(X); ,TRUNC(Y));
U :=U+0,003 ;
UNTIL U)3.14;
I :=0 ;
REPEAT
U :=SIN(I*.523)/COS(0.523*I) ;
R : R/B0/SQRT(U*U+1); S := B0/R1 ;
WRITELN (CHR(I + 65);'(PULS W='S * U, ;RAY=';PHASE=,'-I*30,')'
I : = I +1;
UNTIL I ) 3
U : = SQR(K/B0)-1;U:=SQRT(U);MP:=ATAN(U)*180/3.14;
MP: = MP + 180;
WRITELN ('MARGE DE PHASE='MP,'DEG');
TEXTMODE
END;
```

```

PROCEDURE  MGAIN ;
VAR MG : REAL ;
BEGIN
U := (B0 * A1 + B1 * A0)/A2;  U := SQRT (U/S);
MG:= RAY(U);MG= -20 *LOG(MG);
WRITELN ('MARGE DE GAIN =' ,MG: 'DB');
TEXTMODE
END;
procedure MP2 ;
BEGIN
U 2:= SQR(A1+S/A0)-2;U= SQR(U2)+ 4+SQR(K/AQ)-4;
U:= -U2 + SQRT(U); U:=SQRT ( U/2);
MP := 1.57;
IF U < 1 THEN MP:= PHASE (U) + 3.14;
IF U > 1 THEN MP:= PHASE (U);
MP:= MP *180/3.14;
WRITELN ('MARGE DE PHASE =' ,MP, DEG');
TEXTMODE
END;


---


BEGIN
WRITE ('Indiquer le degré de F(P) ?');
READLN (N) ;
IF N = 1 THEN
BEGIN
WRITELN ('F(P) = K/( B0 + B1 * P)');
WRITELN ('INDIQUER B0,B1,K?');
READLN (B0,B1,K);
AXES;
DEG 1
END;

```

```

IF N = 2 THEN
BEGIN
WRITELN ('F(P)=K/(A0 + A1P + A2 +P * P)');
WRITELN (' INDIQUER A0,A1,A2,K');
AXES;
IF A0 ) 0 THEN
BEGIN
ECHELLE ;
GEN ;
MP 2
END;
ELSE DEG 21P
END;
IF N = 3 THEN
BEGIN
WRITELN ('- SI F(p) =K / (B0 + B1* p) A0 + A1*p + A2*p*p));
WRITELN ('INDIQUER B0 ,B1,A0,A1;A2,K?');
WRITELN('-SI F(p)=K/p(A0 +A1*p+A2* p*p)');
WRITELN ('DONNER 0,1,A0,A1,A2,K?');
READLN (B0,B1,A0;A1,A2,K) ;
AXES;
IF B0 ) 0 THEN
BEGIN
ECHELLE, GEN
END;
END.

```

(* * * * *)

5 eme Partie

Dans cette dernière partie sera traitée un exemple pour chaque type de fonction de transfert susceptible d'être tracé. Nous présenterons les contenus des pages GRAFMODE et TEXTMODE ainsi que le temps d'exécution pour chacun de ces exemples.

Nous donnerons enfin, toutes les conclusions tirées de ce travail.

REMARQUE : Les commandes restent les mêmes quelquesoit le degré de la fonction à tracer.

$$F(p) = \frac{30}{(1 + 0.07p)(1 + 0.0129p + 0.0026p^2)}$$

Apple: Commands: E(DIT); R(UN); F(ILE); X(CUTE); C(OMPILE)

Utilisateur: X

Apple: EXECUTE WHAT FILE?

Utilisateur: #4: NYQUIST (RETURN)

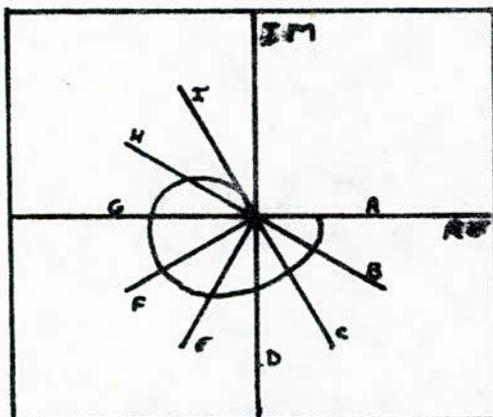
Apple: INDIQUER LE DEGRE DE F(P)?

Utilisateur: 3

Apple: INDIQUER B0, B1, A0, A1, A2, K?

Utilisateur^(*): 1 0.07 1 0.0129 0.0026 30

T_{exec} = 1 min.



GRAF MODE

A	(W=0; R=30; PHI=0)
B	(W=6.7; R=30.6; PHI=-30)
C	(W=13.2; R=38.5; PHI=-60)
D	(W=16.9; R=57.7; PHI=-90)
E	(W=18.7; R=71.4; PHI=-122)
F	(W=12.8; R=48.3; PHI=-150)
G	(W=21.6; R=46.7; PHI=-180)
H	(W=24.5; R=23; PHI=-211)
MARGE DE PHASE =	
MARGE DE GAIN = -34 DB	

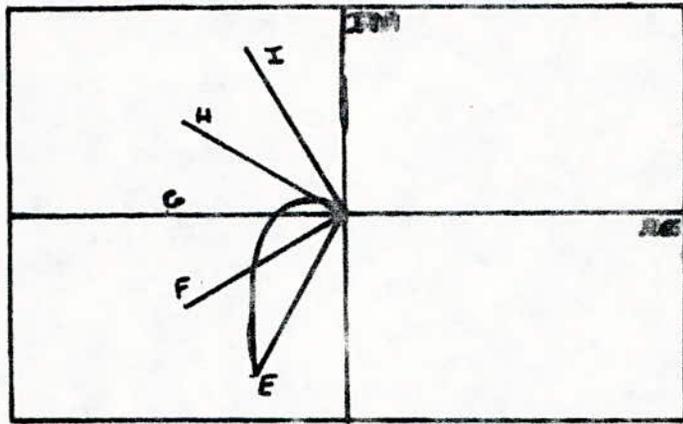
TEXT MODE

(*) : L'utilisateur doit taper 'RETURN' apres chaque donnée introduite.

$$F(p) = \frac{0.175}{p(1 + 8.25p + 2p^2)}$$

Apple: Donner $\phi, 1, A\phi, A1, A2, K$

Utilisateur: ϕ 1 1 8.25 2 0.175 (RETURN)



GRAFMODE

E (W = 1.18 E-2 ; R = 14 ; PHI = -120°)
F (W = 3.30 E-2 ; R = 5.1 ; PHI = -150°)
G (W = 1.19 E-1 ; R = 1.05 ; PHI = -180°)
H (W = 4.37 E-1 ; R = 1.09 E-1 ; PHI = -210°)
I (W = 1.22 ; R = 1.39 E-2 ; PHI = -240°)
MARGE DE PHASE = 42 DEG.
MARGE DE GAIN = 27 DB.

TEXTMODE

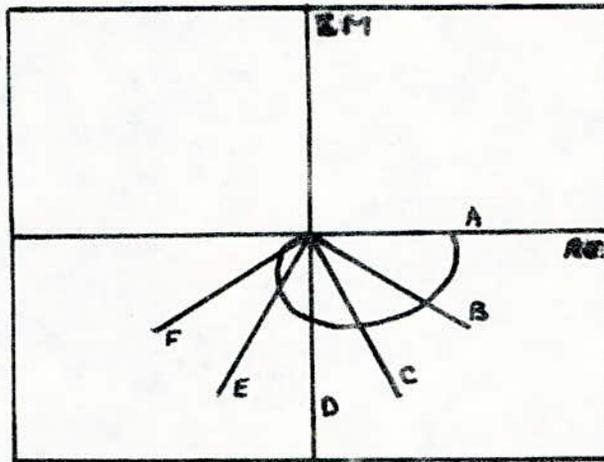
Texec. = 20 sec.

-40-

$$F(p) = \frac{11}{1 + 18p + 20p^2}$$

Apple : INDIQUER A0, A1, A2, K.

Utilisateur : 1 12 20 11 (RETURN)



GRAF MODE.

A (W=0 ; R= 11 ; PHI= 0)
B (W=4.6E-2 ; R= 9.9 ; PHI=-30)
C (W=1.1E-1 ; R= 7.2 ; PHI=-60)
D (W=2.2E-1 ; R= 4.1 ; PHI=-90)
E (W=4.5E-2 ; R= 1.7 ; PHI=-120)
MARGE DE PHASE = 46 DEG.

TEXT MODE.

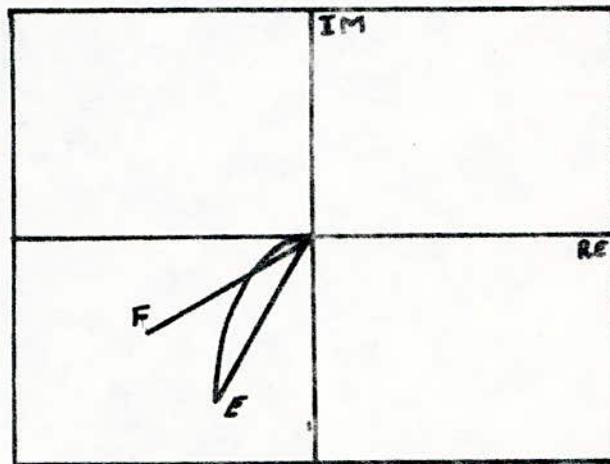
T_{exec.} = 1'35

-41-

$$F(p) = \frac{300}{p(1 + 0.066p)}$$

Apple: INDiquer A0, A1, A2, K

Utilisateur: 0 1 0.066 300 (RETURN).



GRAFMODE

$F(\omega = 9.6; R = 19.5; \text{PHI} = -120^\circ)$

$F(\omega = 26.06; R = 7.3E-1; \text{PHI} = -150)$

MARGE DE PHASE = 12 DEG.

TEXTMODE

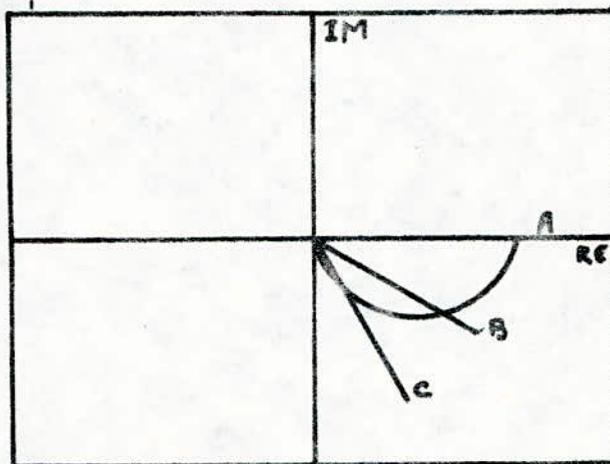
T_{exec.} = 15"

42

$$F(p) = \frac{28}{1 + 0.05p}$$

Apple: INDIQUER B0, B1, K.

Utilisateur: 1 0.05 28 (RETURN)



GRAFMODE

A (W=0 ; R=28 ; PHI=0)

B (W=11.5 ; R=24 ; PHI=30)

C (W=34.5 ; R=14 ; PHI=-60)

MARGE DE PHASE = 92 DEG.

TEXTMODE

T_{exec.} = 10 sec.

UTILISATION DU PROGRAMME NYQUIST

Le programme Nyquist est compilé et sauvegardé dans la disquette nommée FARAH. Aussi pour l'utiliser il faudra :

--Introduire cette disquette dans le driver 1 (driver de gauche) et la disquette APPLE SYSTEM \emptyset dans le driver 2 (driver de droite).

--Mettre le téléviseur et l'APPLE II sous tension

Après un bref instant, l'APPLE indique à l'utilisateur qu'il est sous contrôle de la disquette 1 et lui indique les commandes dont il dispose :

COMMANDS: E(DIT), R(UN), F(ILE), C(OMPILER), X(EXECUTE)

Ainsi commence le dialogue APPLE -- UTILISATEUR:

--UTILISATEUR: X (pour ordonner l'exécution d'un programme)

--APPLE II : EXECUTE WHAT FILE?

--UTILISATEUR: # 4: NYQUIST (RETURN)

--APPLE : RUNNING.....

INDIQUER LE DEGRE DE F(P) ?

A ce moment l'utilisateur donnera le degré du dénominateur de sa fonction de transfert (1, 2 ou 3)

L'APPLE indiquera les formes éventuelles de F(P) et demandera l'introduction des Coefficient de la fonction

L'UTILISATEUR introduira ses coefficients en respectant l'ordre indiqué et en les séparant par une tape RETURN

Alors apparaîtra sur l'écran la page GRAFMODE suivie lorsque le trace est terminé, par la page TEXTMODE .

CONCLUSIONS

Au terme de cette étude, nous concluons :

- Temps d'exécution : T max : 3 minutes

Nous pensons que c'est un temps assez bon dans la mesure où l'utilisateur obtient une courbe graduée en pulsation, rayon et phase. En plus, il trouve ces marges de stabilité calculées, ce qui lui permettra une complète connaissance de son système en un temps relativement court.

- En ce qui concerne le langage de programmation, nous vérifions ici la nette supériorité de l'Apple Pascal (compilé) sur le Basic Applesoft ceci du point de vue souplesse de programme et rapidité d'exécution.

- Pour ce qui est du matériel utilisé, Apple II et périphériques nous aurions préféré disposer pour le tracé d'une table traçante plutôt que d'un écran, ce qui aurait rendu ce travail encore plus pratique.

- Enfin, ce travail nous aura permis de nous familiariser au micro ordinateur Apple II et ses composants (mémoires, périphériques, drivers, disques ...), de mieux connaître des problèmes posés par la programmation et, d'avoir à notre actif deux nouveaux langages de programmation (Basic et Pascal) surtout pour le dernier dont le développement dans divers secteurs industriels devient considérable.

B I B L I O G R A P H I E

1°- Le langage de programmation Pascal

Philippe KRUCHTEN

2°- PASCAL: Manuel de l'utilisateur

K.JENSEN & N.WIRTH

3°- APPLE PASCAL : Langage Reference manual

4°- Microprocesseurs et Microordinateurs

R.LYON-CAEN & J.M. CROZET

5°- Architecture des miniordinateurs

A.G. LIPPIATT

6°- APPLE PASCAL : Opertins System reference manual

