NATIONAL AERONAUTICS & SPACE ADMINISTRATION
Langley Research Center, Hampton, Va.

NSG 1140. COMPUTER AUTOMATION FOR FEEDBACK SYSTEM DESIGN

Semi-annual Report
December 31, 1975

A complete automated computer program was prepared and hand delivered to NASA, Langley. The program included complete listings, a deck of cards, and several detailed examples, all in the form of actual computer outputs.

Mr. Isaac Horowitz, in a visit to NASA, Langley, gave an outline of the program. Mr. Patrick Rosenbaum spent several days at NASA, Langley, advising and assisting NASA personnel in its use by working with them at the NASA computer. A separate outline of the program, including explanation of the various steps, references to the mathematical techniques, used, etc., is included herein.

Present efforts are now in the improvement and refining of the more sophisticated parts of the program -- primarily, the automatic evaluation of a sub-optimal loop transmission, and the translation of time to frequency domain specifications.

MAR 1976
RECEIVED
NASA STI FACILITY
INPUT BRANCH

(NASA-CR-146638) COMPUTER AUTOMATION FOR FEEDBACK SYSTEM DESIGN Semiannual Report (Colorado Univ.) 59 p HC $4.50 CSCL 09B

N76-19835

Unclas

AUTOMATIC DESIGN PROGRAM FOR SINGLE LOOP SYSTEMS WITH UNCERTAIN PLANT PARAMETERS (ADSUP)

Introduction

This program is divided into four parts:

PART 1 is the translation of time domain boundaries (T.D.B.) into frequency domain bounds (F.D.B.).

PART 2 computes, for a given plant, the templates at various frequencies and stores them.

PART 3 derives the optimum $L_2(j\omega)$ (open-loop transfer function with excess of 2 poles over zero) which satisfies the bounds on $L(j\omega)$ due to F.D.B.

PART 4 derives the final $L_5(j\omega)$ (excess of 5 poles over zero). It then derives the prefilter $F(j\omega)$. Both L and F are obtained as rational functions in the complex variable s.

I. PART 1

Philosophy

In the first step, all second order systems (two poles, no zeros) are found, which satisfy the T.D.B. Real poles and zeros are then added, which continue to satisfy the T.D.B. Part 1 consists of the following components.

A. START1

This routine reads the T.D.B. data.

NT : number of T.D.B. points.

DELTA: increment (in seconds) in time .

BMAX : vector of NT maximum tolerable values.

BMIN : vector of NT minimum tolerable values.

Note: If NT<24 linear interpolation is used in order to have at least 24 points. A maximum of 100 points is allowed (NT≤100) .

B. TWO

This routine finds all second order systems which satisfy the T.D.B.

First, KRANGE computes $\omega_{n\,min}$ and $\omega_{n\,max}$ corresponding to

$\omega_{n\,min} < \omega_n < \omega_{n\,max}$ such that $\forall t$ $BMIN(t) \leq \mathcal{L}^{-1}\left(\frac{\omega_n^2}{s(s^2+2\omega_n s+\omega_n^2)}\right) \leq BMAX(t)$.

The vectors TMIN(•) and TMAX(•) are calculated on a set of selected frequencies (see TABCS). TMIN and TMAX denote respectively the minimum and maximum permitted frequency domain values of the system transfer function. Then, for $\omega_n \in [\omega_{n\ min}, \omega_{n\ max}]$, there is found the range (by calling KRANGE) of the damping $[\zeta_{min}$ denoted CMIN and ζ_{max} denoted CMAX] , which satisfies the T.D.B.; ω_n is incremented using step-size DW . At each step TMIN and TMAX are, if necessary, updated.

<u>Note</u>: WO is called the central frequency. If M=128 (number of points) (see TABCS) then $WO \triangleq W(65)$. In general $WO \triangleq W(\frac{M}{2} + 1)$.

1) <u>KRANGE (A,B,N)</u>

Input : A,B are the first guess for the range, N is number of iterations allowed.

Output: A,B is the range of variation.

The method used here is systematic cut and try.

2) <u>MTEST (X)</u>

Let x represent ω_n or ζ :

If KFLAG=1 (in the common block), then $\zeta=1$, and we test the transfer function

$$T(s) = \frac{1}{\frac{s^2}{x^2} + 2\frac{s}{x} + 1} \quad \text{if } \mathcal{L}^{-1}\left(\frac{T(s)}{s}\right) \in \text{T.D.B.}$$

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

If KFLAG=2 then we test if the step response of $T(s) = \dfrac{1}{\dfrac{s^2}{(OMEGA)^2} + 2x\dfrac{s}{OMEGA} + 1}$ lies inside the time domain bounds.

3) KTEST (NQ,NP,Q,P,GAIN,WO)

We test if GAIN*Q(S/WO)/P(S/WO) lies in T.D.B. Q is a polynom of degree NQ . P is a polynom of degree NP . The outputs are shown in Fig. 1.1.

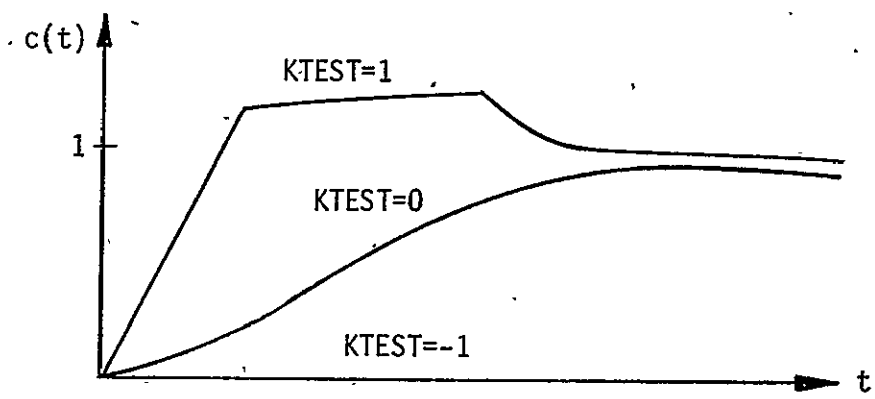


Fig. 1.1

C. STEP and DSTEP

Those routines compute the step response of G*Q(S)/P(S) .

1) STEP (M,N,Q,P,G,DELTA)

Q is a polynom of degree M . P is a polynom of degree N

Let the dynamical system be given by:

$$\dot{x} = A_p x + bu \qquad \text{where:} \qquad b = \begin{bmatrix} 0 \\ G \end{bmatrix}$$

$$A_p = \left[\begin{array}{c|c} 0 & 1 \\ \hline -p_n & \cdots p_1 \end{array}\right]$$

$$y = cx \qquad c = [q_m, q_{m-1}, \cdots, q_1, 1, 0]$$

and let u be a step function. Then $y(s) = \frac{1}{S} C(SI-A_p)^{-1}b$. $H(S) = C(SI-A_p)^{-1}b$ is the Laplace transform of the impulse response.

If δ=DELTA , is the increment in time, let $A=e^{\delta A_p}$. The step response is then:

$$y(t) = \int_0^t Ce^{A_p(t-\tau)} b d\tau$$

$$y(t) = -CA_p^{-1}b + Ce^{A_p t} A_p^{-1} b$$

$$= H(0) + Ce^{A_p t} A_p^{-1} b$$

but $$A_p^{-1}b = \begin{bmatrix} -g/p_n \\ 0 \end{bmatrix}$$ so finally:

$$y(k\delta) = H(0) + (-g/p_n)(CA^k)_1 \tag{1}$$

where the subscript $_1$ refers to the first element of that vector.

The following notation is used:

A <= EXP(DELTA*A_p)

B <= $-g/p_n$

PN <= P(N)

F <= H(0) = q_m*g/p_n

$$C(\cdot) <= [q_m, q_{m-1}, \cdots, q_1, 1, 0]$$

$$AA <= A_p$$

STEP is the value at $t=0$ of the step response.

2) DSTEP (N)

N is the degree of $P(\cdot)$. To calculate the step response at the time $k \times \delta$, we must call k times the function DSTEP , which computes (1).

D. Description of the Package of Subroutines for Stable, Minimum Phase Rational Approximation to a Magnitude Squared Function (RFA)

1) Philosophy

The problem is to construct a rational function

$$H(s) = g \frac{q(s)}{p(s)}$$

whose poles and zeros are in the LHP , such that

$$|H(j\omega)|^2 \approx F_0(\omega)$$

where $F_0(\omega)$ is a given magnitude squared function with a known excess of poles over zeros.

2) Theoretical Preliminaries

The heart of the approximation procedure is based on the following facts:

Let $F(\omega)$ be a positive, even function on $(-\pi,\pi)$ (a "power spectrum") and let its Fourier coefficients be denoted by

$$r_k = \frac{1}{2\pi}\int_{-\pi}^{\pi}\cos(k\omega)F(\omega)d\omega$$

[these form the autocorrelation sequence for a stationary second order random sequence with power spectrum $F(\omega)$]. Let $R(n)$ be the $(n+1)\times(n+1)$ Toeplitz matrix

$$R(n) = \begin{bmatrix} r_0 & r_1 & & r_n \\ r_1 & & & \\ & & & r_1 \\ r_n & & r_1 & r_0 \end{bmatrix}$$

Then $R(n)$ is positive definite for each n. Define $[\alpha_n, p_1(n), \cdots, p_n(n)]$ to be the solution

$$R(n)\begin{bmatrix} p_n(n) \\ p_{n-1}(n) \\ \vdots \\ p_1(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \alpha_n \end{bmatrix}$$

Finally, define

$$P_n(z) = z^n + p_1(n)z^{n-1} + \cdots + p_n(n)$$

$$F_n(\omega) = \frac{\alpha^n}{|P_n(e^{j\omega})|^2} .$$

Then the following are true:

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

(i) $$\alpha_n = \frac{\det R(n)}{\det R(n-1)} > 0$$

(ii) The zeros of $P_n(z)$ are in the open unit circle, i.e.

$$P_n(\lambda) = 0 \Rightarrow |\lambda| < 1$$

(iii) $$\frac{1}{2\pi}\int_{-\pi}^{\pi}\cos(\omega k)F_n(\omega) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\cos(\omega k)F(\omega)d\omega = r_k \quad \text{for} \quad k=0,1,\cdots,n\ .$$

(iv) $$\frac{1}{2\pi}\int_{-\pi}^{\pi}P_i(e^{j\omega})P_j(e^{-j\omega})F(\omega)d\omega = \begin{cases} 0\ , & \text{if } i\neq j \\ \alpha_i\ , & \text{if } i=j \end{cases} \quad \text{(Orthogonal polynomials)}$$

(v) An equivalent version of the above is the following. Let $T(n)$ be the $(n+1)\times(n+1)$ upper triangular matrix whose diagonal elements are ones, defined by

$$T(n)_{ij} = \begin{cases} 0\ , & \text{for } i>j \\ p_i(j-i)\ , & \text{otherwise} \end{cases}$$

Then

$$T^T(n)R(n)T(n) = \mathrm{diag}(\alpha_0,\cdots,\alpha_n)$$

This is a "triangular decomposition" of $R(n)$.

(vi) α_n is the minimum value of $\psi^T R(n)\psi$, taken over all vectors ψ of the form

$$\psi = \begin{bmatrix} x \\ x \\ x \\ \vdots \\ x \\ 1 \end{bmatrix}$$

(vii) The sequence $\alpha_0, \alpha_1, \cdots$ is nonincreasing and has the limit

$$\lim_{n\to\infty} \alpha_n = \exp[\frac{1}{2\pi} \int_{-\pi}^{\pi} \ln F(\omega) d\omega]$$

(This is the geometric mean of the power spectrum F).

(viii) The solutions to equation (1) may be computed recursively in n . This is

<u>Levinson's algorithm</u>:

Initialization: $\alpha_0 = r_0$

Recursion:

$$\begin{cases} \beta_n = \sum_{k=0}^{n} p_k(n) r_{n+1-k} \\ p_k(n+1) = p_k(n) - (\frac{\beta_n}{\alpha_n}) p_{n+1-k}(n) \\ \alpha_{n+1} = \alpha_n [1-(\frac{\beta_n}{\alpha_n})^2] \end{cases}$$

In the above $p_0(n)=1$ for all n .

<u>Remarks</u>: Levinson's algorithm dates from the paper: N. Levinson, "The Wiener rms error criterion in filter design and prediction", <u>J. Math. Phys.</u>, Vol. 25, pp. 261-178, Jan. 1947.

It is reprinted in the book <u>Extrapolation, Interpolation, and Smoothing of Stationary Time Series</u> by N. Wiener, as an appendix.

The result (vii) is due to Szegö, and may be found in the book <u>Toeplitz Forms and their Applications</u> by Grenander and Szegö.

For more references, proofs, and discussions see "The Use of Second-Order Intomation In Recursive Filter Approximation", by T.Mullis and R. Roberts, (Dept. of Electronics and Engineering, University of Colorado, Boulder).

3) Rational Approximation on Unit Circle

This is the discrete-time equivalent of the approximation problem. In other words, given a magnitude squared function $F(\omega)$, $-\pi<\omega<\pi$, find a rational function

$$H(z) = g\,\frac{q(z)}{p(z)}$$

where

$$\begin{cases} q(z) = z^m + q_1 z^{m-1} + \cdots + q_m \\ p(z) = z^n + p_1 z^{n-1} + \cdots + p_n \\ m \leqslant n \text{ , the zeros of both } p \text{ and } q \text{ are in the open unit circle,} \end{cases}$$

such that

$$|H(e^{j\omega})|^2 \approx F(\omega) \ .$$

The all-pole specialization is $m=0$, so that $q(z)=1$. In this case a useful approximation is provided by Levinson's problem. If one sets

$$g = \sqrt{\alpha_n}$$

$$p(z) = P_n(z)$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Update Denominator

Given $F(\omega)$ and $q(z)$, find an all-pole model

$$\frac{\alpha}{|p(e^{j\omega})|^2} \approx \frac{F(\omega)}{|q(e^{j\omega})|^2} \quad ; \quad \text{set} \quad g = \sqrt{\alpha}$$

Update Numerator

Given $F(\omega)$ and $p(z)$, find an all-pole model

$$\frac{\alpha}{|q(e^{j\omega})|^2} \approx \frac{1}{F(\omega)|p(e^{j\omega})|^2} \quad ; \quad \text{set} \quad g = \frac{1}{\sqrt{\alpha}}$$

These two iteration modes are repeated until some degree of accuracy is obtained. Note that one can increase the degrees m , n of the polynomials between iterations. It is also possible to consider approximations of the form

$$g\,\frac{q_1(z)\cdots q_r(z)}{p_1(z)\cdots p_e(z)}$$

then the resulting approximation $|H(e^{j\omega})|^2$ is useful in that Fourier coefficients are matched, i.e.

$$\frac{1}{2\pi}\int_{-\pi}^{\pi}|H(e^{j\omega})|^2\cos(k\omega)d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi}F(\omega)\cos(k\omega)d\omega$$

for $k=0,1,\cdots,n$. Furthermore, the poles of $H(z)$ are indeed in the open unit disc.

Since low order all-pole approximations are not very efficient, it is advisable to consider the case $m>0$, but how can this be done? In RFA an iterative procedure is used. This procedure has a number of modifications, but basically it goes as follows:

Initialize by setting $q(z)=p(z)=1$, and update only one (factor) polynomial at a time, if it is necessary to have a suitably factored approximation.

Each iteration requires the computation of a power spectrum, and a number of numerical integrations to obtain autocorrelation coefficients before Levinson's algorithm can be applied. It is here that most of the computation time is spent. For the computation of the power spectrums, it is necessary to compute $|p(e^{j\omega})|^2$ or $|q(e^{j\omega})|^2$. For this purpose the routine SPE ("Special Polynomial Evaluation") was written for the numerical integration the routines APA and JP were written.

<u>Description of SPE (P,N,CN,SN,X,Y)</u>

Here the inputs are

$$P = (p_1, p_2, \cdots, p_n)$$

$$CN = \cos\omega$$

$$SN = \sin\omega$$

and the outputs are x , y , where

$$x + iy = p(e^{j\omega}) \quad , \text{ where } \quad p(z) = z^n + p_1 z^{n-1} + \cdots$$

The algorithm is fast in that it requires only about n multiplications and $2n$ additions. It is based on the following:

If
$$A = \begin{bmatrix} 0 & 1 \\ -1 & 2\cos\theta \end{bmatrix} , \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix} , \quad C = \begin{bmatrix} 0 & \sin\theta \\ -1 & \cos\theta \end{bmatrix}$$

then for $k>0$

$$CA^{k-1}b = \begin{bmatrix} \sin k\theta \\ \cos k\theta \end{bmatrix}$$

For the system

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

$$\psi(k+1) = A\psi(k) + bp_{k+1}$$

Have

$$\psi(n) = A^n \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \sum_{k=1}^{n} A^{k-1} bp_{n+1-k}$$

and

$$C\psi(n) = \begin{bmatrix} \sin n\theta \\ \cos n\theta \end{bmatrix} + p_1 \begin{bmatrix} \sin[(n-1)\theta] \\ \cos[(n-1)\theta] \end{bmatrix} + \cdots + p_n \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \mathrm{Im}\ p(e^{j\theta}) \\ \mathrm{Re}\ p(e^{j\theta}) \end{bmatrix}$$

4) <u>Rational Approximation on $j\omega$ Axis</u>

The technique is to transform the problem to approximation on the unit circle via the bilinear transform

$$z = \frac{1+s}{1-s} \quad , \quad s = \left(\frac{z-1}{z+1}\right)$$

Define the map $\omega \leftrightarrow \theta$ by $s=j\omega$, $z=e^{j\theta}$ so that

$$\omega = \frac{\sin\theta}{1+\cos\theta} = \tan(\frac{\theta}{2}) \tag{3a}$$

$$\theta = 2\tan^{-1}(\frac{\omega}{n}) \tag{3b}$$

The numerical integrations on $(0,\pi)$ in APA assume a uniform step size, so that θ will take on the values $\{0,\frac{\pi}{N_0},\frac{2\pi}{N_0},\cdots\}$. This generates corresponding values of ω . Relation (3a) is thus used in TABCS to define the set of frequencies. The center frequency (i.e. for $\theta=\frac{\pi}{2}$) will be $\omega=1$. RFA is set up so that N_0 is a power of 2, up to $2^7=128$. The indexing is such that

$$\omega(1) = 0$$

$$\omega(N_0/2+1) = 1$$

$$\omega(N_0+1) = \infty$$

<u>Note</u>: We can multiply all frequencies by the constant factor ω_0 defined above, such that $\omega(N_0/2+1)=\omega_0$, which explain the name given to the central frequency.

The essential dynamic range, in deca , is

$$D = \log_{10}\left[\frac{\omega(N_0)}{\omega(2)}\right] = 2\log_{10}\left[\frac{1+\cos\frac{\pi}{N_0}}{\sin\frac{\pi}{N_0}}\right]$$

N_0	16	32	64	128	256
D	2.02	2.62	3.22	3.82	4.42

The log frequencies are distributed symmetrically about the center frequency, with the density greatest at the center. In other words

$$\log[\omega(k_0+j)] = -\log[\omega(k_0-j)]$$

where $k_0 = \frac{N_0}{2} + 1$.

5) <u>Transformation of Rational Functions According to Bilinear Map - Emphasis of Deemphasis</u>

Let $F_0(\omega)$ be a magnitude squared function on (o,∞) . This generalies the magnitude squared function

$$F(\theta) = F_0(\tan\frac{\theta}{2}) = F_0(\omega)$$

on $(0,\pi)$. If F_0 is the magnitude squared of a rational function with an excess of e poles over zeros, then $F(\theta)$ will have a zero of order $2e$ at π . It is not wise to approximate $F(\theta)$ in this form, since its inverse will have a pole of order $2e$ at π , and the numerator update will be ill-behaved. Hence $F(\theta)$ is multiplied by an "emphasis", so that the result will have no poles or zeros on the interval $(0,\pi)$. A natural choice of emphasis function is suggested by the bilinear transformation. Given a rational approximation for $(0,\pi)$, we must be able to compute the corresponding rational approximation

for $(0,\infty)$. This is the reason for the following operator B , whose inverse is implemented in the program by the function subprograms BT and NORMLZ. A consideration of this operation will lead to a convenient emphasis deemphasis.

For any monic (leading coefficient units) polynomial $\pi(t)$ of degree k , define the monic polynomial

$$(B\pi)(z) = \pi\left(\frac{z-1}{z+1}\right)(z+1)^k$$

$$(B^{-1}\pi)(s) = \pi\left(\frac{1+s}{1-s}\right)(1-s)^k$$

Suppose that $F_0(\omega)=|H(j\omega)|^2$, where

$$H(s) = g\,\frac{q(s)}{p(s)}$$

and q is monic of degree m , zeros in open RHP

p is monic of degree n , zeros in open RHP

$n - m = e$

Then

$$H(s) = g\,\frac{q\left(\frac{z-1}{z+1}\right)(z+1)^m}{p\left(\frac{z-1}{z+1}\right)(z+1)^n}\,(z+1)^e$$

$$= (z+1)^e\,\frac{(B_q)(z)}{(B_p)(z)}$$

For $z=e^{j\theta}$, $|1+z|^2=2(1+\cos\theta)$. Therefore

$$\left|\frac{(B_q)(e^{j\theta})}{(B_p)(e^{j\theta})}\right|^2 = \left(\frac{.5}{1+\cos\theta}\right)^e F_0(\omega)$$

Hence, we may as well use

$$\left(\frac{.5}{1+\cos\theta}\right)^e$$

for emphasis (see TABLE) and approximate

$$\tilde{F}(\theta) = \left(\frac{.5}{1+\cos\theta}\right)^e F_0\left(\tan\frac{\theta}{2}\right)$$

on $(0,\pi)$, note that $\tilde{F}(\theta)$ has no poles or zeros on this interval.

Suppose this is done, i.e. we have found an approximation

$$\tilde{H}(z) = \tilde{g}\,\frac{\tilde{q}(z)}{\tilde{p}(z)},$$

where $\tilde{q}$ is monic of degree m , with zeros in open unit circle and $\tilde{p}$ is monic of degree n , with zeros in open unit circle, for which

$$|\tilde{H}(e^{j\theta})|^2 \approx \tilde{F}(\theta) = \frac{F_0(\omega)}{[2(1+\cos\theta)]^e}$$

How do we recover the approximation to $F_0(\omega)$ which is stable and minimum phase? From the preceding approximation, we have

$$F_0(\omega) = |H(j\omega)|^2 \approx (|1+e^{j\theta}|^2)^e\,\tilde{g}^2 \left|\frac{\tilde{q}(e^{j\theta})}{\tilde{p}(e^{j\theta})}\right|^2$$

Then

$$H(s) \approx (1+z)^{n-m}\,\tilde{g}\,\frac{\tilde{q}(z)}{\tilde{p}(z)}\Bigg|_{z=\frac{1+s}{1=s}}$$

$$= \left(\frac{2}{1-s}\right)^{n-m}\tilde{g}\,\frac{\tilde{q}\left(\frac{1+s}{1-s}\right)}{\tilde{p}\left(\frac{1+s}{1-s}\right)} = [\,2^{n-m}\,\tilde{g}\,]\,\frac{(B^{-1}\tilde{q})(s)}{(B^{-1}\tilde{p})(s)}$$

We may therefore take

$$g = |\ (2)^{n-m}\ \tilde{g}\ |$$

$$q(s) = (B^{-1}\tilde{q})(s)$$

$$p(s) = (B^{-1}\tilde{p})(s)$$

Numerically, what is required in order to transform the $j\omega$ axis problem to the unit circle problem is,

(i) Emphasis, i.e. replace $F_0(\omega)$ with

$$\tilde{F}(\theta_i) = \frac{F_0(\omega_i)}{[2(1+\cos\theta_i)]^{n-m}}$$

(ii) After finding the approximation on $(0,\pi)$, recover g, q, p as above. In order to do this, the function subprogram BT was written.

6) Implementation

a) TABLE (NPQ)

For an excess e=NPQ we store a table in TAB of the values of

$$\left(\frac{.5}{1+\cos\omega}\right)^e$$

b) NORMLZ (FD,F,NPQ)

This subroutine transforms $F(\omega)$ in an emphazised function $FD(\theta)$ with $\theta\in(0,\pi)$.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

c) TABCS

This routine computes a TABLE of Cosine and Sine. With step-size $\frac{\pi}{n}$ (n given) it computes cos(•) , sin(•) and the frequencies

$$W(\cdot) = \frac{SN(\cdot)}{1+CN(\cdot)} .$$

E. RFABND (TM,PNAP,PDAP,NNQ,NNP,GAMA)

Given the frequency response TM (input), a rational approximation GAMA*PNAP(•)/PDAP(•) is calculated, with PNAP of degree NNQ , and PDAP of degree NNP .

Note: RFABND begin to approximate with degrees (1/3) , then (2/4) ••• until a maximum error (EPS) of 1% in the approximation achieved or until a maximum degree of 12 is reached for the denominator PDAP .

F. EXTLWB (NEXCESS)

This routine tries to extend the Lower Bound of the overall transfer function using up to NEXCESS of poles over zeros.

This is done by selecting suitable poles, which added to the upper bound still guarantee a correct behaviour in the time domain.

Systematic cut and try is used here and so, starting from a first guess ISTART (index frequency) we converge toward the final value of ISTART , at which one or several poles can be placed.

W(•) being the frequency-vector (see TABCS) then POLE=W(ISTART) .

1) MUL (P,NP,POLE,PA,NPA)

This routine calculates PA(s)=P(s)*(s+POLE) where P is of degree NP and PA of degree NPA=NP+1 .

2) KEEP5 (BND,POLE,N)

This routine updates the F.D.B.

$$FDUM(\cdot) = BND(\cdot) * \left(\frac{POLE^2}{POLE^2+W(\cdot)^2}\right)^N$$

FDUM is then compared to BNMN (BouNd Min) and BNMX (BouNd Max) which contain the lower and upper limit of the F.D.B.

G. EXTHGB (NEXCESS)

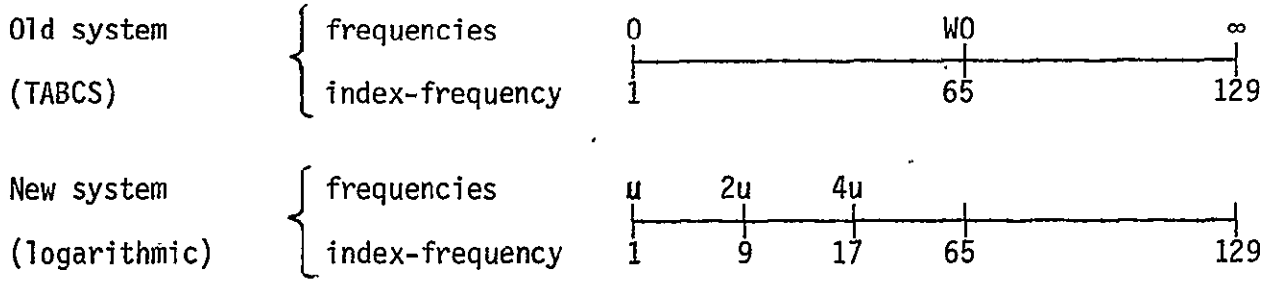
This routine EXTends the HiGher Bounds by placing appropriate zeros on the lowest bound TMIN and works in a similar way to EXTLWB .

H. FREQ

RFA requires a frequency scale, given by TABCS . However Part II uses a logarithmic scale for the frequencies. The routine FREQ does this transformation on the FREQuencies and makes the interpolation for the upper

and lower bounds of the F.D.B., BNMN and BNMX and for the ratio R=BNMX/BNMN .

Example: Suppose that NBR , the number of points/octave derived is 2 . Suppose k=2 , which means that the set of points considered in Part II, is $(1,1+2^2,1+2\times2^2,1+3\times2^2,\cdots)=(1,5,9,13,\cdots)$. In other words k=2 means that an interval of $2^k=4$ is chosen for the index frequency scale.



The logarithmic step is, ingeneral , $2^{\frac{1}{NBR*2^k}}$.

I. Other Subroutines Used

1) INIT

This is just an inialization of BNMN and BNMX , the lower and upper limits of the F.D.B. obtained with the second order model.

The vectors FP(•) and FQ(•) are also initialized to 1 in order to be able to call the routine RFA .

2) SPE2 (P,N,OMEGA,X,Y)

At frequency OMEGA , the polynom P of degree N has for real part X and imaginary part Y,

$$P(jOMEGA) = X + jY .$$

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

II. PART 2

Philosophy

The plant is read in (PLANT) and its range or templates (TMPLT) calculated over a set of frequencies. As an option, the boundary of the template can be plotted (P020) and the bounds on L can be computed and plotted (BNDG) .

A. PLANT Model

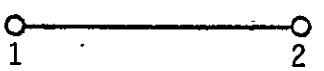


Fig. 2.0

Between the nodes 1-2 , there can exist (Fig. 2.0):

a) a known block of transfer function t(s) . This will be called a LINK ;

b) an unknown gain k . In this case it will be called a KINK .

The rules to describe the plant are the following:

(i) $1 \leq LINKS \leq 10$;

(ii) $1 \leq KINKS \leq 4$;

(iii) No feedback from final node;

(iiii) No feedback into first node;

(j) No trivial KINK , i.e., $1 \leq k \leq 1$;

(jj) All integrations should be included in path from 1 to 2 ; (The number of integrations is called NTYPE);

(jjj) The only branch out from 1 is to 2 .

1) Mathematical Preliminaries

Let us replace all integrations by a unit delay "1" .

Let IC(I) denote the length of the shortest path from node I to the output. Thus IC(•) is a vector of the shortest path to the output, from different nodes. It is seen, then, that the shortest path from input to output is the excess e of poles over zeros of the plant P(s) . Let NPO and NQO denote respectively the degrees of the denominator and numerator of P(s) , so that e=NPO-NQO .

Let IA be a matrix and IA(I,J) represent the length of node I to node J .

Example:

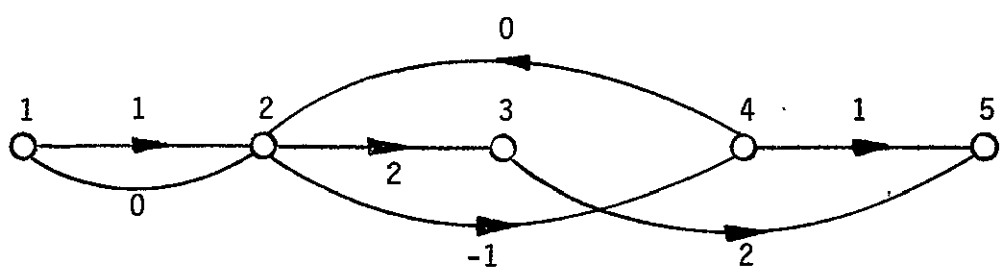


Fig. 2.1

The minimum path (Fig. 2.1) in 1 step only is represented by:

$$IA_1 = \begin{bmatrix} \infty & 1 & \infty & \infty \\ 0 & \infty & 2 & -1 \\ \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \end{bmatrix}$$

The minimum path from node I to the output in 1 step is represented by the vector:

$$IC = C_0 = \begin{bmatrix} \infty \\ \infty \\ 2 \\ 1 \end{bmatrix}$$

Define addition (+) by

$$IA_1(i,j) + IA_2(i,j) = \mathrm{Min}[IA_1(i,j), IA_2(i,j)]$$

and multiplication (*) by

$$IA_1(i,j) * IA_2(i,j) = IA_1(i,j) + IA_2(i,j)$$

Then it can be shown that the set of matrices IA associated with the two laws of additive (+) and multiplicative (*) is a field.

Thus, the minimum path in 2 steps is:

$$\begin{bmatrix} \infty & 1 & \infty & \infty \\ 0 & \infty & 2 & -1 \\ \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \end{bmatrix}$$

$$IA_2 = (IA_1)^2 = \begin{bmatrix} \infty & 1 & \infty & \infty \\ 0 & \infty & 2 & -1 \\ \infty & \infty & \infty & \infty \\ \infty & 0 & \infty & \infty \end{bmatrix} \begin{bmatrix} 1 & \infty & 3 & 0 \\ \infty & -1 & \infty & \infty \\ \infty & \infty & \infty & \infty \\ 0 & \infty & 2 & -1 \end{bmatrix}$$

It can be proven that the minimum path in 2 steps to the output is:

$$C_1 = C_0 + IA_1 * C_0$$

and in general, the minimum path in N steps or less to the output is:

$$C_t = C_{t-1} + IA_1 * C_{t-1}$$

or $$C_t(i) = \mathrm{Min}\{C_{t-1}(i), \min_j [Q(i,j) + C_{t-1}(j)]\}$$

It remains now to note that $e = C_t(1)$ when N is very large.

2) Implementation

∞ in the program is replaced by the number 777 . The preceding formula (same notation) is used in the program to calculate e .

The links are read in the matrix T , which is organized as follows:

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

link # 1	T(1)	T(LINKS+1)		T(126*LINKS+1)
link # 2	T(2)	T(LINKS+2)		T(126*LINKS+2)
⋮	⋮	⋮		⋮
link # LINKS	T(LINKS)	T(2*LINKS)		T(127*LINKS)
frequencies	W(2)	W(3)	----------	W(128)

T(i) is a complex number which represents the transfer function from node k to ℓ at the frequency W(n) .

Then the links are read in and AK and BK are vectors of extremal values for each KINK .

B. <u>RECQPG (NP,NQ,Q,P,G,X)</u>

For the plant condition X , this subroutine RECovers the polynoms Q and P of degree NQ and NP and the gain G , such that G*Q(S)/P(S) match exactly the data contained in the previous matrix set. This problem is known as the Cauchy-interpolation problem and is presented under "<u>Mathematical preliminaries</u>" below.

1) <u>Mathematical Preliminaries</u>

Suppose the data set $\{(\lambda_1,x_1),\cdots,(\lambda_N,x_N)\}$ is given, where the $\{\lambda_i\}$ are distinct.

<u>Define</u> $$\pi(s) = \prod_1^N (s-\lambda_i) = s^N + \pi_1 s^{N-1} + \cdots + \pi_N$$

Define

$$\eta_k = \begin{cases} 0 , & k=0 \\ \sum_{1}^{N} \frac{\lambda_i^{n-1}}{\pi'(\lambda_i)} , & k>0 \end{cases}$$

Then

$$\sum_{k=0}^{\infty} \eta_k s^{-k} = \frac{1}{\pi(s)}$$

and

$$\{\eta_k\} = \{0,0,\cdots,0,\underset{\substack{\uparrow \\ k=N}}{1},\Sigma\lambda_i,\cdots\}$$

Construction

$$c = [1 \quad \bigcirc]$$

$$A = \left[\begin{array}{c|c} 0 & I \\ \hline -\pi_N & -\pi_1 \end{array}\right] \qquad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Then

$$A = \Sigma\lambda_k E_k \quad , \quad cE_k b = \frac{1}{\pi'(\lambda_k)} \quad , \quad cA^k b = \sum_{i=1}^{N} \frac{\lambda_i^k}{\pi'(\lambda_i)}$$

$$\frac{1}{\pi(s)} = c(sI-A)^{-1}b = \frac{cb}{s} + \frac{cAb}{s^2} + \cdots$$

and

$$cA^{k-1}b = 0 \quad \text{for} \quad k < N .$$

Assume that the data satisfies

$$y_i p(\lambda_i) = q(\lambda_i) \quad \text{for} \quad i=1,\cdots,N$$

where

$$p(s) = s^n + a_1 s^{n-1} + \cdots + a_n$$

$$q(s) = q_0 s^m + \cdots + q_m .$$

Note: Distinguish between n and N .

<u>Define</u>

$$g_k = \begin{cases} 0 \ , & k=0 \\ \sum_{i=1}^{N} \dfrac{\lambda_i^{k-1} q(\lambda_i)}{\pi'(\lambda_i)} \ , & k>0 \end{cases}$$

Then

$$\sum_{k=0}^{\infty} g_k s^{-k} = \frac{q(s)}{\pi(s)}$$

and

$$\{g_k\} = \{0,\cdots,0,q_0,\underset{\substack{\uparrow \\ k=N-m}}{q_1}+q_0\Sigma\lambda_i,\cdots\}$$

<u>Define</u>

$$h_k = \sum_{i=1}^{N} \frac{\lambda_i^{k-1} y_i}{\pi'(\lambda_i)}$$

Since $y_i p(\lambda_i)=q(\lambda_i)$ for each i , assuming that $p(\lambda_i)\neq 0$ for each i , we get

$$h_k = \sum_{i=1}^{N} \frac{\lambda_i^{k-1}}{\pi'(\lambda_i)} \left[\frac{q(\lambda_i)}{p(\lambda_i)}\right]$$

Let

$$f(\lambda) = \frac{q(\lambda)}{P(\lambda)} \frac{1}{s-\lambda} = \frac{q(\lambda)}{p(\lambda)} \Sigma \frac{\lambda^{j-1}}{s^j}$$

$$cf(A)b = \Sigma \frac{f(\lambda_i)}{\pi'(\lambda_i)} = \Sigma h_k s^{-k}$$

$$\Sigma h_k s^{-k} = cq(A)[p(A)(sI-A)]^{-1}b$$

Using

$$\sum_{\ell=0}^{n} \lambda_i^{k+\ell-1} a_{n-\ell} = \lambda_i^{k-1} p(\lambda_i)$$

$$\sum_{\ell=0}^{n} h_{k+\ell} a_{n-\ell} = \sum_{i=1}^{N} \frac{\lambda_i^{k-1} q(\lambda_i)}{\pi'(\lambda_i)} = g_k$$

This implies the following

$$\begin{bmatrix} h_1 & h_{n+1} \\ h_2 & h_{n+2} \\ h_3 & h_{n+3} \\ & \\ & \end{bmatrix} \begin{bmatrix} a_n \\ \\ a_1 \\ 1 \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ \\ \\ \end{bmatrix} \tag{3}$$

Since $g_k=0$ for $k<N-m$, if $N-m\geq n$, i.e. $N\geq n+m$, then $p(s)$ is determined from $\{h_1,\cdots,h_{2n}\}$. These numbers depend <u>only</u> on the data, and <u>not</u> on the $\{g_k\}$. Once $p(s)$ is determined, the above equation can generate the sequence $\{g_k\}$ which can be used to generate $q(s)$ in view of the identity $\Sigma g_k s^{-k}=\frac{q(s)}{\pi(s)}$. This is done as follows:

$$\frac{q(s)}{\pi(s)} = \frac{q_0 s^{-(N-m)} + q_1 s^{-(N-m+1)} + \cdots + q_m s^{-N}}{1 + \pi_1 s^{-1} + \cdots + \pi_n s^{-N}}$$

therefore

$$\{0,0,\cdots,0,\underset{\substack{\uparrow \\ \text{index}=N-m}}{q_0},q_1,\cdots,q_m,0,0,0,\cdots\}$$

$$= \{0,0,\cdots,0,g_{N-m},g_{N-m+1},\cdots\} * \{1,\pi_1,\cdots,\pi_n,0,0,0,\cdots\}$$

<u>i.e.</u>

$$q_0 = g_{N-m}$$

$$q_1 = q_{N-m+1} + \pi_1 g_{N-m}$$

$$q_2 = q_{N-m+2} + \pi_1 q_{N-m+1} + \pi_2 g_{N-m}$$

etc.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

<u>Case 1</u> $N \geq 2n'$

$$\{\lambda_i\} = \{(j\omega_1),\cdots,(j\omega_{n'}),(-j\omega_1),\cdots,(-j\omega_{n'})\}$$

$$\{y_i\} = \{(a_1+jb_1),\cdots,(a_{n'}+jb_{n'}),(a_1-jb_1),\cdots,(a_{n'}-jb_{n'})\}$$

$$h_k = \sum_{i=1}^{n'} 2\mathrm{Re}\left[\frac{(a_i+jb_i)(j\omega_i)^{k-1}}{\pi'(j\omega_i)}\right] \quad \text{for} \quad k>0$$

$$\pi(s) = \prod_1^{n'} (s^2+\omega_i^2)$$

$$\pi'(j\omega_\ell) = 2j\omega_\ell \overset{\circ}{\Pi}(\omega_i^2-\omega_\ell^2) \qquad [\overset{\circ}{\Pi} \text{ is a deleted p} \quad]$$

define $$Q_\ell = \frac{2j}{\pi'(j\omega_\ell)} = \frac{1}{\omega_\ell \overset{\circ}{\Pi}(\omega_i^2-\omega_\ell^2)}$$

then $$h_k = \sum_{i=1}^{n'} \mathrm{Re}[Q_\ell(b_\ell-ja_\ell)(j\omega_\ell)^{k-1}]$$

$k = 0 \qquad Q_\ell b_\ell$

$k = 1 \qquad Q_\ell a_\ell \omega_\ell$

$k = 2 \qquad -Q_\ell b_\ell \omega_\ell^2$

$k = 3 \qquad -Q_\ell a_\ell \omega_\ell^3$

$\vdots$

<u>Case 2</u> Same as case one except $\tilde{N}=2n'+1$, add 0 to $\{\lambda_i\}$ $y_0=a_0$

$$\tilde{\pi}(s) = s\prod_1^{n'} (s^2+\omega_i^2) = s\pi(s)$$

$$\tilde{\pi}'(0) = \left(\prod_1^{n'} \omega_i\right)^2$$

$$\tilde{\pi}'(j\omega_\ell) = -2\omega_\ell^2 \overset{\circ}{\Pi}(\omega_i^2-\omega_\ell)^2 = j\omega_\ell \pi'(j\omega_\ell)$$

$$\tilde{h}_k = \frac{a_0}{(\prod_1^{n'} \omega_i)^2} \delta_{k,0} + \sum_{\ell=1}^{n'} \mathrm{Re}[(\frac{a_\ell + jb_\ell}{-\omega_\ell}) Q_\ell (j\omega_\ell)^{k-1}]$$

$k = 0 \qquad \frac{-a_\ell Q_\ell}{\omega_\ell}$

$k = 1 \qquad b_\ell Q_\ell$

$k = 2 \qquad a_\ell Q_\ell \omega_\ell$

$k = 3 \qquad -b_\ell Q_\ell \omega_\ell^2$

$$\tilde{h}_k = \begin{cases} \frac{a_0}{(\prod \omega_i)^2} - \Sigma(\frac{a_\ell Q_\ell}{\omega_\ell}) & , \quad k=1 \\ h_{k-1} & , \quad k>1 \end{cases}$$

2) Implémentation

For the set of select frequencies $W(\cdot)$, we compute $P(j\omega)$. This is done through subroutine PJ .

a) PJ (KW,X,PJW)

Input : KW is index frequency

X is the plant condition

Output: PJW <= P[jW(KW)] = $P(j\omega)$

In the frequency domain (Fig. 2.2), we can write:

$$x = Gx + cu$$

$$y = bx$$

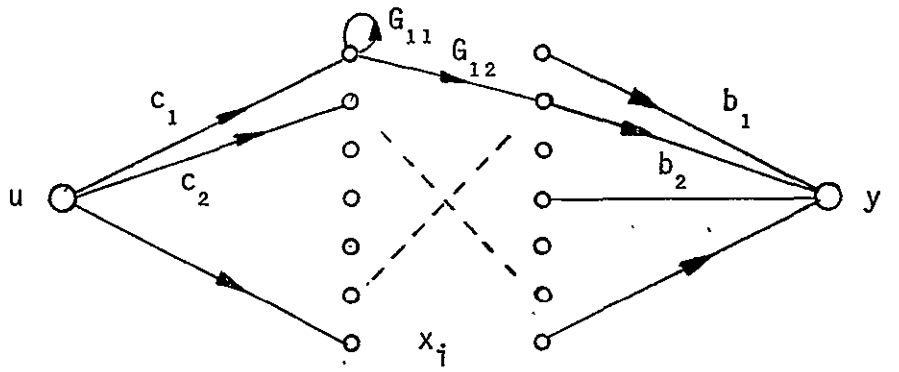


Fig. 2.2

Thus: $$y = c(I-G)^{-1}bu$$

Here $c=[1,0]$ so the transfer function we are looking for is:

$$PJW = [(I-G)^{-1}b]_1$$

where the index $_1$ refers to the first coefficient of the vector under parenthesis. Instead of inverting $(I-G)$, we employ "Crout reduction".

Essentially this leads to the building of the matrix $A<=[I-G|b]$ which is $n+1 \times n$ where $n+1$ is the number of nodes.

"Crout reduction" uses matrix A to compute

$$PJW = [(I-G)^{-1}b]_1 \quad \text{as} \quad [b+Gb+G^2b+\cdots]_1$$

b) <u>JAKRON (KFLAG,NN)</u>

This routine builds up the Hankel matrix

$$H = \begin{bmatrix} h_1 & h_2 & & h_{n+1} \\ h_2 & & & \\ \cdot & & & \\ \cdot & & & \\ \cdot & & & \\ h_{n+1} & & & h_{2n+1} \end{bmatrix}$$

used in equation (3).

c) HANKEL (N1,H,A)

This subroutine solves recursively equation (3) and leads to the computation of the coefficients of the polynom $P(s)$ of degree n .

The sequence $\{g_k\}$ then generates the numerator $g(s)$ whose degree $n-e=m$ is known, and the gain G .

Note: The main reason for using Cauchy interpolation problem is to guarantee the minimum form of the plant $P(s)$.

C. PATTERN (KW,KINKS)

KW : index-frequency

KINKS: number of independent parameters

This routine computes the contour of the template of P at frequency-index KW and stores the necessary points.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

1) Mathematical Preliminaries

Using the bilinear theorem, we know that if one parameter k varies over (k_1, k_2) the plant transfer function describes in the complex plane, an arc of a circle. To represent an arc of circle in the complex plane we use the mapping

$$R \xrightarrow{f} \mathbb{C}$$

$$f(x) \triangleq \frac{ax+b}{cx+1} \qquad \text{with} \qquad \begin{array}{l} x \in R \\ a,b,c \in \mathbb{C} \end{array}$$

If $x \in (0,1)$ we have an arc of circle. The relations between a, b, c and $f(0)$, $f(\frac{1}{2})$, $f(1)$ are given by:

$$f(0) = b$$

$$f(\tfrac{1}{2}) = \frac{\frac{a}{2} + b}{\frac{c}{2} + 1}$$

and

$$f(1) = \frac{a+b}{c+1}$$

The relations between a, b, c and $f(0)$, $f(1)$, $f'(0)$ are given by:

$$f(0) = b$$

$$f(1) = \frac{a+b}{c+1}$$

and

$$f'(0) = a - bc \qquad \left[f'(x) = \frac{a-bc}{(cx+1)^2}\right]$$

Thus we can show that:

$$f(x) = f(0) + \frac{f'(0)x}{cx+1}$$

with $$1 + c = \frac{f'(0)}{f(1)-f(0)}$$

Notation: FO <= f(0)

FH <= f(½) FPO <= f'(0)

F1 <= f(1)

2) TOUR (IN,M,NO)

M is the number of KINKS

NO=2^M is the number of vertices in the space of parameters.

IN is a vector which summarizes the path taken.

Example: M=3 => NO=8

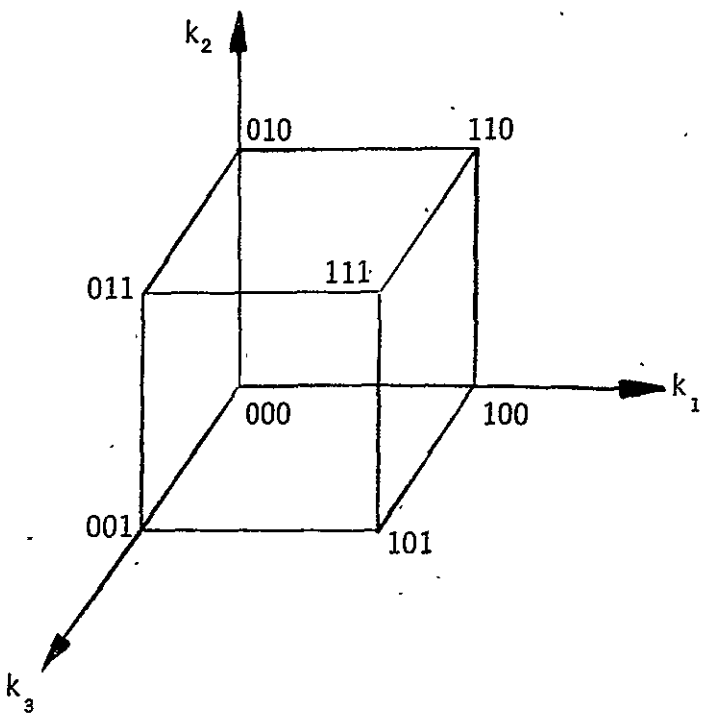
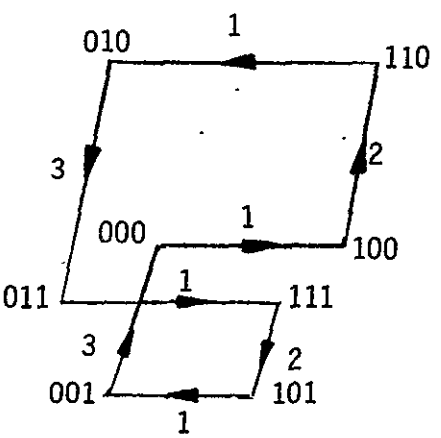


Fig. 2.3a



Path 1 is a tour

Fig. 2.3b

All the corners (Fig. 2.3a) are reached by taking the path 1 (Fig. 2.3b) which can be described by the vector IN={1,2,1,3,1,2,1,3} (called a "tour") which indicates only which of the 3 independent parameters is changed. Thus IN describes a "TOUR" .

Two other tours are possible by single circular permmtation.
In PATTERN the vector IDGE(I,J) reveals whether the edge from I to J has already been examined [IDGE(I,J)>0] , or not [IDGE(I,J)=0] . In each edge, trois points are examined and the parametric linearization is found in ARC .

3) <u>ARC (C,FPO,FO,FH,F1)</u>

Given FO , FH , F1 . This subroutine computes FPO and C , in order to describe the arc of circle $f(z)=f(0)+f'(0)\frac{z}{cz+1}$ with $z \in (0,1)$.

4) <u>IVARC (C,FPO,FO,D,T)</u>

Given C , FPO , FO and D=f(1)-f(0) . This subroutine computes the curvature of the circle and depending on it, a multiple of 2 (points) are added to describe the arc of circle.

IVARC can be 1, 2, 4, 8 .

The eventual added points are set in the vector $T \leftarrow (t_1,\cdots,t_8)$.

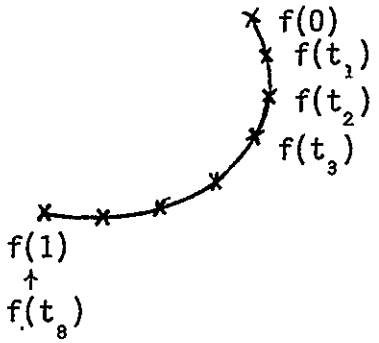


Fig. 2.4

5) <u>KHULL (Z,N,X,I1,I2)</u>

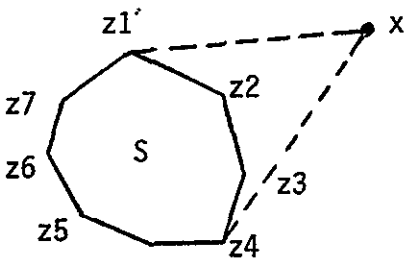


Fig. 2.5

Given some ordered vectors of a convex polygon S , Z(1)•••Z(N) all complex, check if X complex belong to the convex HULL S .

$$\text{KHULL} = \begin{cases} 0 & \text{if} \quad X \in S \\ 1 & \text{if} \quad X \notin S \end{cases}$$

If KHULL=1 , I1 , I2 are the indices corresponding to the end values of the points to be next. (Example: here I1=1 , I2=3).

6) IRT (X,Y,Z)

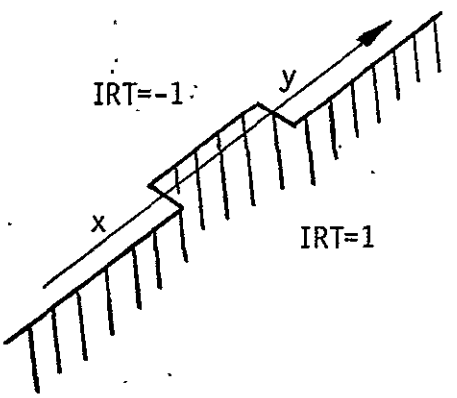


Fig. 2.6

Given X , Y , Z complex, the value of IRT is returned according to the drawing. IRT is used by KHULL to detect if a point is inside or outside a convex hull.

7) PO2O (Z)

This is a plotting routine for the templates in the Nyquist plane. PO21(N,NPT,Z) plots on the Nichols' chart.

8) PO1 (NQ,NP,Q,P,GAIN)

This is a printing routine for the rational function GAIN*Q(S)/P(S)

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

9) DOT (X,Y)

$$\vec{X} \cdot \vec{Y} = DOT(x,y)$$

D. TMPLT (KINKS)

This routine first eliminates all the arcs which are interior to the convex hull returned from the routine PATTERN .

Then the contour of the template is stored in parametrized form.

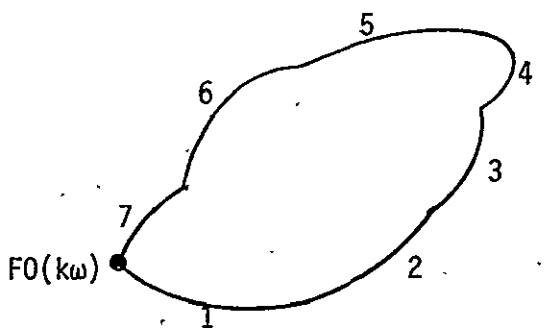


Fig. 2.7

FO(KW) is the starting point of the template at frequency KW . NPT is the number of arc of circles which describe the template at KW . LOC is an adress such that FPO(LOC) contain the value of f'(0) for the arc no.1 ; FPO(LOC+1) contains f'(0) for the arc no.2 ; FPO(LOC+NPT) contains f'(0) for the arc no.NPT .

LOC	NPT

NTMP(KW)

LOC and NPT are stored in arc register NTMP(KW) . They are obtained using the rule:

LOC = NTMP(KW)/128

NPT = NTMP(KW) - 128*LOC

In the parametrization we do not use c , but store h=1/(1+c) in H(•) .

Thus the first arc is described by:

FO(KW) , FPO(LOC) , H(LOC)

It's final point is $f_1(1)$ and will be the initial point of the second arc described by:

$f_1(1)$, FPO(LOC+1) , H(LOC+1)

and so on.

<u>Note</u>: Instead of storing the template of P , we store the template of 1/P .

III. PART 3

Philosophy

We take a first guess for $L_2(j\omega)$ (FRSTG) and do 4 iterations of "local changes". The "local" changes are done at each frequency and the correction then affects all frequencies. This allow rapid initial 'ball-park' convergence.

"Global" changes are then made to improve the solution. "Global" refers to the fact that the corrections are made simultaneously at every frequency.

1) Mathematical Preliminaries

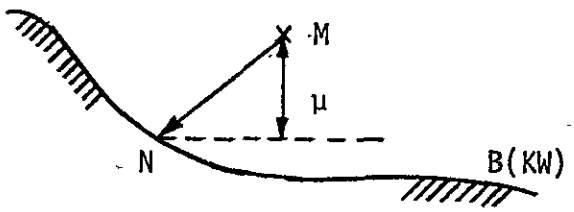


Fig. 3.1

Suppose that $B(KW)$ is the bound on $L(j\omega)$ at the frequency KW, and let M be the value being used at this point for $L(KW)$. The correction from point M to point N is made by means of a lead-lag network, of the form $t_{cor}(s) = \frac{b}{a}\frac{s+a}{s+b}$.

Suppose we want the effect of the network to be maximum at the frequency $W(KW)$. Then we can let $b=x \cdot W(KW)$ and $a=W(KW)/x$ and

$$|t_{cor}[jW(KW)]|^2 = x^2 = \mu$$

because
$$t_{cor}(s) = \frac{x[W(KW)+xs]}{[xW(KW)+s]}$$

We know that for minimum phase system , the phase and magnitude are related through the Bode-integrals. However, to use the Bode-integral we must evaluate singular integrals, difficult to do with precision.

We use instead a technique first proposed by Bode (Bode, p. 343) whereby the given characteristic is approximated by "semi-infinite segments". The phase corresponding to such a characteristic (break-point ω_0) is given by:

$$B(x_c) = \frac{2}{\pi}\left(x_c + \frac{x_c^3}{9} + \frac{x_c^5}{25} + \cdots\right) \quad \text{if} \quad 0 \leq x_c \leq .414$$

$$= \frac{\pi}{4} - \frac{1}{\pi}\log x_c \log\frac{1-x_c}{1+x_c} - \frac{2}{\pi}\frac{1-x_c}{1+x_c} \quad \text{if} \quad .414 \leq x_c < 1$$

where $x_c = \frac{\omega_c}{\omega_0}$.

Our procedure for global changes is as follows. Given $L_2^{(i)}(j\omega)$ obtain after i iterations, we compute the ℓ corrections corresponding to the ℓ different frequencies of interest. Thus we will end with a set of ℓ magnitudes, from which the corresponding ℓ phases will be obtained. The new set of ℓ magnitudes and ℓ phases give new loop transmission $L^{(i+1)}(j\omega)$, etc.

A. PLNMNL

This routine calculates the PLant NoMiNoL for $x=1$, which correspond to the minimum value of all KINKS . The nominal plant is then stored, in PLNL( ) for the magnitude squares, and in PLNFI( ) for the phase in radians.

B. FRSTG (GF,IST2,IST1,NNN)

The first guess of $L_{2\ nom}$ is takes as $\frac{k}{s(s+\alpha)}$. Thus $(L_{2\ nom}=GP_{nom})$, $G_{first\ guess} = \frac{k}{s(s+\alpha)} \frac{1}{P_{nom}(s)}$.

GNVAR denotes variation in the high-frequency gain factor of the plant. IST1 is an index-frequency which is chosen such that

$$R(IST1) = \left.\frac{T_{max}}{T_{min}}\right|_{I=IST1} > GNVAR \qquad \text{but } [R(IST1-1)<GNVAR]$$

Then $\alpha = W(IST1)$

IST2 is an index-frequency calculated in Part I, and which essentially gives the first index-frequency for which $L(j\omega)$ satisfies its bound.

k is then chosen as the minimum gain for which $L_{2\ nom}$ satisfies its bound on the interval (IST2,IST1) .

The procedure now consists of finding $L'_{2\ nom}$ which satisfies its bound on (IST2,IST1) then to increase IST1 until $L_{2\ nom}(IST1)$ is less than GNVAR (the high gain variation). The later will be the solution of our problem if we were satisfied with an excess of 2 poles over zeros.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

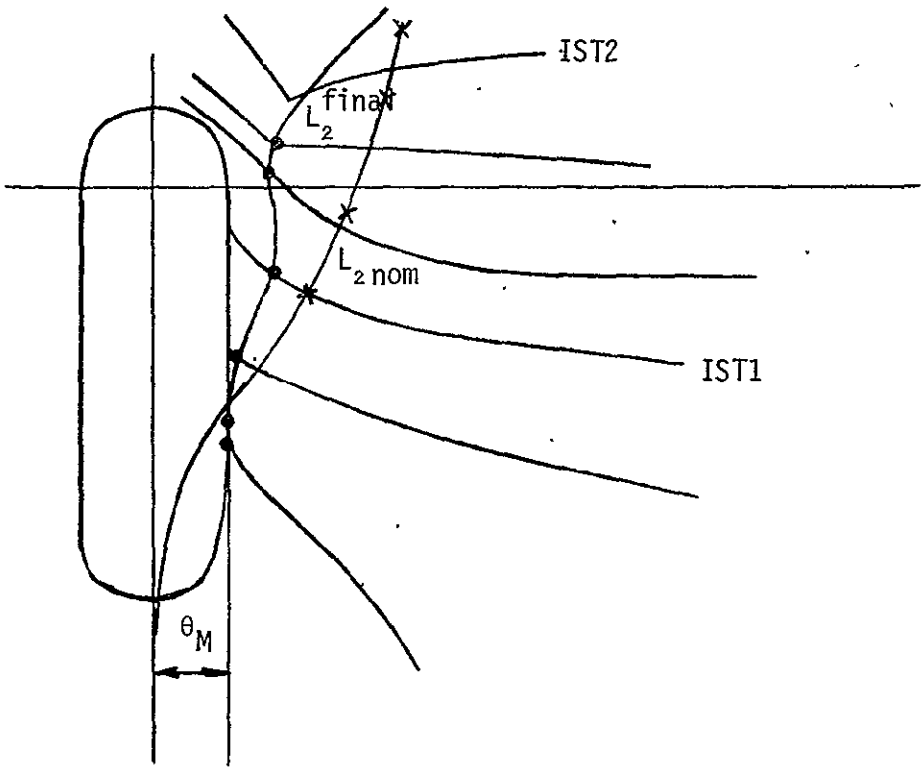


Fig. 3.2

C. BNDL (IST2,IST1,NNN)

This routine computes and plots the first nine bounds on L_{nom} .

D. XLDLAG1 (A,B,I,KL,NFCAG1,IST2,IST1)

At frequency-index $I \in$ (IST2,IST1) , this routine computes the LeaD-LAG $\frac{B}{A}\frac{s+A}{s+B}$ which brings $L_{2\ nom}$ to its boundary. This is done by cut and try. RNG1 (see below) computes T_{max}/T_{min} and accordingly x is increased or decreased.

A and B are then returned to the current program.

Note: $$R = \frac{T^2_{max}}{T^2_{min}} = \frac{|1+L|^2_{max}}{|1+L|^2_{min}} = \frac{|1+GP|^2_{max}}{|1+GP|^2_{min}} = \frac{|G + \frac{1}{P}|^2_{max}}{|G + \frac{1}{P}|^2_{min}}$$

Thus in XLDLAG1 we need to call RNG1 after translating the origin FO by G .

$$PMAX \Leftarrow |G + \frac{1}{P}|^2_{max}$$

$$PMIN \Leftarrow |G + \frac{1}{P}|^2_{min}$$

1) RNG1 (FO,FPO,H,N,ZMIN,ZMAX,PMIN,PMAX)

Given a boundary composed of N distinct arc of circles parametrized by FO,FPO(1),•••,FPO(N),H(1),•••,H(N) . This subroutine determines the points Z_{max} and Z_{min} , with the greatest and least distances from zero.

2) CHK1 (F,P,ZMIN,ZMAX,PMIN,PMAX)

$P=|F|^2$, F complex.

This subroutine does the following:

If P < PMIN then P => PMIN

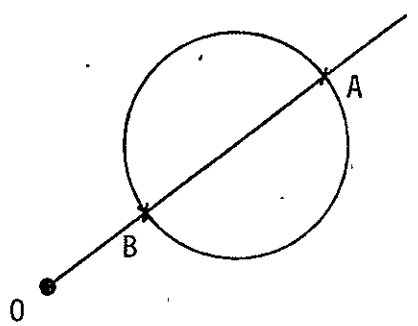
If P > PMAX then P => ZMIN

If PMIN < P < PMAX , nothing happens.

3) MIDPNT (F1,D,H,FPO,F)

If an arc of a circle is parametrized by F1 , D , H , FPO this subroutine returns the number of special points contained in the arc.

$$\text{MIDPNT} = \begin{cases} 0 , & \text{no special point on the arc} \\ 1 , & \text{one special point is contained} \\ 2 , & \text{two special points are contained} \end{cases}$$



A and B are special points

Fig. 3.3

4) KUAD (A,B,C,X1,X2,Y)

Given the polynom $p(s)=As^2+Bs+C$, this routine computes the roots of $p(s)=0$ and stores them in y . KUAD is then the number of roots which lie in the interval $(x1,x2)$.

5) ERROR (I,NFLAG,IST2,IST1)

This routine computes the normalized Euclidienne distance of the point $L_{nom}(j\omega)$ to its boundary at frequency I . Moreover, it stores the maximum

of all distances computed on (IST2,IST1) , with one db considered equivalent in 'distance' to 5 degrees.

NFLAG=7 , indicates that the disturbance bound has been violated.

Note: In XLDLAG1 (except when called for local changes), the value of x is limited to (.95,1.05) (5% of changes allowed at each step) in order to avoid oscillations in the iteration.

An additional refinement was found to be useful in the final stage of the iteration for eliminating oscillations. In Fig. 3.4 suppose the n-th iteration gives the points A , B for $L(j\omega_1)$, $L(j\omega_2)$ respectively. Say the correction corresponding to μ_1 , is made for the new set of magnitudes (for the n+1 trial). This smaller magnitude gives a larger rate of decrease of magnitude near ω_1 , driving $L(j\omega_1)$ in the direction of A' . No correction is persumably needed at ω_2 because B is on the boundary. However, since the magnitude of A' is now being used at ω_1 , if B is used at ω_2 then the average rate of magnitude decrease near ω_2 is now less than before, i.e. mag.(B-A')<mag.(B-A) , leading to phase lead at ω_2 , in the direction of B To prevent this we should lower also the magnitude at ω_2 , in the direction of B' . This refinement is incorporated at present only along the high-frequency disturbance boundary. For example at C,D,···,G in Fig. 3.4 the updates would normally be, say, .95, 1.0, .98, 1.02, 1.05 . Instead we use .95, .95, .93, .95, 1.0 .

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

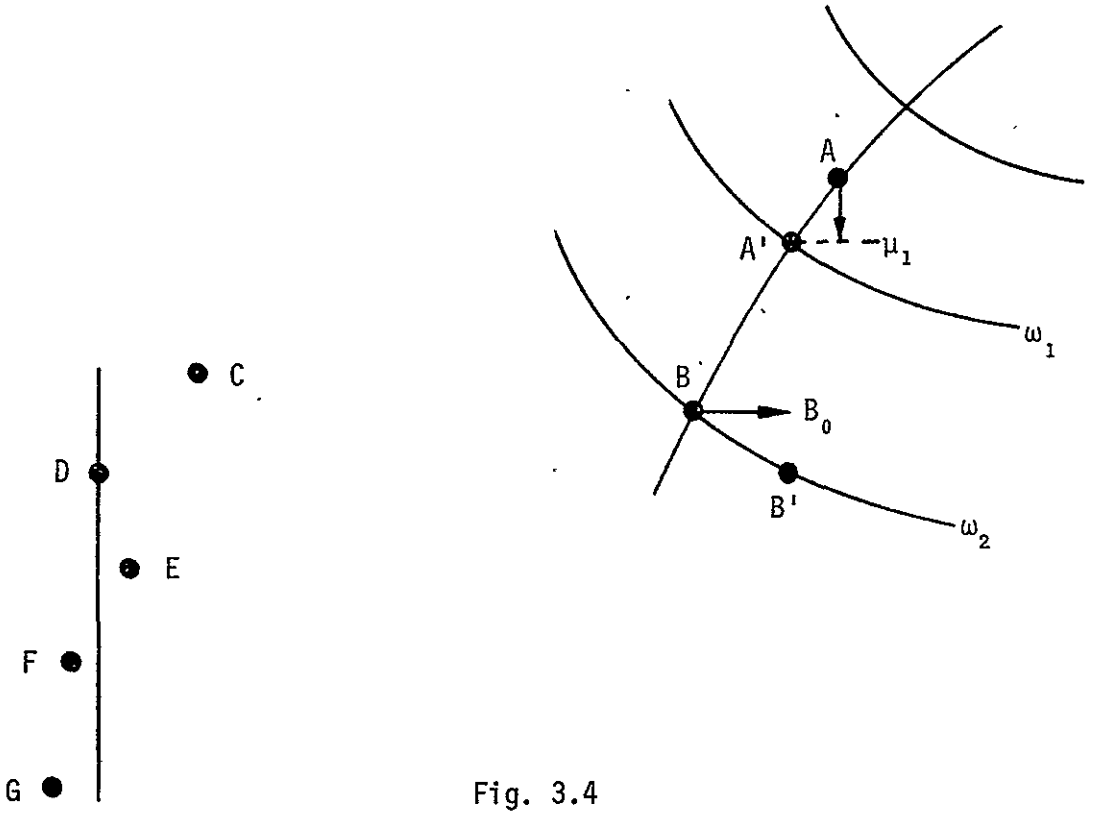


Fig. 3.4

E. KEEPG (A,B)

This routine is used with local changes. It updates the compensation G(s) at every frequency according to the rule

$$G_{new}(s) = G_{old}(s) * \frac{B}{A} * \frac{s+A}{s+B}$$

F. FIXG (A,B,L,IST,IST1,IST2)

This subroutine is used for "global" changes to update the set of magnitudes squared. At index-frequency $L \varepsilon$ (IST2,IST1) the vector F is filled with G^2x^2 . For the frequencies smaller than IST2 , $L_{2\ nom}(j\omega)$ is assumed to have a slope of 6dB/octave and for the frequencies greater than

IST1 , the last slope is extended indefinitely.

G. SLOP (I1,I2,NNN)

Given a set of magnitude square F(•) over (I1,I2) , SLOPE(•) contains the relative change of slope over the previous interval at each frequency.

H. PHASE (W,I1,I2,NNN)

Using the previous results stored in SLOPE(•) , this subroutine computes the phase at frequency W , using the technique previously described.

I. XGAIN (IST2,IST1,NNN)

This subroutine replaces G(•)<=XK*G(•) where XK is a pure gain, obtained such that as many points lie above the boundary below.

J. Other Subroutines

1) DSP (NPT,NY,X,Y,SYM,TITLE,KFLAG,Y1,Y2)

This routine is a plotting routine.

NPT : number of points

Y1,Y2: Min and Max of the ordinates y

NY : number of curves

X : vector of abcissal values

Y : two dimensional vector of curves $y_K=y(\cdot,K)$

TITLE: vector of alphanumeric characters

SYM : vector of symbols for up to 10 curves

KFLAG: 0 , if general
1 , if we want to plot $f(x,y)=0$

In the latter case of should be stored in DSPZ .

2) DSP2 (A,Y1,Y2,X1,X2)

Used when KFLAG=1 in DSP

3) IFOR

Used for scaling purpose.

4) FIXBET (I)

This subroutine calculates the maximum possible phase of $L_2(j\omega)$ over all frequencies and plant-conditions.

IV. PART 4

Philosophy

At the end of Part 3 we do not have as yet the required loop transmission $L(j\omega)$. We have for reasons of convenience a loop transmission (physically unrealiable) with has an excess of $(2-\frac{\theta_M}{90})$ poles over zeros, where θ_M refers to the phase margin of L (see Fig. 4.1).

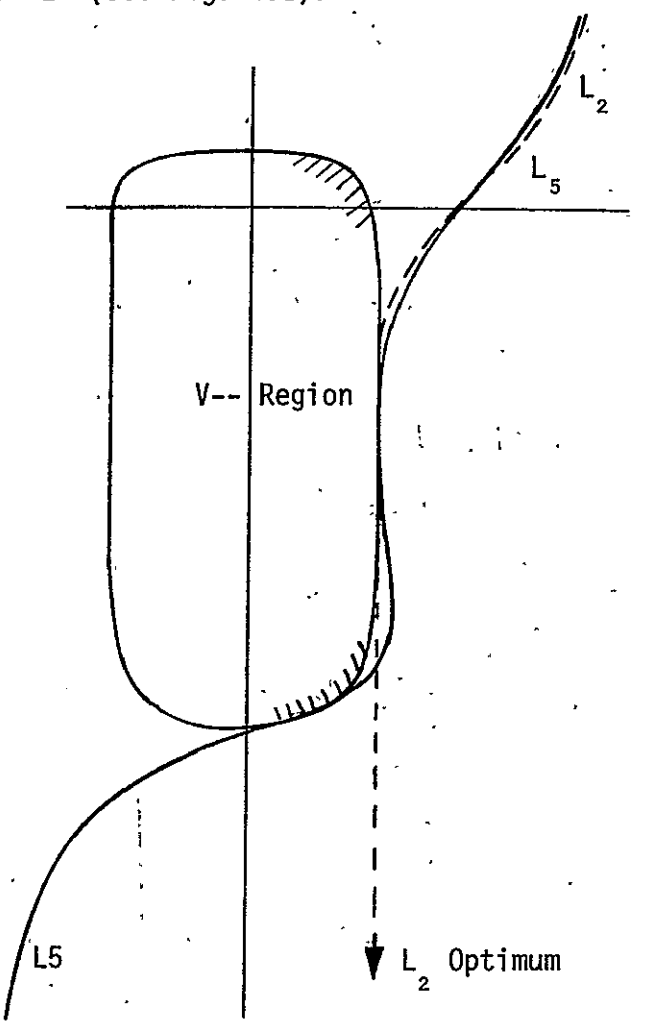


Fig. 4.1

We therefore multiply this by $H=\frac{K(s+z)^{\beta}}{(s^2+2\zeta\omega_n s+\omega_n^2)^2}$ with $\beta=1-\frac{\theta_M}{90}$, in order to obtain a realizable loop transmission with an excess of 5 poles over zeros. Obviously, we do not want to change the gain at low frequencies, $K=\frac{\omega_n^4}{z^{\beta}}$.

We let $\omega_n=\lambda z$, and search over the three parameters z, λ and ζ, given θ_M, in order to find an 'acceptable' characteristic. The search over λ and ζ has been experimentally and charts of suitable $\lambda(\theta_M)$ and $\zeta(\theta_M)$ were prepared and stored in the program (see Fig. 4.2). Therefore, it is necessary only to search over z, given θ_M, $\zeta(\theta_M)$ and $\lambda(\theta_M)$. This is done by cut and try.

When z is found, $L_5(j\omega)$ is available in the form of numerical data, over a set of frequencies. A rational approximation is then made of

$$L_0(s) = L_5(s) \cdot S \cdot \frac{(s^2+2z\omega_n s+\omega_n^2)^2}{\omega_n^2}$$

which is finite non-zero at $s=0$ and at $s=\infty$. The reason for this modification is to retain the double complex factor in the final rational representation of $L(s)$.

A. <u>CHKBND (IST,IST2,IST1)</u>

This routine uses the chart (Fig. 4.2), to compute ζ and λ. Then the value of z is adjusted by cut and try.

The final results are stored in block |BETA1| .

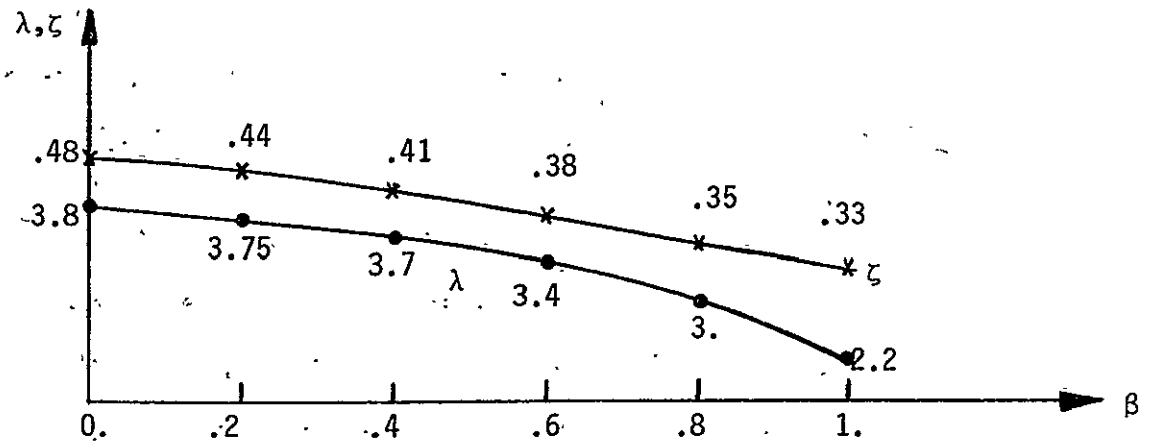


Fig. 4.2

Notations: BETA <= β XLAMDA <= λ

A <= z QSI <= ζ

B. FREQ1 (IST)

This subroutine changes our logarithmic scales for the frequencies into a trigonometric one as defined by TABCS . This is done in order to be able to take a rational approximation of L_s . On that new scale $|G|^2$ is interpolated and started in F(•) .

C. INIT1

This subroutine initializes FP and FQ , for use in RFA .

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

D. RFAFNL

This subroutine computes

$$L_0 = L_5(s) \cdot S \cdot \frac{(s^2+2z\omega_n s+\omega_n^2)}{\omega_n^2}$$

and approximates it by GAMA*PNAP(S)/PDAP(S) , where PNAP is of degree NNQ and PDAP of degree NNP . Then L_5 is approximated and G(s) calculated. $[G(s)=L_5(s)/P_{nom}(s)]$.

1) CNVLTN (A,NA,B,NB,C,KFLAG)

This subroutine computes the product of two polynoms A and B of degree NA and NB and puts the result in C .

E. PREFI (IST,IST1)

The data for the prefilter F is obtained from

$$F = \frac{\sqrt{T_{Min}T_{Max}}}{\sqrt{(\frac{L}{1+L})_{Min}(\frac{L}{1+L})_{Max}}} = \frac{\sqrt{T_{Min}T_{Max}}}{G}\sqrt{(G+\frac{1}{P})_{Min}}\sqrt{(G+\frac{1}{P})_{Max}}$$

Notations:

FM <= F is stored in the block PREF

BB <= TMIN * TMAX

PMIN <= $|G + \frac{1}{P}|_{Min}$

$$PMAX <= |G + \frac{1}{P}|_{Max}$$

$$GMAG <= |G|$$

This subroutine, thus, prepares the data of $|F|^2$. RFA is then used to obtain a rational approximation.

Note: PREFI prints out a summarizing table.

OMEGA	stands for	frequency
DB1	stands for	(T_{Max}/T_{Min}) tolerated in DB
DB	stands for	(T_{Max}/T_{Min}) real in DB
XLMIN	stands for	$(L/1+L)_{Min}$ in mag. square
XLMAX	stands for	$(L/1+L)_{Max}$ in mag. square
DIST	stands for	tolerated disturbance in mag. squares
PREF	stands for	value of the prefilter
BNMN(KW)	is the	actual min. bound of T at KW
TMN	is the	tolerated T_{Min}
TMX	is the	tolerated T_{Max}
BNMX(•)	is the	actual max. bound of T

F. FREQ2 (NEXPF)

This subroutine extends the data for the prefilter, assuming an excess of NEXPF poles over zeros for the prefilter.

G. RFAPF (IST,IST1)

This routine approximates the data contained in FN~GAMA*PNAP(S)/PDAP(S) , which is then stored back in the common block |FINPF| , to be available for printing and plotting of the prefilter response.