TECHNICAL REPORT

THRESHOLD ANALYSIS OF PHASE LOCKED LOOPS

for

National Aeronautics & Space Administration
Electronic Research Center

under

NASA Grant NGR 33-006-020

Prepared by

Paul W. Osborne, Graduate Student
Donald L. Schilling, Associate Professor
Department of Electrical Engineering
Polytechnic Institute of Brooklyn

PIBEE69-002

1969

TECHNICAL REPORT

THRESHOLD ANALYSIS OF PHASE LOCKED LOOPS

for

National Aeronautics & Space Administration
Electronic Research Center

under

NASA Grant NGR 33-006-020

Prepared by

Paul W. Osborne, Graduate Student
Donald L. Schilling, Associate Professor
Department of Electrical Engineering
Polytechnic Institute of Brooklyn

PIBEE69-002

1969

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont"d)

TABLE OF CONTENTS (Cont'd)

LIST OF FIGURES

LIST OF FIGURES (Cont'd)

ABSTRACT

THRESHOLD ANALYSIS OF PHASE LOCKED LOOPS

BY

PAUL W. OSBORNE

ADVISER: DONALD L. SCHILLING

A new method for estimating the expected number of spikes in phase locked loops is presented. Unlike previous analyses, the technique developed here can be applied to phase locked loops of any order, with or without modulation. The modulation must be deterministic, though a random modulating signal can, in principle, be employed if it is approximated by a deterministic quasi-random signal. The Frequency Modulator with Feedback (FMFB) can also be analysed by the method.

To perform the required spike estimation, it is assumed that the noise is a deterministic function, which is defined as "most-likely noise". The real noise driving the phase locked loop is derived from an IF filter driven by white noise. We replace the resulting low-pass equivalent noises x (t) and y (t) by the conditional expectations

$$x_1(t) = E\ [\ x(t) \,/\, x(0),\ \dot{x}(0)\ ]$$

$$y_1(t) = E\ [\ x(t) \,/\, y(0),\ \dot{y}(0)\ ]$$

The time t = 0, is defined as the time at which the noise trajectory crosses the axis of the in-phase noise component y(t). In a FM discriminator, this is the mid-spike or doublet time, depending on whether or not the noise

vector, placed on the end of the carrier vector, does or does not encircle the origin. Thus, for a given filter, $x_1(t)$ and $y_1(t)$ depend only on the four initial conditions $x(0) = 0, \dot{x}(0), y(0)$, and $\dot{y}(0)$. For IF filters of interest (finite bandwidth), the noises $x_1(t)$, $y_1(t)$ approach zero for $|t| >> 0$.

The differential equations of the phase locked loop contain the noises x(t), and y(t). These are replaced by the models $x_1(t)$, and $y_1(t)$. The differential equations are solved on a digital computer, for a particular choice of the parameters $\dot{x}(0)$, $y(0)$, and $\dot{y}(0)$. The solution is started at a time sufficiently negative so that the noise models are close to zero ($t << 0$), and continued to a time when the solution reaches a steady state value, which occurs after the noise models have decayed to zero ($t >> 0$). A fourth order Runge-Kutta starting procedure with a fourth order Moulton's predictor-corrector continuation is used. A hunting procedure in $\dot{x}(0)$, $y(0)$, and $\dot{y}(0)$, is employed, to determine surfaces which define regions in the $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ space, where the steady state phase error between the solution and the modulating phase is either 0, $+2\pi$ or -2π. The expected number of spikes per second is then computed as the expected number of times the random vector $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ **enters** the $+2\pi$ and -2π regions.

The method is applied in this dissertation to the first, second, and third order loops, with and without modulation, for modulation indeces of 3 and 12, and also to the FM Discriminator (which can be represented by a PLL with infinite loop gain). The IF filter used was a cascade of two identical single-tuned stages, each stage, having a normalized 3 db bandwidth of $\pm$ 1 radian per second. The results are presented in normalized form.

Experimental results are presented, which compare favourably to the theoretical results obtained.

CHAPTER 1

INTRODUCTION

1.1 Statement of the Problem

The Phase Locked Loop (PLL) is an oft-used device for obtaining threshold extension in FM demodulation. Threshold occurs when the signal-to-noise ratio improvement of FM demodulation is degraded by some fixed amount (1 db is standard) by the onset of spikes of area $\pm 2\pi$.[1.1] The problem to be solved in this Dissertation, is to estimate the expected number of spikes obtained in a given PLL, with and without modulation. From this, the threshold performance of the PLL can be predicted.

1.2 Summary of Prior Work

Fokker-Planck techniques have been applied to the problem of estimating the threshold performance of PLL by Tikhonov[1.2] and Viterbi.[1.3] Using this technique, an exact solution has been obtained for the first order PLL, and approximate results for the second order PLL, both without modulation (i.e. PLL is used for tracking rather than demodulation), and driven by white noise. It was indicated in Viterbi's paper[1.3] that an approximate solution for the first order PLL with modulation could also be obtained. However, for higher order loops, with modulation, the Fokker-Planck equation becomes intractible. The assumption of white noise input, which is equivalent to saying that the PLL bandwidth is much narrower than the IF bandwidth preceding it, is also restrictive.

Boonton's quasi-linearization technique[1.4] has been applied to the PLL threshold problem by Develet.[1.5] This paper does take the modulation into account, but also assumes a white noise input. His assumption that the phase error of the PLL has an almost-Gaussian density, together with his unusual definition of threshold, yields the un-physical result that the output

noise variance becomes unbounded, when the input signal-to-noise ratio falls below a certain level.

Volterra's functional expansion has been applied to PLL threshold analysis by Van Trees.[1.6] However, the method becomes impractically complex when one attempts to apply it to higher order PLLs, especially if one wishes to take the modulation into account.

R. C. Tausworthe presents another approximate PLL threshold analysis,[1.7] but neglects the effect of modulation. Thus his analysis is applicable to carrier tracking, but not to demodulation.

Schilling and Billig[1.8] attacked the problem by applying Rice's technique[1.9] for a F M D to the PLL. Modulation was considered. In a series of papers,[1.10] Schilling postulated and then showed that the output of a PLL can be considered to consist of the demodulated signal, ordinary FM noise and spike noise. The simple procedure employed to calculate the expected number of spikes/sec, while analytically tractible, resulted in thresholds up to 7 db better than those actually obtained. Schilling also showed, in thse papers, that the PLL demodulation required a larger bandwidth, than the IF filter. It is shown here (Sec. 1.4.3) that this is not strictly correct.

Schilling's representation of the PLL outputs coupled with the use of the "most likely" input noise, form the foundation of this dissertation.

1.3 Summary of Results Obtained

In contradistinction to previous analyses, the effects of the IF filter preceding the PLL (implying non-white noise input) and of the modulation, will be taken into account. Results are presented for a constant-frequency-offset modulation. This is equivalent to square-wave modulation, where the frequency of the square wave is low enough, so that spikes occurring at

transitions of the square wave constitute a negligible proportion of the total number of spikes present. Note, also that the square wave modulation must have rounded corners and finite rise times, so that the FM signal is compatible with IF filtering. Since the number of spikes produced at the output of a PLL increases when the frequency offset increases, the results obtained represent a "worst-case" solution. The magnitude of this constant-frequency-offset modulation is $\Delta\omega = \beta\omega_m$, the peak deviation of a sinusoidal modulation of frequency ω_m radians per second, and a modulation index of β. In Chapter 7, it is shown how to apply this method to any deterministic modulation.

The results obtained assume, that the spikes are independent and have an area of $\pm 2\pi$. Since spikes are caused by rotations of the noise vector about the carrier vector, the spikes must have areas of $2\pi k$ where $k = 1, 2, 3, \ldots$. The experimentally obtained results indicate that multiple rotations ($k > 1$) are relatively infrequent. In addition, the experimental results show that at and above threshold the spikes are relatively independent and do not occur in bursts.

The results of this technique are of course approximate, since deterministic models (see Chapter 2) are used for the noise, and the above approximations are made.

Theoretical results are presented in Chapter 4, and experimental results and verification of the Theory is discussed in Chapter 5. It is shown that threshold performance of the PLL improves as the order of the PLL increases. It is also shown, for the second and third order loops, that the threshold is slightly better for a relative modulation index of 12 than for one of 3. The threshold, with modulation, (for PLL gains suitable for use with a modulation index of 3) is 6 db for a second order PLL, and 5.2 db

for the third order PLL, as compared to a 8.9 db for the F M D. When the PLL Gains are suitable for use with a modulation index of 12, the above figures become 7.5 db, 5 db and 1.5 db for the first, second and third order PLL's respectively, while it is 10.8 db for the F M D. With a constant offset modulation of $\Delta\omega = \beta\omega_m$, thresholds of 7.2 db, and 6.3 db were obtained for the second and third order loops respectively, and a modulation index of 3. For a modulation index of 12, these thresholds become 7.2 db and 4.2 db for the second and third order PLL's respectively, and 10.2 db for the first order PLL.

1.4 Mathematical Formulation of the Problem

1.4.1 FM Discriminator

The output of an FM Discriminator, when integrated is:

$$V_{FMD}(t) = \Phi_m(t) + \arctan\left(\frac{x(t)\cos\Phi_m + y(t)\sin\Phi_m}{1 + x(t)\sin\Phi_m - y(t)\cos\Phi_m}\right) \qquad (1.4.1)$$

where

$\Phi_m(t)$ = phase of the modulating signal

$x(t)$ = quadrature lowpass equivalent noise

$y(t)$ = in phase lowpass equivalent noise

A spike is obtained from the FM Discriminator output, when the arctan $(\cdot)$ jumps $\pm 2\pi$[1.9].

1.4.2 First Order Phase Locked Loop

A block diagram of a first order phase locked loop is shown in Figure 1.4.1. The differential equation describing the loop is easily shown to be

$$\dot{\Phi} + G\sin(\Phi - \Phi_m) = G(x(t)\cos\Phi + y(t)\sin\Phi) \qquad (1.4.2)$$

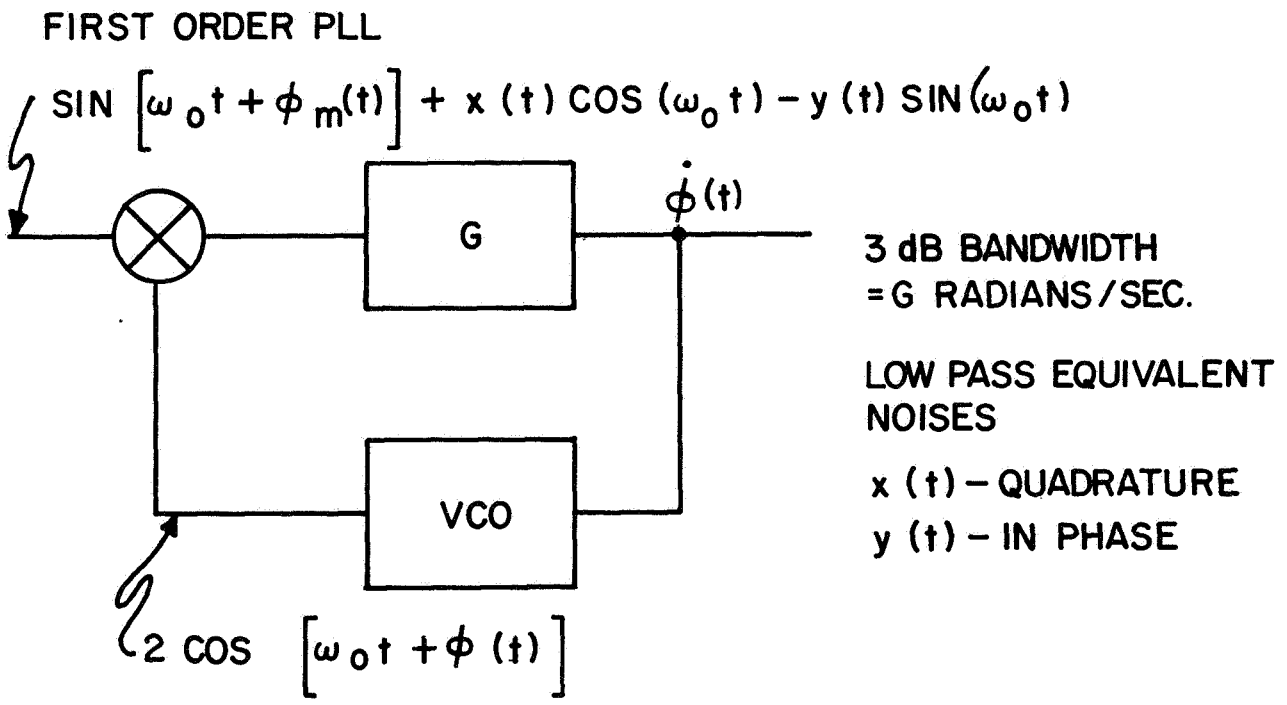

FIRST ORDER PLL
SIN [ω0t + φm(t)] + x(t) COS (ω0t) − y(t) SIN (ω0t)
G
φ̇(t)
VCO
2 COS [ω0t + φ(t)]
3 dB BANDWIDTH
= G RADIANS/SEC.
LOW PASS EQUIVALENT
NOISES
x(t) − QUADRATURE
y(t) − IN PHASE

FIG. 1.4.1

where Φ = phase of VCO output

G = loop gain and 3 db bandwidth of PLL

We let the modulating signal be $2\pi\Delta f$. Thus $\Phi_m(t) = (2\pi\Delta f)t$. When considering noise, this represents a "worst case" solution (see Sec. 5.4.6 where it is shown that the spike rate increases with deviation). The solution of Eq. 1.4.2 with no noise is

$$\Phi(t) = (2\pi\Delta f)t - \arcsin\left(\frac{2\pi\Delta f}{G}\right) \tag{1.4.3}$$

For proper operation of the PLL (low distortion), the error voltage $(\Phi - \Phi_m)$ must be much smaller, in magnitude, than $\pi/2$, or

$$\arcsin\left(\frac{2\pi\Delta f}{G}\right) \ll \frac{\pi}{2} \tag{1.4.4}$$

which implies that

$$G \gg 2\pi\,|\Delta f| \tag{1.4.5}$$

Notice that if the IF half-bandwidth is $(\beta + 1) f_m$ (Carson's rule), and Δf is approximately equal to βf_m, then the bandwidth G of the first order PLL is much greater than the IF bandwidth. (Thus the input noise is certainly not white).

1.4.3 Second Order Phase Locked Loop

A block diagram of a second-order constant-plus-integral phase locked loop is shown in Figure 1.4.2 The differential equation describing the loop is:

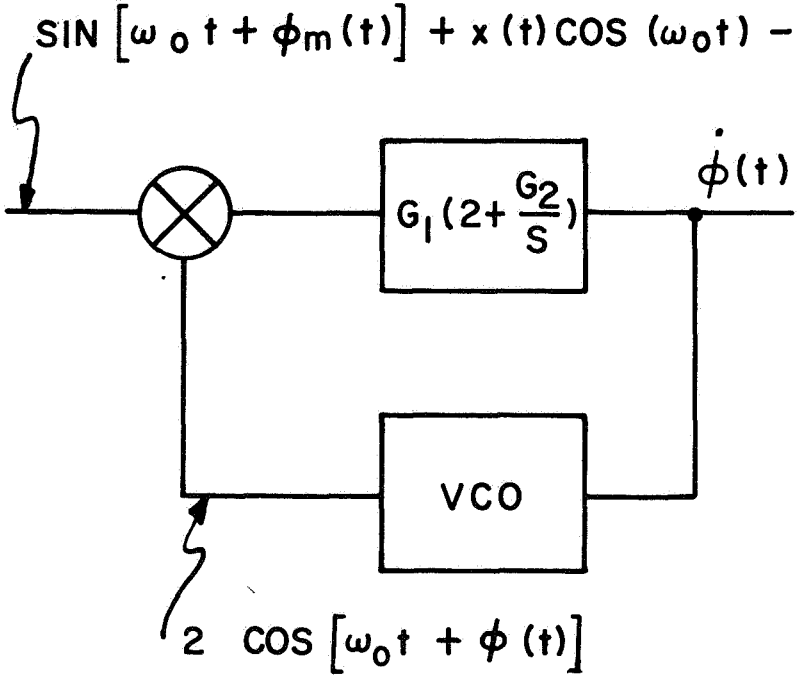
$$\ddot{\Phi} + 2G_1[x(t)\sin\Phi - y(t)\cos\Phi + \cos(\Phi - \Phi_m)]\dot{\Phi} + G_1 G_2 \sin(\Phi - \Phi_m)$$

$$= G_1[(2\dot{x} + G_2 x)\cos\Phi + (2\dot{y} + G_{2y})\sin\Phi + \dot{\Phi}_m \cos(\Phi - \Phi_m)] \tag{1.4.6}$$

where Φ is the phase of the VCO output

$G_1(2 + G_2/S)$ is the transfer function of the constant plus integral

SECOND ORDER PLL (CONSTANT PLUS INTEGRAL)



MAXIMALLY FLAT
WHEN $G_1 = G/2$
$G_2 = 2G$

3 dB BANDWIDTH
= G RADIANS / SEC

LOW PASS EQUIVALENT NOISES

x(t) - QUADRATURE
y(t) - IN PHASE

FIG. 1.4.2

filter.

When there is no noise, and $|\Phi-\Phi_m| << \frac{\pi}{2}$ (so that the distortion is small), the PLL equation becomes

$$\Phi(p) = \frac{2G_1 p + G_1 G_2}{p^2 + 2G_1 p + G_1 G_2} \Phi_m(p) \qquad (1.4.8)$$

If (for reasons given in the next paragraph) we let

$$G_1 = G/\sqrt{2} \qquad (1.4.9a)$$

$$G_2 = G\sqrt{2} \qquad (1.4.9b)$$

Equation 1.4.8 reduces to

$$\Phi(p) = \frac{\sqrt{2}\,Gp + G^2}{p^2 + \sqrt{2}\,Gp + G^2} \Phi_m(p) \qquad (1.4.10)$$

The 3 db bandwidth of the PLL is $\sqrt{\sqrt{5}+2}\;\; G = 2.05\ G$.

A more illuminating way to view this loop is to consider the transfer between the modulating phase and the error phase of the loop, since for proper operation, (low distortion) this error phase must be much less than $\pi/2$ (See Section 1.4.6). Then

$$\Phi_e(p) = \Phi_m(p) - \Phi(p) = \frac{p^2}{(p^2 + \sqrt{2}\,Gp + G^2)} \Phi_m(p) \qquad (1.4.11)$$

This represents a high pass filter with a 3 db cut-off frequency of G radians per second. This means that the phase error is small for modulating frequencies much less than G. The choice of G_1 and G_2 made by Eqs. (1.4.9a) and (1.4.9b), makes this high pass filter maximally flat. This was found to give the best performance, though only slightly better than critically damped or other pole positions which were employed. This result is shown in Appendix F, but which can best be understood after the concept of "Spike Boundaries" have been discussed (Chapter 4). For

proper operation, the modulating frequencies must be considerably less than G, in order to maintain a small phase error (for low distortion). Thus, the error bandwidth G is of more practical interest than the PLL bandwidth. For example, if

$$\Phi_m(t) = \beta \sin \omega_m t.$$

$$\Phi_e(t) = \frac{\beta \omega_m^2}{\sqrt{G^4 + \omega_m^4}} << \frac{\pi}{2} \qquad (1.4.13)$$

$$G >> \left[\frac{4}{\pi^2}\beta^2 - 1\right]^{\frac{1}{4}} \omega_m \qquad (1.4.14)$$

1.4.4 RC Loop Filter

A second order phase locked loop with an RC loop filter was considered, but gave poorer results, since, to have the same distortion performance, its bandwidth is larger.

The differential equation for this PLL is

$$\ddot{\Phi} + \alpha \dot{\Phi} + \alpha G \sin(\Phi - \Phi_m) = \alpha G(x(t)\cos\Phi + y(t)\sin\Phi) \qquad (1.4.15)$$

where $\frac{\alpha}{S+\alpha}$ is the transfer of the RC filter.

For $|\Phi - \Phi_m| << \frac{\pi}{2}$

$$\Phi = \frac{\alpha G}{p^2 + \alpha p + \alpha G} \Phi_m(p) \qquad (1.4.16)$$

This is maximally flat for $\alpha = 2G$. The error phase transfer is given by:

$$\Phi_e = (\Phi_m - \Phi) = \frac{p^2 + 2Gp}{p^2 + 2Gp + 2G^2} \Phi_m(p) \qquad (1.4.17)$$

If $\Phi_m(t) = \beta \sin \omega_m t$ then

$$\Phi_e(t) = \beta \sqrt{\frac{\omega_m^4 + 4G^2\omega_m^2}{\omega_m^4 + G^4}} \sin\left[\omega_m t - \arctan\left(\frac{2G\omega_m}{2G^2 - \omega_m^2}\right) - \arctan\left(\frac{2G}{\omega_m}\right)\right] \qquad (1.4.18)$$

The same distortion is generated in the two loops if the magnitude of the phase errors, given by Eqs. 1.4.12 and 1.4.18, are the same. Thus, equating the squares of these magnitudes, dropping out the β^2 term, and dividing by ω_m^4 we get:

$$\frac{1}{1 + \left(\frac{G_{CI}}{\omega_m}\right)^4} = \frac{1 + 4\left(\frac{G_{RC}}{\omega_m}\right)^2}{1 + 4\left(\frac{G_{RC}}{\omega_m}\right)^4} \qquad (1.4.19)$$

where the subscripts CI and RC refer to constant plus integral and resistance capacitance filters, respectively.

Since the 3 db lower cut-off frequency of the RC second order loop phase error transfer is $\sqrt{2\sqrt{5} + 4}\ G_{RC} = 2.91\ G_{RC}$ radians per second (from Eq. 1.4.17), we rearrange 1.4.19 to get

$$\frac{1}{1 + \left(\frac{G_{CI}}{\omega_m}\right)^4} = \frac{1 + .474\left(\frac{2.91\ G_{RC}}{\omega_m}\right)^2}{1 + .057\left(\frac{2.91\ G_{RC}}{\omega_m}\right)^4} \qquad (1.4.20)$$

In order to satisfy Eq. 1.4.20,

$$(2.91\ G_{RC})\ (.057)^{\frac{1}{4}} > G_{CI} \qquad (1.4.21)$$

or

$$(2.91\ G_{RC}) > 2.05\ G_{CI} \qquad (1.4.22)$$

Since the cut-off frequency of the constant plus integral phase locked loop phase error transfer from Eq. (1.4.11) is just $2.05\ G_{CI}$, this cut-off

frequency of the RC phase locked loop phase error transfer must be more than twice as large as that of the constant plus integral PLL when both loops develop the same distortion.

1.4.5 Third Order Phase Locked Loop

A block diagram of a third order constant-plus-integral plus double integral PLL, is shown in Fig. 1.4.3. The differential Eq. describing the loop is

$$\dddot{\Phi} = G_1 \ddot{e}(t) + G_1G_2 \dot{e}(t) + G_1G_2G_3 e(t) \qquad (1.4.23)$$

where $e(t) = -\sin(\Phi - \Phi_m) + x(t)\cos\Phi + y(t)\sin\Phi$

$G_1 + G_1G_2/S + G_1G_2G_3/S^2$ is the transfer function of the PLL filter.

When there is no noise and $|\Phi - \Phi_m| << \frac{\pi}{2}$, the PLL equation becomes,

$$\dddot{\Phi} + G_1\ddot{\Phi} + G_1G_2\dot{\Phi} + G_1G_2G_3\Phi = G_1\ddot{\Phi}_m + G_1G_2\dot{\Phi}_m + G_1G_2G_3\Phi_m \qquad (1.4.24)$$

Again, considering the phase error, we let $G_1 = 2G$, $G_2 = G$, and $G_3 = G/2$, which makes the phase error transfer maximally flat,

$$\Phi_e(p) = \Phi_m(p) - \Phi(p) = \frac{p^3}{p^3 + 2Gp^2 + 2G^2p + G^3}\,\Phi_m(p) \qquad (1.4.25)$$

This high pass filter has a cut-off frequency of G radians per second. Again, to maintain low distortion, the modulating frequencies must be considerably less than this. If $\Phi_m(t) = \beta \sin \omega_m t$ the phase error developed is

$$\Phi_e(t) = \frac{\beta\omega_m^3}{\sqrt{\omega_m^6 + G^6}} \sin\left[\omega_m t - \arctan\left(\frac{2G^2\omega_m - \omega_m^3}{G^3 - 2G\omega_m^2}\right)\right] \qquad (1.4.26)$$

From the above equation, it is clear that for low distortion, G must be such

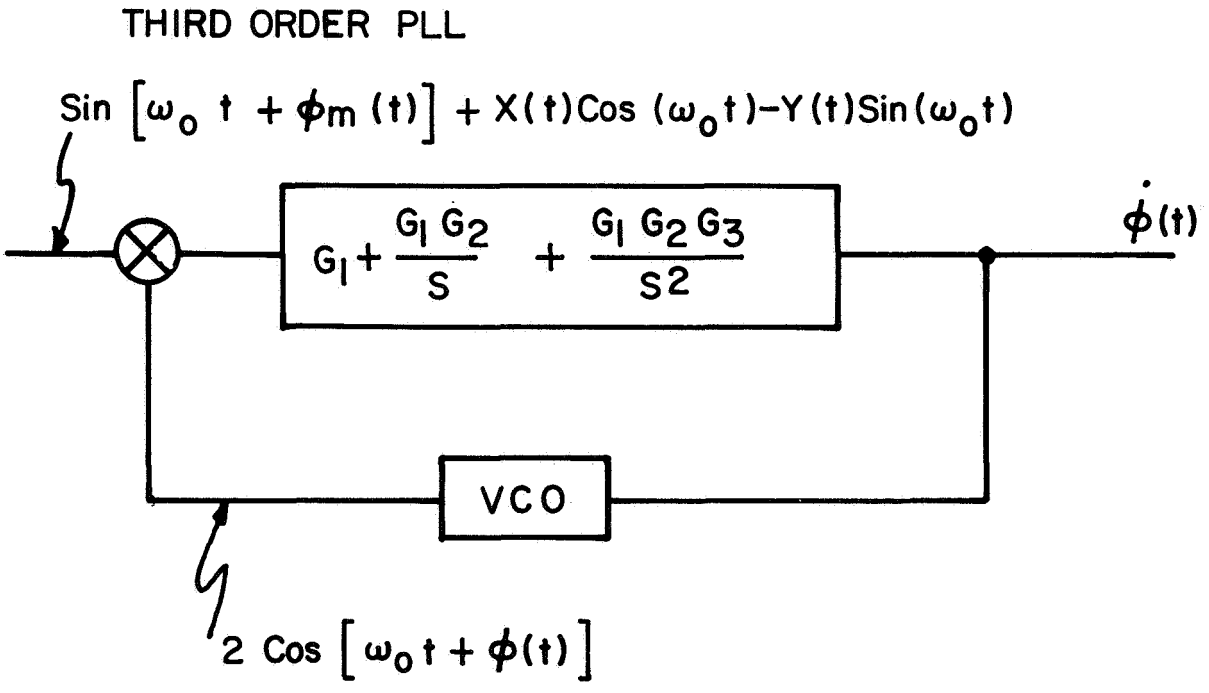

THIRD ORDER PLL
Sin [ω0 t + φm (t)] + X(t)Cos (ω0t)-Y(t)Sin(ω0t)
G1 + G1 G2 / S + G1 G2 G3 / S2
φ̇(t)
VCO
2 Cos [ω0 t + φ(t)]

FIG. 1.4.3

that,

$$\frac{\beta\omega_m^3}{\sqrt{\omega_m^6 + G^6}} << \frac{\pi}{2} \tag{1.4.27}$$

or $$G >> \omega_m \left[\frac{4}{\pi^2} \beta^2 - 1 \right]^{\frac{1}{6}} \tag{1.4.28}$$

In making comparisons between the performances of the second and third order PLL's for different modulation indices β, the gain G in Eqs. 1, 4, 12 and 1.4.26 were adjusted to make the magnitude of the phase errors equal.

1.4.6. On Using the Phase Error Transfer of PLLs

The input-output and phase error amplitude responses (derived from Eqs. 1.4.10 and 1.4.11 respectively) are plotted in Fig. 1.4.4 below. If

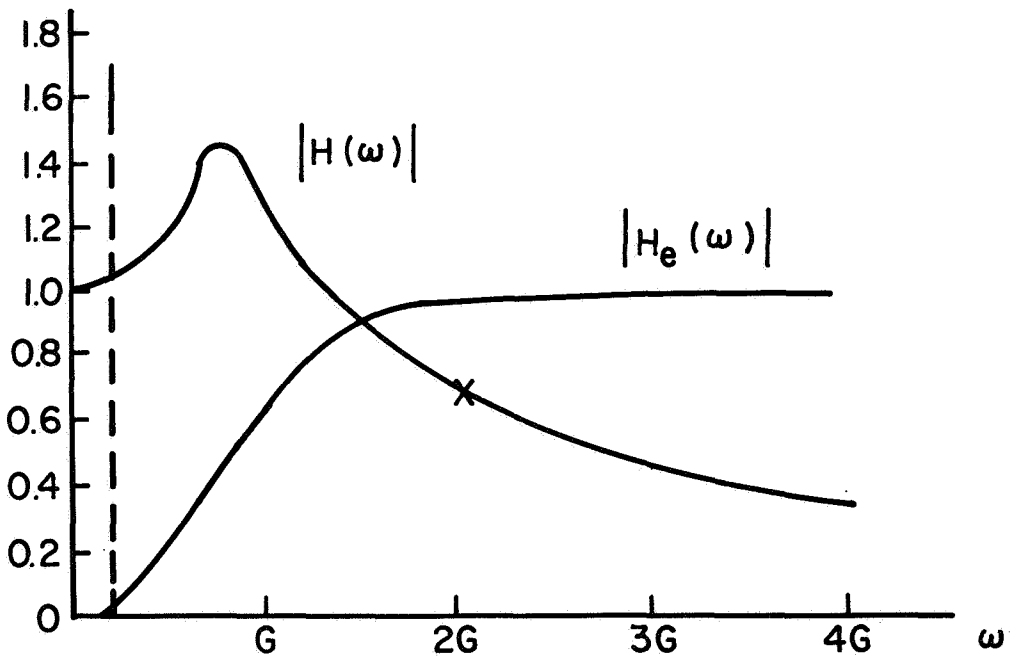


Fig. 1.4.4 Input-Output and Phase Error Responses of Constant-plus-Integral PLL

the phase error is to be less than 0.1 in magnitude, when the modulating

phase is $\beta \sin \omega_m t$, where $\beta = 3$, then the modulating frequencies must lie below that indicated by the broken line in Fig. 1.4.4. The 3 db bandwidth of the loop (indicated by the "x" on the frequency response curve) of about 2G is quite far removed from the frequency indicated by the broked line.

Given the amount of phase error (which is related to the harmonic and intermodulation distortion) one can tolerate, one can see at a glance at the phase error amplitude response, what range of modulating frequencies the PLL will track or demodulate successfully. The ordinary input-output response of the PLL ($|H(\omega)|$), however, does not indicate as clearly what modulation frequencies the PLL will demodulate. Indeed, it is misleading, since the "hump" in that response might lead one to believe that the loop is tracking, while in fact, though it does track, the phase error is so large that appreciable harmonic and intermodulation distortion will result.

For the above reasons, in this dissertation, the phase error response, rather than the input-output response of PLL's is used in comparing and evaluating their "bandwidth".

1.4.7 Computing Threshold from Expected Number of Spikes per Second

The FM demodulator (FM discriminator, or PLL) produces, at a given input carrier-to-noise ratio, an average of N spikes per second. The output signal to noise ratio, can be expressed as a function of N, and the threshold calculated. To maximize the output signal-to-noise ratio, a low-pass ideal filter of bandwidth $f_m H_z$ is placed at the output of the FM demodulator.

If the modulating phase is $\beta \sin \omega_m t$, the output signal is $\beta\omega_m \cos \omega_m t$. The output noise power due to the smooth (FM type) noise is [1.6] (for a carrier amplitude of unity)

$$P_N = \int_{-fm}^{fm} \omega^2 S_x(\omega)\, df \qquad (1.4.29)$$

$S_x(\omega)$ is the spectral density of the quadrature low pass equivalent noise. (watts per Hz). For the examples used in this dissertation, this density is

$$S_x(\omega) = \frac{4\sigma^2}{(\omega^2 + 1^2} \qquad (1.4.30)$$

where σ^2 is the variance of the quadrature noise.

Thus 1.4.29 becomes

$$P_N = \frac{2\sigma^2}{\pi} \left[\arctan \omega_m - \frac{\omega_m}{1 + \omega_m^2} \right] \qquad (1.4.31)$$

It is shown in Appendix B that if symmetrical Poisson distributed spikes (delta functions) of area 2π drive an ideal low pass filter of bandwidth ω_m Hz, the resulting output noise power is:

$$P_S = 4\pi N\omega_m \qquad (1.4.32)$$

where N is the average number of spikes per second. Since the output signal power is $\omega_m^2 \beta^2/2$, the output signal-to-noise ratio is:

$$SNR = \frac{\omega_m^2 \beta^2/2}{\frac{2\sigma^2}{\pi}\left[\arctan \omega_m - \frac{\omega_m}{1+\omega_m^2} \right] + 4\pi N\omega_m} \qquad (1.4.33)$$

Since the amplitude of the carrier is taken to be unity, the carrier-to-noise ratio is CNR = $1/2\sigma^2$, also note that from Eq. 1.4.30 the 3db IF bandwidth is $(\beta + 1)\omega_m = \sqrt{\sqrt{2} - 1}$. Thus Eq. 1.4.33 may be written

$$\mathrm{SNR} = \frac{\left\{ \dfrac{\pi\,(\sqrt{2}-1)\beta^2\,\mathrm{CNR}}{2\left[\arctan\left(\dfrac{\sqrt{\sqrt{2}-1}}{\beta+1}\right) - \dfrac{1}{\left(\dfrac{\sqrt{\sqrt{2}-1}}{\beta+1} + \dfrac{\beta+1}{\sqrt{\sqrt{2}-1}}\right)}\right](\beta+1)^2} \right\}}{\left\{ 1 + \dfrac{4\pi^2 N\sqrt{\sqrt{2}-1}\,\mathrm{CNR}}{(\beta+1\left[\arctan\dfrac{\sqrt{\sqrt{2}-1}}{\beta+1} - \dfrac{1}{\left(\dfrac{\sqrt{\sqrt{2}-1}}{\beta+1} + \dfrac{\beta+1}{\sqrt{\sqrt{2}-1}}\right)}\right]} \right\}} \qquad (1.4.34)$$

When β is greater than unity, which represents the majority of practical systems, $\omega_m \ll 1$, the

$$\arctan\omega_m - \frac{1}{\left(\omega_m + \frac{1}{\omega_m}\right)} \doteq \frac{2}{3}\,\omega_m^3 \qquad (1.4.35)$$

and the output SNR becomes

$$\mathrm{SNR} = \frac{\dfrac{3\pi}{4(\sqrt{2}-1}\,\beta^2\,(\beta+1)\,\mathrm{CNR}}{1 + \dfrac{12\pi^2 N(\beta+1)^2}{(\sqrt{2}-1)}\,\mathrm{CNR}} \qquad (1.4.36)$$

The numerators in Eqs. 1.4.35 and 1.4.36 represent the output signal-to-noise ratio for high input carrier-to-noise ratios. Note, also that the SNR is a linear function of CNR for a given β, so long as the denominator is nearly 1.

Threshold is defined as the CNR occurring when the SNR has decreased below this linear relationship by 1 db. This is equivalent to the denominator being $\simeq 1.25$. Thus the threshold CNR is

$$CNR_{TH} = \frac{(\beta+1)\left[\arctan\left(\frac{\sqrt{\sqrt{2}-1}}{\beta+1}\right) - \frac{1}{\left(\frac{\sqrt{\sqrt{2}-1}}{\beta+1} + \frac{\beta+1}{\sqrt{\sqrt{2}-1}}\right)}\right]}{16\pi^2 N\sqrt{\sqrt{2}-1}} \qquad (1.4.37)$$

From Eq. 1.4.37, we can see that if we have plotted N as a function of CNR, for a particular β (modulation index), the threshold occurs when the N curve intersects the hyperbola defined by Eq. 1.4.37.

CHAPTER 2

THE MOST LIKELY NOISE MODEL

2.1 Bandpass Noise Representation

If a bandpass filter is driven by white Gaussian noise, its output can be represented by (see appendix A)

$$z(t) = x(t)\cos\omega_o t - y(t)\sin\omega_o t \qquad (2.1.1)$$

It is shown in appendix A that if the bandpass filter is symmetric about ω_o, then x(t) and y(t) are independent low pass Gaussian noises. It is also shown that if $S_z(\omega)$ is the noise spectrum of the bandpass filter then

$$S_x(\omega) = S_y(\omega) = S_z(\omega-\omega_o)+S_z(\omega+\omega_o) \text{ when } |\omega|<\omega_o = 0 \qquad (2.1.2)$$
$$= 0 \qquad \text{when } |\omega|>\omega_o$$

Thus the properties of x(t) and y(t) are derived by driving the low pass equivalent of the symmetric bandpass filter with white noise.

The examples used in the computer simulation portion of this dissertation use an IF filter consisting of a cascade of two identical single-tuned stages each having a 3 db bandwidth of $\pm$ 1 radian per second. Such a filter is a symmetric bandpass filter only in the limit as its centre frequency increases without bound (while of course maintaining the same bandwidth). (Since experimentally, an infinite centre frequency is impossible, the assumption of symmetry introduces some error). The Fourier transform of the low pass equivalent of such a filter is

$$H(\omega) = \frac{1}{(j\omega+1)^2} \qquad (2.1.3)$$

from which the spectral densities of x(t), and y(t) are

$$S_x(\omega) = S_y(\omega) = \frac{4\sigma^2}{(\omega^2+1)^2} \qquad (2.1.4)$$

where σ^2 is the variance of x(t) and y(t). Taking inverse Fourrier transforms, we find their autocorrelation to be

$$R_x(t) = R_y(t) = R(t) = \sigma^2 (1 + |t|)e^{-|t|} \tag{2.1.5}$$

2.2 Noise Models[2.1]

We need to evaluate the conditional expectations

$$x_1(t) = E[x(t)/x(0), \dot{x}(0)] \tag{2.2.1a}$$

$$y_1(t) = E[y(t)/y(0), \dot{y}(0)] \tag{2.2.1b}$$

To do this we need the conditional density function

$$f(x(t)/x(0),\dot{x}(0)) = \frac{f(x(t), x(0), \dot{x}(0))}{f(x(0), \dot{x}(0))} \tag{2.2.2}$$

This may be computed by using the standard form of a joint Gaussian density function

$$f(x_1, x_2 \ldots x_n) = \frac{1}{(2\pi)^{\frac{n}{2}} |C|} e^{\frac{1}{2}} \left\{ (x_1, x_2 \ldots x_n) C^{-1} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \right\} \tag{2.2.3}$$

where the elements of matrix C are $c_{ij} = E[x_i x_j]$.

Using the autocorrelation function, defined by Eq. (2.1.5) we find

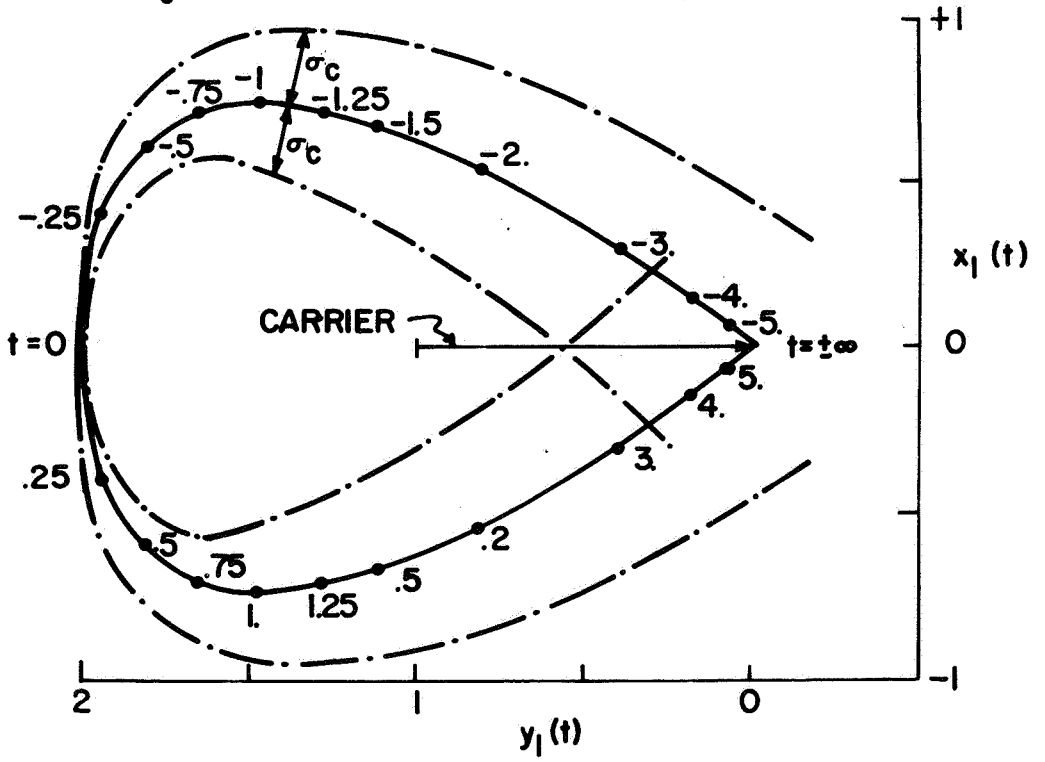
$$\begin{aligned} &E[x(t)^2] = E[x(0)^2] = \sigma^2 \\ &E[x(t)\, x(0)] = R(t) = \sigma^2 (1 + |t|)\, e^{-|t|} \\ &E[x(t)\, x(0)] = -R(t) = \sigma^2\, t\, e^{-|t|} \\ &E[x(0)\, x(0)] = -R(0) = 0 \\ &E[x(0)^2] = -R(0) = \sigma^2 \end{aligned} \tag{2.2.4}$$

Thus, from Eqs. (2.2.2), (2.2.3) and (2.2.4)

POLAR PLOT OF NOISE MODELS WITH BOUNDS EQUAL TO THE SQUARE ROOT OF THE CONDITIONAL VARIANCE

POLAR PLOT OF NOISE MODELS, $\dot{x}(0)=-2$, $\dot{y}(0)=0$, $y(0)=2$. ——●——●——●——●——

BOUNDS σ_c (MEASURED AS SHOWN BELOW) —·—·—·—·—·—



$$\sigma_c = \sigma\sqrt{1-\left\{(1+|t|)^2+t^2\right\}e^{-2|t|}}$$

σ^2 = VARIANCE OF $x(t)$, $y(t)$; $\sigma = 1/\sqrt{8}$. THIS CORRESPONDS TO A CNR $\left[1/2\sigma^2\right]$ OF 4.

FIG. 2.1

$$f\,(x(t)/x(o), \dot{x}\,(o)) =$$

$$\frac{1}{\sqrt{2\pi\left[1-\frac{R^2(t)}{\sigma^4}-\frac{\dot{R}^2(t)}{\sigma^4}\right]\sigma^2}}\; e^{-\frac{\left[x(t)-\left(\frac{R(t)}{\sigma^2}\,x(0)-\frac{\dot{R}(t)}{\sigma^2}\,\dot{x}\,(0)\right)\right]^2}{2\sigma^2\left[1-\frac{R^2(t)}{\sigma^4}-\frac{\dot{R}^2(t)}{\sigma^4}\right]}} \qquad (2.2.5)$$

Since x(t) and y(t) have identical statistics, the conditional density for y(t) has exactly the same form as Eq. (2.2.5).

The noise model for the quadrature noise x(t) is evident from Eq. (2.2.5),

$$x_1(t) = \left(\frac{R(t)}{\sigma^2}\,x\,(0) - \frac{\dot{R}(t)}{\sigma^2}\,x\,(0)\right)$$

$$= [x(0)\,(1+|t|) + \dot{x}\,(0)\,t]\; e^{-|t|} \qquad (2.2.6)$$

Since $t = 0$ is chosen to be the time when $x(0) = 0$, (equivalent to the time when a trajectory of x(t), y(t) crosses the y(t) axis) Eq. (2.2.6) becomes

$$x_1(t) = \dot{x}(0)\,t\,e^{-|t|} \qquad (2.2.7)$$

Similarly the model for y(t), the in phase noise, is

$$y_1(t) = [y(0)(1+|t|) + \dot{y}(0)t]\; e^{-|t|} \qquad (2.2.8)$$

A trajectory of these noise models is shown in Fig. 2.1. Notice that at $t=0$ this trajectory crosses the y(t) axis, i.e. $x(0) = 0$.

A trajectory of these noise models is shown in Fig. 2.1. Notice that at $t=0$ this trajectory crosses the y(t) axis, i.e. $x(0) = 0$.

2.3 Variance of the Conditional Noise about the Noise Models

The variance of the Gaussian density of Eq. (2.2.5) is a measure of how good the noise models of section 2.2 are. This variance is:

$$\sigma\left[1-\frac{R^2(t)}{\sigma^4}-\frac{\dot{R}^2(t)}{\sigma^4}\right] = \sigma^2\left\{1-\left[(1+|t|)^2+t^2\right]e^{-2|t|}\right\} \qquad (2.2.9)$$

The square root of the above variance is used as a bound in Fig. 2.1 indicating the likely variation of the noise (conditioned on $\dot{x}\,(0)$, $y(0)$, $\dot{y}(0) = 0$) from the noise model. In Fig. 2.1, $\sigma = 1/\sqrt{8}$, corresponding to a CNR $= 1/2\sigma^2$

of 4 (6db). Higher Carrier to Noise Ratios will cause the bounds to converge more closely to the noise models. Thus the noise model becomes better and better with increasing CNR's. Note that at this value of CNR, which is a typical threshold value (see Sec. 6.2) the variance is 0 at $t=0$, $0.93\sigma^2$ at $|t| = 1$ and σ^2 at $|t| \rightarrow$ infinity.

CHAPTER 3

COMPUTER PROGRAMMING OF THE PLL USING MOST LIKELY NOISE (MLN)

3.1 General Description

The low-pass equivalent noises x(t), (quadrature noise) and y(t), (in phase noise) which appear in differential equations of the PLL, are replaced by the most-likely noise models (MLN) developed in Chapter 2. Thus, the differential equations are deterministic, with the random parameters $\dot{x}(0)$, y(0), and $\dot{y}(0)$.

Given the above, the computer program achieves the following:

a) It solves the differential equations of the PLL, over a specified interval of time, (I mean, of course, time as the argument of the solution, not as the actual time of computation,) and for values of the parameters $\dot{x}(0)$, y(0), $\dot{y}(0)$ specified by part (b) of the program. The time interval must be such that the initial time is a sufficiently negative time so that the noise models are near zero, and the final time is a sufficiently positive time so that the solution has reached a steady state value. For the examples considered, -5 to + 25 seconds was an adequate time interval.

b) It performs a hunting procedure on the parameters $\dot{x}(0)$, y(0), $\dot{y}(0)$ to determine regions in the space of $\dot{x}$ (0), y(0) and $\dot{y}(0)$, such that the steady state error phase (the steady state solution obtained in part (a) minus the modulating phase) is $+ 2\pi$ or $- 2\pi$. The program prints out values of $\dot{x}(0)$, y(0), and $\dot{y}(0)$ on the boundaries of such regions.

These boundaries or spike surfaces may be plotted using the output of the above program. They can be used to estimate the expected number of spikes of the PLL. A second computer program, described at the end of this chapter, is used to perform the three dimensional integration

required to determine the expected number of spikes.

3.2 Program to Solve PLL Differential Equations (Part (a) of Program)

The differential equation describing the PLL is written as a first order differential vector equation.

$$\frac{d}{dt}[\underline{Z}] = \underline{F}\,(\underline{Z}, t, \dot{x}(0), y(0), \dot{y}(0)) \qquad (3.2.1)$$

The function on the right of Eq. 3.2.1 also contains parameters defining the PLL gain and filter, however, for simplicity, these are not explicitly shown here. As an example, the differential equation describing the second order PLL may be written (Eq. 1.4.6) as

$$\frac{d}{dt}\begin{bmatrix}\Phi \\ \dot{\Phi}\end{bmatrix} = \begin{Bmatrix}\dot{\Phi} \\ G_1[(2y\dot{\Phi} + 2\dot{x} + G_2 x)\cos\phi + (-2x\dot{\Phi} + 2\dot{y} + G_2 y)\sin\Phi \\ \qquad -(\dot{\Phi}-\dot{\Phi}_m)\cos(\Phi-\Phi_m) - G_2\sin(\Phi-\Phi_m)]\end{Bmatrix} \qquad (3.2.2)$$

A flow chart for Part (a) of the Computer Program is shown in Fig. 3.2.1.

Block 1 sets the initial constants. For simplicity, only those constants used in the Flow Chart have been included. The use of the Constants EF_1, EF_2 and DT will become evident later in the Flow Chart.

In block 2, a fourth order Runge-Kutta procedure is used to obtain the values of the solution at T_1, $T_1 + DT$, $T_1 + 2DT$, $T_1 + 3DT$ seconds. The particular Runge-Kutta procedure used[3.1] is as follows:

Referring to the differential Eq. 3.2.1, let $\underline{Z}_1$ be the initial value of the solution, at time T.

Evaluate the constants

$\underline{E}_1 = \underline{F}(\underline{Z}_1, T_1, \dot{x}(0), y(0), \dot{y}(0))$

$\underline{E}_a = \underline{F}(Z_1 + \underline{E}_1 * (.5*DT),\ T_1 + .5*DT,\ \dot{x}(0),\ y(0),\ \dot{y}(0))$

Flow Chart of Part (a) of Computer Program

FIG. 3.2.1

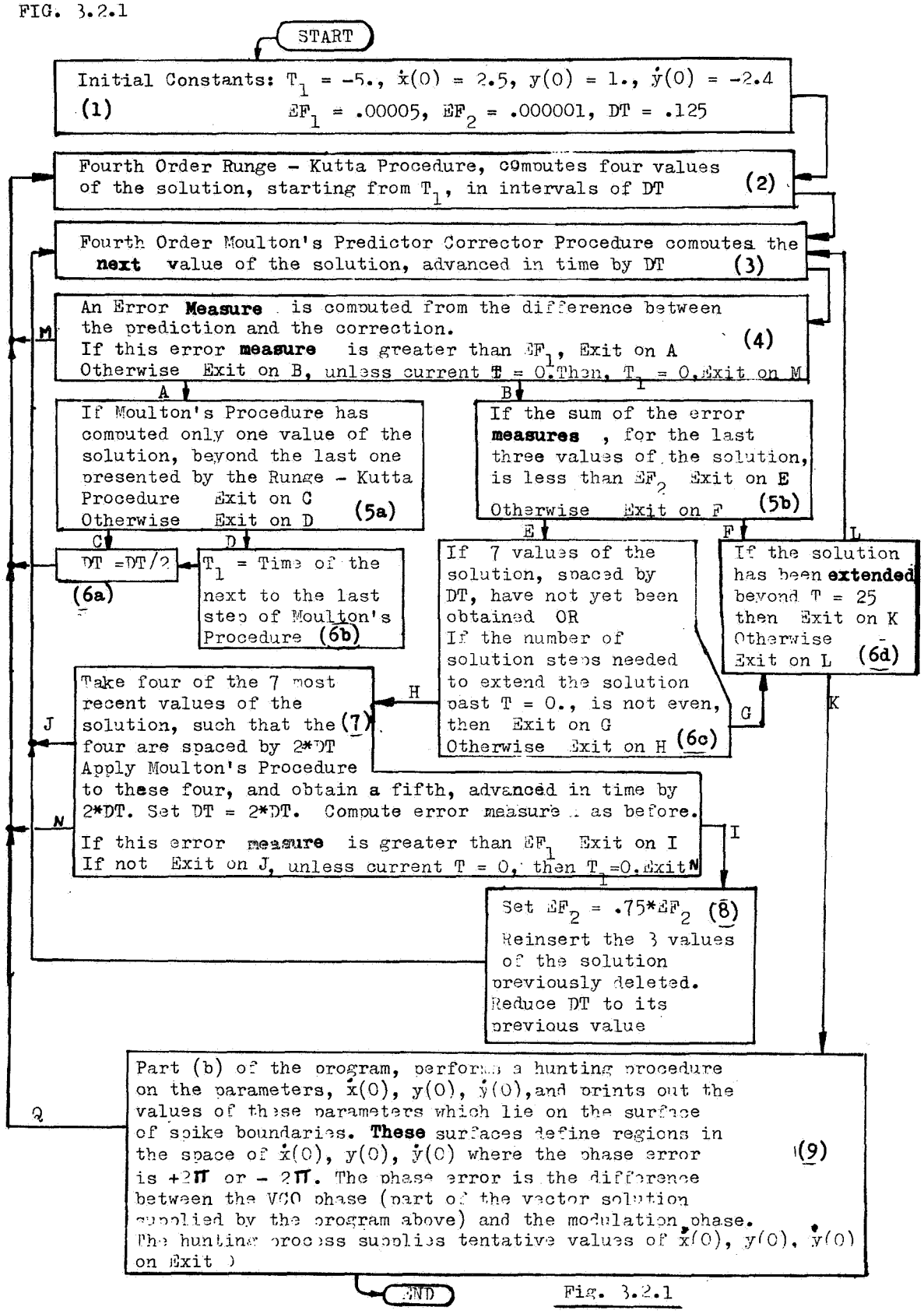


Fig. 3.2.1

$$\underline{E}_b = \underline{F}(\underline{Z}_1 + \underline{E}_a * (.5*DT),\ T_1 + .5*DT,\ \dot{x}(0),\ y(0),\ \dot{y}(0))$$

$$\underline{E}_c = \underline{F}(\underline{Z}_1 + \underline{E}_b *DT,\ T_1 + DT,\ \dot{x}(0),\ y(0),\ \dot{y}(0))$$

Then the value of the solution at $T = T_1 + DT$ is

$$\underline{Z}_2 = \underline{Z}_1 + DT*(\underline{E}_1 + 2*\underline{E}_a + 2*\underline{E}_b + \underline{E}_c)/6 \qquad (3.2.3)$$

The above is repeated three times until values of the solution up to the time $T_1 + 3DT$ seconds have been obtained. In block 3, a fourth-order Moulton's Predictor Corrector procedure is used to compute the value of the solution DT seconds later, i.e. at $T_1 + 4DT$. The Moulton's procedure is as follows:[3.1]

We have the solutions $\underline{Z}_1$, $\underline{Z}_2$, $\underline{Z}_3$, $\underline{Z}_4$ from the Runge-Kutta procedure. Compute

$$\underline{E}_4 = \underline{F}(\underline{Z}_4, T_4, \dot{x}(0), y(0), \dot{y}(0)) \qquad (3.2.4)$$

where $T_4 = T_1 + 3DT$

We make a "prediction" of the next value of the solution:

$$\underline{V} = \underline{Z}_4 + DT*(55*\underline{E}_4 - 59*\underline{E}_3 + 37*\underline{E}_2 - 9*\underline{E}_1)/24 \qquad (3.2.5)$$

Then compute,

$$T_5 = T_4 + DT$$

$$\underline{E}'_5 = \underline{F}(\underline{V}, T_5, \dot{x}(0), y(0), \dot{y}(0))$$

The prime on $\underline{E}'_5$ serves as a reminder that the "predicted" value $\underline{V}$ of the solution at T_5 is used, rather than the final "corrected" value. This final "corrected" value is:

$$\underline{Z}_5 = \underline{Z}_4 + DT*(9*\underline{E}'_5 + 19*\underline{E}_4 - 5*\underline{E}_3 + \underline{E}_2)/24 \qquad (3.2.6)$$

If four values of the solution are known, and equi-spaced in time, then the above procedure can be used to find a new value of the solution one step further out in time.

We now proceed to block 4 in the Flow Chart. The difference between the predicted value of the solution (Eq. 3.2.5) and the corrected one (Eq. 3.2.6)

gives a measure of the error of truncation committed by the process. This error measure is used to adjust the interval of computation, DT. The error-measure is taken to be (in order to give a measure of the relative error)

$$ER = \left| \frac{V' - Z'_n}{Z'_n} \right| \qquad (3.2.7)$$

where the primes are to indicate that the first component of the vectors $\underline{V}$ and $\underline{Z}_n$ are used. The first component of $\underline{Z}$ is the phase of the VCO in the PLL. To avoid dividing by zero when Z'_n is small, Eq. 3.2.7 is changed for Z'_n less than .25, to

$$ER = \frac{|V' - Z'_n|}{.5 - |Z'_n|} \qquad (3.2.8)$$

This error measure, has to be used to control the integration interval, DT, but at the same time, take into account, a number of problems that can arise.

1) Since the noise models contain $e^{-|t|}$, some of the derivatives of the PLL solution will be discontinuous at t = 0. Therefore Moulton's Predictor-Corrector procedure, cannot be used to find a value of the solution at t >0 based on four values of the solution prior to t = 0.

To avoid this difficulty, the solution is restarted at t=0 by the Runge-Kutta procedure (which does not require previous values of the solution, but only an initial value at t=0). Note that an exit on M from block 4 to block 2 is provided when T = 0; T_1 is set equal to zero, since T_1 is the initial time from which the Runge-Kutta procedure begins.

The initial value of the integration interval DT, is so chosen, that an integer number of computation steps will bring the solution exactly to t=0. When the error measure is so small (ER$\leq$ EF, in block 4, and we exist into block 5b via exist B) that we wish to double the value of DT and save computer time, we nevertheless <u>avoid</u> doubling DT if the number of computation steps needed to extend the solution to t=0 is <u>not even</u>. To do so

would cause the program to neglect to evaluate the solution at t=0. Thus, block 6c does not allow an exit on H to block 7, where doubling of DT takes place, unless the number of steps required to extend the solution to t = 0 is even.

2) The error measure ER, computed in block 4, fluctuates in a random manner about some mean value. This presents a minor problem in deciding when the error is small enough so that DT can be doubled, saving computer time, without increasing the error too much. This problem is solved by averaging the values of ER from the last three values of the solution, in block 5b. We consider doubling DT (exit on E to block 6c) when this sum is less than EF_2, a number chosen by trial and error.

3) It is not convenient to double DT until at least 7 values of the solution have been obtained, because to do so would require restarting with Runge-Kutta' s procedure. Thus, block 6c, prevents doubling DT (exit on H to block 7) if this condition is not satisfied.

4) It may happen, that when DT is doubled, and Moulton' s procedure applied, the error is too large. This would imply that we were not really justified in doubling DT, or that EF_2 is too large. Thus, block 7, when it meets this problem, exits on I to block 8, where EF_2 is reduced by 0.75 (so that the program is more conservative in deciding to doube DT in the future) and DT is halved to its original value. In short, the attempt to double DT is given up.

Notice that a successful doubling of DT in block 7 causes an exit at J to the regular Moulton' s procedure (block 3). Notice also, that if t becomes zero, we exit block 7 at N, to restart the solution with the Runge Kutta Method.

5) If the error measure ER is too great ($ER > EF_1$) we must reduce

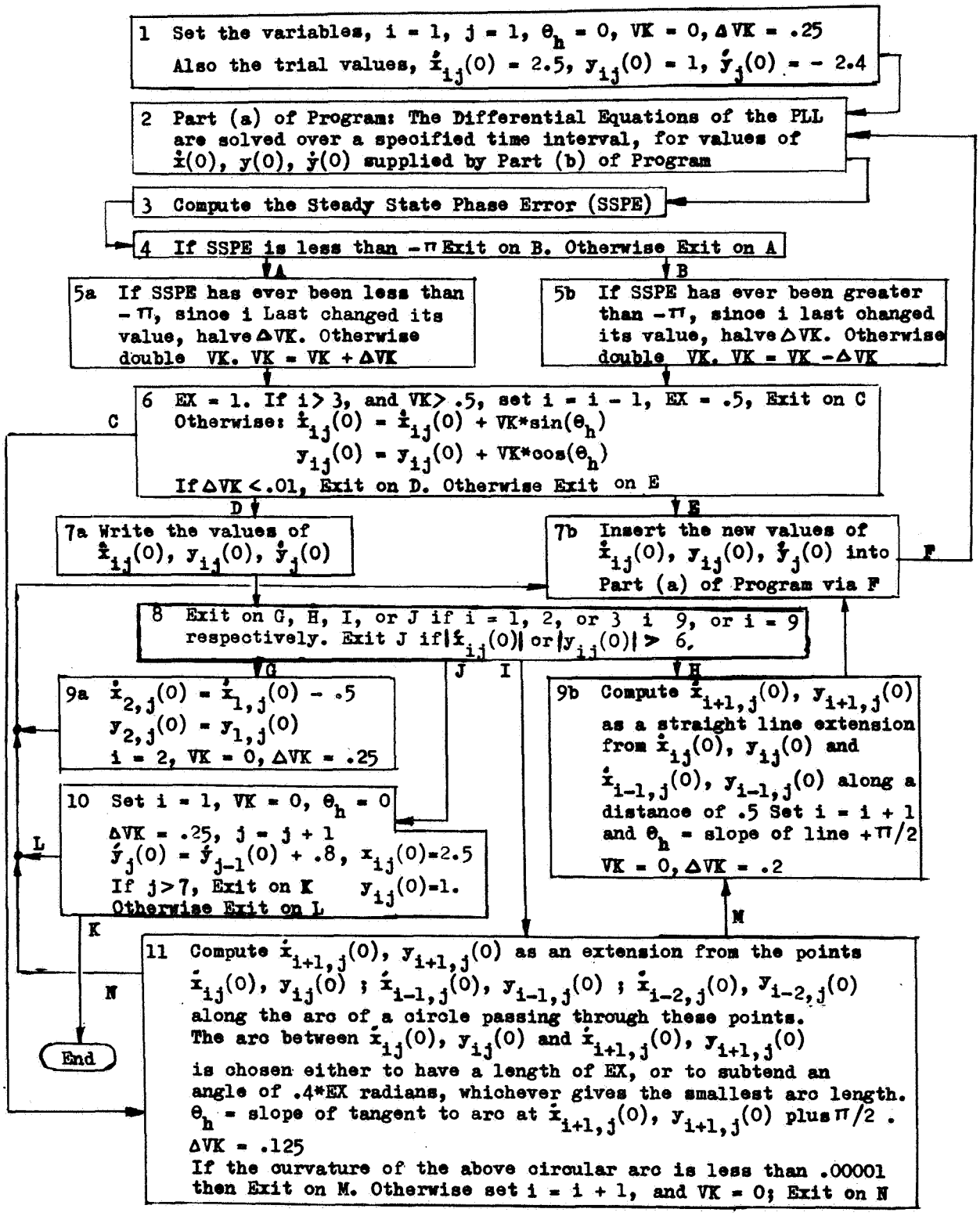
DT, the integration interval. Thus we exit block 4 at exit A into block 5a. However, if the Moulton' s Predictor-Corrector procedure had only computed one value beyond the four solution values presented to it by the Runge-Kutta starting procedure, then too large an error implies that DT was too large for both procedures. Thus, if this is the case, we exit block 5a at C, halve DT, and start all over again with the Runge-Kutta procedure.

However, if the Moulton' s Predictor-Corrector procedure had computed more than one value beyond the four Runge-Kutta values of the solution than too large an error indicates that only the last Moulton' s value of the solution has too large an error. Thus we exit block 5a at D, and set T_1 (the time at which the Runge-Kutta method will begin computing) equal to the time of the second to the last value of the solution supplied by Moulton' s procedure. We then halve DT as before and go to the Runge-Kutta procedure (block 2) and start the solution at the new values of T_1 and DT.

If a change in DT is not being made, (or considered) then the program, after being started by the Runge-Kutta procedure, cycles around blocks 3, 4, 5b and 6d, in the order just listed. Each cycle, provides a new value of the solution, advanced in time by DT. When the solution has been extended beyond t = 25 seconds, block 6d is exited at K and the value of the solution at this final time is presented to Part (b) of the program, block 9. This part of the program will be described in the next section. Here, it should be noted that during the hunting procedure on $\dot{x}(0)$, $y(0)$, and $\dot{y}(0)$, new values of $\dot{x}(0)$, $y(0)$, and $\dot{y}(0)$ are injected via exit Q into the program. The program of Part (a) repeats itself, from the beginning, with the new values of the parameters $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$, as often as Part (b) of the program presents them.

FIG. 3.3.1

Flow Chart of Part (b) of the Computer Program



3.3 Program to find Spike Regions (Part (b) of Program)

A Flow Chart of Part (b) of the Computer Program, which enables us to locate the spike regions, is given in Fig. 3.3.1. This Chart shows how the Computer Program finds regions in $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ space where the Steady State Phase Error (SSPE) is -2π. If it is desired to find the regions for $+2\pi$, it is only necessary to change the sign of the modulation in the PLL differential equation, ($\Phi_m(t) \rightarrow -\Phi_m(t)$); and when the program has printed out the points, $\dot{x}_{ij}(0)$, $y_{ij}(0)$ and $\dot{y}_j(0)$, $(i, j = 1, \ldots 7)$ defining the surface of the spike region, take these points to be $-\dot{x}_{ij}(0)$, $y_{ij(0)}$, and $\dot{y}_j(0)$. For the no modulation case, this is equivalent to pointing out that the region (or regions) for $+2\pi$ spikes is simply a mirror image of those for -2π spikes, below the $y(0)$, $\dot{y}(0)$ plane.

To prove the above, consider that any PLL can be described by the equation

$$\dot{\Phi} = h(t) \circledast [-\sin(\phi - \phi_m) + x \cos\phi + y \sin\phi] \qquad (3.3.1)$$

where $h(t)$ is the impulse response of the PLL filter

$\circledast$ denotes convolution.

other terms defined in Sec. 1.4

If $\Phi(t)$ gives a positive spike at $t \approx 0$, then $-\Phi(t)$ gives a negative one. If $\Phi(t)$ is changed to $-\Phi(t)$ and $\Phi_m(t)$ to $-\Phi_m(t)$ in Eq. 3.3.1 then

$$\dot{\phi} = h(t) \circledast [-\sin(\phi - \phi_m) - x \cos\phi + y \sin\phi] \qquad (3.3.2)$$

Since $x(t)$ is modeled in the program as $\dot{x}(0)\, te^{-|t|}$, Eq. 3.3.2 is the same as Eq. 3.3.1 when $\dot{x}(0) \rightarrow -\dot{x}(0)$.

In the first block of the Flow Chart, our initial trial value for $\dot{x}_{ij}(0)$, $y_{ij}(0)$ and $\dot{y}_j(0)$ is inserted into Part (a) of the Program. The subscripts, i and j (used to enumerate the points to be determined on the spike boundary in $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ space) are both set equal to unity, and the variables

θ_h, VK, and ΔVK are given the numerical values shown. The use of these three variables (and their definition) will become evident later in the Program.

Block 2, comprises Part (a) of the Program described in the previous section. It supplies a steady-state value of the solution of the PLL differential equations to block 3, which then computes the Steady-State Phase-Error (SSPE) by subtracting the modulation phase from the solution.

In block 4, we determine whether or not this SSPE is less than $-\pi$ (indicating a negative phase error jump) or not.

If the SSPE is less than $-\pi$ we enter block 5b via B. If the SSPE has ever been greater than $-\pi$ since i last changed its value, we halve the hunting interval ΔVK, otherwise we double it. In other words, if we have crossed the -2π spike surface at least once during the hunting process to find the point $\dot{x}_{ij}(0)$, $y_{ij}(0)$, $\dot{y}_j(0)$ on the -2π spike surface, we halve the hunting interval ΔVK, otherwise we double it. This increases the hunting efficiency. VK is the net amount the hunting procedure has taken us from the trial value of $\dot{x}_{ij}(0)$, $y_{ij}(0)$, $\dot{y}_j(0)$. ΔVK is the hunting step, i.e., the distance we move from $\dot{x}_1, (0), y_1, (0), \dot{y}, (0)$ in one step of the hunting process. In many steps ΔVK sum to VK. VK is replaced by VK-ΔVK in this block.

If we had entered block 5a via A (because there was no negative phase jump in SSPE), it is again determined whether or not we have crossed the -2π spike surface at least once during the hunting process to find $\dot{x}_{ij}(0)$, $y_{ij}(0), \dot{y}_j(0)$ on the spike boundary. If so, halve ΔVK, if not, double it. In this block we <u>increase</u> VK by ΔVK.

From either block 5a or 5b, we next enter block 6. The first line of this block can be discussed better when discussing block 11. If this first

line does not result in an exit on C, we change the values of $\dot{x}_{ij}(0)$, and $y_{ij}(0)$ as indicated by the two equations * in the block. In Fig. 3.3.2, if the point labeled "1" is the old value of $\dot{x}_{ij}(0)$, $y_{ij}(0)$, ($\dot{y}_j(0)$ does not change during the hunting process) we hunt along the line with inclination θ_h, by amount VK. The point "2" would be the new value of $\dot{x}_{ij}(0)$, $y_{ij}(0)$. If ΔVK is not less than .01 (which does not occur until the hunting procedure for this point is complete) we exit on E to block 7b, where the new value of $\dot{x}_{ij}(0)$, $y_{ij}(0)$, $\dot{y}_j(0)$ are inserted into block 2 via F. The process repeats again from there.

The consecutive and cyclical operation of blocks 2, 3, 4, 5a or 5b, 6, and 7b, perform the core of the hunting procedure. For clarity, we shall discuss a typical example with the aid of Fig. 3.3.2. Three points on the spike boundary are shown. This boundary is indicated by the curved line, and spikes of area-2π occur for values of the parameters, $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ to the right. Suppose that the program is in the process of hunting for the point $\dot{x}_{2,4}(0)$, $y_{2,4}(0)$, $\dot{y}_4(0)$, along the line with slope θ_h, and the initial trial value is the point labeled "1". After completing blocks 2 an 3, we find that SSPE is nearly zero, (no spike is generated since we are to the left of the spike boundary) and therefore block 4 results in entering block 5a. Since SSPE has never been less than $-\pi$ since i last changed its value (from 1 to 2 since the previous point hunted was $x_{1,4}$ (0), $y_{1,4}(0)$, $y_4(0)$) we double ΔVK to .5, increase VK by ΔVK, and enter block 6. Since i is <u>less</u> than 3 and VK is <u>not</u> greater than .5, we do not

*Equations, in computer programs, of the form A=A+B mean that the value of A is changed to its previous value plus B.

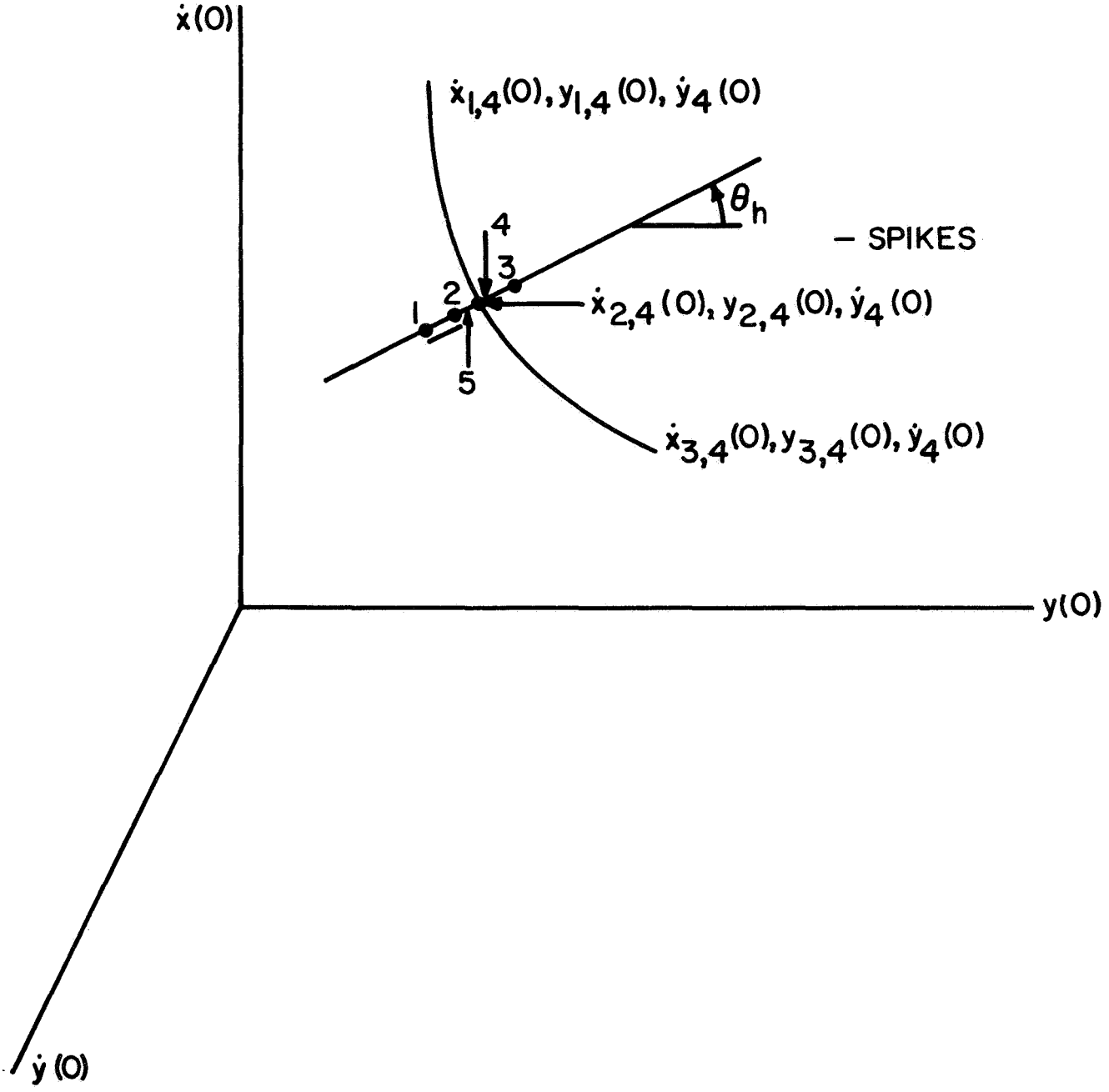


FIG. 3.3.2 SKETCH DEMONSTRATING HUNTING PROCEEDURE

exit on C, but execute the two equations, making the new values of $\dot{x}_{2,4}(0)$, $y_{2,4}(0)$, $\dot{y}_4(0)$ correspond to point "2" on Fig. 3.3.2. Since ΔVK is not less than .01, we exit on E, and insert these new values of $\dot{x}_{2,4}(0)$, $y_{2,4}(0)$, $\dot{y}_4(0)$ into block 2 via F. We go through the same blocks (2, 3, 4, 5a, 6, 7b), doubling ΔVK to 1 and increasing VK by ΔVK in block 5a. Block 6 makes the values of $\dot{x}_{2,4}(0)$, $y_{2,4}(0)$, $\dot{y}_4(0)$ corresponding to point "3" in Fig. 3.3.2. The next time we go through this loop, we enter block 5b (since SSPE is now about -2π or less than $-\pi$). Since we have crossed the spike boundary, (SSPE was greater than $-\pi$ at points "1" and "2"), we halve ΔVK to .5, decrease VK by ΔVK, and block 6 takes us to point "4". Each time we go through the above loop, we will "hop" back and forth by ΔVK, halving ΔVK each time, until ΔVK is less than .01. We then exit block 6 at D, and enter block 7a. When this happens, $\dot{x}_{2,4}(0)$, $y_{2,4}(0)$, $\dot{y}_4(0)$ is within .01 of the actual spike boundary.

Block 7a, writes out the values of the co-ordinates of this point, or, in general, the $\dot{x}_{ij}(0)$, $y_{ij}(0)$, $\dot{y}_j(0)$ supplied by the completion of the above hunting procedure.

The purpose of the rest of the program(except block 10) is to use points previously found by the hunting procedure, to estimate a better trial value for the subsequent point to be hunted.

Block 8, (for a reason which will become evident later) steers the progress of the program through exits G,H, I, or J depending on whether $i = 1$, or 2, or $3 \leq i < 9$, or $i = 9$ respectively. It also exits at J if either $\dot{x}_{ij}(0)$ or $y_{ij}(0)$ is greater than 6.

If $i=1$, we enter block 9a through G. Here the test value for the new point to be hunted, $\dot{x}_{2,j}(0)$, $y_{2,j}(0)$, $\dot{y}_j(0)$, is taken to be $\dot{x}_{1,j}(0) - .5$, $y_{1,j}(0)$, $y_j(0)$ ($\dot{y}_j(0)$ not changed). Since only one point has been found by the hunting

procedure, increasing $\dot{x}(0)$ by .5 and not changing y(0) is about the best new trial value to take. The subscript i is increased to 2, and we set VK=0, and $\Delta VK = .25$ in preparation for a new hunting procedure. Leaving block 9 a, the new trial value from block 7b goes, via Exit F, into block 2, and a new hunting procedure begins.

Returning to block 8, if i is equal to 2, (which means that two points with the same $\dot{y}_j(0)$ have been obtained) we enter block 9 b via exit H. Here, a straight line extension is used, as described in block 9 b, Fig. 3.3.1, to obtain a better estimate for the trial value, $\dot{x}_{i+1,\,j}(0)$, $y_{i+1,\,j}(0)$ from which to start hunting. Since it is desired to hunt in a direction approximately normal to the spike boundary, θ_h is set equal to the slope of this line of extension, rotated by a right angle. VK is set equal to zero and i incremented by 1, but ΔVK is set to 0.2 instead of 0.25, since the trial value is better (closer to the true final value of $\dot{x}_{i+1,\,j}(0)$, $y_{i+1,\,j}(0)$, $\dot{y}_j(0)$ on the spike boundary). The program proceeds to block 7 b again and the new trial value inserted into block 2 via exit F.

Returning again to block 8, if $3 \leq i < 9$, we exit on I into block 11. Here, a trial value for $\dot{x}_{i+1,\,j}(0)$, $y_{i+1,\,j}(0)$ is estimated by using a circular extension from the previous three points found by the hunting procedure. Note that the length of the arc of extension is proportional to the number EX, which is specified by the first line in block 6. What this first line does is as follows. Normally EX is unity. However, if , during the hunting process VK becomes greater than 0.5 while i>3 (the circular extension in block 11 is only used when $i \geq 3$, but it increases i by 1 so that if i was 3 it now becomes 4), EX is set equal to 0.5, i decreased by one and the program exits block 6 via exit C into block 11 where the trial value for $\dot{x}_{i+1,\,j}(0)$, $y_{i+1,\,j}(0)$ is recalculated using the smaller value of EX.

In other words, if VK becomes large, that indicates that the initial trial value is not close to the boundary. The hunting procedure is abandoned, and a new trial value is computed using a smaller extension. The program is started over again. This procedure saves computer time, since estimating trial values via a circular extension (block 11) involves mere arithmetic, while the hunting procedure involves a repetitive series of solutions, (with different parameters) of a PLL differential equation, over a fixed time interval. Block 11, finally, sets θ_h equal to the slope of a tangent to the circular arc at $\dot{x}_{i+1,j}(0)$, $y_{i+1,j}(0)$ rotated by a right angle. It also sets VK = 0, i = i + 1, and $\Delta VK = 0.125$. (ΔVK is still smaller than that set by block 9b, since the circular extension is usually better than a straight line extension). Notice that if the curvature is very small, (indicating that the three points lie on a straight line) we exit block 11 at M and use the straight line extension method of block 9b.

Finally, if we are in block 8 and i = 9, we exit via exit J into block 10. If i = 9, this means that we have discovered 9 values $\dot{x}_{ij}(0)$, $y_{ij}(0)$, $\dot{y}_j(0)$; i = 1, 2, . . . 9 on the spike boundary. We usually will enter block 10 from block 8 with less than 9 points, because of the statement "Exit J if $|\dot{x}_{ij}(0)|$ or $|y_{ij}(0)| > 6$ in block 8"

The latter statement in block 8, stops the hunting procedure (for a particular value of $\dot{y}_j(0)$) for values of $\dot{x}_{ij}(0)$ and for $y_{ij}(0)$ greater than 6 in magnitude, since we are not interested in portions of the spike boundary that remote. Note that all these points share the same $\dot{y}_j(0)$ co-ordinates. Block 10, sets a new value for $\dot{y}_j(0)$ (increasing subscript j by 1) and restarts the hunting procedure to find a new set of at most 9 points on the spike boundary

$$\dot{x}_{i,j+1}(0),\ y_{i,j+1}(0),\ \dot{y}_{j+1}(0);\ i = 1, 2, \ldots \leq 9.$$

Seven values of $\dot{y}_j(0)$ are presented to the program, evenly spaced by .8, starting from -2.4 and continuing to +2.4. The program then exits on K and stops. A total of at most 63 (9*7) points, on the spike boundary are printed out by the program. Less than 63 points are presented, if the program has ever exited block 8, due to the last statement in the block, which is usually the case.

A sample program, using Fortran IV language is presented in Appendix C.

3.4 Program to Find the Expected Number of Spikes, Given the "Spike Regions"

The expected number of times that the Gaussian random vector $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ enters spike regions is (from Rice[1.9], see also Appendix E)

$$N = \int_S |\dot{x} + \dot{\phi}_m y| f_{x\dot{x}y\dot{y}}(x=0, \dot{x}, y, \dot{y})\, d\dot{x}\, dy\, d\dot{y} \qquad (3.4.1)$$

where $f_{x\dot{x}y\dot{y}}(x\dot{x}y\dot{y})$ is the joint Gaussian density of $x(0)$, $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$, N is the expected number of times the vector $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ enters the region (or regions) S.

$\dot{\phi}_m = \dot{\phi}_m(0)$ is the instantaneous frequency deviation (radians/sec) at t=0.

If S consists of the spike regions, determined by the program described in Secs. 3.1 and 3.2, then N is the expected number of spikes per second.

The spike boundary (or boundaries) are plotted (see Secs. 4.2, 4.3, 4.4) and approximated by plane segments as shown in Fig. 3.4.1 (the shaded portion represents a "vertical" drop or cliff). This mode of approximation is equivalent to segmenting the spike surface with planes parallel to the $\dot{x}(0)$, $y(0)$ plane. The shaded portion in Fig. 3.4.1 lies on one of these segmenting planes. A typical plane passes half way

between the $\dot{y}_j(0)$ and $\dot{y}_{j+1}(0)$ supplied by Parts (a) and (b) of the Program (see Secs. 3.1, and 3.2). The spike surface is assumed to be independent of $\dot{y}(0)$ on any of these planes (this is equivalent to a "step" approximation along $\dot{y}(0)$). Then the best straight line approximation is used for the variation of the y(0) co-ordinate of the spike boundary with $\dot{x}(0)$.

If spikes of area- 2π occur for any $\dot{x}(0)$, y(0), $\dot{y}(0)$ to the right of the spike surface, (a typical portion of which is approximated by plane segments in Fig. 3.4.1) then to find the expected number of spikes of area -2π, we can evaluate the integral in Eq. 3.4.1 for the area to the right of each of the plane segments, then add these results to get the total expected number of negative spikes.

For the upper right segment in Fig. 3.4.1,

$$y(0) = A*\dot{x}(0) + B \tag{3.4.2}$$

where $D = \dot{x}_{i+1,\,j}(0) - \dot{x}_{ij}(0)$

$A = (y_{i+1,\,j}(0) - y_{ij}(0))/D$

$B = (\dot{x}_{i+1,\,j}(0)*y_{ij}(0) - \dot{x}_{ij}(0)*y_{i+1,\,j}(0))/D$

Referring to Sec. 2.2, x(0), $\dot{x}(0)$, y(0), $\dot{y}(0)$ are each independent, and have the same variance, σ^2. Thus, their joint density function is

$$f_{x\dot{x}\,y\dot{y}}(x\dot{x}\,y\dot{y}) = \frac{1}{(2\pi\,\sigma^2)^2}\; e^{-\frac{x^2+\dot{x}^2+y^2+\dot{y}^2}{2\sigma^2}} \tag{3.4.3}$$

The integral in Eq. 3.4.1 for this plane segment becomes

$$N_{ij} = \frac{1}{(2\pi\,\sigma^2)^2} \int_{\dot{y}_j(0)-\Delta\dot{y}_-}^{\dot{y}_j(0)+\Delta\dot{y}_+} d\dot{y}\; e^{-\dot{y}^2/2\sigma^2} \int_{\dot{x}_{ij}(0)}^{\dot{x}_{i+1,\,j}(0)} |\dot{x}+\dot{\phi}_m(0)y|\, d\dot{x}\; e^{-\dot{x}^2/2\sigma^2} \int_{A\dot{x}(0)+B}^{\infty} dy\; e^{-y^2/2\sigma^2} \tag{3.4.4}*$$

* $\Delta\dot{y}_+$ and $\Delta\dot{y}_-$ defined in Fig. 3.4.1

This integral can be evaluated in closed form. The computation is however, time consuming*, since $\dot{x}_{i+1,j}(0)$ can be positive while $\dot{x}_{ij}(0)$ is negative, thus requiring the absolute value on $\dot{x}$ in the integral to be contended with. The interested reader will find the details in the sample program in Appendix D.

The co-ordinates of the corners of all the approximating plane segments are read into the program in Appendix D. The value for N_{ij} (Eq. 3.4.4) that results when the integral is performed, for each i and j is then computed. Adding all of the N_{ij}'s together, the expected number of spikes per second, is obtained.

The carrier-to-noise ratio (CNR) is $1/2\sigma^2$ since the carrier has a magnitude of unity. N is a function of this CNR (Eq. 3.4.4 depends upon $1/2\sigma^2$), and the program is made to print out N as a function of CNR. A Flow Chart of this program is presented in Fig. 3.4.2.

*It took 16 Fortran statements and 4 Subroutines to express that integral in the program.

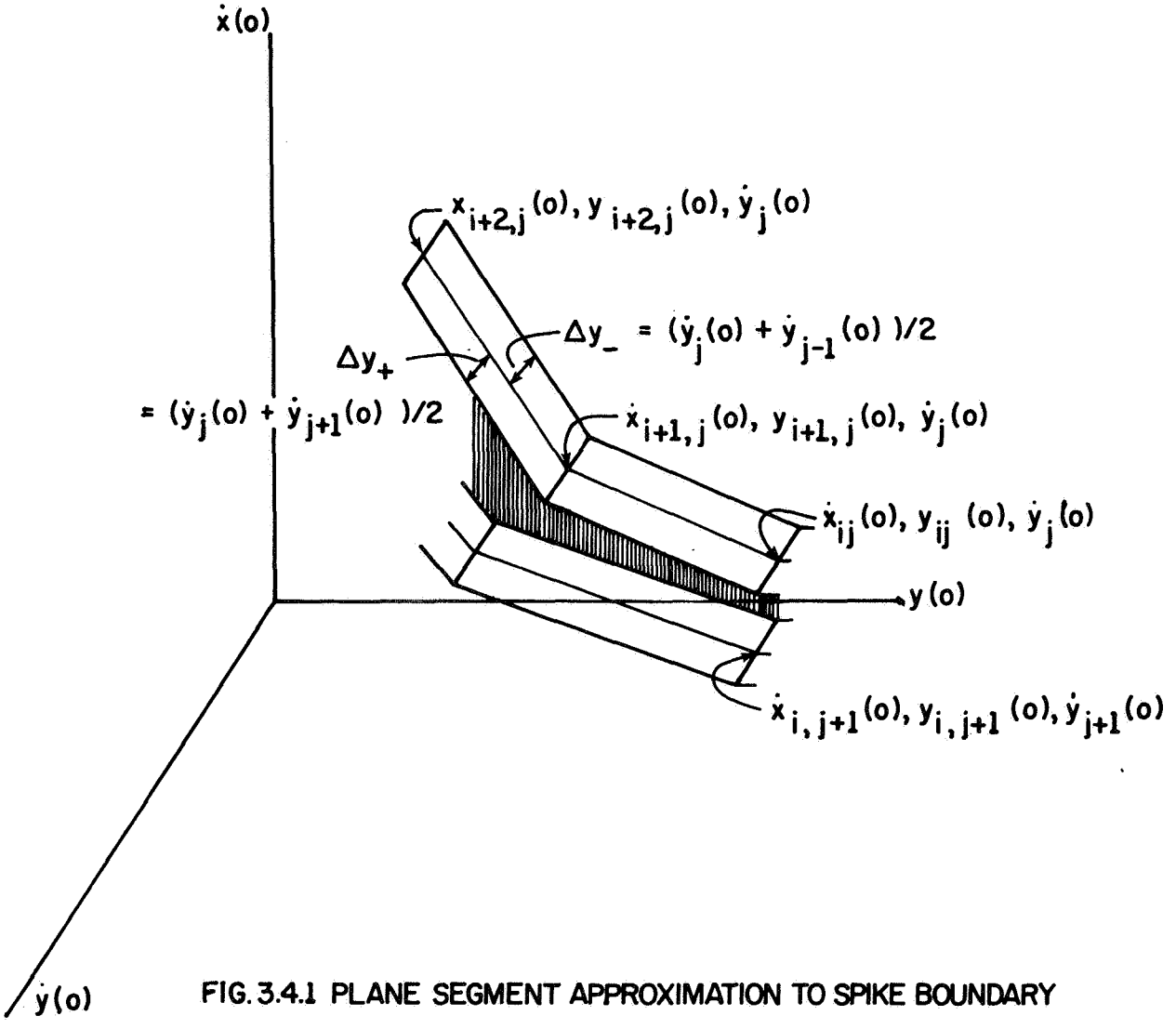
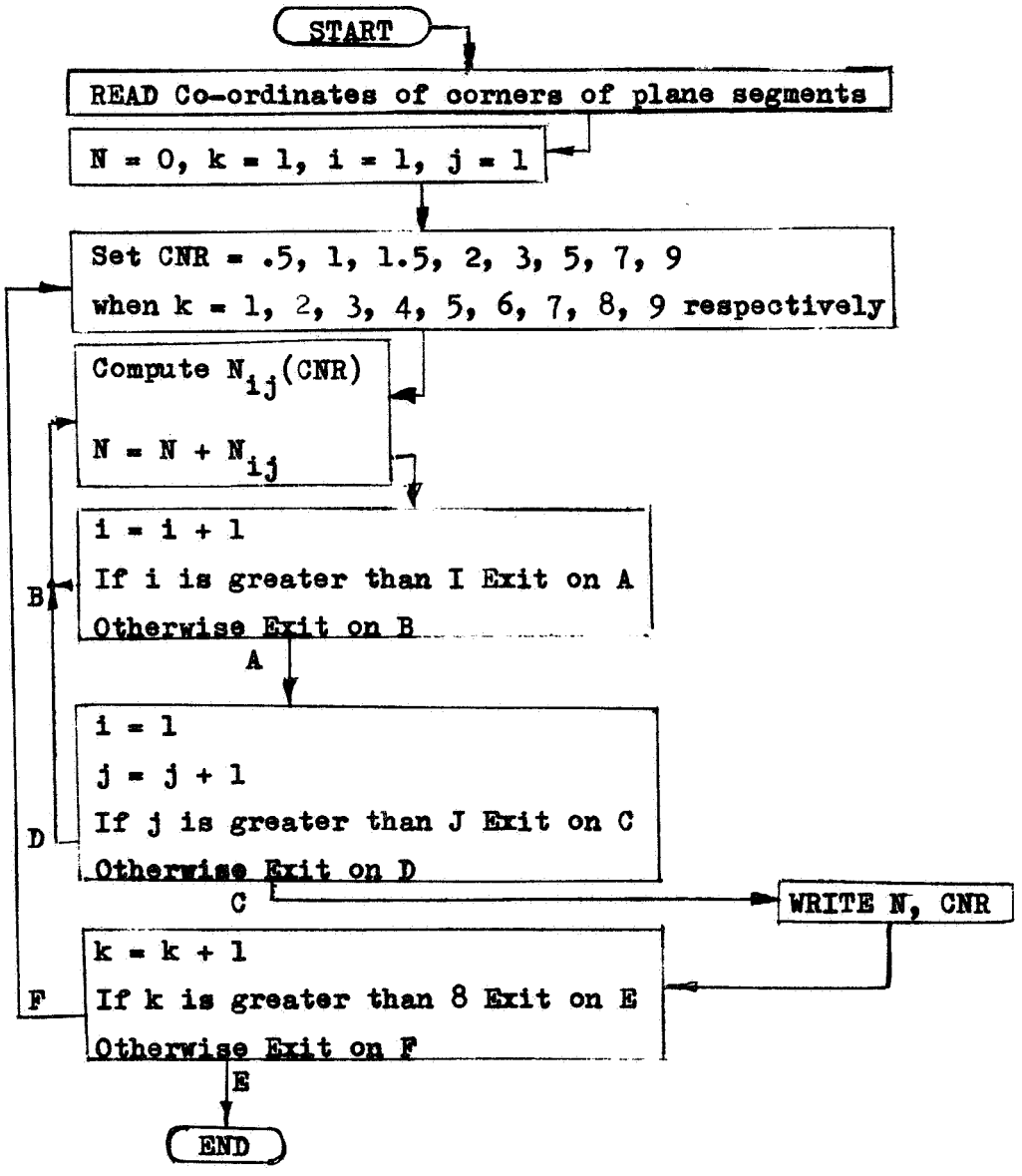


FIG. 3.4.1 PLANE SEGMENT APPROXIMATION TO SPIKE BOUNDARY

FIG. 3.4.2

Flow Chart to Estimate Expected Number of spikes per Second



CHAPTER 4

COMPUTER RESULTS (SPIKE BOUNDARIES)

4.1 FM Discriminator

The spike boundary for the FM discriminator, with a constant offset modulation, is shown in Fig. 4.1.1. It was computed by replacing Part (a) of the Program of Sec. 3.2 with a program which determined the steady state value of the arctan (.)term of equation 1.4.1, at the end of the -5 to + 25 second time interval.

Only the surface for positive $\dot{y}(0)$ is shown, because the surface is symmetrical in $\dot{y}(0)$. It is interesting to note that negative spikes occur not only to the right of the surfaces, but also in a separate region to the left below the $\dot{y}(0)$, $y(0)$ plane. The fact that no regions for positive spikes are shown does not exclude, entirely, the possibility of their occurrence, since there might be a positive spike region in the region $\dot{x}(0) < 3$. This region was not considered, as the number of spikes contained therein must be extremely small.

If there is no modulation, the spike boundary of the FMD is simply the $y(0) = 1$ plane. Any $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ to the right of that plane will result in a phase jump. That phase jump is -2π above the $y(0)$, $\dot{y}(0)$ plane and $+2\pi$ below it. When Eq. 3.4.1 is applied to these regions for the FMD without modulation, the standard result for expected number of spikes per second for FMD due to Rice[1.9] is obtained. $[(1/2\pi)\,\mathrm{erfc}(\sqrt{CNR})]$.

When Eq. 3.4.1 (integration using spike boundary to obtain expected number of spikes per second, N) is applied to the spike surfaces in Fig. 4.4.1, for FMD with modulation, we obtain a slightly higher expected number of spikes per second, versus input carrier-to-noise ratio (CNR), than Rice's result gives for FMD with modulation. (See Figs. 5.4.1a and 5.4.1b.)

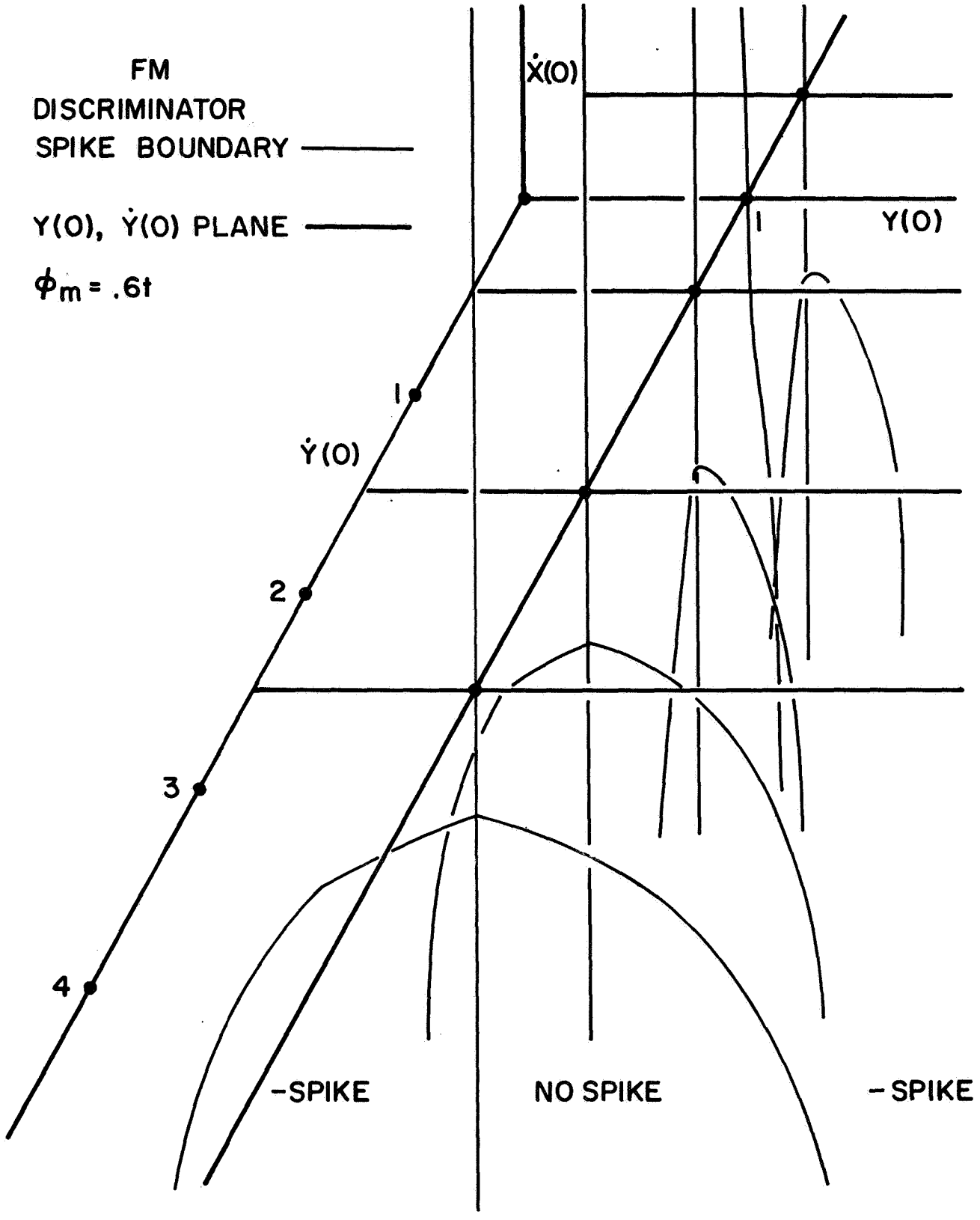

FM
DISCRIMINATOR
SPIKE BOUNDARY
Y(0), Ẏ(0) PLANE
$\phi_m = .6t$
Ẋ(0)
Y(0)
Ẏ(0)
1
2
3
4
-SPIKE
NO SPIKE
-SPIKE

FIG. 4.1.1

With a CNR of 10 (10db) Rices' result for N is 4.8×10^{-6} spikes/sec, while the most-likely noise gives 6×10^{-6} spikes/sec, a negligible difference. The conclusion to be made here, is that the results obtained for the FMD, using the Most-Likely-Noise approach, essentially agree with those obtained by Rice, with or without modulation.

4.2 First Order PLL

4.2.1 First Order PLL Without Modulation

The spike boundary for the first order PLL without modulation is shown in Fig. 4.2.1. The loop had a 3db bandwidth of 5 radian/second. (The results, in terms of expected number of spikes per second versus CNR, are presented in normalized form, in Figs. 5.4.1a and 5.4.1b. They therefore can be employed for all loop bandwidths, having the same loop bandwidth to IF bandwidth ratio, as the loops used in the computer program.) Only the surface for -2π spikes is shown. They occur to the right of the curved boundary. The boundary for -2π spikes is simply a mirror image of the one shown, below the $y(0)$, $\dot{y}(0)$ plane. The boundary was found to be symmetric in $\dot{y}(0)$, thus the Fig. 4.2.1 shows the boundary only for positive $\dot{y}(0)$.

Loops of higher and lower gain, were also tried. The spike boundaries for all of them had the same shape as the one in Fig. 4.2.1, however. The boundary hugs the plane $y(0) = 1$, $\dot{x}(0) > 0$, and the $\dot{x}(0) = 0$ plane more closely, as the gain is increased, thus converging to the spike boundary for FMD without modulation. The reverse occurs when the gain is reduced, i.e. the boundary moves further away, to the right. This indicated, that the loop with lower gain has fewer expected number of spikes per second with a given CNR, since the vector $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$ has extend, further from the origin in order to enter the spike regions. Indeed, the performance of different

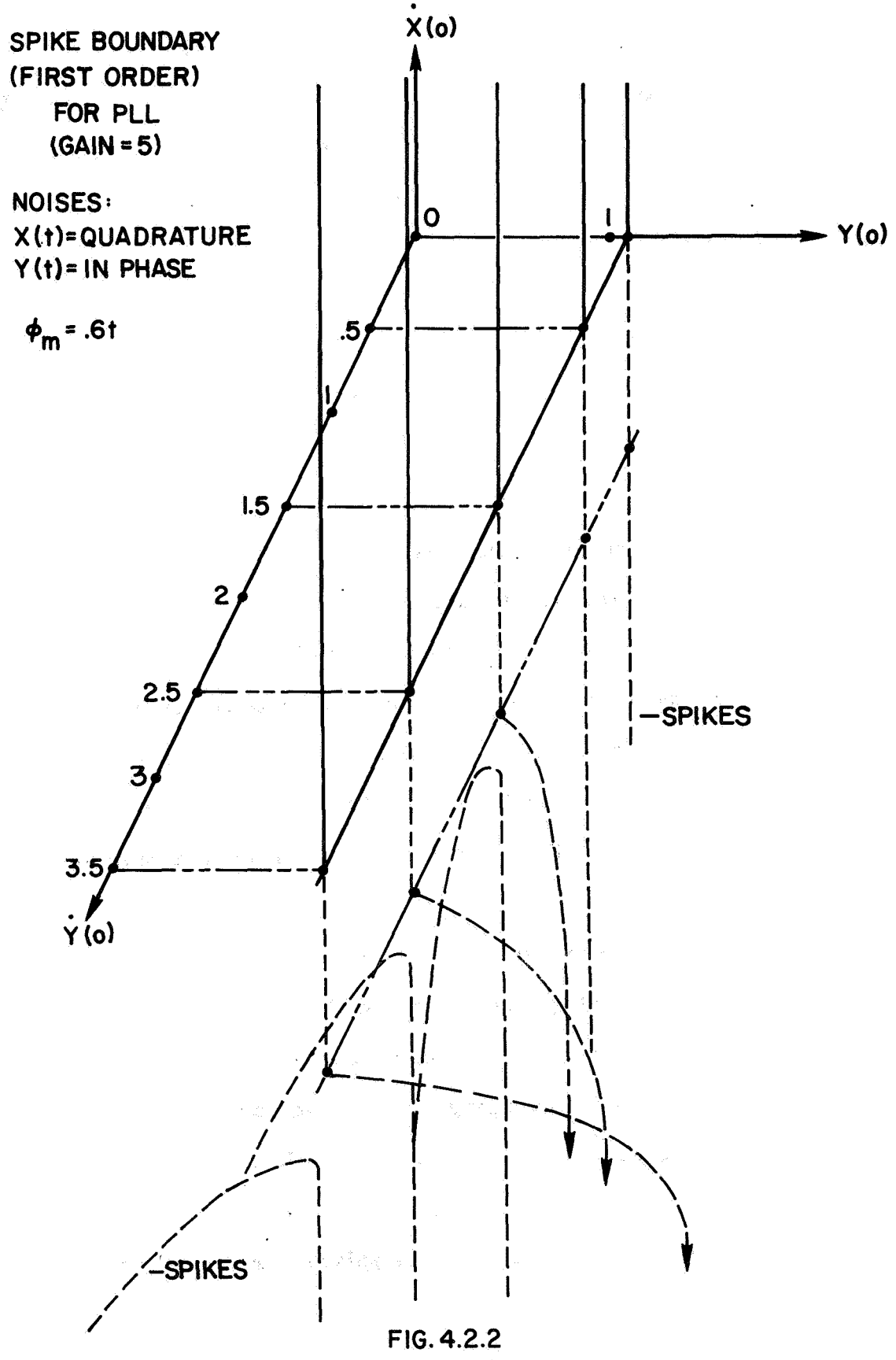

SPIKE BOUNDARY
(FIRST ORDER)
FOR PLL
(GAIN = 5)
NOISES:
X(t) = QUADRATURE
Y(t) = IN PHASE
$\phi_m = .6t$
$\dot{X}(o)$
$\dot{Y}(o)$
Y(o)
0
1
.5
1
1.5
2
2.5
3
3.5
SPIKES
SPIKES

FIG. 4.2.2

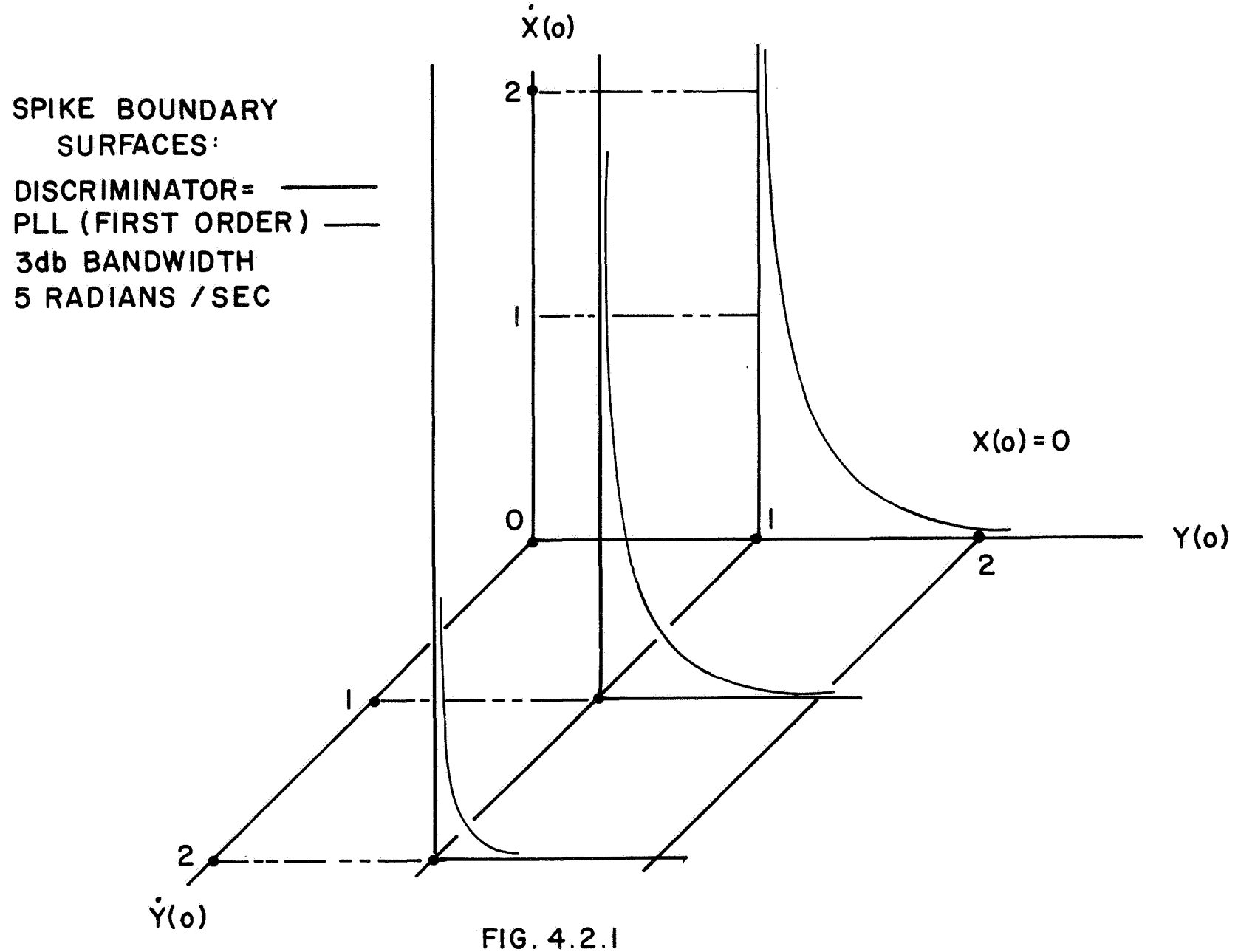

SPIKE BOUNDARY
SURFACES:
DISCRIMINATOR=
PLL (FIRST ORDER)
3db BANDWIDTH
5 RADIANS / SEC
Ẋ(o)
2
1
0
X(o) = 0
1
2
Y(o)
1
2
Ẏ(o)

FIG. 4.2.1

PLL's can be quickly compared, by comparing this aspect of their spike regions. The Expected number of Spikes per second for this loop is plotted in Fig. 5.4.1a and will be discussed in Chapter 5.

4.2.2 First Order PLL with Modulation

The spike boundary for the first order PLL with a constant offest modulation is shown in Fig. 4.2.2. The resulting boundary is very similar to that for the FMD with the same modulation (Fig. 4.1.1). This is as to be expected, since a first-order loop with a high gain (5 radians/second bandwidth, compared to a +.643 radians/second IF bandwidth) has a performance not too different from the FMD. As was the case for the FMD, there are no positive spike regions in Fig. 4.2.2. A detailed discussion of the expected number of spikes per second, and a comparison with experimental results is presented in Chapter 5.

4.3 Second Order PLL

4.3.1 Second Order PLL without Modulation

Figs. 4.3.1a and 4.3.1b show the spike boundaries for a second order, constant-plus-integral PLL with the phase-error transfer function cut-off frequencies at .91 and .56 radians/second respectively. The PLL was designed to have a maximally flat phase-error transfer. In both cases, only the spike boundary for spikes of -2π area are shown since the boundary for $+2\pi$ spikes is a mirror image, with respect to the $y(0)$, $\dot{y}(0)$ plane.

Comparing the two surfaces, we can judge that the lower-gain loop has fewer spikes per second, given the same CNR, since its spike boundary is further from the origin.

It is interesting to note, that the spike surface for the loop of gain .91, crosses to the left of the $y(0) = 1$, plane for $\dot{x}(0) > 2$. For $\dot{x}(0)$, $y(0)$, $\dot{y}(0)$'s in this region, we obtain negative spikes from the PLL that do not occur in

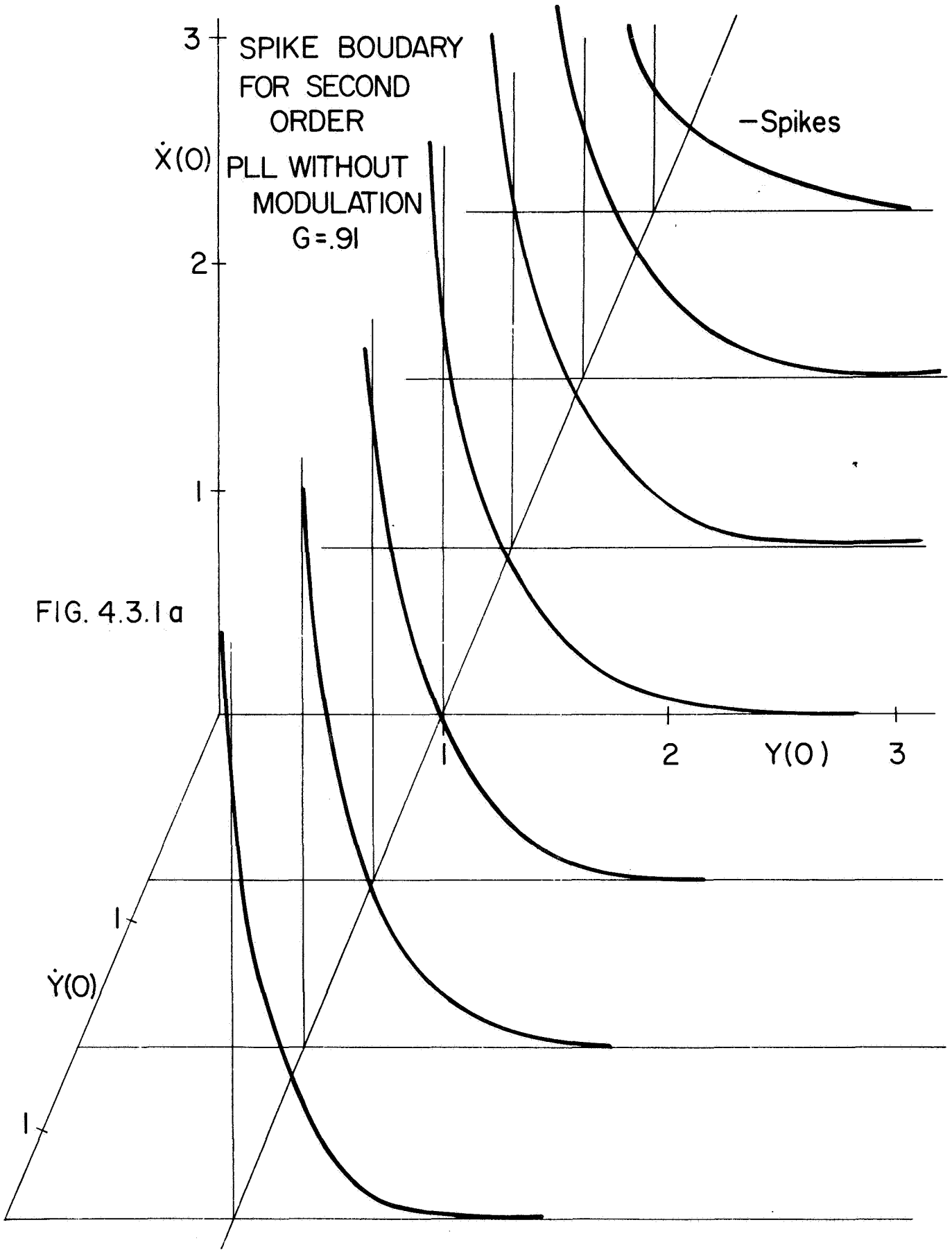

SPIKE BOUDARY
FOR SECOND
ORDER
PLL WITHOUT
MODULATION
G=.91
—Spikes
3
2
1
Ẋ(0)
Ẏ(0)
1
2
3
Y(0)
1
1

FIG. 4.3.1a

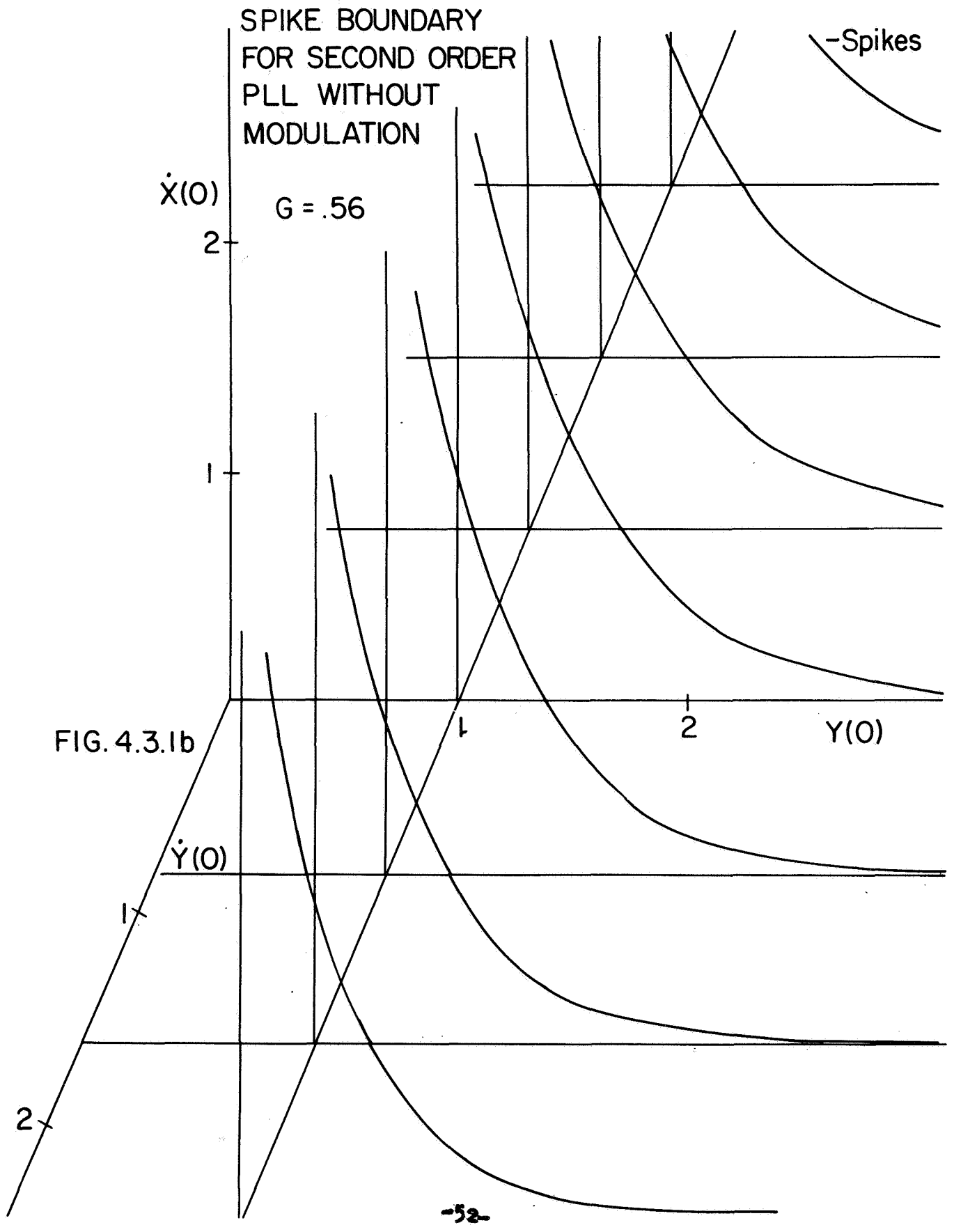

SPIKE BOUNDARY
FOR SECOND ORDER
PLL WITHOUT
MODULATION
G = .56
-Spikes
Ẋ(0)
2
1
1
2
Y(0)
Ẏ(0)
1
2

FIG. 4.3.1b

the FMD. The spike boundary for the loop of gain .56 also illustrates this phenomenon for large $\dot{x}(0)$, when the surface in Fig. 4.31b is extended in the positive $\dot{x}(0)$ direction. It is conjectured, that the above phenomenon takes place when an input phase spike (causing a doublet FMD output) in the noise trajectory, causes the PLL to lose lock and execute a phase jump of 2π.

4.3.2 Second Order PLL with Modulation

Figs. 4.3.2 show the spike boundaries for the constant-plus-integral PLL for a modulation index of 3 and 12 respectively.* The PLL used with a modulation index of 3 has a phase-error transfer function cut-off frequency of .91 radians/second, while when $\beta = 12$, the cut-off frequency was .56 radians/second. These two phase-error cut-off frequencies were chosen so that when the higher gain loop (.91) is driven with a modulation phase of $3 \sin(\omega_m t)$, it develops the same magnitude of phase error as the lower gain loop (.56) does, when it is driven with a modulation phase of $12 \sin(\omega_m t)$, $(\omega_m = [\omega_{IF}/2]/[\beta+1])$. A constant offset modulation of $\beta\omega_m$ is used when determining the spike boundaries. This is equal to the peak value of frequency deviation with a sine wave modulation of modulation index β.

The spike surface is of a much simpler form, than the one for the FMD or first-order PLL with modulation. Any region for positive spikes, it it exists, is outside the range of view of Figs. 4.3.2a or 4.3.2b.

Here again, it is possible to obtain spikes out of the PLL, that do not occur with a FMD. Remember, that in Fig. 4.1.1 for the FMD spike boundary with modulation, that there is a region, inside the lower right

---

*The constant offset modulation used is $\beta\omega_m$ radians/ sec., where $\omega_m = \Delta\omega/(\beta+1)$, $\Delta\omega = \omega_{IF}/2$. The loop gain is adjusted so that with a modulation of $\beta \sin(\omega_m t)$ the amplitude of the phase error is .1.

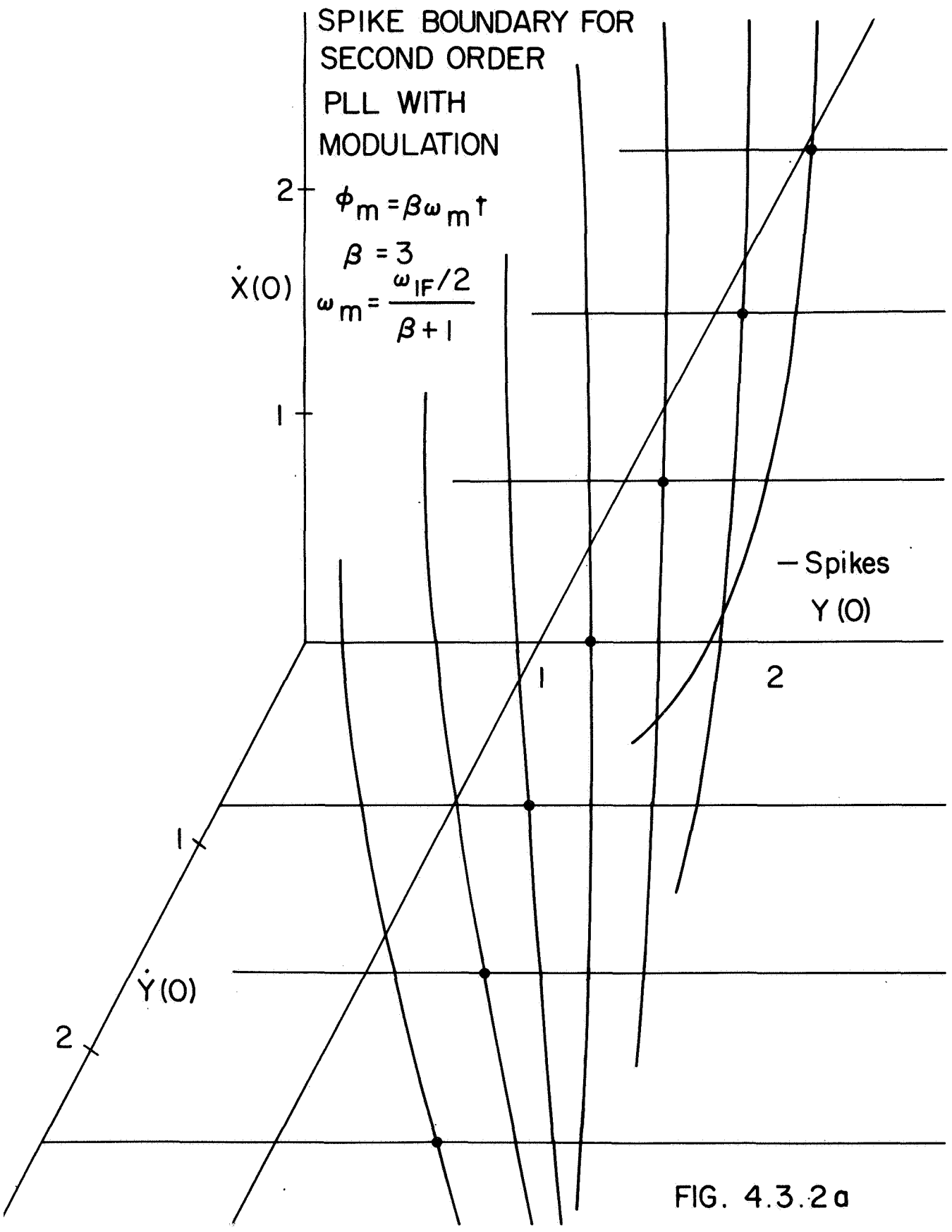

SPIKE BOUNDARY FOR
SECOND ORDER
PLL WITH
MODULATION
$\phi_m = \beta \omega_m t$
$\beta = 3$
$\omega_m = \frac{\omega_{IF}/2}{\beta + 1}$
$\dot{X}(0)$
2
1
Spikes
Y (0)
1
2
1
$\dot{Y}(0)$
2

FIG. 4.3.2a

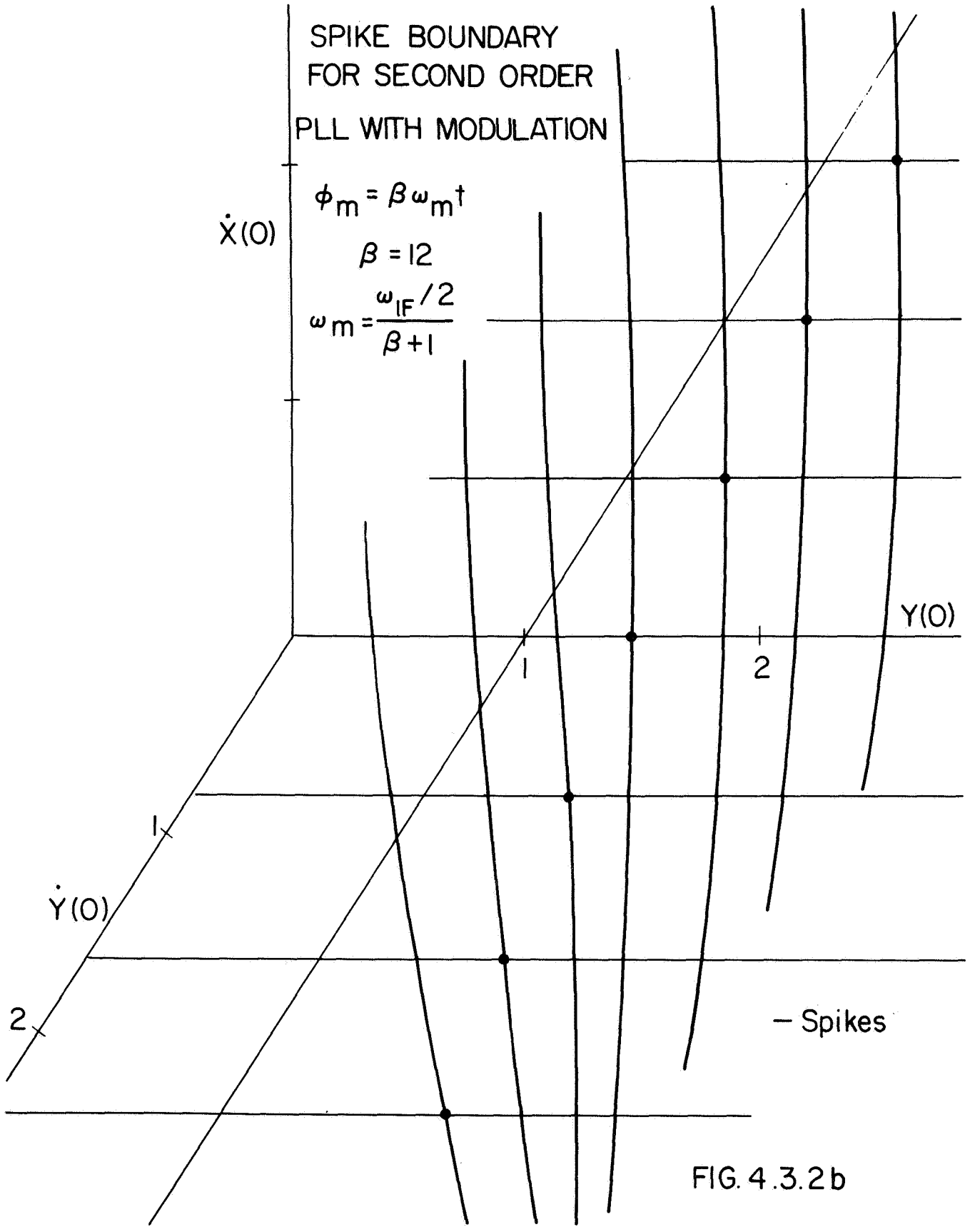

SPIKE BOUNDARY
FOR SECOND ORDER
PLL WITH MODULATION
$\phi_m = \beta \omega_m t$
$\beta = 12$
$\omega_m = \frac{\omega_{IF}/2}{\beta+1}$
$\dot{X}(0)$
$Y(0)$
1
2
$\dot{Y}(0)$
1
2
– Spikes

FIG. 4.3.2b

"hump", below the y(0), $\dot{y}(0)$ plane and slightly to the right of the y(0)=1 plane, where no spikes occur. This same region, in Figs. 4.3.2a and 4.3.2b does indicate negative spikes, especially in the negative $\dot{y}(0)$ side (remember that the FMD boundary is symmetrical in $\dot{y}(0)$).

The expected number of spikes/ second, derived from these surfaces is shown in Figs. 5.4.1a and 5.4.1b and will be discussed in Chapter 6.

4.4 Third Order PLL

4.4.1 Third Order PLL without Modulation

The spike boundaries for two third order constant-plus-integral-plus double-integral PLL's are shown in Figs. 4.4.1a and 4.4.1b. The PLL's have a phase-error transfer-function cut-off frequency of .512 radians/sec. and .25 radians/sec.

Both spike boundaries have roughly the same shape as those for the other PLL's, however, they are further from the origin, indicating a lower spike generation. The lower gain loop (Fig. 4.4.1b) is further from the origin than the higher gain one (Fig. 4.4.1a) indicating that a lower gain loop has less spikes/sec. than the higher gain loop.

4.4.2 Third Order PLL with Modulation

The Third-order PLL with a phase error transfer-function cut-off frequency of .512 radians/sec. when driven with a modulating phase of $3 \sin \omega_m t$ developed the same phase error magnitude as the PLL with a phase-error transfer function cut-off of .25 radians/sec. driven with a $12 \sin \omega_m t$ modulating phase ($\omega_m = .5 * \omega_{IF}/(\beta + 1)$). Incidentally this phase error magnitude is about .1 and is the same as that for both second order PLL's as well.

Figs. 4.4.2a and 4.4.2b show the spike boundaries for the two third order PLL's with a constant offset modulation of $\beta\omega_m$. These surfaces

are further from the origin than the corresponding ones for the second-order PLL's. Also the spike surface of the lower-gain third-order PLL (G=.25) is further out than that of the higher gain loop (G=.512). Again, no regions for positive spikes are shown.

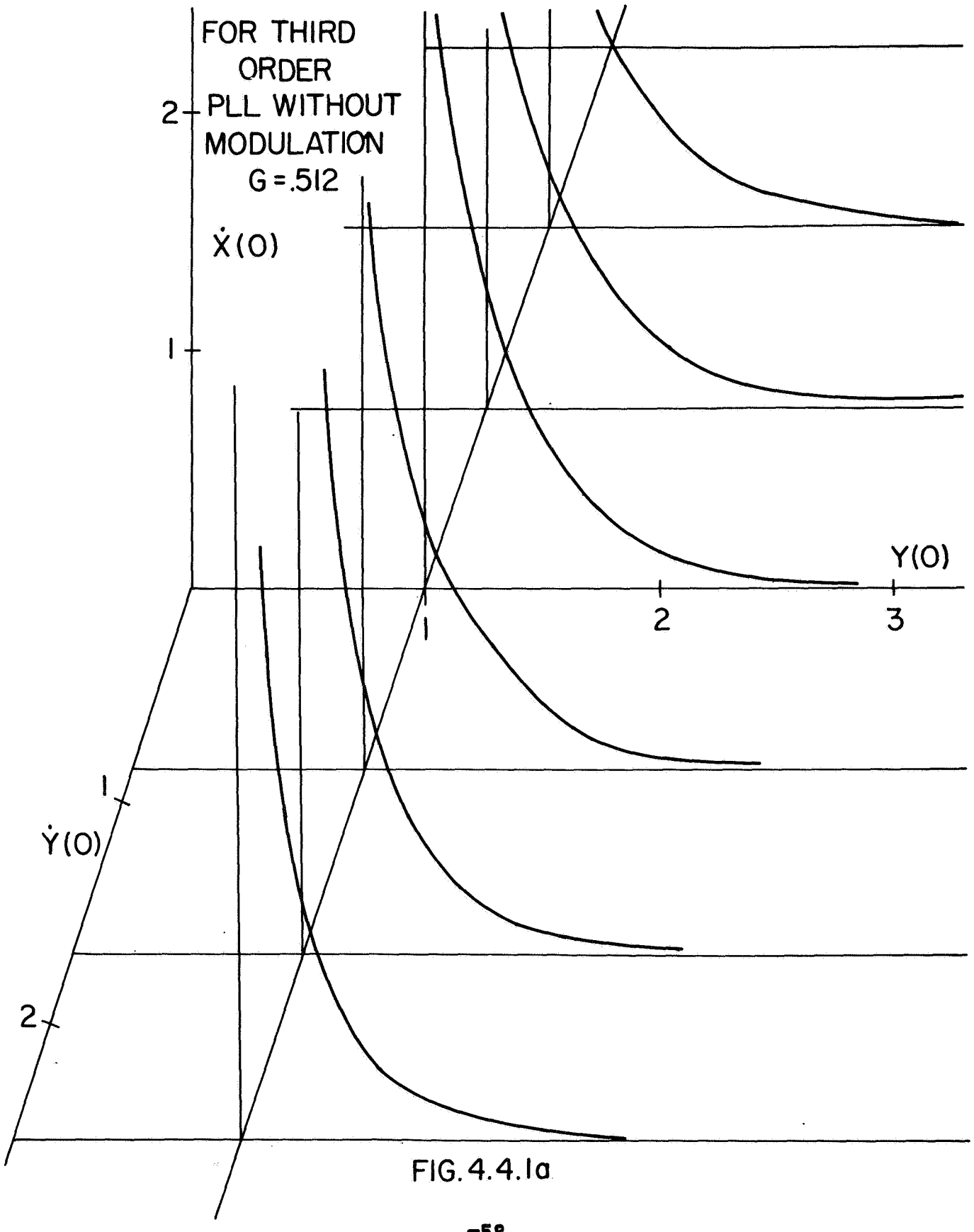

FOR THIRD
ORDER
PLL WITHOUT
MODULATION
G = .512
2
1
Ẋ(0)
Y(0)
1
2
3
1
Ẏ(0)
2

FIG. 4.4.1a

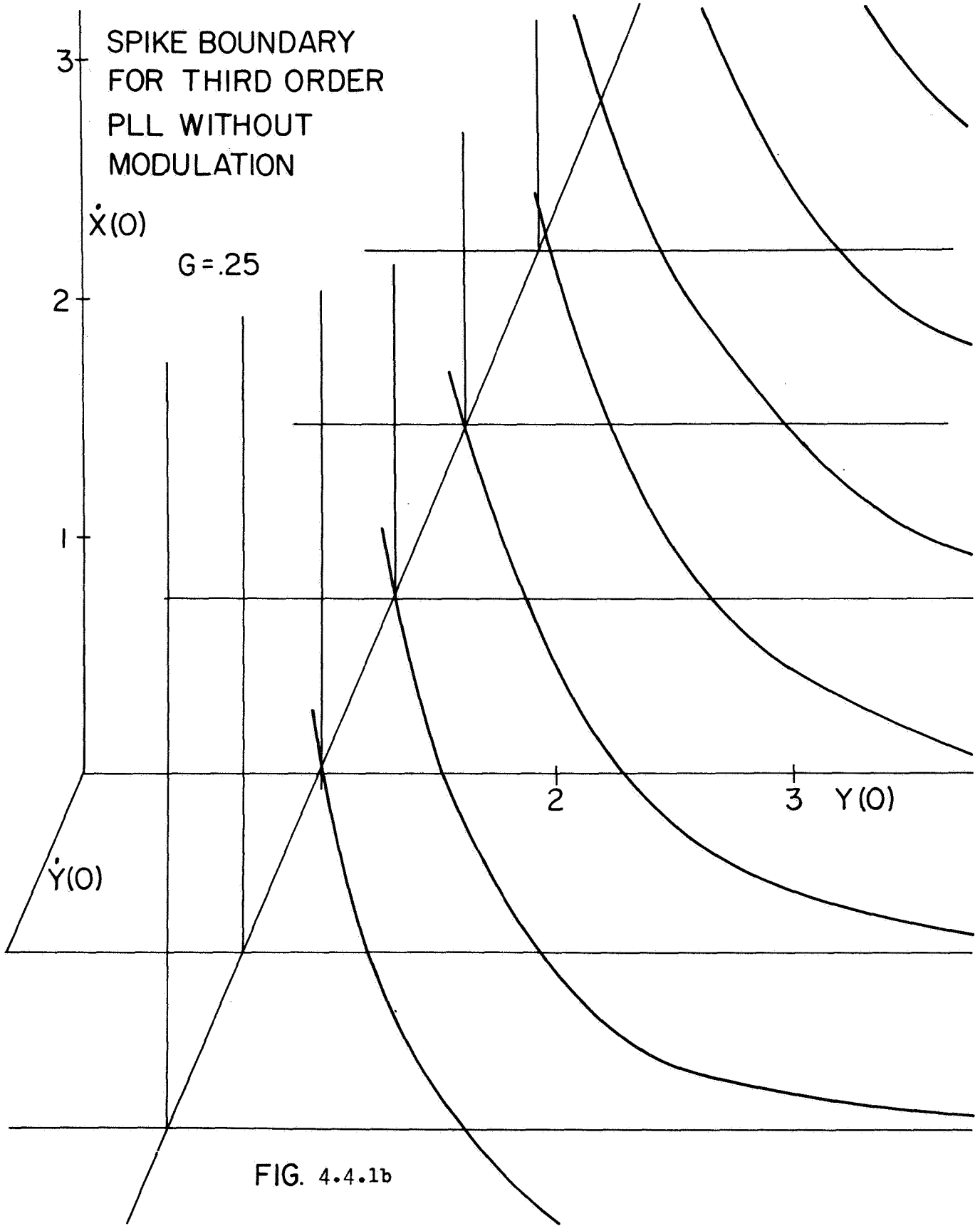

SPIKE BOUNDARY
FOR THIRD ORDER
PLL WITHOUT
MODULATION
$\dot{X}(0)$
G = .25
3
2
1
2
3
Y(0)
$\dot{Y}(0)$

FIG. 4.4.1b

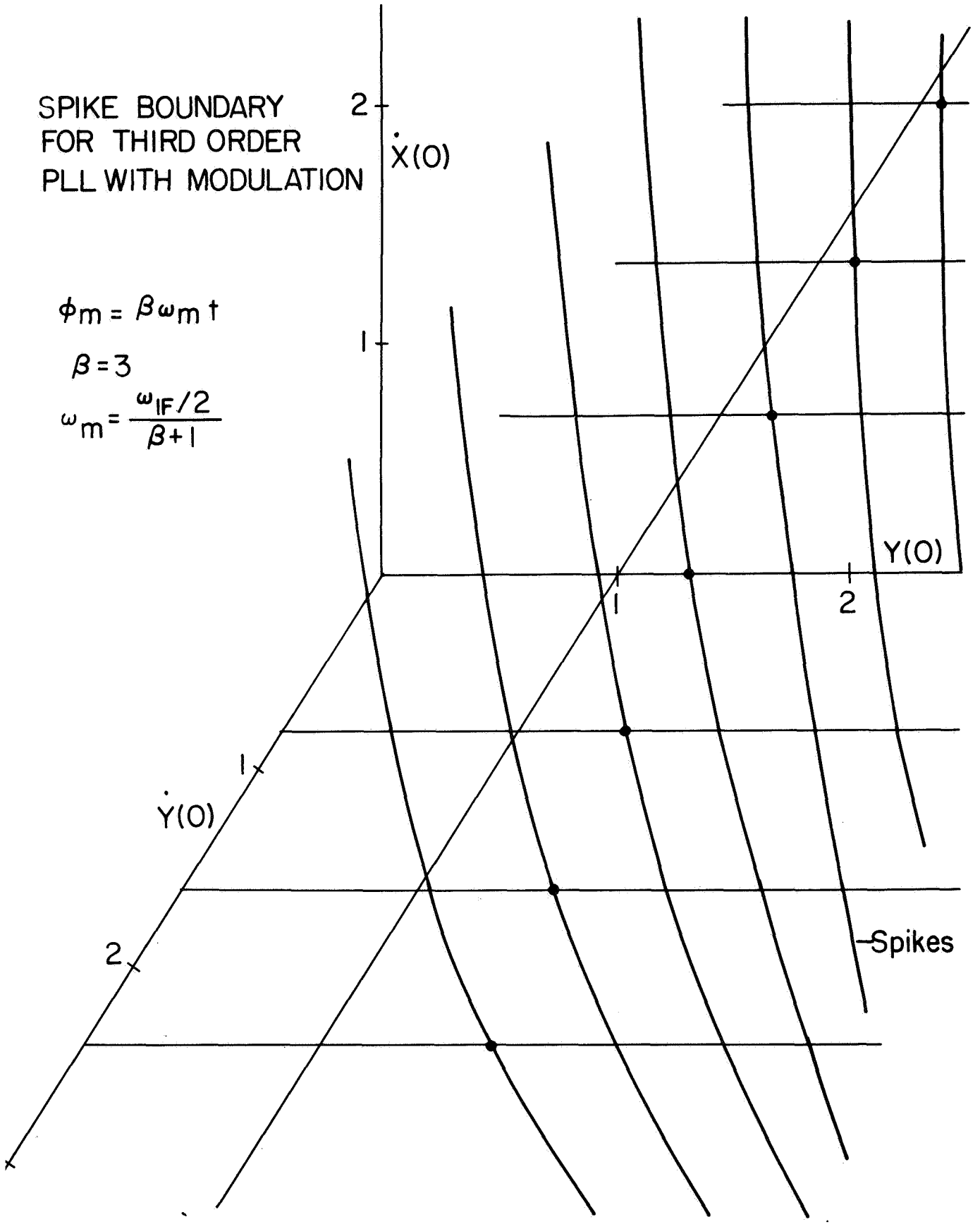

SPIKE BOUNDARY
FOR THIRD ORDER
PLL WITH MODULATION
$\phi_m = \beta\omega_m t$
$\beta = 3$
$\omega_m = \frac{\omega_{IF}/2}{\beta+1}$
2
$\dot{X}(0)$
1
Y(0)
1
2
1
$\dot{Y}(0)$
2
Spikes

FIG. 4.4.2a

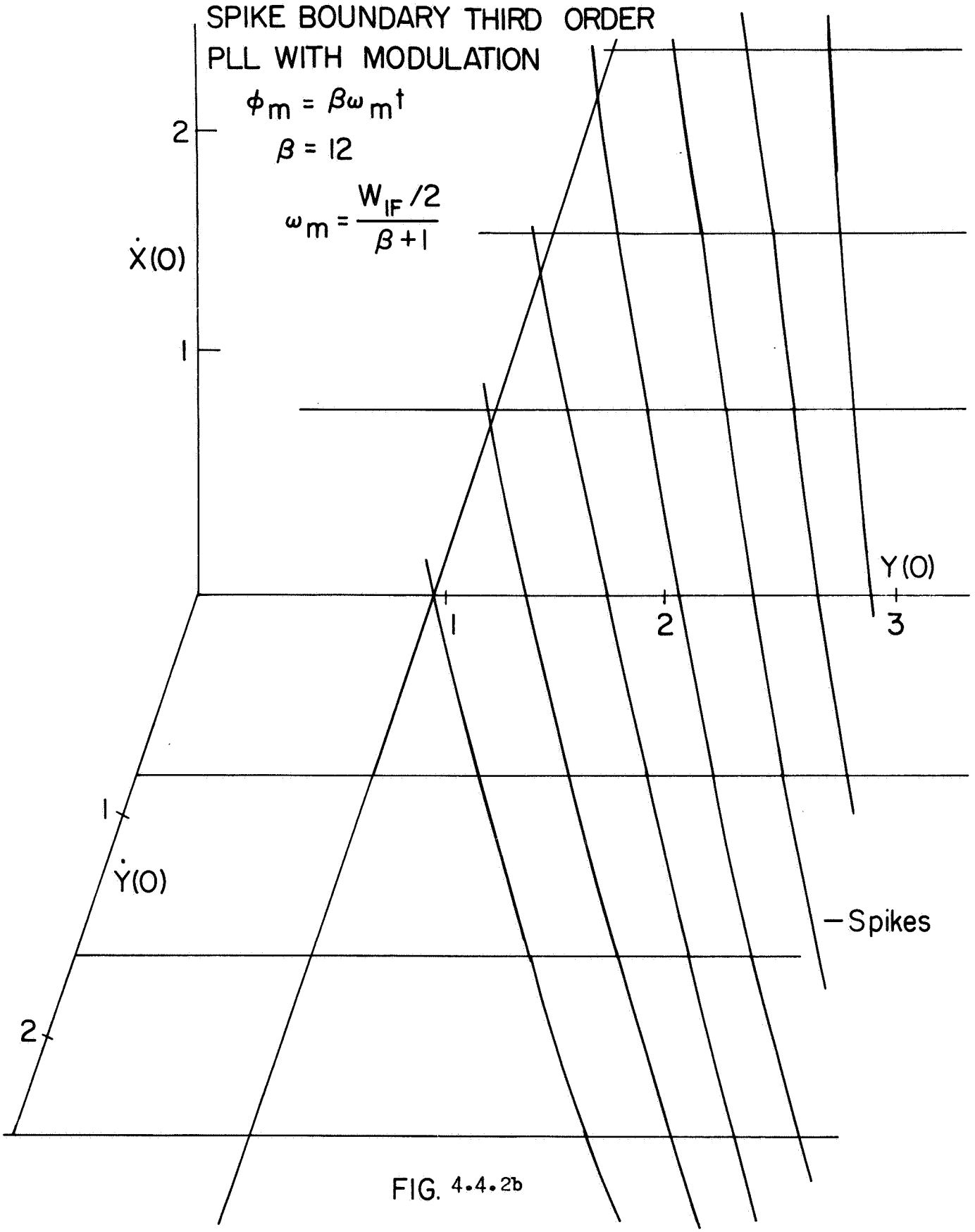

SPIKE BOUNDARY THIRD ORDER
PLL WITH MODULATION
$\phi_m = \beta\omega_m t$
$\beta = 12$
$\omega_m = \frac{W_{IF}/2}{\beta+1}$
2
1
$\dot{X}(0)$
1
2
3
Y(0)
1
2
$\dot{Y}(0)$
Spikes

FIG. 4.4.2b

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 General Description of Experimental Set Up

Fig. 5.1.1 is a block diagram of the PLL's and auxiliary equipment, with which the experimental results were obtained.

A wideband noise generator (General Radio Model 1390-B) having a noise spectrum extending from audio frequencies to 5 MHz, drives a wide bandpass noise amplifier. (See schematic in Fig. 5.1.6). This amplifier has a bandwidth of approximately 20 KHz, so that the noise it presents to the IF filter is virtually "white" noise, with respect to the IF filter which has a BW=800 Hz. The wideband filter is employed so that the noise can be amplified without limiting.

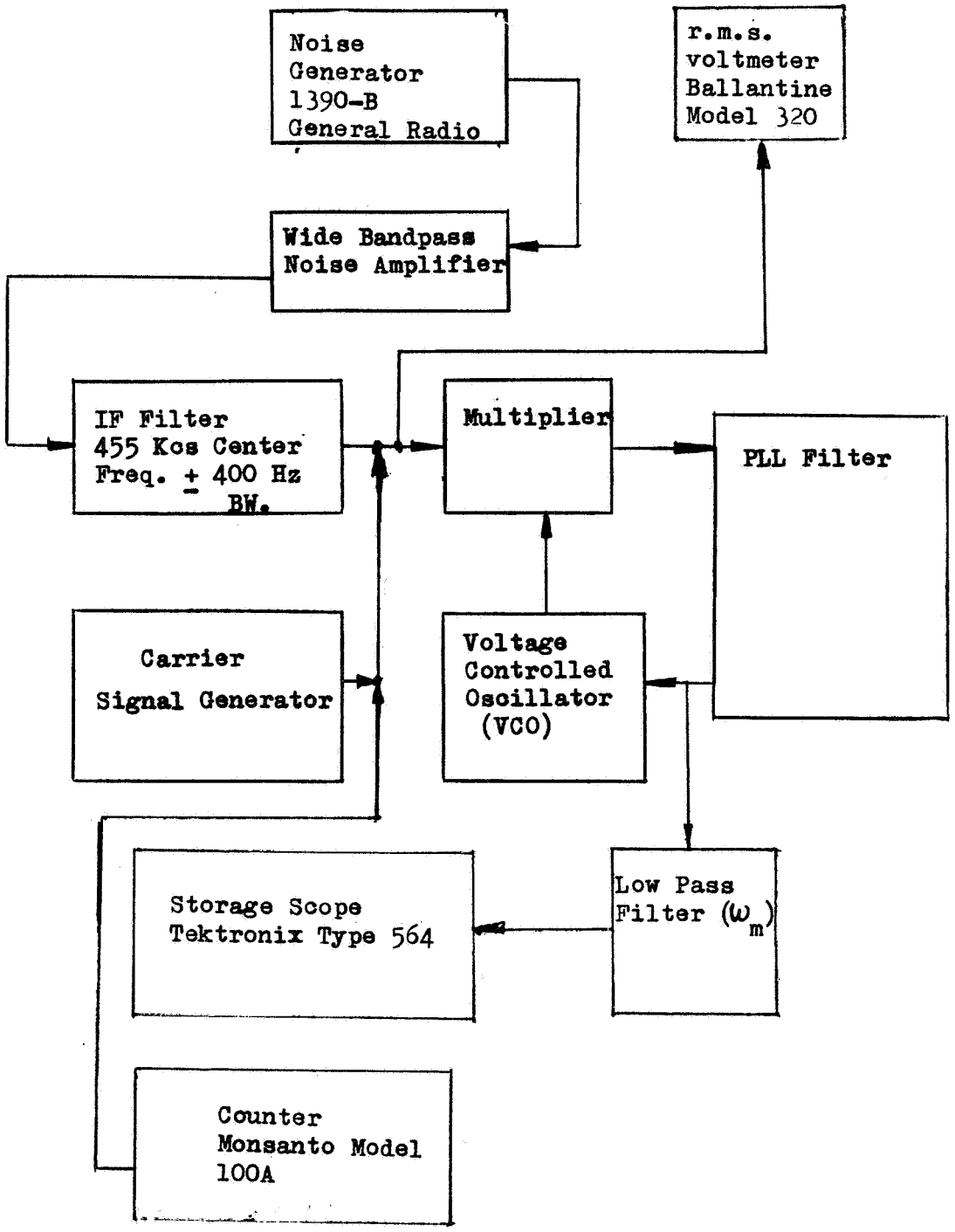
The "white" noise then passes through the IF filter, which has a center frequency of 455 KHz and a 3 db bandwidth of $\pm$ 400 Hz. Since the IF filter consists of a cascade of two identical stages, each with a 3 db bandwidth of $\pm$ 620 Hz, the noise spectrum is shaped to resemble the noise spectrum discussed in Chapter 2.

The carrier, is generated by a (Tektronix type 190A) Signal Generator, and is added to the noise output of the IF filter. The carrier is not passed through the IF filter as that would result in overloading the Filter Amplifiers and hence result in noise limiting. If sinusoidal or Gaussian modulation is employed the modulated carrier must be passed through the filter. Great care must be taken to avoid overload.

An r.m.s. voltmeter (Ballantine Model 320) is used to measure the r.m.s. value of the carrier plus noise. The carrier-to-noise- ratio (CNR) was measured by:

(a) Turning off the noise

FIG. 5.1.1 Block Diagram of PLL



(b) setting the carrier amplitude to a standard value, (270 mv r.m.s.) at which the loop gain is adjusted.

(c) Increasing the noise, until the carrier plus noise, measured on the r.m.s. meter was

$$\text{r.m.s. reading} = .27\sqrt{1 + 1/\text{CNR}} \qquad (5.1.1.)$$

As the highest CNR used in the experimental measurements was about 7.5, the above measurement technique was found to be satisfactory.

5.1.1. Operation of PLL

The carrier plus noise is then supplied to a balanced phase detector (see schematic in Fig. 5.1.7), where it is multiplied with the square wave output of the Voltage Controlled Oscillator (VCO). The multiplier has a small capacitance on its output, to remove the second and higher harmonics, without appreciably affecting the phase error signal.

This error signal is fed to the PLL filter, which may be any of the three filters shown in Fig. 5.1.2, depending upon whether the PLL is a first, second or third order loop.

The output of the PLL filter drives the input of the VCO, thus forming PLL operation. The signal is also supplied to a two-stage RC low pass filter, with a cut-off frequency of 100 Hz. The purpose of this filter, is to attenuate the output Gaussian noise component, accentuating the spikes thereby making them easier to identify. This relative spike accentuation results because the Gaussian noise component of the output has a parabolic frequency spectrum, while the spikes have a flat spectrum.

The spikes are displayed on a Storage Oscilloscope (Tektronix Type 564). The expected number of spikes received are determined as follows:

(a) Set sweep rate at some low value

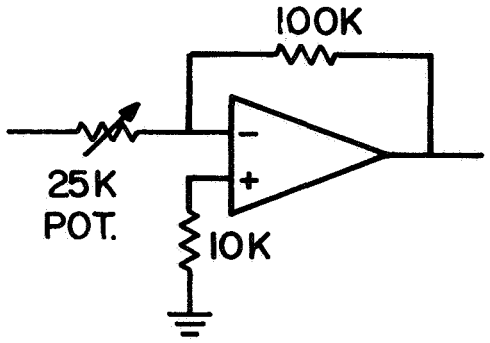


FIG. 5.1.2a
FIRST ORDER PLL FILTER
G_1

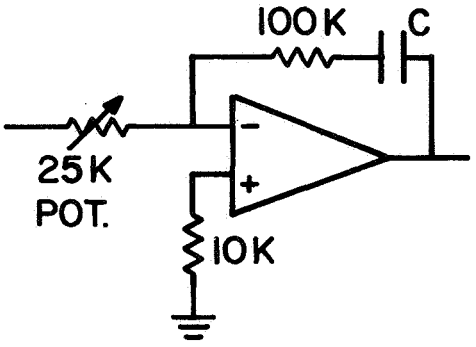


FIG. 5.1.2b
SECOND ORDER PLL FILTER
$G_1 (2 + G_2/S)$

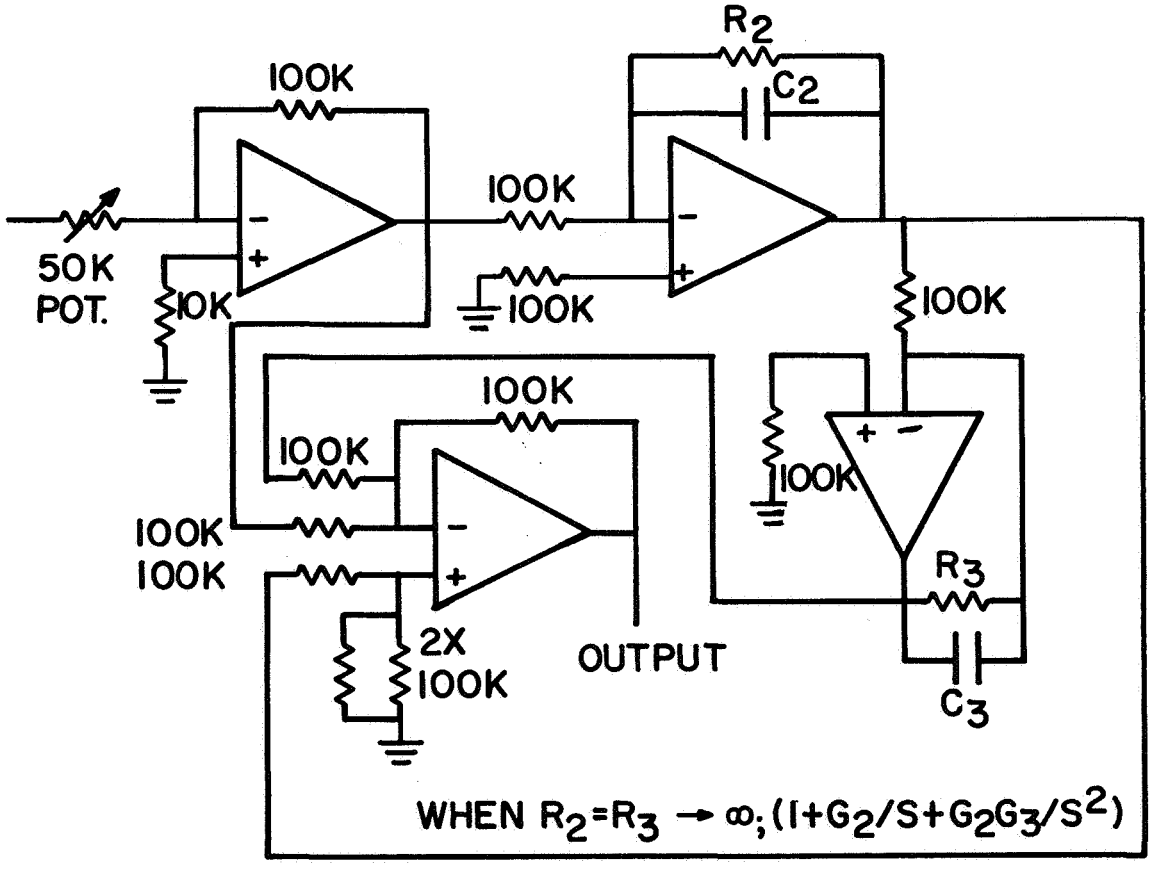


FIG. 5.1.2c THIRD ORDER PLL FILTER

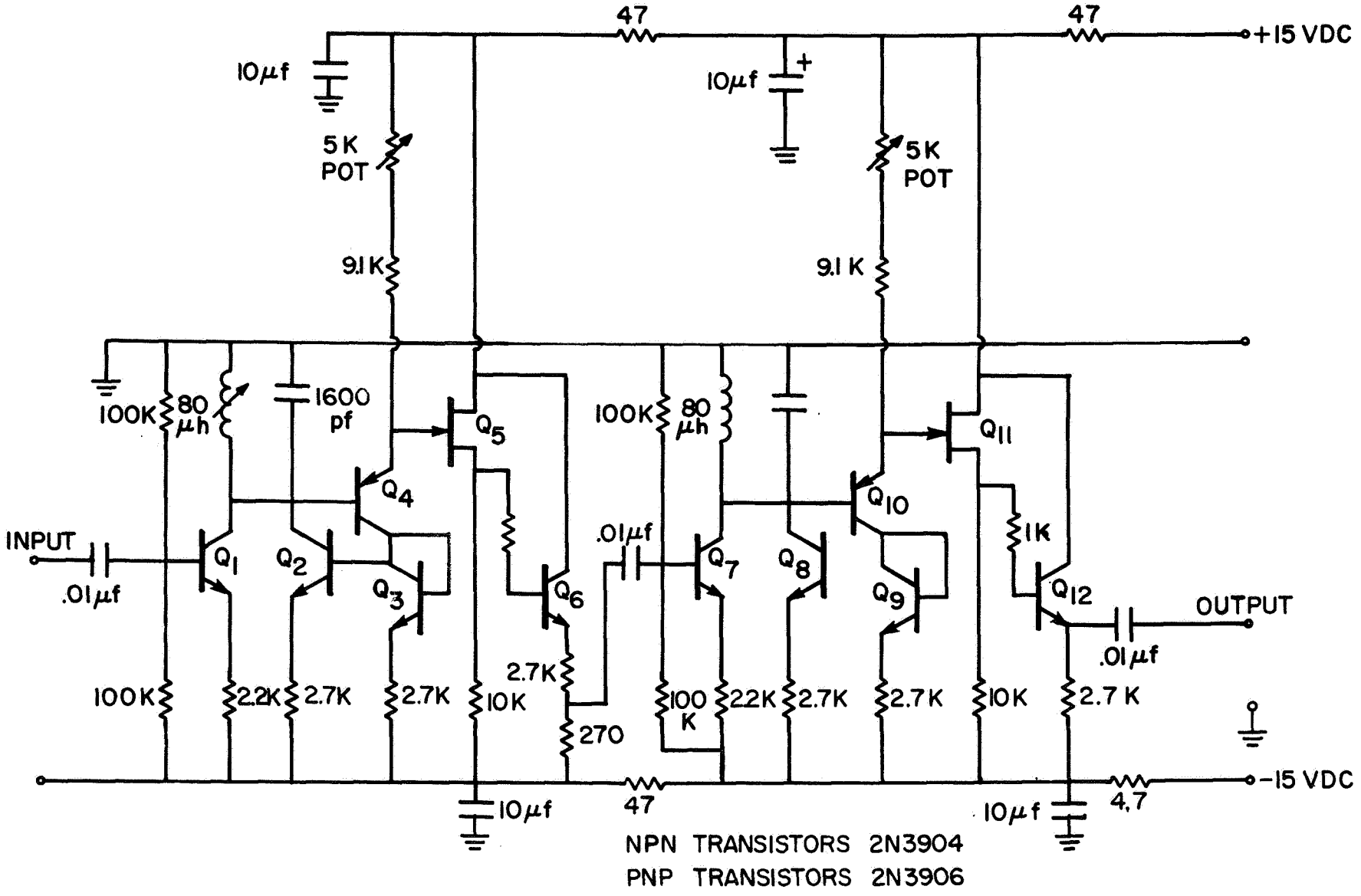


FIG. 5.1.3 IF FILTER 455 Kcs CENTER FREQUENCY

(1 cm/sec. - 5 cm/sec.) so that during a single sweep, a reasonable number of spikes occur.

(b) For each sweep compute N/T, where N is the number of spikes displayed, T is the time taken for one complete sweep to occur.

(c) Repeat until about 50 to 100 spikes have been counted, then average all the N/T's computed for each sweep.

5.2 Operation of IF filter

The schematic of the IF filter used, is shown in Fig. 5.1.3. Transistors Q_1, Q_2, Q_3, Q_4, Q_5, Q_6 comprise the first stage. The rest of the transistors are used for the second stage. As the second stage is essentially identical to the first, it is sufficient to describe the operation of the first stage only.

Transistor Q_1 is a simple common emitter, tuned collector amplifier with an unbypassed emitter resistor.

Transistors Q_2 and Q_4 (Q_3 is used as a compensating diode) form a circuit which presents a voltage stable negative resistance to the tuned circuit. This negative resistance is used to increase the Q of the tuned circuit, permitting narrower bandwidths, than normally attainable. The value of this negative resistance is equal to the 5 k Ω potentiometer plus the 9.1 k Ω fixed resistor.

Transistor Q_5 and Q_6 provide isolation. A voltage divider is used at the output of Q_6, to reduce the gain, and thus the effect of any magnetic

or other coupling between the two stages.

The IF filter is adjusted, as follows:

a) Adjust the first 5 kΩ pot. so that the bandwidth of the first stage is ± 620 Hz.

b) Adjust the second 5 kΩ pot. so that the total bandwidth is ± 400 Hz.

c) Throughout the above procedure, maintain both tuned tank-circuits centered at 455 KHz.

5.3 Operation of the VCO

An astable VCO was originally employed. However, because the IF bandwidth is so narrow, the drift and jitter of the astable VCO was not tolerable.

Fig. 5.1.4 shows the schematic of a voltage-controlled Wien Bridge Oscillator. The Wien Bridge Oscillator is modified so that the resistors in the RC network can be varied.

The operation of this device can be explained by referring to Fig. 5.1.5 as well as the schematic of Fig. 5.1.4.

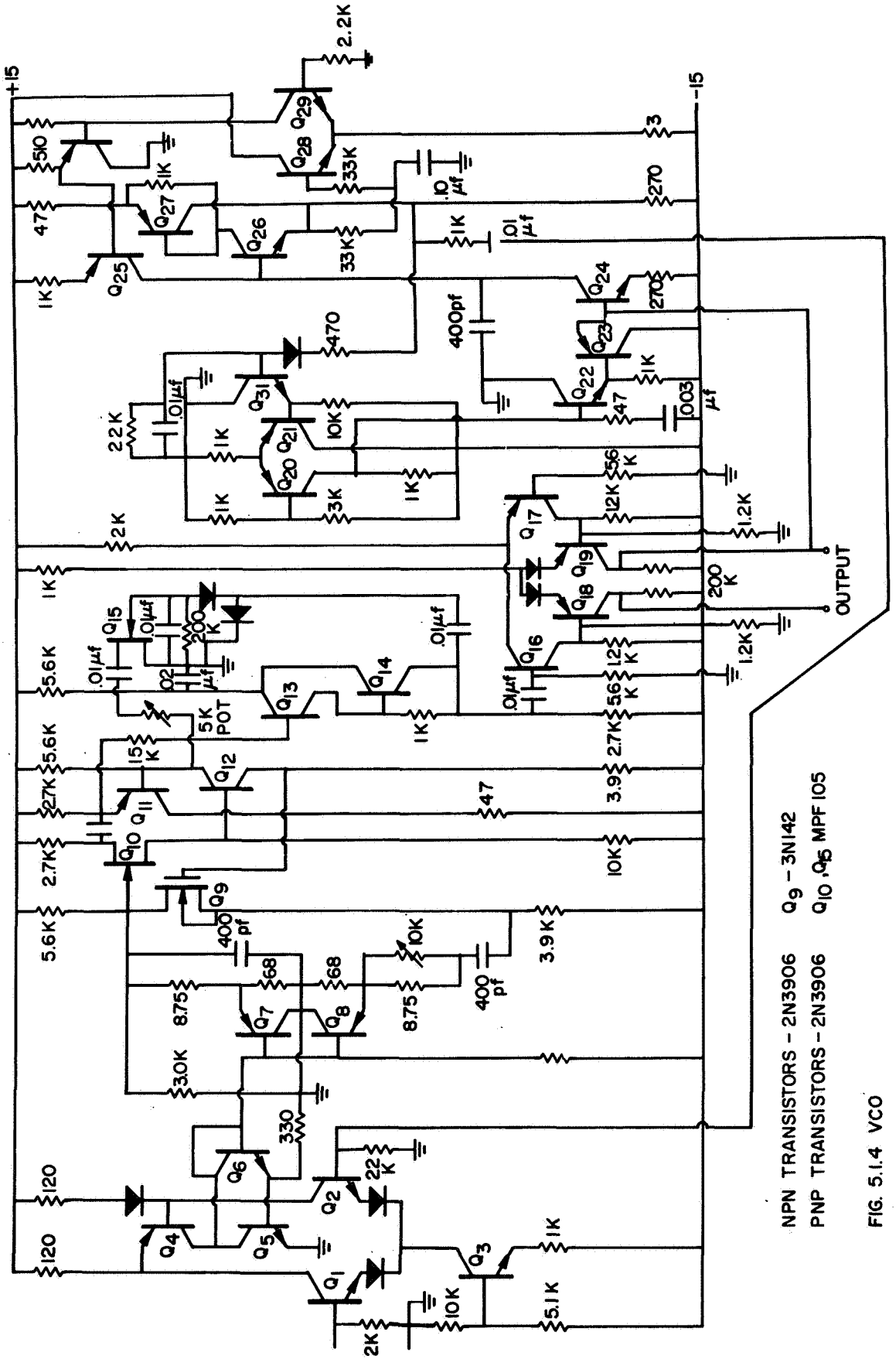
If the two resistances in the Wien Bridge oscillator network are functions of time, the output can easily be shown to be

$$V_{out} = A\sin\left[\int \frac{1}{C}\,\frac{dt}{R(t)} + B\right] \qquad (5.3.1)$$

where the constants A and B deoend upon the initial conditions.

If $1/R(t)$ is $1/R + \Delta R)$ for time T_1 and is $1/R$ for time T_2, and $T_1 + T_2 = T$ where T is the period of the output of Eq. (5.3.1), the resulting signal looks as shown in Fig. 5.1.5c. Part of the signal will be a sine wave of frequency ω_1, and the other part will be a sine wave of frequency ω_2. The frequency of the signal can be determined from Eq. 5.3.1, (by averaging over T)

$$\omega_{av} = \frac{1}{RC}\left[1 - \frac{T_1}{T}\,\frac{\Delta R}{R + \Delta R}\right] \qquad (5.3.2)$$



NPN TRANSISTORS - 2N3906 Q_9 - 3N142

PNP TRANSISTORS - 2N3906 Q_{10}, Q_{15} MPF 105

FIG. 5.1.4 VCO

BLOCK DIAGRAM OF VCO

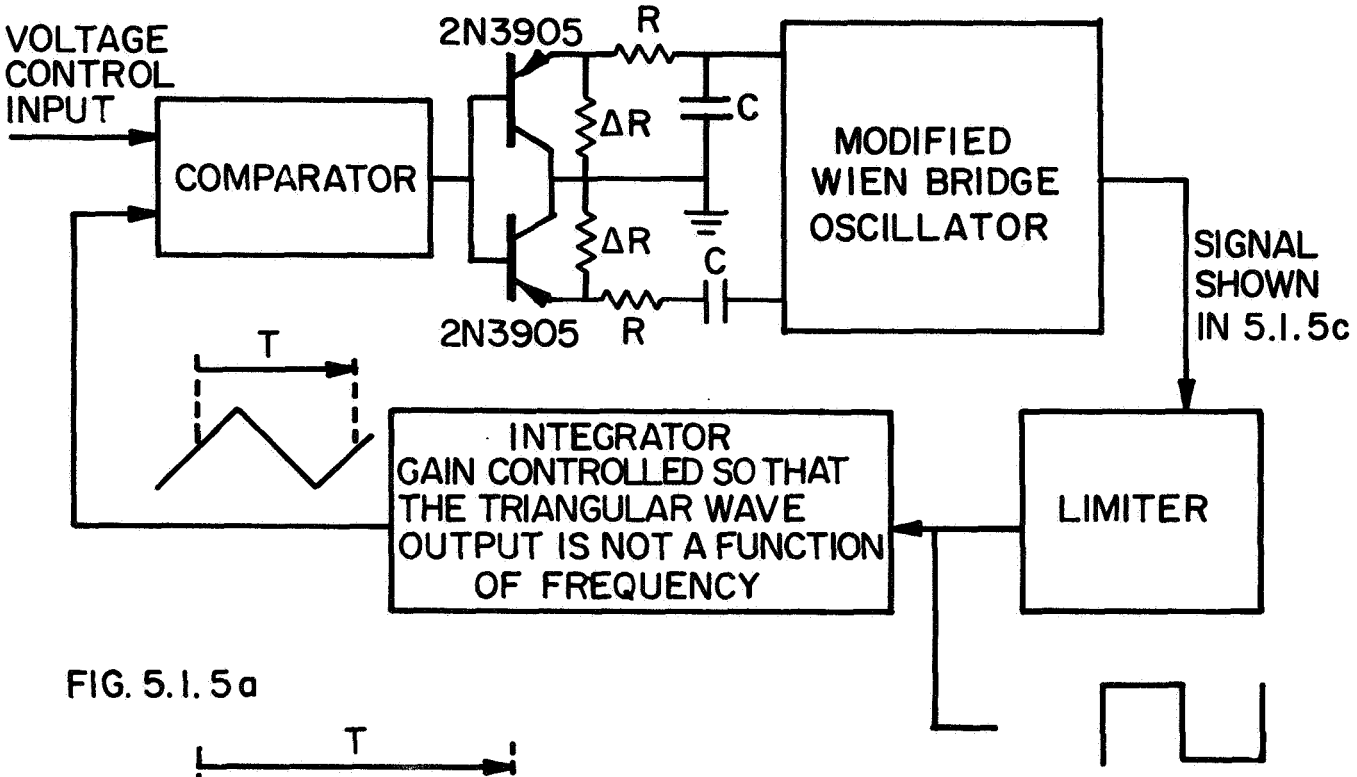


FIG. 5.1.5a

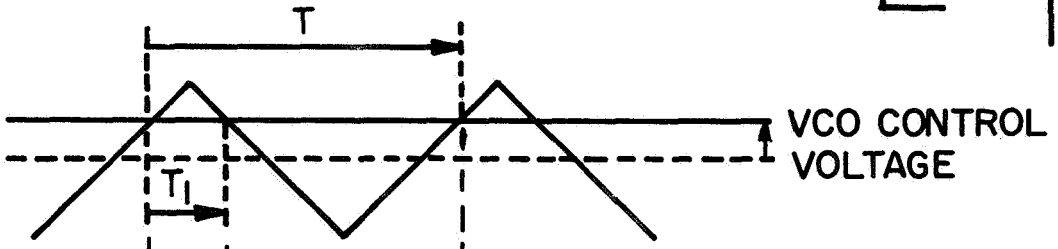


FIG. 5.1.5b

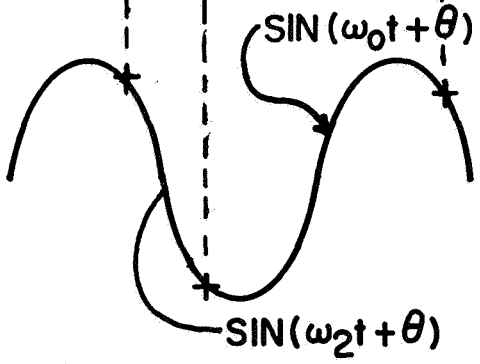


FIG. 5.1.5c

NOISE AMPLIFIER

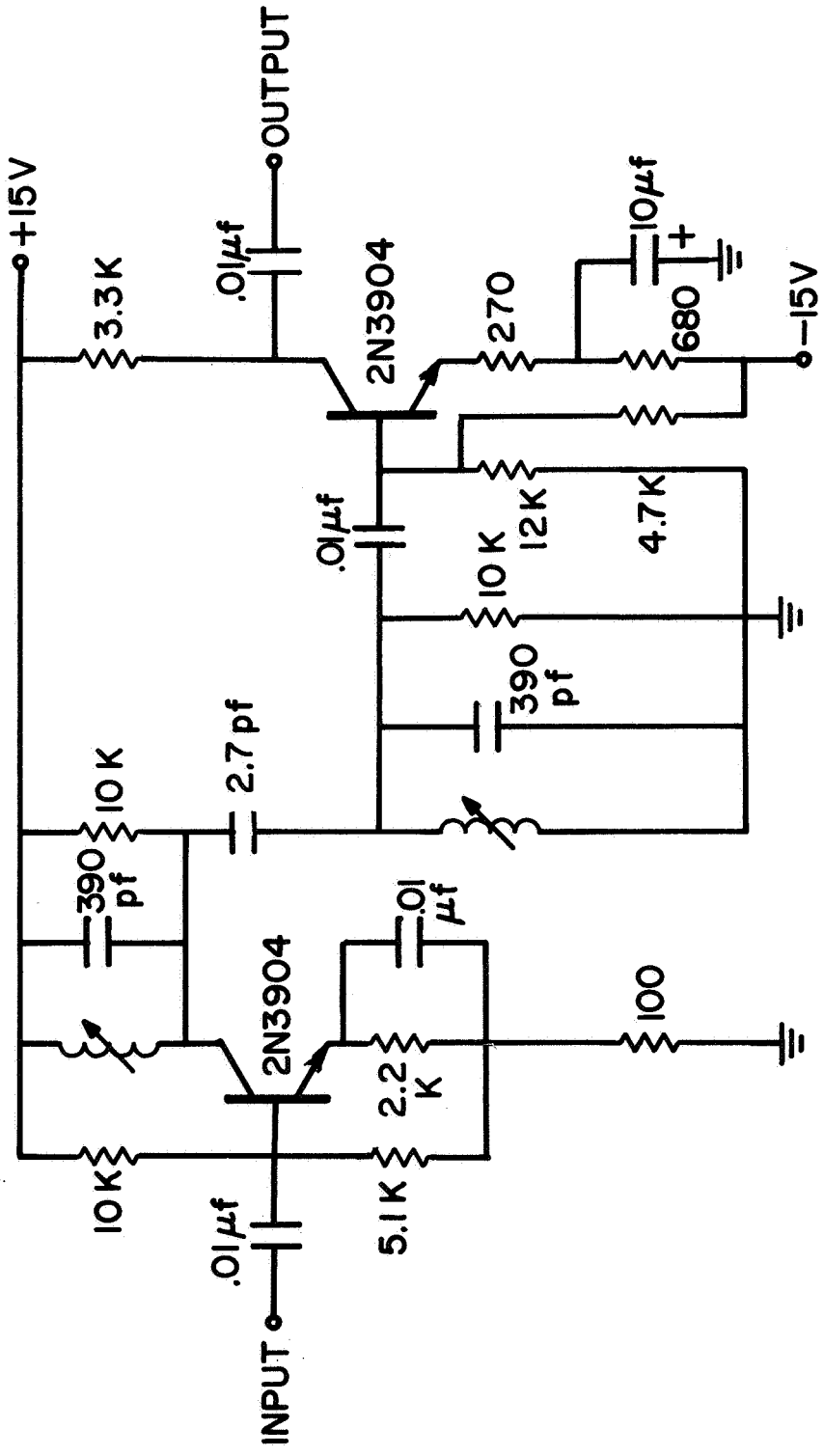
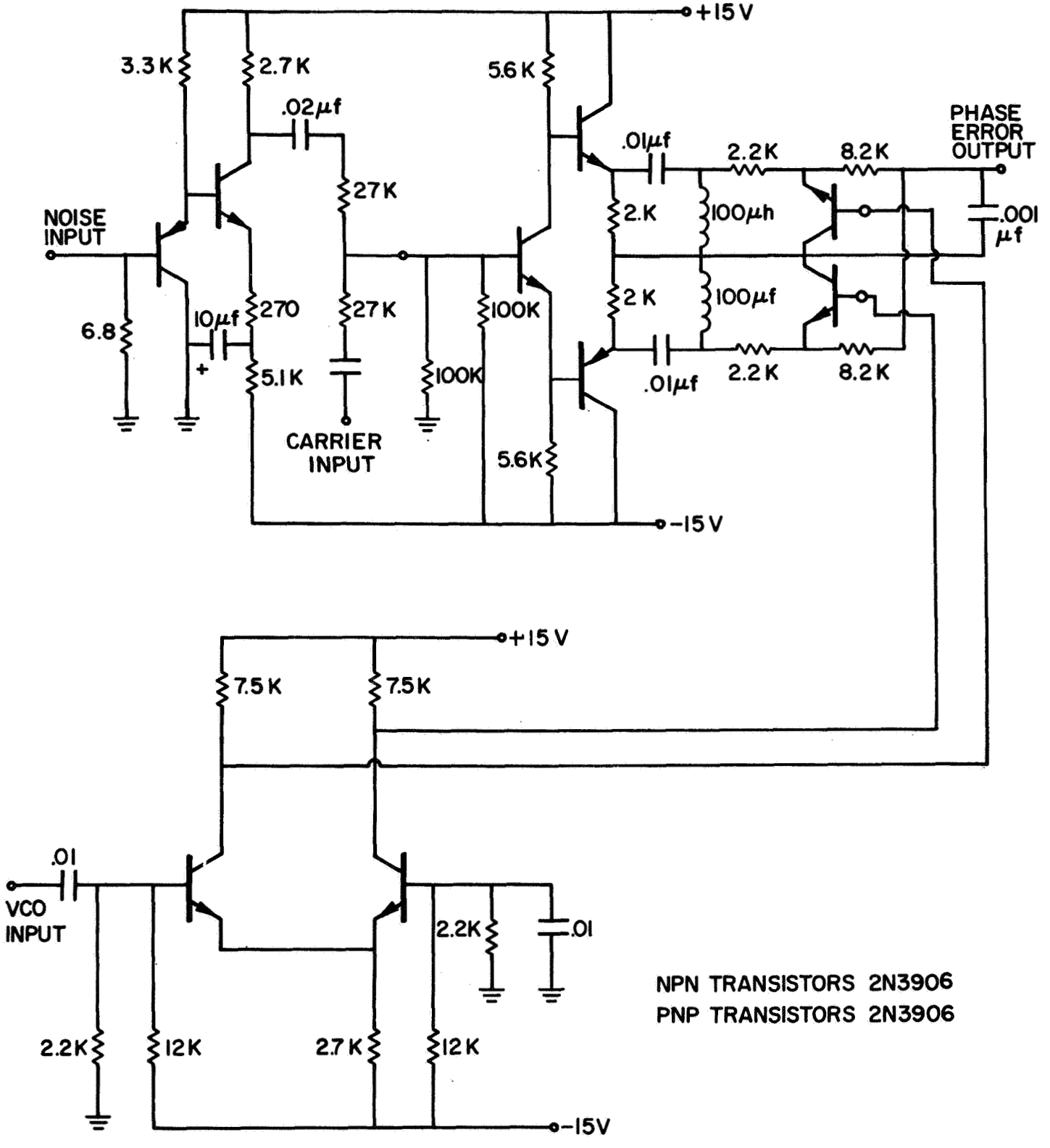


FIG. 5.1.6

FIG. 5.1.7 PHASE DETECTOR (MIXER)



Thus the average frequency is a linear relation with T_1/T. From Fig. 5.1.5b, it can be seen that this will be a linear relation with the control voltage.

Referring to the schematic of Fig. 5.1.4 we see that transistors Q_1, Q_2, Q_3, Q_4 (with Q_5, Q_6 as a voltage clamp) form the comparator (see Fig. 5.1.6a). Transistors Q_7, Q_8 are the switching transistors to change the two resistances of the Wien Bridge network. Transistors Q_9, Q_{10}, Q_{11}, Q_{12} comprise the modified Wien Bridge oscillator.

F.E.T. Q_{15} acts as a gain control of the oscillator to control the amplitude of oscillation. Transistors Q_{13}, Q_{14} are used to amplify the oscillator output. Q_{16}, Q_{17}, Q_{18}, and Q_{19} form an output limiter, which results in an output square wave.

Transistors Q_{24} and Q_{25} act as an integrator of the square wave output, while Q_{26}, Q_{27} serve as a high gain emitter follower, and Q_{28}, Q_{29} maintain proper bias on the integrator (preventing it from drifting into saturation).

Transistors Q_{20}, Q_{21}, Q_{31} act as a differential amplifier, which senses the rectified triangular wave output, and controls the amplitude of the square wave drive to the integrator (via transistors Q_{22} and Q_{23}) so that the amplitude of the triangular wave does not change with frequency.

5.4 Experimental Results for Phase Locked Loops

5.4.1 Normalization of Experimental and Theoretical Results

Experimental and theoretical (Computer-Simulation) results, for the Expected Number of Spikes per second versus Carrier-to-Noise Ratio (CNR), are presented in normalized form in Figs. 5.4.1a and 5.4.1b.

If the differential equation describing the PLL's (Eqs. 1.4.2, 1.4.6,

and 1.4.23) are time scaled by

$$t = T'/\lambda \tag{5.4.1}$$

where t' is the time used in the computer simulation

t is the time without normalization

λ is a normalizing constant

then the IF bandwidth used in the computer-simulation

$$\omega'_{IF} = 2\sqrt{\sqrt{2}-1} \tag{5.4.2}$$

becomes

$$\omega_{IF} = \lambda \omega'_{IF} \quad . \tag{5.4.3}$$

Also the PLL constants

$$G_1 = \lambda G'_1 \tag{5.4.4a}$$

$$G_2 = \lambda G'_2 \tag{5.4.4b}$$

$$G_3 = \lambda G'_3 \tag{5.4.4c}$$

or if the PLL phase-error transfer-function, is chosen to be maximally flat, and $G'_1 = G'/\sqrt{2}$, $G'_2 = G'\sqrt{2}$ for the second order loop, and $G'_1 = 2G'$, $G'_2 = G'$, $G'_3 = G'/2$ for the third order loop then

$$G = \lambda G' \tag{5.4.5}$$

The expected number of spikes/second, received from the time scaled PLL's will be

$$N = \lambda N' \tag{5.4.6}$$

Thus the Computer and Experimental results are normalized, by considering the PLL gains (or rather, phase-error transfer-function cut-off frequency, which is proportional to loop gain) to be that given by Eq. 5.4.5, and instead of the expected number of spikes/second, (in the time scaled PLL) N, $N/\lambda = N\omega'_{IF}/\omega_{IF}$ is plotted.

The noise bandwidth of the low pass equivalent of the IF filter, used in the computer program is defined by:

$$f_{noise} = \frac{1}{2\pi}\sqrt{\frac{\int_{-\infty}^{\infty} \omega^2 \,|H(\omega)|^2 d\omega}{\int_{-\infty}^{\infty} |H(\omega)|^2 \, d\omega}} \qquad (5.4.7)$$

Where $H(\omega)$ is given by Equation 2.3.1, and is the low pass equivalent IF filter. This works out to be $(\frac{1}{2\pi})$ Hz or 1 radian per second. (Remember that the 3 db bandwidth of this low pass equivalent IF filter is $\sqrt{\sqrt{2}-1} \;\hat{=}\; .64$ radians per second).

Thus the Experimental and Theoretical (Computer) results presented in Figs. 5.4.1a and 5.4.1b can be considered to be normalized to unit (in radians per second) noise low pass equivalent IF bandwidth.

Results for other IF filters (e.g. rectangular) can best (but not exactly) compared to the authors results, by normalizing to unit noise bandwidth. For example, experimental results for the first order PLL, $[(2f_1/B\omega) = 5)]$ with a rectangular IF bandwidth of 13 KHz, obtained by Prof. D.T. Hess[5.1] are shown in Fig. 5.4.1a. They agree reasonably well with this author's results.

5.4.2 Experimental Results for the First Order Phase Locked Loop

The experimental apparatus, shown in the block diagram of Fig. 5.1.2, using the PLL filter shown in Fig. 5.1.2a was used to obtain the experimental results shown in Fig. 5.4.1a (square points) for the first order PLL.

The gain of the loop G_1, was adjusted by adjusting the potentiometer in Fig. 5.1.2a so that the loop fell out of lock at $\pm\, G_1/2\pi$ Hz. Since the IF bandwidth, was $\pm$ 4000 Hz, (the Q multipliers in Fig. 5.1.3 , giving the IF filter bandwidth of $\pm$ 400 Hz, was used for the second and third

order PLL's, but not for the first order PLL.)

$$G_1 = 5\lambda = 5*(2\pi\ 4000\ /.643) = 185000 \text{ radians/sec.}$$
$$\text{where } \omega'_{IF}/2 = \sqrt{\sqrt{2}-1} = .643 \quad \text{or } G_1/2\pi = 31 \text{ KHz} \qquad (5.4.8)$$

The expected number of spikes/second was measured using the procedure described in Sec. 5.1.1.

The phase error developed by a first order loop is almost entirely a function of the total deviation (for $\beta > 1$) rather than of the modulating frequencies. (The first order PLL bandwidth has to be much larger than the deviation see Sec. 1.4.2). Thus a first order PLL with a relative gain of 5/(.643) is suitable for any modulation index from 3 to infinity. This is why results for only one value of first order PLL gain are presented.

The results for a constant offset modulation of $\pm$ 4 KHz were obtained by

a) Measuring the Expected Number of Spikes/Sec. with a carrier of 459 KHz.

b) Doing the same at 451 KHz.

c) Averaging the two results

This was done because the results at 459 KHz were less than at 451 KHz. The reasons for this, are given in the discussion of results presented in Sec. 5.4.3.

5.4.3. Experimental Results for the Second Order Constant-plus-Integral Phase Locked Loop

The results for the Second Order Constant-plus-Integral PLL, shown in Figs. 5.4.1a and 5.4.1b (square dots) were obtained using the apparatus shown in Fig. 5.1.1 with the PLL filter shown in 5.1.2b.

Preliminary results were obtained, using a bandwidth of $\pm$ 4 KHz.

Though the IF filter was carefully tuned to 455 KHz, symmetrical spikes were obtained at 456 KHz. The IF filter transfer function was measured and found to be asymmetrical. This resulted in x(t) and y(t), the low pass quadrature and in phase noises to be correlated. It is conjectured that this resulted in the spike asymmetry. A more physical argument is that the noise need deviate less on one side than on the other to produce a spike when there is asymmetry. It was decided to construct a more nearly symmetrical IF filter. To do this, we noted that symmetry improved as the filter Q increased. Thus Q multipliers were added to the IF filter circuit (schematic shown in Fig. 5. 1. 3) and the bandwidth was adjusted to be $\pm$ 400 Hz. The resulting asymmetry was 20 Hz.

This asymmetry was measured by measuring the upper and lower 3 db frequencies of the pass band; the magnitude of the difference between the average of these (one half the sum) and the frequency of maximum amplitude of filter output, was taken as a measure of the asymmetry.

Another difficulty encountered, was that there was coupling between signals in the IF filter, and the carrier added to the output of that filter before being applied to the mixer. This was markedly improved by placing the IF filter circuitry in a shielded container, 1 foot away from the other circuitry, and by power supply decoupling between these other circuits and the IF filter circuit. Even with these precautions, the amplitude of the carrier would vary by approximately 1 or 2 db when its frequency was swept through the IF passband, even though in principle it should not have varied at all since it was added to the noise at the output of the IF filter.

The second order PLL was aligned as follows:

(a) The capacitor, of the series RC network in the feedback path of the OP-AMP shown in Fig. 5. 1. 2b is adjusted so that the RC

time constant of that network is $1/(G_2/2$ times the constant λ). (Note - the desired PLL filter response is $G_1(2 + G_2/S)$)

$$\lambda = \frac{2\pi\ 400}{.643} = 3900$$

Since the resistance of the network is 100 KΩ , ($G_2/2 = \sqrt{2}\ G/2 = G/\sqrt{2}$, and G =. 91 for β = 3, and .56 for β = 12) that capacitor has to be;

$$C = \sqrt{2}\ /\ (3900 * .91 * 10^5) = 4000 \text{ pf}$$
$$C = \sqrt{2}/\ (3900 * .516 * 10^5) = 7030 \text{ pf} \qquad (5.4.9)$$

for a β of 3 and 12 respectively.

(b) When the capacitor is shorted, (the PLL becomes first order with gain 2 G_1) the potentiometer shown in the Fig. 5.1.2b is adjusted to give a lock-in range of 2 G_1 = 2 $G/\sqrt{2}$ = $G/\sqrt{2}$ times the constant λ. This lock-in range (LIR) is:

$$LIR_2 = \frac{2\,G_1\,\lambda}{2\pi} = \frac{(3900)\ (.91)}{2\pi\ \sqrt{2}} = 399 \text{ Hz}$$
$$LIR_2 = \frac{(3900)\ (.516)}{2\pi\ \sqrt{2}} = 226 \text{ Hz} \qquad (5.4.10)$$

for a β (modulation index) of 3 and 12 respectively. If the second order PLL, is pulled out of lock (on either side) when the capacitor is shorted, it will pull back into lock (from either side) when the short is removed only (if it is far enough out of lock), when the offset of the OP-AMP is adjusted so that the net offset of it and the mixer offset is near zero. This adjustment must be re-made periodically since the offsets of the mixer (saturated transistor offset) and the OP-AMP drift in time. If this is not done, and the offset becomes an appreciable fraction of the phase error magnitude the performance of the loop is degraded by an increase in spikes.

5.4.4 Experimental Results for the Third Order Constant-plus-Integral-plus-Double-Integral Phase Locked Loop

The results for the third order loop PLL shown in Figs. 5.4.1a and 5.4.1b (square dots) were obtained using the apparatus shown in Fig. 5.1.1 with the PLL filter shown in Fig. 5.1.2c.

When the loop was first set up, R_2 and R_3 were not used (resulting in a true integral and double integral in the filter). The capacitors were shorted, and the potentiometer (controlling the gain of the PLL filter) adjusted to give a lock in range of $\pm$ G_1 times the constant λ. Since G_1 = 2 G, and G = .512 and .25 for a β (modulation index) of 3 and 12 respectively, the lock in ranges were

$$\begin{aligned} LIR_3 &= \frac{G_1\lambda}{2\pi} = \frac{2*.512*3900}{2\pi} = 635 \text{ Hz} \\ LIR_3 &= \frac{G_1\lambda}{2\pi} = \frac{2*.25*3900}{2\pi} = 310 \text{ Hz} \end{aligned} \tag{5.4.11}$$

for a β of 3 and 12 respectively.

The capacitance of the first integrator was adjusted so that the integrator had a gain of $G_2 = G\lambda$. Thus (since the input resistance to the integrator was 100 kΩ)

$$\begin{aligned} C_2 &= \frac{1}{3900*.512*10^5} = .005\ \mu f \\ C_2 &= \frac{1}{3900*.25*10^5} = .0103\ \mu f \end{aligned} \tag{5.4.12}$$

for a β of 3 and 12 respectively.

Similarly the capacitance of the second integrator was adjusted so that the integrator had a gain of $G_3 = G/2$ times the constant λ. Thus

$$\begin{aligned} C_3 &= 2\,C_2 = .01\,\mu f \\ C_3 &= 2\,C_2 = .0206\,\mu f \end{aligned} \tag{5.4.13}$$

Alignment tests indicated that the loop was not operating properly. The test performed was to offset the loop in each side, so that when both

integrating capacitors C_2 and C_3 were shorted, the loop was out of lock, and let it pull in when the shorts were removed. Due to offsets in the OP-AMPs the loop pulled in from one side only. When the second integrator capacitance was shorted, the loop acted as a constant-plus-integral PLL. The OP-AMP offset for the first integrator could be adjusted so that the PLL (now acting as a second order one) would pull into lock from an out of lock state, from either side. When the short circuit on the second integrator capacitor was removed, its offset could be adjusted, so that the loop would pull into lock, from an out of lock state in both directions. However, this situation, could not be maintained, as drift in the OP-AMPs would (in a few seconds) make the third order PLL be able to pull into lock from one direction only. When noise was applied, some of the spikes had areas of $2n\pi$, (n=3, 4, and larger), and unlike the spikes of area 2π, these larger area spikes occurred in one direction only (depending upon which way OP-AMPs offsets had drifted). Very soon, the largest of these multiple 2π spikes, would cause the loop to lose lock permanently, and the OP-AMP filter would drift off to saturation.

The fact that, whenever the noise caused the loop to loose lock permanently, it always did so in one direction, and that direction of permanent loss of lock could be made to be positive or negative by adjusting OP-AMP offsets, plus the fact that the third order PLL could be made to pull into lock from an unlocked state from any direction, for only a short time by adjusting the OP-AMP offset voltages, indicates that a constant-plus-integral-plus-double-integral PLL may be practical only if chopper-stabilized OP-AMPs are used in constructing the PLL filter, and FET choppers used in the mixer.

The system shown in Fig. 5.1.1 did not employ chopper-stabilization and could not be maintained in balance for a sufficient length of time to

determine the expected number of spikes per second.

In lieu of chopper-stabilization resistances were placed across the integrating capacitances (R_2 and R_3 in Fig. 5.1.2c). When R_2=200 k and R_1=100 k , the tendency of the loop to lose lock permanently was stopped; Cycle slipping was limited to a single cycle, and the symmetry of spikes could be maintained sufficiently long to obtain readings. However, this new PLL is not exactly the same third order PLL used in the computer simulation, which used perfect integrators.

5.4.5 Comments on Other Experimental Difficulties

Three experimental difficulties presented themselves in the second and third order PLL measurements.

The first is that when the carrier generator drifted by as much as 50 Hz, this would cause a five-fold increase in spikes when taking measurements, without modulation. This was a problem that arose when the $\pm$ 400 Hz IF bandwidth was used.

The second, was that a very slight drift in the output of the noise or carrier generator causes a large change in the number of spikes/sec. generated by the loop.

The third, is that to measure N near threshold, especially for a β of 12, one has to measure about 3900* ($5*10^{-6}$) or 0.02 spikes per second. (About 1 spike per minute). Since the apparatus has to be adjusted frequently, this can be very tedious.

The only cure for these difficulties was found to consist of constantly readjusting the frequency of the carrier generator and the output of the carrier and noise generator.

The fact that the output amplitude control of the carrier generator effected its frequency, further complicated experimental procedure.

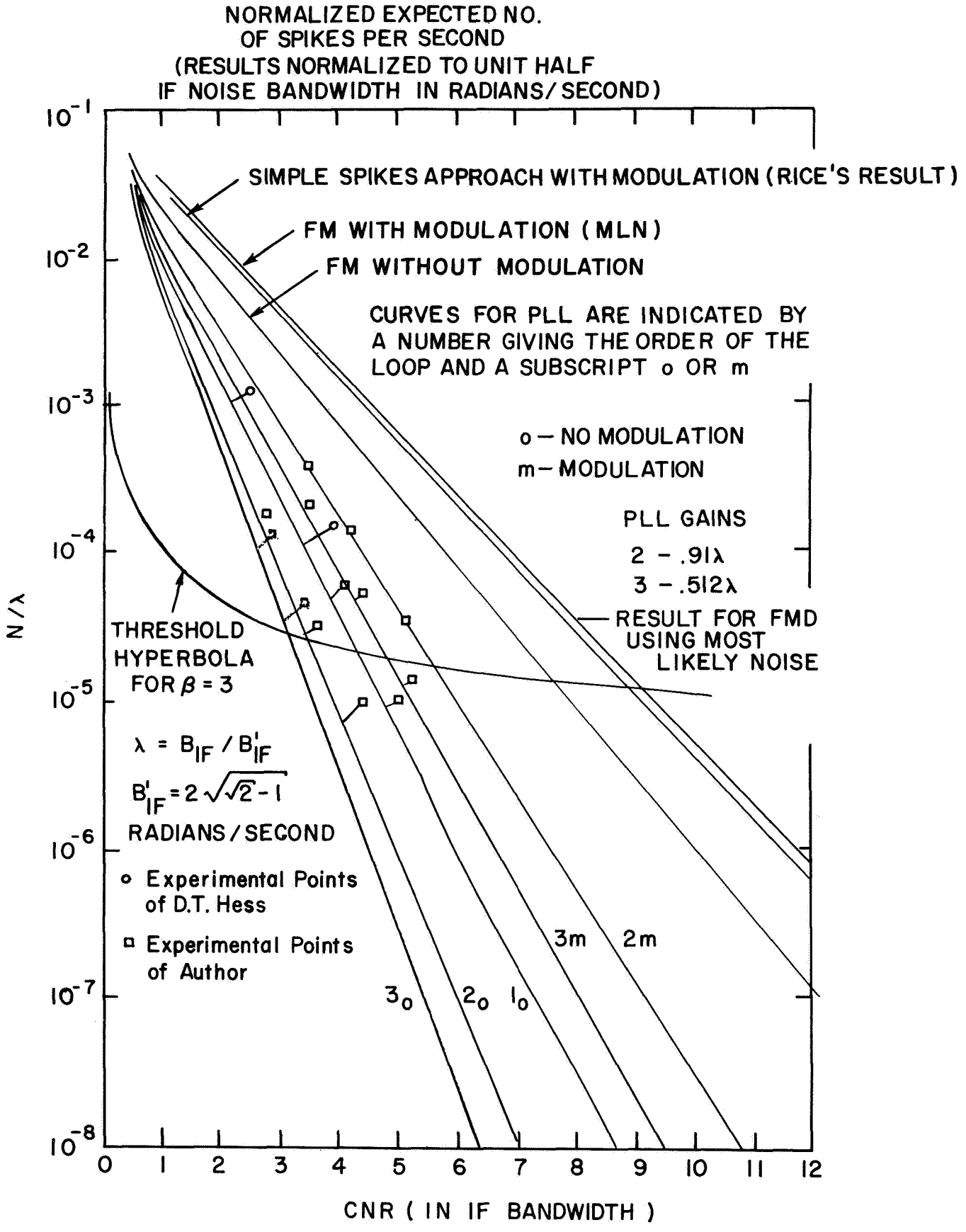

NORMALIZED EXPECTED NO.
OF SPIKES PER SECOND
(RESULTS NORMALIZED TO UNIT HALF
IF NOISE BANDWIDTH IN RADIANS/SECOND)
SIMPLE SPIKES APPROACH WITH MODULATION (RICE'S RESULT)
FM WITH MODULATION (MLN)
FM WITHOUT MODULATION
CURVES FOR PLL ARE INDICATED BY
A NUMBER GIVING THE ORDER OF THE
LOOP AND A SUBSCRIPT o OR m
o — NO MODULATION
m — MODULATION
PLL GAINS
2 - .91λ
3 - .512λ
RESULT FOR FMD
USING MOST
LIKELY NOISE
THRESHOLD
HYPERBOLA
FOR β = 3
λ = B_IF / B'_IF
B'_IF = 2√(√2 - 1)
RADIANS/SECOND
o Experimental Points
of D.T. Hess
□ Experimental Points
of Author
3m
2m
3o
2o
1o
N/λ
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
0
1
2
3
4
5
6
7
8
9
10
11
12
CNR (IN IF BANDWIDTH)

FIG. 5.4.1a

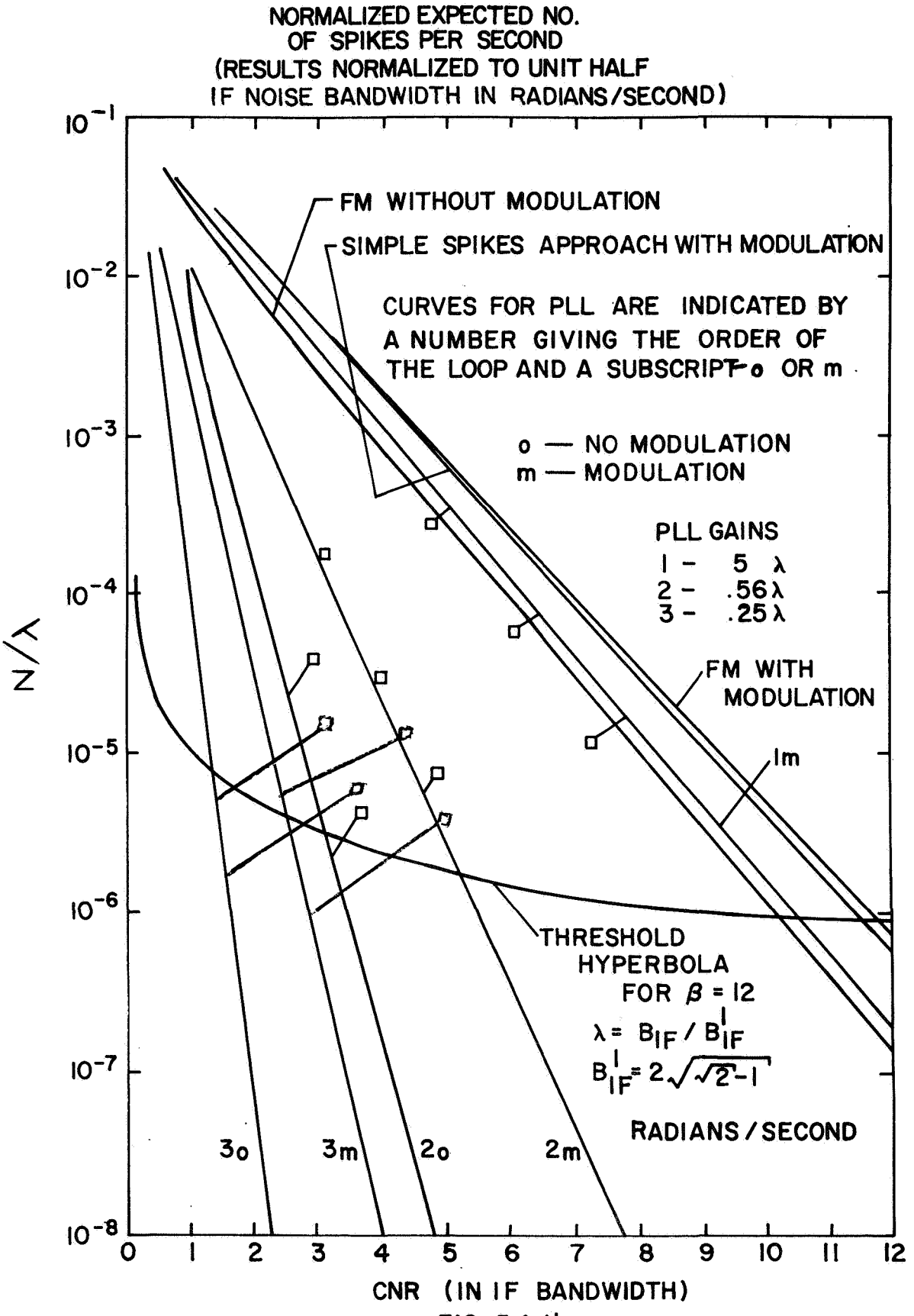

NORMALIZED EXPECTED NO.
OF SPIKES PER SECOND
(RESULTS NORMALIZED TO UNIT HALF
IF NOISE BANDWIDTH IN RADIANS/SECOND)
FM WITHOUT MODULATION
SIMPLE SPIKES APPROACH WITH MODULATION
CURVES FOR PLL ARE INDICATED BY
A NUMBER GIVING THE ORDER OF
THE LOOP AND A SUBSCRIPT o OR m
o — NO MODULATION
m — MODULATION
PLL GAINS
1 - 5 λ
2 - .56λ
3 - .25λ
FM WITH
MODULATION
1m
THRESHOLD
HYPERBOLA
FOR β = 12
λ = B_IF / B'_IF
B'_IF = 2√(√2-1)
RADIANS / SECOND
3o
3m
2o
2m
N/λ
CNR (IN IF BANDWIDTH)

FIG. 5.4.1b

From the above discussion it should be clear that a PLL used in the field probably will not perform as well as the measurements taken here indicate.

5.4.6 Experimental Procedure with Constant offset Modulation for Both the Second and Third Order PLLs.

The constant offest modulation for both the second and third order PLL was taken to be $\beta\ \omega_m$, where $\omega_m = (\omega_{IF} / 2)(\beta + 1)$. Both a positive and negative offset was applied, and results averaged. The deviation was applied, not to the center frequency of the IF bandpass filter, but to a slightly higher frequency, so that the spikes obtained for both positive and negative offsets were approximately equal. This was done, in consideration of the following:

(a) The spikes were found to increase with deviation, very rapidly.

(b) If the IF passband were symmetrical, the amount of spikes/sec. obtained for a deviation positive should be the same as for the same deviation in a negative (lower frequency) direction.

CHAPTER 6

COMPARISON OF EXPERIMENTAL AND THEORETICAL RESULTS

6.1 First Order Phase Locked Loop

The theoretical (or computer) results (solid line) and the experimental results (square dots) for the first order PLL are shown in Fig. 5.4.1b. The agreement between theory and experiment is excellent, both with and without modulation, indicating that Schilling's model of the PLL and the technique of "Most Likely Noise" is valid. An analytic expression for the spikes, obtained from the results is

$$N_o = (.066 + \frac{.004}{G}) e^{-(1.1 + \frac{3.65}{G})} CNR \qquad (6.1.1a)$$

for the no-modulation case, while with modulation the result becomes

$$N_m = \Delta\omega\ (.159)(1 + \frac{0.525}{G}) e^{-(1 + \frac{0.50}{G})} CNR \qquad (6.1.1b)$$

The above two eqs. converge to the results for the FMD as the loop gain G tends to infinity. When G = 5 and the deviation $\Delta\omega$ is .6 radians/sec. Eqs. 6.1.1a and 6.1.1b correspond to the first order PLL results plotted in Fig. 5.4.1a.

6.2. Second Order Phase Locked Loop

Theoretical and experimental results for this PLL are presented in Fig. 5.4.1a and Fig. 5.4.1b for β's of 3 and 12 respectively.

The agreement between theory and experiment for a β of 3 is excellent. The threshold predicted by theory and verified by experiment is 5.6 or 7.5 db with modulation. Indeed, the experimental points fall right on top of the theoretical curve, for modulation, $\beta = 3$ (Fig. 5.4.1a). With no modulation the theoretical threshold is 3.4 or 5.3 db, the corresponding

experimental result is 3.8 or, 5.8 db an error of .5 db.

The agreement between theory and experiment is also excellent for a β of 12, (Fig. 5.4.1b), though not quite as good as for a β of 3. With modulation, the theoretical threshold is 5.2 (CNR) or 7.2 db, while the experimental result is 5.6 or 7.5 db, a 0.3 db error. When there is no modulation, the theoretical threshold is 3.1 or 4.9 db while the corresponding experimental result is 4 or 6 db, an error of 1.1 db.

The results again indicate that the PLL and noise models apply for the second order PLL. The simplest analytic expressions obtainable for the second order PLL results shown in Fig. 5.4.1a and 5.4.1b are:

$$N_o = \left(.066 + \frac{.0135}{G - .391}\right) e^{-\left(1.1 + \frac{1.04}{G - .116}\right)\text{CNR}} \qquad (6.2.1a)$$

for the no modulation case, while with modulation it is

$$N_m = \frac{.102\beta}{\beta + 1}\left(1 - \frac{0.03}{G - 0.726}\right) e^{-\left(1 + \frac{0.276}{G - .312}\right)\text{CNR}} \qquad (6.2.1b)$$

Again these results converge to the results for FMD when G tends to infinity.

6.3 Third Order Phase Locked Loop

Theoretical (straight lines) and experimental (square points) results for this PLL are presented in Figs. 5.4.1a and 5.4.1b for β's of 3 and 12 respectively.

As mentioned in Sec. 5.4.4, the theoretical results used ideal integrators in the PLL filter, while experimental results were obtained with resistors placed across the integrating capacitors. The reasons for this were presented in Sec. 5.4.4. The capacitors have a smaller impedance than their parallel resistors for frequencies greater than 160 Hz for the filter for $\beta = 3$. For $\beta = 12$, this figure is 80 Hz. Since the noise bandwidth (low pass equivalent) is 400 Hz, the PLL filters are

not significantly different from the ideal filter for most of the noise spectrum.

The agreement between theoretical and experimental results for the third order PLL for a β of 3 (Fig. 5.4.1a) is fair. The theoretical threshold with modulation is 4.8 or 6.8 db. The experimental result is 5.2 or 7.3 db and error of 0.5 db. When there is no modulation, the theoretical result is 3.05 or 4.7 db while the corresponding experimental result is 3.9 or 5.9 db an error of 1.2 db. Note however, that while the threshold is not altered significantly, the results indicate no improvement beyond that obtained with the second order PLL.

The agreement between theoretical and experimental results for the third order PLL for a β of 12, is poor. The theoretical threshold, with modulation is 2.6 or 4.2 db, while the experimental result (obtained by extending a line through the two experimental points until it crosses the threshold hyperbola) is 5.2 or 7.2 db, an error of 3.0 db. With no modulation these figures become 1.3 or 1.1 db theoretical, and 4.2 or 6.2 db experimental, an error of 5.1 db.

Again we see that the second and third order loop yield comparable results.

The simplest analytic expressions obtainable for the third order PLL results shown in Figs. 5.4.1a and 5.4.1b are:

$$N_o = \left(.066 + \frac{.0178}{G-.0572}\right) e^{-\left(1.1 + \frac{.71}{G-.169}\right)} \text{ CNR} \qquad (6.3.1a)$$

for the no-modulation case, while with modulation it is

$$N_m = \frac{.102\beta}{\beta+1}\left(1 - \frac{.0248}{G-.364}\right) e^{-\left(1. + \frac{.219}{G-.180}\right)} \text{ CNR} \qquad (6.3.1b)$$

The above two Eqs. converge to the FMD results when G tends to infinity, as is required.

6.3.1 A Look at the Bounds of the Noise Model

The bounds on the noise model trajectory of Fig. 2.1 (square-root of conditional variance) become wider with lower carrier to noise ratios (since the carrier is fixed at an amplitude of unity, lower CNR implies larger σ^2, the noise variance). Since these bounds give a measure of the expected deviation of the noise from the noise model, it is reasonable to expect the "Most-Likely-Noise" method to give less accurate results as the CNR is reduced. This expectation is born out, when we note that the lower the theoretical threshold of the system, the greater the error between experiment and theory.

CHAPTER 7

CONCLUSION

7.1 Conclusions

A computer technique has been presented, which, using the 'Most-Likely Noise' noise model, can give reasonably accurate prediction of threshold in phase locked loops, with and without modulation.

The technique has the advantages, over previous threshold analysis techniques, that, as long as the differential equation of PLL system (or FMFB) can be solved on a digital computer, the method remains tractible, and the effect of modulation and IF filter characteristics are taken into account.

It also provides some qualitative insight to the operation of PLL systems, by the inspection and comparison of the "spike boundaries". For example, it was indicated in Chapter 4 that the second order PLL produced spikes where the FMD did not.

Computer time, to determine the spike boundary varied from 15 minutes (first order PLL) to 35 minutes (third order PLL). About 1 minute of computer time was required to determine the results presented in Figs. 5.4.1a and 5.4.1b from these boundaries. Thus computer time is not excessive.

7.2 Application of Method to Any Deterministic Modulation

In the computer program, the noise disturbance is centered at $t=0$, i.e. all input phase jumps or impulses are delivered to the PLL at or near $t=0$.

If the modulation $\Phi_m(t+t_i)$ is inserted into the PLL differential equations and used in the computer technique, the result will give the expected number of spikes/sec. as a function of CNR given that when

they occur, they occur at $\phi_m(t_i)$.

A probability density function for the modulation $\phi_m(t)$ can be determined by looking at $\phi_m(t)$ with a window of height x and width dx, and averaging the time $\phi_m(t)$ spends passing through the window: Call that time T(x), the density function can be taken to be:

$$f_\phi(x) = M \lim_{\Delta x \to 0} \left[\frac{T(x)}{T}\right] \tag{7.2.1}$$

where

$$\frac{1}{M} = \int_{-\infty}^{\infty} f_\phi(x)\, dx \tag{7.2.2}$$

T is the total time over which

T(x) is determined.

If a number of computer results for expected spikes per second (N_i) are obtained for a selection of $\phi_m(t_i)$, i =... n and these $\phi_m(t_i)$ are spaced closely enough so that the highest frequency component of $\phi_m(t)$ is taken into account, and there are enough $\phi_m(t)$ points taken, to extend over about one cycle of the lowest frequency content of $\phi_m(t)$, then the total expected spikes per second can be taken to be:

$$N = \sum_{i}^{n} N_i \int_{\phi_m(t_i) - |\Delta\phi_i|}^{\phi_m(t_i) + |\Delta\phi_{i+1}|} f_\phi(x)dx \tag{7.2.3}$$

where $\Delta\phi_i = \phi_m(t_i) - \phi_m(t_{i-1})$

For a simple modulation, like a sine wave, N_i(spikes per second), three points along the sine wave would be adaquate (say at 0, $\pi/4$, $\pi/2$), because of its symmetry.

7.3 Suggestions for Future Work

The program of the "Most Likely Noise" computer technique could

be improved. First of all, it could be made more general, so that few changes have to be made, when one wishes to analyze a new system.

The hunting procedure, could be made to find multiple spike regions automatically.

Computer time could perhaps be shortened by optimizing various parameters of the program, for example the initial hunting step.

The program, with slight modification could be used to evaluate signal interference of Phase Locked Loop and FMFB systems. If the main carrier has unit amplitude, suppose there is an interferring carrier offset by $\Delta\omega$ from the main carrier, and having an amplitude of A. The program could be made to hunt for regions in the space of $\Delta\omega$ and A where the PLL or FMFB system does or does not lose the main carrier and lock on to the interferring one.

The FMFB is currently being analyzed using this technique. The technique can also be applied to the MLE where Guida and Schilling have shown that spikes of area 2π also occur.

BIBLIOGRAPHY

1.1 Schwartz, Bennet and Stein, "Communication Systems and Techniques."

1.2 V. I. Tikhonov, "The Effects of Noise on Phase Locked Loop Performance Near Threshold", IEEE Transactions on Communication Theory Technology, August, 1967.

1.3 A. J. Viterbi, "Phase Locked Loop Dynamics in the Presence of Noise by Fokker Planck Techniques" Proc. IEEE - December, 1963, Vol. 51, No. 11 pp. 1737-1753.

1.4 R. C. Boonton, Jr. "Nonlinear Control Systems with Statistical Inputs", Mass. Inst. of Tech., Cambridge, Mass., Dept. No. 66, pp. 1-35, March 1, 1952.

1.5 J. A. Develet, Jr., "A Threshold Criterion for Phase-Lock Demodulation," Proc. IEEE Vo. 51, pp. 349-356; Feb. 1963.

1.6 H. L. Van Trees, "Functional Techniques for the Analysis of the Nonlinear Behaviour of Phase-Locked Loops", presented at WESCON, San Francisco, Calif., Aug. 20-23, 1963.

1.7 R. C. Tausworthe, "A Method for Calculating Phase Locked Loop Performance Near Threshold," IEEE Transactions on Communication Theory Technology, Aug. 1967.

1.8 D. L. Schilling and J. Billig, "A Comparison of the Threshold Performance of the FDFB and the PLL", PIBMRI Report 1207, 1964.

1.9 S. O. Rice, "Noise in FM Receivers" Time Series Analysis, Ch. 25, Wiley, 1963.

1.10 D. L. Schilling and J. Billig, "On the Threshold Extension Capabilities of the PLL and the FDMFB" Proc. IEEE Correspondence, May 1964.

1.11 D. L. Schilling and J. Billig, "Threshold Extension Using the PLL and the FDMFB" presented at the ICMCI Conference in Tokyo, Sept. 1964.

1.12 D. L. Schilling, "Threshold Comparison of the Phase Locked Loop and the Frequency Demodulator Using Feedback" - presented at 1965 International Space Electronics Symposium, November 1965.

2.1 Most Likely Noise Models were suggested by Dr. S. O. Rice of Bell Laboratories in a private correspondence.

3.1 Eugene Isaacson and Herbert Bishop Keller, " Analysis of Numerical Method" Wiley and Sons, N. Y. 1965.

5.1 P. T. Hess, "Cycle Slipping First Order Phase Locked Loops" P. I. B. Progress Report R - 452, 32-67 November 1967.

APPENDIX A

REPRESENTATION OF BANDPASS NOISE

Let Z(t) be the random Gaussian process developed by driving a bandpass filter with white Gaussian noise. The following can always be written;

$$Z(t)=x(t)\cos(\omega_o t) - y(t)\sin(\omega_o t) \tag{a.1}$$

when x(t) and y(t) are defined as

$$x(t) = Z(t)\cos(\omega_o t) + W(t)\sin(\omega_o t) \tag{a.2}$$

$$y(t) = -Z(t)\sin(\omega_o t) + W(t)\cos(\omega_o t) \tag{a.3}$$

W(t) can be considered arbitrary.

The autocorrelations of x(t) and y(t) are

$$\begin{aligned} E[x(t+\tau)x(t)] = E\{ & Z(t+\tau)Z(t)\cos[\omega_o(t+\tau)]\cos(\omega_o t) \\ & + Z(t+\tau)W(t)\cos[\omega_o(t+\tau)]\sin(\omega_o t) \\ & + Z(t)W(t+\tau)\cos(\omega_o t)\sin[\omega_o(t+\tau)] \\ & + W(t+\tau)W(t)\sin[\omega_o(t+\tau)]\sin(\omega_o t)\} \end{aligned} \tag{a.4}$$

$$\begin{aligned} E[y(t+\tau)y(t)] = E\{ & Z(t+\tau)Z(t)\cos[\omega_o(t+\tau)]\cos(\omega_o t) \\ & - Z(t+\tau)W(t)\sin[\omega_o(t+\tau)]\cos(\omega_o t) \\ & - Z(t)W(t+\tau)\sin(\omega_o t)\cos[\omega_o(t+\tau)] \\ & + W(t+\tau)W(t)\sin[\omega_o(t+\tau)]\sin(\omega_o t)\} \end{aligned} \tag{a.5}$$

These autocorrelations become stationary when Z(t) and W(t) are stationary and

$$R_z(\tau) = R_w(\tau) \tag{a.6}$$

$$R_{zw}(\tau) = -R_{wz}(\tau) \tag{a.7}$$

Then Eqs. 4 and 5 become

$$R_x(\tau) = R_y(\tau) = R_z(\tau)\cos(\omega_o\tau) - R_{zw}(\tau)\sin(\omega_o\tau) \tag{a.8}$$

Let the random process W(t) be generated by driving a filter with Z(t); let the Fourier transform of this filter be

$$H(\omega) = -j\,\mathrm{sgn}(\omega) = \begin{cases} -j; \omega > 0 \\ 0; \omega = 0 \\ j; \omega < 0 \end{cases} \qquad \text{(a. 9)}$$

This is admissible since $|H(\omega)|^2 = 1$, satisfying Eq. a. 6, and $H(\omega) = -H^*(\omega)$, satisfying Eq. a. 7.

The cross correlation between x(t) and y(t) is

$$\begin{aligned} R_{xy}(\tau) &= E[x(t+\tau)y(t)] \\ &= E\{-Z(t+\tau)Z(t)\cos[\omega_o(t+\tau)]\sin(\omega_o t) \\ &\quad + Z(t+\tau)W(t)\cos[\omega_o(t+\tau)]\cos(\omega_o t) \\ &\quad - W(t+\tau)Z(t)\sin[\omega_o(t+\tau)]\sin(\omega_o t) \\ &\quad + W(t+\tau)W(t)\sin[\omega_o(t+\tau)]\cos(\omega_o t)\} \end{aligned} \qquad \text{(a. 10)}$$

$$R_{xy}(\tau) = R_z(\tau)\sin\omega_o\tau + R_{zw}(\tau)\cos\omega_o\tau \qquad \text{(a. 11)}$$

From Eq. a. 11 and the Eqs.

$$S_{wz}(\omega) = S_z(\omega)H(\omega) = -S_z(\omega)H^*(\omega) = -S_{zw}(\omega) \qquad \text{(a. 12)}$$

we get for the spectral density,

$$\begin{aligned} S_{xy}(\omega) = \frac{1}{2j}\{ & S_z(\omega-\omega_o) - S_z(\omega+\omega_o) \\ & - S_z(\omega-\omega_o)\mathrm{sgn}(\omega-\omega_o) - S_z(\omega+\omega_o)\mathrm{sgn}(\omega+\omega_o)\} \end{aligned} \qquad \text{(a. 13)}$$

If x(t) and y(t) are uncorrelated (and hence independent since they are Gaussian),

$$S_{xy}(\omega) = 0 \qquad \text{(a. 14)}$$

For $|\omega| < \omega_o$

$$S_{xy}(\omega) = 0 = \frac{1}{j}\{S_z(\omega-\omega_o) - S_z(\omega+\omega_o)\} \qquad \text{(a. 15)}$$

This implies

$$S_z(\omega-\omega_o) = S_z(\omega+\omega_o) \quad |\omega| < \omega_o \qquad \text{(a. 16)}$$

From Eq. a. 13 and a. 14

$$S_z(0) = 2S_z(2\omega_o) \quad \omega = \omega_o \tag{a. 17}$$

$$2S_z(-2\omega_o) = S_z(0) \qquad \omega = -\omega_o \tag{a. 18}$$

$$S_z(\omega - \omega_o) = 0 \qquad \omega < -\omega_o \tag{a. 19}$$

$$S_z(\omega + \omega_o) = 0 \qquad \omega > \omega_o \tag{a. 20}$$

If $S_z(\omega)$ is a bandpass spectral density

$$S_z(0) = 2S_z(2\omega_o) = 2S_z(-2\omega_o) = 0 \tag{a. 21}$$

Eqs. a. 16, a. 19, and a. 20 indicate that $S_z(\omega)$ is a symmetrical bandpass spectrum. Thus we have proved that the condition for the process x(t), and y(t) to be independent is that the bandpass filter be symmetric.

We now turn to the question of what conditions ensure that x(t) and y(t) are low pass random processes. From Eqs. a. 8 and a. 12, the spectral densities of x(t) and y(t) are

$$S_x(\omega) = S_y(\omega) = \frac{1}{2}\{ S_z(\omega - \omega_o) + S_z(\omega + \omega_o) \tag{a. 22}$$

$$-S_z(\omega - \omega_o)\mathrm{sgn}(\omega - \omega_o) + S_z(\omega + \omega_o)\ \mathrm{sgn}\ (\omega + \omega_o)\}$$

For $|\omega| < \omega_o$ Eq. a. 22 is

$$S_x(\omega) = S_y(\omega) = S_z(\omega - \omega_o) + S_z(\omega + \omega_o) \quad |\omega| < \omega_o \tag{a. 23}$$

For $\omega < -\omega_o$ Eq. a. 22 is

$$S_x(\omega) = S_y(\omega) = S_z(\omega - \omega_o) \quad \omega < -\omega_o \tag{a. 24}$$

While for $\omega > \omega_o$ it is

$$S_x(\omega) = S_y(\omega) = S(\omega + \omega_o) \tag{a. 25}$$

Thus we see from Eqs. a. 23, a. 24, and a. 25 that any bandpass spectrum for $S_z(\omega)$ makes x(t) and y(t) lowpass random processes, so long as ω_o is in the passband.

APPENDIX B

OUTPUT POWER OF AN IDEAL LOW PASS FILTER DRIVEN BY SYMMETRIC POISSON DISTRIBUTED IMPULSES OF AREA 2π

Over a time interval T, a series of Symmetric Poisson distributed impulses of area 2π can be represented by

$$f(t) = 2\pi \sum_{i=1}^{n} m_i \delta (t-t_i) \qquad \text{(b. 1)}$$

where n is an integer random variable whose mean is NT which is the expected number of impulses occurring in T sec.

T is the time interval over which f(t) is observed

N is the expected number of impulses per second

t_i is a set of independent random variables uniformly distributed over the interval 0 to T

m_i is a set of independent random variables which can take on the values +1 or -1 with equal probability.

The Fourier Transform of f(t) is

$$F(\omega) = 2\pi \sum_{i=1}^{n} m_i e^{-j\omega t_i} \qquad \text{(b. 2)}$$

When f(t) drives an ideal low pass filter of bandwidth ω_m the output is

$$g(t) = \frac{1}{2\pi} \int_{-\omega_m}^{\omega_m} m_i e^{-j\omega t_i} d\omega \qquad \text{(b. 3)}$$

The total output power is

$$P = \int_{-\infty}^{\infty} g(t)^2 dt = \frac{1}{2\pi} \int_{-\infty}^{\infty} G(\omega) G(-\omega)\, d\omega$$

$$= \frac{4\pi^2}{2\pi} \int_{-\omega_m}^{\omega_m} \sum_{i,k=1}^{n} m_i m_k e^{-j\omega (t_i - t_k)}\, d\omega$$

$$= 4\pi\,\omega_m \sum_{i,k=1}^{n} m_i m_k e^{-j\omega(t_i - t_k)}$$

$$P = 4\pi\,\omega_m n + 4\pi\,\omega_m \sum_{\substack{i,k=1 \\ i \neq k}}^{n} m_i m_k e^{-j\omega(t_i - t_k)} \qquad \text{(b. 4)}$$

The average power is

$$P_s = \lim_{T \to \infty} \frac{E[P]}{T} \qquad \text{(b. 5)}$$

$$P_s = 4\pi\,\omega_m N \qquad \text{(b. 6)}$$

The expectation of the second term on the right of Eq. b. 4 is zero since the m_i are independent, and have zero means.

APPENDIX C

SAMPLE PROGRAM TO FIND SPIKE BOUNDARIES

(FOR SECOND ORDER PLL)

```
DIMENSION E(7), ET(7), U(7), UT(7), T(7), EN(5), ENT(5), UN(5), UNT(5),
1AC(9), BB(9), ER(3)
PI = 4.*ATAN (1.)
S1 = 1./6.
S2 = 1./3.
```

} Constants used in Runge-Kutta procedure

```
Z1 = 55./24.
Z2 = -(59./24.
Z3 = 37./24.
Z4 = -(9./24.)
X1 = -Z4
X2 = 19./24.
X3 = -(5./24.
X4 = 1./24.
```

} Constants used in Moulton's Predictor-Corrector procedure.

```
E1 = 0.
E2 =0.
E3 = 0.
E4 = 0.
```

} These constants are here given numerical values, for the benefit of the Common statement that follows. They shall be defined later in the program.

```
EF1 = .00005
EF2 = .000001
```

} The constants EF1 and EF2 are used in the error control technique.

```
A = 0.
B = 0.
C = 0.
G1 = 0.
G2 = 0.
```

} These constants are here given numerical values, for the benefit of the Common statement that follows. They shall be defined later in the program.

```
OM1 = 0.
OM = 0.
BETA = 0.
```

These constants are here given numerical values, for the benefit of the Common statement that follows. They shall be defined later in the program.

```
COMMON/ROOT/A, B, C, G1, G2, OM1, OM, BETA/TOPT/E1, E2,E3, E4
OM = SQRT(SQRT(2.)-1.)
DO 95 IZ = 3, 12, 9
BETA = IZ
OM1 = OM/(BETA + 1.)
AI1 = SQRT (2.) }
```

OM is the low pass equivalent IF bandwidth, in radians/sec. The "DO" loop to the left, selects modulation indeces of 3 and 12 in succession.

This value of AI1 makes the loop phase-error transfer-function, max. flat

```
BI1=10.
G = OM1*((BI1*BETA)**2 -1.)**.25)
G1 = G/AI1
G2 = G*AI1
```

G (loop gain) is given a value such that the phase error magnitude is .1 (1./BI1)

```
DO 91 MJ = 1, 7
PMJ = MJ-4
C = .8*PMJ
```

The DO loop provides a selection of values for $C(\dot{y}(0))$.

```
VDT = 1.
IX = 2.
IP = 0
SDP = 0.
CDP = 1.
THT = 1.
```

VDT and THT are constants used in the circular extension procedure. The other constants, are used in the hunting procedure, except SDP and CDP define the hunting direction.

```
   A = 2.5
   B = 1.
   VK = .0.
   AC(IK) = A
   BB(IK) = B
   AB = .25
```

A ($\dot{x}(0)$) and B ($\dot{y}(0)$) set an initial trial point. VK now set = 0, will be the distance the hunting process takes us from the trial point. AB is the hunting interval.

```
 6 IF(ABS(VK) . GE. .5 . AND. IK . GE. 4 . AND. ABS(VDT) . GT. . 00001)
 1 GO to 79
   A = VK*SDT + AC(IK)
   B = VK*CDP + BB(IK)
   T(1) = -5.
   U(1) = -5. *BETA*OM1
   UT(1) = BETA*OM1
   H = .125
   GO TO 13
 8 T(1) = T(N)
   U(1) = U(N)
   UT(1) = UT(N)
 9 H = H*. 5
   GO TO 13
10 T(1) = T(N+1)
   U(1) = U(N+1)
   UT(1) = UT(N+1)
```

The "IF" statement executes the first line of Block 6 in Fig. 3.3.1.

U and UT are the phase and frequency of the PLL VCO respectively.

H is the differential equation integration interval.

```
13 CALL RT(T(1))
   DO 25 M = 1, 3
   P = M - 1
   T(M) = P*H + T(1)
   E(M) = UT(M)
   ET(M) = F(U(M), UT(M))
   TV1 = T(M) + .5*H
   CALL RT(TV1)
   EB = UT(M) + .5*ET(M)*H
   EBT = F(U(M) + .5*E(M)*H, EB)
   EC = UT(M) + .5*EBT*H
   ECT = F(U(M) + .5*EB*H, EC)
   TV2 = T(M) + H
   CALL RT(TV2)
   ED = UT(M) + ECT*H
   EDT = F(U(M) + EC*H, ED)
   U(M + 1) = U(M) + H*(S1*E(M) + S2*EB + S2*EC + S1*ED)
25 UT(M + 1) = UT(M) + H*(S1*ET(M) + S2*EBT + S2*ECT + S1*EDT)
```

Statements 13 to 25 computes 3 values of the solution beyond the initial value.

```
   N = 3
   T(4) = T(3) + H
   ER(2) = 0.
   ER(3) = 0.
27 N = N + 1
   IF(N .EQ. 7) N = N - 1
   E(N) = UT(N)
   ET(N) = F(U(N), UT(N))
   V = U(N) + H*(Z1*E(N) + Z2*E(N-1) + Z3*E(N-2) + Z4*E(N-3))
   VT = UT(N) + H*(Z1*ET(N) + Z2*ET(N-1) + Z3*ET(N-2) + . Z4*ET(N - 3))
   T(N+1) = T(N) + H
   CALL RT(T(N+1))
   E(N+1) = VT
   ET(N+1) = F(V, VT)
   U(N+1) = U(N) + H*(X1*E(N+1) + X2*E(N) + X3*E(N-1) + X4*E(N-2))
   UT(N+1) = UT(N) + H*(X1*ET(N+1) + X2*ET(N) + X3*ET(N-1) + X4*ET(N-2))
```

Statements 27 to the end of this page, comprise one step of the Moulton's Predictor-Corrector Procedure.

```
    IF(ABS(U(N+1)) .GT. .25) ER(N-3)=ABS((V-U(N+1))/U(N+1))
    IF(ABS(U(N+1)) .LE. .25) ER(N-3)=ABS(V-U(N+L))/(.5-ABS(U(N+1)))
    IF(ER(N-3) .GT. EF1 .AND. N .LT. 5) GO TO 9
 31 IF(ER(N-3) .GT. EF1 .AND. N .GE. 5) GO TO 8
    IF(ABS(T(N+1)) .LT. .0001) GO TO 10
    IF(T(N+1) .GE. VNI) GO TO 67
    ITT=-T(N+1)/H+.1
    IF(MOD(IABS(ITT),2) .NE. 0 .AND. T(N+1) .LT. 0.) GO TO 45
    IF((ER(1)+ER(2)+ER(3)) .LT. EF2 .AND. N .EQ. 6) GO TO 33
    GO TO 45
 33 H=H*2.
    DO 34J=1,4
    UN(J)=U(J+J-1)
    UNT(J)=UT(J+J-1)
    EN(J)=E(J+J-1)
 34 ENT(J)=ET(J+J-1)
    N=4
    V=UN(4)+H*(Z1*EN(4)+Z2*EN(3)+Z3*EN(2)+Z4*EN(1))
    VT=UNT(4)+H*(Z1*ENT(4)+Z2*ENT(3)+Z3*ENT(2)+Z4*ENT(1))
    TT=T(7)+H
    CALL RT(TT)
    EN(5)=VT
    ENT(5)=F(V,VT)
    UN(5)=UN(4)+H*(X1*EN(5)+X2*EN(4)+X3*EN(3)+X4*EN(2))
    UNT(5)=UNT(4)+H*(X1*ENT(5)+X2*ENT(4)+X3*ENT(3)+X4*ENT(2))
    IF(ABS(UN(5)) .GT. .25) ERRT=ABS((V-UN(5))/UN(5))
    IF(ABS(UN(5)) .LE. .25) ERRT=ABS(V-UN(5))/(.5-ABS(UN(5)))
    IF(ERRT .GT. EF1) GO TO 37
    DO 35J=1,5
    RTT=5-J
    T(J)=TT-H*RTT
    U(J)=UN(J)
    UT(J)=UNT(J)
    E(J)=EN(J)
 35 ET(J)=ENT(J)
    IF(ABS(T(N+1)) .LT. .0001) GO TO 10
    ER(1)=ERRT
    ER(2)=0.
    ER(3)=0.
    GO TO 27
 37 EF2=EF2*.75
    N=6
    H=H*.5
 45 IF(N .EQ. 6) GO TO 61
    GO TO 27
 61 DO 63L=1,6
    T(L)=T(L+1)
    U(L)=U(L+1)
    UT(L)=UT(L+1)
    E(L)=E(L+1)
 63 ET(L)=ET(L+1)
    ER(1)=ER(2)
    ER(2)=ER(3)
    GO TO 27
```

The first 10 statements, constitute the error control of H, the integration interval. It also controls the restarting procedure at T=0.

The sixth statement controls the termination of the solution at t > 15 sec.

The rest of the statements, from 33 on, to 45, comprise the procedure for doubling H

Statement 45 to the end of the page, recirculate the program to the MOULTON'S procedure.

```
67  IF(AB .LE. .01) GO TO 73
    UV=U(N+1) -BETA*OM1*T(N+1)
    IF(UV .GE. (-PI) GO TO 69
    GO TO 71
69  IF (IP  .EQ. 2) IX=1
    IF(IX .NE. 1) GO TO 70
    AB=AB*.5
    VK=VK+AB
    GO TO 6
70  AB=AB*2.
    IP=1
    IF(ABS(A)  .GE.15.  .OR. ABS(B) .GE.20.) GO TO 82
    VK=VK+AB
    GO TO 6
71  IF(IP .EQ. 1) IX=1
    IF(IX .NE. 1) GO TO 72
    AB=AB*.5
    VK=VK-AB
    GO TO 6
72  IP=2
    AB=AB*2
    IF(ABS(A) .GE.15.  .OR. ABS(B) .GE.20.) GO TO 82
    VK=VK-AB
    GO TO 6
73  UV=U(N+1) - BETA*OM2*T(N+1)
    WRITE(t,75) T(N+1),UV,UT(N+1),A,C,B } points[2(0), y(0) ẏ(0)]on the
75  FORMAT(6E16.7)                       } "Spike Surface" written here
    TAT=.5
    IF(IK .EQ. 9 .CR. ABS(A) .GE.14.8 .OR. ABS(B) .GE.17.5) GO TO 91
    AC(IK)=A
    BB(IK)=B
    IK=IK+1
    IF(IK .GE. 4) GO TO 78
    IF(IK .EQ. 3) GO TO 80
    IF(IK .EQ. 2) GO TO 81
78  AXT1=AC(IK-3)-AC(IK-2)
    AXT2=AC(IK-2)-AC(IK-1)
    AXT3=AC(IK-1)-AC(IK-3)
    AYY1=BB(IK-3)-BB(IK-2)
    AYY2=BB(IK-2)-BB(IK-1)
    AYY3=BB(IK-1)-BB(IK-3)
    ATA1=(AC(IK-3)**2)+(BB(IK-3)**2)
    ATA2=(AC(IK-2)**2)+(BB(IK-2)**2)
    ATA3=(AC(IK-1)**2)+(BB(IK-1)**2)
    VDT=2.*(AXT1*AYY2-AXT2*AYY1)
    IF(ABS(VDT)  .LE.  .00001) GO TO 80
    VAT=(ATA1*AYY2+ATA2*AYY3+ATA3*AYY1)/VDT
    VBT=-(ATA1*AXT2+ATA2*AXT3+ATA3*AXT1)/VDT
    VCX=VAT-AC(IK-1)
    VCY=VBT-BB(IK-1)
    VRR=SQRT((VCX**2)+(VCY**2))
    THT=THT*2.
    IF(THT .GT. 1.)THT=1.
79  IF(THT .LT. .0625) GO TO 84
    THETA=.4*THT
    IF(VRR .GE. 2.5) THETA=THT/VRR
```

Statements 67 to 73 comprise the hunting procedure

Statements from 78 to the one just before 80 (next page) comprise the circular extension systems (Block 11 in Fig. 3.3.1 Chapter 3)

```
      VCX2=VAT-AC(IK-2)
      VCY2=VBT-BB(IK-2)
      PIP=1.
      ALPHA=ATAN2((VCY2*VCX-VCY*VCX2),(VCX2*VCX+VCY2*VCY))
      IF(ALPHA .LT. 0.)PIP=-1
      THETA=PIP*THETA
      ALPA1=2.*(SIN(THETA/2.)**2)
      ALPA2=SIN(THETA)
      ALPA3=COS(THETA)
      AAA=VAT*ALPA1-VCY*ALPA2+AC(IK-1)*ALPA3
      BBB=VBT*ALPA1+VCX*ALPA2+BB(IK-1)*ALPA3
      THT=THT*.5
      IF(ABS(AAA) .GE.15.  .OR. ABS(BBB)  .GE.20.) GO TO 79
      SDP=PIP*(VAT-AAA)/VRR
      CDP=PIP*(VBT-BBB)/VRR
      BB(IK)=BBB
      AC(IK)=AAA
      AB=.125
      VK=0.
      IX=3
      IP=0
      GO TO 6
   80 DNP=BB(IK-1)-BB(IK-2)
      DNN=AC(IK-2)-AC(IK-1)
      DDP=SQRT((DNP**2)+(DNN**2))
      CDP=DNN/DDP
      SDP=DNP/DDP
      PIP=1
      BBB=BB(IK-1)+PIP*SOP
      AAA=AC*IK-1)-PIP*CDP
      BB*IK)=BBB
      AC(IK)=AAA
      AB=.2
      IF(ABS(AAA) .GE.15. .OR. ABS(BBB) .GE.20) GO TO 82
      VK = 0.
      IX=3
      IP=0
      GO TO 6
   81 AC(IK)=1.7
      BB(IK)=BB(IK-1)
      AB=.25
      VK =0.
      IX=3
      IP=0
      GO TO 6
   82 WRITE(6,83) T(N+1), UV,UT(N+1),A,C,B,IX
   83 FORMAT(6E16.7,15)
      IF(IK  .EQ. 1) GO TO 91
      BB(IK) =BB(IK-1)+TAT*(BBB-BB(IK-1))
      AC(IK=AC(IK-1)+(AAA-AC(IK-1))*TAT
      TAT=TAT*.5
      AB=.25
      VK =0.
      IF(TAT .LT. .125 ) GO TO 91
      GO TO 6
   84 WRITE(6,85) T(N+1), UV,UT(N+1),A,C,B.THT
```

Straight line extension system. (Block 96 in Fig. 3.3.1)

Second trial point provided. (Block 9a in Fig. 3.3.1)

If hunting procedure fails after a trial point was provided by the straight line extension method, here, the amount of the extension is halved, and the hunting procedure repeated

If hunting procedure fails, this result is printed.

```
85 FORMAT(7E16.7)
91 CONTINUE
   WRITE(6,92) G
92 FORMAT(E16.7)
95 CONTINUE
   STOP
   END

   SUBROUTINE RT(T)
   COMMON/ROOT/A, B, C, G1, G2, OM1, OM, BETA/TOPT/E1, E2, E3, E4
   AZ = ABS(T)
   AZ1 = 1. + AZ
   AZ2 = 1. - AZ
   AEZ = EXP(-AZ)
   X = A*T*AEZ
   Y = (B*AZ1 + C*T)*AEZ
   XT = A*AZ2*AEZ
   YT = ( - B*T + C*AZ2)*AEZ
   PMT = BETA*OM1
   PM = PMT*T
   CPM = COS(PM)
   SPM = SIN(PM)
   GG1 = 2*G1
   GG12 = G1*G2
   E1 = GG1*(XT + PMT*CPM) + GG12*(X + SPM)
   E3 = - GG1*(X + SPM)
   E4 = GG1*(YT + PMT*SPM) + GG12*(Y - CPM)
   RETURN
   END

   FUNCTION F(U,UT)
   COMMON/TOPT/E1, E2, E3, E4
   F = (E1*UT + E2)*COS(U) + (E3*UT + E4)*SIN(U)
   RETURN
   END
```

These two subroutines compute the right side of Eq. 3.2.1.

APPENDIX D

SAMPLE PROGRAM TO COMPUTE EXPECTED EXPECTED NUMBER OF SPIKES PER SECOND FROM SPIKE BOUNDARIES

```
      DIMENSION CV(9), YT(18), Y(10,17), XT(10,17),P(9,17),BETA(4)
      READ(5,1) BETA,CV,YT
    1 FORMAT(4F2.0/9F3.1/F4.0,16F4.1,F4.0
      DO2K=1,18
    2 YT(K)=YT(K)/12
      DO16K=1,4
      READ(5,3)(Y(I,1),I=1,10),((Y(I,J),I=1,10),J=3,15,3),
     1 (Y(I,17),I=1,1C),(XT(I,1),I=1,10),((XT(I,J),I=1,10),J=3,15,3),
     2 (XT(I,17),I=1,10)
    3 FORMAT(10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/
     1 10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/10F5.2/
      PI=4.*ATAN(1.)
      AA=(BETA(K)/(BETA(K)+1.))*SQRT(SQRT(2.)-1.)
      DO7J=1,18
      DO7I=1,10
      GO TO (7,4,7,5,6,7,5,6,7,5,6,7,5,6,7,4,7,7)J
    4 Y(I,J)=.5*(Y(I,J-1)+Y(I,J+1))
      XT(I,J)=.5*(XT(I,J-1)+XT(I,J+1))
      GO TO 7
    5 Y(I,J)=(2.*Y(I,J-1)+Y(I,J+2))/3.
      XT(I,J)=(2.*XT(I,J-1)+XT(I,J+2))/3.
      GO TO7
    6 Y(I,J)=(Y(I,J-2)+2.*Y(I,J+1))/3.
      XT(I,J)=(XT(I,J-2)+2.*XT(I,J+1))/3.
    7 CONTINUE
      DO13IK=1,9
      CNR=CV(IK)
      CRT=SQRT(CNR)
      PN=0.
      DO11J=1,17
      DO111=1,9
      D=XT(I+1,J)-XT(I,J)
      A=(Y(I+1,J)-Y(I,J))/D
      B=(XT(I+1,J)*Y(I,J)-XT(I,J)*Y(I+1,J))/D
      C-A*A+1.
      E=SQRT(C)
      F=A*B/E
      G=B*B/C
      AV1=1.
      AV2=1.
      AV3=1.
      Q2=(A-AA)/E
      XTT1=XT(I+1,J)+AA*Y(I+1,J)
      XTT2=XT(I,J)+AA*Y(I,J)
      SS1=SGN(XTT1)
      SS2=SGN(XTT2)
      SS3=SGN2(XTT1,XTT2)
      ROCK=ROF(YT(J+1),YT(J),CRT)
      SO=SGN(ROCK)
      IF(SO.EQ.O.)GO TO 10
      AVO=ALOG(ABS(ROCK))-ALOG(8.)-ALOG(PI)
      RR1=SS2*ERFC(Y(I,J)*CRT)
      SI=SGN(RR1)
      IF(S1.NE.o.)AV1=AVO+ALOG(ABS(RR1))-CNR*(XT(I,J)**2)
      RR2=SS1*ERFC(Y(I+1,J)*CRT)
      SO-SGN(RR2)
```

[A A is $\dot{\Phi}_m(0)$]

Comments

CV ≡ CNR, carrier-to-noise ratio

YT is $\dot{y}(0)$

Y is $y(0)$

XT is $\dot{x}(0)$

XT(I,J),Y(I,J),

YT(J) are the coordinates of the approximating plane segments.

Intergral (Equation 3.4.4) is evaluated in closed form. Termso of the resulting complicated expression which multiply each other, are evaluated in this program in terms of logarithims, to avoid underflow

```
      IF(S2.NE.O.)AV2=AVO+ALOG(ABS(RR2))-CNR*(XT(I+1,J)**2)
      ABC=(1.+AA*A)
      ABB=AA*B/ABC
      RR3=ERFC(B*CRT/ABC)*SS3
      S3=SGN(RR3)
      IF*S3.NE.O.)AV3=AVO+ALOG(ABS(RR3))-CNR*(ABB**2)
      P(I,J)=SO*(S1*EXP(AVI)-S2*EXP(AV2)+S3*EXP(AV3))
      SQ2=SGN(Q2)
      IF(SQ2.EQ.O.)GO TO 11
      RR4=ROFT(SS3,SS1*(E*XT(I+1,J)+F),SS2*(E*XT(I,J)+F),(F-E*ABB),CRT)
      S4=SGN(RR4)
      IF(S4.EQ.O.)GO TO 11
      AV4=AVO+ALOG(ABS(Q2))+ALOG(ABS(RR4))-G*CNR
      P(I,J)=P(I,J)-SQ2*S4*EXP(AV4)
      GO TO 11
   10 P(I,J)=0.
   11 PN=PN+P(I,J)
      WRITE(6,12)PN,CNR
   12 FORMAT (2E16.7)
   13 CONTINUE
      WRITE(6,14) BETA(K),K
   14 FORMAT(F5.4,15)
   16 CONTINUE
      STOP
      END

      FUNCTION ROF(X,Y,Z)
      XX=X*Z
      YY=Y*Z
      IF((ABS(XX).LE.L.).AND.(ABS(YY).LE.1.))GO TO 1
      GO TO 2
    1 ROF=ERF(XX)-ERF(YY)
      RETURN
    2 IF((XX.GT.1.).AND.(YY.GT.1.))GO TO 3
      GO TO 4
    3 ROF=ERFC(YY)-ERFC(XX)
      RETURN
    4 IF((XX.LT.(-1)).AND.(YY.LT.(-1.)))GO TO 5
      GO TO 3
    5 ROF=ERFC(-XX)-ERFC(-YY)
      RETURN
      END



      FUNCTION ROFT(S,X,Y,Z,C)
      IF(S.EQ.O.)GO TO 1
      GO TO 2
    1 ROFT=ROF(X,Y,C)
      RETURN
    2 IF(S.GT.O.)GO TO 3
      GO TO 4
    3 ROFT=ROF(X,Z,C)-ROF(Y,(-Z),C)
      RETURN
    4 ROFT=ROF(X,(-Z),C)-ROF(Y,Z,C)
      RETURN
      END
```

Note also that $YT(J+1) \equiv \dot{y}(0)+\Delta\dot{y}_+$ and $YT(J) \equiv \dot{y}(0)-\Delta\dot{y}_-$ when referring to Fig.3.4.1 in text.

Result of N for each plane segment summed here.

```
      FUNCTION SGN2(X,Y)
      TT=X*Y
      IF(TT.GT.0.)GO TO 1
      GO TO 2
    1 SGN2=0
      RETURN
      END


      FUNCTION SGN(X)
      IF(X.GT.0.)GO TO 1
      TO TO 2
    1 SGN=1.
      RETURN
    2 IF(X.LT.0.)GO TO 3
      GO TO 4
    3 SGN=-1.
      RETURN
    4 SGN=0.
      RETURN
      END
```

APPENDIX E

Derivation of Integral to Evaluate Expected Number of Spikes per Second.

The output of an FM Discriminator, when integrated is: (Equation 1.4.1 in text)

$$V_{FMD}(t) = \Phi_m(t) + \arctan\left(\frac{x(t)\cos\phi_m + y(t)\sin\phi_m}{1+x(t)\sin\Phi_m - y(t)\cos\Phi_m}\right) \qquad \text{(E1)}$$

Where

$\Phi_m(t)$ = phase of the modulating signal.

$x(t)$ = quadrature low pass equivalent noise.

$y(t)$ = in phase low pass equivalent noise.

Let us introduce, two new random processes,

$$\begin{aligned} x'(t) &= x(t)\cos\Phi_m(t) + y(t)\sin\Phi_m(t) \\ y'(t) &= y(t)\cos\Phi_m(t) - x(t)\sin\Phi_m(t) \end{aligned} \qquad \text{(E2)}$$

$x'(t)$, and $y'(t)$, are independent Gaussian random processes, have zero means, and have variances $\{\sigma^2 = E[x^2(t)] = E[y^2(t)]\}$ the same as $x(t)$ and $y(t)$ if

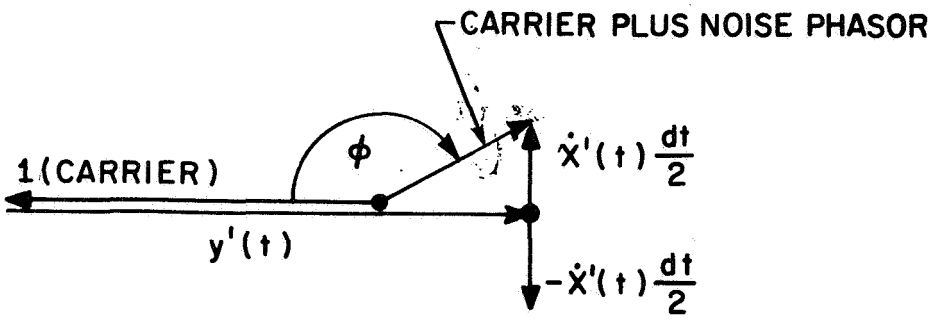
1) $x(t)$ and $y(t)$ are Gaussian, zero mean, variance σ^2, and independent.

2) $\Phi_m(t)$ is any deterministic function of time.

Substituting Equation E2 in E1 we get:

$$V_{FMD}(t) = \Phi_m(t) + \arctan\left[\frac{x'(t)}{1-y'(t)}\right] \qquad \text{(E3)}$$

From Rice[1.9], the probability of receiving a positive spike in time dt, is the probability that $y'(t) \geq 1$, $\dot{x}'(t) \leq 0$, and $+\dot{x}'(t)\frac{dt}{2} < x'(t) < -\dot{x}'(t)\frac{dt}{2}$. This can be seen from the figure below:



positive spike means ϕ increases by 2π radians

Fig. E1.1, Phasor Diagram Illustrating Spike Generation

From this, he gets the result:

$$N_+ = \int_1^\infty \int_0^\infty |\dot{x}'(t)|\, f_{x'\dot{x}'y'}\,(x(t)=0, \dot{x}'(t), y'(t))\, d\dot{x}'(t),\ dy'(t) \qquad \text{(E4)}$$

where: N+ = Expected number of positive spikes/sec.

$f_{x'\dot{x}'y'}\,(x, \dot{x}' y')$ joint density function.

However, in the Most Likely Noise approach, spikes are implied by spike regions in $\dot{x}, y, \dot{y}$ space, or by transformation, $\dot{x}', y', \dot{y}'$ space. We still consider the occurrance of a spike by observing the noise as it crosses the carrier vector during a time interval, dt, thus implying the condition $-|\dot{x}'(t)|\ \frac{dt}{2} < x'(t) < |\dot{x}'(t)|\ \frac{dt}{2}$. (absolute value signs account for the possibility of both positive and negative $\dot{x}(t)$.) However we will now consider that $y'(t)$, $\dot{x}'(t)$ and $\dot{y}'(t)$ lie within a "spike region" S'. Thus the expected number of spikes per second is:

$$N = \int_{S'} |\dot{x}'(t)|\, f_{x'\dot{x}'y'\dot{y}'}\,(x'(t)=0, \dot{x}'(t), y'(t)\ \dot{y}'(t))\, d\dot{x}'(t), dy'(t), d\dot{y}'(t) \qquad \text{(E5)}$$

Since initial time, or the time of the occurrance of a spike is arbitrary, we can set t=0 in Equation E5. Also, from Equation E2,

$$\dot{x}'(0) = \dot{x}(0) + \dot{\Phi}_m(0)\, y(0) \qquad \text{(E6)}$$

Transforming Equation E5 to the x(t), y(t) random processes we get the desired result (Equation 3.4.1 in text)

$$N = \int_S |\dot{x} + \dot{\Phi}_m y|\, f_{x\dot{x}y\dot{y}}\,(x=0, \dot{x}, y, \dot{y})\, d\dot{x}\, dy\, d\dot{y} \qquad \text{(E7)}$$

where S is now the spike region (or regions) in the $\dot{x}$ (o), y(0), $\dot{y}$ (0) space.

APPENDIX F

Comparison of Spike Boundaries of Phase Locked Loops with Different Filter Pole Positions

It can be seen from Figs. F1. 1 and F 1. 2 that the Spike Boundaries, corresponding to a maximally flat response, is slightly further from the origin than are the Spike Boundaries for other responses. This is true for both the second and the third order loops. (Note; for ease of comparison, only the section of the spike boundary for which $\dot{y}(0) = 0$ is shown).

The conclusion to drawn from this is that a maximally flat response should give slightly better response than other responses.

Incidentally, distance of a spike boundary to the origin, should be taken as the radius of a circle, centered at the origin, and tangent to the spike boundary.

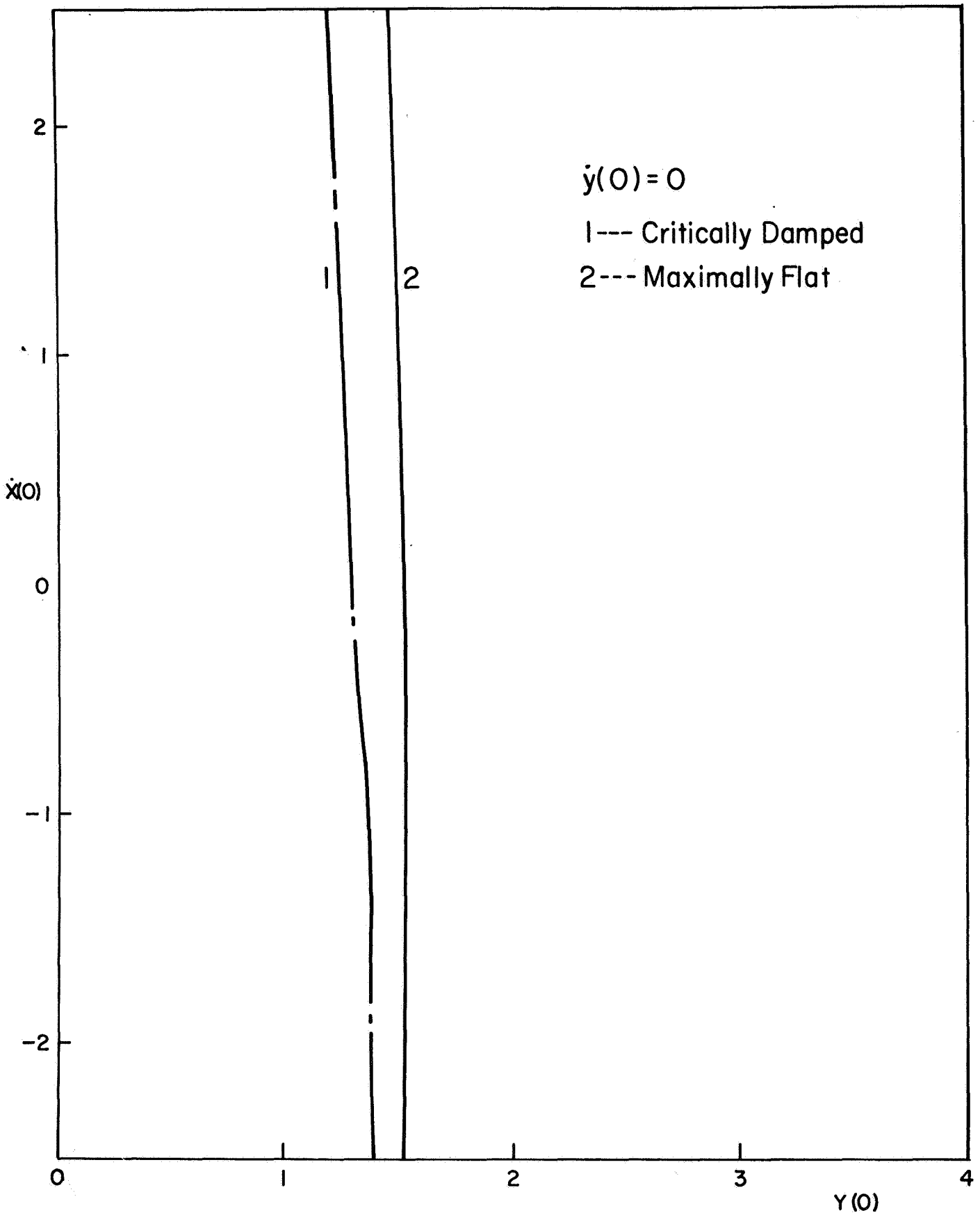


FIG. F1.1 COMPARISON OF MAXIMALLY FLAT AND CRITICALLY DAMPED SPIKE BOUNDARIES FOR SECOND ORDER PLL

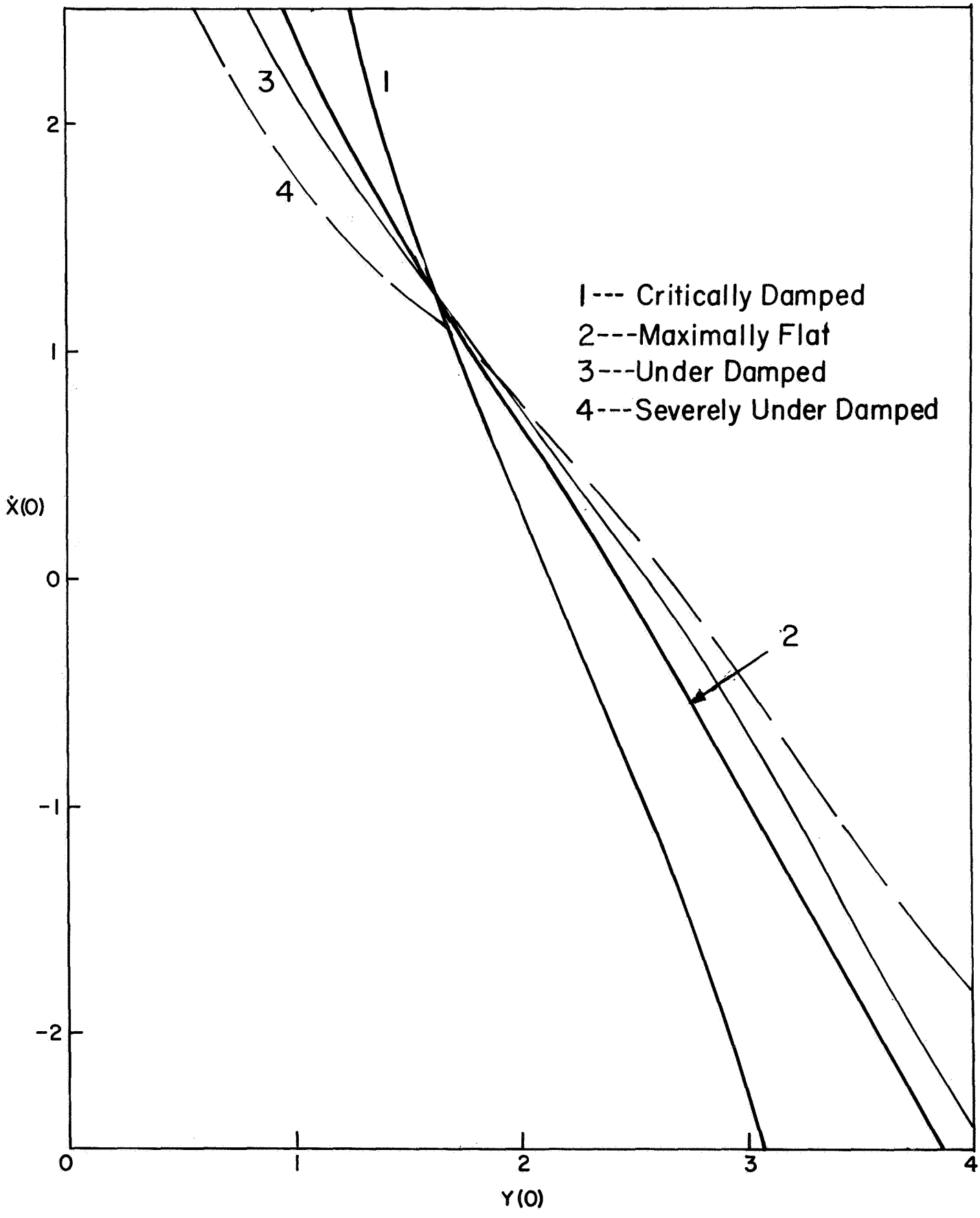


FIG. F1.2 COMPARISON OF DIFFERENT PLL FILTERS IN TERMS OF THEIR SPIKE BOUNDARIES FOR THIRD ORDER PLL