General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

TECHNICAL MEMORANDUM NASA 49

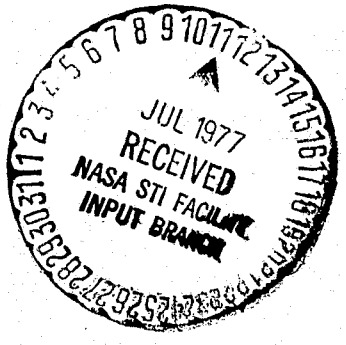
SIMULATION OF DIGITAL PHASE-LOCKED LOOPS

This technical memorandum deals with development of simulation equations for first- and second-order digital phase-locked loops. In addition, examples of loop simulation are given to determine loop performance with respect to several loop parameters.

by

Paul R. Blasche

Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701



May 1977

Supported by

National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia
Grant NGR 36-009-017

I. CONCLUSIONS

A difference equation has been derived that describes the performance of a digital phase-locked loop of order one or two. A computer program has been written that simulates the difference equation and hence simulates the DPLL itself. Using this simulation program, the performance characteristics of the DPLL for an input signal corrupted by noise with respect to several loop parameters were determined. In particular, the steady-state phase error as a function of signal-to-noise ratio for a constant phase input and the mean time-to-lock for a initial phase offset have been determined by means of this simulation program.

II. INTRODUCTION

The lack of suitable mathematical models for digital phase-locked loops of order higher than one makes it necessary to evaluate such loops solely by means of simulation. Two notable derivations by Reddy and Gupta [1] and Gill and Gupta [2] of simulation models have appeared in the literature recently that describe the simulation of first-and second-order digital phase-locked loops of a somewhat general configuration. The purpose of this paper is to apply the results of [1] and [2] to a particular DPLL configuration. Slight changes to the material referenced will be required to accommodate the particular DPLL configuration considered here; therefore, the simulation equations used in this paper will be derived in detail.

In Section IV these simulation equations will be applied to various loops to determine the effects of the several loop parameters upon the output phase. Specifically, the steady-state phase error and the mean time-to-lock for various loops will be examined.

III. SIMULATION EQUATIONS

The DPLL used in the simulation is given in Figure 1. For this loop, the input is sampled by the phase detector at the positive-going zero crossing of the reference clock. The output of the phase detector is quantized to plus or minus one depending on the sign of the sample. Sampling in this manner, the phase difference between the reference clock and the input signal is essentially reduced to a determination of phase lead or lag of the reference clock with respect to the input. The two forward paths represent a filter for the output of the phase detector in the sense that they alter the detector outputs to create a signal suitable for adjustment of the reference clock phase.

Operation of the loop can best be described by considering the special case in which the noise is assumed to be zero. The input signal then is

$$s(t) = \sin \omega_o t$$

let the reference clock be quantized to n distinct phase states so that the reference clock can be described by

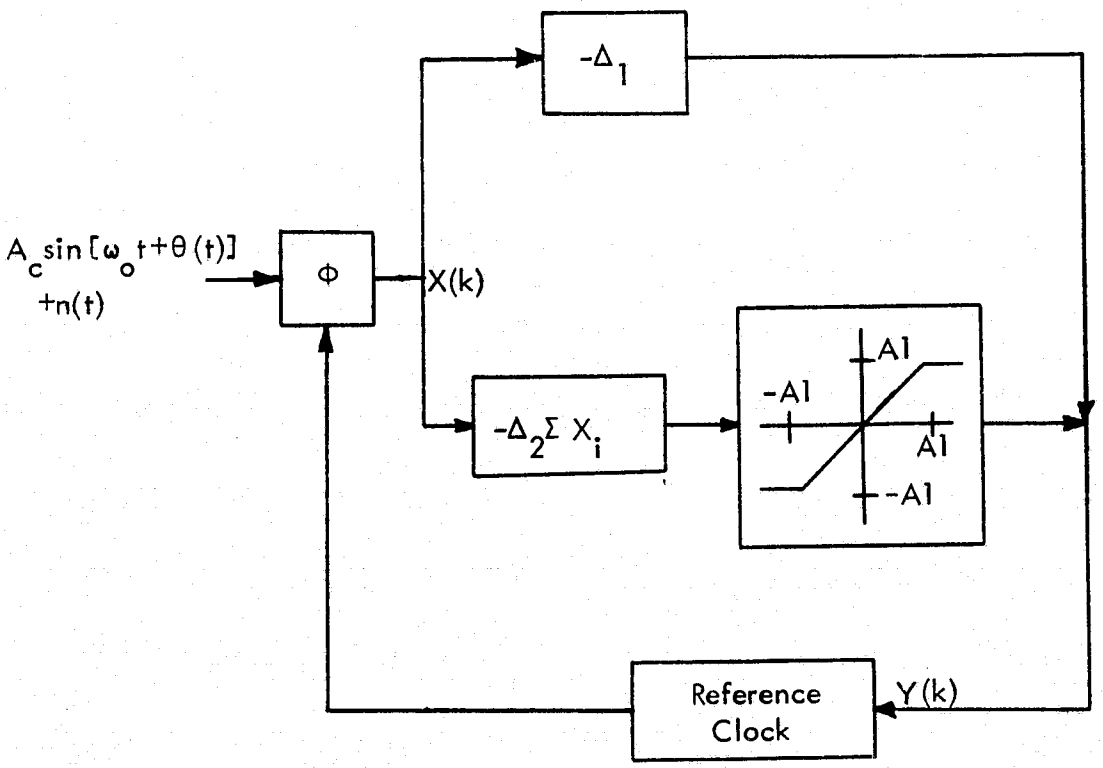


Figure 1. Second-Order DPLL.

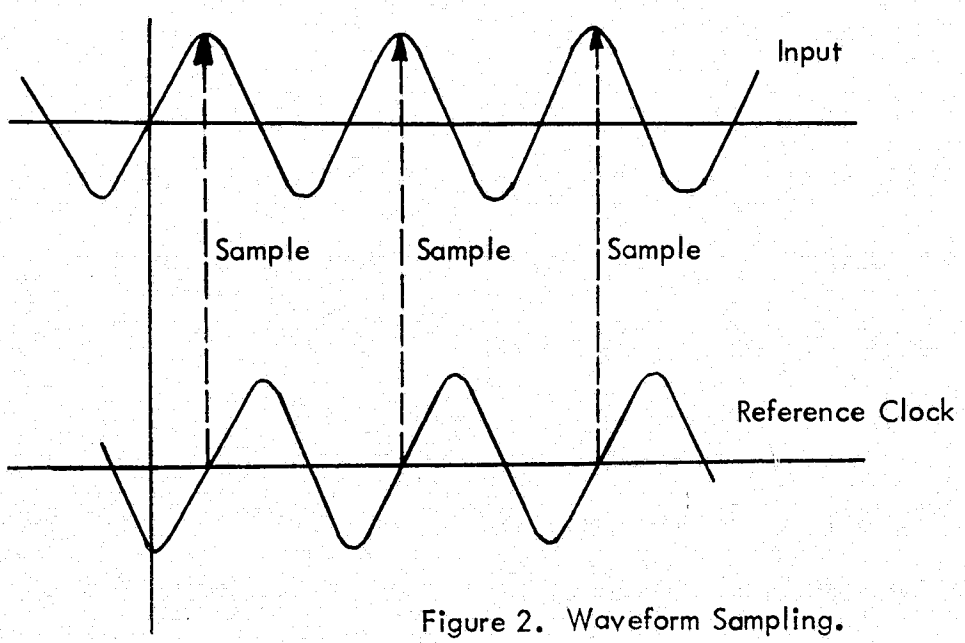


Figure 2. Waveform Sampling.

$$r(t) = \sin\left(\omega_o t - \frac{2\pi i}{N} - \frac{\pi}{N} + \pi\right) \qquad i = 1,2,\ldots N$$

assume that initially the reference clock is given by

$$r(t) = \sin\left(\omega_o t - \pi\,\frac{N-2}{2N}\right)$$

the reference clock then lags the input by

$$\frac{N-2}{2N}\,\pi \quad \text{Radians.}$$

The relative phase of the two signals is depicted in Figure 2. Since the input is sampled at the positive going zero crossing of the reference clock, from Figure 2, the first sample is positive in sign and the output of the phase detector is +1. For the forward path of Figure 1 containing the $\Delta 1$ block, this +1 output is multiplied by the value $\Delta 1$ radians. The phase detector outputs are also summed in the other forward path, multiplied by the constant $\Delta 2$, and then passed through the nonlinearity. For the case considered here, after the first sample is taken the value at the output of the nonlinearity is $\Delta 2$ radians. The signal Y(k) is then used to alter the phase of the reference clock by

$$\Delta 1 + \Delta 2 \quad \text{Rad.}$$

before the next sample is taken. Obviously, $\Delta 1$ and $\Delta 2$ must be related to the quantization of the loop. If at the second sample, the reference clock still lags the input the reference clock phase will be increased by

$$\Delta 1 + 2 * \Delta 2 \qquad \text{Rad.}$$

This continues until the j th sample, at which time the reference clock phase leads the input signal phase and the phase is altered by

$$-\Delta 1 + \Delta 2 \sum_{i=1}^{J} x(i)$$

Finally as the loop achieves lock, the value of the output of the nonlinearity is less than or equal to $\Delta 2$ for all successive samples.

Following the derivation of (2), the k^{th} sample occurs at time t(k) and the time between samples is

$$T(k) = t(k) - t(k-1) \tag{1}$$

Since the output of the phase detector occurs only at the times t(k), then X(k) is

$$X(k) = \text{sgn}\{A_c \sin[\omega_o t + \theta(t(k))] + n[t(k)]\} \tag{2}$$

Also, let the k^{th} sample interval be given by

$$T(k) = T - Y(k) \tag{3}$$

where T is the period of the input signal. Then the actual time of the k^{th} sample is

$$\begin{aligned} t(k) &= \sum_{i=1}^{k} T(k) \\ &= kT - \sum_{i=0}^{k} Y(i) \end{aligned} \tag{4}$$

and the k^{th} phase detector output is

$$X(k) = \text{sgn}\{A_c \sin[\theta(t(k)) - \omega_o \sum_{i=0}^{k-1} Y(i)] + n[t(k)]\} \tag{5}$$

If the phase error is defined as

$$\Phi[t(k)] = \theta[t(k)] - \omega_o \sum_{i=0}^{k-1} Y(i) \tag{6}$$

then

$$\Phi(k+1) - \Phi(k) = \theta(k+1) - \theta(k) - \omega_o Y(k) \tag{7}$$

where

$$\theta[t(k)] = \theta(k) \quad \text{and} \quad \Phi[t(k)] = \Phi(k)$$

But Y(k) is defined as

$$Y(k) = \Delta_1 X(k) + f\{\Delta_2 \sum_{i=0}^{k} X(i)\} \tag{8}$$

so that Equation (7) can be written as

$$\begin{aligned} \Phi(k+1) - \Phi(k) = \theta(k+1) - \theta(k) - \omega_o \Delta_1 \text{sgn}[\Phi(k) + n(k)] \\ - \omega_o f\{\Delta_2 \sum_{i=0}^{k} \text{sgn}[\Phi(i) + n(i)]\} \end{aligned} \tag{9}$$

This final equation then provides a means of simulating the phase error of the loop, and hence the phase output of the loop, for any function of phase input.

IV. DPLL SIMULATION

Using Equation (9), the DPLL of Figure 1 was simulated via a computer program to determine the characteristics of the DPLL for a signal plus wideband Gaussian noise input. In particular, the steady-state phase error was determined for a constant input phase of zero radians and the mean time-to-lock for an initial phase offset were determined. The programs used for the simulation may be found in the Appendix.

Figures 3, 4, and 5 give the steady-state phase error response for different loop constants. In Figure 3, the reference clock phase was quantized to 32 distinct values while the constants $\Delta 1$ and $\Delta 2$ were both set to $2\pi/128$ radians, where 128 is the number of distinct loop states. The value of A1 (the nonlinearity) was varied to values of 0, $\Delta 2$ and $2*\Delta 2$. For the case of $A1 = 0$, the loop was acting as a first-order loop only, while for $A1 = \Delta 2$ and $2*\Delta 2$, the loop is capable of second-order phase updating. From Figure 3, it is apparent that for signal-to-noise ratios less than 20 dB that significant steady-state error is introduced by the second-order updating of the loop. For signal-to-noise ratios higher than 20 dB, the performance of the loop in terms of steady-state error is essentially the same for all three values of A1. The limiting value of the steady-state phase error for signal-to-noise ratios higher than 20 dB is a function of the number of quantized values of the loop output phase. As the signal-to-noise ratio decreases below -40 dB, the loop output error approaches that as would be found with a uniform phase distribution regardless of the value of A1. Figure 4 is a similar plot except that $\Delta 1$ and $\Delta 2$ have been set to $2\pi/256$ radians. Comparison of Figures 3 and 4 reveals that for corresponding values of A1 and SNR, the steady-state phase error given in Figure 4 will be less than or equal to that given in Figure 3. This is caused by the smaller values used for $\Delta 1$ and $\Delta 2$ in the case given in Figure 4 which essentially indicates that the phase detector outputs are filtered more for this case. Figure 5 gives the steady-state phase error for a loop in which the output phase is quantized to 64 distinct values and the constants $\Delta 1$ and $\Delta 2$ are set to $2\pi/512$ radians. Notice that in this case, for signal-to-noise ratios above 20 dB the steady-state error approaches a value one-half that of Figures 3 and 4 and is less than that obtained for Figures 2 and 3 for all corresponding values of A1 and SNR.

The mean time-to-lock for the DPLL's considered above were also determined for a signal-to-noise ratio of 30 dB to check the settling time of the loops. In Figure 6, the mean time-to-lock is given for the same loop constants as were used in Figure 3. Notice that for small values of initial phase error that the second-order loop with $A1 = 2$ actually takes longer to lock than for the other two cases. This is caused by the maximum value building up in the summation block of the second-order forward path and requiring additional samples to settle out. This same characteristic is seen in Figure 7 where the loop constants are the same as used in Figure 4. In Figure 8, the mean time-to-lock for a loop quantized to 64 phase states is given.

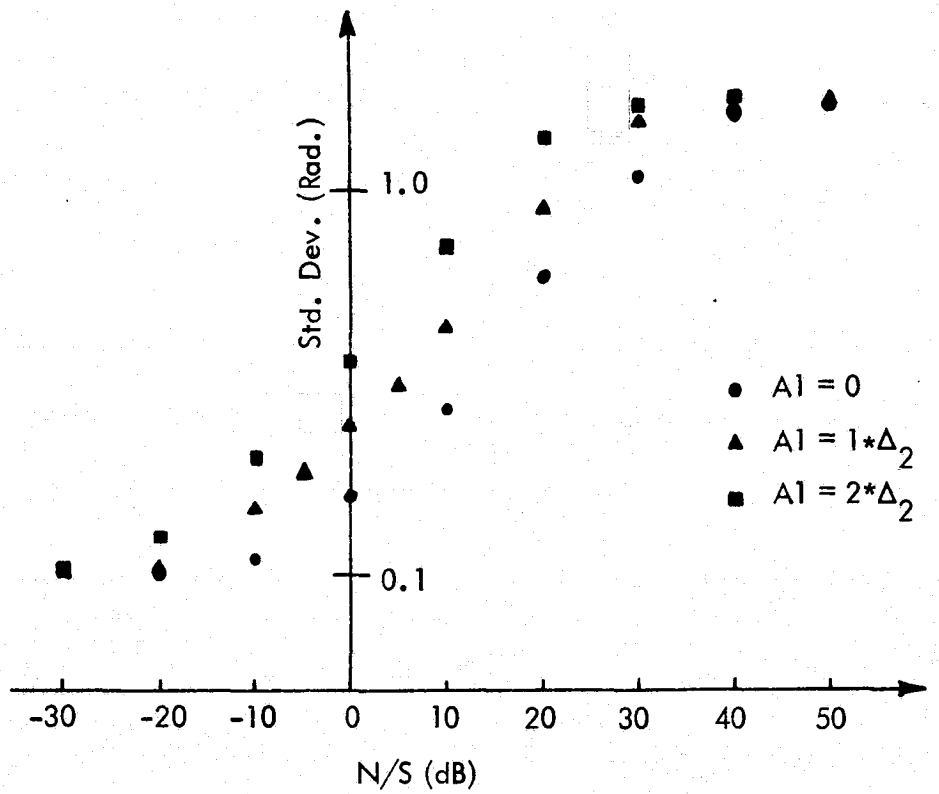


Figure 3. Standard Deviation of Steady-State Phase Output. $M = 4$, $N = 32$, $\Delta_1 = \Delta_2 = 0.049$ Rad.

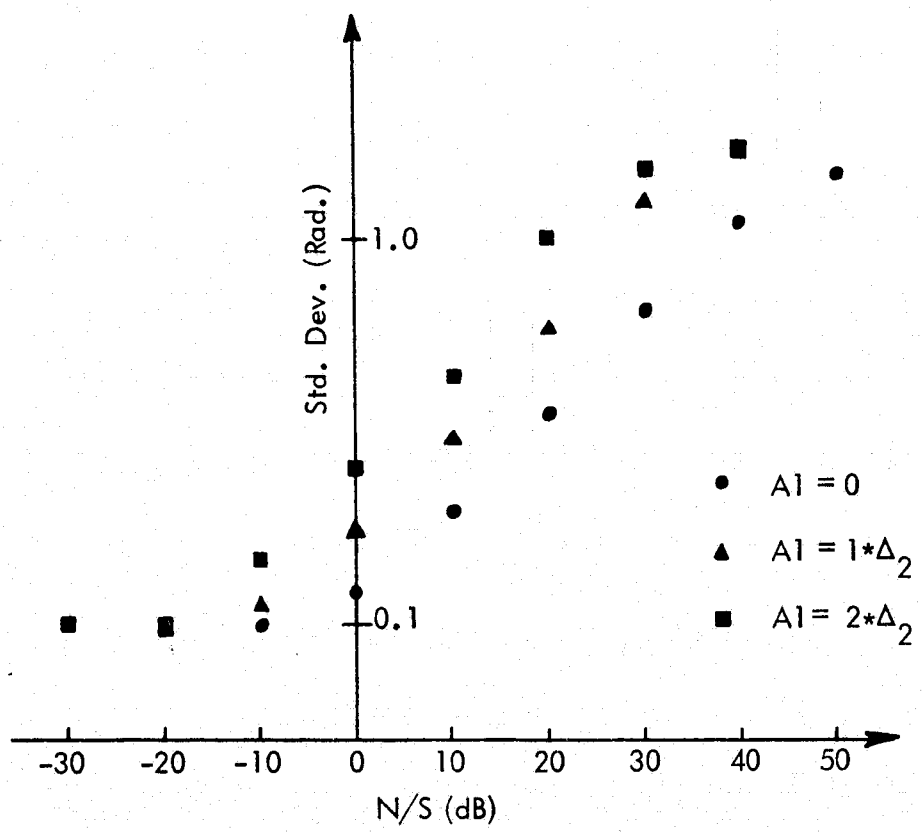


Figure 4. Standard Deviation of Steady-State Phase Output. $M = 8$, $N = 32$, $\Delta_1 = \Delta_2 = 0.0245$.

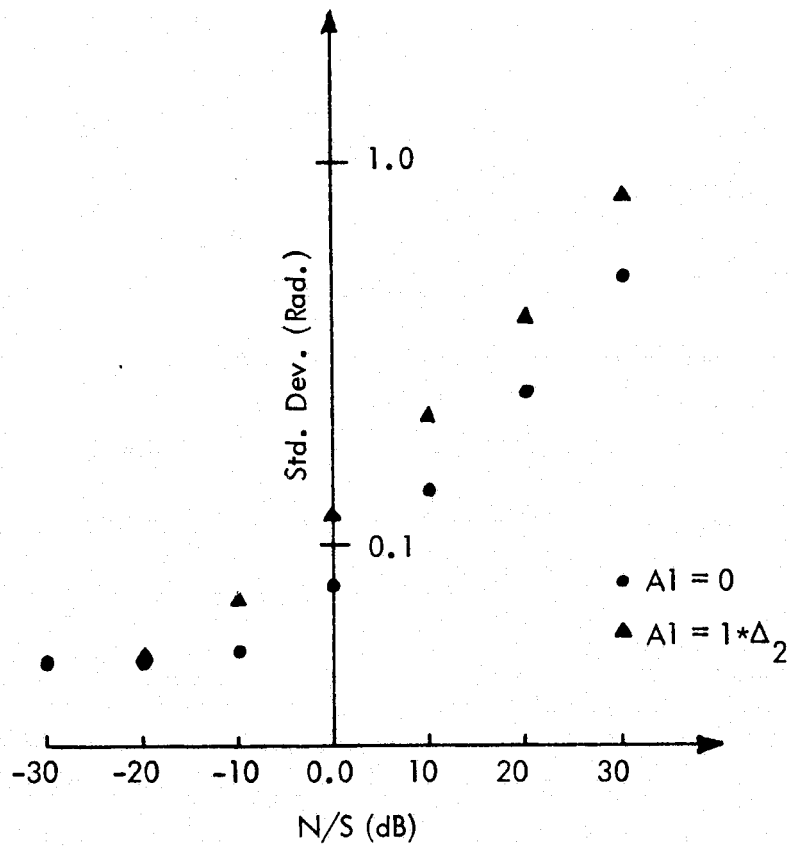


Figure 5. Standard Deviation of Steady-State Phase Output. $M = 8$, $N = 32$, $\Delta_1 = \Delta_2 = 0.0122$.

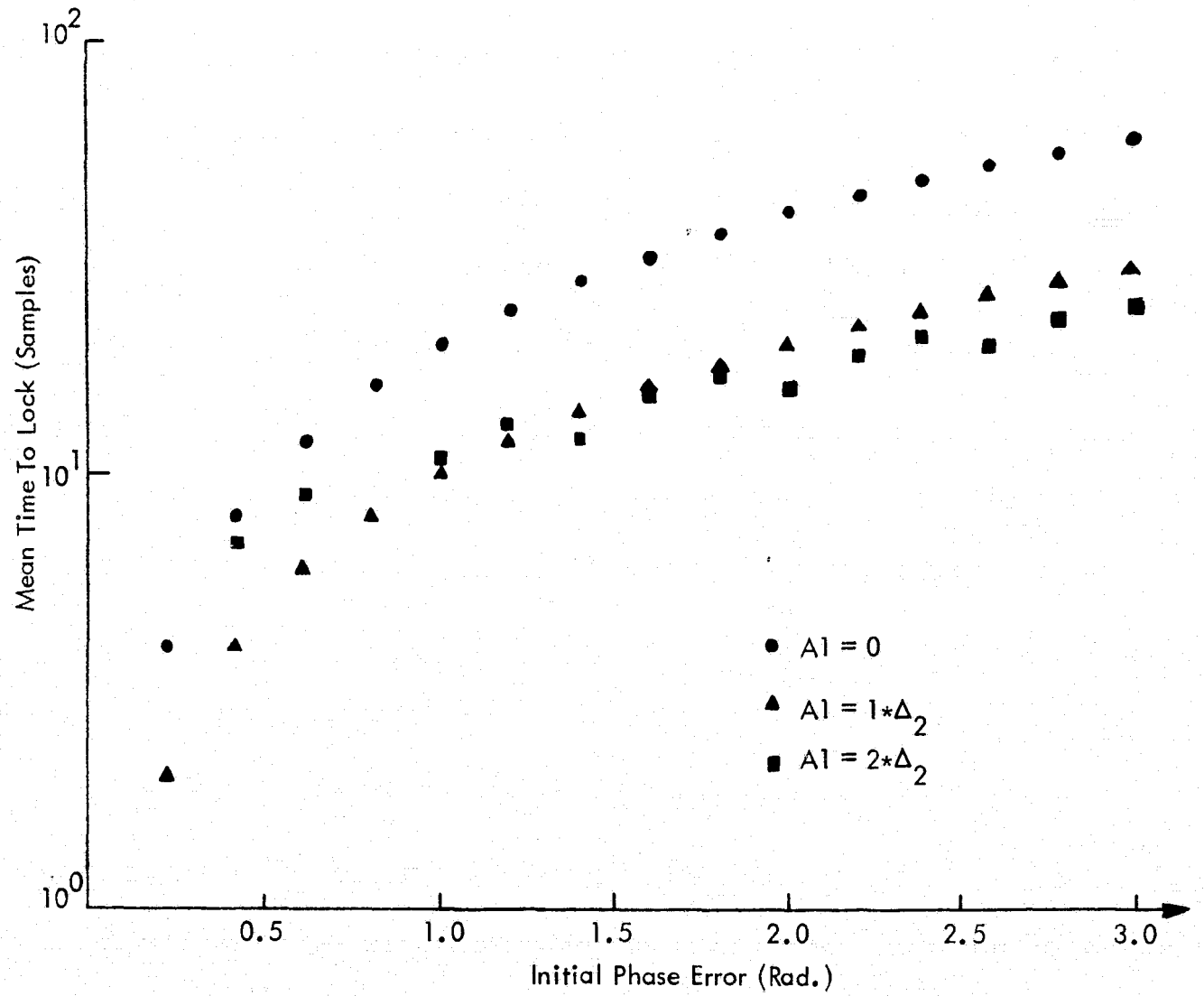


Figure 6. DPLL Transient Response.
$M = 4$, $N = 32$, $\Delta_1 = \Delta_2 = 0.049$
SNR = 30 dB.

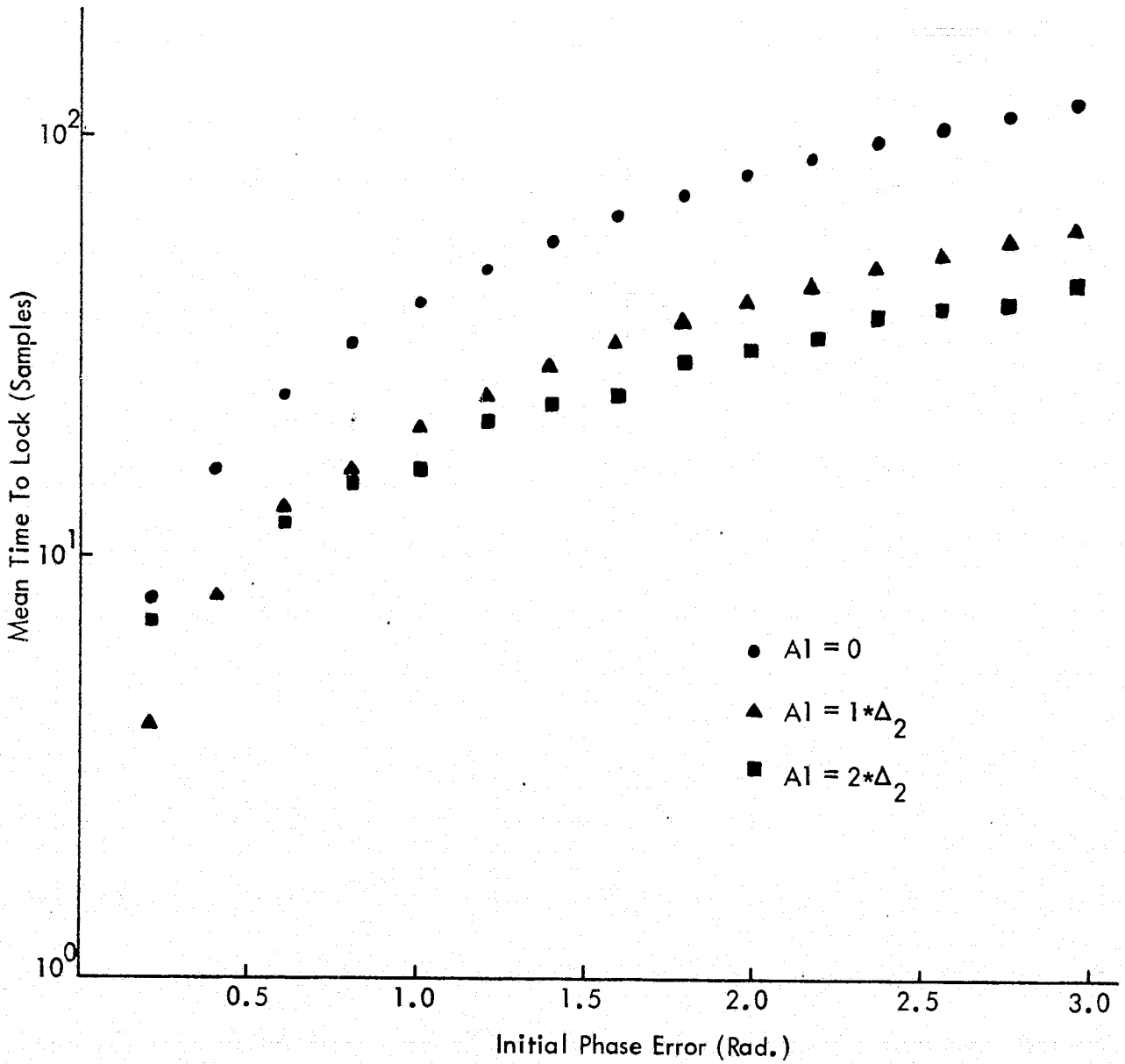


Figure 7. DPLL Transient Response.
M = 8, N = 32, $\Delta_1 = \Delta_2 = 0.0245$
SNR = 30 dB.

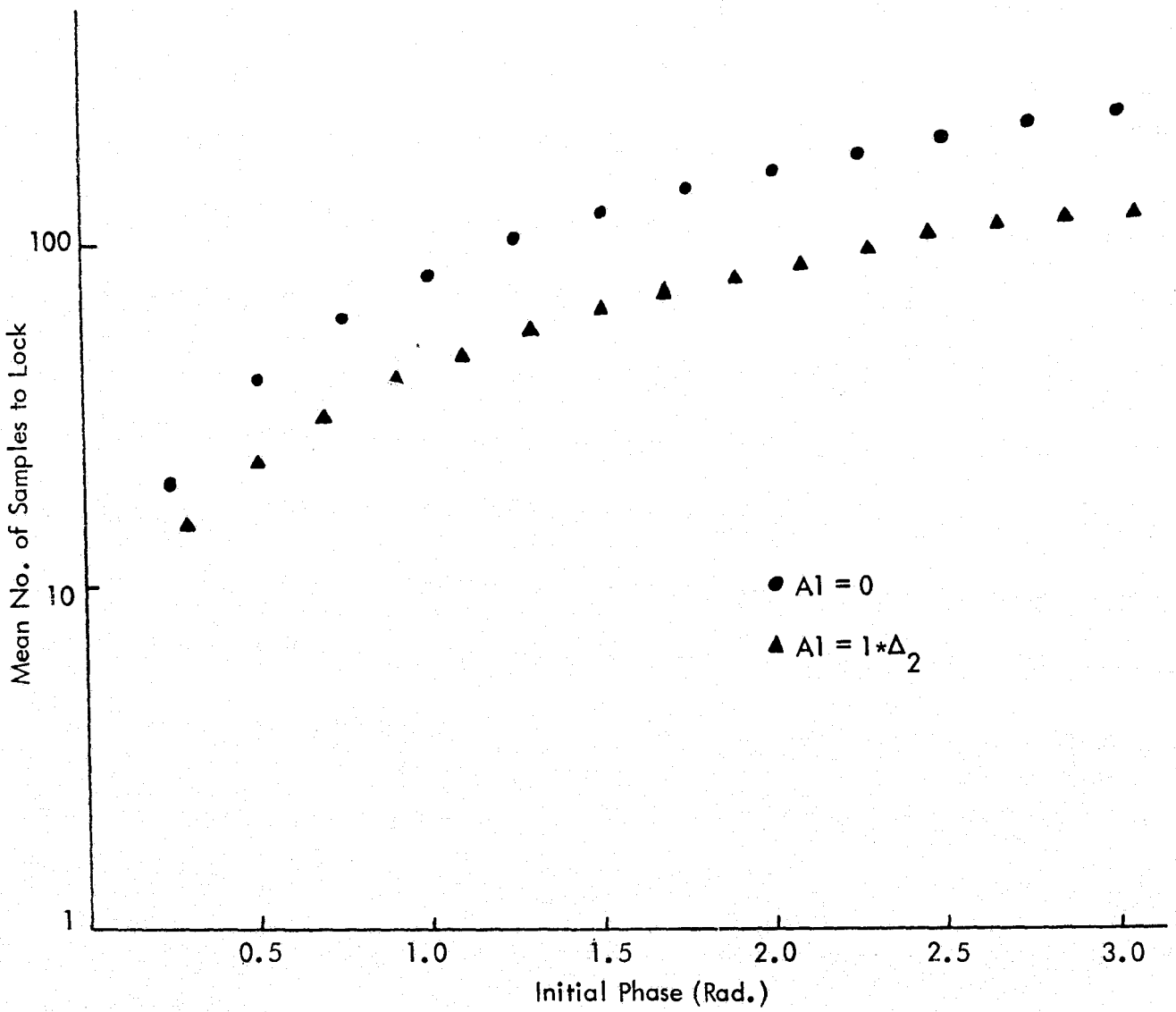


Figure 8. DPLL Transient Response.
M = 8, N = 64, $\Delta_1 = \Delta_2 = 0.0122$
SNR = 20 dB.

V. REFERENCES

[1] Reddy, C. P. and S. C. Gupta, "A Class of All Digital Phase-Locked Loops: Modeling and Analysis", IEEE Trans. on Ind. Elect. and Cont. Inst., Volume pp. 239-251, November 1973.

[2] Gill, G. S. and S. C. Gupta, "On Higher-Order Discrete Phase-Locked Loops", IEEE Trans. on Aero. and Elec. Syst., Volume AES-8, No. 5, pp. 615-623, September, 1973.

APPENDIX

A. Steady-State Error Program

```
      DIMENSION TOT(64)
      DATA NSEED,PI,STD,TOT/137,3.14159265,0.,64*0./
      INTEGER STATE
      WRITE(6,100)
  100 FORMAT(1X,'ENTER M-REGISTER LENGTH')
      READ(5,101)(M)
  101 FORMAT(I5)
      WRITE(6,102)
  102 FORMAT(1X,'ENTER N-REGISTER LENGTH')
      READ(5,101)(N)
      MN=M*N
      WRITE(6,6)
    6 FORMAT(1X,'ENTER SNR IN DB')
      READ(5,104)(SNR)
      A=SQRT(2.)*10.**(SNR/20.)
      WRITE(6,103)
  103 FORMAT(1X,'ENTER W*DEL1')
      READ(5,104)(WDEL1)
  104 FORMAT(F10.0)
      WRITE(6,105)
  105 FORMAT(1X,'ENTER W*DEL2')
      READ(5,104)(WDEL2)
      WRITE (6,7)
    7 FORMAT(1X,'ENTER MAX 2ND ORDER EFFECT')
      READ(5,104)(A1)
      PHI=1.*PI/MN
      K=PHI/(M*WDEL1)
      PHIQ=PI/N
      IF(PHI.LT.0) PHIQ=-PI/N
      IF(K.GT.0) PHIQ=PI/N+K*2.*PI/N
      IF(K.LT.0) PHIQ=-PI/N+2.*K*PI/N
      TEMP2=0.
      IF(TEMP2.EQ.0.) GO TO 5
      TEMP2=ABS(TEMP2)/TEMP2
    5 DO 10 I=1,1000000
      CALL GAUSS(NSEED,1.,0.,V)
      TEMP=A*SIN(PHIQ)+V
      IF(TEMP.EQ.0.) GO TO 20
      TEMP=ABS(TEMP)/TEMP
   20 TEMP2=TEMP2+TEMP
      IF(TEMP2.GT.A1) TEMP2=A1
      IF(TEMP2.LT.-A1) TEMP2=-A1
      PHI=PHI-WDEL1*TEMP-WDEL2*TEMP2
      IF(PHI.GT.PI) PHI=PHI-2.*PI
      IF(PHI.LT.-PI) PHI=PHI+2.*PI
      K=PHI/(M*WDEL1)
      PHIQ=PI/N
      IF(PHI.LT.0.) PHIQ=-PI/N
      IF(K.GT.0) PHIQ=PI/N+2.*K*PI/N
      IF(K.LT.0) PHIQ=-PI/N+2.*K*PI/N
      TEMP1=(PHIQ+PI)/(2.*PI/N)+0.51
      STD=STD+(PHIQ)**2
      STATE=TEMP1
      TOT(STATE)=TOT(STATE)+1
      IF(MOD(I,10000).NE.0) GO TO 10
      RI=I
      STDO=SQRT(STD/RI)
      WRITE(6,1)(I,STDO)
    1 FORMAT(1X,'NO.OF ITERATIONS=',I6,5X,'PHASE OUTPUT STD DEV='
     1,E12.4)
   10 CONTINUE
```

```
      STOP
      END
      SUBROUTINE GAUSS(IX,S,AM,V)
      A=0.
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.)*S+AM
      RETURN
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*0.465661E-9
      RETURN
      END
```

R;

B. Transient Response Program

```
C       1ST AND 2ND ORDER DPLL SIMULATION
        DATA NSEED,PI/137,3.14159265/
        DIMENSION TOT(64)
        INTEGER STATE
        NSEED=139
        WRITE(6,8)
      8 FORMAT(1X,'ENTER M-REGISTER LENGTH')
        READ(5,3)(M)
        WRITE(6,9)
      9 FORMAT(1X,'ENTER N-REGISTER LENGTH')
        READ(5,3)(N)
      3 FORMAT(I5)
        MN=M*N
        WRITE(6,6)
      6 FORMAT(1X,'ENTER SNR IN DB')
        READ(5,2)(SNR)
      2 FORMAT(F5.0)
        A=SQRT(2.)*10.**(SNR/20.)
        WDEL1=2.*PI/MN
        WDEL2=2.*PI/MN
        WRITE (6,7)
      7 FORMAT(1X,'ENTER MAX 2ND ORDER EFFECT')
        READ(5,2)(A1)
        WRITE(8,201)(M,N)
    201 FORMAT('1',' M-REGISTER LENGTH=',I3,5X,'N-REGISTER LENGTH='
       1,I4)
        WRITE(8,202)(WDEL1,WDEL2)
    202 FORMAT(1X,'W*DEL1=',E12.4,5X,'W*DEL2=',E12.4)
        WRITE(8,203)(SNR,A1)
    203 FORMAT(1X,'SNR=',F6.0,5X,'2ND ORDER MAX=',F5.0)
        PHIM=PI/N
        NN=N/2-1
        DO 200 MM=1,16
        VAR=0.
        RMEAN=0.
        DO 100 JJ=1,200
        PHI=PHIM
        PHII=PHI
        K=PHI/(M*WDEL1)
        PHIQ=PI/N
        IF(PHI.LT.0) PHIQ=-PI/N
        IF(K.GT.0) PHIQ=PI/N+K*2.*PI/N
        IF(K.LT.0) PHIQ=-PI/N+2.*K*PI/N
        TEMP2=0.
        IF(TEMP2.EQ.0.) GO TO 5
        TEMP2=ABS(TEMP2)/TEMP2
      5 DO 10 I=1,1000000
        CALL GAUSS(NSEED,1.,0.,V)
        TEMP=A*SIN(PHIQ)+V
        IF(TEMP.EQ.0.) GO TO 20
        TEMP=ABS(TEMP)/TEMP
     20 TEMP2=TEMP2+TEMP
        IF(TEMP2.GT.A1) TEMP2=A1
        IF(TEMP2.LT.-A1) TEMP2=-A1
        PHI=PHI-WDEL1*TEMP-WDEL2*TEMP2
        IF(A1.EQ.0.0.AND.PHI.LE.PI/MN.AND.PHI.GE.-PI/MN) GO TO 11
```

```
      K=PHI/(M*WDEL1)
      PHIQ=PI/N
      IF(PHI.LT.0.) PHIQ=-PI/N
      IF(K.GT.0) PHIQ=PI/N+2.*K*PI/N
      IF(K.LT.0) PHIQ=-PI/N+2.*K*PI/N
   10 CONTINUE
   11 VAR=VAR+I**2
      WRITE(6,4)(I)
    4 FORMAT(1X,I5)
      RMEAN=RMEAN+I
  100 CONTINUE
      RMEAN=(RMEAN/200.)
      VAR=(VAR/200.)-RMEAN**2
      WRITE(8,13)
   13 FORMAT(1X,//,'MEAN NO.OF SAMPLES TO LOCK (200 TRIALS)')
      WRITE(8,14)(PHIM)
   14 FORMAT(1X,'PHIM=',F5.1,' RAD')
      WRITE(8,15)(RMEAN,VAR)
   15 FORMAT(1X,'SAMPLES=',E11.3,' WITH A VAR OF',E11.3)
  200 PHIM=PHIM+2.*PI/N
      STOP
      END
      SUBROUTINE GAUSS(IX,S,AM,V)
      A=0.
      DO 50 I=1,12
      CALL RANDU(IX,IY,Y)
      IX=IY
   50 A=A+Y
      V=(A-6.)*S+AM
      RETURN
      END
      SUBROUTINE RANDU(IX,IY,YFL)
      IY=IX*65539
      IF(IY) 5,6,6
    5 IY=IY+2147483647+1
    6 YFL=IY
      YFL=YFL*0.465661E-9
      RETURN
      END
R$
```