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.

Produced by the NASA Center for Aerospace Information (CASI)
ANALYSIS AND DESIGN OF A SECOND-ORDER DIGITAL PHASE-LOCKED LOOP

A second-order digital phase-locked loop is analyzed by application of a Markov chain model with alternatives. Steady-state loop error statistics and mean transient time are determined for various loop parameters. In addition a hardware digital phase-locked loop was constructed and tested to demonstrate the applicability of the Markov chain model.

by
Paul R. Blasche
Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701

March, 1979

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

(NASA-CR-158393) ANALYSIS AND DESIGN OF A SECOND-ORDER DIGITAL PHASE-LOCKED LOOP (Ohio Univ.) 40 p HC A03/MP A01 CSCL 17G
I. CONCLUSIONS

A specific second-order digital phase-locked loop (DPLL) has been modeled as a first-order Markov chain with alternatives. From the matrix of transition probabilities of the Markov chain, the steady-state phase error of the DPLL was determined. In a similar manner the loop's transient response was calculated for a fading input.

Additionally, a hardware DPLL was constructed and tested to provide a comparison to the results obtained from the Markov chain model. In all cases tested, good agreement was found between the theoretical predictions and the experimental data.

II. INTRODUCTION

The phase-locked loop has long been recognized as a circuit with many important applications, and as such, the description of analog phase-locked loops (APLL's) has become well known as a large volume of material has been published to facilitate their use. In recent years, there has been an increasing use of various types of loops employing discrete elements. Among these have been hybrid PLL schemes that utilize both analog and digital circuitry. Newer loop realizations have been circuits composed entirely of digital elements (DPLL's). The importance of these types of configurations lies in the relative ease of design and construction and, of equal importance, the ease in which such circuits can be maintained.

Unfortunately, the very attributes that make DPLL's attractive from a design standpoint also contribute to the difficulties in the theoretical analysis of DPLL operation. In this area the available literature is relatively thin. For the area of analysis with a fading input, Weinburg and Lin [1] derive steady-state results for general first- and second-order DPLL's, but the analysis is limited by the assumption of small phase error values. In [2] and [3] steady-state loop performance in the presence of Gaussian noise is discussed for specific digital loop configurations. However, for both of these papers the analysis is limited to first-order DPLL's only.

In a previous Technical Memorandum [4], the analysis of a first-order DPLL was performed by modeling the loop as a first-order Markov chain. In the following, the ideas and methods used in this previous Technical Memorandum will be modified to allow a similar approach to be used for the analysis of second-order DPLL's. It will be shown that a second-order DPLL can be modeled as a first-order Markov chain with alternatives and that these alternatives themselves can be thought of as states in a first-order Markov chain. The steady-state distribution of the Markov chain alternatives can be determined and from this distribution it is possible to find the steady-state phase error of the DPLL. The transient response of the loop is determined in a similar manner.

Also detailed in the following is the design and testing of a hardware DPLL capable of either first- or second-order operation. A primary design objective followed in the construction of the hardware loop was to allow easy alteration of important loop parameters.
to provide insight into their effects on loop performance. The general range of these parameters, however, was limited to values thought to be useful for an Omega navigation receiver application. It must be emphasized, however, that neither the theory developed in this TM nor the DPLL design are in any way limited to this particular application.

III. MARKOV CHAIN MODEL OF SECOND-ORDER DPLL

A. Second-Order DPLL Configuration. A block of the second-order DPLL configuration considered here is shown in Figure 1. This configuration differs from the first-order loop presented in [4] only by the addition of the S-bit counter and adder blocks. For this loop, the input is sampled at the positive-going zero crossing of the reference clock and then quantized to a value of ±1. The quantized signal is applied directly to the divide-by-2 \((M + N)\) up/down counter (count up one on plus one and count down one on minus one) giving first-order phase updating. The quantized hard-limiter output is also applied to the input of the divide-by-2 \(S\) up/down counter which has the additional property that the counter will saturate at the limit points ±2 \(S-1\). Once the counters have been updated their values are summed and the sum loaded into the \((M + N)\)-bit counter. The phase of the reference clock is then set to a value determined by the contents of the most significant bits of the \((M + N)\) bit counter.

B. Markov Chains with Alternatives. A Markov chain can be characterized by a system containing a number of distinguishable states (finite or infinite) which transition to a new state denoted \(s_j\) from any present state \(s_i\) depends solely on the present state \(s_i\). For the present, assume that the number of states is \(N\); then for each state \(s_i\), \(i = 1, 2, \ldots, N\) a vector of transition probabilities may be written as,

\[
P_i = (p_{i1}, p_{i2}, \ldots, p_{IN}), \quad i = 1, 2, \ldots, N
\]  

(1)

where \(p_{ij}\) is the probability of transition to state \(s_j\) given that the present state is \(s_i\). Note that the \(N\) transition probability vectors must satisfy the requirement

\[
p_{ij} \geq 0 \quad i, j = 1, 2, \ldots, N
\]

and

\[
\sum_{j=1}^{N} p_{ij} = 1.0, \quad i = 1, 2, \ldots, N
\]

(2)

and are called stochastic vectors. These \(N\) vectors may be arranged in a matrix of size \(N \times N\) to give the matrix of transition probabilities \(P\).
Figure 1. Block Diagram of Second-Order DPLL.
Note that this matrix defines all of the state-to-state transition probabilities for the Markov chain with a finite number of states.

A Markov chain can also be specified for which each state has one or more possible vectors of transition probabilities. For each state \( s_i \), the possible vectors are called alternative vectors and transition from the present state to some new state is governed by one and only one of the alternative vectors associated with the present state. In this case, for each state \( s_i \) there exists \( K_i \) alternative vectors,

\[
p_i = \begin{bmatrix}
  p_{i1} & p_{i2} & \cdots & p_{iN} \\
  p_{i1} & p_{i2} & \cdots & p_{iN} \\
  \vdots & \vdots & \ddots & \vdots \\
  p_{i1} & p_{i2} & \cdots & p_{iN}
\end{bmatrix}
\]

Note that this matrix defines all of the state-to-state transition probabilities for the Markov chain with a finite number of states.

A Markov chain can also be specified for which each state has one or more possible vectors of transition probabilities. For each state \( s_i \), the possible vectors are called alternative vectors and transition from the present state to some new state is governed by one and only one of the alternative vectors associated with the present state. In this case, for each state \( s_i \) there exists \( K_i \) alternative vectors,

\[
k_{p_{ij}} = (k_{p_{i1}}, k_{p_{i2}}, \ldots, k_{p_{iN}}) \quad i = 1, 2, \ldots, N
\]

where \( k_{p_{ij}} \) is the probability that the system will make a transition to state \( s_j \) given that the present state is \( s_i \) and \( k \)th alternative for \( s_i \) is used. As before, each of the alternative vectors of transition probabilities are stochastic vectors and must satisfy the conditions,

\[
k_{p_{ij}} > 0 \quad i, j = 1, 2, \ldots, N \quad \text{and} \quad k = 1, 2, \ldots, K_i
\]

and

\[
\sum_{j=1}^{N} k_{p_{ij}} = 1.0 \quad i = 1, 2, \ldots, N \quad \text{and} \quad k = 1, 2, \ldots, K_i
\]

The vectors may be combined to give the \( K \times N \) stochastic matrix, \( P \).
$$\varphi = \begin{bmatrix}
1_{p_{11}} & 1_{p_{12}} & \cdots & 1_{p_{1N}} \\
2_{p_{11}} & 2_{p_{12}} & \cdots & 2_{p_{1N}} \\
\vdots & \vdots & \ddots & \vdots \\
K_1_{p_{11}} & K_1_{p_{12}} & \cdots & K_1_{p_{1N}} \\
1_{p_{21}} & 1_{p_{22}} & \cdots & 1_{p_{2N}} \\
2_{p_{21}} & 2_{p_{22}} & \cdots & 2_{p_{2N}} \\
\vdots & \vdots & \ddots & \vdots \\
K_2_{p_{21}} & K_2_{p_{22}} & \cdots & K_2_{p_{2N}} \\
\vdots & \vdots & \ddots & \vdots \\
K_N_{p_{N1}} & K_N_{p_{N2}} & \cdots & K_N_{p_{NN}}
\end{bmatrix}$$
C. Markov Chain Model of DPLL. Use of the Markov chain with alternatives in the modeling of second-order DPLL's will be shown with an example. Consider the DPLL of Figure 1 with $M=1$, $N=2$, and $S=3$. Defining the possible states of the $(M+N)$-bit counter as loop states then the loop will have eight loop states while the reference clock is quantized to $2^N = 4$ possible states. These eight loop states form the states of a first-order Markov chain. However, transfer from loop state to loop state is dependent not only upon the present loop state but also on the value of the $S$-bit counter. For example, if the value of the $S$-bit counter is zero, then a $+1$ phase detector output will cause the $(M+N)$ counter to be incremented by a value of $+2$. Similarly, a $-1$ phase detector when the $S$-bit counter is zero will cause the $(M+N)$-bit counter to decrement by a value of 2. However, if the value of the $S$-bit counter is $+1$, then a $+1$ phase detector output will cause the $(M+N)$-bit counter to increment by a value of 3 while a $-1$ phase detector output causes the counter to decrement by a value of 1. Since the possible values of the $(M+N)$-bit counter are associated with the states of a Markov chain, then the possible values of the $S$-bit counter can be associated with alternative actions for each loop state. Thus for the present case of $S=3$, there are seven alternative actions associated with each loop state.

A state diagram of the Markov chain model with alternatives for the DPLL under consideration is shown in Figure 2. Since there are eight possible loop states with each having seven possible alternative actions, there are a total of 56 alternative actions for the entire loop; or in terms of Markov chain notation, there are 56 alternative vectors of transition probabilities. For each loop state the alternative vector to be used is uniquely defined by the value of the $S$-bit counter so that the alternative vectors themselves can be thought of as states in a first-order Markov chain. That is, instead of considering the transitions from loop state to loop state, the transitions from loop alternative to loop alternative are considered. Thus in the state diagram, Figure 2, the loop alternatives are successively numbered $s_1, s_2, \ldots, s_{56}$ and the possible transition from loop alternative to loop alternative is assigned a probability as indicated by the directed arrows. The values of the indicated probabilities are dependent on the state of the loop reference clock only. The method for determining their value was discussed in [4].

Denoting the steady-state probability of loop occupancy of state $s_i$ as $a_i$, then from Figure 2,

$$a_1 = a_{30}p_3 + a_{29}p_3$$

It is clear that there will be 56 such equations which can be put in matrix form as

$$[A][a] = [0]$$

Where

$$[a] = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_{56} \end{bmatrix}$$

$$[A] = \begin{bmatrix} a_{30}p_3 & a_{29}p_3 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \cdots \\ \cdots & \cdots & \ddots & \ddots & a_{56} \end{bmatrix}$$

-6-
for which a unique solution does not exist. However, since \( a_i \) represents a steady-state probability then

\[
\sum_{i=1}^{56} a_i = 1.0
\]  

so that the above homogeneous system in conjunction with this additional constraint can be solved for the steady-state probabilities \( a_i \). If the steady-state probability of the occurrence of the \( k \)th clock state is denoted as \( R_k \) then

\[
R_k = \sum_{i=L}^{L+2M+S-2M} a_i \
\text{for } K = 1, 2, \ldots, 2^N
\]  

where

\[
L = (k-1) \cdot (2^M + S - 2M) + 1
\]  

The distribution of \( R_k \) then establishes the statistics for the phase error of the DPLL.

A similar method is used to find the mean time to loop lock-up for a given initial state. For the loop represented by Figure 2 with a noiseless input signal,

\[
p_1 = p_2 = q_3 = q_4 = 1
\]

and

\[
p_3 = p_4 = q_1 = q_2 = 0
\]

and it is easy to see that regardless of initial state the loop will move to a stable condition involving a transition loop of,

\[
S_{38} \leftrightarrow S_{32} \leftrightarrow S_{19} \leftrightarrow S_{25}
\]

Thus, occurrence of any of these four states can be used as defining the lock-up condition for the DPLL. Denoting \( T_i \) as the mean number of loop transitions to a lock-up condition for some initial state \( s_i \), then a set of difference equations can be written in terms of mean time to loop lock as was done for the determination of steady-state loop probabilities. For instance if the loop is initially in state \( S_7 \), then

\[
T_7 = p_1 T_{55} + q_1 T_{35} + 1
\]  

Clearly, a similar equation can be written for each initial state. Note, for state \( S_{38}, S_{32}, S_{19}, \) and \( S_{25} \) then

\[
T_{38} = T_{32} = T_{19} = T_{25} = 0
\]

since the loop is defined as being initially in a lock condition.
D. Application of the Markov Chain Model. The DPLL's steady-state distribution and transient properties are determined by a solution of the systems of equations given by (8) and (9), respectively. A computer program PBSTGEN was written to establish the state-to-state transitions (i.e., the systems of equations) for arbitrary values of M, N, and S. Using the output of PBSTGEN a second program, PBDPL2, solved the system of equations to find the steady-state phase distribution for the DPLL. Similarly, the output of PBSTGEN by another program, PBDPL2T, to solve for the mean time to lock for an arbitrary initial phase offset. In both PBDPL2 and DPL2T, Jacobi's iterative algorithm was used to solve the system of equations. These programs may be found in Appendix A. Results obtained from these programs will be presented in a later section of this TM.

IV. DIGITAL PHASE-LOCKED LOOP DESIGN

To demonstrate experimentally the validity of the Markov chain model described in the previous section, a hardware DPLL was designed and constructed. A block diagram of the hardware loop is shown in Figure 3. All register lengths shown in the block diagram indicate the maximum values and during testing the actual register lengths were altered to verify the effects of various parameters upon loop performance.

Referring to Figure 3, the phase detector will sample the incoming binary signal of frequency \( f_c \) and then output a count-down signal if the sample is a ZERO or a count-up signal if the input is ONE. The count-up or count-down signal is then applied to both the S-bit saturating counter and the \((M + N)\)-bit counter. Inhibiting logic is included with the S-bit counter so that the counter will saturate at selectable values of \( \pm (2^i - 1) \), \( i = 1, 2, \ldots, 7 \). The sample command also initiates the control logic so that the new value of the \((M + N)\)-bit counter (following the count-up or count-down signal) is loaded into the 12-bit buffer by means of the LOAD signal. After settling, the output of the 12-bit adder will contain the sum of the S-bit saturating counter and the \((M + N)\)-bit counter. This value is then loaded into the \((M + N)\)-bit counter by means of the LOAD signal. Note, the value of the N most significant bits represents the phase estimate of the DPLL. To establish variable phase reference clock, this phase estimate is compared to the value of an N-bit counter being clocked at a \( 2^N \cdot f_c \) rate by an N-bit binary magnitude comparator. Upon coincidence of the two input words the magnitude comparator output takes on a value of ONE. Note that this pulse output occurs at an \( f_c \) rate. Notice also that this loop will operate in the first-order mode simply by inhibiting the LOAD signal to the \((M + N)\) counter. Detailed schematics for the DPLL may be found in Appendix B.

The basic test configuration used for all DPLL experiments is shown in Figure 4. Since it was primarily desired to determine the steady-state phase error and the mean time to lock for an initial phase offset, it was possible to use the same clock source for both the DPLL input and the external reference source. A Sulzer temperature-compensated crystal oscillator was used as the reference source to minimize effects caused by frequency drift of this source. This source was quantized to binary levels and applied to both the DPLL variable phase reference clock and a phase shifting network. The phase shifted clock was then filtered, mixed with noise, and hardlimited to provide a noisy binary input to
Figure 3. Hardware DPLL Block Diagram.
the DPLL. The phase estimate of the loop is continuously stored in a buffer register to provide a parallel digital phase output. This value is also applied to a D/A converter to provide an analog phase output for monitoring purposes.

V. ANALYSIS AND HARDWARE TEST RESULTS

Solutions to the systems of equations describing the Markov chain model for the DPLL were performed for various loop parameter values by means of the computer programs found in Appendix A. Of prime concern was the steady-state phase error and the transient response of the DPLL for a fading input. In addition, data was taken from experiments involving the hardware DPLL to determine both of the parameters for comparison to the results obtained from the Markov chain model.

Figures 5, 6, and 7 are plots of standard deviation of the phase error for the DPLL as a function of noise-to-signal ratio for various loop parameters. The results obtained from the Markov chain model are overplotted with the experimental results and, as can be seen, there is close agreement between the two. Notice for the case of \( S=1 \) the loop is operating in the first-order mode while for \( S \) greater than one the loop is operating second-order. Referring to these three figures it is seen that for constant \( M \) and \( N \) there is a reduction of 5 to 10 dB in loop performance as \( S \) increases in unit steps. Similarly, as \( S \) and \( N \) are held constant, there is a 5 to 10 dB increase in loop performance as \( M \) increases in unit steps.

Figures 8-14 are plots of the loop transient response for various loop parameters and signal-to-noise ratios. Again the experimental results are overplotted on the theoretical predictions. Each experimental data point represents the mean of at least 500 trials and once again there is good agreement between experimental and theoretical data. A particularly significant result may be found from a comparison of Figures 8-10. From these figures it is seen that for small values of initial phase offset, the mean time to lock increases with an increasing value of \( S \). That is, for small phase offsets the first-order DPLL will achieve lock in less time than will a second-order DPLL. This is likewise found to be true when comparing Figures 11-13 and can be a significant factor in selection of loop parameters for an application such as an Omega navigation receiver.

VI. REFERENCES


Figure 8. Loop Transient Response, $M=0$, $S=1$, $N=6$
Figure 9. Loop Transient Response, $M=0$, $N=6$, $S=2$. 

- Mean No. of Samples to Lock

- Initial Phase Offset (RAD.)

- SNR = -20 dB
- SNR = 0 dB
- SNR = +20 dB

- Theoretical
- Δ - Experimental Data

Figure 9. Loop Transient Response, $M=0$, $N=6$, $S=2$. 

-18-
Figure 10. Loop Transient Response, M=0, N=6, S=3.
Figure 11. Loop Transient Response, M=1, N=6, S=1.
Figure 12. Loop Transient Response, M=1, N=6, S=2.
Figure 13. Loop Transient Response, M=1, N=6, S=3.
Figure 14. Loop Transient Response, $M=2$, $N=6$, $S=2$.
VII. APPENDICES

A. Fortran Programs Used for Solution of Markov Chain Model.
   Reference Section III - C and D.
**CALCULATION OF STATE TRANSFER VECTORS FOR DPLL**

**MARPOV CHAIN VUCEL**. THIS PROGRAM TO BE USED IN CONJUNCTION WITH DPLL AND PROPLT.

**ORIGINAL PAGE E OF POOR QUALITY**
CALCULATION OF STATE TRANSFER VECTORS FOR PLL
HARPOV-CHAIN MODEL. THIS PROGRAM TO BE USED IN
CONJUNCTION WITH PROPL2 AND PROPL2.

INTEGER PTR(7164),NTR(7164),IN(7,5,2),ALT,ALT?

SET PLL PARAMETERS:

M=1
N=6
ALT=3
WRITE(K,2)M*,ALT
WRITE(K,2)N*,ALT
2 FORMAT(1X,N=M*,K=K)
MN=M*N
DO 10 I=1,ALT
K=1
DO 20 J=1,MN
10 CONTINUE
L=1
ALT=(ALT+1.)/2.
ALT2=ALT/2.
K=1
DO 40 J=1,ALT2
DO 30 T=1,ALT
R=T
IF(I-ALT) EQ.0.GT.0
50 IF(I.EQ.1) GO TO 31
TK=I-1
JK=J*ALT2+1
IKI=J+1
JKI=J*ALT2-1-I
GO TO 31
51 TK=I
JK=J*ALT2+1
TKI=J+1
JKI=J*ALT2-2
GO TO 36
60 IF(ALT*EQ.1) GO TO 31
IK=I-1
JK=J*+1
IKI=J+1
JKI=J-2
GO TO 31
61 TK=I
JK=J+1
IKI=I
JKI=J-1
GO TO 31
70 IF(I.EQ.0) GO TO 71
   IK=I-1
   JK=J-I+ALT+2+3
   TK1=I+7
   JK1=J-I+ALT2-1
   GO TO 80
71   IK=I-1
   JK=J-ALT2+2
   IK1=I
   JK1=J-ALT2-1
80 IF(JK.GT.RMN) JK=JK-'W'
   IF(JK1.GT.RMN) JK1=JK1-'W'
   IF(JK.LT.1) JK=JK+'W'
   IF(JK1.LT.1) JK1=JK1+'W'
   PTR1(K)=IN(JK,JK)
   PTR1(K+1)=IN(JK1,JK1)
   K=K+2
30 CONTINUE
   IF(MOD(J,4).EQ.1) L=L+1
40 CONTINUE
   NST=K-1
   K=1
   LC=1
   LC1=0
93 DO 90 J=1,NST
   IF(MOD(J,2).NE.0) GO TO 94
   IF(LC.LT.NST) GO TO 95
94 IF(PTR1(I).NE.(11/2)) GO TO 92
   T1(K)=(I+1)/2
   PTR2(K)=(((T1(K)+N-1)/2)+ALT-1)/ALT
   K=K+1
   LC=LC+1
   LC1=1
95 DO 99 LC=LC1
   LC1=0
   GO TO 93
99 CONTINUE
   IF(K.GT.RMN) GO TO 96
   LC1=1
   GO TO 93
96 DO 100 I=2,NST+2
100 PTR2(I)=PTR2(I-1)+N
   IF(ALT.EQ.1) GO TO 101
   MM=-ALT
   DO 110 I=2,NST,'W'
110 PTR2(I)=PTR2(I-1)+N
   MM1=MM-1
   DO 120 I=MM1,NST,'M'
120 PTR2(I)=PTR2(I-1)+N
101 DO 95 T=1,NST
   T2(I)=PTR1(I)
   PTR1(I)=T1(I)
   T1(I)=T2(I)
95 CONTINUE
C STUDY STATE DISTRIBUTION DATA
C PTR1=STATE TO STATE TRANSITION VECTOR.
C PTR2=PROBABILITIES VECTOR FOR STATE TO STATE TRANSITIONS.
C
WRITE(9,1)(PTR1(I),I=1,NST)
WRITE(9,1)(PTR2(I),I=1,NST)
1 FORMAT(1X,20I5)
K=1
DO 130 I=1,NST+2
   T2(I)=K
   T2(I+1)=K+N
   KK=MOD((I+1)/2,N+1)
   IF(KK.EQ.0) K=K+1
130 CONTINUE
C
C MEAN TRANSIENT RESPONSE DATA
C T1=STATE TO STATE TRANSITION VECTOR
C T2=PROBABILITY VECTOR FOR STATE TO STATE TRANSITIONS
C
WRITE(9,1)(T1(I),I=1,NST)
WRITE(9,1)(T2(I),I=1,NST)
STOP
END
FILE: PBDPL? FORTRAN A 2/11/79 OHIO UNIVERSITY AVIONICS ENGINEERING CENTER

DETERMINATION OF LOOP PHASE PDF AND STEADY-STATE ERROR FROM MARKOV CHAIN MODEL.

IMPLICIT REAL*8(A-H,O-Z)
INTEGER POINT1(7164), POINT2(7164), ALI
DIMENSION POLD(3584), PHEW(3584), AC(54), A1(64), A(128), THETA(64)
DIMENSION PHASE(64), CHRM(3584)

SET DLL PARAMETERS
READ(8,7) N,NALT
7 FORMAT(5X,I3,7X,I3,5X,I3)
SNR=-40.
MN=N*N
IST=NALT+N
NST=2*IST
DO 20 I=1,IST
20 POLO(I)=1./IST

READ STATE TRANSFER VECTORS
READ(8,1) POINT1(I), I=1,NST)
READ(8,1) POINT2(I), I=1,NST)
1 FORMAT(1X,25I5)

CALCULATE STATE TRANSFER PROBABILITIES
DO 500 KM=1,I
PI=3.14159265
AC=SQRT(2.*DO)+1.00..(SNR/2.0)
PSI=PI+PI/N
DO 10 I=1,N
RMEAN=AC+SIN(PSI)/SQRT(2.*DO)
AL(I)=0.5+0.5*DERF(RMEAN)
AO(I)=1.-A(L)
THETA(I)=PSI
10 PSI=PSI+2.*PI/N
DO 12 T=1,N
A(I)=AO(I)
A(I+N)=A(I)
12 CONTINUE
JU=1

CALCULATE STEADY-STATE LOOP STATE PROBABILITIES BY ITERATION.
30 DIFF=0.0
JU=JU+1
K=1
J=1
DO 40 I=1,IST
PHEW(I)=A(POINT2(J))*POLD(POINT1(I))*A(POINT2(J+1))
40 CONTINUE
IPOLD(POINT1(J+1))
    IF(PNEW(I),LT,1.0-28) PNEW(I)=0.000
41 J=J+2
40 CONTINUE
    SUM=0.
    DO 50 I=1,IST
50 SUM=SUM+PNEW(I)
    DO 60 T=1,IST
60 PNEW(I)=PNEW(I)/SUM
    DO 61 T=1,IST
61 PHASE(I)=POLD(I)
    DO 70 T=1,IST
70 CN=ABS(PNEW(I)-POLD(I))
    IF(CNS,ST,DIFF) DIFF=CN
70 POLD(I)=PNEW(I)
    IF(JJ.EQ.10000) GO TO 30
    IF(DIFF.GT.1.0E-3) GO TO 31
80 SUM=0.
C
C    CALCULATE STEADY-STATE LOOP PHASE PROBABILITIES
C    AND STANDARD DEVIATION OF LOOP PHASE
C
    STD=0.
    PSI=PI+PI/K
    K=1
    DO 90 T=1,IST
90 SUM=SUM+POLD(I)
    IF(MOD(I,*ALT),NE,1) GO TO 92
    PHASE(I)=SUM
    K=K+1
    STD=STD+SUM*PSI**2
    SUM=0.
    PSI=PSI+2.*DT/2.
    CONTINUE
    STD=DSQRT(STD)
91 WRITE(4,2)
2 FORMAT(*IPOLL PARAMETERS*)
    WRITE(6,6) SNR,NM,ALT
6 FORMAT(*SNR,STAR,ALT*)
9 FORMAT(1X,*SNR=*,F7.1,5X,*NM=*,I3,5X,*ALT=*,I3/)
    WRITE(6,3) STD,JD,DIF
3 FORMAT(1X,*STD DEV=*,F7.1,5X,*NO. OF ITER=*,I6,5X,*DIFF=*,E12.4/) 
    WRITE(6,4) THETA(1),PHASE(1),I=1,N
4 FORMAT(1X,*THETA(1),PHASE(I),I=1,N)
500 SNR=SNR+5
100 STOP
END
DETERMINATION OF MEAN TIME TO LOCK FOR INITIAL PHASE OFFSET
FROM MARKOV CHAIN MODEL

IMPLICIT REAL*8(A-H,O-Z)
INTEGER POINT1(7168),POINT2(7168),ALT
DIMENSION POLD(584),NEW(584),AD(64),A1(64),A12(4),THETA(64)
DIMENSION PHASE(64),PTEM(384)

SET DPLL PARAMETERS

READ(8,7) N,M,ALT
7 FORMAT(3X,I13,7X,I3,3X,I3)
SNR=20.
NWE=N
IST=ALT+N
NST=N+IST
DO 20 I=1,IST
20 POLD(I)=1.
JL=IST/2-(ALT/2)
JLI=JL-(I-1)*ALT
JLU=JL1+(2*M)-1*ALT

READ STATE TRANSFER VECTORS

READ(8,1)(POINT1(I),I=1,NST)
READ(8,1)(POINT2(I),I=1,NST)
1 FORMAT(1X,2*I5)

CALCULATE STATE TRANSFER PROBABILITIES

C=& 500 MM=I,13
PSI=3.14159266
AC=DSQRT(2.00)*10.**(SNR/20.)
PSI=-PI+PI/N
DO 10 I=1,N
RMEAN=AC*OSIN(PSI)/DSQRT(2.00)
A(I)=0.5*OSIN(RMEAN)
AG(I)=I-AIL(I)
THETA(I)=PSI
10 PSI=PSI+2.*PI/N
DO 12 I=1,N
A(I)=AG(I)
A(I+N)=A(I)
12 CONTINUE
JJ=1

CALCULATE MEAN TIME TO LOCK

30 DIFF=0.0
JJE=JJ+1
K=1
J=1
DO 40 I=1,1ST
PNEN(I)=POINT2(J), POLD(POINT1(J))+4(POINT2(J+1))
1POLD(POINT1(J+1))+1.
IF(PNEN(I)<TSTJ-2), PNEN(I)=0.000
41 J=J+2
40 CONTINUE
DO 71 JJ=JL1, JLU, ALT
71 PNEN(JJ)=0.
DO 70 I=1,1ST
CNG=DABS(PNEN(I)-POLD(I))
IF(CNG<ST+DIFF) DIFF=CNG
70 POLD(I)=PNEN(I)
IF(JJ<ER-10000) GO TO 30
IF(DIFF<ST+1.E-5) GO TO 30
80 SUM=0.
K1=ALT/2+1
K2=ALT
J=1
DO 110 I=K1,1ST*K2
POLD(J)=PNEN(I)
110 J=J+1
91 WRITE(6,2)
2 FORMAT(*1*,*OPLL PARAMETER*)
WRITE(6,9) SNR*N*M/ALT
R FORMAT(1X*,SNR=*,E15.6,X*,M=*,I3,X*,N=*,I3,5X,*,ALT=*,I3,1/)
WRITE(6,3) JJ, DIFF
3 FORMAT(1X*,NO. OF ITER=*,E16.5,X*,DIFF=*,E12.4,1/)
WRITE(6,4) (THETA(I)+POLD(I),I=1,N)
4 FORMAT(1X*,THETA DIFF=*,E12.4,5X*,MEAN NO. SAMP=*,E12.4)
500 SNR=SNR-5
100 STOP
END
B. Binary Phase-Locked Loop Design.

The following sections discuss in detail the design of the hardware DPLL used to validate the Markov chain model. The block diagram for the DPLL may be found in Section IV, Figure 3 of the main text.

a. Binary Phase Detector. Referring to the schematic of Figure B-1, the binary phase detector operates by sampling the binary input signal $f_c$ and producing a complemented pulse output on either the count-up or count-down line. This function is produced by three D-type flip-flops as follows. The OPEN SW signal is applied to the clock input of ff B3 while the binary signal $f_c$ is applied to the data input of the same ff. Thus on a positive transition of OPEN SW the value of $f_c$ is latched, giving the sampled value IN at B3's Q output. The two ff's of A6 are initially set to the ONE state so that when SET LAT latches the input values to the ff's (IN and IN), one ff goes to the ZERO state while the other remains in the ONE state. The SET LAT signal is followed by the CLR LAT signal which sets both ff's to the ONE state. Thus the count-up or count-down are produced in a mutually exclusive manner when one of the ff's of A6 toggles HIGH LOW HIGH while the other remains HIGH.

b. S-bit Saturating Counter. The saturating up-down S-bit counter is shown in Figure B-2. The counting function is performed by two serially connected 74193 4-bit synchronous up-down counters. The output states of the counter is detected by a logic network to produce the INHIB UP and INHIB DN signals that will inhibit the UP and DN clock signals respectively. For example, if switches SW1 through SW6 are closed, then for a counter state of 0000 0001 the INHIB UP signal will be TRUE, thus inhibiting the UP clock signal and saturating the counter at that value. Note, a DN clock signal will still count the counter to the 0000 0000 state. Similarly, for all switches closed and the counter in state 1111 1111 the INHIB DN signal will be TRUE and the counter is saturated at that state. If switch SW1 is opened while all other switches remain closed, then the counter saturation states will be 0000 0001 and 1111 1101 (±3). Continuing in this manner, the counter saturation states may be selected to be $\pm 2^{i-1}$, $i = 1, 2, \ldots, 7$ by opening switches SW1 through SW (i-1) while all other switches remain closed.

c. (M + N)-Bit Counter. The schematic for the (M + N)-bit counter along with the buffer register and adder are shown in Figure B-3. The (M + N)-bit counter consists of three serially connected 74193 4-bit up-down counters. Following an UP or DN clock signal the value of the (M + N)-bit counter is loaded into the 12-bit buffer register composed of two 74174 hex D-type flip-flops by the LOAD signal. The values of the 12-bit buffer and the S-bit saturating counter compose the inputs to the adder circuit consisting of three 7483 4-bit binary adders. The output of the 12-bit adder is applied to the preset terminals of the (M + N)-bit counter. If the type select switch is set for first-order operation, the adder output will not be loaded into the (M + N)-bit counter. If a second-order loop is selected, the LOAD signal will be applied to the load inputs of the 74193's thus presetting the (M + N)-bit counter to the value of the 12-bit adder output.
d. Variable Phase reference Clock. Referring to Figure B-4, six selectable contiguous bits from the (M + N)-bit counter are applied to one input side of a binary magnitude comparator formed by two 7485 4-bit magnitude comparators. The other input to the comparator is obtained from the lower N bits of the two series-connected 74193 counters that are being clocked at a rate of $2^N$ time the input frequency. Thus the EQUALS output pin 6 of A31, of the magnitude comparator will be a ONE when the two inputs are equal and this will occur at a rate equal the input frequency $f_c$. The circuit formed by A33 and A34 prevents the sample output from occurring at a rate greater than $f_c$. For example, if pin 8 of A34 is HIGH then when the magnitude comparator detects the equal condition, pin 5 of A34 is set HIGH causing the phase detector to sample the input signal. The SAMPLE signal is reset LOW by the LOADT signal at the end of the loop phase update. However, since the (M + N)-bit counter may have counted up during the last phase update and the counters of the reference clock have also counted up, it is possible for the magnitude comparator to detect equal states occurring at a rate of $2^N * f_c$. Thus it is necessary for the LOADT signal to reset pin 8 of A33 to a ZERO so that another sample cannot occur until the reference clock counter loads a ZERO into that flip-flop.

e. Control Logic. A timing diagram for the necessary control waveforms and the logic schematic used in their generation are given in Figures B-5 and B-6, respectively. Use of the control waveforms has been described in the previous sections.
Figure B-1. DPLL Binary Phase Detector.
Figure B-2. S-Bit Saturating Counter.
Figure B-3. (M+N) Bit Counter with Adder Function.
Figure B-4. DPLL Variable Phase Reference Clock.
Figure B-4. DPLL Variable Phase Reference Clock.
Figure B-5. Control Waveforms.
Figure B-6. Control Logic.
Figure B-6. Control Logic.