SECURE RANDOM NUMBER GENERATION USING CHAOTIC CIRCUITS

by

Greg M. Bernstein and M. A. Lieberman

Memorandum No. UCB/ERL M89/38

17 April 1989

SECURE RANDOM NUMBER GENERATION USING CHAOTIC CIRCUITS

by

Greg M. Bernstein and M. A. Lieberman

Memorandum No. UCB/ERL M89/38

17 April 1989

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Secure Random Number Generation using Chaotic Circuits *

Greg M. Bernstein
Ford Aerospace Corporation
Western Development Laboratories Division
3825 Fabian Way M/S X21
Palo Alto CA 94303

M. A. Lieberman
Dept. of Electrical Engineering and Computer Science
and the Electronics Research Laboratory
University of California
Berkeley CA 94720

March 21, 1989

Abstract

Random number generators are widely used in simulation, testing and communications. Some applications, such as key generation, require a *secure*, that is, unpredictable, source of random numbers. In this paper we show how to use a chaotic circuit as a secure random number generator and give an example using a first order, nonuniformly sampling, digital phase locked loop operating in a chaotic regime.

1 Introduction

Random and pseudo-random numbers are used for purposes such as: test data generation, Monte-Carlo simulation techniques, generation of spreading sequences for spread spectrum communications, and cryptography.

*This work supported in part by Office of Naval Research Contract N00014-84-J-1097 and National Science Foundation Grant ECS-8517364.

The myriad applications put various constraints on how random numbers are generated. A main design criterion is whether the sequence needs to be repeatable; e.g., the pseudo-random spreading sequences used in spread spectrum communications must be repeatable, while for most simulations using random numbers this is not necessary. Repeatable pseudo-random number generators are implemented in digital hardware or software.

The security of a pseudo-random number generator, particularly repeatable generators, is of paramount importance to the field of cryptography, where it is equivalent to the problem of finding a secure encryption method [1, 2]. By the security of a pseudo-random or random number generator we mean, roughly, how difficult it is, based on past values of the sequence, to predict future values of the sequence. The level of difficulty may be defined in computational or probabilistic terms depending upon the type of generator.

For some applications repeatability is not necessary, but security is a major concern. These include key generation and various aspects of key management [1]. A typical application which uses a secure nonrepeatable pseudo-random number generator and an encryption algorithm is the Keyed-Access EPROM reported in [3].

Typically such applications sample noise from reversed biased diodes, oscillator phase noise [4], or other physical phenomena. However, due to the difficulties encountered in dealing with diode noise sources and other *natural* sources, alternative deterministic circuits have been developed [5, 3]. Two issues are immediately raised by these deterministic generators: (1) Is the pseudo-random sequence secure or even sufficiently random? (2) Are there simpler, i.e., easier to implement and smaller, circuits that can serve this same function? We note that the pseudo-random number generator in [3] uses three ring oscillators, a 32-bit shift register and associated support circuitry.

The nonlinear phenomenon of chaos poses a promising alternative for psuedo-random number generation due to its characteristic unpredictable behavior. The connection between random number generation and chaos has been made before. In [6] Tang et. al. noted the similarity between their map approximating a nonlinear forced oscillator and the map describing a linear congruential psuedo-random number generator, and in [7] Oishi and Inoue showed how to use chaotic first order difference equations to generate pseudo-random sequences with a prescribed distribution function. However, the security/predictability issue was not addressed. In this paper we consider the use of chaotic circuits as secure non-repeatable pseudo-random number generators.

2 Chaotic Circuits

2.1 Overview

The simplest circuits that exhibit chaotic behavior can be described, with no approximations, by discrete time mappings. Nonlinear switched capacitor circuits have been used to implement various mappings of an interval or real line that are known to be chaotic, such as the logistic map [8], while a first order DPLL circuit gives circle maps with well known chaotic properties. Forced chaotic circuits include forced relaxation oscillators, such as the forced multi-vibrator circuit of Tang [6] and the simple forced circuit of Hassler et. al. [9]. Finally, we have autonomous continuous time circuits that exhibit chaos, such as the "Double Scroll" circuit of Matsumoto et. al. [10].

The first order DPLL is the simplest synchronization system [11] that exhibits chaos. The block diagram for a first order, nonuniformly sampling, digital phase locked loop is shown in Fig. 1. The input signal is defined by $s(t) = h(\omega_1 t + \theta_0)$, where $h(\cdot)$ is a 2π periodic function. The input signal $s(t)$ has angular frequency ω_1 and initial phase angle θ_0. The sampler is assumed to be ideal and is clocked by the square wave output of the *variable frequency oscillator* (VFO) at time t_n. The output of the sampler block is $s_n = h(\omega_1 t_n + \theta_0)$ and may be a voltage, a current, a digital word, etc., depending on the type of VFO. The VFO consists of a control input which sets the period of the oscillator, $T_{n+1} = t_{n+1} - t_n = g(s_n)$, and a square wave output.

The equation for the $(n+1)$th period, T_{n+1}, is

$$T_{n+1} = g(h(\omega_1 t_n + \theta_0)). \tag{1}$$

Define $\phi_n = \omega_1 t_n + \theta_0$ to be the phase error variable. Then we obtain

$$\phi_{n+1} = \phi_n + \omega_1 g(h(\phi_n)). \tag{2}$$

Let the input signal be $s(t) = A_m \sin(\omega_1 t + \theta_0)$. For an *experimental* DPLL developed in [11], the VFO is a *voltage controlled oscillator* (VCO) whose *frequency* is linearly related to its control voltage: $T_{n+1} = 1/(f_{off} + bs_n)$, where f_{off} is the offset frequency and b is the voltage-to-frequency conversion constant of the VCO. The circle map defined by this DPLL is

$$\phi_{n+1} = \phi_n + \frac{k}{1 - A\sin\phi_n} \equiv f(\phi_n), \tag{3}$$

where $k = \omega_1/f_{off}$ and $A = -bA_m/f_{off}$.

3 Random Number Generation

Fundamental to most definitions of chaos is the concept of *sensitivity to initial conditions*, that is, two trajectories of the system, no matter how closely they start to one another, will eventually diverge. Furthermore, this divergence is of *exponential* order. A measure of the exponential divergence of trajectories in a dynamical system is the Lyapunov exponent, which measures the average rate of divergence of nearby trajectories. A positive Lyapunov exponent indicates chaos.

Using the parameter values $A = -0.25$ and $k = 8.5$ in equation (3), we find that the DPLL is chaotic. Its Lyapunov exponent can be calculated from [12]

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} \log_2 |df/d\phi(\phi_i)|. \tag{4}$$

Letting $n = 100000$ in equation (4) yields $\lambda = 0.760$, a positive Lyapunov exponent, independent of the initial choice of ϕ.

Most chaotic circle maps are *ergodic* and possess an equilibrium invariant distribution. This distribution can be obtained from techniques discussed and justified in [13]. Essentially one iterates the probability distribution for ϕ, i.e.,

$$p_{n+1}(\phi_{n+1}) = \frac{p_n(\phi_{n1})}{|df/dx|_{\phi_{n1}}|} + \frac{p_n(\phi_{n2})}{|df/dx|_{\phi_{n2}}|} + \ldots \tag{5}$$

where $f()$ is the function that takes ϕ_n to ϕ_{n+1}, and ϕ_{n1}, ϕ_{n2}, ... are the points such that $\phi_{n+1} = f(\phi_{n1}) = f(\phi_{n2}) = \ldots$. In Fig. 2 we show the invariant distribution obtained by starting with a uniform density, 1000 points, and iterating equation (5) twenty times. Ergodicity implies that time averages equal space averages. Hence, in addition to equation (4) we can obtain the Lyapunov exponent from

$$\lambda = \int_0^{2\pi} \log_2 |df/d\phi(\phi)| p_{inv}(\phi) d\phi \tag{6}$$

where $p_{inv}(\phi)$ is the equilibrium invariant distribution. Calculated by this method the Lyapunov exponent is $\lambda = 0.759$.

To obtain random bits from the DPLL, we look at the output of the sampler shown in Fig. 1 and assign bit values of zero and one based on whether the output of the sampler is positive or negative. Hence, $\phi \in [0, \pi)$ corresponds to a "1" and $\phi \in [\pi, 2\pi)$ corresponds to a "0", assuming A_m is positive. From the invariant distribution of Fig. 2 we find that

the probability of obtaining a "1" is 0.5136. Although not perfectly symmetrical there exist techniques to rectify this situation [14], if necessary.

We use information-theoretic notions to discuss the predictability of a random number generated from the above circuit. At a given instant of time we possess certain information about the state of our circuit. As time progresses and in the absence of further information, e.g. more observation, this information decreases in a chaotic circuit. Given an experiment, if we assign probabilities p_i to the m possible outcomes, the information associated with an outcome or measurement is defined as

$$H = -\sum_{i=1}^{m} p_i \log_2 p_i. \tag{7}$$

Increasing the accuracy of the measurement increases the information obtained in the following way. Suppose the outcome of the measurement lies in an interval of the real line, and we break that interval up into m equally sized pieces, with the probability of the outcome in any particular piece being $1/m$. Then the information associated with the measurement is:

$$H = -\sum_{i=1}^{m} 1/m \log_2(1/m) = \log_2 m. \tag{8}$$

So, the finer our measurement resolution, the more information we obtain about our system. For the DPLL we can express our knowledge of the state of the system in terms of a probability distribution $p(\phi)$, and with that distribution we can calculate the amount of information we know about the state, relative to the invariant distribution [12, 15]:

$$H_p = \int_0^{2\pi} p(\phi) \log_2[p(\phi)/p_{inv}(\phi)] d\phi \tag{9}$$

The above integral is evaluated over the subset of $[0, 2\pi)$ where $p_{inv}(\phi) \neq 0$.

In [12], Shaw shows that the average rate of information loss in a chaotic system is equal to the Lyapunov exponent. Thus, the number of iterations n_{loss} for complete loss of information H_p is

$$n_{loss} \approx \frac{H_p}{\lambda}, \tag{10}$$

This formula is valid when λ is much less than the initial information; actually, there is a significant reduction in the rate of loss of information as the information asymptotically approaches zero.

Iteration	Prob($\phi \in [0, \pi)$)	Information (H)
0	1.000	0.9612
1	0.5171	0.5132
2	0.4567	0.1807
3	0.4601	0.1227
4	0.5600	0.08184
5	0.5612	0.05517
6	0.4529	0.02937
7	0.5083	0.01627
8	0.5336	0.00803

Table 1: Change in information and probabilities with iterations of a chaotic DPLL. Observation of a "1" bit at iteration 0.

3.1 Continuously Running Circuit

Suppose the chaotic circuit has been running for a long time; also assume that we have no prior knowledge of initial conditions and we have not been looking at the generated bit stream. Under these circumstances the equilibrium invariant distribution of Fig. 2 is the probability density of ϕ. If we make an observation or take a bit — for example, we see a "1"— then this changes the amount of information we have about the system. The probability distribution corresponding to this information is shown in Fig. 3(a). This is just the invariant distribution truncated and rescaled corresponding to our knowledge that $\phi \in [0, \pi)$, i.e., a "1" bit. The information in this distribution can be computed using equation (9). If we iterate the system, (run the circuit) without making subsequent observations, then the information decreases, as shown in Table 1. In addition, we also show how the probability of seeing a "1" changes as we iterate the system. In Figs. 3(b)-(e) we show how the probability distribution of Fig. 3(a) relaxes towards the invariant distribution as the circuit is iterated. Note that since λ is relatively close to the value of the initial information, equation (10) is not applicable; however, from Figs. 3(a)-(e) we can see that the distribution quickly converges to the invariant distribution of Fig. 2. Hence, to use this circuit as a secure random number generator it is best to wait four to eight iterations before

taking another bit rather than simply the two iterations indicated by equation (10).

3.2 Information at Circuit Startup

Now suppose that when we turn the circuit on we know approximately the initial conditions (we can never exactly know the initial conditions due to measurement error or noise). For example, suppose the initial ϕ is in the vicinity of zero, i.e., the initial probability distribution of ϕ is

$$p_{initial}(\phi) = \begin{cases} 100/2\pi & \text{if } |\phi| < 2\pi/100 \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

From equation (9) the initial information is $H_{initial} = 7.37$ bits. In Table 2 we show how this initial information is lost as the circuit runs, in the absence of additional observations.

In Fig. 4 we plot how the information (in the absence of any additional observations) decreases with each iteration. Equation (10) gives $n_{loss} = 9.71$ or approximately 10 iterations. From Table 2, we can see that after 10 iterations the information left is roughly equal to the Lyapunov exponent. From this point, the rate of loss of information decreases asymptotically towards zero; this can be observed in the plot shown in Fig. 4. By $2 * n_{loss}$ (19 or 20) iterations, the initial probability has essentially relaxed to the invariant distribution, and hence, we can safely start taking bits. It is interesting to note that although the initial probability of a "1" is 0.500, we know with complete certainty what the next eight bits will be. Hence, from a security standpoint it is important to wait for the initial probability distribution to relax to the invariant distribution.

4 Practical Issues

One aspect of chaotic circuits not mentioned so far is the sensitive dependence of the Lyapunov exponent on the value of system parameters. For our application we are interested in determining where in parameter space a given circle map has a positive Lyapunov exponent. In Fig. 5(a) we plot, with black squares, points in the k, A plane where the map of the experimental DPLL has a Lyapunov exponent greater than 10^{-3}. We can see boundaries of regions containing periodic orbits; for example, the region $2\pi(1 - A) \leq k \leq 2\pi(1 + A)$ contains a fixed point which gives the main white section in the middle of the figure. If we examine closely a region in parameter space containing positive Lyapunov exponents (see

Iteration	Prob($\phi \in [0, \pi)$)	Information (H)
0	0.5000	7.3679
1	1.0000	6.3039
2	1.0000	4.8427
3	0.0000	4.8030
4	0.0000	4.2232
5	1.0000	4.1946
6	1.0000	3.8269
7	1.0000	3.3719
8	0.0000	2.5883
9	0.1601	2.5470
10	0.5641	1.3303
11	0.1602	1.1035
12	0.5130	0.5482
13	0.3637	0.4969
14	0.6748	0.2954
15	0.6056	0.1406
16	0.5076	0.08795
17	0.5055	0.05270
18	0.4970	0.03467

Table 2: Change in initial information and probabilities with iterations of a chaotic DPLL

Fig. 5(b)), we observe thin white regions; these are periodic windows that possess negative exponents. For our application we must operate in a region having relatively large positive Lyapunov exponent, far from significant stable periodic orbits. As long as the periodic windows are thin enough, experimentally we will never notice them due to the fluctuations in the system parameters.

A schematic of the actual PLL circuit built is shown in Fig. 6. The Lyapunov exponents determined from experimental data are shown in Fig. 7(a). The parameter k was varied by changing the frequency of the input signal. The algorithm of Wolf et. al. [16] was used to estimate the Lyapunov exponent from the measured data series. In Fig. 7(b) we show the Lyapunov exponents calculated directly from the mapping. The agreement between simulation and experiment is good except for a mismatch in the parameter values which give negative Lyapunov exponents. This is due to errors in measuring the parameter values in the experimental circuit.

Additionaly, one needs to be concerned with the changes in the invariant distribution as circuit parameters change. The Lyapunov exponent is used as a rough indicator of this, since a change in Lyapunov exponent from positive to negative changes an invariant distribution that is nonzero on at least some interval to an invariant distribution that is a finite sum of delta functions.

5 Conclusion

We have suggested a class of circuits for generating secure psuedo-random numbers and estimated the security of these generators from the information loss property of chaotic systems. For a generator implemented using a chaotic DPLL, we considered two important cases. (i) Given no prior information concerning the initial conditions of a continuously running circuit, we established how long one should wait after taking a bit before one can securely take another bit. (ii) Given knowledge of the initial conditions at startup (up to measurement and noise uncertainty), we showed how long one should wait before starting the bit sampling.

References

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions*

on Information Theory, vol. IT-22, November 1976.

[2] A. C. Yao, "Theory and applications of trapdoor functions," in *23rd Annual Symposium on Foundations of Computer Science*, pp. 80–91, November 1982.

[3] L. Letham, D. Hoff, and A. Folmsbee, "A 128K EPROM using encryption of pseudorandom numbers to enable read access," *IEEE Journal of Solid-State Circuits*, vol. SC-21, October 1986.

[4] AT&T, *AT&T Component Products Selection Guide.* AT&T, 1987.

[5] J. D. Long, "VCOs generate selectable pseudo-random noise," *Electronics*, vol. 51, September 1978.

[6] Y. S. Tang, A. I. Mees, and L. O. Chua, "Synchronization and chaos," *IEEE Transactions on Circuits and Systems*, vol. CAS-30, September 1983.

[7] S. Oishi and H. Inoue, "Pseudo-random number generators and chaos," *The Transactions of the IECE of Japan*, vol. E 65, September 1982.

[8] A. B. Rodriguez-Vazquez, J. L. Huertas, and L. O. Chua, "Chaos in a switched capacitor circuit," *IEEE Transactions on Circuits and Systems*, vol. CAS-32, October 1985.

[9] A. Azzouz, R. Duhr, and M. Hassler, "Transition to chaos in a simple nonlinear circuit driven by a sinusoidal voltage source," *IEEE Transactions on Circuits and Systems*, vol. CAS-30, December 1983.

[10] T. Matsumoto, L. O. Chua, and M. Komuro, "The double scroll," *IEEE Transactions on Circuits and Systems*, vol. CAS-32, pp. 797–818, August 1985.

[11] G. M. Bernstein, *Nonlinear Oscillations, Synchronization and Chaos.* PhD thesis, University of California, Berkeley, 1988.

[12] R. Shaw, "Strange attractors, chaotic behavior, and information flow," *Z. Naturforsch.*, vol. 36a, pp. 80–112, 1981.

[13] A. J. Lichtenberg and M. A. Lieberman, *Regular and Stochastic Motion.* Springer-Verlag, 1983.

[14] M. Santha and U. V. Vazirani, "Generating quasi–random sequences from slightly–random sources," in *25th Annual Symposium on Foundations of Computer Science*, pp. 434–439, October 1984.

[15] C. E. Shannon and W. Weaver, *The Mathematical Theory of Communications.* University of Illinois Press, 1949.

[16] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining lyapunov exponents from a time series," *Physica 16D*, pp. 285–317, 1985.

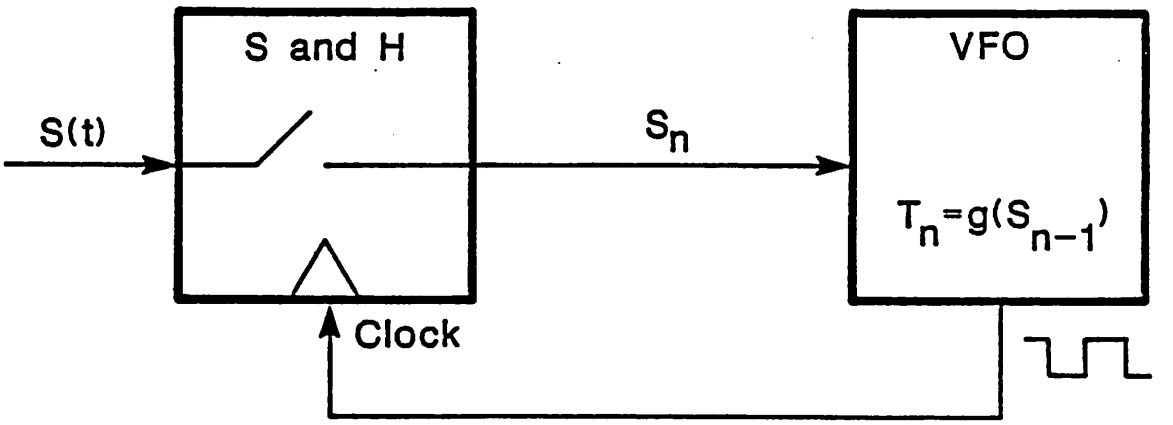

Figure 1. Block diagram of a nonuniformly sampling first order digital phase locked loop.

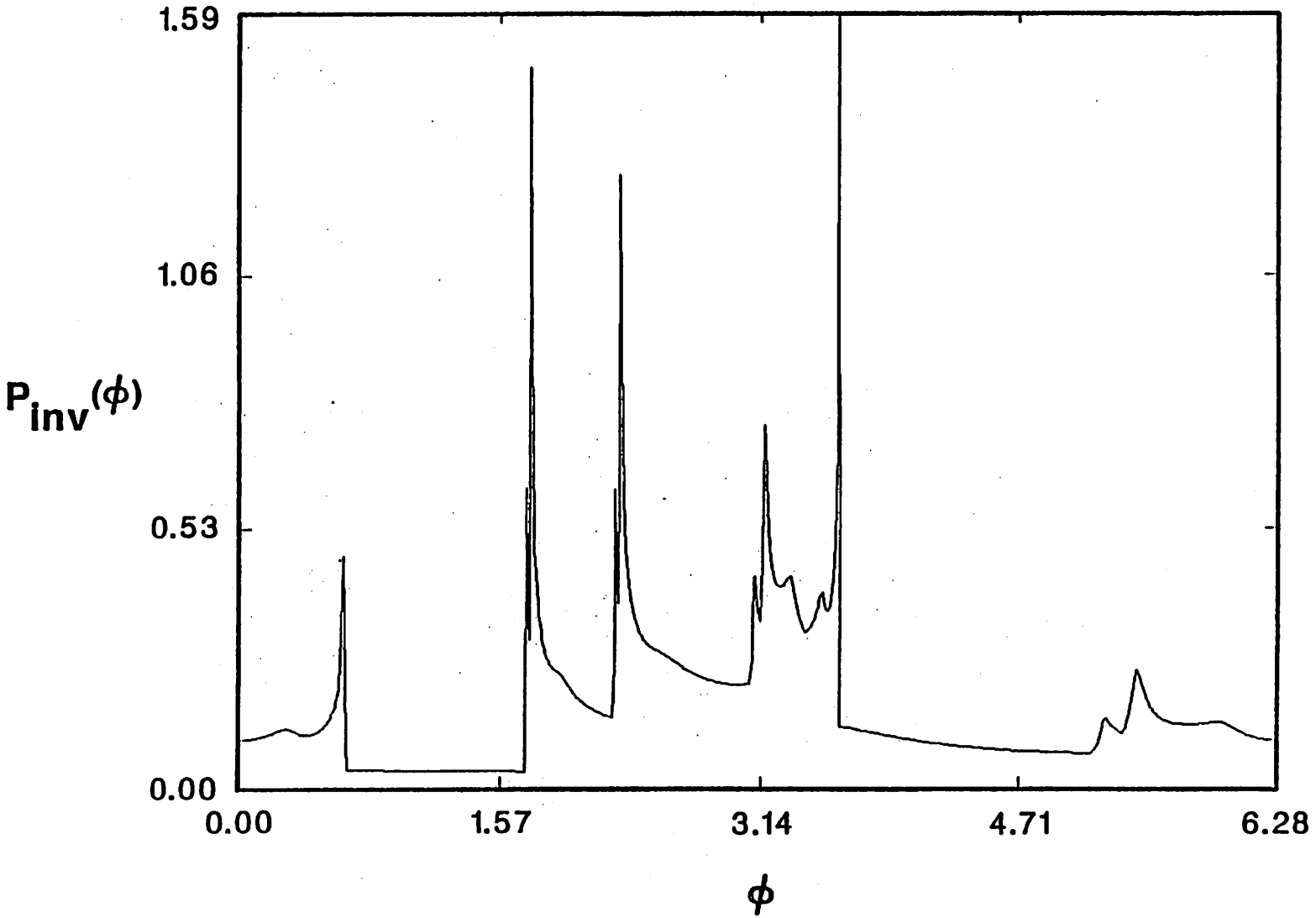

Figure 2. Invariant distribution (density) for the first order DPLL with $A = -0.25$ and $k = 8.5$.

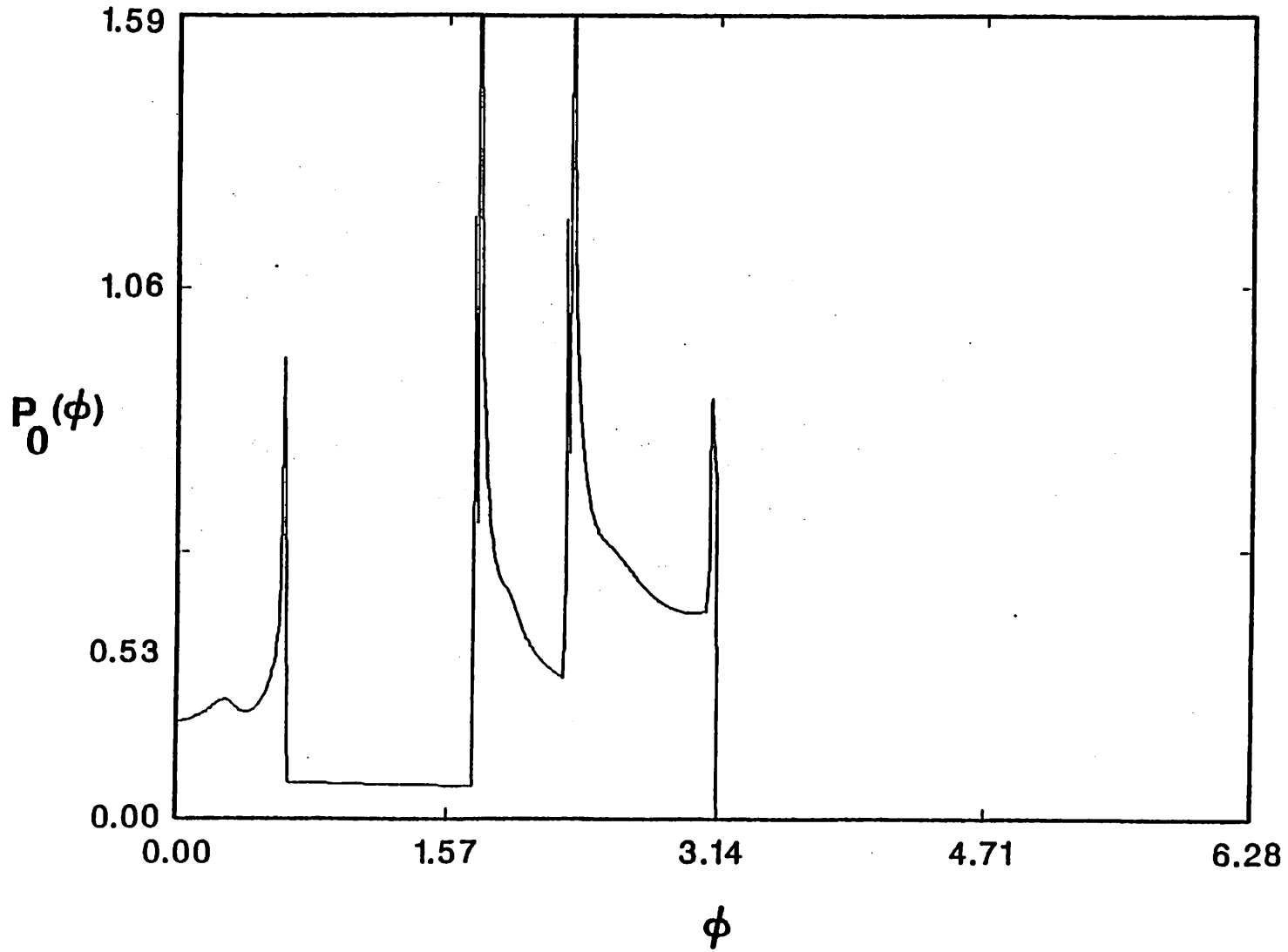

Figure 3. (a) Distribution (density) of the DPLL based on observation of a "1". (b)-(e) Subsequent distributions after iterating the circuit in the absence of additional observations. Distributions relax to the invariant distribution of Fig. 2.

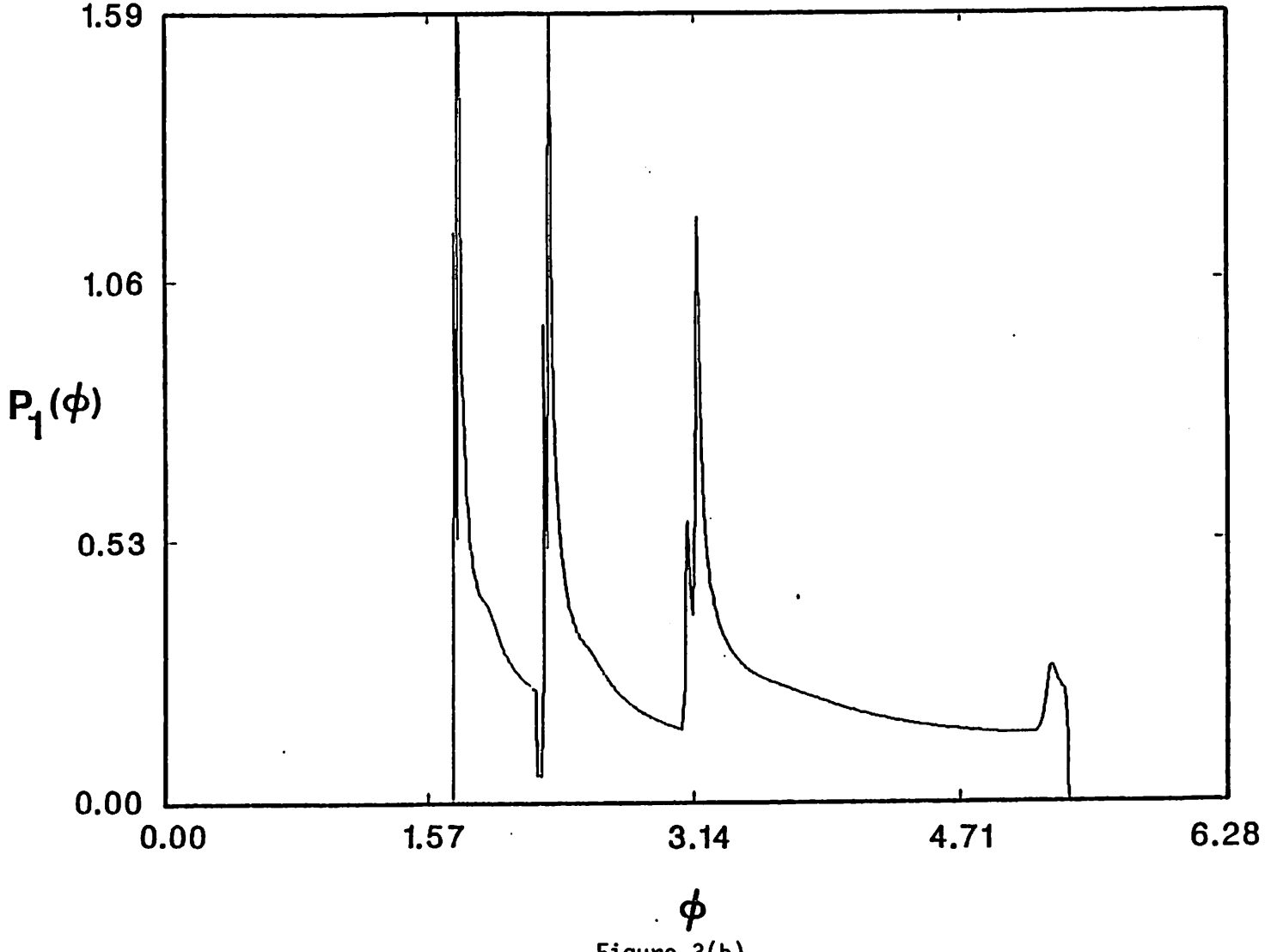

Figure 3(b)

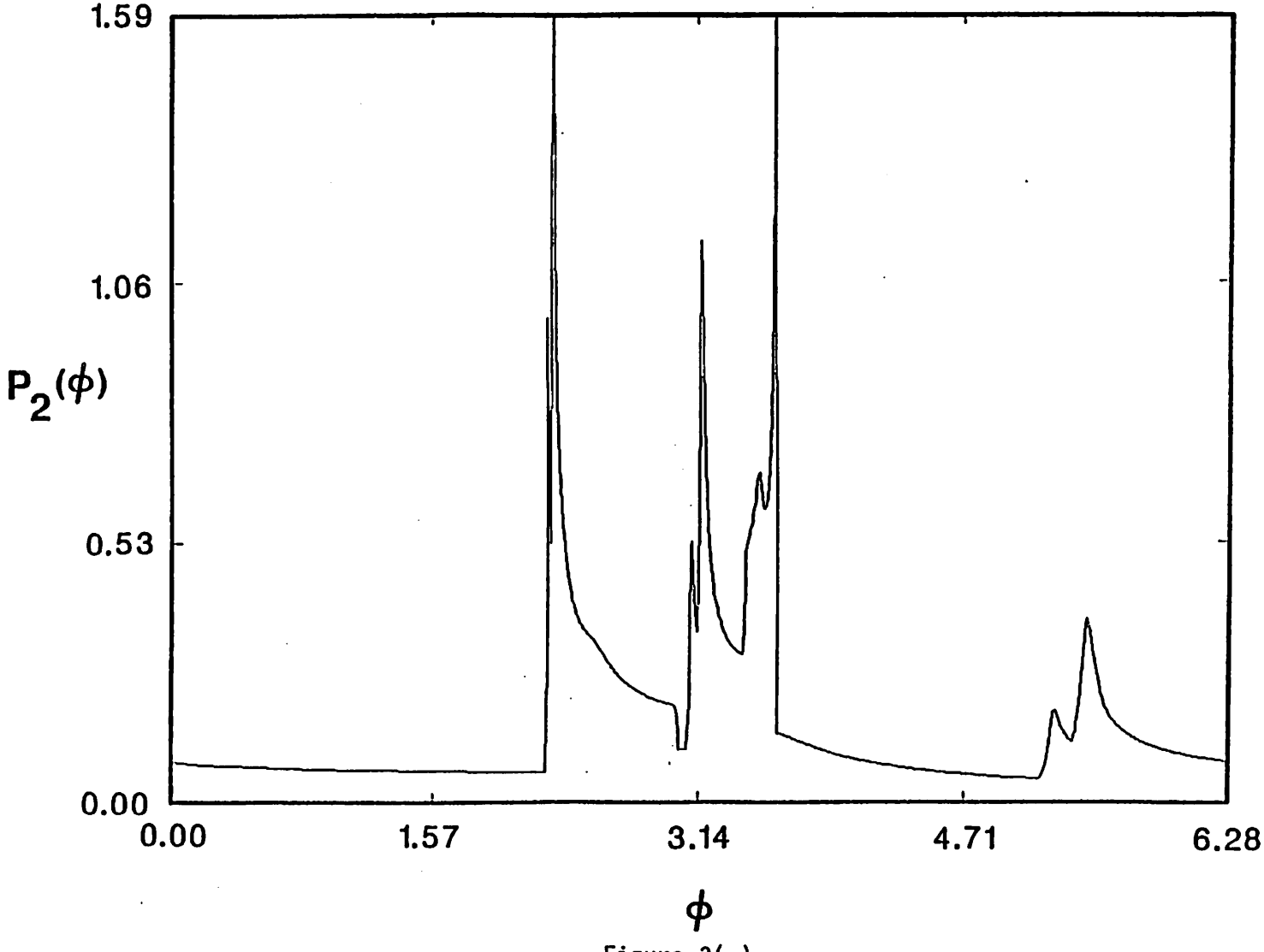

Figure 3(c)

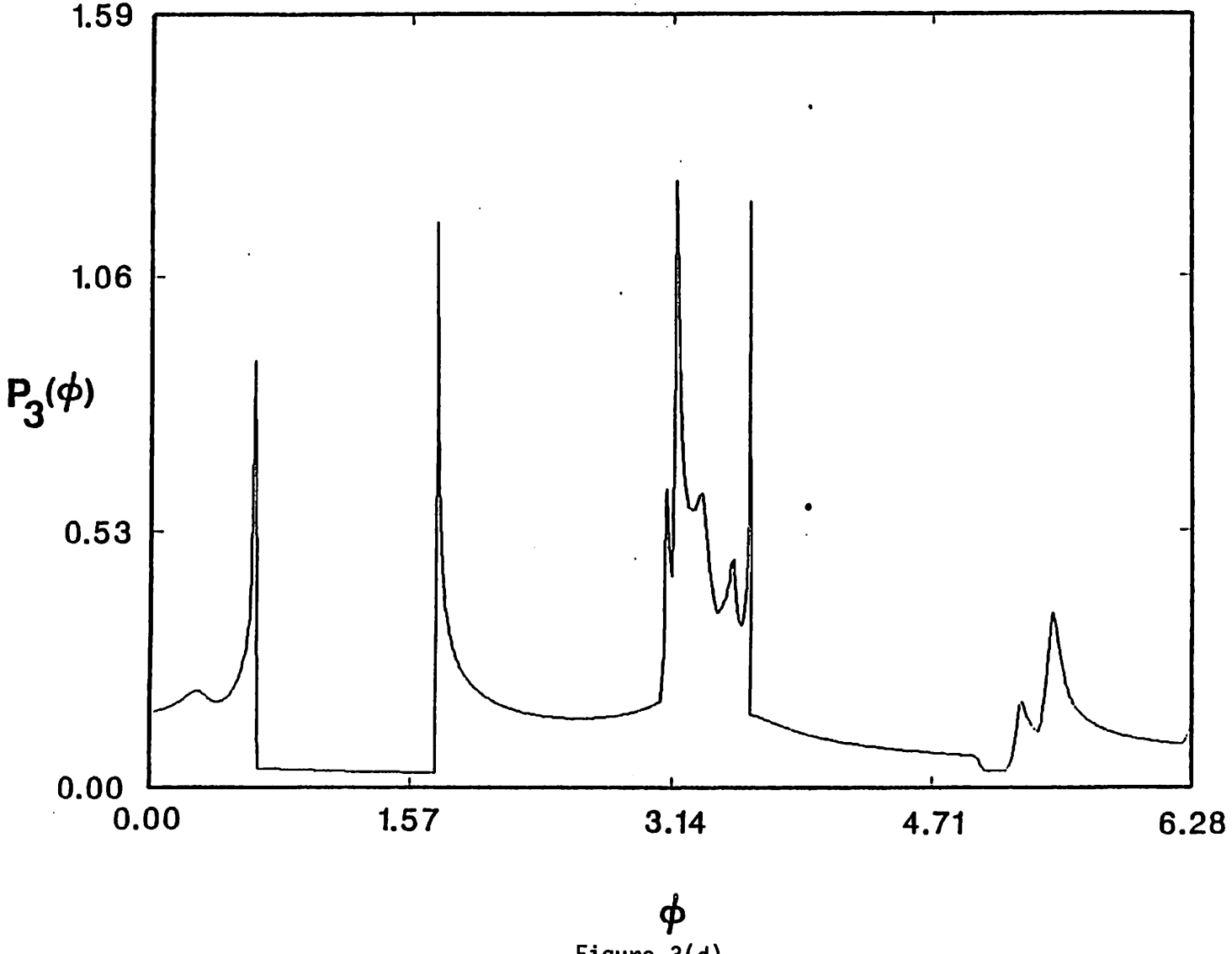

Figure 3(d)

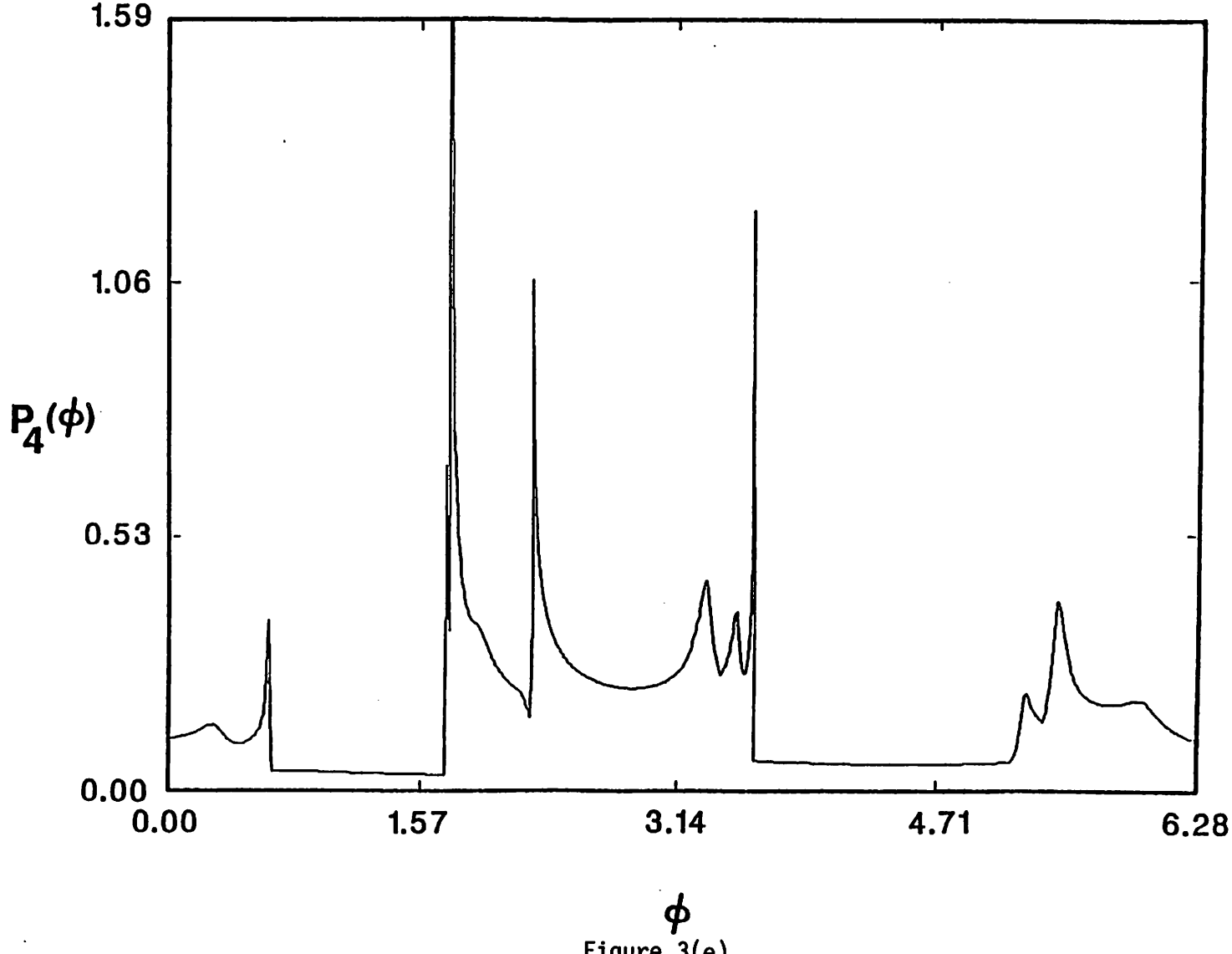

Figure 3(e)

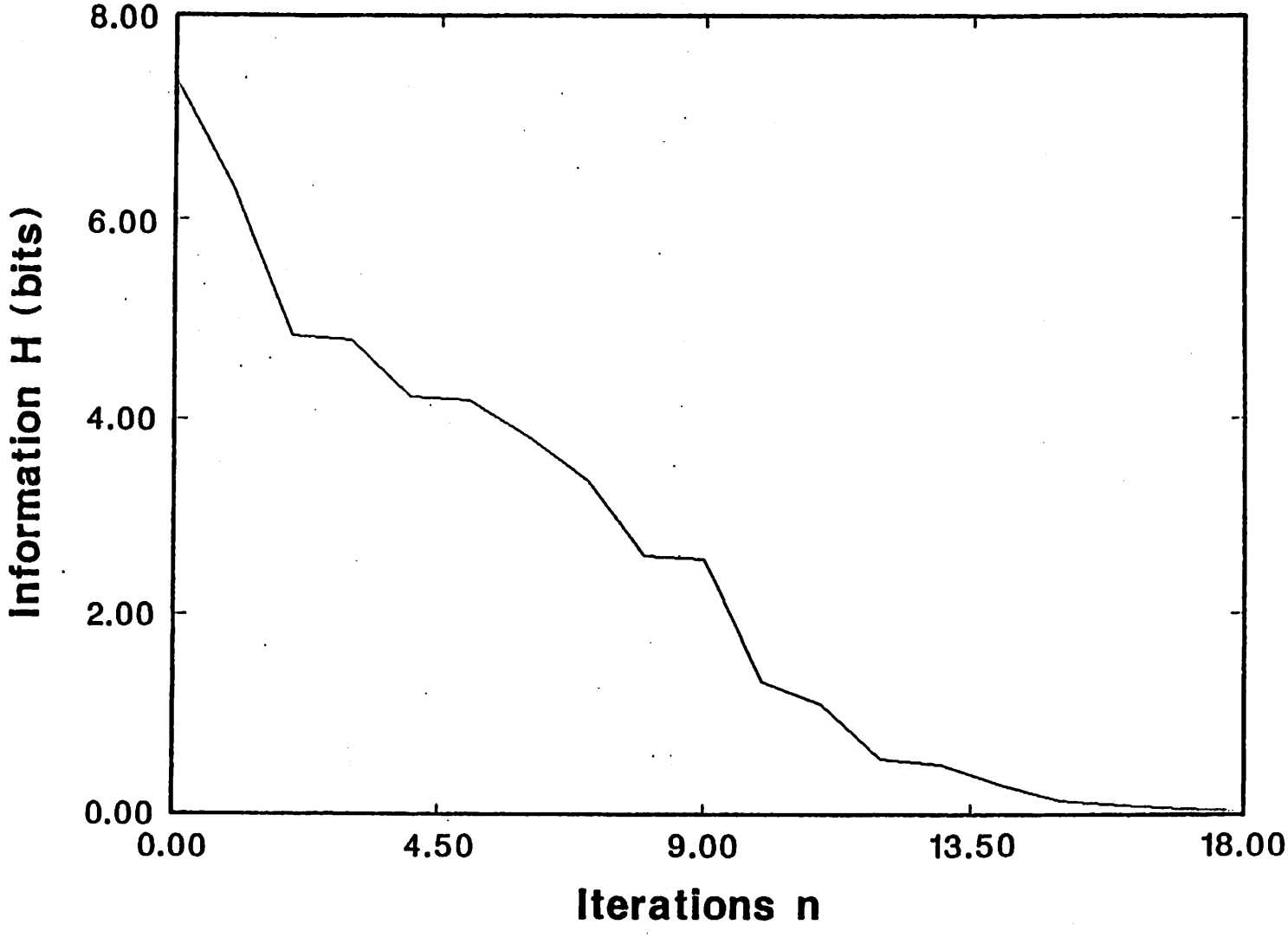

Figure 4. Loss of information about the initial state of the circuit as time progresses (in the absence of additional observations).

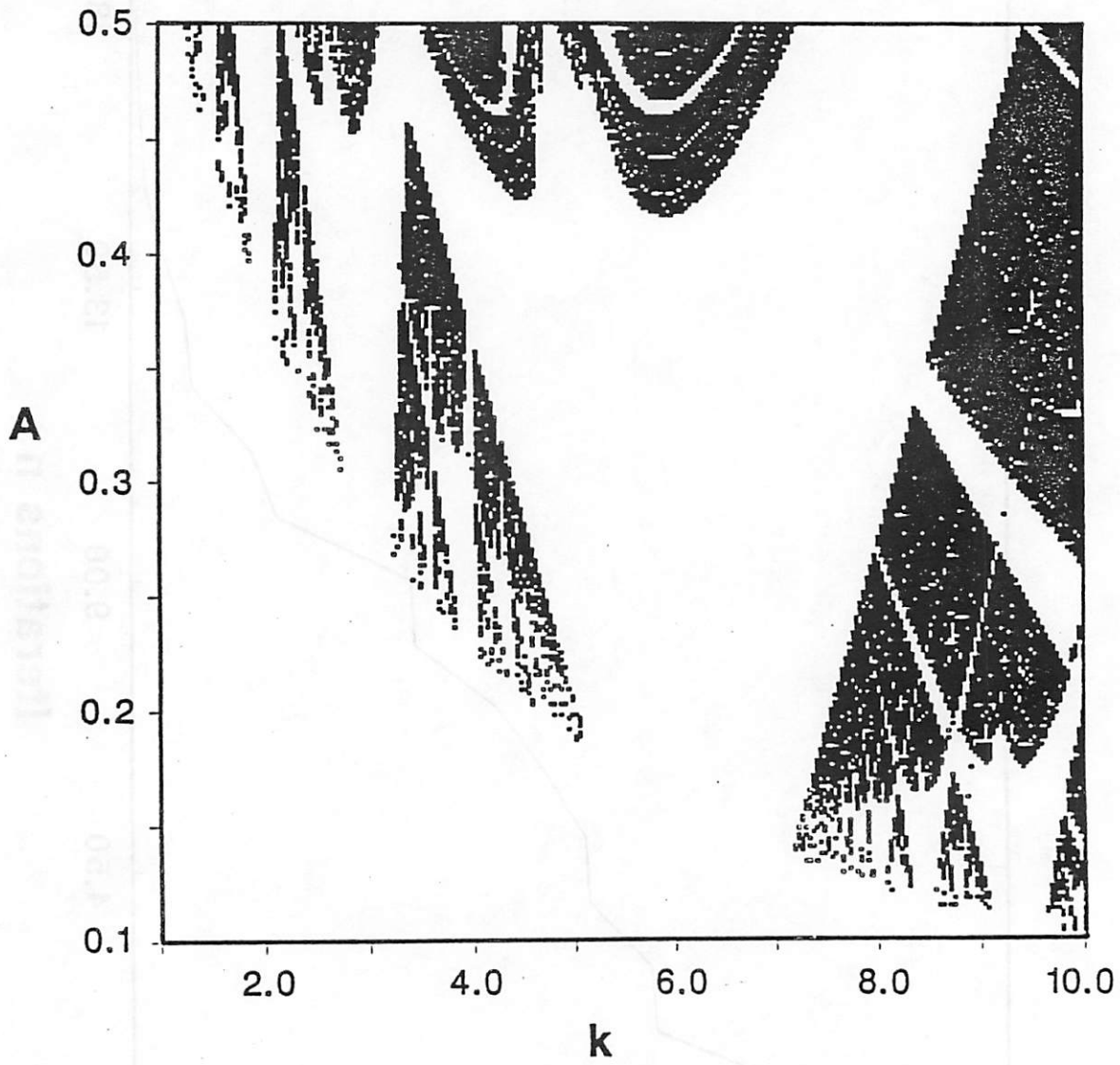

Figure 5. (a) Plot of significantly positive Lyapunov exponents for the DPLL. Black squares indicate regions of parameter space where $\lambda \geq 0.001$; 20,000 iterations were used to calculate λ. (b) Expanded plot of significantly positive Lyapunov exponents for the DPLL.

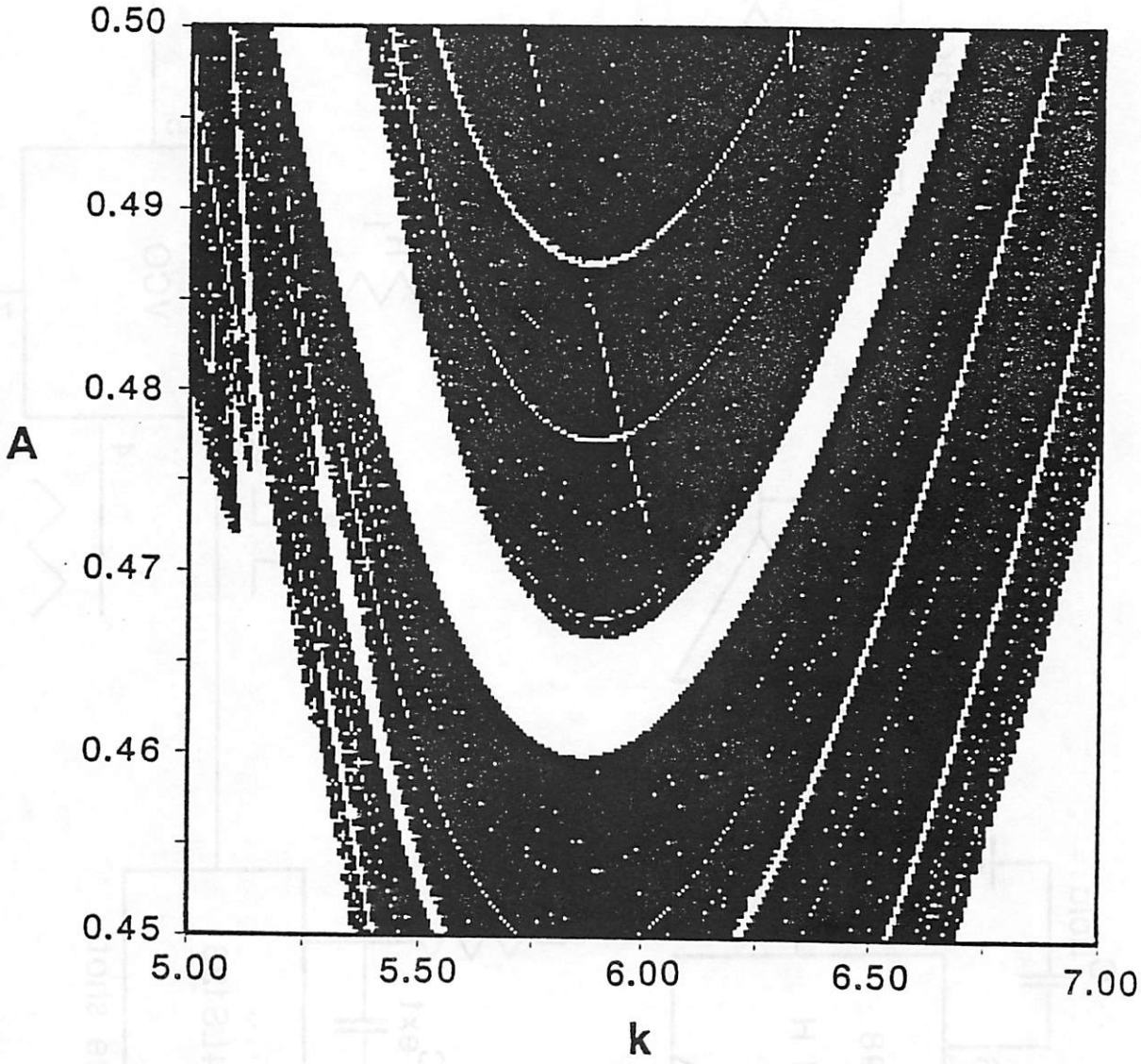

Figure 5(b)

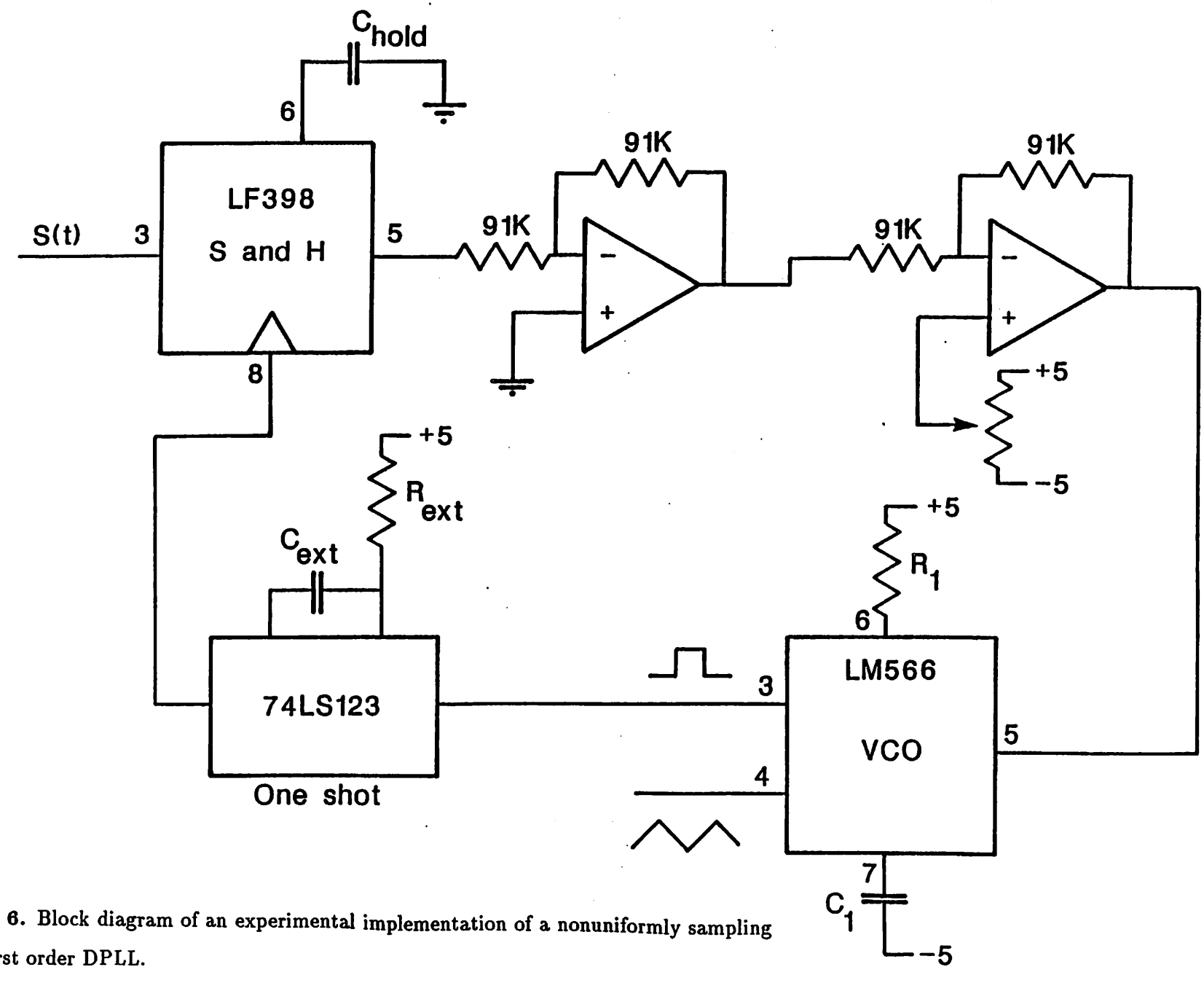

Figure 6. Block diagram of an experimental implementation of a nonuniformly sampling first order DPLL.

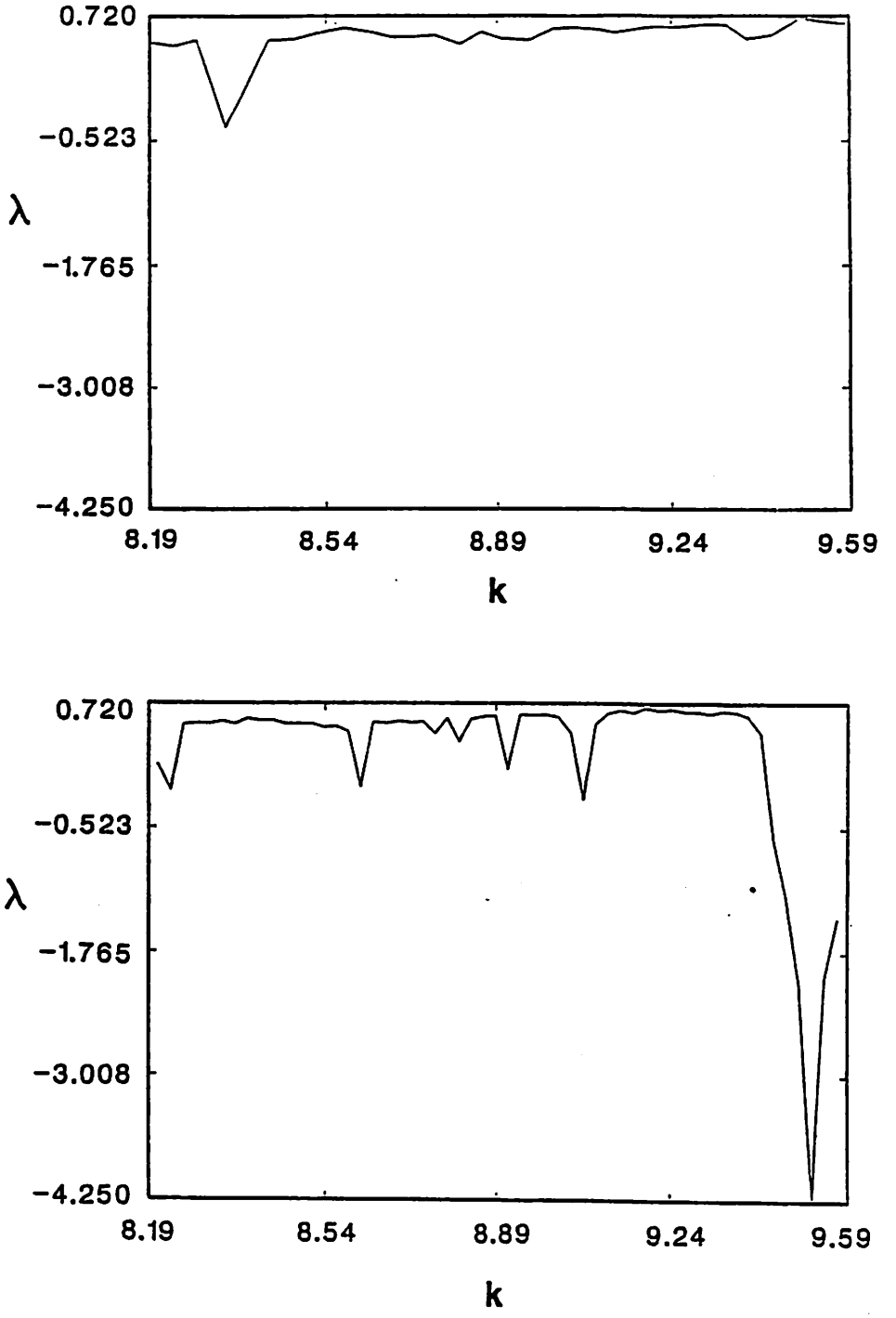

Figure 7. (a) Lyapunov exponents estimated from experimental data obtained from the circuit of Fig. 6. In this plot 30 data files containing 1000 points each were used to obtain the above estimates with $A = -0.25$. (b) Lyapunov exponents obtained from simulations. In this plot $A = -0.25$ and 40,000 iterations were used to obtain λ.