111-43
30535
66P

NASA Contractor Report 194985

Investigation of Allan Variance for Determining Noise Spectral Forms With Application to Microwave Radiometry

William D. Stanley
Old Dominion University Research Foundation, Norfolk, Virginia

(NASA-CR-194985) INVESTIGATION OF ALLAN VARIANCE FOR DETERMINING NOISE SPECTRAL FORMS WITH APPLICATION TO MICROWAVE RADIOMETRY Final Report (Old Dominion Univ.) 66 p

N95-15727

Unclas

G3/43 0030535

Contract NAS1-19858
November 1994

National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

Investigation of Allan Variance for Determining Noise Spectral Forms With Application to Microwave Radiometry

by

William D. Stanley

Abstract

An investigation of the Allan Variance method as a possible means for characterizing fluctuations in radiometric noise diodes has been performed. The goal is to separate fluctuation components into white noise, flicker noise, and random-walk noise. The primary means is by discrete-time processing, and the study focused primarily on the digital processes involved. Noise satisfying the requirements was generated by direct convolution, Fast Fourier transformation (FFT) processing in the time domain, and FFT processing in the frequency domain. Some of the numerous results obtained are presented along with the programs used in the study.

Introduction

The application of the Allan Variance method to determine the stability of precision oscillator circuits has been well documented in the literature (refs. 1–7). R. W. Lawrence of NASA Langley Research Center (ref. 8) has proposed to apply the same concept to the determination of the long-term stability of noise diodes for precision radiometric applications. This concept has been investigated in this report and some of the findings have been documented here.

Sample Mean and Variance of Random Process

Assume a random process $x(t)$ defined over some time interval, and let $\hat{x}_k(\tau)$ represent the time average of this function over the interval $t_k < t < t_k + \tau$ as illustrated in figure 1(a). This operation can be performed by the integrate-and-dump process, which can be described as

$$\hat{x}_k(\tau) = \frac{1}{\tau}\int_{t_k}^{t_k+\tau} x(t)dt \tag{1}$$

The value $\hat{x}_k(\tau)$ is the time-average or mean over an interval with duration τ.

Assume now that the function is sampled at equally-spaced intervals T seconds apart as shown in figure 1(b). For a given interval having N discrete samples, a discrete-time approximation to the continuous-time formulation of (1) is the sample mean as defined by

$$\bar{x}_k = \frac{1}{N}\sum_{k}^{k+N-1} x_j \tag{2}$$

This operation can also be referred to in signal-processing terminology as a sum-and-dump algorithm.

The sample variance of this process is denoted as $v_k(N,\tau)$ and is given by

$$v_k(N,\tau) = \frac{1}{N-1}\sum_{k}^{k+N-1}(x_j - \bar{x}_j)^2 \tag{3a}$$

$$= \frac{1}{N-1}\left[\sum_{k}^{k+N-1} x_j^2 - \frac{1}{N}\left(\sum_{k}^{k+N-1} x_j\right)^2\right] \tag{3b}$$

The proceeding formulation is based on the statistics of a so-called sample population, from which information about the process can be determined from the sample mean and sample variance. The term "Allan Variance" refers to a specific manner in which the sample variance is determined as a function of the integration time τ to sort out processes exhibiting different types of power spectra.

Assume that the power spectrum $S(f)$ is of the form

$$S(f)=\frac{K_1}{f^k} \quad \text{for} \quad 0<f<\infty \tag{4}$$

where K_1 is a constant. The theoretical work of Allan and others establishes the basis for the variance $v(N,\tau)$ to have the form

$$v(N,\tau)=K_2\tau^{k-1} \tag{5}$$

where K_2 is another constant.

Some of the underlying theory, however, involves improper integrals and diverging functions so it is not always clear how the proceeding process will behave with finite summations and discrete-time processes.

From (4) and (5), several forms were of primary interest in this investigation. They are as follows:

<u>White Noise</u>:

$$S(f)=K_1 \tag{6}$$

$$v(N,\tau)=\frac{K_2}{\tau} \tag{7}$$

<u>Flicker Noise</u>:

$$S(f)=\frac{K_1}{f} \tag{8}$$

$$v(N,\tau)=K_2 \tag{9}$$

<u>Random Walk Noise</u>:

$$S(f)=\frac{K_1}{f^2} \tag{10}$$

$$v(N,\tau) = K_2\tau \tag{11}$$

By determining $v(N,\tau)$ as a function of τ, it may be possible to sort out the spectral terms according to the behavior of the variance.

It can be shown that the expected value of $v_k(N,\tau)$ is given by

$$E\left[v_k(N,\tau)\right] = \overline{x^2} - E\left[x_k x_\ell\right] = \overline{x^2} - \Re_x(\tau) \tag{12}$$

where $\overline{x^2}$ is the mean-square value of the process and $\Re_x(\tau) = E\left[x_k x_\ell\right]$ is the autocorrelation function corresponding to a time shift of τ.

In general, the mean-square value can be expressed as

$$\overline{x^2} = \sigma_x^2 + \mu_x^2 \tag{13}$$

where σ_x^2 is the variance and μ_x is the mean or dc value. Similarly, the autocorrelation function can be expressed as

$$\Re_x(\tau) = R_x(\tau) + \mu_x^2 \tag{14}$$

where $R_x(\tau)$ is the autocovariance function. Substitution of (13) and (14) in (12) results in

$$E\left[v_k(N,\tau)\right] = \sigma_x^2 + \mu_x^2 - R_x(\tau) - \mu_x^2 \tag{15a}$$

$$= \sigma_x^2 - R_x(\tau) \tag{15b}$$

Thus, the effect of any dc component in the random process cancels out and does not affect the variance.

For any realistic processes, $R_x(\tau)$ approaches zero as τ increases. This means that the preceding expectation tends to approach

$$E[v_k(N,\tau)] \approx \sigma_x^2 \tag{16}$$

as τ increases.

After studying the pattern of the algorithm for different values of N, it was decided to restrict most of the study to a two-point variance as suggested by Lawrence. While it is possible to produce slightly lower fluctuations in the variance estimates by using more points in the algorithm, there is a tradeoff with respect to resolution. Further, the complexity of implementation of the algorithm increases markedly with increasing N. Thus, $N = 2$ was judged to be a good choice, and in subsequent expressions, the variance rotation will be simplified to $v(N,\tau) = v(2,\tau) = v(\tau)$, with $N = 2$ understood. (Later in the report, N will be used as the number of points in a record length.)

White Noise Reduction by Integration

It is instructive to initially review the procedure by which the variance of a white noise process can be reduced by integration or averaging. Consider white noise having a flat power spectrum as shown in figure 2. It has been assumed here that the noise has been sampled at intervals of T seconds apart and, hence, the sampled spectrum is defined over the frequency range $-1/2T < f < 1/2T$. The power spectrum has a constant value $S(f) = N_0/2$ over the region, and the sampling process creates aliases at integer multiples of the sampling frequency. The variance of the noise can be expressed as

$$\sigma_x^2 = \frac{N_0}{2} \cdot 2B = N_0 B = \frac{N_0}{2T} \tag{17}$$

Assume that the function $x(t)$ is successively integrated or averaged over an interval τ with τ increasing. The equivalent noise bandwidth of the process is then $0.5/\tau$. Letting $v(\tau)$ represent the variance of the filtered process, it can be readily shown that

$$v(\tau) = \frac{T}{\tau}\sigma_x^2 \tag{18}$$

The conclusion is that the variance of a white-noise process is inversely proportional to the integration time for an integrate and dump or sum and dump filter. The process is illustrated in figure 3.

Two-Point Allan Variance

Consider a random process $x(t)$ for which N equally-spaced samples are taken at intervals of T seconds over a total epoch length of t_p seconds; i.e.,

$$t_p = NT \tag{19}$$

Assume that N is selected as an integer power of 2; i.e.,

$$N = 2^l \tag{20}$$

where l is an integer. Consider the first two points x_1 and x_2 and let

$$\bar{x} = \frac{x_1 + x_2}{2} \tag{21}$$

The sample variance $v_{1,1}$ of this two-point distribution is

$$v_{1,1} = \frac{1}{1}\sum_1^2 (x_i - \bar{x})^2 = \sum_1^2 (x_i - \bar{x})^2 \tag{22}$$

Substitution of (21) in (22) and simplification lead to

$$v_{1.1} = \frac{1}{2}(x_1 - x_2)^2 \tag{23}$$

The preceding process is repeated at $N/2-1$ intervals of width $2T$ to yield a sequence of two-point variance samples as follows:

$$v_{1.2} = \frac{1}{2}(x_3 - x_4)^2 \tag{24}$$

$$v_{1.3} = \frac{1}{2}(x_5 - x_6)^2 \tag{25}$$

$$\vdots \qquad \vdots$$

$$v_{1.j} = \frac{1}{2}(x_{2j-1} - x_{2.j})^2 \tag{26}$$

$$\vdots \qquad \vdots$$

$$v_{1.N/2} = \frac{1}{2}(x_{N-1} - x_N)^2 \tag{27}$$

Next, the mean value $\overline{v}_1$ of the $N/2$ variances is formed as follows:

$$\overline{v}_1 = \frac{1}{N/2}\sum_{1}^{N/2}\overline{v}_{1.j} \tag{28}$$

The quantity $\overline{v}_1$ is the average or mean variance associated with the short integration interval $\tau = T$. This constitutes the completion of the first cycle.

The second cycle of the computational process is initiated by first doubling the "integration" or averaging interval to $2T$ by successive linear averages. Let

$$x_{2.1} = \frac{x_1 + x_2}{2} \tag{29}$$

$$x_{2,2} = \frac{x_3 + x_4}{2} \tag{30}$$

$$\vdots \qquad \vdots$$

$$x_{2,j} = \frac{x_{2j-1} + x_{2,j}}{2} \tag{31}$$

$$\vdots \qquad \vdots$$

$$x_{2,N/2} = \frac{x_{N-1} + x_N}{2} \tag{32}$$

The variance estimates for the longer intervals are now computed by

$$v_{2,1} = \frac{1}{2}(x_{2,1} - x_{2,2})^2 \tag{33}$$

$$v_{2,2} = \frac{1}{2}(x_{2,3} - x_{2,4})^2 \tag{34}$$

$$\vdots \qquad \vdots$$

$$v_{2,j} = \frac{1}{2}(x_{2,2j-1} - x_{2,2j})^2 \tag{35}$$

$$v_{2,N/4} = \frac{1}{2}(x_{2,N/2-1} - x_{2,N/2})^2 \tag{36}$$

The mean variance $\bar{v}_2$ is now computed as

$$\bar{v}_2 = \frac{1}{N/4}\sum_{1}^{N/4} \bar{v}_{2,j} \tag{37}$$

Without showing the intermediate steps, there will be a total of $L = \log_2 N$ cycles. On the last cycle, the interval is $NT/2$ and there are only two samples in which the variance can be computed. Hence

$$\bar{v}_L = \frac{1}{2}(x_{L,1} - x_{L,2})^2 \tag{38}$$

Algorithm for Two-Point Variance

The analysis in this section will be limited to the two-point variance algorithm, for which most of the data obtained are based. The basic process in cyclical form is illustrated in figure 4. Basically, there are $L = \log_2 N$ cycles with three nearly identical steps except for the first and last cycles.

The first cycle consists of the following three steps:

(1) transfer of N samples of $x(t)$ to $y(t)$ preserve original array intact

(2) computation of $N/2$ separate estimates of the variance $v_{1,j}$ by the two-point algorithm and

(3) computation of the mean variance $\bar{v}$ from the $N/2$ separate variances based on an integration time T.

The first step in the second though the $L-1$ cycles consists of replacing the N points of y with $N/2$ points based on doubling the sampling time interval. Steps (2) and (3) are identical to those of the first cycle except that the number of points used in the separate variance and mean variance estimates become successively smaller in each cycle.

The first two steps in the Lth or last cycle follow the preceding pattern except that there is only one variance estimate v_L in this case. Thus, there is no mean variance compilation in this case and $\bar{v}_L = v_L$. The equivalent integration time is $NT/2$.

As a further aid in visualizing the process, refer to the chart of figure 5 based on $N = 16$ points. Obviously, this record length would be far too short to obtain any statistical reliability, but the chart provides an interesting visual representation of the process. In this case $L = \log_2 16 = 4$. The various integration times in this process are T, $2T$, $4T$, and $8T$. There are four cycles and a total of 12 steps.

A computer flow chart for the Allan Variance algorithm is shown in figure 6. Some of the notation varies slightly from that previously utilized due to the programming constraints. In fact, the two-dimensional variables $y_{i,j}$ and $v_{i,j}$ can be replaced by a one dimensional variable by replacing the array with new data following each cycle.

Statistical Analysis of the Allan Variance

The extent to which the Allan variance algorithm is statistically significant will now be investigated. Let $v_{i,j}$ represent the ith cycle of the algorithm based on the jth segment, which will be denoted for this development simply as $v(N,\tau)$. The expression for $v(N,\tau)$ is

$$v(N,\tau)=\frac{1}{N-1}\sum_{1}^{N}(y_k-\bar{y})^2 \tag{39}$$

Assume that $y(t)$ is gaussian distributed white noise with mean μ_y and variance σ_y^2. The Allan variance can then be expressed as

$$v(N,\tau)=\frac{\sigma_y^2}{N-1}\chi_{N-1}^2 \tag{40}$$

where χ_{N-1}^2 is a chi-squared distribution with $N-1$ degrees of freedom as defined by

$$\chi_{N-1}^2=x_1^2+x_2^2+\ \ldots\ +x_N^2 \tag{41}$$

Each x_n is gaussian distributed with mean 0 and variance 1.

The mean value and variance of the chi-square distribution are denoted, respectively, as $\mu_{\chi_{N-1}^2}$ and $\sigma_{\chi_{N-1}^2}^2$, and they can be shown to be

$$\mu_{\chi_{N-1}^2}=N-1 \tag{42}$$

$$\sigma_{\chi_{N-1}^2}^2=2(N-1) \tag{43}$$

The expected value of $v(N,\tau)$ is then given by

$$E[v(N,\tau)] = \mu_v = \frac{\sigma_y^2}{N-1} \times (N-1) = \sigma_y^2 \tag{44}$$

The variance of $v(N,\tau)$ is

$$E\{[v(N,\tau) - \mu_v]^2\} = \frac{2(\sigma_y^2)^2(N-1)}{(N-1)^2} = \frac{2\sigma_y^4}{N-1} \tag{45}$$

The mean value $\bar{v}(\tau)$ based on N_1 variance computations is

$$\bar{v}(\tau) = \frac{1}{N_1}\sum_{1}^{N_1} v_k(N,\tau) \tag{46}$$

It can be shown that

$$E[\bar{v}(\tau)] = \sigma_y^2 = \mu_v \tag{47}$$

and

$$E[(\bar{v} - \mu_v)^2] = \sigma_{\bar{v}}^2 = \frac{2\sigma_y^4}{N_1(N-1)} \tag{48}$$

The preceding results have been analyzed to determine the various statistical parameters inherent in the Allan variance algorithm. A tabular formulation of these results is presented in table 1 for a record length of N points satisfying the requirement that $N = N_P^L$, where N_P is the number of points in each cycle chosen as an integer power of 2 and L is the number of cycles in the algorithm.

From the results of the table, it is clear that the mean value of the variance as well as the mean value of the averaged variance both decrease inversely with the order of the cycle. This represents the expected trend for white noise since the order of the cycle is proportional to the integration time.

The variance σ_v^2 of the variance computation decreases inversely with the number of points in each cycle and with the square of the cycle number. The variance $\sigma_{\bar{v}}^2$ of the averaged variance decreases inversely with the number of points in each cycle and with the order of the cycle.

A statistical merit factor that was found to be very useful is the ratio $\sigma_{\bar{v}} / \mu_{\bar{v}}$, i.e., the ratio of the standard deviation of the averaged variance to the value of the averaged variance. This parameter provides a measure of the relative uncertainty of the averaged variance. This parameter varies inversely with the square root of the number of points in a given cycle. In all cases, this parameter is smaller for small n, but it increases as the number of points used in the estimate decreases. In fact for a two-point algorithm, the merit factor reaches a level of $\sqrt{2}$ for the last cycle. This extreme case means that the relative uncertainty on the last cycle is about $\pm 140\,\%$ of the estimated variance!

A significant conclusion of the preceding analysis is that the record length must be considerably longer than the desired integration time over which the estimate is to be made. Depending on the desired accuracy, it could be required to be 100 or more times the desired integration time. Another way of looking at this is to say that the relative accuracy of the last portion of the record length is considerably uncertain.

A similar tabulation for two-point variance is shown in table 2. To illustrate some actual numbers, table 3 was constructed for $N = 64$ and two-point variance. Finally, table 4 illustrates the results that would apply for four-point variance and for $N = 64$.

Variance Analysis with Discrete Operations

The original work performed by Allan and others primarily utilized Fourier theorems based on classical continuous functions. While this approach is more mathematically rigorous, it is intuitively void because of the divergence of some of the functions involved at $t = 0$ and/or $f = 0$. Examples of some of the pertinent Fourier pairs are shown in figure 7. Inasmuch as any modern measurements would likely be made utilizing discrete sampling processes (possibly with an FFT), it seems prudent to investigate the phenomena involved based on a finite number of sampled points.

Consider the three forms of noise spectra depicted in figure 8. The functions of (a), (b), and (c) are assumed, respectively, as power spectra of white noise, flicker noise, and random-walk noise. Each spectrum is presumed to be derived from a process sampled at a rate of $f_s = 1/T$. However, the power spectra fold about the folding frequency $f_o = f_s/2$, and a unique spectrum exists only from dc to f_o. The effect of the Allan variance will be analyzed for each case based on a discrete summation of the power under each curve.

White Noise. For white noise, the power spectrum will be expressed as

$$S(f) = \frac{\sigma^2}{f_o} \qquad \text{for} \qquad 0 < f < f_o \tag{49}$$

where σ^2 is the variance of the process. Let $v(\tau)$ represent the Allan Variance based on an integration time τ_1 The corresponding frequency resolution Δf_1 is

$$\Delta f_1 = \frac{1}{\tau_1} \tag{50}$$

The variance of the unmodified function is determined by taking the area under the curve between 0 and f_o and it is simply

$$\begin{aligned} v(\tau_1) &= \frac{\sigma^2}{f_o}\Delta f_1 + \frac{\sigma^2}{f_o}\Delta f_1 + \cdots \\ &= \frac{N_H}{f_o}\sigma^2 \Delta f_1 = \frac{N_H}{f_o}\sigma^2\left(\frac{1}{\tau_1}\right) \end{aligned} \tag{51}$$

where $N_H = N/2$. In the initial case, $N_H\ \Delta f_1 = f_o$ and the result is σ^2 as expected.

Next, assume that the integration time is doubled, i.e., let $\tau_2 = 2\tau_1$, and let $\Delta f_2 = 1/\tau_2 = 1/2\tau_1$. The corresponding frequency increment is now half as large as before, and the variance can now be written as

$$v(\tau_2) = \frac{\sigma^2}{f_o}\Delta f_2 + \frac{\sigma^2}{f_o}\Delta f_2 + \cdots$$

$$= \frac{N_H \sigma^2}{f_o}\Delta f_2 = \left(\frac{N_H \sigma^2}{f_o}\right)\frac{1}{2\tau_1} \tag{52}$$

It should be noted that f_o is based on the original sampling rate and is a constant. Thus, the variance has been reduced by one-half and is $\sigma^2/2$.

It is easy to generalize this process at this point and write for any τ

$$v(\tau) = \frac{K}{\tau} \tag{53}$$

Thus, it is expected that the Allan variance is inversely proportional to the integration time for white noise.

Flicker Noise. Now assume that the power spectrum is of the form

$$S(f) = \frac{K_1}{f} \qquad 0 < f < f_o \tag{54}$$

This form is depicted in figure (b). Because of the divergence of (45) at $f = 0$, the lowest frequency indicated is Δf in which $S(\Delta f) = K/\Delta f$.

The variance in this case can be written as

$$v(\tau) = S(\Delta f)\Delta f + S(2\Delta f)\Delta f + \mathrm{L} \quad + S(N\Delta f)\Delta f$$

$$= K_1\left(\frac{1}{\Delta f} + \frac{1}{2\Delta f} + \frac{1}{3\Delta f} + \;....\; \frac{1}{N_H\Delta f}\right)\Delta f$$

$$= K_1\left(1 + \frac{1}{2} + \frac{1}{3} + \;....\; + \frac{1}{N_H}\right)$$

$$= K_1K_2 \tag{55}$$

where K_2 is a constant representing the summation. The expected result is a constant value independent of the integration time!

Random Walk Noise. Finally, assume that the spectrum is of the form

$$S(f) = \frac{K_3}{f^2} \qquad 0 < f < f_o \tag{56}$$

This form is illustrated in figure (c). Again, because of the divergence at $f = 0$, the lowest frequency component is Δf in which $S(f) = K/(\Delta f)^2$.

The variance in this case can be written as

$$v(\tau) = S(\Delta f)\Delta f + S(2\Delta f)\Delta f + S(3\Delta f)\Delta f + \;....\; + S(N\Delta f)\Delta f$$

$$= K_3\left[\frac{1}{\left(\Delta f^2\right)} + \frac{1}{(2\Delta f)^2} + \frac{1}{(3\Delta f)^2} + \;....\; + \frac{1}{(N_H\Delta f)^2}\right]\Delta f$$

$$= K_3\left[1 + \frac{1}{4} + \frac{1}{9} + \;....\; + \frac{1}{N^2}\right] \cdot \frac{1}{\Delta f}$$

$$= \frac{K_3K_4}{\Delta f} = K_3K_4\tau \tag{57}$$

This result indicates that the variance of random-walk noise should increase linearly with the integration time τ.

Resolution. It should be understood that as the integration time increases, the number of points used in the variance estimate decreases, and the resulting uncertainty of the estimate increases. A different way of looking at this is that the lowest frequency in the spectrum is $\Delta f = 1/\tau$ and this is the region where the uncertainty is greatest. Consequently, the integration time must be much larger than the reciprocal of the lowest frequency at which an estimate is desired.

Computer Programs Developed

Numerous computer programs were written during the course of this study, but four particular programs deserve special attention due to their extensive utilization within the study. These four programs were written in QBASIC ("Quick Basic") and are identified by the following four titles: (1) ALLAN.BAS, (2) ALLCON.BAS, (3) ALLFFT.BAS, and (4) ALLFF1.BAS. The first one explicitly uses the name Allan in reference to Allan Variance, and the others use the abbreviation "ALL" for Allan at the beginning of the title. All of the programs utilize the Allan variance algorithm, but the last three provide much more. Each will be discussed briefly, but detailed program listings will be delayed until the last three sections.

(1) ALLAN.BAS. This program provides a direct computation of the Allan Variance of a signal having 2^l points, where l is an integer. In theory l could be set to any arbitrary integer, but a practical limit is imposed by the storage capacity of the computer used. The output is a tabulation of the Allan variance as a function of the integration time τ. Although any input data could be utilized, most of the studies utilizing this program directly employed white noise.

(2) ALLCON.BAS. This program performs a direct convolution in the time domain between white noise and a specified impulse response whose Fourier transform has the

proper spectral shape required to produce different types of noise. The key feature here is that all of the analysis is done directly in the time domain. The last part of the program tabulates the Allan variance as a function of the integration time.

(3) ALLFFT.BAS. This program starts with an input random process and an impulse response as in the previous program but does not convolve in the time domain. Instead, the FFT of the input signal and the FFT of the impulse response are first computed. Next the FFT's are multiplied at each frequency using complex arithmetic, and the inverse FFT is computed. This signal is then applied to the Allan variance process, and the Allan variance as a function of the integration time is then computed.

(4) ALLFFT1.BAS. This program starts with an input random signal and computes the FFT. Unlike the previous two programs, an impulse response is not provided. Instead, the FFT of the input signal is adjusted in the frequency domain to the desired spectral shape. The inverse FFT is then computed, and the Allan variance algorithm is applied. The output is a tabulation of the Allan variance as a function of the integration time.

White Noise Simulation

White noise can be directly simulated with either of the four programs, although it is on "overkill" with the latter three. The basis for the random signal in each case was the random number generator RND in QBASIC. This generator produces a random variable that is uniformly distributed from 0 to 1 with a mean value of 0.5 and a standard deviation of $0.5/\sqrt{3}$ (or a variance of 1/12). After a reasonable number of averaging operations, a gaussian distribution may be assumed. In all simulations, the variable was shifted to produce a mean of zero and multiplied by 100. The result is a distribution having a mean value of zero and a variance of 2500/3 = 833.3.

The experimental system employed in the laboratory utilized a sample time $T = 0.2$ s. Consequently, the output data format for integration time was adjusted to create values of time that were multiples of the sampling time.

A listing of the program ALLAN and the corresponding simulation results for white noise with $N = 2^{13} = 8192$ points are shown in figure 9. An associated plot is shown in figure 10. For the initial integration time of $0.2s$, the computed Allan variance is about 819.8, which differs from the theoretical value of 833.3 by about 1.62%. From the plot, the variance as a function of integration time follows closely the expected $1/\tau$ dependence up to about $\tau = 100$ s, at which time the variation begins to deviate significantly.

Although "clumsy" because of the additional features, it was decided as a test to compare all four programs with white noise as the desired input variable. In this test, the number of points was limited to $2^{10} = 1024$ points due to memory restrictions. The program ALLAN was run again, and a program listing along with the output data are shown in figure 11. The corresponding plot is shown in figure 12. Once again, the derivation from the expected behavior becomes apparent as τ approaches the limit of the time range.

Next, the program ALLCON was adjusted for white noise. This was achieved by setting the impulse response simply to $H(0) = 1$ and $H(N) = 0$ for $N \neq 0$. The program listing and output data are shown in figure 13. These results are in perfect agreement with those of the preceding program and, therefore, were not plotted.

The third run of the same input signal was made with ALLFFT using the same noise generator as in the preceding run. Once again, the impulse response was established as a single point with non-zero value, i.e. $H(0) = 1$. The program listing is shown in figure 14 and the output data are shown in figure 15. Once again, the results are the same.

The fourth run of the same data was made with ALLFF1. In this case, only the input random signal was specified in the time domain. The approach in this case was to specify the spectral shaping. This is achieved with the use of the functions $HM(M)$ and $HP(M)$. The quantity $HM(M)$ represents the frequency domain filter magnitude response and $HP(M)$ represents the phase response. These functions were established for this run as follows:

$$HM(M) = 1 \quad \text{for} \quad 0 \le M \le 2043$$

$$HP(0) = 0 \quad \text{for} \quad 0 \le M \le 2043 \tag{58}$$

The program listing is shown in figure 16, and the data are shown in figure 17. The results are again in agreement.

Flicker Noise Simulation

The generation of flicker noise and the measurement of the Allan Variance was achieved separately with the three programs ALLCON, ALLFFT, and ALLFFT1. Flicker noise possesses a power spectrum proportional to $1/f$. The process used to create the desired noise is based on the following Fourier transform pair:

$$\Im\left[\frac{1}{\sqrt{t}}\right] = \frac{1}{\sqrt{f}} \tag{59}$$

Since convolution in the time domain is equivalent to multiplication in the frequency domain, one has the choice of either convolving the white noise at the input with an impulse response proportional to $1/\sqrt{t}$ $\left(\text{or } 1/\sqrt{N}\right)$ or by multiplying the white noise spectrum by a transfer function proportional to $1/\sqrt{f}$ $\left(\text{or } 1/\sqrt{M}\right)$.

When the linear spectrum is modified by a function proportinal to $1/\sqrt{f}$, the power spectrum becomes proportional to $1/f$, which is the desired outcome.

First, ALLCON was used for an 8192-point signal. The impulse response was defined as

$$H(0) = 0$$

$$H(N) = 1/\sqrt{N} \quad for \quad 1 \le N \le 8192 \tag{60}$$

Note that it is necessary to define $H(0)$ separately due to the divergence of the spectral form at $N = 0$. What this means is that the impulse response is invalid at the first point, but this limitation did not seem to create any problems with the results.

The program listing and a table of the Allan variance versus integration time are shown in figure 18. A corresponding plot is shown in figure 19. The ideal expected values of the Allan variance should be essentially a constant, and this trend is definitely observable up to about $\tau = 25$ s. For the last decade or so, however, the process is erratic as expected.

Next, a run was taken with ALLCON based on a 1024-point array corresponding to $\tau = 102.4$ s. The program listing and data shown in figure 20. The corresponding plot is shown in figure 21. The nearly constant behavior of the variance at small values of τ is evident, but the results become unstable near the end of the time interval.

The preceding signal was then applied to the program ALLFFT. To avoid overlap, the signal was first padded with zeros. Specifically the following input was created:

$$X(N) = \text{ random noise for } 0 \le N \le 1023$$

$$= 0 \quad \text{for } 1024 \le N \le 2047 \tag{61}$$

$$H(0) = 0 \tag{62}$$

$$H(N) = 1 / \sqrt{N} \quad \text{for} \quad 1 \le N \le 1023$$

$$= 0 \quad \text{for} \quad 1024 \le N \le 2047 \qquad (63)$$

A program listing is shown in figure 22, and the output data are shown in figure 23. These results are in exact agreement with the direct convolution approach in the time domain. Note, however, that due to the padding, a 2048-point FFT was used to generate a 1024-point result for variance analysis.

Finally, ALLFFT1 was used for a 1024-point signal. The input random signal was padded with zeros from $N = 1024$ to 2047 as in the previous case. However, the spectral shaping in this case was performed in the frequency domain by the use of the following transfer function:

$$HM(0) = 0 \qquad (64)$$

$$HM(M) = 1 / \sqrt{M} \quad \text{for} \quad 1 \le M \le 1023 \qquad (65)$$

$$HM(1024) = 0 \qquad (66)$$

$$HM(M) = HM(2048 - M) \quad for \quad 1025 \le M \le 2047 \qquad (67)$$

and

$$HP(M) = 0 \quad \text{for} \quad 0 \le M \le 2047 \qquad (68)$$

Note that the strategy in the frequency domain is to create a function whose magnitude response is an even function about the folding frequency, which corresponds to $M = 1024$. This is necessary for the corresponding impulse response to be a real function. If the phase response had been specified as anything other than zero, it would be necessary to force it to be an odd function about $M = 1024$.

The program listing is shown in figure 24, and the output data are shown in figure 25. In this case, the level of the output signal is quite different than with the use of ALLCON and ALLFFT. The reason is that there was no attempt made to adjust the spectral level to correspond with that obtained in the preceding two runs. However, the relative level is the quantity of interest, and a plot is shown in figure 26. The relative level is observed to be nearly a constant over a wide range of τ as expected.

Random Walk Noise Simulation

Continuing in the tradition established earlier, random walk noise was generated in the time domain by convolving white noise with a constant value for the response. A program utilizing ALLCON for a duration of 409.6 s and the output data are shown in figure 27. The plot of these data is shown in figure 28. The upward trend of the variance as τ increases is evident.

The array was reduced to 1024 points, constituting a time of 102.4 s, and the program and data are shown in figure 29. The corresponding plot is shown in figure 30.

Next, the program ALLFFT was applied to the same random signal. In this case, a 2048-point FFT was used with zero padding as discussed earlier. The output data are shown in figure 31. For all practical purposes, these results are in perfect agreement with those of the preceding case.

Finally, ALLFFT1 was used for the same signal by performing spectral alteration in the frequency domain. Random walk noise possesses a power spectrum proportional to $1/f^2$. The linear spectrum can then be adjusted by multiplying by $1/f$ or, equivalently in form $1/M$. Specifically, the following transfer function was used:

$$HM(0) = 0 \tag{69}$$

$$HM(M) = 1/M \quad \text{for} \quad 1 \leq M \leq 1023 \tag{70}$$

$$HM(1024) = 0 \tag{71}$$

$$HM(M) = HM(2048 - M) \quad \text{for} \quad 1025 \leq M \leq 2047 \tag{72}$$

$$HP(M) = 0 \quad \text{for} \quad 0 \leq M \leq 2047 \tag{73}$$

A program listing is shown in figure 32, and the output data are shown in figure 33. A plot of the results is shown in figure 34. As in the case of flicker noise, the level is quite different here but the relative shape displays the correct pattern for most of the time range.

Summary and Conclusions

The Allan variance process has been investigated as a means for characterizing the power spectra of random noise processes. Extensive simulations were performed using white noise, flicker noise, and random walk noise. Discrete time operations utilizing both convolution and FFT processes were utilized, and several computer programs were written to facilitate the study.

In general, the Allan variance estimates followed the theoretical processes expected over much of the integration time range. However, two conditions were clearly evident from the results: (1) the record length must be considerably longer than the interval over which the integration time results must be evaluated. Said differently, the last 90 percent or so of the record length exhibits wide fluctuations and (2) the record length must be considerable larger than the reciprocal of the lowest frequency at which the spectrum is to be evaluated.

References

1. D. W. Allan, "Statistics of atomic frequency standards." *Proc. IEEE*, vol. 54, no. 2, pp. 221–230, Feb. 1986.
2. W. F. Egan, "An efficient algorithm to compute Allan Variance from spectral density." *IEEE Trans. on Instrumentation and Measurement,* vol. 37, no. 2, pp. 240–244, June 1988.
3. K. W. Wan et al., "Extended variances and autoregressive/moving average algorithm for the measurement and synthesis of oscillator phase noise." 43rd Annual Synposium on Frequency Control. 1989.
4. B. L. Convoy and D. Le, "Measurement of Allan Variance and phase noise at fractions of a millihertz." *Rev. Sci. Instrum.*, vol. 61, no. 6, June 1990.
5. D. B. Percival, Characterization of frequency stability: frequency-domain estimation of stability measures." *Proc. IEEE,* vol. 79, no. 6, June 1991.
6. J. Rutman and F. L. Walls, "Characterization of frequency stability in precision frequency sources." *Proc. IEEE,* vol. 79, no. 7, July 1991.
7. S. Goldman, "Technique measures accuracy of Allan Variance of systems." *Microwaves & RF.* March 1993.
8. R. W. Lawrence, M. J. Scherner, and B. M. Grady, "Measurement of Calibration Stability of Radiometer Systems." 1993 SPIE International Symposium on Optical Engineering and Photonics in Aerospace and Remote Sensing. April 12-16, 1993.

Table 1. Statistical properties as a function of general parameters

	μ_v	σ_v^2	$\mu_{\bar{v}}$	$\sigma_{\bar{v}}^2$	$\sigma_{\bar{v}}/\mu_{\bar{v}}$
$n=1$	σ^2	$\dfrac{2\sigma^4}{N_p-1}$	σ^2	$\dfrac{2\sigma^4}{N_d(N_p-1)}$	$\sqrt{\dfrac{2}{N_d(N_p-1)}}$
General	$\dfrac{\sigma^2}{n}$	$\dfrac{2\sigma^4}{n^2(N_p-1)}$	$\dfrac{\sigma^2}{n}$	$\dfrac{2\sigma^4}{nN_d(N_p-1)}$	$\sqrt{\dfrac{2n}{N_d(N_p-1)}}$
Final $n=N_d$	$\dfrac{\sigma^2}{N_d}$	$\dfrac{2\sigma^4}{N_d^2(N_p-1)}$	$\dfrac{\sigma^2}{N_d}$	$\dfrac{2\sigma^4}{N_d^2(N_p-1)}$	$\sqrt{\dfrac{2}{N_p-1}}$

n = variable assuming integer values N_p^{l-1} for $1 \le l \le L$.

N = total number of points chosen as an integer power of N_p.

N_p = number of points in each cycle chosen as an integer power of 2.

N_d = N/N_p = largest number of points used in computing $\bar{v}$.

Table 2. Statistical properties for general two-point algorithms

	μ_v	σ_v^2	$\mu_{\bar{v}}$	$\sigma_{\bar{v}}^2$	$\sigma_{\bar{v}}/\mu_{\bar{v}}$
$n=1$	$2\sigma^2$	$2\sigma^4$	σ^2	$\frac{2\sigma^4}{N_d}$	$\sqrt{\frac{2}{N_d}}$
General	$\frac{\sigma^2}{n}$	$\frac{2\sigma^4}{n^2}$	$\frac{\sigma^2}{n}$	$\frac{2\sigma^4}{nN_d}$	$\sqrt{\frac{2n}{N_d}}$
Final $n=N_d$ $=N/2$	$\frac{\sigma^2}{N_d}$	$\frac{2\sigma^4}{N_d^2}$	$\frac{\sigma^2}{N_d}$	$\frac{2\sigma^4}{N_d^2}$	$\sqrt{2}$

n = variable assuming integer values 2^{l-1} for $1 \le l \le M$.

N = total number of points chosen as an integer power of 2.

N_p = 2 = number of points in each variance computation.

N_d = largest number of points used in computing mean variance.

$= N/2$

Table 3. Statistical properties for $N = 64$ and $N_p = 2$

n	μ_v	σ_v^2	$\mu_{\bar{v}}$	$\sigma_{\bar{v}}^2$	$\sigma_{\bar{v}}/\mu_{\bar{v}}$
1	σ^2	$2\sigma^4$	σ^2	$\frac{2\sigma^4}{32}$	$\sqrt{\frac{2}{32}} = 0.25$
2	$\frac{\sigma^2}{2}$	$\frac{2\sigma^4}{4}$	$\frac{\sigma^2}{2}$	$\frac{2\sigma^4}{2 \bullet 32}$	$\sqrt{\frac{2 \bullet 2}{32}} = 0.354$
4	$\frac{\sigma^2}{4}$	$\frac{2\sigma^4}{16}$	$\frac{\sigma^2}{4}$	$\frac{2\sigma^4}{4 \bullet 32}$	$\sqrt{\frac{2 \bullet 4}{32}} = 0.5$
8	$\frac{\sigma^2}{8}$	$\frac{2\sigma^4}{64}$	$\frac{\sigma^2}{8}$	$\frac{2\sigma^4}{8 \bullet 32}$	$\sqrt{\frac{2 \bullet 8}{32}} = 0.707$
16	$\frac{\sigma^2}{16}$	$\frac{2\sigma^2}{256}$	$\frac{\sigma^2}{16}$	$\frac{2\sigma^4}{16 \bullet 32}$	$\sqrt{\frac{2 \bullet 16}{32}} = 1$
32	$\frac{\sigma^2}{32}$	$\frac{2\sigma^2}{1024}$	$\frac{\sigma^2}{32}$	$\frac{2\sigma^4}{32 \bullet 32}$	$\sqrt{\frac{2 \bullet 32}{32}} = \sqrt{2}$

n = variable assuming integer values 2^{l-1} for $1 \le l \le 6$.

$N = 64$, $N_p = 2$ $N_d = 64/2 = 32$.

Table 4. Statistical properties for $N = 64$ and $N_p = 4$

n	μ_v	σ_v^2	$\mu_{\bar{v}}$	$\sigma_{\bar{v}}^2$	$\sigma_{\bar{v}}/\mu_{\bar{v}}$
1	σ^2	$\frac{2\sigma^4}{3}$	σ^2	$\frac{2\sigma^4}{3 \bullet 16}$	$\sqrt{\frac{2}{3 \bullet 16}} = 0.204$
4	$\frac{\sigma^2}{4}$	$\frac{2\sigma^4}{3 \bullet 16}$	$\frac{\sigma^2}{4}$	$\frac{2\sigma^4}{3 \bullet 4 \bullet 16}$	$\sqrt{\frac{2 \bullet 4}{3 \bullet 16}} = 0.408$
16	$\frac{\sigma^2}{16}$	$\frac{2\sigma^4}{3 \bullet 256}$	$\frac{\sigma^2}{16}$	$\frac{2\sigma^4}{3 \bullet 256}$	$\sqrt{\frac{2}{3}} = 0.816$

n = variable assuming values 4^{l-1} for $1 \le l \le 3$.

$N = 64$.

$N_p = 4$.

$N_d = 64 / 4 = 16$.

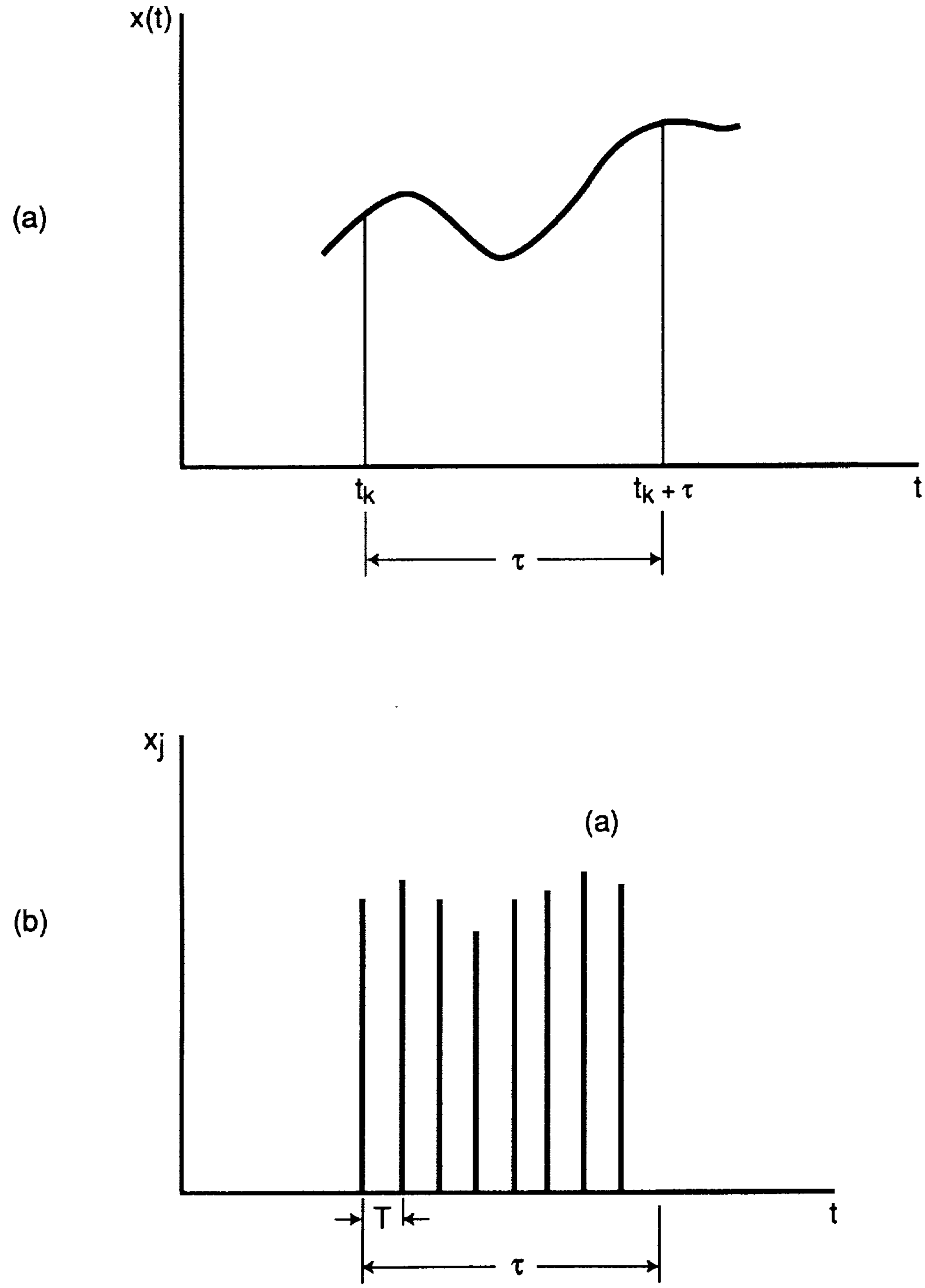


Figure 1. Continuous-time and discrete time functions for which the mean and variance are to be estimated.

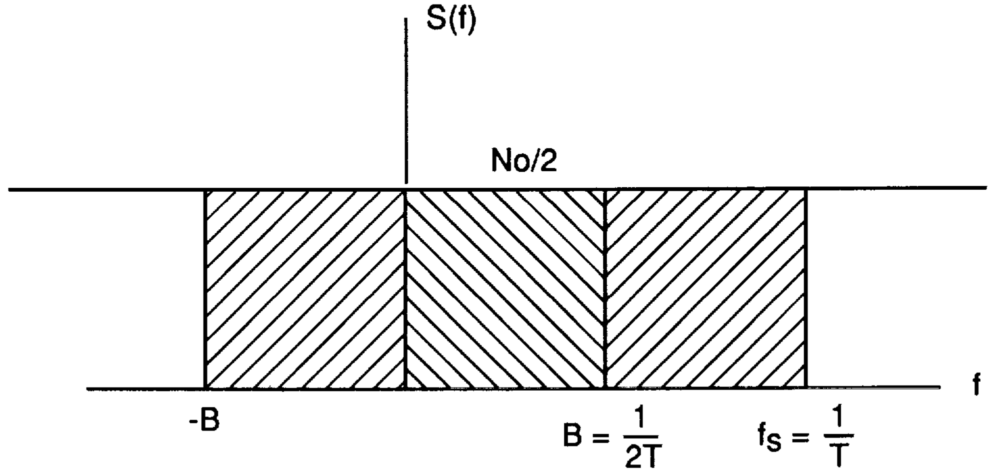


Figure 2. White noise spectral form after sampling.

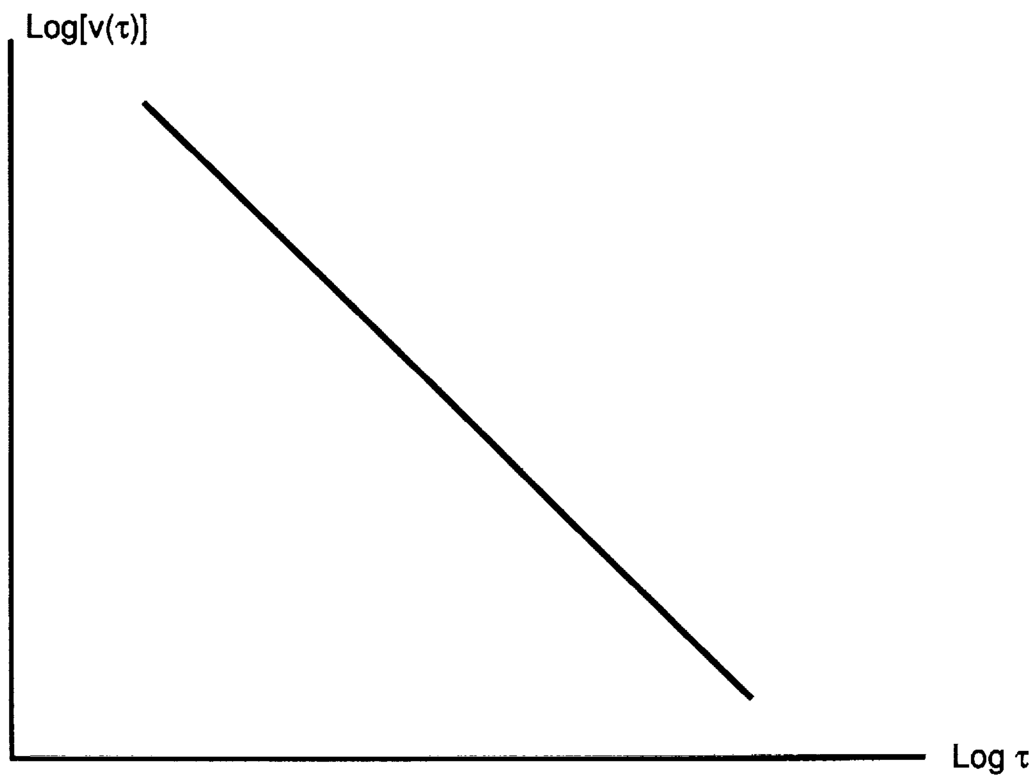


Figure 3. Variation of white noise variances as a function of integration time.

Integration Time

$\downarrow$

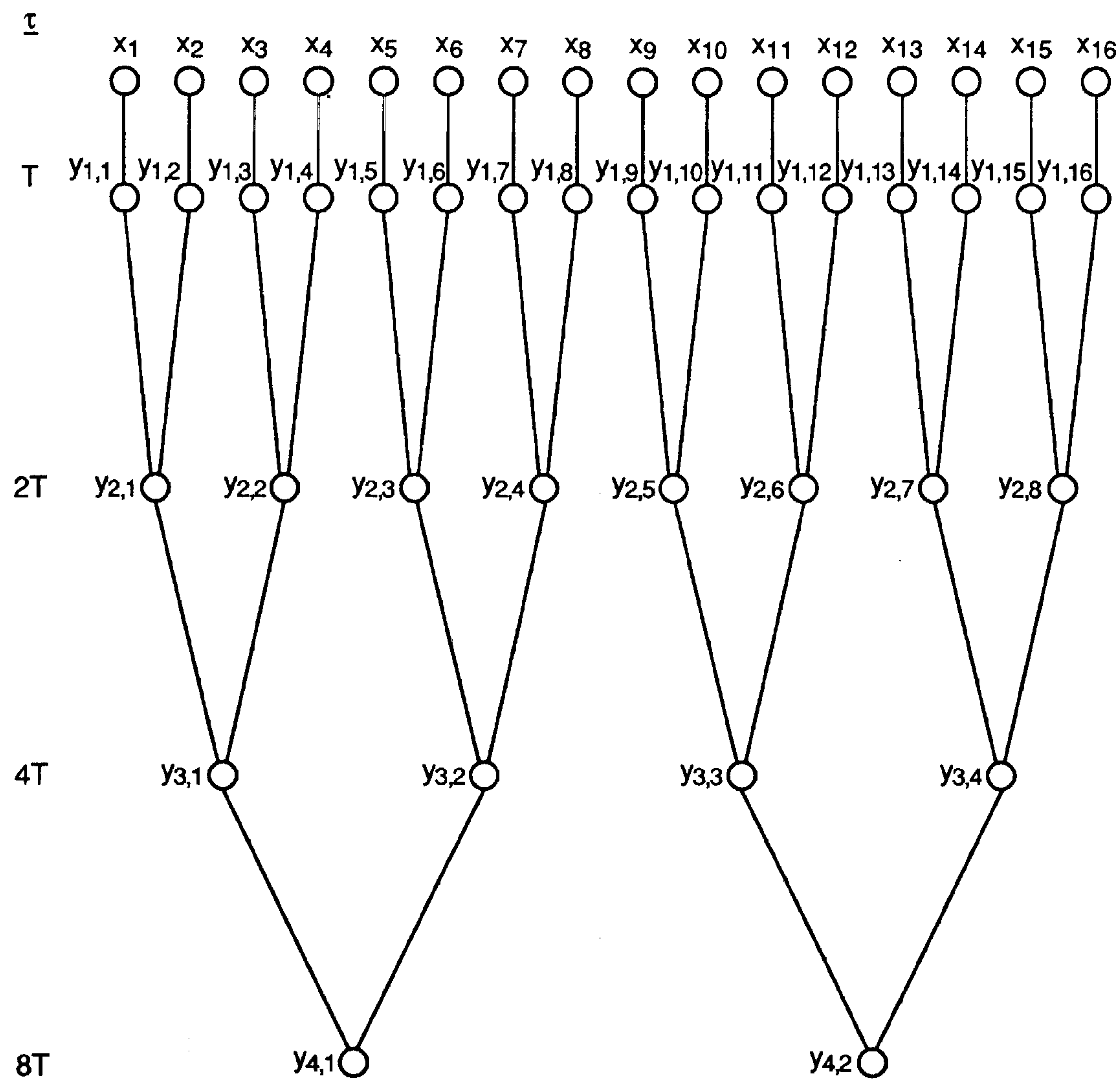
$$T \quad \left[\begin{array}{ll} y_{1,j} = x_j & \text{for } j = 1, N \\ v_{1,j} = \frac{1}{2}(y_{2j-1} - y_{2j})^2 & \text{for } j = 1, N_d \text{ (or } N/2) \\ \bar{v}_1 = \frac{1}{N_d} \sum_{1}^{N_d} v_{1,j} & \end{array} \right.$$

$$2T \quad \left[\begin{array}{ll} y_{2,j} = \frac{1}{2}(y_{2j-1} + y_{2j}) & \text{for } j = 1, N_d \\ v_{2,j} = \frac{1}{2}(y_{2j-1} - y_{2j})^2 & \text{for } j = 1, N_d/2 \text{ (or } N/4) \\ \bar{v}_2 = \frac{1}{N_d/2} \sum_{1}^{N_d/2} v_{2,j} & \end{array} \right.$$

$$4T \quad \left[\begin{array}{ll} y_{3,j} = \frac{1}{2}(y_{2j-1} + y_{2j}) & \text{for } j = 1, N_d/2 \\ v_{3,j} = \frac{1}{2}(y_{2j-1} - y_{2j})^2 & \text{for } j = 1, N_d/4 \text{ (or } N/8) \\ \bar{v}_3 = \frac{1}{N_d/4} \sum_{1}^{N_d/4} v_{3,j} & \end{array} \right.$$

$$\frac{NT}{2} \quad \left[\begin{array}{ll} y_{L,j} = \frac{1}{2}(y_{2j-1} + y_{2j}) & \text{for } j = 1, 2 \\ v_{L,j} = \frac{1}{2}(y_1 - y_2)^2 & \text{for } j = 1, 1 \text{ (or } N/N) \\ \bar{v}_L = v_L & \end{array} \right.$$

Figure 4. Algorithm for two-point variance computations.



$y_{1,j} = x_j$ for $j = 1, 16$

$v_{1,j} = \frac{1}{2}(y_{1,2j-1} - y_{1,2j})^2$ for $j = 1, 8$

$\bar{v}_1 = \frac{1}{8}\sum_{j=1}^{8} v_{1,j}$

$y_{2,j} = \frac{1}{2}(y_{1,2j-1} + y_{1,2j})$ for $j = 1, 8$

$v_{2,j} = \frac{1}{2}(y_{2,2j-1} - y_{2,2j})^2$ for $j = 1, 4$

$\bar{v}_2 = \frac{1}{4}\sum_{j=1}^{4} v_{2,j}$

$y_{3,j} = \frac{1}{2}(y_{2,2j-1} + y_{2,2j})$ for $j = 1, 4$

$v_{3,j} = \frac{1}{2}(y_{3,2j-1} - y_{3,2j})^2$ for $j = 1, 2$

$\bar{v}_3 = \frac{1}{2}\sum_{j=1}^{2} v_{3,j}$

$y_{4,j} = \frac{1}{2}(y_{3,2j-1} + y_{3,2j})$ for $j = 1, 2$

$v_{4,j} = \frac{1}{2}(y_{4,2j-1} - y_{4,2j})^2$ for $j = 1, 1$

$\bar{v}_4 = v_{4,1}$

Figure 5. Variance computation chart for N = 16.

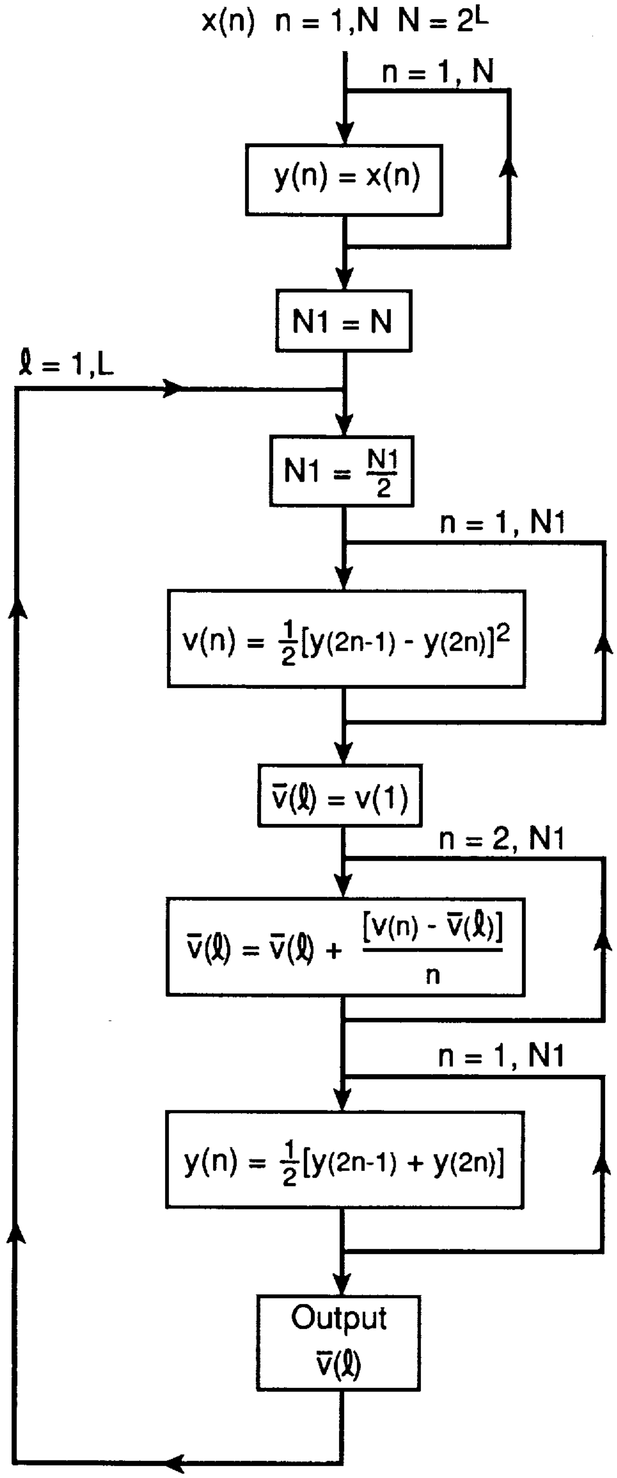


Figure 6. Flow chart for two-point variance computation.

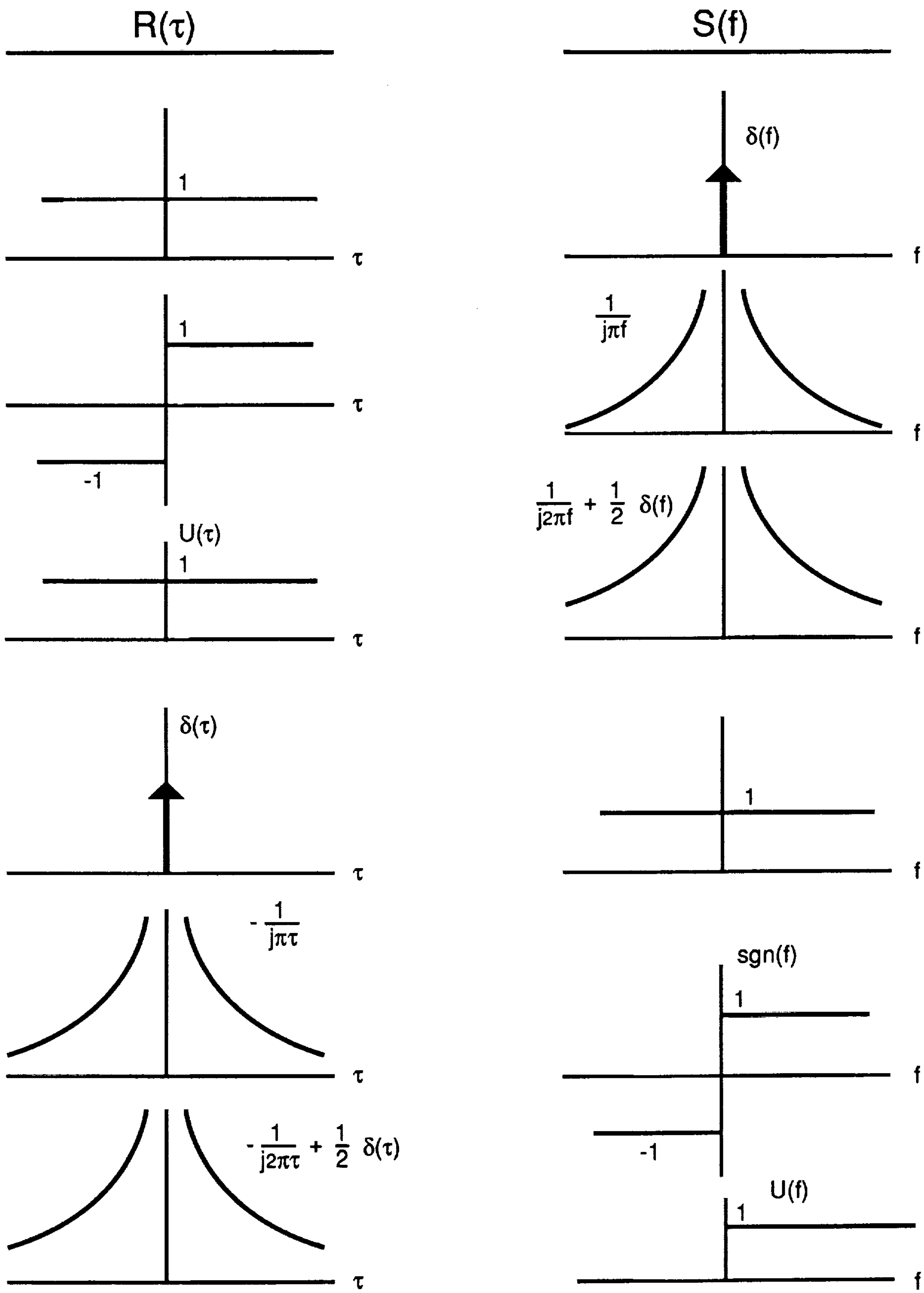


Figure 7. Some autocorrelation Fourier transform pairs.

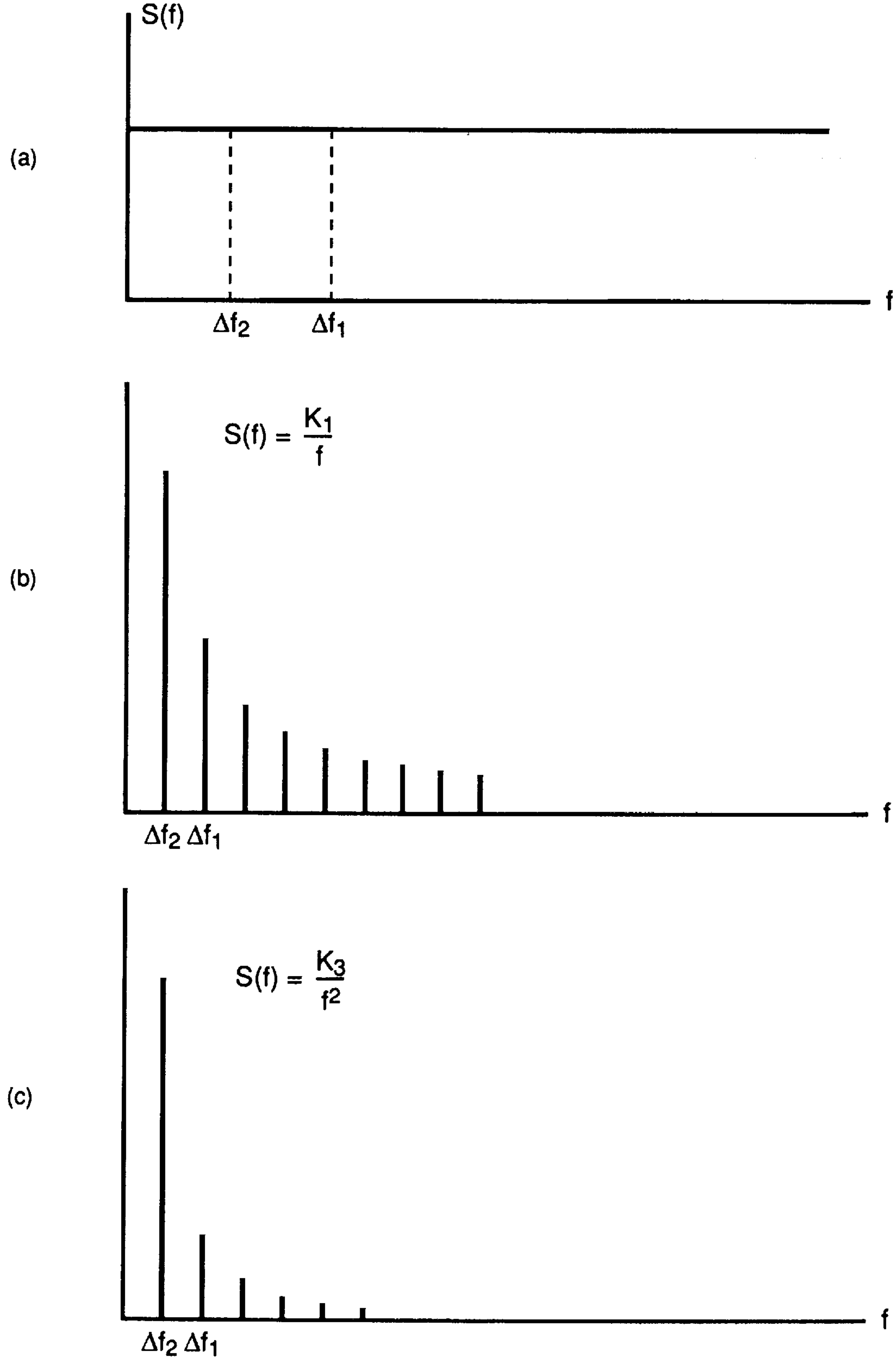


Figure 8. Three noise forms considered in study.

```
CLS
DIM X(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLAN TO DETERMINE SAMPLE VARIANCE": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 8192: LP = 13
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5): Y(N) = X(N)
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N
N1 = NP
FOR L = 1 TO LP
N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L
BEEP
```

PROGRAM ALLAN TO DETERMINE SAMPLE VARIANCE

L	INT TIME	VARIANCE
1	.2	819.8405
2	.4	422.2106
3	.8	214.7032
4	1.6	103.3248
5	3.2	52.65189
6	6.4	27.02878
7	12.8	14.13634
8	25.6	6.824592
9	51.2	2.61482
10	102.4	1.387154
11	204.8	.3692694
12	409.6	.442937
13	819.2	.603497

Figure 9. Program and data for white noise with $\tau = 819.2$ s and ALLAN.

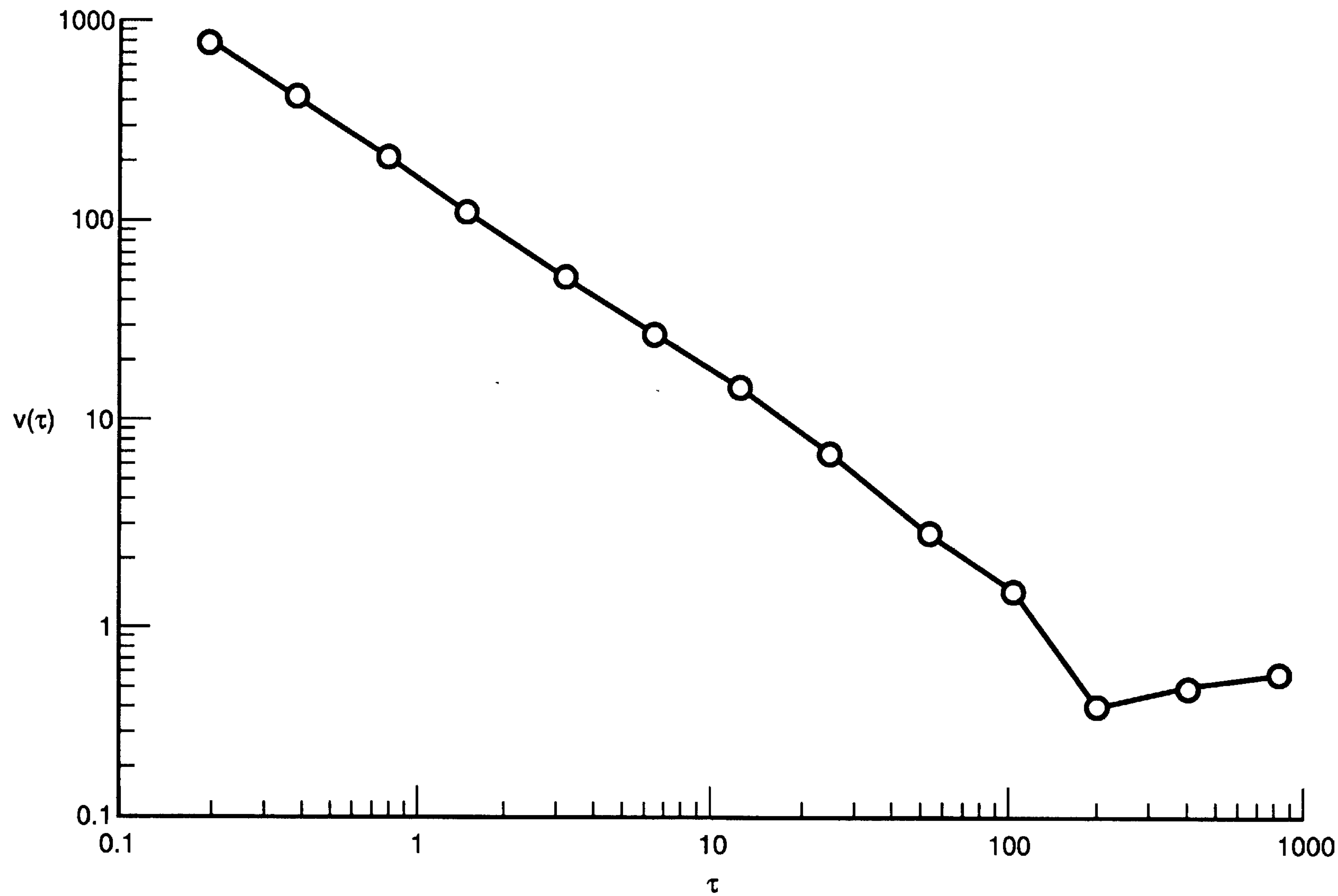


Figure 10. Plot for results of Figure 9.

```
CLS
DIM X(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLAN TO DETERMINE SAMPLE VARIANCE": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 1024: LP = 10
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5): Y(N) = X(N)
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N
N1 = NP
FOR L = 1 TO LP
N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L
BEEP
```

```
PROGRAM ALLAN TO DETERMINE SAMPLE VARIANCE

L              INT TIME       VARIANCE

 1              .2             761.5204
 2              .4             406.3706
 3              .8             212.9356
 4              1.6            98.75756
 5              3.2            44.2822
 6              6.4            13.76261
 7              12.8           18.59924
 8              25.6           8.317465
 9              51.2           3.744104
 10             102.4          .2659905
```

Figure 11. Program and data for white noise with $\tau = 102.4$ s and ALLAN.

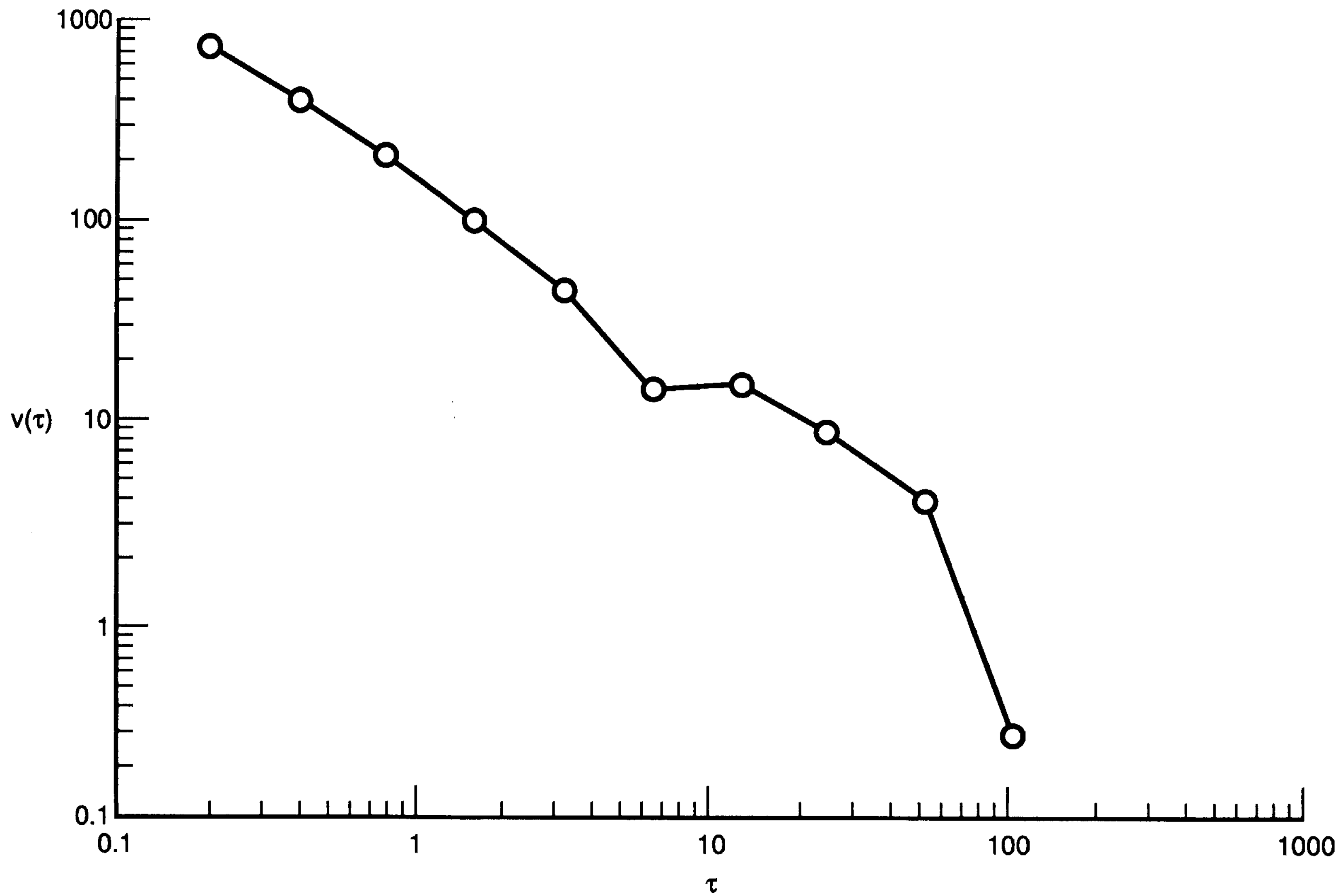


Figure 12. Plot for results of Figure 11.

```
CLS
DIM X(8192), H(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLCON USING DIRECT CONVOLUTION": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 1024: LP = 10
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5)
H(N) = 0: Y(N) = 0
NEXT N
H(0) = 1
FOR N = 0 TO NP - 1
FOR I = 0 TO N
Y(N) = Y(N) + X(I) * H(N - I)
NEXT I
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N: N1 = NP
FOR L = 1 TO LP: N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L: BEEP
```

```
PROGRAM ALLCON USING DIRECT CONVOLUTION

L               INT TIME       VARIANCE

 1               .2             761.5204
 2               .4             406.3706
 3               .8             212.9356
 4               1.6            98.75756
 5               3.2            44.2822
 6               6.4            13.76261
 7               12.8           18.59924
 8               25.6           8.317465
 9               51.2           3.744104
 10              102.4          .2659905
```

Figure 13. Program and data for white noise with $\tau = 102.4$ s and ALLCON.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048)
DIM H(2048), H1R(2048), H1I(2048), H2R(2048), H2I(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
H(0) = 1: FOR N = 1 TO NH - 1: H(N) = 0: NEXT N
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): NEXT N
FOR N = NH TO NP - 1: X(N) = 0: H(N) = 0: NEXT N
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: H1R(M) = H(M): H1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1: AR = H1R(M + IC): AI = H1I(M + IC)
BR = ZR * H1R(M + ID) - ZI * H1I(M + ID)
BI = ZI * H1R(M + ID) + ZR * H1I(M + ID)
H2R(M) = AR + BR: H2I(M) = AI + BI
H2R(M + NH) = AR - BR: H2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: H1R(M) = H2R(M): H1I(M) = H2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1
Y1R(M) = X1R(M) * H1R(M) - X1I(M) * H1I(M)
Y1I(M) = X1R(M) * H1I(M) + X1I(M) * H1R(M): NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 14. Program and data for white noise with $\tau = 102.4$ s and ALLFFT.

PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION

N	INT TIME	VARIANCE
1	.2	761.5205
2	.4	406.3707
3	.8	212.9357
4	1.6	98.75756
5	3.2	44.28219
6	6.4	13.76262
7	12.8	18.59924
8	25.6	8.317466
9	51.2	3.744104
10	102.4	.2659903

Figure 15. Data for white noise with $\tau = 102.4$ s and ALLFFT.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048), HM(2048), HP(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): NEXT N
FOR N = NH TO NP - 1: X(N) = 0: NEXT N
FOR M = 1 TO NH - 1: HM(M) = 1: HP(M) = 0: NEXT M
HM(0) = 1: HP(0) = 0: HM(NH) = 1: HP(NH) = 0
FOR M = NH + 1 TO NP - 1: HM(M) = HM(NP - M): HP(M) = 0: NEXT M
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: HR = HM(M) * COS(HP(M)): HI = HM(M) * SIN(HP(M))
Y1R(M) = X1R(M) * HR - X1I(M) * HI
Y1I(M) = X1R(M) * HI + X1I(M) * HR: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 16. Program for white noise with $\tau = 102.4$ s and ALLFFT1.

PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING

N	INT TIME	VARIANCE
1	.2	761.5205
2	.4	406.3707
3	.8	212.9357
4	1.6	98.75756
5	3.2	44.28219
6	6.4	13.76262
7	12.8	18.59924
8	25.6	8.317466
9	51.2	3.744104
10	102.4	.2659903

Figure 17. Data for white noise with τ = 102.4 s and ALLFFT1.

```
CLS
DIM X(8192), H(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLCON USING DIRECT CONVOLUTION": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 4096: LP = 12
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5): Y(N) = 0: NEXT N
FOR N = 1 TO NP - 1: H(N) = 1 / SQR(N): NEXT N: H(0) = 0
FOR N = 0 TO NP - 1
FOR I = 0 TO N
Y(N) = Y(N) + X(I) * H(N - I)
NEXT I
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N: N1 = NP
FOR L = 1 TO LP: N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L: BEEP
```

```
PROGRAM ALLCON USING DIRECT CONVOLUTION

L             INT TIME      VARIANCE

 1             .2            465.0879
 2             .4            483.4081
 3             .8            511.6693
 4             1.6           599.207
 5             3.2           747.6578
 6             6.4           800.0687
 7             12.8          1009.559
 8             25.6          1217.807
 9             51.2          179.5898
 10            102.4         87.83582
 11            204.8         956.2097
 12            409.6         3456.044
```

Figure 18. Program and data for flicker noise with τ = 409.2 s and ALLCON.

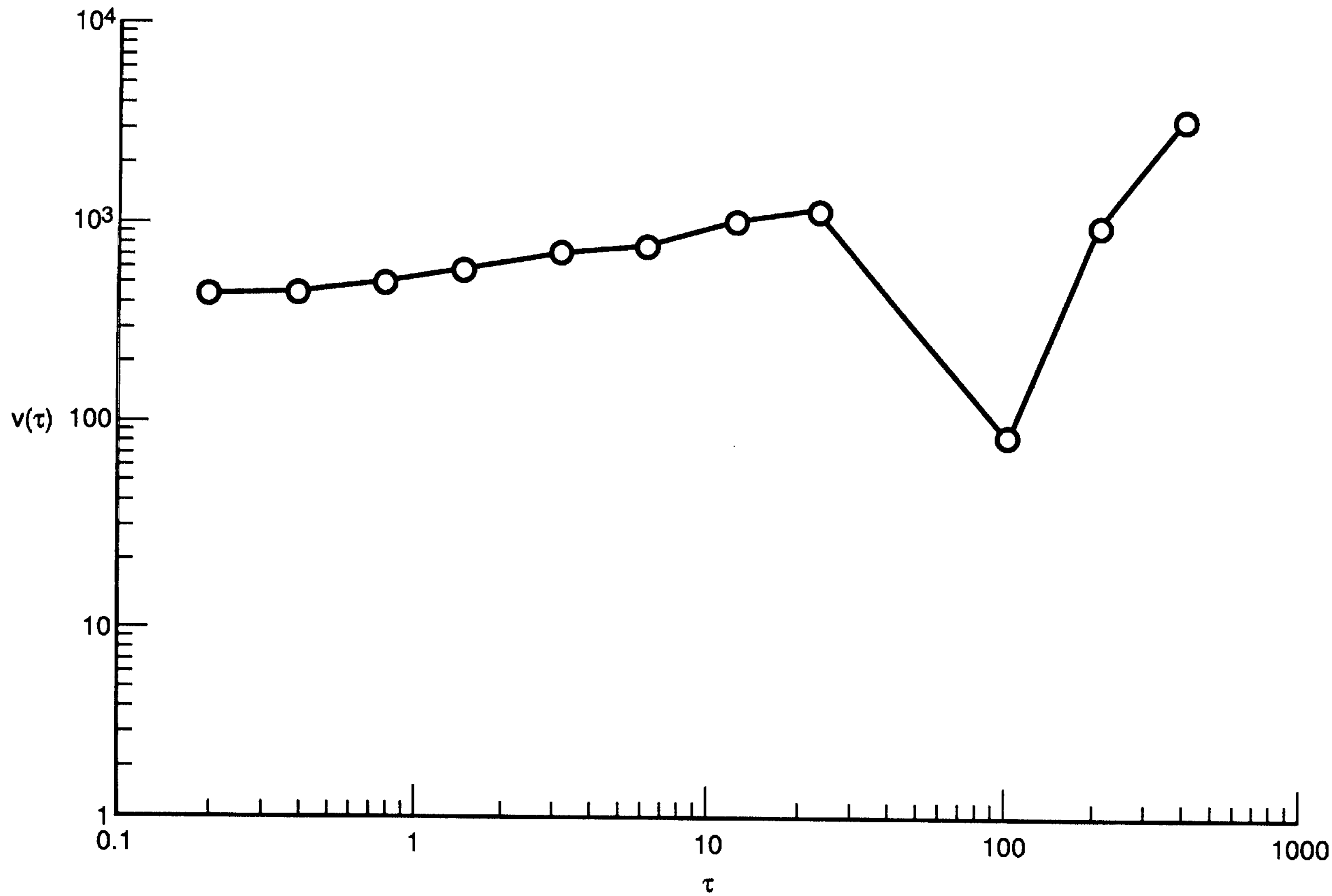


Figure 19. Plot for results of Figure 18.

```
CLS
DIM X(8192), H(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLCON USING DIRECT CONVOLUTION": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 1024: LP = 10
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5): Y(N) = 0: NEXT N
FOR N = 1 TO NP - 1: H(N) = 1 / SQR(N): NEXT N: H(0) = 0
FOR N = 0 TO NP - 1
FOR I = 0 TO N
Y(N) = Y(N) + X(I) * H(N - I)
NEXT I
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N: N1 = NP
FOR L = 1 TO LP: N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L: BEEP
```

```
PROGRAM ALLCON USING DIRECT CONVOLUTION

L             INT TIME      VARIANCE

 1             .2            437.8973
 2             .4            440.3997
 3             .8            532.6196
 4             1.6           517.8219
 5             3.2           800.6263
 6             6.4           846.5082
 7             12.8          1505.267
 8             25.6          1135.48
 9             51.2          5.159271
 10            102.4         60.71339
```

Figure 20. Program and data for flicker noise with $\tau = 102.4$ s and ALLCON.

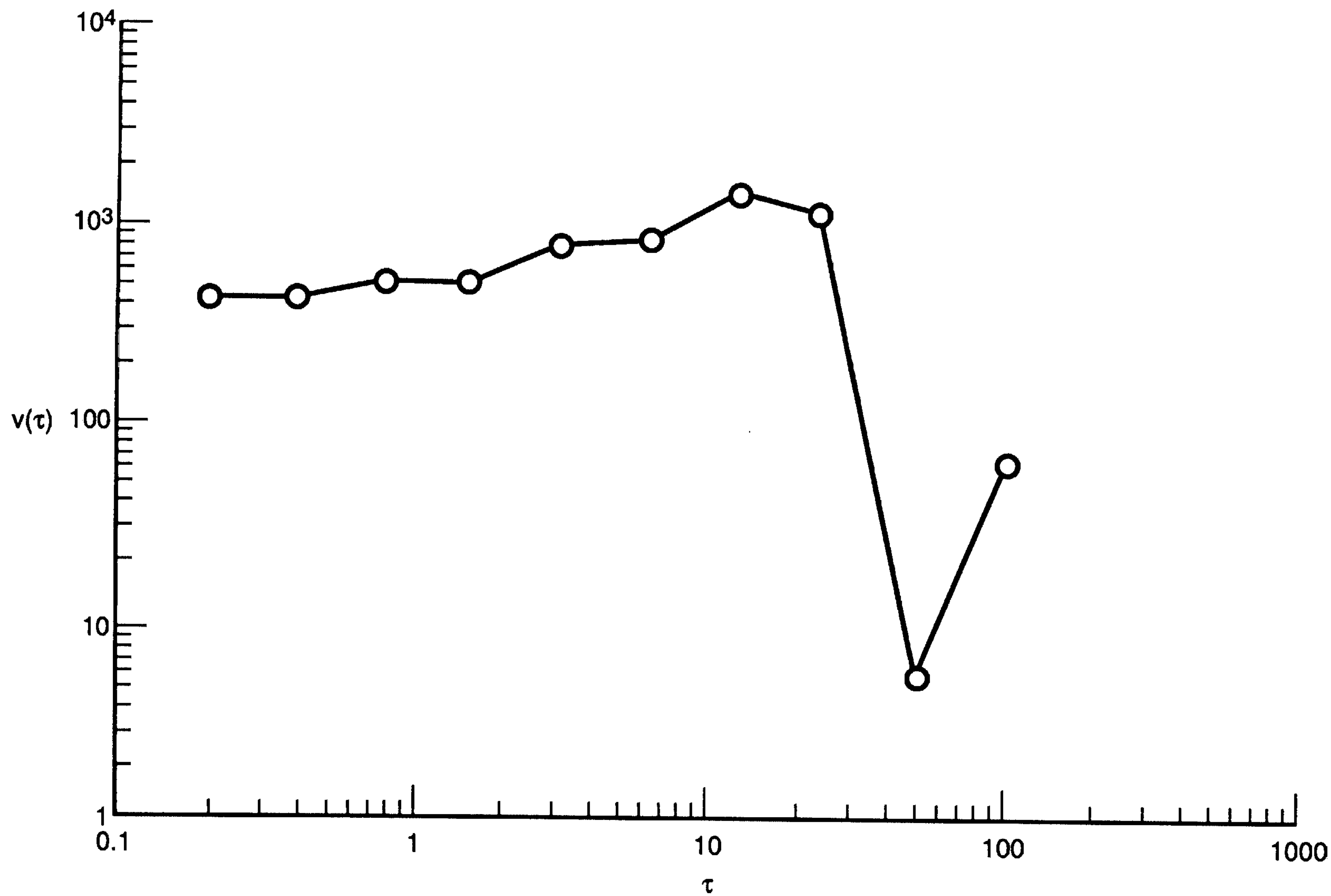


Figure 21. Plot for results of Figure 20.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048)
DIM H(2048), H1R(2048), H1I(2048), H2R(2048), H2I(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
H(0) = 0: FOR N = 1 TO NH - 1: H(N) = 1 / SQR(N): NEXT N
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): NEXT N
FOR N = NH TO NP - 1: X(N) = 0: H(N) = 0: NEXT N
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: H1R(M) = H(M): H1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1: AR = H1R(M + IC): AI = H1I(M + IC)
BR = ZR * H1R(M + ID) - ZI * H1I(M + ID)
BI = ZI * H1R(M + ID) + ZR * H1I(M + ID)
H2R(M) = AR + BR: H2I(M) = AI + BI
H2R(M + NH) = AR - BR: H2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: H1R(M) = H2R(M): H1I(M) = H2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1
Y1R(M) = X1R(M) * H1R(M) - X1I(M) * H1I(M)
Y1I(M) = X1R(M) * H1I(M) + X1I(M) * H1R(M): NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 22. Program for flicker noise with $\tau = 102.4$ s and ALLFFT.

PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION

N	INT TIME	VARIANCE
1	.2	437.8973
2	.4	440.4008
3	.8	532.6213
4	1.6	517.8229
5	3.2	800.6278
6	6.4	846.5118
7	12.8	1505.262
8	25.6	1135.482
9	51.2	5.160076
10	102.4	60.71655

Figure 23. Data for flicker noise with $\tau = 102.4$ s and ALLFFT.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048), HM(2048), HP(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): NEXT N
FOR N = NH TO NP - 1: X(N) = 0: NEXT N
FOR M = 1 TO NH - 1: HM(M) = 1 / SQR(M): HP(M) = 0: NEXT M
HM(0) = 0: HP(0) = 0: HM(NH) = 0: HP(NH) = 0
FOR M = NH + 1 TO NP - 1: HM(M) = HM(NP - M): HP(M) = 0: NEXT M
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: HR = HM(M) * COS(HP(M)): HI = HM(M) * SIN(HP(M))
Y1R(M) = X1R(M) * HR - X1I(M) * HI
Y1I(M) = X1R(M) * HI + X1I(M) * HR: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 24. Program for flicker noise with $\tau = 102.4$ s and ALLFFT1.

PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING

N	INT TIME	VARIANCE
1	.2	1.219962
2	.4	1.214046
3	.8	1.171432
4	1.6	1.17254
5	3.2	.9797807
6	6.4	.7463165
7	12.8	1.43384
8	25.6	1.68266
9	51.2	.9090664
10	102.4	.3821753

Figure 25. Data for flicker noise with $\tau = 102.4$ s and ALLFFT1.

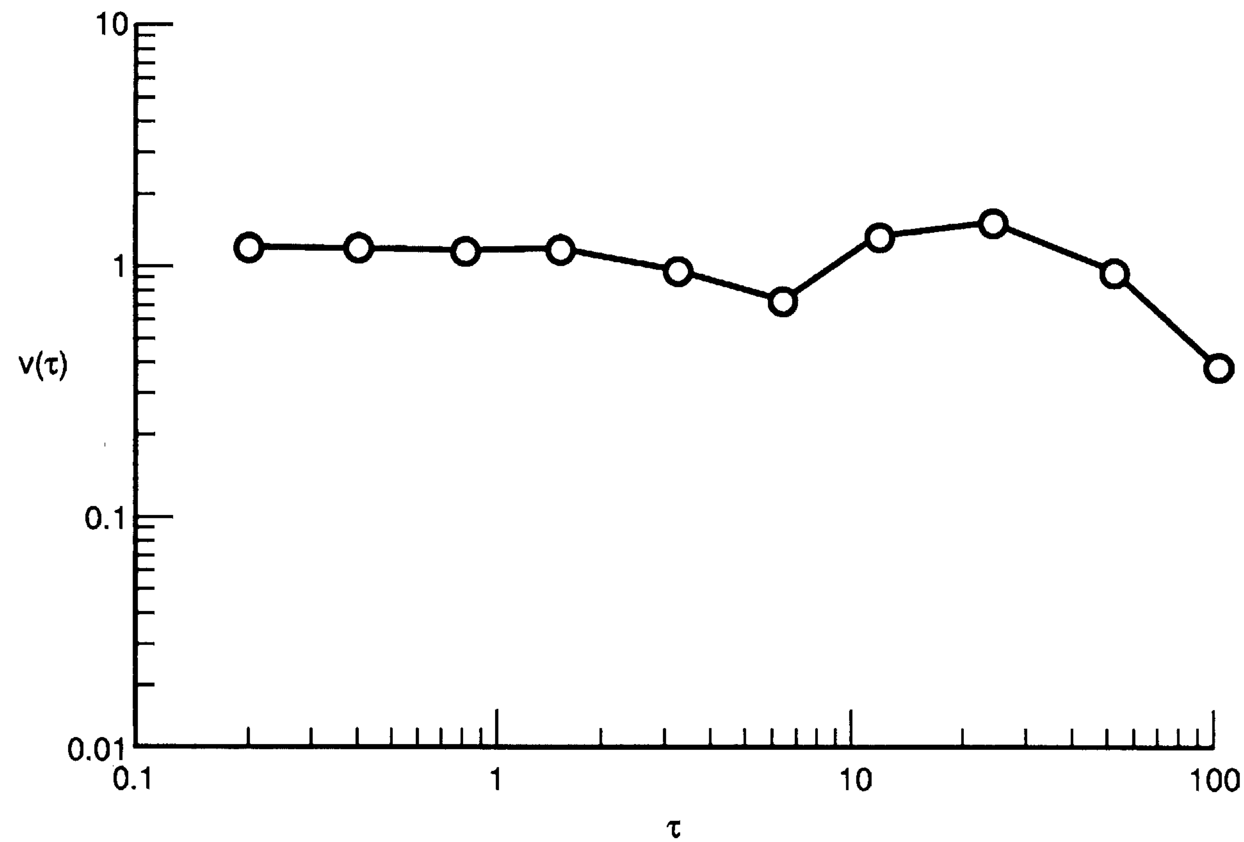


Figure 26. Plot for results of Figure 25.

```
CLS
DIM X(8192), H(8192), Y(8192), V(8192)
PRINT "PROGRAM ALLCON USING DIRECT CONVOLUTION": PRINT
PRINT "L", "INT TIME", "VARIANCE": PRINT
NP = 4096: LP = 12
FOR N = 0 TO NP - 1
X(N) = 100 * (RND - .5): Y(N) = 0: H(N) = 1: NEXT N
FOR N = 0 TO NP - 1
FOR I = 0 TO N
Y(N) = Y(N) + X(I) * H(N - I)
NEXT I
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N: N1 = NP
FOR L = 1 TO LP: N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT L, .2 * (2 ^ (L - 1)), VBAR
NEXT L: BEEP
```

PROGRAM ALLCON USING DIRECT CONVOLUTION

L	INT TIME	VARIANCE
1	.2	404.4132
2	.4	621.1478
3	.8	1031.323
4	1.6	1914.698
5	3.2	3831.717
6	6.4	8453.081
7	12.8	18015.49
8	25.6	25252.21
9	51.2	28800.24
10	102.4	90826.44
11	204.8	309915.6
12	409.6	699118.1

Figure 27. Program and data for radom walk noise with τ = 409.6 s and ALLCON.

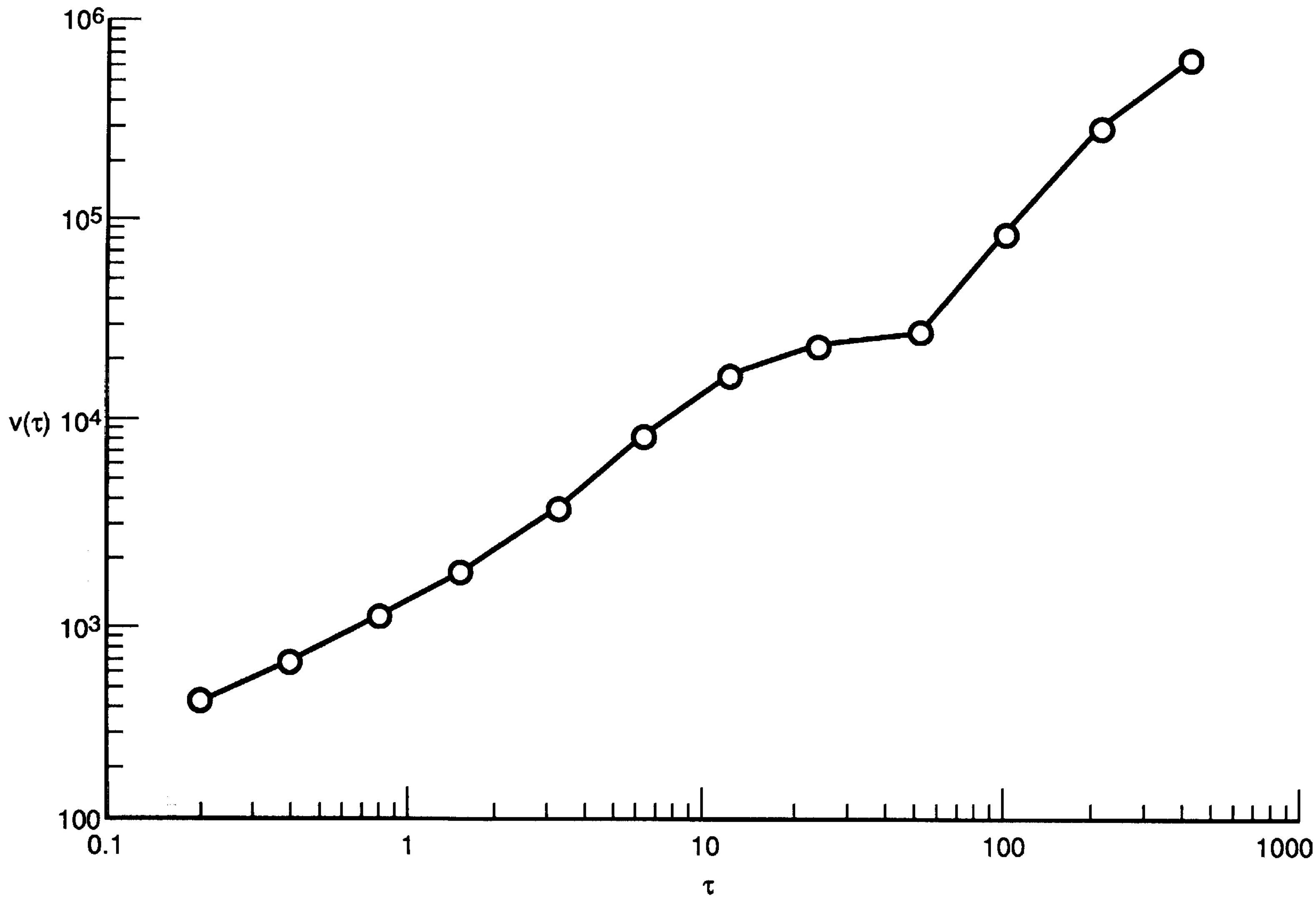


Figure 28. Plot for results of Figure 27.

```
CLS
DIM X(1024), H(1024), Y(1024), V(1024)
PRINT "M", "INT TIME", "VARIANCE"
PRINT
NP = 1024
FOR N = 0 TO 511
X(N) = 100 * (RND - .5): H(N) = 1: Y(N) = 0
NEXT N
FOR N = 512 TO NP - 1
X(N) = 100 * (RND - .5): H(N) = 1: Y(N) = 0
NEXT N
FOR N = 0 TO NP - 1
FOR I = 0 TO N
Y(N) = Y(N) + X(I) * H(N - I)
NEXT I
NEXT N
FOR N = NP TO 1 STEP -1
Y(N) = Y(N - 1)
NEXT N
N1 = NP
FOR M = 1 TO 10
N1 = N1 / 2
FOR N = 1 TO N1
V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2
NEXT N
VBAR = V(1)
FOR N = 2 TO N1
VBAR = VBAR + (V(N) - VBAR) / N
NEXT N
FOR N = 1 TO N1
Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N))
NEXT N
PRINT M, .2 * (2 ^ (M - 1)), VBAR
NEXT M
BEEP
```

M	INT TIME	VARIANCE
1	.2	394.7494
2	.4	610.4443
3	.8	1093.376
4	1.6	1712.925
5	3.2	3951.782
6	6.4	12355.76
7	12.8	15130.16
8	25.6	38779.27
9	51.2	64480.73
10	102.4	117919.4

Figure 29. Program and data for random walk noise with $\tau = 102.4$ s and ALLCON.

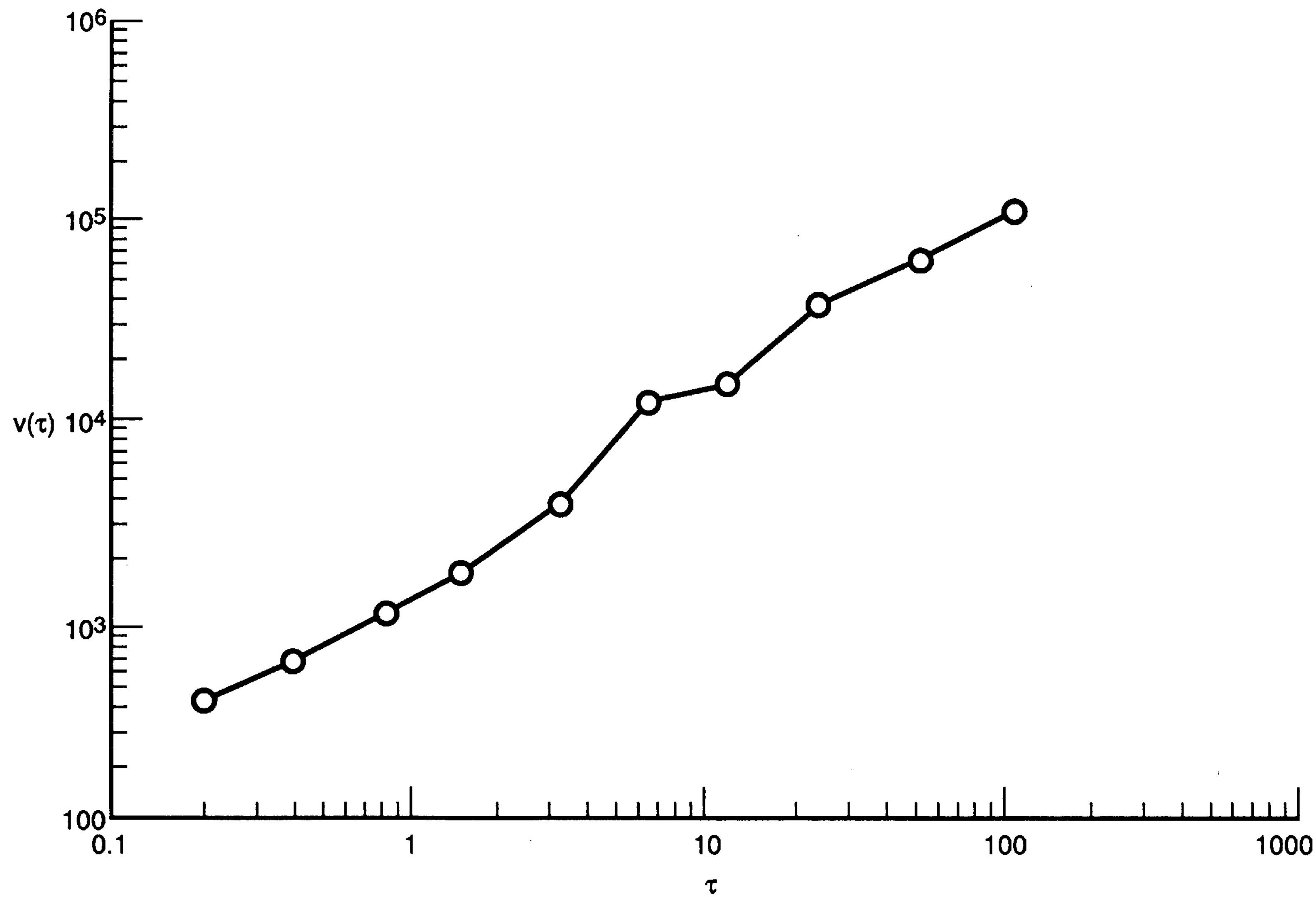


Figure 30. Plot for results of Figure 29.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048)
DIM H(2048), H1R(2048), H1I(2048), H2R(2048), H2I(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): H(N) = 1: NEXT N
FOR N = NH TO NP - 1: X(N) = 0: H(N) = 0: NEXT N
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: H1R(M) = H(M): H1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1: AR = H1R(M + IC): AI = H1I(M + IC)
BR = ZR * H1R(M + ID) - ZI * H1I(M + ID)
BI = ZI * H1R(M + ID) + ZR * H1I(M + ID)
H2R(M) = AR + BR: H2I(M) = AI + BI
H2R(M + NH) = AR - BR: H2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: H1R(M) = H2R(M): H1I(M) = H2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1
Y1R(M) = X1R(M) * H1R(M) - X1I(M) * H1I(M)
Y1I(M) = X1R(M) * H1I(M) + X1I(M) * H1R(M): NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 31. Program for random walk noise with τ = 102.4 s and ALLFFT.

PROGRAM ALLFFT USING FFT TO PERFORM CONVOLUTION

N	INT TIME	VARIANCE
1	.2	394.7613
2	.4	610.4606
3	.8	1093.399
4	1.6	1712.953
5	3.2	3951.828
6	6.4	12355.85
7	12.8	15130.14
8	25.6	38779.55
9	51.2	64481.73
10	102.4	117921.1

Figure 32. Data for random walk noise with τ = 102.4 s and ALLFFT.

```
CLS
DIM X(2048), X1R(2048), X1I(2048), X2R(2048), X2I(2048), HM(2048), HP(2048)
DIM Y(2048), Y1R(2048), Y1I(2048), Y2R(2048), Y2I(2048), V(2048)
PRINT "PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING": PRINT
PRINT "N", "INT TIME", "VARIANCE": PRINT : T = .2
NP = 2048: LP = 11: NH = NP / 2: PI = 3.14159: U = 2 * PI / NP
FOR N = 0 TO NH - 1: X(N) = 100 * (RND - .5): NEXT N
FOR N = NH TO NP - 1: X(N) = 0: NEXT N
FOR M = 1 TO NH - 1: HM(M) = 1 / M: HP(M) = 0: NEXT M
HM(0) = 0: HP(0) = 0: HM(NH) = 0: HP(NH) = 0
FOR M = NH + 1 TO NP - 1: HM(M) = HM(NP - M): HP(M) = 0: NEXT M
FOR M = 0 TO NP - 1: X1R(M) = X(M): X1I(M) = 0: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = -SIN(IC * U)
FOR M = IC TO ID - 1
AR = X1R(M + IC): AI = X1I(M + IC)
BR = ZR * X1R(M + ID) - ZI * X1I(M + ID)
BI = ZI * X1R(M + ID) + ZR * X1I(M + ID)
X2R(M) = AR + BR: X2I(M) = AI + BI
X2R(M + NH) = AR - BR: X2I(M + NH) = AI - BI: NEXT M
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR M = 0 TO NP - 1: X1R(M) = X2R(M): X1I(M) = X2I(M): NEXT M: NEXT L
FOR M = 0 TO NP - 1: HR = HM(M) * COS(HP(M)): HI = HM(M) * SIN(HP(M))
Y1R(M) = X1R(M) * HR - X1I(M) * HI
Y1I(M) = X1R(M) * HI + X1I(M) * HR: NEXT M: IA = NP / 2: IB = 1
FOR L = 1 TO LP: IC = 0: ID = IA
FOR K = 1 TO IB: ZR = COS(IC * U): ZI = SIN(IC * U)
FOR N = IC TO ID - 1: AR = Y1R(N + IC): AI = Y1I(N + IC)
BR = ZR * Y1R(N + ID) - ZI * Y1I(N + ID)
BI = ZI * Y1R(N + ID) + ZR * Y1I(N + ID)
Y2R(N) = AR + BR: Y2I(N) = AI + BI
Y2R(N + NH) = AR - BR: Y2I(N + NH) = AI - BI: NEXT N
IC = ID: ID = ID + IA: NEXT K: IA = IA / 2: IB = 2 * IB
FOR N = 0 TO NP - 1: Y1R(N) = Y2R(N): Y1I(N) = Y2I(N): NEXT N: NEXT L
FOR N = NP TO 1 STEP -1: Y(N) = Y1R(N - 1) / NP: NEXT N: N1 = NH
FOR L = 1 TO LP - 1: N1 = N1 / 2
FOR N = 1 TO N1: V(N) = .5 * (Y(2 * N - 1) - Y(2 * N)) ^ 2: NEXT N:
VBAR = V(1)
FOR N = 2 TO N1: VBAR = VBAR + (V(N) - VBAR) / N: NEXT N
FOR N = 1 TO N1: Y(N) = .5 * (Y(2 * N - 1) + Y(2 * N)): NEXT N
PRINT L, T * (2 ^ (L - 1)), VBAR: NEXT L: BEEP
```

Figure 33. Program for random walk noise with $\tau = 102.4$ s and ALLFFT1.

PROGRAM ALLFFT1 USING FFT WITH SPECTRUM SHAPING

N	INT TIME	VARIANCE
1	.2	2.777584E-03
2	.4	5.458743E-03
3	.8	1.051909E-02
4	1.6	2.125172E-02
5	3.2	3.561426E-02
6	6.4	7.147162E-02
7	12.8	.1857618
8	25.6	.4147886
9	51.2	.3673056
10	102.4	.3560083

Figure 34. Data for random walk noise with $\tau = 102.4$ s and ALLFFT1.

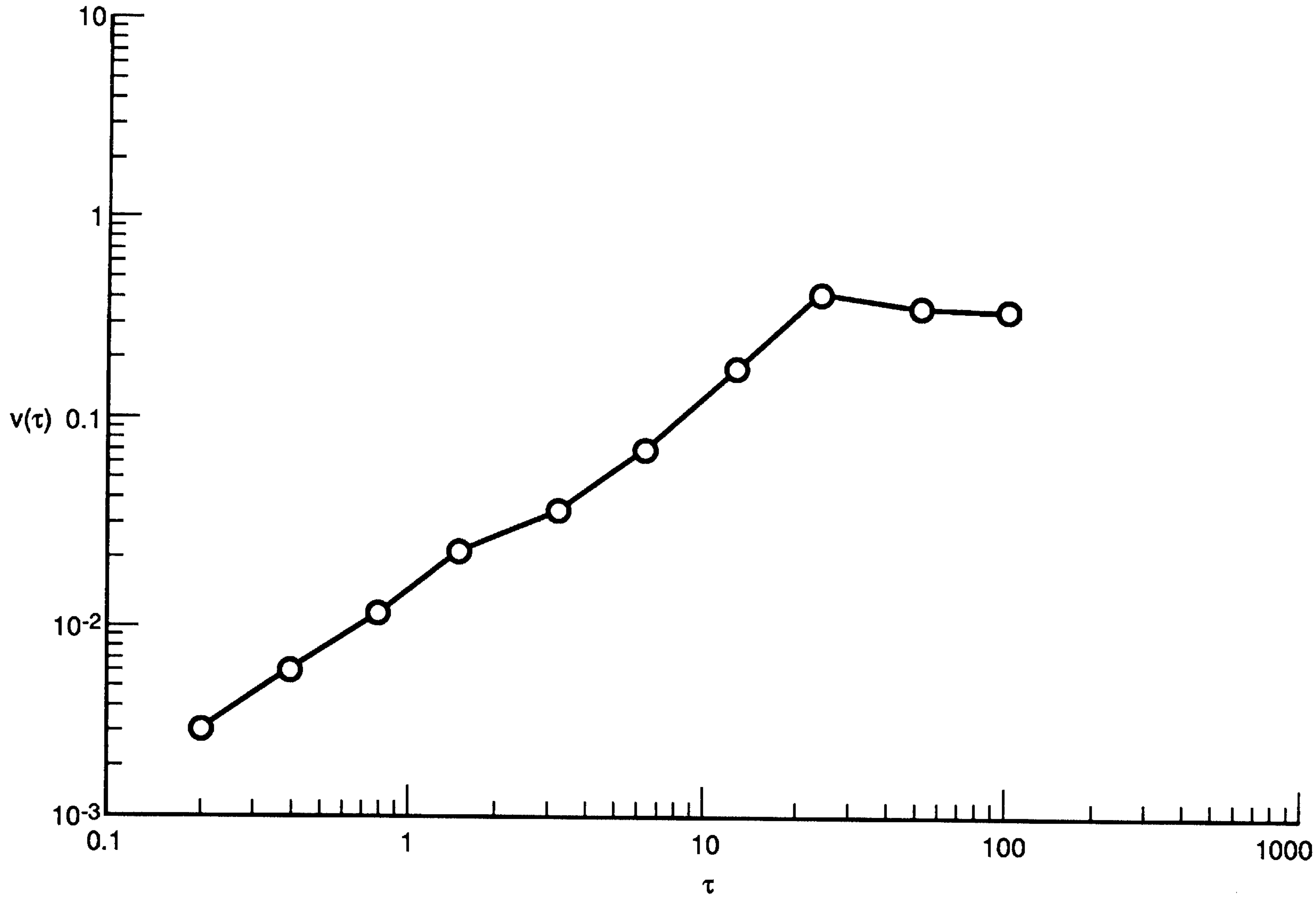


Figure 35. Plot for results of Figure 33.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
November 1994

3. REPORT TYPE AND DATES COVERED
Contractor Report

4. TITLE AND SUBTITLE
Investigation of Allan Variance for Determining Noise Spectral Forms with Application to Microwave Radiometry

5. FUNDING NUMBERS
C NAS1-19858
TA 12
WU 233-01-03-22

6. AUTHOR(S)
William D. Stanley

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Old Dominion University Research Foundation
Department of Engineering Technology
College of Engineering & Technology
Norfolk, Virginia 23529

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23681-0001

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
NASA CR-194985

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: L. C. Schroeder
Final Report - Task 12

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Unclassifed - Unlimited

Subject Category 43

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
An investigation of the Allan Variance method as a possible means for characterizing fluctuations in radiometric noise diodes has been performed. The goal is to separate fluctuation components into white noise, flicker noise, and random-walk noise. The primary means is by discrete-time processing, and the study focused primarily on the digital processes involved. Noise satisfying the requirements was generated by direct convolution, Fast Fourier Transformation (FFT) processing in the time domain, and FFT processing in the frequency domain. Some of the numerous results obtained are presented along with the programs used in the study.

14. SUBJECT TERMS
microwave radiometry, noise characterization, noise diode stability

15. NUMBER OF PAGES
65

16. PRICE CODE
A04

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		