Efficient detection of a continuous-wave signal with a linear frequency drift.

David H. Bailey [1] and Paul N. Swarztrauber [2,3]

SIAM J. Sci. Comput., **16**(1995), pp. 1233-1239

ABSTRACT

An efficient method is presented for the detection of a continuous-wave (CW) signal with a frequency drift that is linear in time. Signals of this type occur if the transmitter and receiver are rapidly accelerating with respect to one another, for example, as in interplanetary and space communications. We assume that both the frequency and the drift are unknown. We also assume that the signal is weak compared to the Gaussian noise. The signal is partitioned into subsequences whose discrete Fourier transforms provide a sequence of instantaneous spectra at equal time intervals. These spectra are then accumulated with a shift that is proportional to time. When the shift is equal to the frequency drift, the signal-to-noise ratio increases and detection occurs. In this paper, we show how to compute these accumulations for many shifts in an efficient manner using a variant of the FFT. Computing time is proportional to $L \log L$, where L is the length of the time series. Computational results are presented.

1 NASA Ames Research Center, Moffett Field, California 94035.

2 National Center for Atmospheric Research, Boulder, Colorado 80307, which is sponsored by the National Science Foundation.

3 This work was supported by the NAS Systems Division via Cooperative Agreement NCC 2-387 between NASA and the University Space Research Association (USRA). This work was performed while this author was visiting the Research Institute for Advanced Computer Science (RIACS), NASA Ames Research Center, Moffett Field, California 94035.

1. Introduction

The authors first learned of the problem of detecting signals with linearly drifting frequencies in discussions with scientists associated with the Search for Extraterrestrial Intelligence (SETI) project [3] at NASA Ames Research Center (see Acknowledgements). Signals with drifting frequencies result from continuous-wave (CW) transmissions between any locations that are accelerating with respect to one another. They are also of interest for applications where the acceleration is considerably reduced but a high level of accuracy is required, such as satellite-based aircraft navigational systems and similar systems that are currently under development for automobiles. We make the assumption that the period of observation is sufficiently brief that the drift is approximately linear.

It will be shown that detecting such a signal can be reduced to the problem of detecting a straight line in a noisy two-dimensional array. The techniques developed here for the efficient solution of this problem can be applied to a variety of problems, including the detecting of lines in a digitized photograph. The fast fractional Fourier transform (FFFT) is also based on these techniques [1].

For $l = 0, \ldots, L-1$, we are given a time series y_l that is generated from a noisy CW signal whose frequency is drifting linearly with time; that is,

$$y_l = Ae^{2\pi i(\omega_0 + \omega_1 t_l)t_l} + g_l \, . \tag{1.1}$$

The frequency at time $t_0 = 0$ is ω_0 and the frequency drift is ω_1. Both are measured in cycles per unit time. The term g_l is Gaussian noise with mean zero and standard deviation σ. We are interested in determining ω_0 and ω_1 in applications where A is much less than unity; that is, the amplitude of the signal is much less than the amplitude of the noise.

THE PROBLEM:

Given the time series y_l in (1.1), determine ω_0 and ω_1.

Once a signal is detected, its verification and analysis is relatively simple using tuned versions of the algorithm that will be presented in the later sections of this paper. Therefore, detection becomes the fundamental problem, and the primary goal is to maximize its likelihood. With a fixed computing resource, the following options are available:

1. Develop a new method of detection.

2. Speed the computation of an existing method and use higher resolution.

3. Improve the accuracy of the existing method.

In this paper, we will pursue options 2 and 3 in the context of an existing method described in [3] and [4] where the fast Fourier transform (FFT) is used to compute, but not accumulate, the spectra. We show that the FFT can be used in all phases of the computation. In particular, the FFT is used to compute certain spectral shifts with a high degree of accuracy at the same time that it is used to accumulate the shifts.

The approach in Cullers, Linscott, and Oliver, [3] is the accumulation of spectral subsequences to increase the signal-to-noise ratio. If L has factors $L = MN$, we can define M subsequences $y_{m,n} = y_l$, where $l = n + mN$, for $m = 0, \ldots, M-1$ and $n = 0, \ldots, N-1$. Each subsequence has N elements. The first step is to compute an approximate instantaneous spectra by computing the discrete Fourier transform (DFT) of each row of $y_{m,n}$; that is, compute M row transforms of length N,

$$Y_{m,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} y_{m,n} e^{-ikn\frac{2\pi}{N}} . \tag{1.2}$$

Next, compute the squared amplitude (power) of the spectra,

$$X_{m,k} = Y_{m,k} \overline{Y}_{m,k} . \tag{1.3}$$

If the frequency of the signal were constant (i.e., $\omega_1 = 0$), detection of the signal could be achieved by merely summing the columns of $X_{j,k}$ to increase the signal-to-noise ratio. The column $k_{\max}$ with the largest sum would correspond to the detected frequency. However as the frequency is not constant ($\omega_1 \neq 0$), this method fails because the column containing the signal differs from row to row and hence the column sums would not have a statistically large entry. Nevertheless, the signal is present in this array as a "line" with some slope that must also be determined. To this end, we select a test shift α, which is used to shift the mth row by $k+\alpha m$ and sum the resulting columns. That is, we compute

$$S_k(\alpha) = \sum_{m=0}^{M-1} X_{m,\,k+\alpha m}\,. \tag{1.4}$$

The shift α can be any real number because the shifted array is computed accurately using interpolation. If the correct shift α is selected, then the contribution of the signal will occur in the same column $k_{\max}$ for all m and will therefore be statistically evident as a maximum in the sums $S_k(\alpha)$. Therefore, the procedure is to try many values α_j and search the resulting array $S_k(\alpha_j)$ for a maximum, say $S_{k_{\max}}(\alpha_{\max})$. In what follows we will show that $\max\limits_{k,\alpha} S_k(\alpha) \approx S_{\omega_0}(2\omega_1)$. Therefore, once $k_{\max}$ and $\alpha_{\max}$ are determined, then $\omega_0 = k_{\max}$ and $\omega_1 = \alpha_{\max}/2$.

Without loss of generality, we can assume that the sequence y_l is tabulated at times $t_l = (n + mN)/N$ for which (1.1) takes the form

$$y_{m,n} = Ae^{2\pi i m(\omega_0 + m\omega_1)} e^{in\frac{2\pi}{N}(\omega_0 + 2m\omega_1 + \frac{n}{N}\omega_1)} + g_{m,n}\,. \tag{1.5}$$

We will assume that the frequency drift is negligible on each subsequence; that is, assume $n\omega_1/N = 0$ in (1.5). Then the DFT $Y_{m,k}$ provides an approximation to the "instantaneous" noisy spectra at time $t_l = m$. Further, from (1.2) and (1.5) it can be determined that the magnitude of signal in $Y_{m,k}$ has the following maximum at $k = \omega_0 + 2m\omega_1$:

$$Y_{m,\,\omega_0 + 2m\omega_1} = Ae^{2\pi i m(\omega_0 + m\omega_1)} + \text{noise}\,. \tag{1.6}$$

Substituting (1.6) into (1.3), we obtain

$$X_{m,\,\omega_0 + 2m\,\omega_1} = \left| A \right|^2 + \text{noise}\,. \tag{1.7}$$

From (1.7) and (1.4) the signal-to-noise ratio in $S_k(\alpha)$ is maximum at $S_{\omega_0}(2\omega_1)$. Therefore, as indicated, once $k_{\max}$ and $\alpha_{\max}$ are determined, then $\omega_0 = k_{\max}$ and $\omega_1 = \alpha_{\max}/2$.

2. The Algorithm.

The maximum of $S_k(\alpha)$ is located by tabulating $S_k(\alpha)$ at a set of points $\alpha_j = \alpha_0 + j\,\delta$ for $j=0, \ldots, M-1$. If we set $S_{j,k} = S_k(\alpha_j)$, then in the sections that follow we will be concerned with the efficient tabulation of a discrete version of (1.4), namely, (2.1).

Given α_0, δ, and the $M \times N$ matrix $X_{m,n}$ computed from (1.3), then we wish to compute another $M \times N$ matrix

$$S_{j,k} = \sum_{m=0}^{M-1} X_{m,\,k+m(\alpha_0+j\,\delta)}\,. \tag{2.1}$$

The jth row of $S_{j,k}$ contains the sum of the rows $X_{m,k}$ shifted by $m(\alpha_0+j\,\delta)$. The smallest shift between any two adjacent rows is δ. Fractional shifts are permitted and computed with a high degree of accuracy using trigonometric interpolation and the FFT.

Let $x_{m,n}$ be the row-wise forward Fourier transform of $X_{m,k}$. Then

$$X_{m,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x_{m,n}\, e^{ikn\frac{2\pi}{N}} \tag{2.2}$$

and

$$X_{m,\,k+m(\alpha_0+j\,\delta)} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} x_{m,n}\, e^{i[k+m(\alpha_0+j\,\delta)]n\frac{2\pi}{N}}\,. \tag{2.3}$$

From (2.1)

$$S_{j,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} \left[\sum_{m=0}^{M-1} \left(e^{imn\alpha_0 \frac{2\pi}{N}} x_{m,n} \right) e^{i\delta jmn \frac{2\pi}{N}} \right] e^{ikn \frac{2\pi}{N}} \qquad (2.4)$$

or

$$S_{j,k} = \frac{1}{N} \sum_{n=-N/2}^{N/2-1} s_{j,n} e^{ikn \frac{2\pi}{N}} \qquad (2.5)$$

where

$$s_{j,n} = \sum_{m=0}^{M-1} z_{m,n} e^{i\delta jmn \frac{2\pi}{N}} \qquad (2.6)$$

and

$$z_{m,n} = e^{imn\alpha_0 \frac{2\pi}{N}} x_{m,n} . \qquad (2.7)$$

Given α_0, δ, and $X_{m,k}$, the algorithm consists of the following steps:

A. Compute $x_{m,n}$ from the inverse of (2.2) using the FFT and compute $z_{m,n}$ from (2.7).

B. Next, compute $s_{j,n}$ using (2.6). With fixed n, each column can be transformed independently using a method that will be presented below.

C. Compute $S_{j,k}$ from (2.5) using the backward FFT applied to each row.

D. If $S_{jmax,\,kmax} = \max_{j,k} S_{j,k}$ then $\omega_0 = kmax$ and $\omega_1 = (\alpha_0 + jmax\,\delta)/2$

The FFT can be used for steps A and C. However, it cannot be used for step B because δ is not necessarily an integer. Nevertheless, we will show that (2.6) can be computed as efficiently as an M-point linear convolution using the FFT. We use a technique introduced by Bluestein [2] that enabled him to develop an FFT for arbitrary M including large prime numbers. The details of this algorithm are presented in [5] together with its implementation on multiprocessors. Bluestein's technique was also used to develop the fast fractional Fourier transform, which is described in [1].

The same transform (2.6) is applied to each column of the array $z_{m,n}$ and hence the exposition can be simplified by removing the subscript n. Therefore, we wish to compute

$$s_j = \sum_{m=0}^{M-1} z_m e^{i\beta jm \frac{2\pi}{N}} \tag{2.8}$$

where $\beta = \delta n$. We begin with the following identity:

$$2jm = j^2 + m^2 - (j-m)^2 \ . \tag{2.9}$$

Substituting into (2.8), we obtain

$$s_j = e^{i\beta j^2 \frac{\pi}{N}} \sum_{m=0}^{M-1} (e^{i\beta m^2 \frac{\pi}{N}} z_m) e^{-i\beta (j-m)^2 \frac{\pi}{N}} \tag{2.10}$$

or

$$s_j = b_j \sum_{m=0}^{M-1} b^{-1}_{|j-m|} c_m \tag{2.11}$$

where

$$b_m = e^{i\beta m^2 \frac{\pi}{N}} \ , \tag{2.12}$$

$$c_m = b_m z_m \ . \tag{2.13}$$

Equation (2.11) is preferable to (2.8) for computing s_j because (2.11) can be evaluated efficiently using the FFT. However an indirect approach is required because (2.11) is not a circular convolution; that is, the subscript $j-m$ is not interpreted modulo M. If $j-m$ is negative, then it is replaced by $m-j$ rather than $j-m+M$. Nevertheless, one can construct an equivalent circular convolution.

Select $P \geq 2M-2$ as a highly factorizable integer, such as a power of 2. Next, extend b_m and c_m to sequences with P elements as follows:

$$b_m = 0 \qquad M \leq m < P-M \,, \tag{2.15a}$$

$$b_m = b_{P-m} \qquad P-M \leq m < P \,, \tag{2.15b}$$

$$c_m = 0 \qquad M \leq m < P \tag{2.15c}$$

Define

$$\hat{s}_j = b_j^{-1} \sum_{m=0}^{P-1} b_{j-m} c_m \,, \tag{2.16}$$

where the subscripts are now interpreted modulo P. It can be determined by inspection that $s_j = \hat{s}_j$ for $j = 0, \ldots, M-1$. As (2.16) is a P-point circular convolution, it can be evaluated using the familiar FFT procedure [7]. The algorithm for fast evaluation of (2.11) can now be summarized.

B.0 Compute b_m from (2.12) and extend to length P using (2.15a) and (2.15b). Perform a P-point forward FFT and call the result B_k.

B.1 Compute c_m from (2.13) and extend to length P using (2.15c). Perform a P-point forward FFT and call the result C_k.

B.2 Compute $D_k = B_k C_k$. Perform a backward P-point FFT and call the result d_k.

B.3 Multiply the first M elements of d_k by b_j^{-1} to obtain $s_j = \hat{s}_j$. The remaining $P-M$ elements of d_k may be discarded.

Note that step B.0 is an initialization step that does not have to be repeated for the analysis of any subsequent time series.

3. Computational Results

The algorithm described in the preceding section has been implemented for test purposes. The resulting program is written entirely in Fortran without assembly code or assembly-coded library routines. Some modest effort was made to insure that the code is suitable for reasonably high performance execution on vector computers, but in general it is an entirely straightforward implementation of the algorithm presented in section 2.

Two details of the implementation are worth mentioning because they can result in a considerable savings of computing time. First, every FFT operation, including those that are a part of the convolution, was performed using ‘‘simultaneous’’ FFT routines. In other words, the FFTs were performed vectorwise instead of individually. Such an implementation is highly suited for a vector computer, but it is also quite appropriate for implementation on a parallel computer system. Second, it is important to note that a substantial amount of computing time can saved by employing real-to-complex FFTs in step A and by employing complex-to-real FFTs in step C above [6]. A by-product of this modification is that the FFTs required for the fractional transforms in step B need only be performed on $N/2+1$ columns.

The detection of the true signal base frequency and drift rate is enhanced in this implementation by employing a weighted score. This is motivated by the observation that a known input signal without noise will generate an output array with a peak that is smeared over a number of nearby elements. This pattern was used to design the improved detection filter in (3.1). A "bow-tie" stencil was used, and the coefficients were determined by observing the computed output from a known signal. The only coefficients that were significantly different from unity are written as such. This somewhat heuristic approach yielded the following weighted-score formula

$$T_{j,k} = 1.052 S_{j-2,k+1} + S_{j-1,k} + S_{j-1,k+1} + 1.216 S_{j,k} + S_{j+1,k-1} + S_{j+1,k} + 1.052 S_{j+2,k-1} \quad (3.1)$$

where $S_{j,k}$ is the final result of step C above. The statistical significances of these weighted scores are determined by explicitly computing the mean and standard deviation of all $T_{j,k}$.

Table 1 contains a detailed accounting of the computational cost of the complete detection algorithm. C-C, R-C, and C-R denote the three types of FFTs: complex-to-complex, real-to-complex, and complex-to-real. The column headed "Ref." lists references to specific equations and algorithm steps. The column headed "Operation Counts" contains the number of real floating-point arithmetic operations for each step, with adds, subtracts, and multiplies each counting as one operation.

Table 1
Floating-Point Operation Counts

Computational Step	Ref.	Operation Count
Initial row-wise C-C FFTs	(1.2)	$5MN\log_2 N$
Amplitude squared (power)	(1.3)	$3MN$
Forward row-wise R-C FFTs	A	$(5MN\log_2 N)/2+MN$
Columnwise multiplications	B.1	$3MN$
Columnwise forward C-C FFTs	B.1	$5MN\log_2 M+5MN$
Pointwise multiplications	B.2	$6MN$
Columnwise inverse C-C FFTs	B.2	$5MN\log_2 M+5MN$
Columnwise multiplications	B.3	$3MN$
Backward row-wise C-R FFTs	C	$(5MN\log_2 N)/2+MN$
Weighted scores and statistics	(3.1)	$13MN$
Total		$MN(10\log_2 MN+55)$

For exposition we assumed that $X_{m,k}$ is complex; however, a real version of the algorithm could be developed that would reduce the amount of computation. In particular, a real-to-complex conjugate FFT would take about half the time required by the complex FFT [6].

This algorithm was exercised by performing a series of tests with pseudorandomly selected initial frequencies and drift rates. In these tests, M and N were each set to

1,024. The selection $M = N$ is plausible but otherwise arbitrary. The initial value of the drift rate α was set to $-1/2$, and the drift rate increment δ was set to $-1/M$. Thus, the sums $S_k(\alpha)$ in (1.4) were computed for all drift rates α from $-1/2$ to $1/2$. For each test, a linearly drifting signal was added to complex Gaussian pseudorandom data of the form $g_l = (e_l + i f_l)/\sqrt{2}$, where e_l and f_l are real Gaussian data with zero mean and unit variance. The ratio of the amplitudes of the noise and the signal was 56.6, so that the power of the signal was lower than that of the noise by a ratio of 3,200. This extremely low signal-to-noise level is approximately the level that the Cyclops SETI system hopes to detect [3].

Table 2
Algorithm Test Results

Trial No.	Generated Base Freq.	Generated Drift Rate	Detected Base Freq.	Detected Drift Rate	Z-score	Time
1	119.842	-0.26187	119.00	-0.2607	7.812	2.342
2	967.298	0.49238	968.00	0.4922	7.266	2.423
3	706.734	0.04648	706.00	0.0479	6.976	2.364
4	828.500	0.05203	828.00	0.0537	7.177	2.293
5	97.555	0.35731	98.00	0.3564	7.124	2.430
6	626.842	-0.25255	627.00	-0.2529	6.485	2.472
7	989.664	-0.14073	989.00	-0.1396	7.836	2.476
8	203.058	-0.42424	203.00	-0.4248	8.902	2.592
9	1022.170	-0.10787	1022.00	-0.1074	8.451	2.315
10	921.632	0.43326	922.00	0.4336	10.460	2.550
Ave.					7.849	2.429

These tests were performed on the Cray-2 operated by the NAS Systems Division at NASA Ames. The results of these tests are displayed in Table 2. The column headed ‘‘Base Freq.’’ is ω_0 and the column headed ‘‘Drift Rate’’ is $2\omega_1$. The column headed ‘‘Z-score’’ gives the Z-score of the detection; that is, the number of standard deviations above the mean. The column headed ‘‘Time’’ contains single-processor CPU times in seconds. These results indicate that the detection algorithm is effective in determining the unknown base frequency and drift rate. In every trial, the element of the final array with the highest weighted score was within one interval both in drift rate and frequency from the true value. These Z-scores are well above the level of random scores, as a random Z-score would not be expected to exceed 5 in a 1024×1024 array.

The average processing CPU time, 2.429 seconds, corresponds to a performance rate of approximately 109 MFLOPS, based on the total number of floating-point operations given in Table 1. This performance rate could be increased by employing assembly-coded simultaneous FFT routines and by employing all four of the Cray-2 processors. In any event, these timings indicate that such computations are feasible given suitably fast computer hardware.

As was mentioned, the central problem is making an accurate initial detection. The algorithm presented here accomplishes this basic objective. Once a detection has been made, its confidence level can be increased in a number of ways, such as by repeating our algorithm with higher resolution near the detection, or by employing a matched filter keyed to the frequency and drift rate. Perhaps the most effective approach is to multiply both sides of (1.5) by $\exp[-in\frac{2\pi}{N}\frac{n}{N}\omega_1]$ and repeat the algorithm to compute a new approximation to both ω_0 and ω_1. This defines an iterative procedure that removes the assumption made following equation (1.5). Although this variant can be used to improve the confidence level it does not improve the likelihood of detection. However, we expect to increase the probability of detection with future improvements in the algorithm.

Acknowledgements

We wish, in memoriam, to thank Oscar Buneman for his valuable contributions to this manuscript. We also wish to thank Palmer Dyal for bringing the SETI problem to our attention and both Kent Cullers and Peter Backus for providing the details of the problem and reference [3].

References

[1] D. H. Bailey and P. N. Swarztrauber, The fast fractional Fourier transform and applications, *SIAM Rev.*, 33(1991), pp. 389-404.

[2] L. I. Bluestein, A linear filtering approach to the computation of the discrete Fourier transform, *IEEE Trans. Audio Electroacoust.*, vol AU-18, 4(1970), pp. 451-455. Reprinted in Digital Signal Processing, ed. L. R. Rabiner and C. M. Rader, pp. 317-321, New York: IEEE press, 1972.

[3] D. K. Cullers, I. R. Linscott, and B. M. Oliver, Signal processing in SETI, *Comm. ACM* 28(1985), pp. 1151-1163.

[4] I. Gertner, Optimal detection and separation of chirp signals, *IEEE Int. Conf. Acoust. Speech Signal Proc.*, April 3-4, 1990, Albuquerque, N.M.

[5] H. J. Nussbaumer, *Fast Fourier Transform and Convolution Algorithms*, New York: Springer-Verlag, 1981.

[6] P. N. Swarztrauber, Symmetric FFTs, *Math. Comp.* 47(1986), pp. 323-346.

[7] P. N. Swarztrauber, R. A. Sweet, W. L. Briggs, V. E. Henson and J. Otto, Bluestein's FFT for arbitrary N on the hypercube, *Parallel Computing* 17(1991), pp. 607-617.