An Algorithm for Extraction of Periodic Signals From Sparse, Irregularly Sampled Data

J. Z. Wilcox
Tracking Systems and Applications Section

Temporal gaps in discrete sampling sequences produce spurious Fourier components at the intermodulation frequencies of an oscillatory signal and the temporal gaps, thus significantly complicating spectral analysis of such sparsely sampled data. A new fast Fourier transform (FFT)-based algorithm has been developed, suitable for spectral analysis of sparsely sampled data with a relatively small number of oscillatory components buried in background noise. The algorithm's principal idea has its origin in the so-called "clean" algorithm used to sharpen images of scenes corrupted by atmospheric and sensor aperture effects. It identifies as the signal's "true" frequency that oscillatory component which, when passed through the same sampling sequence as the original data, produces a Fourier image that is the best match to the original Fourier map. Unlike the clean algorithm, it performs the search in the Fourier space. The algorithm has generally met with success on trials with simulated data with a low signal-to-noise ratio, including those of a type similar to hourly residuals for Earth orientation parameters extracted from VLBI data. For eight oscillatory components in the diurnal and semidiurnal bands, all components with an amplitude-noise ratio greater than 0.2 were successfully extracted for all sequences and duty cycles (greater than 0.1) tested; the amplitude-noise ratios of the extracted signals were as low as 0.05 for high duty cycles and long sampling sequences. When, in addition to these high frequencies, strong low-frequency components are present in the data, the low-frequency components are generally eliminated first, by employing a version of the algorithm that searches for noninteger multiples of the discrete FFT minimum frequency.

I. Introduction

In observational sciences like astronomy, it frequently happens that data are available at an unevenly spaced set of sampling times. The analyst may wish to extract from these data certain periodic components for which the frequencies are not precisely known. In fact, signal extraction from sparse or unevenly sampled data may prove crucial to the successful exploitation of large data sets assembled at considerable expense over many years. If the data acquisition process is characterized by a regular or nearly periodic succession of gaps during which no data are taken, sidelobes will occur at frequencies corresponding to the intermodulation products of the various signal and gap frequencies, giving rise to spurious signals in the Fourier map. The situation is aggravated by the inevitable presence of noise. In recent years, a number of authors have developed methods based on Lomb's normalized periodogram to deal with problems of this kind. A set of references and an implementation of Lomb's method are provided in [1]. A recent

article studied the effect of temporal windows on observed solar oscillation parameters [2]. However, there appears to be no currently available method to search for periodic signals with a low signal-to-noise ratio and unknown frequencies spanning a potentially broad spectral range; nor does there exist a clear understanding of the mathematical limits on the kinds of information that might be extracted from such data.

This article describes a new mathematical procedure for extraction of periodic signals from incomplete data sets. The procedure addresses a situation of particular interest in which only a relatively small number of periodic components are present within a noisy background. Although it may not be possible even in principle to extract a large array of broadband signals from sparsely sampled data, the fact that only a small number of components are present makes it possible to extract the components in a manner that effectively eliminates ambiguity. The main idea of the new procedure comes from the so-called "clean" algorithm [3], which has been extensively used within the astronomical community to sharpen images of scenes corrupted by atmospheric and sensor aperture effects. Clean algorithms have been most successful in cleaning up maps of images created by a small number of intense sources. The location of the most intense point source is identified with the highest peak in the scene's "dirty" map. By modeling the effects of the atmosphere and sensor aperture (or other intermediate filters) on the image of a single localized source, the resulting point spread function is subtracted off, and a point reconstruction added to a clean map. The process is repeated for successively smaller peaks.

Analogous to the clean algorithm, the new algorithm identifies the strongest oscillatory component and produces a Fourier image of this signal as it would be registered if it were sampled in the same manner as the actual data. Unlike in the clean algorithm, a search for the component is performed in Fourier space. That frequency is identified as the frequency of the strongest oscillatory component whose Fourier image is the best match to the Fourier map of the actual data. The image is subtracted from the original Fourier map. The difference between the old map and the subtracted image is a new residual map, and the amplitude and frequency of the generating signal are recorded as contributors to the clean map.

The algorithm has generally met with success in trials with simulated data, including those of a type similar to hourly residuals for Earth orientation parameters extracted from VLBI data. Some eight signal components in the diurnal and semidiurnal bands with amplitudes varying by an order of magnitude were added to a background of Gaussian noise on the same order of magnitude as the oscillatory signal. For a 2-year-long sampling sequence and 20-percent duty cycle (the duty cycle is defined as the ratio of the length of time during which the data was taken to the sequence's total length), all components with an amplitude–noise ratio greater than about 0.2 were successfully extracted; for a 100-percent duty cycle, the extracted amplitude–noise ratio was as low as 0.05. To minimize central processing unit (CPU) time (important for long data sequences), the algorithm uses fast Fourier transforms (FFTs) on uniformly spaced time markers. The data arrays have zeroes added into the temporal gaps and into the appended region (which is added to make the array size equal to a power of two). The CPU time scales as a constant $\times N \times \log\ N$, where the constant depends on the efficiency of the search for the signal oscillatory frequencies. For $N = 2^{18}$ time markers and 8 cleaning steps, the required CPU time on a VAX-class machine was on the order of several minutes. The frequency resolution is $1/T$, where T is the sampling sequence length. For cases when a more precise frequency determination is required, or when in addition to the high (i.e., diurnal and semidiurnal) frequencies, the data contains low-frequency components, a noninteger version of the algorithm has been developed. The noninteger version constructs the Fourier image as a function of frequencies that are not integer multiples of the discrete FFT minimum frequency. An additional search, performed by using a functional minimum-finding algorithm, extends the search to noninteger frequency values. Compared to the integer algorithm, the noninteger algorithm determines the signal frequencies with greatly improved accuracy. It also consumes more CPU time, depending on the desired accuracy and on the spectral content of the original data.

Section II describes the algorithm; Section III describes results of simulation experiments; and the Appendix describes the algorithm extension to frequencies that are not equal to discrete FFT frequencies.

II. The Algorithm

The data are sampled in a sequence that consists of discrete observation sessions (or windows, on the order of one day) between which there are large gaps (on the order of several days). Within each session, data are sampled at uniformly spaced discrete time intervals that are integers of some minimum δt (in our case, 1 hr). The set of hour markers at which the data are taken defines the sampling function $S(t_n)$, which assumes values of 1 or 0 depending upon whether or not a data value was acquired at that particular time. The data define the array $D(t_n)$, in which the entries where no data have been taken have been set to zero. The hour markers are $t_n = n\ \delta t$, $n = 0, 1, \cdots, N-1$, where N is the total number of markers in the sampling sequence of duration $T = N\ \delta t$. Note that, to use the FFT, it is most convenient for N to be chosen equal to a power of 2.

Sampling at discrete intervals creates a situation whereby frequencies beyond the maximum resolved frequency (Nyquist, equal in our case to $1/2\delta t$) are aliased (or "folded back") into the lowest frequency region. In addition, the presence of periodic or nearly periodic gaps in the data sequence leads to the appearance of sidelobes and spurious spectral peaks at the signal and gap intermodulation frequencies. The sidelobes are also aliased into the lowest spectral region. To remove the sidelobe effect, the algorithm produces a Fourier map of the actual data and identifies the strongest oscillatory component. It calculates a Fourier image of this signal filtered through the same sequence, $S(t_n)$, as the actual data, and subtracts it from the original Fourier map to produce a residual map. The most obvious candidate for the strongest component is a frequency with the biggest peak in the original Fourier map. Occasionally, however, the biggest peak occurs at one of the intermodulation frequencies. Therefore, the "true" modulation frequency is found to be that frequency which when passed through $S(t_n)$ is the best match (including the sidelobe structure) for the original Fourier map.

Let M_l designate the least squares function:

$$M_l \ \equiv \ \sum_k^N |D^S(k) - \sum_{\pm} a_{\pm l}\ f^S_{\pm l}(k)|^2 \tag{1}$$

where the sum over k is a sum over all (discrete) frequencies, $\omega_k \equiv 2\pi k/(N\delta t)$. The $D^S(k)$ is a discrete Fourier transform of $D(t_n)$, and the second term on the right-hand side of Eq. (1) is a Fourier representation of a candidate oscillatory signal of frequency ω_l taken over the sampling function $S(t_n)$. That is, $f^S_l(k)$ is a Fourier transform of $f_l(t_n)\ S(t_n)$, where the periodic function $f_l(t_n) = \exp\left(-2\pi\ i\frac{nl}{N}\right)$. Note that if there were no gaps in the data (i.e., if $S(t_n)$ consisted of an uninterrupted series of data points), then $f^S_l(k)$ would consist of a single Kronecker delta, $\delta_{l,k}$. However, because of the window-gap structure of $S(t_n)$, the $f^S_l(k)$'s will have sideband components, similar to sideband components of $D^S(k)$.

The a_l's specify amplitudes of the Fourier components $f^S_l(k)$. It is important that the Fourier representation of a candidate oscillatory signal sums over both the positive and negative frequency indexes to account for signal phase. To see this, assume that the contribution of the lth oscillatory component to data is

$$A_l \cos\left(2\pi\ \frac{n\ l}{N} + \phi_l\right) \tag{2}$$

where ϕ_l is phase. The contribution to the Fourier map is

$$\frac{A_l}{2}\left(e^{i\phi_l} f^S_{-l}(k) + e^{-i\phi_l} f^S_l(k)\right) \tag{3}$$

Because Eq. (3) consists of both a positive and its complex conjugate negative frequency components, so must the Fourier image, $\sum_{\pm} a_{\pm l}\ f^S_{\pm l}(k)$, in Eq. (1).

The index l that minimizes M_l as a function of l is identified as the signal's true frequency. To perform the minimization, $D^S(k)$'s and $f^S_l(k)$'s are computed (for all k's and candidate l's). The coefficients $a_{\pm l}$ are computed from the following complex conjugate relationships for spectral strengths at $\pm l$ locations in the Fourier map:

$$D^S(l) \ = a_{-l} f^S_{-l}(l) + a_l\ f^S_l(l) \tag{4a}$$

$$D^S(-l) \ = a_{-l} f^S_{-l}(-l) + a_l\ f^S_l(-l) \tag{4b}$$

By combining Eqs. (4a) and (4b), the $a_{\pm l}$'s are computed for the candidate l. The computed $a_{\pm l}$'s are substituted into Eq. (1), whose minimization yields the optimal index l for the frequency of the strongest oscillatory component. Note that the use of the positive and negative l allows for reconstruction of the signal phase. By comparing Eqs. (3) and (4), the signal phase ϕ_l is given by the following expression:

$$\tan \phi_l = \frac{\text{Im } a_{-l}}{\text{Re } a_{-l}} \tag{5}$$

Hence, the use of the above-described procedure allows one to determine signal phase.

The above relationships, Eqs. (4) and (5), were derived by relating the spectral strengths at $\pm l$ locations. It is possible to formally derive Eq. (4) by minimization of Eq. (1) as a function of a_l. The minimization is performed in the Appendix, together with the extension of the algorithm to noninteger l values. Note also that the use of the $a_{\pm l}$'s computed from Eq. (4) permits complete subtraction of the spectral strengths at the $\pm l$ locations; if the data consist of a single frequency, then one subtraction yields a residual at the level of the machine round-off error or background noise, whichever is bigger.

For data modulated with several frequencies, multiple applications of the algorithm are necessary. Equations (1)–(5) will apply, except that on the $(i+1)$th iteration, $D^S(k)$ is replaced by a residual array $(y_i(k))$ that has resulted from the ith iteration, $y_i(k) \equiv y_{i-1}(k) - \sum_{\pm}\ a_{\pm l_i}\ f^S_{\pm l_i}(k)$, where l_i is the optimal index that minimizes M_l at the ith iteration, $y_{i-1}(k)$ is the residual array that has resulted from the $(i-1)$th iteration, and $y_o(k) \equiv D^S(k)$ is the Fourier transform of the original data. With multiple iterations, it may sometimes be desirable to subtract the spectral peaks in fractions. That is (as is a common practice in the application of the clean algorithm), one subtracts at each iteration only a fraction of the image that has been constructed by using the peak's full strength. However, the phase relationship between the subtracted $a_{\pm l}$'s should be the same as that computed from Eq. (4). Sidelobes associated with one frequency can contribute signal at another frequency. Subtracting the peaks in fractions permits one to correct for sidelobe contributions associated with other signal frequencies. In any case, the algorithm first removes (or partially removes) the most intense component (including sidebands due to the window-gap structure), forms the residual array, and treats the residual as a new set of data. The process is iterated to yield a desired number of oscillatory components.

III. Results and Discussion

Two versions of the above-described general procedure have been implemented and tested on simulated data. In the first version, the oscillatory signal that minimizes Eq. (1) consists of two integers, $\pm l$, as in Eq. (3). This version works very well when all modulation frequencies are equal (or nearly equal) to discrete multiples of $1/T$. If T is the total length of the sampling sequence, then the spacing for the

resolved frequencies is $1/T$, corresponding to the resolution parameter of τ^2/T for an oscillatory signal with period τ. Thus, this version provides a sufficient resolution for relatively high frequencies, say, $\tau < 0.01\ T$ (in which case, the resolution is better than 1 percent). Because the strongest components are extracted first, the accuracy with which these components are extracted impacts the extraction success of less intense components. The integer version works best when all (relatively rapid) modulation amplitudes are about the same order of magnitude. The second version of the algorithm searches for frequencies that are not integer multiples of $1/T$ (or, equivalently, the frequency indices are not integer multiples of $1/N$). This version allows one, at the expense of increased CPU time, to determine the oscillatory frequencies with greatly improved accuracy. This version must be used for analysis of data with strong low-frequency components.

A. Integer Frequency Index

The integer version was tested with success on many different types of simulated monochromatic as well as multifrequency signals; the modulations were imbedded in Gaussian noise. The success of frequency extraction depends on the number of modulation frequencies, the length and structure of the sampling sequence (including length, number, and regularity of individual observational sessions), and noise level.

For a monochromatic signal (data sampled hourly), the modulation frequencies ranged from those corresponding to periods of 4 hr to those corresponding to the length T of the data sequence. Table 1 summarizes the results of the simulations for $S(t_n)$ consisting of 20 and 200 1-day-long, periodically repeating sessions separated by several day-long gaps. The residual has been reduced to a background level by using a single subtraction step for signal-to-noise amplitude ratios as low as 0.17 and 0.04, respectively, when the duty cycle was 1; the minimum values of the successfully extracted ratios were 0.3 and 0.1, and 0.5 and 0.125, for the duty cycles of 0.2 and 0.1, respectively. Here, the duty cycle has been defined as the ratio of the number of days on which data were recorded to the total number of days in the observation period (e.g., for a 1-day-long observation session registered on every 5th day, the duty cycle is 1/5). Results of simulations for nonperiodic sampling sequences were similar, except that the extraction tended to be successful even at somewhat higher noise levels. This is not unexpected, since constructive interference (leading to strong sidelobes) is less.

Results of simulation experiments for a signal consisting of several modulation frequencies are summarized in Tables 2 and 3 and Fig. 1. Eight oscillatory components (four in the semidiurnal band and four in the diurnal band) with amplitudes varying by an order of magnitude between the strongest and weakest component were added to a background of Gaussian noise. Typical $S(t_n)$'s consisted of 70, 140, and 740 1-day-long observation sessions (data sampled hourly) separated by 4-day- (duty cycle 0.2) and 9-day-long (duty cycle 0.1) gaps. The noise amplitude was as high as 100 (the amplitudes of the oscillatory components ranged from 3 to 23). The amplitudes and relative phase of the simulated signals are summarized in Table 2. To demonstrate the resolution for the extracted signals, the table also summarizes the results of the extraction process for the 140–data day, 0.2–duty cycle sampling sequence, with the noise level equal to the amplitude of the strongest component. All eight components were extracted within their resolution accuracy (for $T = 2^{16}$ hr, the resolution $\tau^2/T \simeq 0.002$ and 0.008 for semidiurnal and diurnal bands, respectively) in eight iteration steps. Figure 1(a) shows the original data, and Figs. 1(b), (c), and (d) show, respectively, the original, residual, and clean Fourier maps.

Results of simulations for the same oscillatory components but different observation parameters are summarized in Table 3. To investigate the effect of excess iterations, the number of cleaning steps used was 10. In Table 3, the symbol s designates an oscillatory component extracted within its expected resolution, c designates an extraction of a previously extracted component, i designates extraction at one of the intermodulation frequencies, and n designates noise (any frequency higher than $1/6\ \mathrm{h}^{-1}$ was considered a noise). A component may be extracted twice (or several times) because of our use of the full spectral strength $D^S(\pm l)$ [Eq. (4)] for the subtraction of the Fourier image. The strength $D^S(\pm l)$ consists of contributions from the lth oscillatory component as well as from sidelobes produced by other frequencies

Table 1. Extraction results for simulated monochromatic data.[a]

Duty cycle	Minimum extracted signal–noise amplitude ratio	
	N data days	
	20	200
1	≥0.17	≥0.04
0.2	≥0.3	≥0.1
0.1	≥0.5	≥0.125

[a] The listed numbers are the minimum values for the signal–noise amplitude ratio for which the extraction was successful. (The signal periodicity was $\tau = 11.9672$ hr, data sampled hourly.)

Table 2. Extraction results for simulated data with four oscillatory components in the diurnal and four in the semidiurnal band.[a]

Periodicity, hr	Relative amplitude	Phase, deg
Simulated data		
11.9672	17.0	126
12.0000	8.6	92
12.4206	3.7	86
12.6583	3.2	103
23.9345	23.0	200
24.0659	7.0	240
25.8193	19.0	69
26.8684	5.3	230
Extracted signals		
11.9679	16.4	139
12.0007	9.2	104
12.4215	4.1	103
12.6591	4.2	106
23.9357	22.8	205
24.0676	6.0	243
25.8219	20.0	80
26.8700	6.2	243

[a] The background noise relative amplitude was 30. The sampling sequence consisted of 140 data days separated by 4-day-long gaps (duty cycle 0.2). Data were sampled hourly; total number of hour markers in the array was 2^{16}). The frequencies were extracted within the resolution of $12^2/2^{16}$ and $24^2/2^{16}$ for the semidiurnal and diurnal bands, respectively.

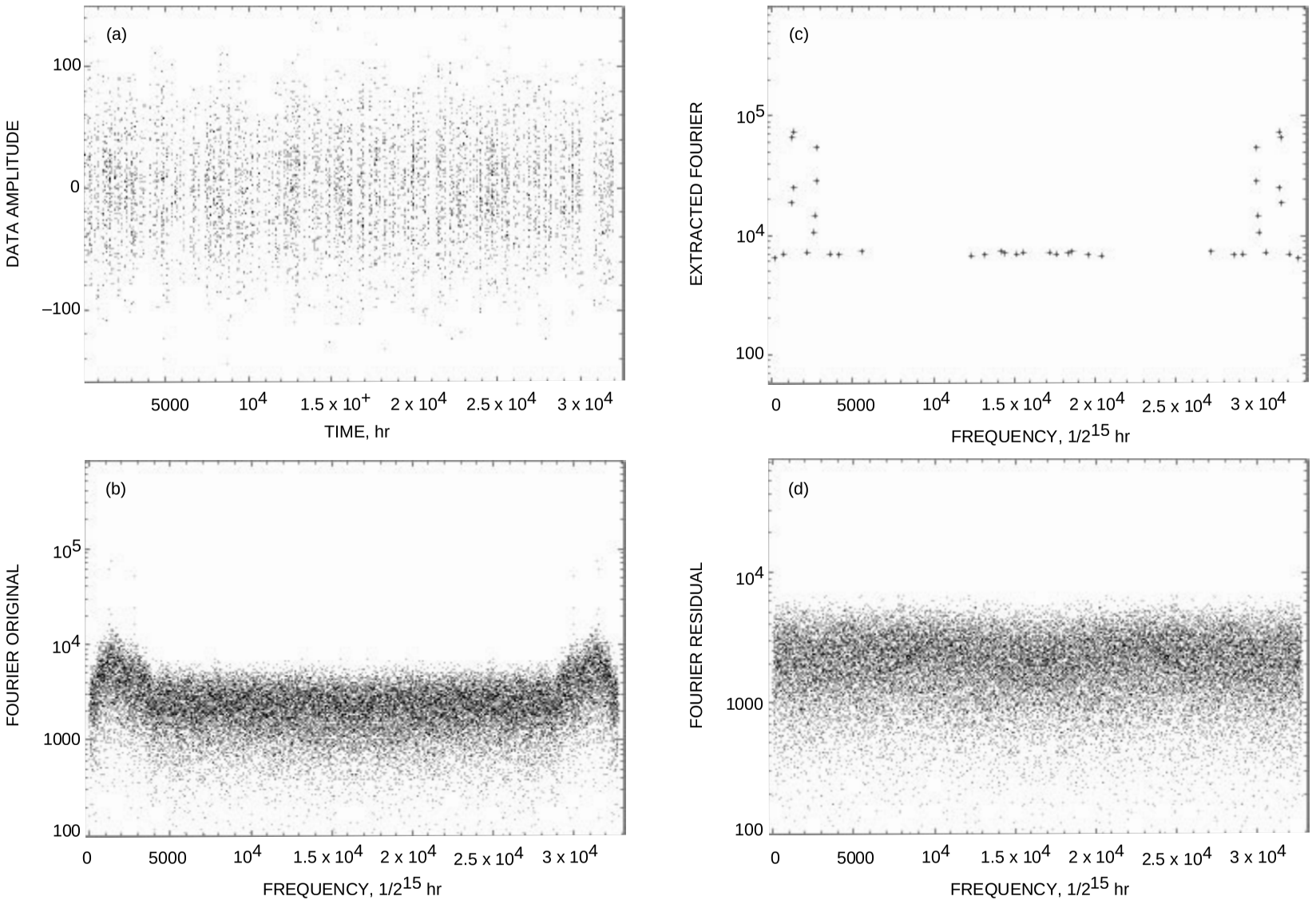

Fig. 1. Results of simulation experiments: (a) the original data, (b) the original Fourier map, (c) the clean Fourier map, (d) the residual Fourier map for simulated data with four diurnal and four semidiurnal signals (parameters of Table 2) buried in the background of Gaussian noise. The noise level was 30, the duty cycle was 0.2, the number of data days was 270 (data sampled hourly). The signals were extracted in eight iteration steps.

Table 3. Extraction results for the same eight oscillatory components as in Table 2 for periodically repeating observational sessions.

Noise level	Duty cycle	N data days		
		740	140	70
0	1	$8s$, $2c$	$8s$, $2c$	$8s$, $2i$
	0.2	$8s$, $2c$	$8s$, $2c$	$8s$, $2c$
	0.1	$8s$, $2c$	$8s$, $2c$	$7s$, $3c$
10	1	$8s$, $2c$	$8s$, $2c$	$8s$, $2i$
	0.2	$8s$, $2c$	$8s$, $2c$	$8s$, $1c$, $1i$
	0.1	$1i$, $3s$, $5i$, $1s$ [$5s$, $2c$, $3s$)	$8s$, $2c$ [$8s$, $2c$]	$8s$, $2i$ [$8s$, $1c$, $1i$]
30	1	$8s$, $2c$	$8s$, $2n$	$6s$, $4n$
	0.2	$8s$, $2c$ [$8s$, $2c$]	$7s$, $1i$, $2n$ [$8s$, $2n$]	$6s$, $1i$, $3n$ [$7s$, $3n$]
	0.1	$1i$, $3s$, $5i$, $1s$ [$3s$, $1c$, $5s$, $1c$]	$6s$, $2i$, $2n$ [$8s$, $2n$]	$5s$, $1i$, $4n$ [$6s$, $1i$, $3n$]
100	1	$6s$, $2i$, $2n$	$4s$, $6n$	$3s$, $7n$
	0.2	$5s$, $1c$, $3i$, $1n$ [$6s$, $2n$, $2i$]	$4s$, $1n$, $5i$ [$4s$, $1i$, $5n$]	$3s$, $7n$ [$3s$, $7n$]
	0.1	$3s$, $1n$, $6i$ [$5s$, $2c$, $3n$]	$3s$, $2n$, $5i$ [$3s$, $7n$]	$2s$, $3i$, $5n$ [$4s$, $2i$, $4n$]

(sidelobes produced by one frequency contribute spurious strength at other frequencies). Therefore, the use of the full $D^S(\pm l)$ may add (or subtract) signal strength at the lth or other frequency. The appearance of c's (c stands for correction) in Table 3 signifies that some of the excess strength has been recovered on subsequent iteration of the algorithm (other side effects may be errors at other frequencies or noise). The number of successful extractions decreases with decreasing T, decreasing duty cycle, and increasing noise. In Table 3, successful extractions range from all eight (low noise, long T case) to two components (70 data days, a 0.1 duty cycle, noise amplitude of 100, corresponding to the minimum extracted amplitude–noise ratio of 0.2). For 140 and 740 data days, the critical amplitude–noise ratio is 0.1 and 0.07, respectively, when the duty cycle is 0.2; the ratio is somewhat less (higher) for higher (lower) duty cycles. Note that in case of zero noise, the extraction was complete in most cases.

Because of high sidelobes, the most unfavorable situation for extraction is a periodic sampling sequence. In Table 3, results are also listed (in square brackets) for nonperiodic sequences characterized by the nonperiodicity parameter σ_p of about 1 (corresponding to randomly distributed observation sessions). Here, σ_p is the normalized root-mean-square deviation of the separation p_i between the beginning times for subsequent sessions,

$$\sigma_p \equiv \sqrt{\frac{1}{N_d}\sum_i^{N_d}\left(\frac{p_i}{<p>}-1\right)^2} \tag{6}$$

where $<p>$ is the average separation, $<p>=\sum_i^{N_d} p_i/N_d$, where N_d is the total number of (1-day-long) observation sessions. The main effect of a finite (nonzero) σ_p is to somewhat suppress the height of the spectral peaks at the signal and gap intermodulation frequencies. The suppression is most effective for

long sampling sequences, because (for periodically repeating sessions) the sidelobes are biggest there. For example, for 740 data days, a 10-percent duty cycle, and a noise level of 100, the randomization increased (Table 3) the number of extracted frequencies from 3 to 5. The randomization makes the extraction easier to obtain if the sampling sequence is long and random.

In the majority of the simulations, the optimal frequency index was found to coincide with the frequency of the highest spectral peak in the current Fourier map. Occasionally, however, the second or third (or tenth) strongest peak actually produced the smallest quadratic residual function M_l, suggesting that during the search for the optimal frequency, one should test some finite number (N_p) of the biggest spectral peaks. For the sequences tested, we have found that $N_p \leq 50$ was quite sufficient. The fact that only a relatively small number of candidate l's need to be inspected makes the minimization of the M_l rather efficient. For N_p inspected peaks, the CPU time scales as $N_c \times N_p \times N \times \log N$, where N_c is the number of iterations (typically not higher than several tens), and where $N \times \log N$ is the scale factor for CPU time for FFT of data arrays of dimension N. (If all indices l were to be tested, the CPU time would scale as $N_c\ N^2\ \log N$, which for large arrays would be excessive.) For $N_c = 8$, $N_p = 10$, and $N = 10^{18}$, the CPU time was about 1 minute on a VAX 3000/400 and several minutes on a VAX 4000/90.

B. Noninteger Frequency Index

The precision of the extracted frequencies affects the accuracy of the extracted phase and, ultimately, the success of retrieving oscillatory components with lower amplitudes. To improve the extraction accuracy, the total length of the sampling sequence T can be increased (by padding it with zeroes) to a higher power of two. Or, if the array size is limited, a second version of the algorithm can be used that involves nonintegral l's. This noninteger version is especially useful if, in addition to high frequencies, the data contain slowly varying signals (for τ on the order of or longer than T, the resolution τ^2/T resulting from the use of the standard discrete FFT would be on the order of or greater than τ itself). The use of the noninteger algorithm allows for elimination of strong low-frequency components before the extraction of high-frequency components.

If the data include frequencies not equal to an integer multiple of $1/N$, the residual is minimum for a set $\pm(l + \Delta l)$, where $0 \leq \Delta l \leq 1$. The noninteger algorithm searches for the optimal Δl by using a noninteger FFT together with a minimization scheme that does not require construction of derivatives. The main mathematical basis for the algorithm is described in the Appendix. The noninteger algorithm was tested by using the same simulated data as for the integer algorithm. The results were similar in that when one algorithm met with success, so did the other algorithm. The trade-off was a bigger array size for the integer algorithm versus longer CPU time for the noninteger algorithm. To achieve a fractional accuracy of $\delta\tau/\tau$ for an oscillatory signal with the period τ, the array size (i.e., the total length of the sampling sequence T) used by the integer algorithm must be greater than $\tau(\tau/\delta\tau)$. For the noninteger algorithm, the required array size is less; however, the CPU time is increased due to the use of the search and minimization procedure that optimizes Δl. For the eight oscillatory components in Table 2, CPU time for the noninteger algorithm was typically a factor of 10 bigger, while the frequency accuracy increased by one significant place for the same array size in both algorithms.

The principal application of the noninteger algorithm is for spectral analysis of data with both rapid and slow modulations present. To simulate a train of hourly residuals of Earth orientation parameters obtained from reduced VLBI data, the simulated data consisted of the same four near-diurnal and four near-semidiurnal oscillatory frequencies as in Table 2, superimposed on a slowly varying signal that was represented by a sum of 1-month, 1-year, and 18-year periodic components. The signal was imbedded in a background of Gaussian noise on the same order of magnitude as the strongest diurnal component. The extraction success depends on the length and structure of the sampling sequence, noise level, and the number and amplitudes of low-frequency components. A typical sampling sequence consisted of 270 data days (data sampled hourly) separated by 4-day-long gaps (with a corresponding duty cycle of 0.2 and a total sampling sequence length of $T = 3.5$ yr). Table 4 summarizes the simulation parameters for

three cases: In the first case, the amplitudes of the 1-month, 1-year, and 18-year components were 500, 2000, and 0, respectively; in the second case, the amplitudes were 500, 2000, and 200,000, respectively; in the third case, the amplitudes were 500, 20,000, and 500,000, respectively. The noise level was 30, and the eight diurnal and semidiurnal signals were the same as in Table 2 (amplitudes between 3.2 and 23). Figures 2, 3, and 4 show the corresponding original data and the original, residual, and clean Fourier maps. On comparison, the figures illustrate the effect of the number and amplitudes of low-frequency components on the success of extraction of high-frequency components.

Table 4. Extraction results when the data also contained strong low-frequency components (corresponding to Figs. 1, 2, 3, and 4).[a]

Relative ampitudes of low-frequency components		Iterations for extraction of high-frequency components
(18 yr):	0	
(1 yr):	0	1, 2, 3, 4, 5, 6, 7, 8
(1 mo):	0	
(18 yr):	0	
(1 yr):	2,000	4, 5, 6, 7, 8, 9, 10, 11
(1 mo):	500	
(18 yr):	200,000	
(1 yr):	2,000	11, 14, 15, 18, 20, 22, 25, 31
(1 mo):	500	
(18 yr):	500,000	
(1 yr):	20,000	26, 28, 31, 47, 53, 65, 72, 86
(1 mo):	500	

[a] The listed numbers designate iterations that extracted one of the high-frequency components. The high-frequency components are the same as in Table 2, the number of data days was 270 (data sampled hourly), the duty cycle was 0.2, and the noise level was 30.

In Figs. 2–4, iterative application of the algorithm has resulted in successful extractions of the diurnal and semidiurnal frequencies in all cases; however, the extraction process was lengthier (resulting in richer recovered spectral content) when the number and amplitudes of the original low-frequency components were higher. In general, the algorithm finds the strongest (in these cases, the low-frequency) components first. However, since the values of the extracted frequencies and phase are always somewhat in error (the accuracy will eventually be limited by the available numerical accuracy), the effect of errors in the values of the extracted parameters is made up by retrieving, in successive iterations, additional (false) frequencies, until the slow variations are matched sufficiently well for the high-frequency modulations to become the strongest Fourier components in the current residual map. This process of matching the time sequence with a richer Fourier spectrum becomes more intricate with increasing amplitudes and complexity of the original spectrum. In Fig. 2, the original spectrum contains no 18-year component. The high-frequency signals were found immediately after the low (1-year and 1-month) frequency signals, except that (due to a finite length T of the data array) the algorithm also found (as it usually does when variations on a scale comparable to or longer that T are present) a frequency $\simeq 1/T$. In Figs. 3 and 4, the original spectrum includes an 18-year-long component with the amplitude four orders of magnitude higher than

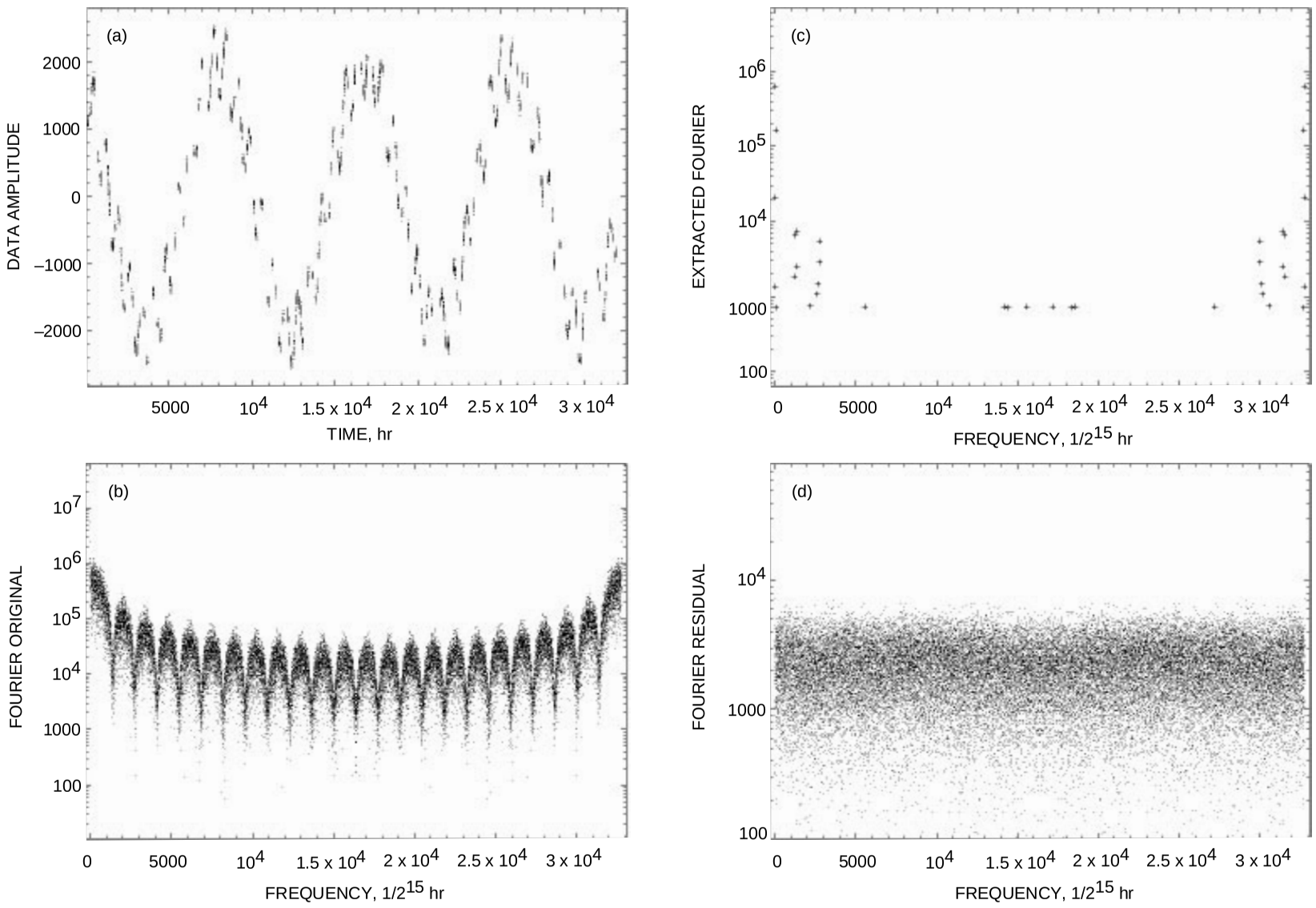

Fig. 2. Results for the same simulated data as in Fig. 1, but with the diurnal and semidiurnal components (Table 2) superimposed on a slowly varying signal with the amplitudes of 500 and 2000 for 1-mo and 1-yr periodic components respectively: (a) the original data, (b) the original Fourier map, (c) the clean Fourier map, (d) the residual Fourier maps.

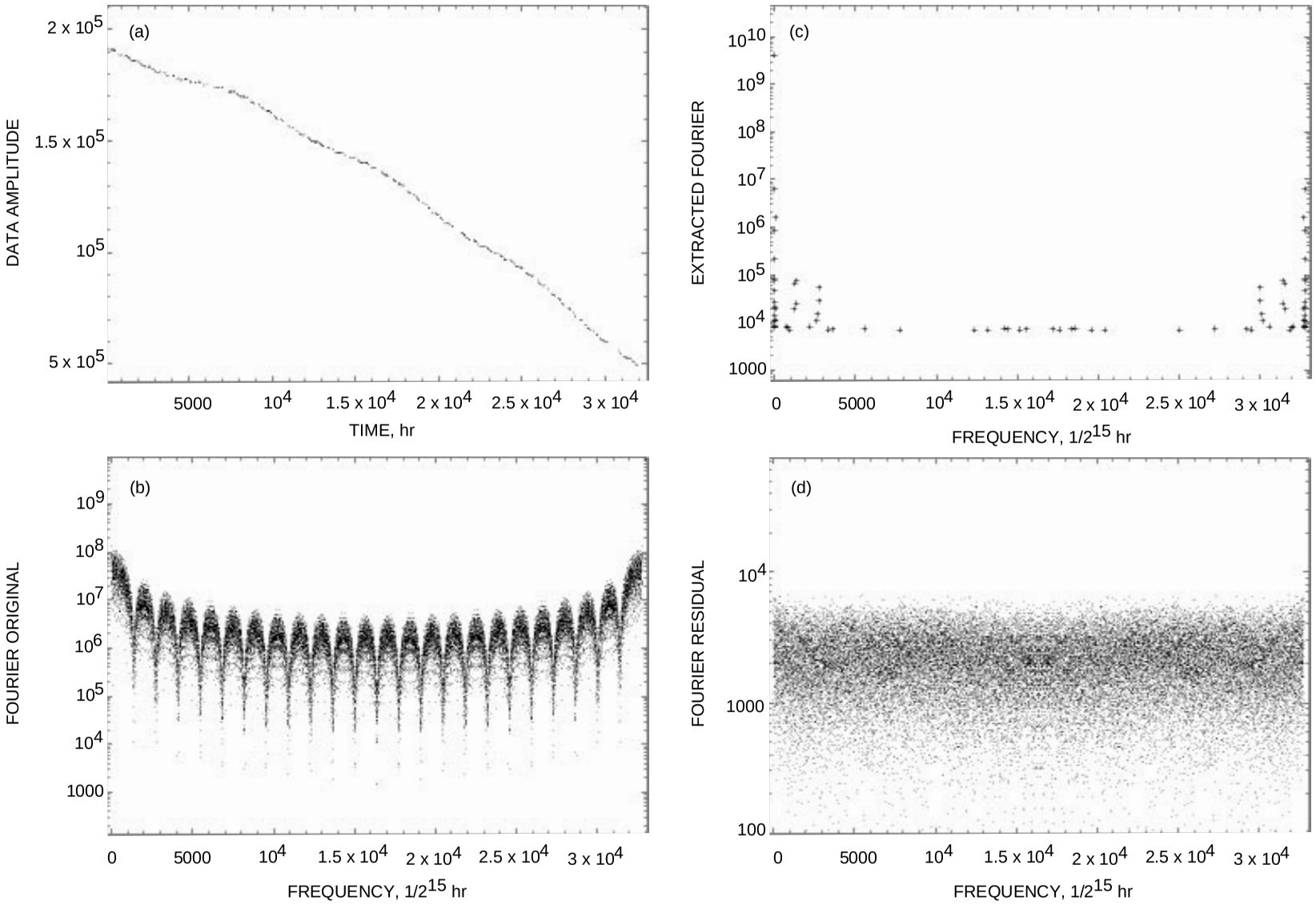

Fig. 3. Results for the same simulated data as in Figs. 1 and 2, but with the diurnal and semidiurnal components (Table 2) superimposed on a slowly varying signal with the amplitudes of 500, 2000, and 200,000 for a 1-mo, 1-yr, and 18-yr periodic components respectively: (a) the original data, (b) the original Fourier map, (c) the clean Fourier map, and (d) the residual Fourier map.

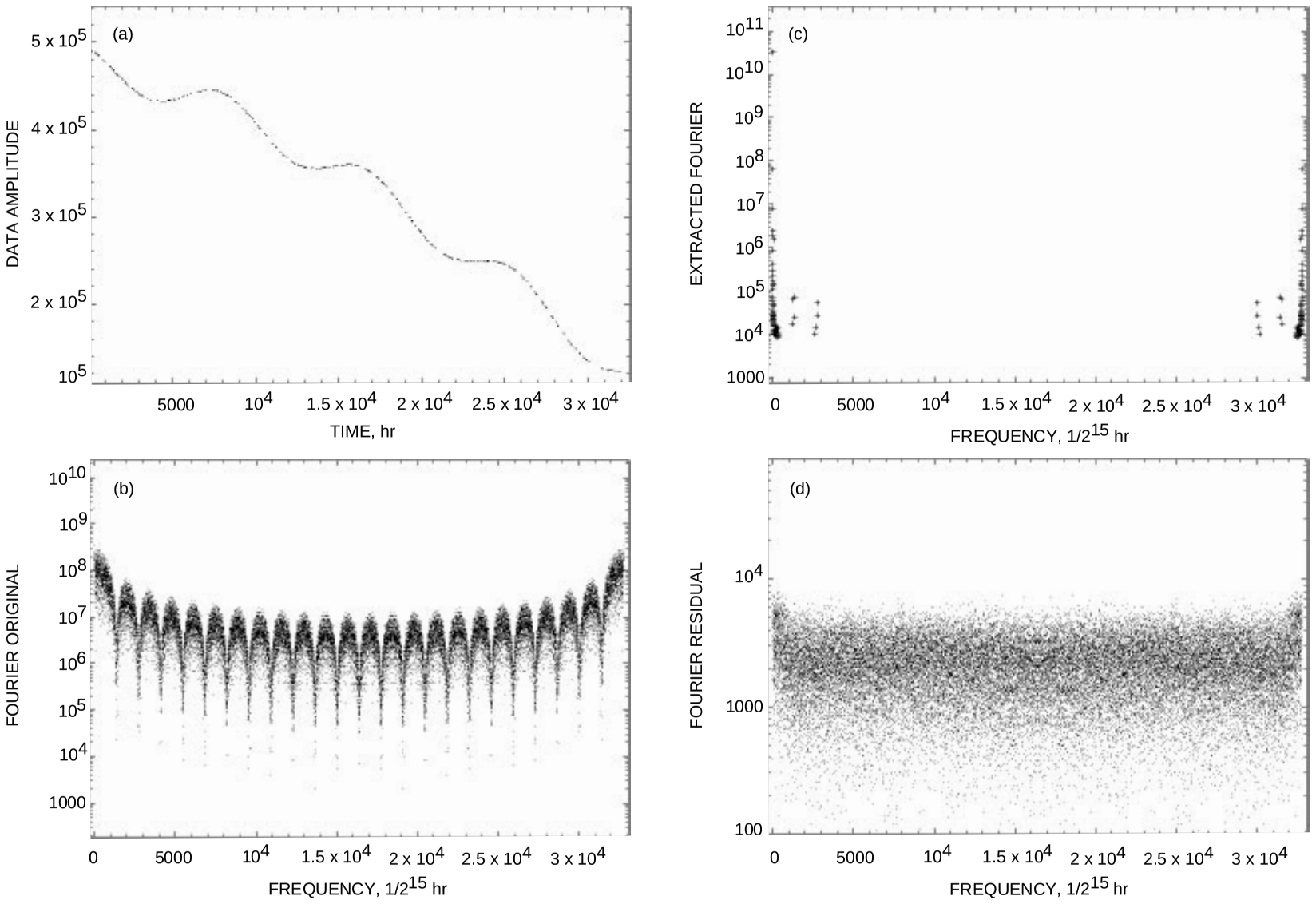

Fig. 4. Results for the same data as in Figs. 1, 2, and 3, but with the diurnal and semidiurnal components (Table 2) superimposed on a slowly varying signal with amplitudes of 500, 20,000, and 500,000 for 1-mo, 1-yr, and 18-yr periodic components, respectively: (a) the original data, (b) the original Fourier map, (c) the clean Fourier map, and (d) the residual Fourier map.

the strongest diurnal component. The algorithm also fits the slow variations first. However, because the variations are not fit exactly (the 18-year period was found with 2- and 3-percent accuracy in Figs. 3 and 4, respectively, and the 1-year and 1-month periods were found with 0.7- and 0.1-percent accuracy, respectively), the algorithm continues to search for additional frequencies until the fit is sufficiently tight. In Figs. 3 and 4, the high frequencies were extracted at the 11th, 14th, 15th, 18th, 20th, 22nd, 25th, and 31st iterations, and at the 26th, 28th, 31st, 47th, 53rd, 65th, 72nd, and 86th iterations, respectively. Thus, the complexity of the low-frequency content will play a significant role in establishing limits on the algorithm applicability.

IV. Summary and Conclusions

A new algorithm has been developed for spectral analysis of sparse, irregularly sampled data with low signal-to-noise ratios; the extracted parameters include frequency, phase, and amplitude of the signal components. The algorithm identifies the strongest component as that component which when passed through the same sampling sequence as the original data produces a Fourier image that is best matched to the current residual map. The algorithm has met with success in trials with simulated data, including those of a type similar to hourly residuals for Earth orientation parameters. The simulated data consisted of four near-diurnal and four semidiurnal oscillatory components with amplitudes varying by an order of magnitude. The background noise was on the same order of magnitude as the strongest high-frequency component. The number of data days ranged from 70 to 740 (data sampled hourly), and the duty cycle for the observation sessions varied between 1 and 0.1. Two versions of the algorithm were developed. In the integer version, the frequency solution is limited to $1/T$, where T is the sampling sequence length. With only the high-frequency components present in the data, the integer version was successful in extracting all components with an amplitude–noise ratio greater than about 0.2; the extraction success was higher (less) for longer (shorter) data sequences and higher (lower) duty cycles. Sessions that are not periodic make the extraction easier to achieve since the sidelobes (produced by the interference with the average periodicity of the sessions) are less than for the periodic case. Sidelobes associated with one frequency can contribute signal at other frequencies. If the residual map has been constructed by completely subtracting the identified spectral peak, the subtracted sidelobes can produce errors in this and other estimates. The solution to this problem is to subtract the peak in fractions. That is, at multiple applications of the algorithm, one subtracts from the current residual map only a fraction of the image that has been constructed by using the peak full strength. The effect of subtracted fractions on the extraction results could be used to achieve confidence in the found solutions. Another test of confidence can be obtained by modifying the observation strategy, e.g., by changing the duty cycle, the length of the sampling sequence, and the gap randomness. The frequency resolution can be increased by padding the data array with zeroes to a higher power of two (higher T).

For signal frequencies that are not integer multiples of the minimum discrete FFT frequency, the residual map will be minimized by a noninteger frequency value. We have implemented (and tested) an algorithm version that searches for a noninteger frequency index. The resolution is greatly improved over $1/T$ without requiring an increase in the array length. However, the CPU time is increased. In simulations with diurnal and semidiurnal bands, the typical CPU time was longer by a factor of 10, while the extracted frequency accuracy increased by one significant place relative to results obtained with the integer algorithm. Many real data also contain (in addition to the high frequencies) very strong low-frequency components. Unless some other scheme is used to eliminate the slow variations first, when applied, the algorithm fits the slowly varying signal with low frequencies before it searches for the high frequencies. The increase in the extraction accuracy comes at the expense of increased CPU time. The CPU time is increased depending on the signal frequencies, strength, and number of low-frequency components. Several types of simulated data with strong 18-year, 1-year, and 1-month periodic components were analyzed in detail in Section III.B (Figs. 2, 3, and 4). The accuracy of the extracted frequencies influences the efficiency of retrieval of components with lower amplitudes.

For uniformly spaced time markers, the CPU time scales as a constant $\times N_c \times N_p \times N \times \log N$, where N_c is a number of iterations of the algorithm (to find multiple oscillatory components), N_p is a number of candidate spectral peaks that must be inspected to minimize the residual sum, and N is the array total size chosen as a power of 2. For 2^{18} time markers in the sampling sequence and eight cleaning steps, typical CPU time for the integer algorithm was on the order of several minutes on a VAX-class machine. For the noninteger algorithm, the CPU time can exceed several hours, depending on the complexity of the overall frequency content.

The algorithm is currently applied to actual hourly Earth orientation parameters for the available International Radio Interferometric Surveying (IRIS), Crustal Dynamics Project (CDP), and DSN VLBI data.

Acknowledgments

The author is grateful to O. Sovers and C. Jacobs for encouragement and to R. Gross for critical reading of the manuscript.

References

[1] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in Fortran*, 2nd ed., Cambridge: Cambridge University Press, Ch. 13, pp. 530–602, 1992, and references therein.

[2] M. Lazrek and F. Hill, "Temporal Window Effects and Their Deconvolution From Solar Oscillation Spectra," *Astron. Astrophys.*, vol. 280, pp. 704–714, 1993.

[3] J. A. Hogbom, "Aperture Synthesis With a Non-Regular Distribution of Interferometer Baselines," *Astron. and Astrophys. Supplement*, vol. 15, pp. 417–426, 1974.

Appendix

Spectral Analysis for a Noninteger Frequency Index

The method outlined in Section II is frequently sufficient provided that the periodic components are sufficiently few in number, well separated in frequency, not too disparate in the relative magnitude of their amplitudes, and not too corrupted by the presence of noise. If stressing conditions are present, the method can be enhanced by introducing fractional frequencies. The Fourier map becomes a function of the fractional value Δl. An additional search is performed on Δl to identify the residual minimum by using some standard function minimum-finding algorithm.

Similarly to Eq. (1) of the main text, define the minimal function

$$M_{l+\Delta l} \equiv \sum_{k}^{N} |D^S(k) - \sum_{\pm} a_{\pm(l+\Delta l)}\ f^S_{\pm(l+\Delta l)}(k)|^2 \tag{A-1}$$

where all terms have the same meaning as in Eq. (1) except that the second term on the right-hand side of Eq. (A-1) represents a signal at the candidate frequency $\omega_{l+\Delta l} = 2\pi(l+\Delta l)/(N\delta t)$, and $f^S_{l+\Delta l}(k)$ is the Fourier transform of $f_{l+\Delta l}(t_n)\ S(t_n)$, where the periodic function $f_{l+\Delta l}(t_n) \equiv \exp(-2\pi\ i(n(l+\Delta l)/N))\ S(t_n)$.

As in the main text, the optimal value of $l+\Delta l$ is found as that frequency which when filtered through the same sampling sequence $S(t_n)$ as the original data provides the best match to the original Fourier map. The algorithm computes the Fourier map, $D^S(k)$, selects a value of Δl, $0 \le \Delta l \le 1$, and, for the selected Δl, computes the functions $f^S_{l+\Delta l}(k)$. To evaluate $M_{l+\Delta l}$, the values of the amplitudes $a_{l+\Delta l}$ are determined from the following least-square minimization principle:

$$\frac{\partial M_l}{\partial a^*_m} = \sum_{k}^{N} \left[D^S(k) - \sum_{\pm} a_{\pm(l+\Delta l)} f^S_{\pm(l+\Delta l)}(k) \right]\ f^{S\,*}_m(k) = 0 \tag{A-2}$$

For $m = \pm(l+\Delta l)$, Eq. (A-2) defines a linear system of two equations from which the optimal magnitudes of the unknown $a_{\pm(l+\Delta l)}$'s are determined. By using these optimal $a_{\pm(l+\Delta l)}$'s in Eq. (A-1), the algorithm searches for an optimal l that minimizes M_l. (Similar to that in the main text, this minimization is most efficiently achieved by inspecting several of the highest peaks in the original Fourier map.) The above-described procedure yields an optimal l for any given Δl. The algorithm performs a systematic search for a Δl that minimizes $M_{l+\Delta l}$ as a function of Δl by using Brent's method (it does not require construction of derivatives; see [1]).

To compute the various functions required in the evaluation of Eqs. (A-1) and (A-2), the following relationships are used to minimize the number of the computations. Let F denote the discrete Fourier transform operator. Then the Fourier transform of $f_{l+\Delta l}(t_n)S(t_n)$ is

$$f^S_{l+\Delta l}(k) \equiv F_k \left[e^{-2\pi i \frac{n(l+\Delta l)}{N}} S(t_n) \right] = \sum_{n}^{N} e^{2\pi i \frac{n(k-l)}{N}} \left[S(t_n) e^{-2\pi i \frac{n\Delta l}{N}} \right]$$

$$= F_{k-l} \left[S(t_n)\ e^{\frac{-2\pi i n \Delta l}{N}} \right] \equiv S(k-l-\Delta l) \tag{A-3}$$

That is to say, the kth Fourier component of $f_{l+\Delta l}(t_n)S(t_n)$ is the $(k-l)$th Fourier component of $S(t_n)\exp^{-2\pi i(n\Delta l/N)}$.

By using relationships similar to those used to derive Eq. (A-3), Eq. (A-2) is simplified:

$$\sum_{k}^{N} f^{S}_{l+\Delta l}(k) f^{S^*}_{m}(k) \;=\; \sum_{k,n,n'}^{N} S(t_n)\; e^{2\pi i \frac{n(k-l)}{N}} S(t'_n)\; e^{-2\pi i \frac{n'(k-m)}{N}}\; e^{-2\pi i \frac{n\ \Delta l}{N}}$$

$$= \sum_{n,n'}^{N} \delta_{n,n'} S^2(t_n)\; e^{-2\pi i \frac{n\ \Delta l}{N}} = N\ S(m-l-\Delta l) \tag{A-4}$$

and, similarly,

$$\sum_{k}^{N} D^S(k) f^{S^*}_{m}(k) = N\ D^S(m) \tag{A-5}$$

By using Eqs. (A-4) and (A-5), Eq. (A-2) simplifies to

$$\sum_{\pm} S(m \mp (l+\Delta l))\ a_{\pm(l+\Delta l)} = D^S(m) \tag{A-6}$$

For $m = \pm(l+\Delta l)$, Eq. (A-6) is solved for a pair of coefficients $a_{\pm(l+\Delta l)}$. Note that for $\Delta l = 0$, Eq. (A-6) reduces to Eqs. (4a) and (4b) of the main text.

Other useful relationships involve the evaluation of the noninteger Fourier components of the sampling function $S(t_n)$:

$$S(k+\Delta l) = F_k\ \left[S(t_n)\; e^{2\pi i \frac{n\Delta l}{N}}\right] \tag{A-7}$$

$$S(2(l+\Delta l)) = F_{2l}\ \left[S(t_n)\; e^{4\pi i \frac{n\Delta l}{N}}\right] \tag{A-8}$$

$$D^S(l+\Delta l) = F_l\left[F_n^{-1}[D^S(k)]\; e^{-2\pi i \frac{n\Delta l}{N}}\right] \tag{A-9}$$

The use of Eqs. (A-3)–(A-5) and (A-7)–(A-9) minimizes the number of computations required to recompute Eqs. (A-1) and (A-6) at each iteration step to minimize M_l.