Digital Signal Processing in the Radio Science Stability Analyzer

C. A. Greenhall
Communications Systems Research Section

The Telecommunications Division has built a stability analyzer for testing Deep Space Network installations during flight radio science experiments. The low-frequency part of the analyzer operates by digitizing sine wave signals with bandwidths between 80 Hz and 45 kHz. Processed outputs include spectra of signal, phase, amplitude, and differential phase; time series of the same quantities; and Allan deviation of phase and differential phase. This article documents the digital signal-processing methods programmed into the analyzer.

I. Introduction

The recently developed radio science stability analyzer (RSA) is an instrument for real-time testing and certification of Deep Space Network (DSN) equipment to be used during gravity wave and planetary occultation experiments [1]. Two sets of equipment can be tested: (1) the radio science open-loop receiver and (2) the 100-MHz frequency standards and distribution network of the DSN frequency and timing system (FTS). Signals from either of these two sources are downconverted to low-frequency band-limited sine wave signals. The last stage of the open-loop receiver, called radio science intermediate frequency to video (RIV), produces sine wave signals with frequencies and bandwidths ranging from 150 Hz in an 82-Hz band to 275 kHz in a 45-kHz band; these depend on the choice of RIV filter. RIV signals are processed directly by the low-frequency RSA circuitry. Pairs of 100-MHz FTS signals are processed in a portion of the RSA called the 100-MHz interface assembly (100 MHz IA), which resides near the frequency standards. The 100 MHz IA mixes the two signals at 10 GHz and downconverts the mixer output to a 100-kHz sine wave signal in a 30-kHz bandwidth, which is sent over a fiber-optic cable to the low-frequency RSA circuitry.

The low-frequency circuitry has two methods for converting a band-limited sine wave signal to digital information. First, the signal can be sampled with a 16-bit analog-to-digital (A–D) converter clocked by a synthesizer. In this mode, two signal channels can be accommodated with the aim of extracting their differential phase. The maximum total data rate is about 230 kilosamples per second. Second, if the carrier frequency is known within approximately 0.1 Hz, it can be mixed with the output of another synthesizer set to this frequency minus 1 Hz. The 1-Hz mixer output is filtered and hard limited by a zero-crossing detector, and the up-crossing times of the resulting sequence of pulses are captured by a time-interval counter according to the “picket fence” method [4].

The principal aim of processing the A–D data is to reduce their bandwidth by a user-selected factor, and to extract the amplitude and phase modulations that constitute the sidebands of the sine wave signal.

The phase of two channels can be combined into differential phase. Three output types can be generated: spectrum of the signal and its modulations, time series of the modulations, and Allan deviation of phase. As described below, the digital signal processing operates in three alternate modes, called full band, medium band, and narrow band. The choice among these depends on the desired bandwidth reduction factor. The 1-Hz zero-crossing data are processed in the same way as sequences of phase residuals produced by narrow-band processing.

The digital signal processing (DSP) methods are designed to take advantage of the architecture of a floating-point vector processor based on the 40-MHz Intel I860. Most of the heavy lifting is done by manufacturer-supplied vector library routines, which include fast Fourier transform (FFT) and finite impulse response (FIR) filtering routines. Throughputs of approximately 25 million floating-point operations per second were achieved.

The remainder of this article explains the DSP methods in some detail.

II. Signal Properties

A. Radio Frequencies

In any test setup, there are two radio frequencies of interest. Let $f_{\rm mix}$ be the frequency at which the primary comparative mixing takes place, and let $f_{\rm ref}$ be the reference frequency for phase noise and Allan deviation. For a RIV test, $f_{\rm mix} = f_{\rm ref} = 2295$ MHz (S-band) or 8415 MHz (X-band). For an FTS test, $f_{\rm mix} = 9.9$ GHz, $f_{\rm ref} = 100$ MHz. This is because the phase of the 100-kHz output of the 100 MHz IA is approximately 99 times the difference between the phases of the two 100-MHz inputs. Phase results are scaled by $f_{\rm ref}/f_{\rm mix}$.

B. Analog Sine Wave Signal

The downconverted signal is assumed to lie in an analog frequency band with the center at $f_{\rm ofst}$ and width $W_{\rm vid} < f_{\rm ofst}$, which are parameters of the RIV filter or the 100-MHz IA. The frequency $f_{\rm ofst}$ can be positive or negative; see the discussion of polarity below. Somewhere in this band is the carrier. Except in full-band processing, it is assumed that the signal consists of a carrier with weak sidebands; the total carrier-to-noise ratio should be at least about 30 dB. (This instrument is a stability analyzer, not a receiver.)

C. Digitized Sine Wave Signal

The analog signal is sampled by a 16-bit A–D converter at the sample rate f_s, which has to be chosen so that the analog frequency band is aliased into the Nyquist band $(0, f_s/2)$ or $(-f_s/2, 0)$. In this way, both sidebands of the carrier are preserved. Each RIV filter is designed for a certain f_s. In any case, an acceptable f_s can be obtained from the formulas

$$m = {\rm int}\left(\frac{|f_{\rm ofst}|}{W_{\rm vid}} - 0.5\right), \qquad f_s = \frac{4\,|f_{\rm ofst}|}{2m+1}$$

where int (x) is the integer part of x. This choice of f_s centers the aliased signal band in the Nyquist band. If the actual carrier frequency is close to $f_{\rm ofst}$, however, then distortion in the analog signal or A–D converter may cause spurious harmonics to appear near the carrier. To push the images of the lowest harmonics away from the carrier, one can offset the sample rate slightly, according to the formulas

$$a = 0.944272, \qquad m = {\rm int}\left(a\left(\frac{|f_{\rm ofst}|}{W_{\rm vid}} - 0.5\right)\right), \qquad f_s = \frac{4\,|f_{\rm ofst}|}{2m+a}$$

The number a is related to the golden ratio $\left(\sqrt{5}-1\right)/2$.

D. Polarity

In the radio science receiver, the 2.3-GHz or 8.4-GHz signal is downconverted and filtered three times until the carrier is at 10 MHz $+f_{\text{ofst}}$, where f_{ofst} can be positive or negative. At this point, the spectrum or phase polarity of the signal is positive, i.e., the same as the radio frequency (RF) signal. The fourth downconversion by the 10-MHz local oscillator and subsequent filtering, therefore, yield a signal whose polarity equals the sign of f_{ofst}. Moreover, the sampling can flip the polarity again. To make better sense of this, it is good to think about the two-sided representation of the signal. One side of the signal has the right polarity (positive), and the other side has the wrong polarity. If we let

$$n_{\text{base}} = \text{nint}\left(\frac{f_{\text{ofst}}}{f_s}\right), \qquad s_{\text{pol}} = \text{sign}\left(f_{\text{ofst}} - n_{\text{base}} f_s\right)$$

where nint (x) is the nearest integer to x, then s_{pol} is the polarity of the digitized signal, the side of the analog signal with the right polarity lies between $n_{\text{base}} f_s$ and $\left(n_{\text{base}} + s_{\text{pol}}/2\right) f_s$, and the side of the digitized signal with the right polarity lies between 0 and $s_{\text{pol}} f_s/2$. The user has the responsibility of entering f_{ofst} with the correct sign.

III. Full-Band Processing

This mode allows the user to see a snapshot of the signal in the time and frequency domains before proceeding to a closer view. The user selects an FFT size N (2048 or 4096). A frame of A–D data $x[0], \cdots, x[N-1]$ is collected. These can be plotted against elapsed time in the frame, after scaling them back to volts at the A–D input (10 V = 32,768). A spectral estimate of the frame is computed by scaling the frame so that $\sum x[n]^2 = 1$ and calculating

$$S_x[k] = \frac{2}{f_s N} \left| \sum_{n=0}^{N-1} x[n] u_0[n; N, 5] \exp\left(-i 2\pi n k/N\right) \right|^2, \qquad k = 0, \cdots, N/2 \tag{1}$$

where $u_0[n; N, 5]$ is the 0th-order, N-point "trig prolate" data taper [5] with bandwidth parameter $w = 5$ (Appendix B), scaled so that $\sum u_0[n]^2 = N$. The sidelobes of this taper (Ω_{05} in Fig. 1) are low enough so that no leakage from the carrier should be visible in the sidebands. The array $10 \log_{10} S_x[k]$ (labeled dBc/Hz) is plotted against the frequency array

$$f[k] = f_s\left(n_{\text{base}} + s_{\text{pol}} k/N\right), \qquad k = 0, \cdots, N/2$$

which shows the side of the signal with the correct polarity. The user chooses how many of these frame spectra are averaged into a run spectrum. The frames do not have to be adjacent; it is all right to lose data while processing the previous frame.

The resolution bandwidth of the spectral estimate, given by

$$W_{\text{nb}} = \frac{f_s N}{\left(\sum u_0[n]\right)^2}$$

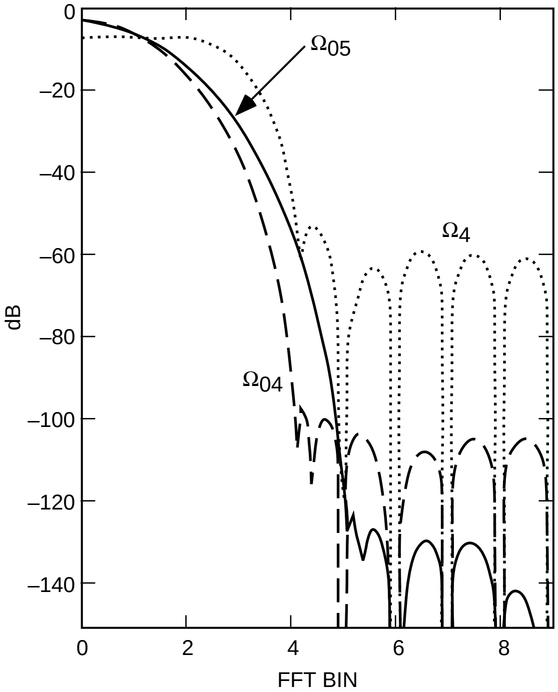

Fig. 1. Spectral windows: full band Ω_{05}, medium band Ω_{04}, and narrow band Ω_{4}.

has two purposes: (1) It gives the user a rough idea of the resolution of the spectral plot, and (2) it allows the user to estimate the power of a bright line (narrower than $W_{\rm nb}$) in dBc by adding $10\log_{10}W_{\rm nb}$ to the dBc/Hz reading at the peak of the line.

Because the main purpose of this function is a check on what sort of signal is actually in the Nyquist band, it might be preferable to scale the spectrum to dBm/Hz or dBV^2/Hz instead of scaling the frame to power 1 and claiming that we are seeing dBc/Hz. Then, for example, if no signal were present, the display would show the correct spectral density level of the noise.

IV. Medium-Band Processing

In this mode of processing, we assume that the sampled signal consists of a carrier with weak sidebands. The purpose of the processing is to reduce the bandwidth of the signal by a modest amount (up to 128 with current parameters), remove the carrier, and measure properties of the sidebands.

A. z-Frame Production

The user having selected an FFT size $N_{\rm fft}$ and a decimation factor r, both powers of 2, define the frame size $N_{\rm xf} = rN_{\rm fft}$. In order to limit memory usage, the frame is divided into $n_{\rm bf}$ adjacent batches of size $N_{\rm xb}$, a divisor of $N_{\rm xf}$ that is not more than some maximum batch size (currently 8192). One batch at a time is processed. We use the first batch to measure the carrier frequency by a simple vector computation called "Pony, Part 1" (Appendix A). Let $\hat{o}$ be the measured frequency in radians per sample, the sign of $\hat{o}$ being $s_{\rm pol}$, and let $u = \exp(-i\hat{o})$. Let $x[n],\ n = 0, \cdots, N_{\rm xf} - 1$, be the A–D x-frame. A complex z-frame $z[m]$ of size $N_{\rm zf} < N_{\rm fft}$ is computed by

$$z_1[n] = x[n]u^{-n}, \qquad n = 0, \cdots, N_{\rm xf} - 1 \tag{2}$$

$$z[n] = \sum_{k=0}^{n_h-1} h_r[k] z_1[rn+k], \qquad n = 0, \cdots, N_{\text{zf}} - 1 \tag{3}$$

where h_r is a lowpass FIR filter designed for decimation by r (Appendix B). Its length n_h is assumed to be a multiple of r (currently $16r$), and it follows that we can take $N_{\text{zf}} = N_{\text{fft}} - n_h/r + 1$. The ripples of the frequency response of h_r above the decimated Nyquist frequency (Fig. 2) are low enough so that the aliased image of the wrong side of the carrier at $-\hat{o}$ barely appears above the 16-bit quantization noise in a spectrum output with simulated data.

The computation in Eqs. (2) and (3) is carried out batch by batch, the z-frame being built up in n_{bf} steps by an overlap-add operation. The result is a complex representation of the carrier (at zero frequency now) and sidebands within $f_s/(2r)$ of the carrier. Because frames are processed independently, it is all right to lose A–D data between frames while carrying out further processing on completed z-frames.

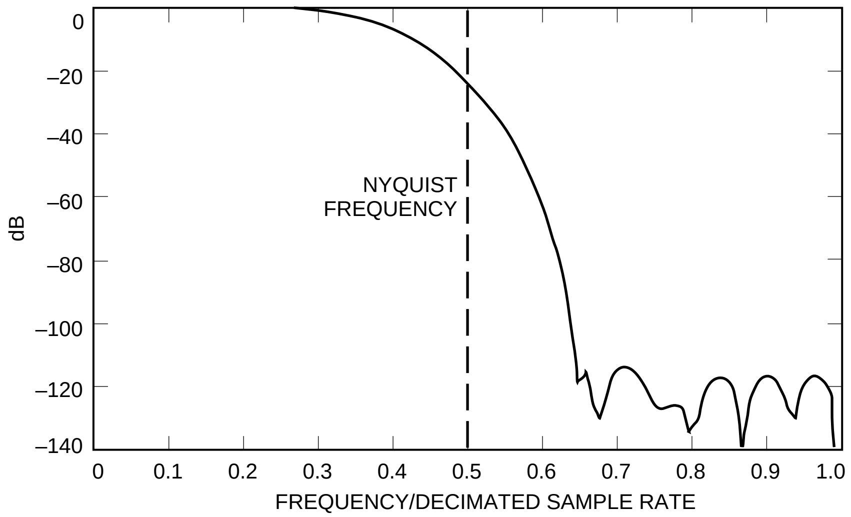

Fig. 2. Frequency response of the FIR filter for lowpass decimation.

B. Signal Spectrum

The signal spectrum is obtained as a two-sided spectrum of the z-frame. First, the z-frame is scaled to unit energy. Most of the energy is in the carrier, which is now at dc (zero frequency). To prevent this dc energy from leaking into the rest of the spectrum, we get rid of most of it by removing a linear fit from the frame. We call this kind of preconditioning operation a calibration. The specific example used here can be defined on a general array $y[0], \cdots, y[N-1]$ as follows: Let M be an integer approximately equal to $N/6$. Compute the centroid points

$$(t_0, y_0) = \frac{1}{M} \sum_{n=0}^{M-1} (n, y[n]), \qquad (t_1, y_1) = \frac{1}{M} \sum_{n=N-M}^{N-1} (n, y[n])$$

and pass a straight line $c_0 + c_1 n$ through them. The calibrated array is given by $y_0[n] = y[n] - c_0 - c_1 n$. If y itself is a straight line, then $y_0 = 0$.

The choice of this particular operation (especially the $N/6$) for spectral preconditioning is admittedly seat-of-the-pants engineering. Perhaps removing a conventional least-squares fit would do as well. To deal with time series modeled by processes that are possibly nonstationary but do have stationary first or second increments, it is desirable to subtract *some* linear fit, not just the mean. This makes all frames statistically identical, so that the average of the spectral estimates of J disjoint frames converges as $J \to \infty$, just as in the theory of stationary-process spectral estimates.

The spectrum and frequency arrays are now given in terms of the calibrated array z_0 by

$$\begin{aligned} S_z[k] &= \frac{r}{f_s N_{\text{zf}} \left|H_r(f[k])\right|^2} \left| \sum_{n=0}^{N_{\text{zf}}-1} z_0[n] u_0[n; N_{\text{zf}}, 4] \exp\left(\frac{-i2\pi nk}{N_{\text{fft}}}\right) \right|^2 \\ f[k] &= \frac{f_s}{r N_{\text{fft}}} k \end{aligned}$$

where $k = -N_{\text{fft}}/2 + 1, \cdots, N_{\text{fft}}/2$. The squared magnitude of

$$H_r(f) = \sum_{n=0}^{n_h - 1} h_r[n] \exp\left(\frac{-i2\pi nf}{f_s}\right)$$

is used for equalizing the spectrum against the lowpass decimation filter. As before, a plot of $10 \log_{10} S_z[k]$ is labeled dBc/Hz. Points corresponding to frequencies with absolute value below $4f_s/(N_{\text{zf}} r)$ or above 95 percent of the Nyquist frequency $0.5 f_s/r$ are not displayed. The low cutoff hides doubtful values near dc; the high cutoff hides a 3-dB rise at the Nyquist frequency caused by the combination of lowpass decimation, noise folding at the Nyquist frequency, and equalization. The user chooses how many of these frame spectra are averaged into a run spectrum. The resolution bandwidth is given by

$$W_{\text{nb}} = \frac{f_s N_{\text{zf}}}{r \left(\sum u_0[n]\right)^2} \tag{4}$$

C. Amplitude and Phase

Extraction of amplitude and phase residuals starts with a rectangular-to-polar operation on the z-frame. The result is a complex "amplitude-phase" frame ap$[n]$, $n = 0, \cdots, N_{\text{zf}} - 1$, whose real part is the amplitude of $z[n]$ and whose imaginary part $\theta[n]$ is the phase of $z[n]$ wrapped into $[-\pi, \pi]$. The amplitudes are replaced by their fractional deviations from the mean. The phases are unwrapped into phase deviations $\phi[n]$ (replacing $\theta[n]$ in the ap array) by the following algorithm:

$$\phi[0] = 0, \qquad \phi[n] = \phi[n-1] + \text{ mods } (\theta[n] - \theta[n-1], 2\pi), \qquad n = 1, \cdots, N_{\text{zf}} - 1$$

The symmetric residue function mods is defined by

$$\text{mods}(x, a) = x - a \text{ nint } (x/a) \tag{5}$$

The correctness of this algorithm requires only that $|\Delta\phi[n]| < \pi$. The mods function also plays the central role in the unwrapping algorithm described in Appendix C.

The amplitude or phase residuals can be displayed as time series for the frame. Often, the phase residuals are dominated by a ramp (a frequency offset), so that it is desirable to subtract a linear fit to reveal the random fluctuations. This can be done with the calibration operation described above in connection with spectral preconditioning.

For better or worse, amplitude and phase spectra are computed together by a single complex FFT instead of two real FFTs. The real and imaginary parts of the ap$[n]$ array are calibrated as above, tapered by $u_0[n; N_{\rm zf}, 4]$, and zero-padded to $N_{\rm fft}$ elements. Let AP$[k]$, $k = 0, \cdots, N_{\rm fft} - 1$ be the complex Fourier transform of the resulting array. The transforms of the real amplitude and phase frames are given by

$$A[k] = \frac{1}{2} \left(\mathrm{AP}[k] + \ \mathrm{AP}[N_{\rm fft} - k]^* \right), \qquad \Phi[k] = \frac{1}{2i} \left(\mathrm{AP}[k] - \ \mathrm{AP}[N_{\rm fft} - k]^* \right)$$

for $k = 0, \cdots, N_{\rm fft}/2$, where AP$[N_{\rm fft}]$ is defined to be AP$[0]$. The one-sided amplitude and phase spectra for the frame are given by

$$S_a[k] = \frac{r}{f_s N_{\rm zf} \left| H_r(f[k]) \right|^2} \left| A[k] \right|^2, \qquad S_\phi[k] = \frac{r}{f_s N_{\rm zf} \left| H_r(f[k]) \right|^2} \left| \Phi[k] \right|^2 \tag{6}$$

with frequency array $f[k] = (f_s/(rN_{\rm fft}))k$. We apply the same low- and high-frequency cutoffs as we did with the medium-band signal spectrum. The absence of a factor of 2 in the scaling factor of Eq. (6) [see Eq. (1)] gives a single-sideband presentation of the spectra, so that they can be labeled dBc/Hz when converted to dB. If a factor of 2 were present in the numerators, the unit for S_ϕ would have to be rad^2/Hz. As before, a number of frame spectra can be averaged into a run spectrum. The resolution bandwidth is given by Eq. (4).

V. Narrow-Band Processing

This processing mode also assumes that the signal consists of a carrier with weak sidebands. Its purpose is to achieve an arbitrarily large reduction in data rate, limited only by the user's patience. The stream of A–D data is reduced to a sequence of average amplitude and phase residuals, the averaging time being chosen by the user. The phase residuals from two channels can be combined into a differential phase. These streams of band-reduced data can be processed into time series, spectra, or Allan deviations (phase or differential phase only).

A. Amplitude and Phase Extraction

The stream of A–D data is divided into batches of size $N_{\rm xb}$, which must be adjacent for the entire run. There is a minimum and maximum batch size (now 200 and 8192). A frame consists of $n_{\rm bf}$ batches, or $N_{\rm xf} = n_{\rm bf} N_{\rm xb}$ A–D data, where $n_{\rm bf}$ can be any positive integer. Each batch is reduced to one sample of average amplitude and phase, and $n_{\rm bf}$ batch samples are averaged to produce a frame sample. The user has to choose $N_{\rm xb}$ and $n_{\rm bf}$ (with the bounds on $N_{\rm xb}$ enforced by the user interface) to achieve the desired reduced sample rate $f_s/N_{\rm xf}$. Unless there are phase tracking problems (see below), the results for a fixed frame size should depend little on the number of batches per frame.

Let us represent the digitized signal by

$$x(t) = A(t) \cos \Phi(t)$$

where $\Phi(t)$ is the total phase, which one can think of as $\omega t + \theta + \phi(t)$, where $\phi(t)$ is a phase residual. The point is that $\Phi(t)$ is an intrinsic part of the signal (except for an unknowable additive constant

$2\pi n_0$), while ω and $\phi(t)$ trade off with each other. We assume that $A(t)$ and $\Phi(t)$ satisfy two imprecisely given conditions, called here the assumptions of small local variations: (1) Over a batch, the fractional variations of $A(t)$ from its mean are much less than 1, and (2) over at least two batches, the total phase differs by much less than one radian from a first-degree polynomial fit (constant phase offset plus constant frequency). Over longer time spans, the phase might deviate from a straight-line fit by many radians.

Let the batches of a run be indexed by k, $k = 0, 1, \cdots$. Batch k starts at time $t_k = kN_{\mathrm{xb}}/f_s$. For the moment, let t run over the sequence of times $t_k + n/f_s, n = 0, \cdots, N_{\mathrm{xb}} - 1$, in batch k. The Pony computation (Parts 1 and 2) of Appendix A is used to estimate the local frequency, amplitude, and phase of the batch. It gives $\hat{o}_k$ (radians per cycle), $\hat{A}_k$, and $\hat{\theta}_k$ such that

$$x(t) \approx \hat{A}_k \cos\left(\hat{o}_k f_s (t - t_k) + \hat{\theta}_k\right) \tag{7}$$

(The sign of $\hat{o}_k$ is taken to be the same as the polarity s_{pol}.) Write $\hat{\omega}_k = \hat{o}_k f_s$. With the assumptions of small local variations, it turns out that, to first order in these variations,

$$\hat{A}_k \approx \bar{A}_k \tag{8}$$

$$\psi_k := \hat{\omega}_k(\bar{t}_k - t_k) + \hat{\theta}_k \approx \bar{\Phi}_k \pmod{2\pi} \tag{9}$$

where $\bar{t}_k$, $\bar{A}_k$, and $\bar{\Phi}_k$ are the averages of t, $A(t)$, and $\Phi(t)$ over batch k. It is important to note that the approximation [Eq. (9)] of ψ_k to $\bar{\Phi}_k \pmod{2\pi}$ is better than the approximation of the phase on the right side of Eq. (7) to $\Phi(t)$ because the errors in $\hat{o}_k$ and $\hat{\theta}_k$ tend to compensate each other in just the right way.

The average amplitude residual for batch k is computed by $a_k = \hat{A}_k/\hat{A}_0 - 1$. The computation of phase residuals is more delicate. According to Eq. (9), ψ_k, to first order, is the average total phase of the signal in batch k, modulo 2π. There are two problems. First, there is the 2π ambiguity. Second, we would like to have a phase *residual* instead of the large total phase. Let us use the initial measured frequency $\hat{\omega}_0$ and phase $\hat{\theta}_0$ to calibrate the total phase to a phase residual

$$\hat{\phi}(t) = \Phi(t) - \hat{\omega}_0 (t - t_0) - \hat{\theta}_0 \tag{10}$$

where t now runs over all time beyond the starting time t_0 of the run. Note that $\hat{\phi}(t)$ depends on the calibration parameters $\hat{\omega}_0$, $\hat{\theta}_0$, so it is not intrinsic. Its average over batch k is

$$\hat{\phi}_k = \bar{\Phi}_k - \hat{\omega}_0 (\bar{t}_k - t_0) - \hat{\theta}_0 \tag{11}$$

These are the batch phase residuals that we would like to compute. From Eq. (9) it follows that, to first order,

$$\left.\begin{aligned} \hat{\phi}_k &\approx \psi_k - \hat{\omega}_0 (\bar{t}_k - t_0) - \hat{\theta}_0 \pmod{2\pi} \\ \hat{\phi}_0 &\approx 0 \pmod{2\pi} \end{aligned}\right\} \tag{12}$$

To a good approximation, then, we know the $\hat{\phi}_k$, modulo 2π. Because of the assumption of small local variations, we also can predict, with an error $< \pi$, how many radians the average total phase advances from one batch to the next, given its previous behavior. With this information, and with the measured $\hat{\phi}_0$ assumed to be 0, the 2π ambiguity can be removed sequentially from all the $\hat{\phi}_k$ by means of a second-order unwrapping algorithm given in Appendix C. It is the same algorithm, with different parameters, that is used for unwrapping the picket fence time-interval measurements that capture the 1-Hz zero crossings.

The algorithm also produces a sequence of prediction errors z_k that satisfies $|z_k| \leq \pi$. It measures how much the current phase differs from what we think it should be, based on the behavior of the previous batches. If any $|z_k|$ exceeds a certain threshold, now set at $\pi/2$, a caution is issued to the user. Perhaps the frequency is changing so fast that the assumption of small variations fails for the batch length N_{xb}. In effect, the analyzer may be losing phase lock, like a phase-locked loop whose bandwidth is too small. If this happens, the user can try decreasing N_{xb}. As mentioned above, the amplitude and phase residual averages for a frame are obtained simply by averaging n_{bf} batch values. Thus, if the user has to decrease N_{xb} to keep the analyzer in lock, he can maintain his chosen averaging time by increasing n_{bf}.

B. Differential Phase

By differential phase we mean some method of subtracting the phases of two channels that are being sampled simultaneously at the same rate. There are two flavors of differential phase processing. In S–S or X–X differential phase, it is assumed that both channels (1 and 2) originate at the same RF band and are downconverted to the same frequency. In this case, the total phases should not be too far apart, and so it makes sense to compute the batch averages

$$\delta\Phi_k = \Phi_k(1) - \Phi_k(2) - 2\pi n_0$$

where (1) and (2) identify the two channels and n_0 is the integer that makes $-\pi < \delta\Phi_0 \leq \pi$. Applying Eq. (11) to both channels, we obtain

$$\delta\Phi_k = \hat{\phi}_k(1) - \hat{\phi}_k(2) + (\hat{\omega}_0(1) - \hat{\omega}_0(2))\,(\bar{t}_k - t_0) + \hat{\theta}_0(1) - \hat{\theta}_0(2) - 2\pi n_0 \tag{13}$$

which gives the intrinsic quantity $\delta\Phi_k$ in terms of measured quantities.

The original design of the analyzer included a sample-and-hold unit so that channels 1 and 2 could be sampled simultaneously. This is no longer the case and, hence, the channel samples have to be interleaved at total rate $2f_s$ through the A–D converter: (1), (2), (1), (2), $\cdots$, where a channel 1 sample is paired with the *following* channel 2 sample. To deal with this situation, we use current batch frequency estimates to adjust the total phases of the two channels as if they were sampled halfway between the channel 1 sample time and the channel 2 sample time. The phase advance of channel 1 over a delay $1/(4f_s)$ is estimated as $\pi f_{\mathrm{vid}}(1)/(2f_s)$, where $f_{\mathrm{vid}}(1)$, the current estimate of the analog carrier frequency of channel 1, is computed by $f_{\mathrm{vid}}(1) = f_s(n_{\mathrm{base}}(1) + \hat{o}_k(1)/(2\pi))$. A similar correction of opposite sign is applied to the channel 2 total phase. Consequently, a correction

$$\frac{\pi}{2}\,(n_{\mathrm{base}}(1) + n_{\mathrm{base}}(2)) + \frac{1}{4}\,(\hat{o}_k(1) + \hat{o}_k(2))$$

has to be added to $\delta\Phi_k$.

In S–X differential phase, channel 1 is downconverted from 2295 MHz (S-band), channel 2 from 8415 MHz (X-band), or the reverse, and we are required to produce some version of

$$\text{S band phase } - \frac{3}{11}\text{ (X band phase)}$$

In a preliminary design, the analyzer simply computed the nonintrinsic quantity $\hat{\phi}_k(1) - (3/11)\hat{\phi}_k(2)$, which depends on the initial measured frequencies $\hat{\omega}_0(i)$, $i = 1, \cdots, 2$, and which, if a linear fit is not removed, has a random ramp component that depends on these measured frequencies. The current design uses a more objective method in which the measured frequencies are replaced by a priori known design frequencies $\omega_0(i) = f_s o_0(i)$. These are computed from the user-provided analog offset frequencies $f_{\text{ofst}}(i)$ by $\omega_0(i) = 2\pi(f_{\text{ofst}}(i) - f_s n_{\text{base}}(i))$. One can then produce phase residuals $\phi_k(i) = \hat{\phi}_k(i) + (\hat{\omega}_0(i) - \omega_0(i))(t_k - t_0)$ that start at zero but show ramps if the actual channel frequencies differ from the design frequencies. S–X differential phase is now just $\phi_k(1) - (3/11)\phi_k(2)$, which shows a ramp if the frequencies of the S- and X-channels are not related in exactly the right way. In contrast with the S–S or X–X situations, the first sample of this differential phase is zero; we are calibrating for frequency only and not attempting to measure the absolute synchronization of the two channels.

As with amplitude and phase, the batch averages of differential phase are combined into frame averages.

C. Time Series

The stream of narrow-band samples (frame average amplitude residuals, phase residuals, or differential phases) can be collected into a buffer and plotted against time. In the present software, we use a buffer management scheme that automatically subsamples the buffer by a factor of 2 when it fills up, crunches it to half its size, and begins to accept data at half the previous rate. At any time during the run, the buffer contains a record of the entire data stream, subsampled by some power of 2. Because phase residuals and differential phases are likely to be dominated by a straight line, we normally apply the calibration operation described in Section V.B before plotting them so that random fluctuations can be seen.

D. Spectrum

Any of the streams of narrow-band samples can be subjected to the same spectral estimation process. Because it takes longer to collect the data arrays, there is incentive to use the narrow-band data more efficiently than the medium-band data. In compensation, there is more processor time available per A–D sample for expensive postprocessing. We use an unweighted Thomson multitaper spectral estimator [10,7 (Chapter 7)] with orthogonal data tapers (trig prolates) computed by the author [5] (Appendix B). The user chooses a FFT size N_{fft}, a power of 2. At the start of the test, we compute an array of K orthogonal data tapers $u_k[n; N_{\text{fft}}, w], n = 0, \cdots, N_{\text{fft}} - 1, k = 0, \cdots, K - 1$. The value of K depends on w and on the sidelobe level we wish to tolerate in the frequency responses of the u_k. In the present design, $w = 4$, $K = 4$. An array of samples $x[0], \cdots, x[N_{\text{fft}} - 1]$, called a "narrow-band frame" (nbframe), is preconditioned by the calibration operation of Section V.B. Then K distinct "eigenspectra" $S_0, \cdots, S_{K-1}$ are computed by applying the tapers and a real FFT, giving

$$S_k[m] = \frac{N_{\text{xf}}}{N_{\text{fft}} f_s} \left| \sum_{n=0}^{N_{\text{fft}}-1} x[n] u_k[n; N_{\text{fft}}, w] \exp\left(-i2\pi nm/N_{\text{fft}}\right) \right|^2 \tag{14}$$

with frequency array $f[m] = (f_s/(N_{\text{xf}} N_{\text{fft}}))m,\ m = 0, \cdots, N_{\text{fft}}/2$. The spectrum of the nbframe is computed by averaging the eigenspectra:

$$S[m] = \frac{1}{K} \sum_{k=0}^{K-1} S_k[m] \tag{15}$$

and the overall run spectrum is computed by accumulating and averaging all the nbframe spectra. One advantage of this method is that, over smooth regions of the true spectrum, the variance of $S[m]$ is about K times smaller than the variance of each $S_k[m]$. With a single-taper method, variance could be reduced by using shorter nbframes or averaging the spectrum over frequency. Either of these methods increases the resolution bandwidth.

To prepare the spectrum for display, we cut off frequencies below $(f_s/(N_{\text{xf}}N_{\text{fft}}))w$ and do the usual conversion to dBc/Hz. The resolution bandwidth W_{nb} is given by

$$\frac{1}{W_{\text{nb}}} = \frac{1}{K}\sum_{k=0}^{K-1}\frac{1}{W_{\text{nb},k}}$$

where

$$W_{\text{nb},k} = \frac{f_s N_{\text{fft}}}{N_{\text{xf}}\left(\sum u_k[n]\right)^2}$$

is the resolution bandwidth of S_k. Although it is not apparent, W_{nb} is proportional to $1/N_{\text{fft}}$; one can use N_{fft} to trade off resolution against run length.

The user should be aware that the spectral window of this method is not bell shaped but approximately rectangular with ripples across the top. If the spectrum has a bright line whose width is of the order of one FFT bin or less, the image of the line may appear to have four small peaks at the top. These are artifacts of the method and do not indicate a splitting of the line. (See Appendix B and Fig. 3.)

In the current version of narrow-band processing, we have achieved bandwidth reduction by unweighted averaging: The batch samples of amplitude and phase are, to first order, unweighted averages of these quantities, and frame samples are unweighted averages of batch samples. Consequently, a calculated

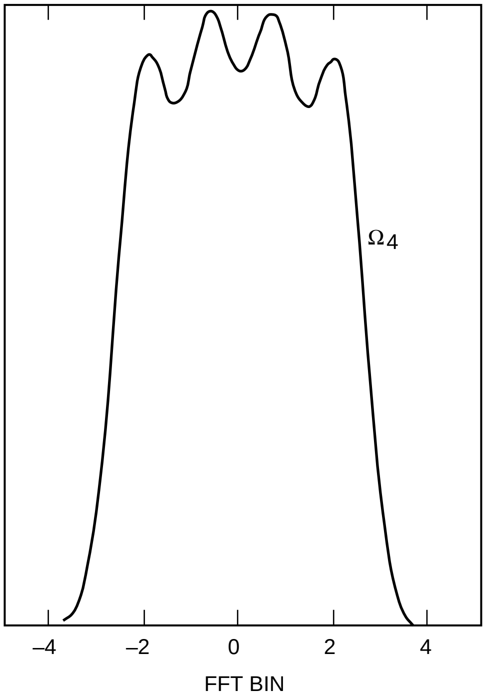

Fig. 3. Shape of a bright line for narrow-band spectrum.

spectrum for frequencies between 0 and the Nyquist frequency $f_s/(2N_{\mathrm{xf}})$ is not, strictly speaking, an estimate of the spectrum of the quantity in that frequency range, but rather an estimate of the spectrum of the averages of the quantity over an averaging time $\tau_0 = N_{\mathrm{xf}}/f_s$, sampled at rate $1/\tau_0$. This spectrum suffers from both aliasing and distortion. The Pony method of extracting batch samples of amplitude and phase leads inherently to this situation for frames consisting of one batch. The main decision was how to deal with further bandwidth reduction: whether to use a lowpass decimation filter, a bank of such filters, or simply to extend the situation with unweighted averaging. The advantages of the chosen design are simplicity, consistency, and flexibility in the choice of decimation factor (frame length), which can be large enough to exhaust the patience of any user.

E. Allan Deviation

The stability analyzer can compute the Allan deviation of frame samples of phase or differential phase for an array of averaging times τ that are powers of 2 times the frame duration τ_0. It was required to remove an estimate of linear frequency drift from the results. For a drift estimator, we use the simple three-point estimator suggested by Weiss [11]. Although the basic method is covered in [2] and [3], we run through the computations for a particular value of $\tau = n\tau_0$. Let the stream of phase samples be $\phi_0,\ \phi_1, \cdots$. At a given point in the run, we have accumulated sums of the first and second powers of m second differences of ϕ_i with stride n, namely,

$$s_p = \sum_{j=2}^{m+1} \left(\Delta_n^2 \phi_{nj}\right)^p, \qquad p = 1,\ 2$$

where $m \geq 4$. (The author realizes that the sum for $p = 1$ telescopes.) We have also collected a subsampled version

$$\phi_0, \phi_d, \phi_{2d}, \cdots, \phi_{Id}$$

of the whole run so far by the same buffer mechanism used for time series above. The calculations proceed as follows:

$$n_c = \left\lfloor \frac{I}{2} \right\rfloor d$$

$$D_c = \phi_{2n_c} - 2\phi_{n_c} + \phi_0 \qquad \text{(unscaled drift estimate)}$$

$$V = \frac{s_2}{m} - \left(\frac{s_1}{m}\right)^2 \qquad \text{(sample variance)}$$

$$V = V + \left(\frac{s_1}{m} - D_c \left(\frac{n}{n_c}\right)^2\right)^2 \qquad \text{(drift correction)}$$

$$\nu = (m-1)\left(0.8776 + 0.0643 e^{-(1/2)(m-4)}\right) \qquad \text{(degrees of freedom)}$$

$$V^{\pm} = V\left(1 \pm \sqrt{\frac{2}{\nu}}\right)$$

$$\sigma_y(\tau) = \frac{\sqrt{V}}{\sqrt{2}\; 2\pi f_{\mathrm{ref}} \tau}, \qquad \sigma_y^{\pm}(\tau) = \frac{\sqrt{V^{\pm}}}{\sqrt{2}\; 2\pi f_{\mathrm{ref}} \tau} \qquad \text{(Allan deviation with error bar)}$$

The formula for degrees of freedom is an empirical formula fitted to the author's numerical results for the random-walk-of-frequency model of phase deviations (f^{-4} noise). The error bars, which are really the square roots of "one-sigma" error bars for $\sigma_y^2(\tau)$, should be conservative for f^{β} noise, $\beta > -4$.

VI. Zero-Crossing Processing

To capture the up-crossing times of the 1-Hz square wave, a preliminary measurement of the nominal period p of the square wave is taken with the interval timer, which is then set to measure the time intervals between each subsequent up-crossing and the next pulse of a 10-Hz train of reference pulses, the "picket fence." These readings are unwrapped into a sequence of time residuals, as described in [4]. The algorithm, which is really the same as the one used for unwrapping the narrow-band phase deviations (Appendix C), need not be reproduced here. The time deviations produced by this algorithm are multiplied by the scale factor

$$\frac{2\pi f_{\mathrm{ref}}}{f_{\mathrm{mix}} p}$$

to give phase deviations that can be used like the batch averages of phase deviation that come from the narrow-band process. For time series and Allan deviation, we allow only one batch per frame, as the 1-second period is natural for the user. For spectrum, an arbitrary number of batches per frame is allowed so that users can shrink the Nyquist frequency below 0.5 Hz as much as they want.

References

[1] J. C. Breidenthal, C. A. Greenhall, R. L. Hamell, and P. F. Kuhnle, "The Deep Space Network Stability Analyzer," *Proceedings of the 26th Annual Precise Time and Time Interval Applications and Planning Meeting*, Reston, Virginia, December 5–8, 1994, in press.

[2] C. Greenhall, "Removal of Drift From Frequency Stability Measurements," *The Telecommunications and Data Acquisition Progress Report 42-65, July–August 1981*, Jet Propulsion Laboratory, Pasadena, California, pp. 127–131, October 15, 1981.

[3] C. Greenhall, "Frequency Stability Review," *The Telecommunications and Data Acquisition Progress Report 42-88, October–December 1986*, Jet Propulsion Laboratory, Pasadena, California, pp. 200–212, February 15, 1987.

[4] C. Greenhall, "A Method for Using a Time Interval Counter to Measure Frequency Stability," *IEEE Trans. Ultrason. Ferroelec. Freq. Control*, vol. 36, pp. 478–480, 1989.

[5] C. Greenhall, "Orthogonal Sets of Data Windows Constructed From Trigonometric Polynomials," *IEEE Trans. Acoust. Speech. Sig. Proc.*, vol. 38, pp. 870–872, 1990.

[6] S. Marple, *Digital Spectral Analysis With Applications*, Englewood Cliffs, New Jersey: Prentice Hall, 1987.

[7] D. Percival and A. Walden, *Spectral Analysis for Physical Applications*, Cambridge, United Kingdom: Cambridge University Press, 1993.

[8] T. Pham, J. Breidenthal, and T. Peng, "Stability Measurements of the Radio Science System at the 34-m High-Efficiency Antennas," *The Telecommunications and Data Acquisition Progress Report 42-114, April–June 1993*, Jet Propulsion Laboratory, Pasadena, California, pp. 112–139, August 15, 1993.

[9] D. Slepian, "Prolate Spheroidal Wave Functions, Fourier Analysis, and Uncertainty—V: The Discrete Case," *Bell System Tech. J.*, vol. 57, pp. 1371–1430, 1978.

[10] D. Thomson, "Spectrum Estimation and Harmonic Analysis," *Proc. IEEE*, vol. 70, pp. 1055–1096, 1982.

[11] M. Weiss and C. Hackman, "Confidence on the Three-Point Estimator of Frequency Drift," *Proceedings of the 24th Annual Precise Time and Time Interval Applications and Planning Meeting*, McLean, Virginia, pp. 451–455, 1992.

Appendix A

The Pony Calculation

This is a batch method for computing the frequency, amplitude, and phase of a sampled sine wave. It comes from a method of harmonic analysis called Prony's method [6 (Chapter 11)], which analyzes a waveform into the sum of n sine waves. The calculation we call "Pony" is simply a modification of Prony's method for $n = 1$.

I. Part 1: Frequency

Let the data array be $x[0], \cdots, x[N-1]$. If $x[n]$ were exactly of form $A \cos\,(on + \theta)$, then we would have

$$x[n+1] + x[n-1] = (2\cos o)x[n], \qquad n = 1, \cdots, N-2 \tag{A-1}$$

On the other hand, if $x[n]$ is a noisy cosine wave, then let us estimate $\cos o$ by projecting the vector $x[n+1] + x[n-1]$ orthogonally onto the vector $x[n]$. The computation is

$$c = \frac{(1/2)\,(x[0]x[1] + x[N-2]x[N-1]) + \sum_{n=1}^{N-3} x[n]x[n+1]}{\sum_{n=1}^{N-2} x[n]^2}$$

$$o = \arccos\,(c) \quad \text{in } [0, \pi]$$

if $|c| \leq 1$, else o goes to the nearest port in the storm, 0 or π. One may also change the sign of o according to polarity considerations.

For use in Part 2 and elsewhere, a single-precision complex array of powers $u^n,\ n = 0, \cdots, N-1$, where $u = \exp\,(-io)$, is generated by a vectorized algorithm that we illustrate for the case $N = 16$. Compute the dyadic powers u^2, u^4, u^8 in double precision and convert them to single precision. Lay down the powers u^0, u^8 in the array, multiply them by u^4, and lay down the products to give u^0, u^4, u^8, u^{12}. Multiply by u^2 to give $u^0, u^2, \cdots, u^{14}$. Multiply by u to give the desired array. For large N, the successive steps get more and more efficient for a vector processor.

II. Part 2: Amplitude and Phase

Having estimated the frequency, we use it to estimate amplitude and phase. Let $a = A\cos\theta$, $b = A\sin\theta$, and solve the least-squares problem

$$x[n] \approx a\cos\ on\ - b\sin\ on, \qquad n = 0, \cdots, N-1$$

for the parameters a and b. The coefficients of the normal matrix can easily be expressed in closed form, and the solution computed as follows:

$$x_c = \sum_{n=0}^{N-1} x[n]\cos\ on, \qquad x_s = -\sum_{n=0}^{N-1} x[n]\sin\ on \tag{A-2}$$

$$\mathrm{cc} = \frac{1}{2}\left[N + \cos\ (o(N-1))\frac{\sin\ oN}{\sin\ o}\right], \qquad \mathrm{ss} = \frac{1}{2}\left[N - \cos\ (o(N-1))\frac{\sin\ oN}{\sin\ o}\right]$$

$$\mathrm{cs} = \frac{1}{2}\sin\ (o(N-1))\frac{\sin\ oN}{\sin\ o}$$

$$D = \mathrm{cc}\cdot\mathrm{ss} - \mathrm{cs}^2$$

$$a = \frac{\mathrm{ss}\cdot x_c + \mathrm{cs}\cdot x_s}{D}, \qquad b = \frac{\mathrm{cs}\cdot x_c + \mathrm{cc}\cdot x_s}{D}$$

$$A = \sqrt{a^2 + b^2}, \qquad \theta = \mathrm{angle}(a + ib)$$

Most of the work is in the in-phase and quadrature mixing operation [Eq. (A-2)], which uses the array u^n whose generation is described in Part 1.

The calculation given here can be regarded as an improvement on the approximations

$$a \approx \frac{2}{N}x_c, \qquad b \approx \frac{2}{N}x_s$$

which are exact if oN is an integer. It has been observed [8] that these approximations are inadequate if oN is not an integer, because the double-frequency terms have not entirely been eliminated by the mixing–filtering operation, Eq. (A-2).

Appendix B
Windows and Filter

The data tapers and lowpass decimation filter are based upon the author's "trig prolate" approximations [5] to the discrete prolate spheroidal sequences of Slepian [9]. The notation $u_k[n; N, w]$ is used here in place of the notation $u_k[n; N, w/N, w]$ in [5].

Figure 1 shows the frequency responses (spectral windows) of the data tapers used for spectral estimation. The Ω_{05} curve applies to full-band spectrum, Ω_{04} to medium-band spectra, and Ω_4 to narrow-band spectra. Note that Ω_4 is the average of the windows of the four eigenspectra, Eq. (14), that are averaged into the total spectrum, Eq. (15). The expectation of a spectral estimate is the convolution of the true spectrum with the spectral window. The Ω_{0w} windows are bell shaped. Figure 3 plots Ω_4 on a linear scale against a two-sided frequency axis to show how a narrow bright line would appear in the spectral estimate if it were plotted on a linear scale. The ripples at the top will not be so prominent on a typical dB scale.

The N-point FIR lowpass filter used in medium-band processing before decimation by r is built in a conventional way from the trig prolate window $u_0[n; N, w]$. The formula for it is

$$h_r[n] = u_0[n; N, w] \operatorname{sinc}\left(2\pi f_h \left(n - \frac{N-1}{2}\right)\right), \qquad n = 0, \cdots, N-1$$

normalized so that $\sum h_r[n] = 1$, where

$$w = 4, \qquad N = 16r, \qquad f_h = \frac{0.4}{r}, \qquad \operatorname{sinc} x = \frac{\sin x}{x}$$

Figure 2 shows the frequency response of this filter for $r = 2$. The response is essentially the same for all r if frequency is scaled according to the x-axis of Fig. 2. Only one table is needed to represent the frequency response for the purpose of equalizing the medium-band spectra.

Appendix C
Phase Unwrapping Algorithm

This algorithm produces the narrow-band phase residuals $\hat{\phi}_k$ from the carrier frequency and phase estimates extracted from each batch by the Pony calculation. It is assumed that the batches all have length N_{xb} and are adjacent. Recall the definition, Eq. (5), of the mods function. Let the damping constant λ be a number between 0 and 1. In the following algorithm, ψ'_k is related to ψ_k of the main text by $\psi'_k = \psi_k - \hat{o}_0(N_{\mathrm{xb}} - 1)/2$.

```
$\psi'_0 = \hat{\theta}_0$
$z_0 = 0, \hat{\phi}_0 = 0, q_0 = 0$
For $k =\ 1, 2, \cdots$

    Obtain the batch frequency and phase $\hat{o}_k$, $\hat{\theta}_k$.
    $\psi'_k = (\hat{o}_k - \hat{o}_0)(N_{\mathrm{xb}} - 1)/2 + \hat{\theta}_k$      ! $\psi_k$ is total phase $\Phi_k$ mod $2\pi$.
    $z_k =$ mods $(\psi'_k - \psi'_{k-1} - \hat{o}_0 N_{\mathrm{xb}} - q_{k-1},\ 2\pi)$      ! prediction error.
    If $|z_k| > \pi/2$ (say), then issue caution "losing lock" to user.
    $\hat{\phi}_k = \hat{\phi}_{k-1} + q_{k-1} + z_k$      ! output phase residual.
    $q_k = q_{k-1} + \lambda z_k$      ! low pass-filtered $\Delta\hat{\phi}_k$.

Next $k$.
```

Note that q_k, z_k satisfy

$$q_k = (1 - \lambda) q_{k-1} + \lambda \Delta\hat{\phi}_k, \quad z_k = \Delta\hat{\phi}_k - q_{k-1}$$

This says that q_k is a lowpass-filtered version of $\Delta\hat{\phi}_k$, and z_k is a prediction error for $\Delta\hat{\phi}_k$. The basis of the algorithm is (1) the assumption that $|z_k| < \pi$ and (2) the knowledge of z_k modulo 2π, namely,

$$z_k = \Delta\Phi_k - \hat{\omega}_0 \Delta\bar{t}_k - q_{k-1} \cong \Delta\psi_k - \hat{o}_0 N_{\mathrm{xb}} - q_{k-1} \qquad (\mathrm{mod}\ 2\pi)$$

Any value for λ in [0, 1] is meaningful. If $0 < \lambda < 1$, then, in effect, a weighted average of previous phase advances, with weights $(1 - \lambda)^n$, is used to judge what the current phase advance should be. In the script files that drive the software, λ has been set to 1/10. This provides some stability against large errors while maintaining the ability of the algorithm to follow frequency drifts.