Development of an Algorithm for the Detection of Coherency in Radar Signal Waveforms

by

Spyridon Mathew Alifrangis

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering

APPROVED:

F. Ricci, Chairman

P. Knight D. Schaefer

December 1989
Blacksburg, Virginia
Development of an Algorithm for the Detection of Coherency in Radar Signal Waveforms

by

Spyridon Mathew Alifrangis

Committee Chairman: Fred Ricci
Electrical Engineering

(ABSTRACT)

The estimation of the stability of radar emissions is of considerable interest in the evaluation of radar clutter rejection performance and also for the general knowledge of the waveform required for the design of threat simulators. It should be stressed that for the estimation of clutter rejection capability, it is the stability of the entire waveform that is of general importance, although the stability of parameters such as phase, Pulse Repetition Interval (PRI) and amplitude are typically specified because of the ease in instrumenting the measurement. The parametric estimates are indeed the most useful in describing the characteristics of the waveform but not necessarily for evaluating clutter rejection performance.

Two broad categories into which radar emissions can be subdivided are coherent and non-coherent RF. A great deal of confusion often surrounds the use of these terms, especially among those who measure radar emissions rather than those who build the radar sets. For the purposes of this paper, coherence will be defined in terms of the square root of the variance of the first pulse-to-pulse phase difference, $\sigma(\Delta\theta)$. For the case where $\sigma(\Delta\theta) << 1$ radian, the signal will be considered coherent. When the phase is uniformly distributed over $2\pi$ radians, the signal will be considered non-coherent. Since it is likely
that, for most practical signals, the signal will be well within one of these two categories, ambiguity will be unlikely.

If a radar emission is observed to be coherent, it implies that the radar uses this property for Moving Target Indication (MTI) processing. The performance of the MTI will probably, but not necessarily, depend on the pulse-to-pulse phase stability as the most critical parameter for this type of system. Alternatively, if the radar emission is observed to be non-coherent, it implies that if the radar has an MTI processor, it is likely that it is of the stored reference variety. The performance of the MTI will probably, but again not necessarily, depend on the pulse-to-pulse RF stability as the most critical parameter.

The common thread between the two types of systems which indicates clutter rejection performance is the repeatability of adjacent pulse waveforms regardless of phase. This is not to imply that phase is not critical; it is important for determining the type of processor. The difference lies in the fact that for the internally coherent system, the phase information of the coherent reference oscillator is not observable as it is for the externally coherent system. Hence, the only hint that such an emitter has an MTI processor is contained in the repeatability of adjacent pulse waveforms.

This paper addresses the general problems of detecting coherence, estimating MTI performance, and estimating the phase stability, frequency stability and PRI stability using sample data derived from a system based on the IBM-PC. Both the analysis and radar waveform generation systems were implemented in software utilizing Microsoft Fortran and Microsoft C compilers.
# TABLE OF CONTENTS

**CHAPTER 1**  
INTRODUCTION ............................................ 1

**CHAPTER 2**  
THEORY .................................................. 4  
  2.1 Initial Signal Processing ................................. 4  
  2.2 Direct Estimation of Phase and Frequency Stability ....... 9  
  2.3 Cancellation Ratio Minimization ...................... 23

**CHAPTER 3**  
ACCURACY ................................................. 28  
  3.1 Internal Errors ................................... 28  
    3.1.1 Cancellation Ratio Minimization ................. 29  
    3.1.2 Minimum Variance Bounds for Δθ and T .......... 31  
    3.1.3 Direct Estimates of Phase, Frequency and PRI ..... 33  
    3.1.4 Cancellation Ratio Measurement Bounds .......... 34  
    3.1.5 Prefiltering .................................. 36  
    3.1.6 Signal-to-Noise Ratio Limitations ............... 39  
      3.1.6.1 Digitization Noise ................ 40  
      3.1.6.2 Summary of S/N Effects .............. 46  
  3.2 External Errors .................................. 51  
    3.2.1 Amplitude Variation ........................... 51  
    3.2.2 Aircraft Emitter Phase Measurement Error ....... 58  
    3.2.3 Effects of Scanning Emitter .................. 63  
    3.2.4 Propagation Path Effects .................. 68
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Non-linear transformation of real signal f(t) into enveloped phase.</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Presentation of predetected signal, envelope and phase sample data.</td>
<td>11</td>
</tr>
<tr>
<td>3</td>
<td>Illustration of difference between actual and measured phase.</td>
<td>14</td>
</tr>
<tr>
<td>4</td>
<td>Sample phase advance of a coherent signal where the PRI is constant.</td>
<td>16</td>
</tr>
<tr>
<td>5</td>
<td>Illustration of aperture error.</td>
<td>42</td>
</tr>
<tr>
<td>6</td>
<td>Quantizer characteristic (with rounding).</td>
<td>43</td>
</tr>
<tr>
<td>7</td>
<td>Probability density function for roundoff error.</td>
<td>44</td>
</tr>
<tr>
<td>8</td>
<td>The standard deviation of the estimate of PRI vs. signal-to-noise ratio in a 20MHz bandpass.</td>
<td>47</td>
</tr>
<tr>
<td>9</td>
<td>The standard deviation of the estimate of first phase difference vs. signal-to-noise ratio in a 20 MHz bandpass.</td>
<td>48</td>
</tr>
<tr>
<td>10</td>
<td>The expected value of the cancellation ratio vs. signal-to-noise ratio in a 20 MHz bandpass.</td>
<td>49</td>
</tr>
<tr>
<td>11</td>
<td>The standard deviation of the intrapulse frequency estimate vs. signal-to-noise ratio.</td>
<td>50</td>
</tr>
<tr>
<td>12</td>
<td>Mainlobe and first two sidelobes of scan pattern.</td>
<td>55</td>
</tr>
<tr>
<td>13</td>
<td>Cancellation ratio limit due to scan induced amplitude variation.</td>
<td>56</td>
</tr>
<tr>
<td>14</td>
<td>Airborne target collection geometry.</td>
<td>59</td>
</tr>
<tr>
<td>15</td>
<td>Antenna scan geometry.</td>
<td>64</td>
</tr>
<tr>
<td>16</td>
<td>Geometry of two path propagation model.</td>
<td>70</td>
</tr>
</tbody>
</table>
17  Illustration of the effect of PRI variation on the direct sampling approach (random variations in phase, \( \theta(t) \), has been set to zero for the sake of illustration). .................................. 75
18  Region of summation for \( i \) and \( n \) (\( N=4 \)). ........................................... 85
19  Illustration of the dependency of the bias on the number of pulses and noise process type. ........................................... 91
20  Cancellation ratio estimation limit vs. \( \sigma(\Delta f)\tau \). ........................................ 94
21  Limitation of cancellation ratio vs. \( \sigma(\Delta \theta) \). ........................................... 96
22  Positive and negative 1\( \sigma \) limits of cancellation ratio estimate due to pulse-to-pulse frequency variation. ................................. 99

LIST OF DIAGRAMS

<table>
<thead>
<tr>
<th>DIAGRAM</th>
<th>TITLE</th>
<th>PAGE NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Functional Block Diagram.</td>
<td>3</td>
</tr>
</tbody>
</table>
CHAPTER 1 INTRODUCTION

This paper presents measurement techniques for the estimation of radar parameter stability. The characterization of parameter stability is of primary interest to a wide range of consumers within the ELINT community. A list of the fields of interests includes ECM design, weapons systems threat assessment, and digital communications. In each field, the objective is to assess system performance based on measures of parameter stability. The goal is to develop a useful and comprehensive characterization of parameter stability that can be understood and applied uniformly by the various communities.

The standardization of the measurement of parameter stability has become critical as an increasing number of precision digital collectors are deployed. Previous methods of stability estimation have varied widely in responding to the requirement for expedient analysis of particular signals. As a result, it is difficult to evaluate comparatively these stability metrics. It is, therefore, desirable that radar parameter stability estimation be accomplished by standardized procedures which are well documented.

The underlying prerequisites for measurements of stability depend on the purpose and type of processing performed in the collection of specific threat signals. This information, particularly for "new" signals, is often unavailable when the measurements are made. Clearly, the requirements for stability estimation call for measurements which are repeatable, and which can be scientifically and comparatively evaluated with similar measurements obtained from other collection systems. In addition, it is important to be able to estimate the accuracy of the stability measurement based on the number of data samples used in the computation.

ELINT requirements call for both long and short term measurements of stability with the measurement time specified. This paper will be concerned only with the short
term stability. The terms "coherence" and "short term RF stability" are used interchangeably. The term "coherence" implies an ability to predict the future waveform of a signal based on a past observation of it. In other words, if the RF and phase of a carrier frequency can be determined during one period of time, an ideal waveform could be completely predictable for all time in the future.

Coherence is a very important property of some radar signals. This is true for radars designed to measure or to discriminate on the basis of target velocity. In this case, the reflected signal, as Doppler shifted by the target's motion, must be analyzed. If the signal is not coherent (stable), its spectrum will have sidebands of energy near the carrier frequency. These sidebands, when reflected by non-moving targets, can produce strong clutter returns which appear in the same part of the frequency spectrum as that of the Doppler shifted target echo signal. There are several ways to measure coherence, which is really the same thing as short term frequency stability. These include spectrum analysis, as well as a number of more specialized techniques. The development of the stability estimation procedures is given in Chapter 3. Although emphasis is placed on estimation of phase and frequency stability of a source radar, the estimation techniques presented may be applied to any measurable radar parameter. A major concern of this paper is to present the processing techniques which have been investigated and shown to provide good results. From these, only one was chosen to be implemented in a software system since it proved to be both realizable and practical. The source code for this technique is given in the Appendix. Its related functional block diagram can be found in Diagram 1 on the following page. Another major concern is to present the results of an investigation of the accuracy of proposed MTI performance metrics and parameter estimation techniques. The error sources (such as antenna scan motion, aircraft doppler and multipath) are certainly not exhaustive, but are considered significant.
Diagram 1  Functional Block Diagram.
CHAPTER 2  THEORY

2.1 Initial Signal Processing

Much of the signal processing associated with this analysis system consists of determining the envelope and phase of a (real) signal [1]. It can be shown that a real signal \( f(t) \) can be represented by

\[
f(t) = \text{Re} \left\{ f(t) \ e^{j\Omega t} \right\}
\]

where \( f(t) \) is the complex envelope which is a slowly varying function of time if \( f(t) \) is narrowband. This implies that \( F(\omega) \), the Fourier transform of \( f(t) \), is identically zero for \( \omega < -\Omega \).

The analytic representation of \( f(t) \) is

\[
g(t) = f(t) + j\hat{f}(t)
\]

where \( \hat{f}(t) \) is the Hilbert transform or quadrature component of \( f(t) \) defined by

\[
\hat{f}(t) = H \left\{ f(t) \right\} = f(t) \ast \frac{1}{\pi t}
\]

where the asterisk represents convolution. It can also be shown that

\[
g(t) = f(t) \ e^{j\Omega t}
\]
Therefore, the envelope and phase advance, $\Omega t$, can be obtained from operations on the analytic signal by

$$ f(t) = \left| g(t) \right| = \left( f^2(t) + \hat{f}^2(t) \right)^{1/2} \quad \text{2.5a} $$

$$ \Omega = \tan^{-1} \left[ \frac{\hat{f}(t)}{f(t)} \right] \quad \text{2.5b} $$

More specifically consider the narrowband signal

$$ f(t) = A(t) \cos \left( \omega_0 t + \theta(t) \right) \quad \text{2.6} $$

where $\omega_0$ is the carrier frequency, $\theta(t)$ is an undetermined phase component and $A(t)$ is real. Using the Hilbert transform it can be found that

$$ \hat{f}(t) = \mathcal{H} \left\{ f(t) \right\} = \mathcal{H} \left\{ A(t) \cos(\omega_0 t + \theta(t)) \right\} \quad \text{2.7} $$

$$ = A(t) \sin \left( \omega_0 t + \theta(t) \right) $$

Therefore, equations 2.5a and 2.5b become

$$ f(t) = \left( A^2(t) \sin^2(\omega_0 + \theta(t)) + A^2(t) \cos^2(\omega_0 + \theta(t)) \right)^{1/2} $$

$$ = A(t) $$

and

$$ \Phi(t) = \tan^{-1} \left\{ \frac{A(t) \sin(\omega_0 t + \theta(t))}{A(t) \cos(\omega_0 t + \theta(t))} \right\} \quad \text{2.8} $$

$$ = \omega_0 t + \theta(t) $$
The instantaneous frequency is defined by

\[ f = \frac{\partial \Phi(t)}{\partial t} = \frac{\omega_0}{2\pi} + \frac{1}{2\pi} \frac{\partial \theta(t)}{\partial t} \]  

and the average frequency over a time period \( T \) is

\[ f_T = \frac{\Phi(t) - \Phi(t - T)}{2\pi T} = \frac{\omega_0}{2\pi} + \frac{\theta(t) - \theta(t - T)}{2\pi T} \]  

The variance of the random phase component is given by

\[ \sigma^2_\theta = \mathbb{E} \left\{ (\theta(t) - \mathbb{E} \{ \theta(t) \})^2 \right\} = R_{\theta}(0) \]  

where \( \mathbb{E} \{ \cdot \} \) is the expected value and \( R_{\theta}(\tau=0) \) is the autocovariance function of the random phase component evaluated at zero lag. The variance of the first phase difference is an important phase stability estimation and is given by

\[ \sigma^2(\Delta \theta) = \mathbb{E} \left\{ (\Phi(t) - \Phi(t - T))^2 \right\} \]  

\[ = \mathbb{E} \left\{ (\theta(t) - \theta(t - T))^2 \right\} \]  

\[ = 2 \left( R_{\theta}(0) - R_{\theta}(T) \right) \]  

Note that 2.12a and 2.12b are equivalent because the difference of \( \Phi(t) \) and \( \Phi(t - T) \) has a zero mean and \( \theta(t) \) is a zero mean process. From equations 2.10 and 2.12b the variance of the average frequency change is
\[ \sigma^2_{\theta} = \mathbb{E} \left\{ \left( \Phi(t) - \Phi(t-T) \right) - \mathbb{E} \left\{ \Phi(t) - \Phi(t-T) \right\} \right\}^2 \frac{1}{2\pi T} \]

\[ = \frac{\sigma^2(\Delta \theta)}{2\pi T} \tag{2.13} \]

Important to the determination of the parameter accuracy estimates (Chapter 4) is the effect of the addition of noise in the directly sampled signal which is primary thermal noise from the receiver RF section. For this analysis, the noise will be assumed white (uncorrelated). Including the additive noise, the in-phase and quadrature components of the signal can be written

\[ X(t) = f(t) + n(t) \tag{2.14a} \]

\[ \hat{X}(t) = \hat{f}(t) + \hat{n}(t) \tag{2.14b} \]

where \( n(t) \) and \( \hat{n}(t) \) are zero mean white noise components with \( \sigma^2_n = \sigma^2_x = \sigma^2_{\hat{n}} = \sigma^2_{\hat{x}} \), the noise power. From equations 2.5a and 2.5b, the phase and amplitude can be written

\[ \Phi(t) = \tan^{-1} \left[ \frac{\hat{X}(t)}{X(t)} \right] \tag{2.15a} \]

\[ \rho(t) = \left( x^2(t) + \hat{x}^2(t) \right)^{1/2} \tag{2.15b} \]

The problem is then one of the non-linear transformation of variables. It can be shown that, if a function \( g(x, \hat{x}) \) is "smooth," the variance of \( g(x, \hat{x}) \) containing two random variables is
\[
\sigma_g^2 (x, \hat{x}) = \left[ \frac{\partial g(x)}{\partial x} \right]_{x=x}^2 \sigma_x^2 + \left[ \frac{\partial g(\hat{x})}{\partial \hat{x}} \right]_{\hat{x}=\hat{x}}^2 \sigma_{\hat{x}}^2 \\
+ \frac{\partial g(x)}{\partial x} \bigg|_{x=x} \frac{\partial g(\hat{x})}{\partial \hat{x}} \bigg|_{\hat{x}=\hat{x}} \sigma_x \sigma_{\hat{x}} 
\]

where \( \sigma_{x\hat{x}}^2 \) is the covariance of \( x \) and \( \hat{x} \), and where \( \bar{x} \) and \( \bar{\hat{x}} \) are the mean values of \( x \) and \( \hat{x} \) respectively [2].

Consider the typical signal of the form

\[
f(t) = A \cos(\omega t + \theta(t))
\]

Following through the operation indicated by equation 2.16, noting that \( \sigma_{x\hat{x}}^2 = 0 \), then since \( x(t) \) and \( \hat{x}(t) \) are orthogonal, it is found that

\[
\sigma_g^2 = \frac{\sigma_n^2}{2(A^2/2)} = \frac{1}{2S/N} \quad 2.18a
\]

\[
\sigma_p^2 = \sigma_n^2 \quad 2.18b
\]

Therefore, a major step in the processing of the data is the implementation of the Hilbert Transform. The convolution involved by the Hilbert Transform implies integration over all time. This is obviously not strictly practical. The Hilbert Transform has, therefore, been approximated by a Finite Impulse Response (FIR) digital filter which satisfies certain maximum mean square error criteria. An explanation of the particular implementation that is used is given in [3].
The FIR filter operation consists of digitally convolving the FIR filter sequence with the data. The resulting sequence is an approximation to the imaginary part of the analytic signal which is often referred to as the quadrature component. The original signal is often called the in-phase component. This preprocessing of the data is illustrated in Figure 1. An example of the envelope and phase derived by application of the Hilbert Transformer to predetected samples of a single pulse is shown in Figure 2. From the intrapulse and interpulse phase samples, the phase and frequency stability of the signal can be estimated, at least for discrete time intervals corresponding to multiples of the Pulse Repetition Interval (PRI) and sampling interval.

2.2 Direct Estimation of Phase and Frequency Stability

In this section a method will be presented by which the interpulse phase variation can be estimated directly by obtaining the sample phase data and removing the modulo $2\pi$ ambiguities. By assuming the functional nature of the intrapulse phase advance, the variance of the pulse phase estimates can be reduced by linear regression. Another technique for obtaining the same type of estimates is presented in section 2.4. This technique overcomes many of the shortcomings of the direct approach, especially in the case of non-linear intrapulse phase advance (FMOP, etc.) where the direct approach becomes problematic, and especially when more than one intrapulse phase is used to obtain the pulse phase estimate.
f(t) = A \cos(\omega t + \Theta(t))

\Phi(t) = \text{Signal Phase}

A(t) = \text{Signal Amplitude}

Figure 1  Non-linear transformation of real signal f(t) into enveloped phase.
Figure 2  Presentation of predetected signal, envelope and phase sample data.
Assume a signal of the form

\[ y(t) = A P_T(t) \sin \Phi(t) \]  \hspace{1cm} 2.19a

where

- \( A \) = constant amplitude
- \( P_T(t) \) = pulse train of infinite extent with PRI = T
- \( \Phi(t) = \omega_0 + \theta(t) \), the phase advance of \( y(t) \).  \hspace{1cm} 2.19b
- \( \omega_0 \) = the down converted Intermediate Frequency (IF) (generally non-zero and dependent on tuning error but within a few percent of the reciprocal of the pulse duration, \( \tau \)).
- \( \theta(t) \) = random phase.

When attempting to measure the phase advance of \( y(t) \) as a function of time, one can only sample \( \Phi(t) \) over a pulse duration, \( \tau \), at intervals of the PRI, T. Therefore, the phase function available for measurement is,

\[ \Phi_o(t) = P_T(t) \Phi(t) \]  \hspace{1cm} 2.20

where

\[ P_T(t) = \sum_{n=-\infty}^{\infty} P_\tau(t - n\tau) \]  \hspace{1cm} 2.21a

and

\[ P_\tau(t) = \begin{cases} 1 & 0 \leq t < \tau \\ 0 & \text{elsewhere} \end{cases} \]  \hspace{1cm} 2.21b
Moreover, the phase as measured by observing a sinusoidal waveform, is modulo $2\pi$ and, hence, the measured phase advance is

$$\hat{\Phi}_o(t) = \left\{ P_T(t) \Phi(t) \right\}_{2\pi}$$

where $\{ \cdot \}_{2\pi}$ indicates modulo $2\pi$ and the cap represents an actual measurement. This is illustrated in Figure 3.

The task then becomes one of measuring the statistics of $\theta(t)$ over intervals of $T$, in particular, the variance of $\theta(t)$ and the variance of $\theta(t) - \theta(t - T)$.

By using a sample data measurement system, the signal available for measurement of $\theta(t)$ is:

$$y(m\Delta) = A P_T(m\Delta) \sin\left(\omega_o m\Delta + \theta(m\Delta)\right)$$

where $m = 1, 2, 3 \ldots N_s$

$N_s$ = total number of samples

$\Delta$ = Sample interval

Assume that $\Phi_o(m\Delta)$ can be derived from equation 2.23 using a technique such as that employing a Hilbert transform approximation (or perhaps by complex demodulation). Because of quantization, $\Phi_o(m\Delta)$ has a random error associated with each sample in addition to other statistical sample errors, all of which will be assumed uncorrelated at least for the intrapulse sampling. In order to reduce the errors, the phase advance over a pulse can be estimated by linear regression of an order corresponding to the assumed intrapulse phase behavior, e.g., if linear FMOP is known to exist one might attempt a second order
\[ \Phi(t) = \omega_0 t + \theta(t) \]

- \( \{\Phi(t) P_T(t)\}_{2\pi} \) = Measured Phase Advance
- \( \Phi(t) = \omega_0 + \theta(t) \) = Signal Phase Advance
- \( \omega_0 \) = Down Converted Intermediate Frequency
- \( \theta(t) \) = Random Phase Component
- \( P_T \) = Pulse Train of Infinite Extent with Pulse Repetition Interval = T

Figure 3 Illustration of difference between actual and measured phase.
fit. For the present, however, the discussion will be limited to only a linear intrapulse phase advance. Hence, if the PRI variation is small such that the instantaneous PRI deviation from the mean over a sample set of $N$ pulses is never greater than the pulse width, then the phase measurement samples at intervals $T$ seconds apart are

$$\dot{\Phi}_o(nT) = \{ \Phi(nT) \}_{2\pi}$$

where $n = 1, 2, 3, \ldots N$

Therefore

$$\dot{\Phi}_o(nT) = \{ \omega_o nT + \theta(nT) \}_{2\pi}$$

If $\omega_o$ could be adjusted to zero by down converting the signal in equation 2.19a by $\omega_o$ (zero beat Intermediate Frequency (IF)) then

$$\dot{\Phi}_o(nT) = \{ \Phi(nT) \}_{2\pi}$$

If the signal is coherent then $\sigma_\theta^2 << 1$ (radian)$^2$ and hence

$$\dot{\Phi}_o(nT) = \theta(nT)$$

This situation is illustrated in Figure 4.

Unfortunately, however, the exact value of $\omega_o$ cannot be precisely determined because of finite signal-to-noise ratio (SNR) and limited observation time. Hence, a precise zero beat may never occur with absolute certainty. The consequences of this shall be investigated.
Using a sample data measuring system, the signal available for measurement of $\theta(t)$ is:

\[ y(m\Delta) = AP_1(m\Delta)\sin(\omega_p M\Delta + \theta(m\Delta)) \]

where $m = 1, 2, 3 \ldots N_s$

$N_s = \text{Total Number of Samples}$

$\Delta = \text{Sample Interval}$

$\Phi(m\Delta) = \text{Sampled Measured Phase Advance}$

**Figure 4** Sample phase advance of a coherent signal where the PRI is constant.
If \( \omega_0 < \frac{2\pi}{NT} \) (near zero IF) then

\[
\hat{\phi}_0(nT) = \omega_0 nT + \theta(nT)
\]

Additionally, there are no modulo 2\( \pi \) ambiguities (no greater than one roll over will occur) as illustrated in Figure 4(B).

In general, if \( \frac{2\pi k - 2\pi}{T} \omega_0 < \frac{2\pi k + 2\pi}{NT} \frac{2\pi}{T} \) \( \frac{2\pi}{NT} \omega_0 \) \( \frac{2\pi}{NT} \)

\[
\hat{\phi}_0(nT) = \omega_\alpha nT + \theta(nT)
\]

where \( \omega_\alpha = \omega_0 - \frac{2\pi k}{T} \) and \( -\frac{2\pi}{NT} \omega_\alpha < \frac{2\pi}{NT} \omega_\alpha \).

This corresponds to tuning to near a PRF harmonic at \( \omega_\alpha \) and may be thought of as an aliasing effect of the sampled phase.

At this point it should be clear that if zero beat cannot be achieved, the phase advances given by the interpulse phase samples will have a different slope than the phase advance given by the interpulse phase samples. If the PRI is not constant this may lead to considerable error. The reason for determining the phase \( \hat{\phi}_0(nT) \) is to estimate the statistics of the random phase component \( \theta(t) \) from \( \hat{\phi}_0(nT) \). To estimate the autocovariance \( R_\theta(\tau) \)
and hence $\text{VAR} \theta(t)$, etc. it would be necessary to either zero beat precisely or induce aliasing such that the signal is essentially tuned to a PRF line component. This will remove the trend in the phase advance which is necessary if $R_\theta(t)$ is to be accurately estimated. If the central moments of the nth differences in $\hat{\Phi}_o(nT)$ are to be estimated, however, it is not necessary to zero beat or tune to a PRF line if the signal is coherent and the modulo $2\pi$ ambiguities can be properly removed.\(^1\)

Of particular interest is the variance of the first phase difference defined by

$$\sigma^2(\Delta \theta) = E \left\{ (\theta(t) - \theta(t - T))^2 \right\} = \text{VAR} \left[ \theta(t) - \theta(t - T) \right]$$

$$= E \left\{ (\theta(nt) - \theta((n - 1)T))^2 \right\}$$

where $\theta(t)$ is assumed to be a zero mean random process.

For the condition given by equation 2.29 the variance is defined as

$$\text{VAR} \left( \hat{\Phi}_o(nT) - \hat{\Phi}_o((n - 1)T) \right)$$

$$= \text{VAR} \left( \omega_\alpha nT - \omega_\alpha(n - 1)T + \theta(nT) - \theta(n - 1)T \right)$$

$$= \text{VAR} \left( \theta(nT) - \theta(n - 1) \right) = \sigma^2(\Delta \theta)$$

\(^1\) Removal of the modulo $2\pi$ ambiguities is considered routine if the signal is coherent.
since $\omega_0 T$ is constant (assuming either invariant PRI or that T is the average PRI under the conditions of PRI variation considered above). Therefore, if $\omega_0$ can be adjusted such that

$$|\omega_0| < \frac{2\pi}{NT}$$

then there is no phase ambiguity and $\theta(t)$ can be estimated from $\Phi_o(nT)$ of an N pulse sample.\(^1\)

In order to adjust $\omega_0$ in accordance with the above condition, however, $\omega_0$ must be estimated such that

$$\left(\text{VAR} (\omega_o)\right)^{1/2} \ll \frac{2\pi}{NT} \quad 2.31$$

where $\hat{\omega}_o$ is the estimate of $\omega_o$.

By assuming a linear phase advance and using linear regression over the above samples within a pulse, it can be shown that [4]

$$\text{VAR} (\hat{\omega}_o) = \frac{12\sigma^2_0}{\Delta^2 n^3} \quad 2.32$$

where $\sigma^2_0$ is the phase variance (due to the signal and/or the system) and $n$ is the number of phase samples per pulse used in establishing the regression lines. If the estimates of frequency are averaged over $N$ pulses, assuming the frequency samples are Gaussian distributed and uncorrelated (white), then from equation 2.32

---

\(^1\) The bias error caused by a finite number of samples will be discussed in the next section.
Applying the condition given in equation 2.31 to 2.33, it is found that

\[ \sigma_\theta = \frac{\pi (\tau/T) \sqrt{n}}{(3N)^{1/2}} \]

where \( \tau/T \) is the duty cycle. The limit of \( \sigma_\theta \) due to white additive Gaussian distributed noise is given in equation 2.18a as

\[ \sigma_\theta = \frac{1}{(2 \ S/N)^{1/2}} \]

From 2.34 and 2.35 it can be determined that the signal-to-noise ratio must be in the range

\[ S/N >> \frac{3N}{2\pi^2 (\tau/T)^2 n} \]

Typical signal parameters for signal processing are:

- \( \tau = 1 \mu\text{sec} \)
- \( n = 100 \)
- \( \Delta = 10^{-8} \text{ sec} \)
- \( T = 1\text{msec} \)
- \( N = 20 \)

For these parameters the signal must have \( S/N \) of greater than 45dB as specified by equation 2.36.

\[ S/N >> 45\text{dB} \]

For example, in a 6 bit A/D converter the maximum achievable signal-to-noise ratio from equation 3.35 is approximately 36 dB. Hence, the condition given by equation 2.36 will not be met except for duty cycles much larger than are likely to be observed even if the \( S/N \)
of the signal in the receiver is much greater than that given by equation 2.36. It can, therefore, be assumed that the signal can never be perfectly zero beated for most signals of interest.

Through iterative techniques using interpulse phase sampling, it has been demonstrated that a signal can be easily "zero beated" to one of the spectral line components by application of interpulse linear regression. This is to eliminate the longer term trends in the interpulse phase advance (Modulo $2\pi$) which are necessary for accurately estimating the interpulse phase statistics.

The above analysis effectively assumes that the phase advance is constant within the pulse and hence coherent phase measurements from equidistant time samples are decoupled from variations in PRI if they are not greater than the pulse width. A constant phase advance does not occur in signals with intrapulse Frequency Modulation (FMOP). Moreover, observation of a variety of both coherent and non-coherent RADAR signals has shown that large amounts of apparently unintentional intrapulse phase variation is common and, hence, changes in the PRI of these signals can cause phase measurement errors in the application of the above technique. It is, therefore, more reasonable to fix the phase sampling to the pulse envelope (e.g., the centroid). In section 3.2.5 it is shown that if the phase is sampled in this way, the variance of the first phase difference at the average PRI is subject to a bias error $\varepsilon^2$ dependent on the residual frequency offset $\Delta \omega_0 = \omega_0 - \hat{\omega}_0$ and the variance of the PRI, $\sigma_t^2$. The bias error is given by
\[
\varepsilon_0^2 = (\Delta \omega_0) \sigma_f^2
\]  
2.37

For most signals of interest (including scanning signals) the bias error can be made less than about 10 KHz (see section 3.1.3). Additionally, as shown in section 3.2.5, the phase stability measurement error is negligible.

The question may also arise as to what value will the estimate of \( \sigma^2(\Delta \theta) \) converge as the phase instability increases. In the limit it will be assumed that the phase samples \( \hat{\Phi}_0(nT) \) will be uniformly distributed over \((0, 2\pi)\) and independent (due to multiple \(2\pi\) rollovers). Hence,

\[
\hat{\Phi}_0(nT) \sim U(0, 2\pi) \quad n = 1, 2, 3, \ldots N
\]

and from equations 2.29 and 2.30 the variance of the first phase difference is therefore

\[
\sigma^2(\Delta \theta) = E \left\{ \left( \theta(nT) - \theta((n+1)T) \right)^2 \right\} - E^2 \left\{ \theta(nT) - \theta((n+1)T) \right\}
\]

\[
= E \left\{ \theta^2(nT) \right\} - 2E \left\{ \theta(nT)\theta((n+1)T) \right\} + E \left\{ \theta^2((n+1)T) \right\}
\]

\[
= 2E \left\{ \theta^2(nT) \right\} - 2E \left\{ \theta(nT) \right\} E \left\{ \theta((n+1)T) \right\}
\]

\[
= 2 \left[ \frac{(2\pi)^2}{3} \right] - 2\pi^2
\]

\[
= \frac{2\pi^2}{3}
\]  
2.38
Therefore, for a non-coherent signal $\sigma(\Delta \theta)$ from equation 2.38 converted to degrees should be around $146^\circ$.

### 2.3 Cancellation Ratio Minimization

Since most radar information is extracted from changes in the signal from pulse-to-pulse, it is then desirable to estimate the stability of the pulsed waveform on the same basis. In particular the determination of the cancellation ratio $C$ to estimate MTI performance and the stability of individual parameters such as phase, amplitude and timing from pulse-to-pulse is an aid in estimating the type of signal processing employed by the radar set.

By minimizing the calculated $C$, with respect to phase and time, one may obtain an optimized estimate of $C$ in addition to very accurate estimates of the first difference of phase and time (PRI). This method will be termed the "optimum" processor technique such as that which minimizes $C$ with respect to time and phase in order to achieve the best clutter rejection for MTI processing. Assuming that the radar is employing a relatively standard processor, then this estimate of $C$ should be a good indicator of MTI performance regardless of whether or not the radar is coherent on receive or transmit.

The cancellation ratio in terms of continuous waveforms is given by

$$C = \frac{\int_T^{T_0} (V(t) - U(t))^* (V(t) - U(t)) \, dt}{\int_0^{T_0} \left\{ |V(t)|^2 + |U(t)|^2 \right\} \, dt}$$

2.39
where

\[ V(t) = A(t)e^{i\alpha(t)} \quad 2.40a \]
\[ U(t) = B(t)e^{i\beta(t)} \quad 2.40b \]

\( T_o \) = observation time

and \( | \cdot | \) represents the modulus of a complex number. The asterisk represents complex conjugate while both \( V(t) \) and \( U(t) \) are representations of the complex envelope waveforms of adjacent pulses. Initially assume that the signal-to-noise ratio is large enough so that the effect of noise can be neglected. Also, it will be assumed that the observation interval \( T_o \) completely includes the pulse and that the time shift between the pulses is zero.

From equation 2.39

\[ C = 1 - \frac{1}{D} \int_0^{T_o} (U(t)V^*(t) + V(t)U^*(t)) \, dt \quad 2.41 \]

where

\[ D = \int_0^{T_o} \left( | V(t) |^2 + | U(t) |^2 \right) \, dt, \text{ the total energy.} \quad 2.42 \]

Substituting for \( V(t) \) and \( U(t) \)

\[ C = 1 - \frac{1}{D} \int_0^{T_o} \left\{ (A(t)B(t)e^{i(\beta(t) - \alpha(t))} + A(t)B(t)e^{-i(\beta(t) - \alpha(t))} \right\} \, dt \]

\[ = 1 - \frac{2}{D} \int_0^{T_o} A(t)B(t)\cos(\beta(t) - \alpha(t)) \, dt \quad 2.43 \]

If the RADAR processor memorizes or otherwise has knowledge of the phase and timing between pulses, then \( C \) may be smaller than the result given by equation 2.43. The
desired result is the minimized $C$ with respect to both time difference between pulses $\tau$ and phase difference $\Phi$. It suffices to maximize the integral in equation 2.43 with respect to parameters $\tau$ and $\Phi$ or

$$\max \{ \chi \}$$

where

$$\chi = \int_{0}^{T_0} \text{Re} \left\{ U(t) V^*(t + \tau) e^{i\Phi} \right\} \, dt$$

2.44

It is also shown in section 3.1.1 that the $\tau$ and $\Phi$ which satisfy 2.44 are the maximum likelihood estimates of these quantities denoted $\hat{\tau}$ and $\hat{\Phi}$ respectively if the noise is additive white, Gaussian and band-limited, and if there are no "stray" parameters. Such estimates have the minimum possible variance. The problem then appears to be equivalent to one of the simultaneous estimations of more than one parameter. In this case it is the estimation of the time difference (or difference in time-of-arrival) and the first pulse phase difference that is of interest. If other parameters are varying from pulse-to-pulse, then the variance of the estimates of $\tau$ and $\Phi$ obtained by minimizing $C$, will in general be greater, as Manasse points out [5].

To find the maximum of $\chi$, one can seek the condition

$$\frac{\partial \chi}{\partial \tau} = 0 \text{ and } \frac{\partial \chi}{\partial \Phi} = 0$$

The resulting solutions for $\tau$ and $\Phi$ will be the estimates denoted by $\hat{\tau}$ and $\hat{\Phi}$ respectively.
Therefore,

\[ \frac{\partial \chi}{\partial \Phi} = \int_{0}^{T_0} A(t)B(t+Q) \sin (\beta(t+Q) - \alpha(t) + \Phi) \, dt = 0 \]  \hspace{1cm} 2.45

By using trigonometric expansion of the sine factor and solving for $\Phi$ in equation 2.45, the result is

\[ \Phi = \tan^{-1} \left\{ \frac{\int_{0}^{T_0} A(t)B(t+Q) \sin (\beta(t+Q) - \alpha(t)) \, dt}{\int_{0}^{T_0} A(t)B(t+Q) \cos (\beta(t+Q) - \alpha(t)) \, dt} \right\} \]  \hspace{1cm} 2.46

Imposing the other condition, one finds that

\[ \frac{\partial \chi}{\partial \tau} \bigg|_{\tau=\hat{\tau}} = \int_{0}^{T_0} A(t) \frac{\partial B(t+Q)}{\partial \hat{\tau}} \cos (\beta(t+Q) - \alpha(t) + \Phi) \, dt \\
- \int_{0}^{T_0} A(t) B(t+Q) \sin (\beta(t+Q) - \alpha(t)) \frac{\partial \beta(t+Q)}{\partial \hat{\tau}} \, dt = 0 \]  \hspace{1cm} 2.47

At this point the objective of determining the optimum $C$ with respect to phase and time difference and of estimating the phase and time difference should not be lost. Merely determining $\max \{ \chi \}$ is a result unto itself; that is, from equation 2.43

\[ \min \{ C \} = 1 - \frac{\hat{\tau}}{D} \max \{ \chi \} \]  \hspace{1cm} 2.48

Quantitatively this result does not imply a particular dependence on any parameterization characteristics of the pulsed signal, although for most signals of interest, parameterization is valid. Therefore, assume that there are no stray parameters except for perhaps a pulse-to-pulse change in amplitude. That is,
\[ \beta(t) = \Delta \theta + \alpha(t) \quad 2.49a \]

and

\[ B(t) = \rho A(t) \quad 2.49b \]

where \( \rho \) and \( \Delta \theta \) are constants for a particular pulse pair. The quantity \( \rho \) would probably be caused by scan induced amplitude modulation and \( \Delta \theta \) is the pulse-to-pulse phase difference (first phase difference). Using 2.49a and 2.46 one finds that

\[ \phi = \tan^{-1} \left\{ \frac{\sin \Delta \theta \int_0^T \rho A(t) A(t + \hat{\tau}) \, dt}{\cos \Delta \theta \int_0^T \rho A(t) A(t + \hat{\tau}) \, dt} \right\} \]

\[ \hat{\phi} = \Delta \theta \quad 2.50 \]

From equation 2.47 by substituting 2.49a and 2.49b one can show that

\[ \frac{\partial T}{\partial \tau} = \int_0^T A(t) A(t + \tau) \, dt \bigg|_{\tau=\hat{\tau}=0} = 0 \quad 2.51 \]

Hence, if there are no stray parameters other than pulse amplitude then \( \tau \) is an unbiased estimate of \( \Delta \text{TOA} \) and \( \Phi \) is an unbiased estimate of \( \Delta \theta \). In Chapter 3 it will be shown that these estimates are maximum likelihood estimates which have the desirable property of approaching the minimum achievable variance.
CHAPTER 3  ACCURACY

The factors that determine the accuracy of the parametric estimates, as well as the cancellation ratio estimates, can be logically subdivided into two categories. Those which are external to the collection and processing system and those which are internal. The external factors include propagation path conditions, antenna motion, scan motion and modulation. The internal factors include collection system effects such as quantization, local oscillator stability, time base stability, additive noise, and processing effects such as the method with which the estimates are obtained. To the greatest possible extent, each of these influences is investigated in an attempt to provide a reasonably detailed description of how the accuracy of the data derived stability estimates will be evaluated.

3.1 Internal Errors

For the purpose of this paper the Collection system will be treated as a black box introducing measurement errors from sources unknown with the exception of A/D conversion errors. The only receiver parameters that will be of concern are the additive thermal noise and L.O. instability, both of which are likely limiting factors in the phase stability measurement accuracy.

The investigation of the accuracy of the parameter estimates centers on the method by which the estimates are extracted from the digital data. Of particular concern here is the estimate of the minimum cancellation ratio, the estimates of first phase differences and PRI obtained by minimizing the cancellation ratio with respect to these parameters, and estimates of pulse phase and frequency by linear regression over the intrapulse phase history. In the following section it will be demonstrated that the cancellation ratio
minimization method gives the best possible parameter estimates, at least in the mean square sense. The estimates by first order linear regression also give the best results if the deviation from linear is uncorrelated. Since the pulse-to-pulse parameter estimates are shown to be unbiased, these estimates then have the minimum possible variance. This is highly desirable since variance is the metric by which the stability will be evaluated.

Although the individual intrapulse phase and frequency estimates are unbiased because of the assumed uncorrelated nature of the additive noise, which is the primary noise source for such short intervals, the pulse-to-pulse difference estimates of frequency, phase and TOA are likely to be somewhat biased. This is because of the highly correlated nature of many of the noise sources in the oscillators being measured. This will be explained later.

3.1.1 Cancellation Ratio Minimization

By minimizing the cancellation ratio, C, with respect to phase and TOA differences, one not only obtains a realistic measure upon which to base an MTI capability estimate, but also obtains very accurate estimates of these two parameters. It will be shown that the quantities $\tau$ and $\hat{\phi}$, which in Section 2.3 were shown to minimize C, are the maximum likelihood estimates (MLE) of the PRI, T, and the first phase difference, ($\Delta \theta$) respectively. This is an important conclusion since MLE's lead to mean square measurement errors which approach the Cramer-Rao bound (minimum variance) for large S/N [6]. Stray unknown parameters such as amplitude (for the estimate of C only) and pulse duration
will be ignored here, but as Whalen points out, if they have unknown random variations the effect is to increase the measurement error [7].

The likelihood function for white additive Gaussian noise is given by [8]

\[ L = F \exp \left\{ -\frac{1}{N_0} \int_0^\tau \left| U(t) - V(t, \theta_1, \theta_2) \right|^2 \, dt \right\} \]  

where \( F \) is an undetermined constant which is not of interest, \( N_0 \) is the noise spectral density, \( U(t) \) is the observation which is a complex signal representation, \( V(t, \theta_1, \theta_2) \) is a signal replica containing all known information about the signal in addition to those parameters that are to be estimated, \( \theta_1 \) and \( \theta_2 \), and \( \tau \) is the pulse duration. A maximum likelihood estimate of parameters \( \theta_1 \) and \( \theta_2 \) is obtained by those values \( \hat{\theta}_1 \) and \( \hat{\theta}_2 \) respectively, which maximize the likelihood function and more specifically minimize \( \psi \), where

\[ \psi = \int_0^{T_0} \left| U(t) - V(t, \theta_1, \theta_2) \right|^2 \, dt \]  

as one can see from equation 3.1.

Hence, the MLE of \( \theta_1 \) and \( \theta_2 \) is obtained by the simultaneous solution of two equations

\[ \frac{\partial \psi}{\partial \theta_1} = 0 \quad \text{and} \quad \frac{\partial \psi}{\partial \theta_2} = 0 \]  

Recall that the cancellation ratio is minimized if the numerator of equation 2.39 is minimized with respect to \( \tau \) and \( \Phi \), i.e.,
Assuming that there are no stray parameters, by comparing equations 3.2 and 3.4 it can be seen that the values of \( \hat{\tau} \) and \( \hat{\Delta \theta} \) are maximum likelihood estimates of \( \tau \) and \( \Delta \theta \) respectively. (See Section 2.3). It should be reiterated that this conclusion is only correct if there are no stray variable parameters. One might assume that it is more likely that there are no stray parameters if the signal is coherent then if it is non-coherent. Moreover, departure from a S/N limited value will indicate that there are stray parameters and that the variance of these estimates will be greater than the Cramer-Rao minimum for the simultaneous estimates of \( \Delta \theta \) and \( \tau \).

### 3.1.2 Minimum Variance Bounds for \( \Delta \theta \) and \( \tau \)

Since the quantity of interest for stability measurement is variance, it is advantageous to determine the minimum variance of the individual parameter estimates. If the estimators are unbiased (as they have been shown to be) the Cramer-Rao inequality states the lower bound for the error variance of simultaneous parameter estimates, regardless of the method used to obtain them.

Using the Cramer-Rao condition and assuming white additive Gaussian noise it can be shown that the variances of \( \hat{\Delta \theta} \) and \( \hat{\tau} \) obtained by simultaneous estimation (i.e., both are unknown) are [8]
\[ \sigma^2 (T) = \frac{1}{\beta^2 (2E/N_o)} \] 3.5

\[ \sigma^2 (\Delta \theta) = \frac{1}{(2E/N_o)} \left( 1 + \left( \frac{\omega_o}{\beta} \right)^2 \right) \] 3.6

where \( \omega_o \) is the residual radian frequency of the pulse, \( N_o \) is the noise spectral density, \( E \) is

the signal energy

\[ \beta^2 = \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega^2 |G(j\omega)|^2 d\omega - \frac{1}{2\pi} \int_{-\infty}^{\infty} \omega |G(j\omega)|^2 d\omega \]

\[ \frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \]

3.7

and \( |G(j\omega)|^2 \) is the signal power spectrum. Note from equation 3.6 that it is advantageous
to make \( \frac{\omega_o}{\beta} \ll 1 \). For this reason, the analytic signal is down-converted to near
zero

frequency by complex down conversion. The term containing \( \omega_o \) in equation 3.6 vanishes
resulting in

\[ \sigma^2 (\Delta \theta) = \frac{1}{(2E/N_o)} \]

For rectangular pulse, \( \beta \) is infinite, which cannot be correct since the signal is band limited.

Using an approximate analysis [5], it has been shown that

\[ \sigma^2 (T) = \frac{2\tau^2}{(2E/N_o)} \] 3.8

where \( \tau \) is the pulse duration.
3.1.3 Direct Estimates of Phase, Frequency and PRI

Direct estimates (i.e., those not obtained by minimization of C) of pulse phase and frequency are obtained by first order linear regression over the intrapulse advance of the phase samples obtained from digital Hilbert transformer derived analytic signal samples. The slope and mean of the linear regression line are obtained by minimizing the mean square error between a linear function and the data (least squares estimation (LSE)).

It can be shown that LSE is identical to MLE if the residual errors are normally distributed [6]. Since the phase advance is approximated as linear, this condition is violated for phase history functions of an FM chirped signal or any phase history that exhibits a trend other than linear. It will be assumed however, that the residual errors are normally distributed to provide a lower bound on the variance of the estimates. This being the case, the estimates of the intrapulse slope and mean of the phase history model of equation 2.19b have been derived and found to be [4,8]

\[ \sigma^2(\hat{\phi}) = \frac{1}{2(S/N) n} \]

\[ \sigma^2(\Delta \hat{\phi}) = \frac{1}{(S/N) n} \]

\[ \sigma^2(\hat{f}) = \frac{3}{2n(\pi \tau)^2(S/N)} \]

1 The phase advance, can be modeled by a polynomial of any order; and, if the residual errors are normally distributed, the LSE will be a MLE.
where \( n \) is the number of samples per pulse and \( S/N \) is the signal-to-noise power ratio in the measurement channel. The \( \Delta \) TOA (or PRI) measurement from a given digitizer is obtained by a leading edge threshold crossing; and, hence, for this measurement [6].

\[
\sigma^2(T) = 2 \left( \frac{1}{2B^2S/N} \right)
\]

where \( B \) is the approximate bandwidth of the IF channel.

These expressions are valid for large signal-to-noise-ratios as measured in the measurement channel including both receiver and digitization noise.

### 3.1.4 Cancellation Ratio Measurement Bounds

The limits of the measurement of \( C \) will now be developed. The additive noise from the receiver and from quantization are of primary concern. It is also important to know the individual effect of certain parameter variations on the results; and, therefore, the influence of phase noise, frequency noise and amplitude variations will be considered.

Assume that there is white additive noise, then

\[
\begin{align*}
V(t) &= V'(t) + n_v(t) \\
U(t) &= U'(t) + n_u(t)
\end{align*}
\]

where \( n_v(t) \) and \( n_u(t) \) represent the noise. In order to calculate just the effect of the noise, let \( V'(t) = U'(t) \) and substitute 3.13a and 3.13b into equation 2.39. Consequently, the estimate of \( C \) is

\[
\hat{C} = \frac{\int_0^\tau |n_v(t) - n_u(t)|^2}{4E}
\]

3.14
by assuming that the signal-to-noise ratio is large and letting
\[ E = \frac{1}{2} \int_0^\tau |V(t)|^2 \, dt = \frac{1}{2} \int_0^\tau |U(t)|^2 \, dt \]  \hspace{1cm} (3.15)
which is the energy in the signal per pulse.

Writing out the absolute value operation in 3.14 one finds that
\[ \hat{C} = \frac{1}{4E} \int_0^\tau \left( |n_v(t)|^2 + |n_u(t)|^2 - 2\text{Re} (n_v^* (t) n_u(t)) \right) \, dt \]  \hspace{1cm} (3.16)
where the asterisk represents complex conjugate and \( \text{Re}(\cdot) \) represents the real part of a complex number.

Taking the expected value of this estimate
\[ E \{ \hat{C} \} = \frac{1}{4E} \int_0^\tau \left( E \{ |n_v(t)|^2 \} + E \{ |n_u(t)|^2 \} \right) \, dt \]  \hspace{1cm} (3.17)
where the expected value of the cross product is zero because of the uncorrelated nature of the noise. The noise terms \( |n_v(t)| \) and \( |n_u(t)| \) are Rayleigh distributed and hence [7]
\[ E \{ |n_v(t)|^2 \} = E \{ |n_u(t)|^2 \} = 2\sigma^2 \]
where \( \sigma^2 \) is the noise variance. Therefore, from equation 3.17
\[ E \{ \hat{C} \} = \frac{4\sigma^2 \tau}{4E}, \quad \frac{\sigma^2 \tau}{S\tau} = \frac{1}{S/N} \]  \hspace{1cm} (3.18)
where \( S \) is the signal power.
3.1.5 Prefiltering

It has been shown that the estimates made of phase difference, frequency, and PRI are maximum likelihood estimates. Moreover, it is well known that for large signal-to-noise ratios maximum likelihood estimates lead to measurement errors that approach the Cramer-Rao bounds. These bounds have been previously derived directly from the likelihood function based on the assumption of white Gaussian additive noise [8]. It is significant to note that the lower variance bounds for the estimates of phase difference and $\Delta$ TOA for $C$ minimization do not depend on the noise bandwidth but on the signal characteristics and in particular the energy of the signal. In view of this, one should anticipate the "best" (lowest variance) estimates from this method with no further improvement possible. The estimates from the direct measurement technique, however, depend on the signal-to-noise ratio which can be improved by prefiltering.

It is, however, hard to believe that no further improvement can be made especially if the signal-to-noise ratio can be improved by "matching" the IF filters more closely to the actual bandwidth of the signal of interest. Detection Theory states that by implementing a matched filter in the signal processing sequence the signal-to-noise ratio is maximized in a sufficient way. In other terms, this means that any valuable information contained in the signal is not disposed of. If the noise is uncorrelated, then the matched filter is simply the signal reversed in time,

$$h(t) = S(T_0 - t)$$  \hspace{1cm} 3.19

where $h(t)$ is the impulse response of the matched filter and $S(t)$ is the signal of interest.

It can be shown that the output of the matched filter is equivalent to the output of a correlator at time $T_0$ (only) i.e.,
Further, it has been shown from equation 3.1 the likelihood estimates are equivalently obtained by maximizing the integral
\[ \int_{0}^{T_0} U(t)V^*(t) \, dt \]  \hspace{1cm} 3.21
with respect to the parameters to be estimated. By comparing equations 3.20 and 3.21, it can be seen that by maximizing 3.21 with respect to the unknown parameters the matched filter operation of equation 3.20 is performed. The signal-to-noise ratio of the measurement is maximized if the noise is white; prefiltering the signal before measurement will provide no additional improvement in the variance of the parameter estimates.

This can be demonstrated by considering a scheme (erroneous) by which the likelihood estimates are to be improved by a prefiltering operation. The pulse envelope and frequency of the signal of interest are known, and the phase difference and Δ TOA are to be estimated. Since the phase and Δ TOA are unknown, the frequency and pulse envelope are the only parameters which can be used to build a "matched" filter and consequently \( h(t) = A(\tau - t) \), where \( A(t) \) is the pulse envelope. But even here a reduction in noise bandwidth is promising. Using the complex envelope representation of the two adjacent pulses the output of the matched filter for each pulse is
\[ U'(t) = \int_{0}^{\tau} h(t_1) U(t - t_1) \, dt_1 = \int_{0}^{\tau} A(\tau - t_1) A(t - t_1)e^{j\theta} \, dt_1 \]  \hspace{1cm} 3.22a
where $t_1$ and $t_2$ are dummy variables.

The function that must be maximized in order to obtain the MLE and, therefore, minimize the cancellation ratio is from equation 2.39 given by

\[ \Psi = \frac{2 \int_0^\tau \text{Re} \left\{ U'(t) V^*(t) \right\} dt}{\int_0^\tau |U'(t)|^2 dt + \int_0^\tau |V'(t)|^2 dt} \]

where the primes reflect the use of the matched filter outputs.

Substituting 3.22a and 3.22b results in

\[ \Psi = \frac{2 \int_0^\tau \text{Re} \left\{ \int_0^\tau \int_0^\tau A(\tau-t_1)A(\tau-t_2)A(\tau-t_2)e^{j(\alpha-\beta)} dt_1 dt_2 \right\} dt}{2 \int_0^\tau \int_0^\tau \int_0^\tau A(\tau-t_1)A(\tau-t_1)A(\tau-t_2)A(\tau-t_2) dt_1 dt_2 dt} \]

Evaluating this expression at $t = \tau$ (where the matched filter has maximum output) and using the variable change $x = \tau - t$ results in

\[ \Psi = \frac{\int_0^\tau A^2(\tau - t_1) dt_1 \int_0^\tau \text{Re} \left\{ A(x)e^{j\alpha}A(x - T)e^{-j\beta} \right\} dx}{\left( \int_0^\tau A^2(\tau - t_1) dt_1 \right)^2} \]
It can be seen that 3.25 has the same form as 3.23 but it contains the signals without the prefiltering. Hence, prefiltering with a filter matched to the envelope of the signal does not improve the parameter estimates obtained by minimizing the cancellation ratio, C. Estimates of C, however, can be improved by "matched" filtering. If the bandwidth of the signal can be estimated, the noise power contributions from both digitization noise and receiver noise can be reduced by an amount equal to the ratio of the filter bandwidth and the noise bandwidth. This will be described in the next section.

3.1.6 Signal-to-Noise Ratio Limitations

In the following discussion the concern is with obtaining a reasonably accurate and practical means of accessing the performance of the parameter estimates in terms of IF signal-to-noise ratio (S/N). The IF S/N is a measurable quantity, although estimates of S/N under actual field conditions may be problematic. Firstly, the factor $2E/N_0$, which appears in expressions for the noise limited minimum variance bound, will be related to IF S/N.

Assume that the total noise spectral density in the measurement channel of a given digitizer having a 20MHZ bandwidth and including both receiver noise and digitization noise is given by
\[
\frac{N_o}{2} = 5\Delta\sigma_n^2 + N_d
\]

where \(\sigma_n^2\) is the thermal noise power (noise variance) in a channel bandwidth of one-fifth the sampling frequency and \(N_d\) is the digitization noise spectral density. Therefore,

\[
\frac{2E}{N_o} = \frac{S_T}{5\Delta\sigma_n^2 + N_d} = \frac{Sn}{5\sigma_n^2 + \frac{N_d}{\Delta}} = \frac{Sn}{5\sigma_n^2 + \sigma_d^2}
\]

Equation 3.27 can also be written in terms of individual contributions to the signal-to-noise ratio by

\[
\frac{2E}{N_o} = \frac{n}{5\frac{S}{(S/N)_n} + \frac{1}{(S/N)_d}}
\]

where \((S/N)_n\) is the thermal noise contribution from the IF bandpass and \((S/N)_d\) is the digitization noise component.

### 3.1.6.1 Digitization Noise

There are two major sources of error in the analog to digital conversion process, aperture error and quantization error. Both sources contribute to a decrease in the signal-to-noise ratio although quantization error is predominant for practical state-of-the-art high speed (100 MHz) A/D converters.

Aperture error is the result of the uncertainty of the instant of time in which the sample is obtained. Perhaps more concisely it may be considered as random jitter of the sample period which can be quantitatively observed as additive noise in the sampled
waveform. This is illustrated in Figure 5 which shows the relationship between amplitude and sample time variation. From Figure 5 it can be observed that

\[ \sigma_a^2 = \left( \frac{\partial f(t)}{\partial t} \right)^2 \sigma_t^2 \]  

where \( \sigma_a^2 \) is the amplitude variance, 
\( f(t) \) is the sampled waveform and
\( \sigma_t^2 \) is the variance of the aperture time.

Considering the sampling of a pulsed sinewave, if the slight edge effects is neglected, then the sample variance of amplitude exhibited over the entire pulse is

\[ \overline{\sigma_a^2} = \left[ \frac{1}{\tau} \int_0^\tau \left( \frac{\partial A \cos 2\pi f_0 t}{\partial t} \right)^2 \, dt \right] \sigma_t^2 \]  

\[ = \frac{A^2}{2} (2\pi f_0)^2 \sigma_t^2 \]  

Figures 6 and 7 illustrate the input/output characteristic of the quantizer and the quantization error distribution respectively. The variance of the quantization noise is given by

\[ \sigma_Q^2 = \frac{Q^2}{12} \]  

---

\[ \int_0^\tau \sin^2 \chi t \, dt = \frac{\tau}{2} \quad \text{if} \quad \tau \gg \frac{1}{\chi} \]
$f(t) = \text{Sampled Waveform}$

$a = \text{Amplitude}$

$\tau = \text{Aperture Time}$

$\frac{\partial f(t)}{\partial t} = \text{Sampled Time Variation}$

Figure 5 Illustration of aperture error.
Figure 6  Quantizer characteristic (with rounding).

Q = Quantization Level Size
A = Peak Amplitude of the Input Waveform
Figure 7 Probability density function for roundoff error.
where $Q$ is the quantization level size and $\sigma_Q^2$ is the quantization noise power. If it is assumed that the input waveform occupies the entire dynamic range of the quantizer with a peak amplitude $A$, then $Q = \frac{2A}{N_Q}$, where $N_Q = 2^b$, the number of quantization levels imposed by $b$ bits. Hence,

$$\sigma_Q^2 = \frac{A^2}{3N_Q} = \frac{A^2}{3 \cdot 2^b}$$

The total digitization noise power is, therefore,

$$\sigma_d^2 = \sigma_Q^2 + \sigma_a^2 = \frac{A^2}{3 \cdot 2^b} + \frac{A^2}{2} (2\pi f_o)^2 \sigma_t^2$$

and hence

$$\frac{(S/N)_d}{\sigma_d^2} = \frac{A^2 / 2}{(2\pi f_o)^2 \sigma_t^2} + \frac{1}{3 \cdot 2^b}$$

For our assumed digitizing system

- $b = 6$
- $\sigma_t = 40$ psec
- $f_o = 25$MHz

The S/N contribution due to digitization is then $(S/N)_d = 4944$ or 37 dB.
3.1.6.2 Summary of S/N Effects

The minimum variance bounds for the four estimates made from rectangular pulses are given by

\[ \sigma^2(\Delta \hat{\theta}) = \frac{1}{2E/N} \quad \omega_o^2 \ll \beta^2 \]  

3.36a

\[ \sigma^2(\hat{T}) = \frac{2\tau^2}{(2E/N_o)^2} = 2\Delta^2 \left( \frac{5}{(S/N)_n} + \frac{1}{(S/N)_d} \right)^2 \]  

3.36b

\[ \sigma^2(\hat{\tau}) = \frac{3}{2(\pi \tau)^2 S/N} = \frac{3}{2(\pi \tau)^2} \left( \frac{1}{(S/N)_n} + \frac{1}{(S/N)_d} \right) \]  

3.36c

\[ E\left\{ \hat{\tau} \right\} = \frac{\tau}{(E/\sigma^2_\tau)} = \frac{1}{S/N} = \left( \frac{1}{(S/N)_n} + \frac{1}{(S/N)_d} \right) \]  

3.36d

The expressions are generally accurate for large signal-to-noise ratios. By substituting equation 3.28 into equation 3.36a,b, the variance bounds in terms of the IF signal-to-noise ratio can be found. These quantities are plotted in Figures 8 through 11. Also plotted in Figure 8 is the variance of the PRI estimate based on the leading edge threshold crossing method given in equation 3.12 to illustrate the dramatic improvement in the measurement resolution theoretically possible from a maximum likelihood estimate.
Figure 8  The standard deviation of the estimate of PRI vs. signal-to-noise ratio in a 20MHz bandpass.
Figure 9 The standard deviation of the estimate of first phase difference vs. signal-to-noise ratio in a 20 MHz bandpass.
Figure 10  The expected value of the cancellation ratio vs. signal-to-noise ratio in a 20 MHz bandpass.
Figure 11 The standard deviation of the intrapulse frequency estimate vs. signal-to-noise ratio.
3.2 External Errors

3.2.1 Amplitude Variation

The effect of narrowband white noise on the measurement of cancellation ratio has already been considered. The bandwidth of this noise was on the order of, or greater than, the bandwidth of the signal. Now consider the effect of amplitude modulation superimposed on the RADAR waveform which has a bandwidth of much less than that of the RADAR emission. This AM could possibly be caused by antenna scanning, antenna wind movement or signal fading.

Let the adjacent pulse be represented by

\[ V(t) = C(t) A(t) e^{j\alpha(t)} \]  \hspace{1cm} 3.37a
\[ U(t) = C(t - T) B(t) e^{j\beta(t)} \]  \hspace{1cm} 3.37b

where \( T = PRI \) and \( V(t) \) and \( U(t) \) are as previously defined except for the lower frequency modulating function \( C(t) \).

The cancellation ratio given by equation 2.43 then becomes

\[ \hat{\mathcal{C}} = 1 - \frac{2}{\mathcal{D}} \int_0^\tau C(t) C(t - T) A(t) B(t) \cos(\beta(t) - \alpha(t)) \, dt \]  \hspace{1cm} 3.38

where \( \tau \) is the pulse duration and

\[ \mathcal{D} = \int_0^\tau \left( C^2(t) A^2(t) + C^2(t - T) B^2(t) \right) \, dt \]  \hspace{1cm} 3.39
Since $T \gg \tau$ usually, and if it is assumed that

$$2E = \int_0^\tau A^2(t) \, dt = \int_0^\tau B^2(t) \, dt$$  \hfill (3.40)

then

$$C = 1 - \frac{2C(t) C(t - T)}{D} \int_0^\tau A(t) B(t) \cos\left(\beta(t) - \alpha(t)\right) \, dt$$  \hfill (3.41)

where

$$D = 2E \left( C^2(t) + C^2(t - T) \right)$$  \hfill (3.42)

Investigating the case where the measurement of $C$ is limited only by the interpulse amplitude modulation (AM),

where

$$A(t) = B(t)$$  \hfill (3.43a)
$$\alpha(t) = \beta(t)$$  \hfill (3.43b)

Equation 3.41 becomes

$$\hat{C} = 1 - \frac{2C(t) C(t - T)}{C^2(t) + C^2(t - T)} = \frac{(R - 1)^2}{1 + R^2}$$  \hfill (3.44)

where $R = C(t - T) / C(t)$. 

Using the first order approximation $C(t - T) = C(t) - T \frac{\partial C(t)}{\partial t}$, then

$$R = 1 - T \left| \frac{\partial C(t)}{C(t)} \right| \bigg|_{t = t_0}$$ \hspace{1cm} 3.46

where $t_0$ is the time at which $C$ is to be evaluated.

Substituting equation 3.46 into 3.45 results in

$$\hat{C} = \frac{\left(T \frac{\partial C(t_0)}{C(t)}\right)^2}{\left(1 - T \frac{\partial C(t_0)}{C(t_0)}\right)^2 + 1}$$ \hspace{1cm} 3.47

The most common type of AM observed under typical intercept conditions is due to transmit antenna scan. The effect of a typical scan induced AM on the cancellation ratio estimate is the main lobe which will be investigated. Before doing so, however, it should be made clear that the effects of AM could be reduced by hardlimiting (either in hardware or software), as is probably done by the radar system, or by optimizing with respect to pulse amplitude. The latter type of compensation will be discussed later, but first the uncompensated case for scan induced AM will be developed.

Assume that the AM caused by scan near the main lobe can be approximated by

$$C(t) = \beta \cos \left( \frac{\pi t}{2NT} \right)$$ \hspace{1cm} 3.48
where \( T \) is the PRI,
\[
N \text{ is the number of pulse per main lobe, and}
\]
\( \beta \) is a constant which is of no interest.

This is illustrated in Figure 12. Differentiating 3.48 and substituting the result into equation 3.47 and evaluating for the worst case where \( t_0 = \pm NT \) results in
\[
\frac{\partial C(t)}{\partial t} \bigg|_{C(t)} = \frac{-\pi}{2NT}
\]
3.49

and
\[
\hat{\chi} = \left( -\frac{\pi}{2N} \right)^2 \left( 1 - \frac{\pi}{2N} \right)^2 + 1
\]
3.50

This is plotted in Figure 13 over typically encountered number of pulses per beamwidth. Note that a signal would need to have at least 100 pulses per beamwidth in order not to be limited by amplitude variation (and hence limited only by digital quantization). This indicates that optimization with respect to amplitude or some other type of compensation is desirable.

Since it has been shown in Section 3.1 that the phase and TOA difference (PRI) estimates obtained by minimizing \( C \) are independent of the interpulse amplitude variations, one may calculate the optimum amplitude scaling to minimize \( C \) without affecting the parameter estimates by merely maximizing \( \chi \) of equation 3.44 by
\[
\left. \frac{\partial \chi}{\partial \Lambda} \right|_{\Lambda = \hat{\Lambda}} = 0
\]
3.51
$A(t) = \cos \frac{2\pi t}{2NT}$

**Figure 12** Mainlobe and first two sidelobes of scan pattern.
Approximate Quantization Noise Limitation

Figure 13  Cancellation ratio limit due to scan induced amplitude variation.
where $\Lambda = \frac{C(t-T)}{C(t)}$

Substituting equations 3.37a and 3.37b into 3.38 results in

$$\frac{W}{D} = \frac{\Lambda \int_0^{T_0} A(t)B(t)\cos\left(\beta(t+\tau) - \alpha(t) - \theta\right) \, dt}{\int_0^{T_0} |V(t)|^2 \, dt + \Lambda^2 \int_0^{T_0} |U(t)|^2 \, dt}$$

3.52

Performing the operation indicated by equation 3.51 on equation 3.52 finds that the $\Lambda$ for which $C$ is minimized is

$$\Lambda = \left(\frac{\int_0^{T_0} |V(t)|^2 \, dt}{\int_0^{T_0} |U(t)|^2 \, dt}\right)^{1/2}$$

3.53

Substituting equation 3.53 into 3.52 where $\Lambda = \Lambda$ finds that

$$\frac{X}{D} = \frac{\int_0^{T_0} A(t)B(t)\cos\left(\beta(t+\tau) - \alpha(t) - \theta\right) \, dt}{\left(\int_0^{T_0} |V(t)|^2 \, dt \int_0^{T_0} |U(t)|^2 \, dt\right)^{1/2}}$$

3.54

$$\frac{X}{D} = \frac{\int_0^{T_0} A(t)B(t)\cos\left(\beta(t+\tau) - \alpha(t) - \theta\right) \, dt}{2 \left(\int_0^{T_0} |V(t)|^2 \, dt \int_0^{T_0} |U(t)|^2 \, dt\right)^{1/2}}$$

Substituting equation 3.54 into 2.43 the cancellation ratio is
Equation 3.55 provides the estimate of cancellation ratio which is independent of and optimized in terms of the interpulse amplitude variation.

3.2.2 Aircraft Emitter Phase Measurement Error

If the doppler shift of the emission from an airborne emitter is not constant in time, a phase stability measurement error will result. The magnitude of the error in the variance of the first phase difference is proportional to the frequency difference between adjacent pulses. Since the rate of change of doppler frequency is maximum nearer the closest point of approach (CPA), a worst case error will be obtained by evaluating the error about this critical point.

Consider the airborne emitter collection situation illustrated in Figure 14 where $V$ is the tangential velocity of an aircraft at point $P$ and $V_r$ is the aircraft radial velocity. For narrowband signals the doppler frequency $f_d$ can be shown to be

$$f_d = \frac{f_o V_t}{C((R/V)^2 + r^2)^{1/2}}$$  

3.56
CPA = Closest Point of Approach
V_r = Aircraft's Radial Velocity
V = Aircraft's Tangential Velocity
R = Antenna Distance to CPA

Figure 14  Airborne target collection geometry.
where $f_0$ is the center frequency of the emission

t is the time with $t=0$ at CPA

$C$ is the velocity of light.

The phase variation that would be observed in the signal during the observation time interval $t - t_o$ as the aircraft passes through the CPA is, therefore,

$$\theta(t_o) = 2\pi \int_{0}^{t_o} f_d \, dt = \frac{2\pi f_0 V}{C} \int_{0}^{t_o} \frac{t}{((R/V)^2 + t^2)^{1/2}} \, dt$$  \hspace{1cm} 3.57

Evaluating the rightmost integral in 3.57 results in

$$\theta(t_o) = \pm \frac{2\pi f_0 V}{C} \left[ \left( \frac{R}{V} \right)^2 + t_o^2 \right]^{1/2} + C_o \hspace{1cm} 3.58$$

where $C_o$ is an arbitrary constant that is found to be equal to $-\frac{R}{V} = \left( \frac{2\pi f_0 V}{C} \right)$ from the fact that $\theta(0) = 0$; therefore,

$$\theta(t_o) = \pm \frac{2\pi f_0 V}{C} \left\{ \left[ \left( \frac{R}{V} \right)^2 + t_o^2 \right]^{1/2} - \frac{R}{V} \right\}$$

$$= \pm \frac{2\pi f_0 R}{C} \left\{ \left[ 1 + \left( \frac{t_o V}{R} \right)^2 \right]^{1/2} - 1 \right\} \hspace{1cm} 3.59$$
Since $\frac{t_o V}{R} \ll 1$ for observations near CPA under most practical conditions, the approximation $(1 + X)^{1/2} = 1 + 0.5X$ can be used. Accordingly

$$\theta(t_o) = \begin{cases} \frac{\pi f_o V^2 t_o^2}{CR} & t_o > 0 \\ -\frac{\pi f_o V^2 t_o^2}{CR} & t_o < 0 \end{cases}$$

Since it is discrete samples of phase that will be dealt with, let $t_o = nT$ where $n = 0, 1, 2, 3 \ldots N$ and $T$ is the PRI. The phase variation function is, therefore,

$$\theta(nt) = \pm \frac{\pi f_o V^2 (nT)^2}{CR}$$

The first phase differences are then

$$\Delta\theta(nT) = \begin{cases} +\frac{\pi f_o V^2 (2n - 1)^2 T^2}{CR} & n \geq 0 \\ -\frac{\pi f_o V^2 (2n - 1)^2 T^2}{CR} & n < 0 \end{cases}$$

The sample mean of the first phase difference is then

$$\overline{\Delta\theta(nT)} = \begin{cases} \frac{1}{N} \sum_{n=-(N-1)/2}^{N-1} \Delta\theta(nT) & N \text{ odd} \\ \frac{1}{N} \sum_{n=-(N/2)}^{N/2} \Delta\theta[(n + 1/2)T] & N \text{ even} \end{cases}$$
If \( N >> 1 \) (greater than about 10) then

\[
\Delta \theta(nT) = \frac{\pi f_0 V^2 T^2 N}{CR}
\]

The variance of the first phase difference can then be calculated for large \( N \) by

\[
\sigma^2(\Delta \theta) = \frac{1}{N - 1} \sum_{n=\frac{N}{2}}^{N} \left( \Delta \theta(nT) - \overline{\Delta \theta(nT)} \right)^2
\]

\[
= \left( \frac{\pi f_0 V^2 T^2}{CR} \right)^2 \frac{1}{N - 1} \left\{ \sum_{n=1}^{N} ((2n - 1) - N)^2 + \sum_{n=1}^{N} ((2n + 1) - N)^2 \right\}
\]

\[
\approx \frac{1.2 \pi f_0 V^2 T^2 N}{CR} \quad \text{(radians}^2)\)
\]

Equation 3.64 represents a worst case situation where the aircraft is observed near CPA. This, in general, is unlikely since most observations will be made when the aircraft heading is very close to its reciprocal line of bearing. Nevertheless, to get a feel for the magnitude of this error, typical airborne radar parameters of \( f_0 = 9 \) GHz, \( N = 20 \), \( T = 1 \) msec, \( R = 100 \) mi and \( V = 500 \) MPH where used to evaluate 3.64 resulting in \( \sigma(\Delta \theta) \) = 1.5°. Since a worst case is represented, this error is certainly tolerable, but it is obvious that problems could develop from close-in collections situations.
3.2.3 Effects of Scanning Emitter

Any movement of the phase center of the radiating antenna about a center of motion is likely to have a radial velocity component which varies with time. Such a situation occurs when there is roll, pitch or yaw movement of an airborne radar platform or when an emitter is scanning.

Consider a scanning antenna as illustrated in Figure 15. The doppler frequency shift is

$$f_d = \left(\frac{f_0}{C}\right) V_r$$  

and from Figure 15 assuming $R >> r$ then

$$V_r = \dot{\theta} r \sin \theta$$

Assuming the phase center is rotating at a constant rate, then $\theta = \theta t$. Using equations 3.65 and 3.66, the phase at time $t_0$ as the antenna scans is

$$\theta(t_0) = 2\pi \int_0^{t_0} f_d \, dt = \frac{2\pi f_0 r}{C} \int_0^{t_0} \dot{\theta} \sin \dot{\theta} \, t \, dt$$

where $t_0 = 0$ corresponds to $\theta = 0$. 
Figure 15  Antenna scan geometry.

\[ V = \text{Tangential Velocity} \]
\[ R = \text{Antenna Distance to the Center of Motion} \]
\[ \dot{\theta} r = \text{Phase Difference} \]
\[ r = \text{Distance of Antenna's Phase Center to the Center of Motion} \]
Evaluating 3.67 results in

\[
\theta(t_o) = \begin{cases} 
+ \frac{2\pi f_o r}{C} (1 - \cos \theta t_o) & t_o > 0 \\
- \frac{2\pi f_o r}{C} (1 - \cos \theta t_o) & t_o < 0
\end{cases}
\]

Since sampling this function will be in effect at the PRI, let \( t_o = nT, n = 1,2,3, \ldots \). The phase difference is, therefore,

\[
\Delta \theta(nt) = \theta(nt) - \theta((n - 1)T)
\]

\[
= \pm \frac{2\pi f_o r}{C} \left[ (1 - \cos \theta nT) - (1 - \cos \theta (n - 1)T) \right]
\]

\[
= \pm \frac{2\pi f_o r}{C} (\cos \theta nT \cos \theta T + \sin \theta nT \sin \theta T - \cos \theta nT)
\]

For most practical radars, \( \theta T \) is small. Moreover, in measurements over a range within a few beamwidths of the boresight condition, it can safely be assumed that \( \theta nT \) is small. Hence, using the small angle approximations of \( \sin(x) \approx x \) and \( \cos(x) \approx (1 - x^2)^{1/2} = 1 \) equation 3.69 reduces to

\[
\Delta \theta(nt) = \frac{2\pi f_o r (\dot{\theta} T)^2}{C} \left| n \right|
\]
The sample mean of the first phase difference is, therefore,

$$\bar{\Delta \theta(nt)} = \frac{1}{N} \sum_{n=0}^{N/2} \Delta \theta(nt) = \frac{4 \pi f_{\alpha} r (\dot{\theta}T)^2}{CN} \sum_{n=1}^{N} n$$

$$= \frac{\pi f_{\alpha} r (\dot{\theta}T)^2 (N + 2)}{C}$$

$$= \frac{\pi f_{\alpha} r (\dot{\theta}T)^2 N}{C} \quad \text{(for large N)}$$

The sample variance of the first phase difference can, therefore, be written from equations 3.71 and 3.69

$$\delta^2(\Delta \theta) = \left[ \frac{\pi f_{\alpha} r (\dot{\theta}T)^2}{C} \right]^2 \frac{1}{N-1} \sum_{n=1}^{N/2} (2 | n | - N)^2$$

$$= \frac{1}{N-1} \left[ \frac{\pi f_{\alpha} r (\dot{\theta}T)^2}{C} \right]^2 \left[ N^2 + 2 \sum_{n=1}^{N/2} (2n - N)^2 \right]$$

Assuming N is greater than 10, all but the highest order terms in 3.72 which contain N can be neglected resulting in

$$\delta(\Delta \theta) = \frac{0.91 \pi f_{\alpha} r (\dot{\theta}T)^2 T^2 N}{Cr} \quad \text{(for moderate typical N)}$$
Since $\theta_f = V$, equation 3.73 is in the same form as equation 3.64 (the square root of the variance error caused by aircraft forward motion) except for the leading factor; and, since $C = \lambda f$, where $\lambda$ is the wavelength, the 3.73 can be written as

$$\hat{\sigma}(\Delta\theta) = 0.91\pi \left(\frac{L}{\lambda}\right) \hat{\dot{\theta}}^2 T^2 N$$

3.74

Making some assumptions about the emitter, 3.74 can be reduced to a more simple form. For many RADARS the moment arm length $r$ is nearly equal to the focal length $l$; and, assuming circular aperture antenna with a cosine amplitude taper, then the beamwidth $B$ is approximately given by

$$B = 1.2\lambda \approx \frac{1.2\lambda}{r} \text{ (radians)}$$

3.75

Again assuming a constant angular scan velocity and letting $N_B$ be the number of pulses per beamwidth, the result is

$$\hat{\dot{\theta}} = \frac{B}{N_B T}$$

3.76

Therefore, by substituting 3.75 and 3.76 into 3.74 and letting $N = N_B$, it is found that $\sigma(\Delta\theta)$ estimated over a beamwidth near boresight is

$$\hat{\sigma}(\Delta\theta) = \frac{\pi B}{N_B}$$

3.77

For a radar with $B = 2^\circ$ and $N_B = 20$ then $\hat{\sigma}(\Delta\theta) = 0.31^\circ$. 
A final comment regarding the bias in the estimate of $\sigma(\Delta \theta)$ due to both scan and aircraft motion effects needs to be made. Observe that the bias in $\sigma(\Delta \theta)$ estimation given in both equations 3.64 and 3.74 is directly proportional to $N$, the number of pulses over which the statistics are derived. In section 3.3 it is shown that, depending on the correlation properties of the noise, estimates of $\sigma(\Delta \theta)$ are also biased because of a small sample size in inverse proportions to $N$. This suggests that there is an optimum number of pulses from which to obtain the statistics, at least from bias error considerations. The problem with this reasoning is that, because the error due to scan is likely to be predominant, the optimum $N$ will be small and unfortunately the statistical estimates which are based on this smaller sample size will have a larger variance. Therefore, each of these factors must be carefully considered when choosing the best sample size for a particular processing requirement.

3.2.4 Propagation Path Effects

First of all, if multipath conditions exist and the path length difference is small such that the reflected signal coincides (at least partially) with the pulse, then the possibility for phase noise measurement error exists.\(^1\) Secondly, additional signal phase variations could conceivably be induced by air turbulence and/or precipitation along the path which would cause measurement error.

\(^1\) This usually corresponds to ground reflection at small grazing angles.
Consider the direct and reflected signal for a two path propagation model illustrated in Figure 16.

\[ E_D = A_D(\theta) e^{j(\omega t + \theta(t) + \theta_o)} \tag{3.78a} \]

\[ E_M = \rho_e(\psi,t) A_R(\theta) e^{j(\omega t + \theta(t) + \phi(\psi,t) + \theta_o)} \tag{3.78b} \]

where \( E_D \) and \( E_M \) are the complex representations of the direct and multipath signals respectively and

- \( A_D(\theta) \) is the amplitude of the direct component which is a function of the transmit antenna scan angle \( \theta \).

- \( A_R(\theta) \) is the amplitude of the multipath component before reflection which is a function of \( \theta \).

- \( \rho_e(\psi,t) \) is the magnitude of the effective reflection coefficient which includes the effects of surface scattering and is dependent on grazing angle, frequency and is, in general, dependent on time (e.g., sea surface reflection).

- \( \Phi(\psi,t) \) is the phase shift of the effective reflection coefficient and is, in general, dependent on time surface scattering, frequency, and grazing angle.
Figure 16  Geometry of two path propagation model.

$\psi =$ Grazing Angle
The received signal \( E_R \) is, therefore, the addition of equations 3.78a and 3.78b.

\[
E_R = A_D e^{j(\omega t + \theta(t) + \theta_0)} \left( 1 + \frac{A_R}{A_D} \rho_e(\psi,t) e^{j\Phi(\psi,t)} \right)
\]

\[
= \gamma A_D e^{j(\omega t + \theta(t) + \theta_0)}
\]

where \( \gamma \) is complex and equivalent to the factor in parentheses. Further, \( \gamma \) can be written as

\[
\gamma = \left[ \left( 1 + \frac{A_R}{A_D} \rho_e(\psi,t)\cos(\psi,t) \right)^2 + \left( \frac{A_R}{A_D} \sin(\psi,t) \right)^2 \right]^{1/2} e^{j\beta(\psi,t)}
\]

\[
3.80a
\]

where \( \beta(\psi,t) = \tan^{-1} \left[ \frac{\frac{A_R}{A_D} \rho_e(\psi,t)\sin(\psi,t)}{1 + \frac{A_R}{A_D} \rho_e(\psi,t)\cos(\psi,t)} \right] \)

\[
3.80b
\]

If the multipath is due to surface reflection and the grazing angle is much smaller than the Brewster angle (which is usually the case), then \( \Phi(\psi,t) \) is near 180°; and, hence, from 3.80b

\[
\beta(\psi,t) = -\frac{\frac{A_R}{A_D} \rho_e(\psi,t)\phi(\psi,t)}{1 + \frac{A_R}{A_D} \rho_e(\psi,t)}
\]

\[
3.81
\]
This is the phase term induced by multipath. If each of the factors in 3.81 did not vary as a function of time, then $\beta(\psi,t)$ would be constant (independent of time) for a given grazing angle and would not cause measurement error. Each of the factors in the denominator of 3.81 can be examined individually.

3.2.5 Zero IF Down Conversions (Direct Sampling)

Phase measurements made from RF pulse trains present little problem when the PRI is constant. Pulse-to-pulse phase difference measurements on the order of a few tenths of a degree have been achieved by using a linear regression over the intrapulse phase history. If the phase sampling interval is constant, small amounts of pulse-to-pulse PRI variation, commonly called jitter, can be tolerated if the PRI variation is bounded by one-half of a pulse width and the phase history is approximately constant. In this case the pulse phase samples are obtained at equal intervals (average PRI of pulse sample set). These pulse phase samples may be either single or from a linear regression of multiple phase samples depending on the extent of the PRI perturbation. In many practical field measurement situations, the PRI variation is not small (e.g., stagger) and the phase history is not a constant. In such a situation the pulse-to-pulse phase measurements may become severely limited by PRI jitter and stagger in both accuracy and interpretation.

Some phase noise measurements assume constant PRI which allows the phase noise (residual phase) to be extracted from the phase advance without conversion to zero IF. This can be achieved in effect by aliasing the modulo $2\pi$ phase advance obtained from a four quadrant arctangent of the ratio of the quadrature signal components after complex down conversion from IF by 25 MHz. Actual phase advance of the sinusoid within the pulse due to imperfect zero beat down conversion (which generally occurs) has a different
slopes than the apparent pulse-to-pulse phase advance. If the pulse phase samples are equidistantly spaced in time then there is no problem. However, if there is a sufficient amount of jitter to prevent equidistant sampling, the sample interval must depend on the individual pulse-to-pulse sample PRI; and, hence, the phase deviation from the linear phase advance will apparently have a component that is dependent on PRI with a constant of proportionality equal to the radian frequency of the down conversion.

Consider a signal \( y(t) \) of the form

\[
y(t) = A \sin[\Omega(t)]
\]

where \( A \) is a constant

\[
\Omega(t) = \omega_\alpha t + \theta(t) + \theta_o \quad \text{the phase}
\]

\[
\omega_\alpha = \text{radian frequency zero IF tuning error}
\]

\[
\theta(t) = \text{random phase}
\]

\[
\theta_o = \text{constant phase}
\]

This effectively represents the continuous sinusoid output of the RADAR oscillator that is pulsed, intercepted by the receiver, down converted to IF, sampled and digitally down converted to near zero IF (as determined by tuning). The fact that only portions (equal to a pulse duration) of \( y(t) \), which are sampled at a rate equal to the PRF, are not represented at this point is because the phase stability of \( y(t) \) is ultimately the object of measurement. The sampling can be introduced in the phase function \( \Omega(t) \) without loss of continuity.
The variance of the first phase difference $\sigma^2(\Delta \theta)$ of the random phase component is determined over an interval equal to the PRI (or average PRI for jittered or staggered pulsing). If the PRI is constant then

$$\text{VAR } \Omega(t) = E \left\{ \left( \Omega(t) - \Omega(t - T) - E(\Omega(t) - \Omega(t - T)) \right)^2 \right\}$$

$$= E \left\{ \left( \Delta \omega_0 T + \theta(t) - \theta(t - T) - \Delta \omega_0 T \right)^2 \right\}$$

$$= E \left\{ \left( \theta(t) - \theta(t - T) \right)^2 \right\}$$

$$= \sigma^2(\Delta \theta)$$  \hspace{1cm} (3.83)

Note that since the PRI is constant, $\omega_0 T$ is a constant (see equation 3.30) and the variance of the first difference of the random phase component can be evaluated without regard to imperfect zero IF down conversion (i.e., non-zero $\omega_o$ or $\omega_\alpha$).

Now let the PRI vary from pulse-to-pulse such that by selecting a phase sample of $y(t)$ corresponding to a fixed point in the pulse, the measurement is (see Figure 17 where the sampled case is illustrated)
Figure 17 Illustration of the effect of PRI variation on the direct sampling approach (random variations in phase, \( \theta(t) \), has been set to zero for the sake of illustration).
\[ \hat{\sigma}^2(\Delta \theta) = E \left\{ \left( \Omega(t) - \Omega(t - \Delta \theta) \right)^2 \right\} \]

\[ = E \left\{ \left( \omega_\alpha T + \omega_\varepsilon \varepsilon + \theta(t) - \theta(t - \Delta \theta) - \omega_\alpha T \right)^2 \right\} \]

3.84

where \( \varepsilon \) is a zero mean random variable representing variations in the PRI and \( \hat{\sigma}^2(\Delta \theta) \) is the estimate of \( \sigma^2(\Delta \theta) \). Therefore,

\[ \hat{\sigma}^2(\Delta \theta) = \omega_\varepsilon^2 E\left\{ \varepsilon^2 \right\} + E\left\{ \left( \theta(t) - \theta(t - \Delta \theta) \right)^2 \right\} \]

3.85

by assuming that \( \varepsilon \) and \( \theta(t) \) are uncorrelated.

The rightmost term of equation 3.85 can be evaluated on the condition that \( \varepsilon \) assumes a fixed value. The expected value of the result for variable \( \varepsilon \) can then be obtained. That is,

\[ E \left\{ E[\theta(t) - \theta(t - \Delta \theta) \mid \varepsilon] \right\} = E \left\{ 2R_\theta(0) - 2R_\theta(T + \varepsilon) \right\} \]

\[ = 2R_\theta(0) - 2 \int_{-\infty}^{\infty} R_\theta(T + \varepsilon) p(\varepsilon) \, d\varepsilon \]

3.86

where \( T \) is the lag interval and \( P(\varepsilon) \) is the probability density of \( \varepsilon \).
In order to evaluate the rightmost integral in equation 3.86, \( R(T + \varepsilon) \) is expanded in a Taylor series.

\[
R_\theta(T + \varepsilon) = \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n R_\theta(T)}{dT^n} \varepsilon^n
\]

3.87

and

\[
\int_{-\infty}^{\infty} R_\theta(T + \varepsilon) p(\varepsilon) d\varepsilon = \int_{-\infty}^{\infty} \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n R_\theta(T)}{dT^n} \varepsilon^n p(\varepsilon) d\varepsilon
\]

\[
= \sum_{n=0}^{\infty} \frac{1}{n!} \frac{d^n R_\theta(T)}{dT^n} \int_{-\infty}^{\infty} \varepsilon^n p(\varepsilon) d\varepsilon
\]

3.88

If \( T \) is the PRI then by substituting with equation 3.88, equation 3.85 becomes

\[
\hat{\theta}(\Delta \theta) = \omega_0^2 E \{ \varepsilon^2 \} + 2R_\theta(0)
\]

\[
-2 \sum_{n=1}^{\infty} \frac{1}{n!} \frac{d^n R_\theta(T)}{dT^n} \int_{-\infty}^{\infty} \varepsilon^n p(\varepsilon) d\varepsilon
\]

\[
= 2R_\theta(0) - 2R_\theta(T) - 2 \sum_{n=1}^{\infty} \frac{1}{n!} \frac{d^n R_\theta(T)}{dT^n} E \{ \varepsilon^n \}
\]

3.89

Note the \( E \{ \varepsilon^n \} \) is related to the characteristic function by

\[
E \{ \varepsilon^n \} = \left. \frac{1}{(-j)^n} \frac{d^n C(j\omega)}{d\omega^n} \right|_{\omega=0}
\]

3.90
where $C(j\omega)$ is the characteristic function defined by

$$C(j\omega) = \int_{-\infty}^{\infty} P(x) e^{j\omega x} \, dx$$

Substituting 3.90 into 3.89 results in

$$\hat{\sigma}(\Delta\theta) = \sigma_T^2(\Delta\theta) + (\omega_o)^2 \mathbb{E}\left\{ \varepsilon^2 \right\}$$

$$- 2 \sum_{n=1}^{\infty} \frac{1}{n!} \frac{d^n\sigma_\theta(T)}{dT^n} \frac{1}{(-j)^n} \frac{d^nC(j\omega)}{d\omega^n} \bigg|_{\omega=0}$$

where the subscript $T$ is used to preserve the dependence of the first difference phase variance on the mean PRI, $E(T + \varepsilon) = T$.

In order to be able to routinely handle the general case of stagger/jitter and to obtain the most accurate estimate of pulse phase based on the entire pulse phase history, it is advantageous to associate the pulse phase sample with a fixed point within the pulse (i.e., amplitude centroid). In this scheme, equation 3.92 shows the dependence of the measure $\sigma^2(\Delta\theta)$ on PRI variation and tuning error. In the event the PRI variance is zero ($E \{ \varepsilon \} = 0$) then $\sigma^2(\Delta\theta) = \sigma_T^2(\Delta\theta)$. If, however, $E \{ \varepsilon^2 \} = 0$ but $\omega_o = 0$, then there is still a bias error which depends on the functions $R_\theta(\tau)$ and $P(\varepsilon)$, although for continuous noise processes this error is negligible for the three noise types which could be predominant at typical PRI values (white phase noise and white frequency noise).
The noise model is

$$R_\theta(t) = a_0(\delta \tau) + a_2 |\tau|$$

3.93

where $a_0$ corresponds to the white phase noise term

$a_2$ corresponds to the white frequency noise term

For the white phase noise all derivatives of $R_\theta(\tau)$ are zero when evaluated at $\tau \neq 0$. For white frequency noise all derivatives of order greater than one are zero. Hence, for common noise processes the error contribution of the terms in equation 3.92 containing the derivatives of $R_\theta(\tau)$ is apparently negligible for continuous noise processes and again

$$\sigma^2(\Delta \theta) = \sigma_T^2(\Delta \theta).$$

If $\varepsilon$ assumes discrete values (i.e., an n position stagger) the probability density function is of the form

$$p(\varepsilon) = \sum_{n=1}^{N} a_n \delta(\varepsilon_n)$$

3.94

where

$$\varepsilon_n = T_n - T$$

$$T_n = \text{nth interval}$$

$$T = \sum_{n=1}^{N} T_n$$

$$N = \text{total number of intervals}$$
\[ a_n \text{ = relative frequency of occurrence of} \]
\[ T_n = \frac{\text{# of occurrences}}{N} \]
\[ \delta(\cdot) = \text{Dirac Delta Function} \]

Evaluating 3.85 after substituting 3.94 into 3.86 letting \( \tau = T \), we find that

\[
\hat{\sigma}^2 (\Delta \theta) = (\omega_o)^2 E\{ \epsilon^2 \} + 2 \left[ R_\theta(0) - \sum_{n=1}^{N} a_n R_\theta(T + \varepsilon_n) \right]
\]

\[
= (\omega_o)^2 E\{ \epsilon^2 \} + \frac{1}{N} \sum_{n=1}^{N} a_n \sigma_{\tau+\varepsilon_n}^2 (\Delta \theta)
\]

In either case, if \( E\{ \epsilon^2 \} \ll T \) and \( \frac{dR_\theta(\tau)}{d\tau} \) is small,

\[
\hat{\sigma}^2 (\Delta \theta) = (\omega_o)^2 E\{ \epsilon^2 \} + \sigma_\tau^2 (\Delta \theta)
\]

and the tuning error term is predominant.

---

1. Recall that \[ \int_{-\infty}^{\infty} \delta(x)f(x) \, dx = f(0) \]
Therefore, it was shown that in order to most accurately estimate $\sigma^2(\Delta \theta)$ when pulse jitter is present, $\omega_0$ must be minimized, that is, the signal must be zero beat as accurately as possible.

3.3 Phase Noise Measurement Bias Error

Let $\theta(t)$ be the random phase noise component where the signal is defined as

$$y(t) = A \sin(\omega t + \theta(t) + \theta_0)$$

and $A$, $\omega$ and $\theta_0$ are constant.

As previously discussed, only phase samples within each pulse of a pulse train of finite extent are available for analysis. That is, the packets of data spaced by the pulse repetition interval and being dealt with or data of the form

$$\begin{align*}
\theta(t_1), & \quad \theta(t_2), & \quad \theta(t_3) & \cdots \quad \theta(t_N) \\
\theta(t_1-T), & \quad \theta(t_2-T), & \quad \theta(t_3-T) & \cdots \quad \theta(t_N-T) \\
\theta(t_1-2T), & \quad \theta(t_2-2T), & \quad \theta(t_3-2T) & \cdots \quad \theta(t_N-2T) \\
\vdots & \quad \vdots & \quad \vdots & \quad \vdots \\
\theta(t_1-MT), & \quad \theta(t_2-MT), & \quad \theta(t_3-MT) & \cdots \quad \theta(t_N-MT)
\end{align*}$$
where \( N \) is the number of phase samples per pulse,

\( M \) is the number of pulses,

\( t_n \) is the intrapulse time referenced to the pulse envelope.

In the following discussion it will be assumed that the pulse repetition interval and pulse envelope are constant. Moreover, it will also be assumed that intrapulse phase modulation\(^1\), if it is present, is constant from pulse-to-pulse such that

\[
\sum_{n=1}^{N} \left( \theta(t_n + mT) - \theta(t_n + (m - 1)T) \right) = 0
\]

when there are absolutely no random variations due to master oscillator instabilities. The effect of lifting these assumptions will be discussed elsewhere.

Using intrapulse phase sample averaging over \( N \) intrapulse samples, the variance of the first phase difference is estimated from a pulse train of finite extent by

\[
\hat{\sigma}^2 (\Delta \theta) = \frac{1}{M-1} \sum_{k=1}^{M} \left[ \frac{1}{N} \sum_{n=1}^{N} \left( \theta(t_n - (k - 1)T) - \theta(t_n - kT) \right) \right]^2
\]

\[
- \left[ \frac{1}{M} \sum_{k=1}^{M} \left[ \frac{1}{N} \sum_{n=1}^{N} \left( \theta(t_n - (k - 1)T) - \theta(t_n - kT) \right) \right] \right]^2
\]

\(^1\) i.e., caused by system components.
Presumably a large number of estimates (>10) of this form will be averaged; and, therefore, the estimate is \( E \{ \sigma^2(\Delta \theta) \} \) where \( E \{ \cdot \} \) is the expected value (or ensemble average). In order to determine if and how this estimate differs from the true value, equation 3.100 is first expanded. From equation 3.100

\[
E \left\{ \frac{\hat{\sigma}_N^2}{\Delta \theta} \right\} = \frac{1}{M-1} \left[ \frac{E \{ A \}}{N^2} - \frac{E \{ B \}}{(NM)^2} \right]
\]

(3.101a)

where

\[
A = \sum_{m=1}^{M} \left[ \sum_{n=1}^{N} \left( \theta(t_n - (m-1)T) - \theta(t_n - mT) \right) \right]^2
\]

(3.101b)

\[
B = \left[ \sum_{m=1}^{N} \sum_{n=1}^{N} \left( \theta(t_n - (m-1)T) - \theta(t_n - mT) \right) \right]^2
\]

(3.101c)

and the subscript \( N \) indicates a finite sample size.

Expanding \( A \),

\[
A = \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[ \theta(t_i - (m-1)T)\theta(t_j - (m-1)T) + \theta(t_i - mT)\theta(t_j - mT) \right. \\
- \theta(t_i - (m-1)T)\theta(t_j - mT) - \theta(t_i - mT)\theta(t_j - (m-1)T) \left. \right]
\]

(3.102)
Taking the expected value and assuming $\theta(t)$ is from a stationary process,

$$\mathbb{E}\left\{ A \right\} = \sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} \left( R_\theta(t_i - t_j) + R_\theta(t_j - t_i) ight)$$

$$- R_\theta(t_i - t_j + T) - R_\theta(t_j - t_i + T)$$

where $R_\theta(\tau)$ is the autocorrelation function of $\theta(t)$.

Let $i - j = n$ and $t_i - t_j = n\Delta$ where $\Delta$ is the sampling interval. Therefore, since $R_\theta(\tau)$ is an even function

$$\mathbb{E}\left\{ A \right\} = \sum_{m=1}^{M} \sum_{n=1}^{N} \sum_{i=1}^{N} \sum_{j=1}^{1-N} \left( 2R_\theta(n\Delta) - R_\theta(n\Delta + T) - R_\theta(-n\Delta + T) \right)$$

Shown in Figure 18 is the region of summation for $i$ and $n$. Since the summand is independent of $i$, the order of summation can be interchanged and the sum over $i$ can be done first. Also, since $R_\theta(n\Delta)$ and the sum $R_\theta(n\Delta + T) + R_\theta(-n\Delta + T)$ are even functions of $n$, the contributions for $n < 0$ are equal to those for $n > 0$ and the term for $n = 0$ can be pulled out. Hence,

$$\mathbb{E}\left\{ A \right\} = \sum_{m=1}^{M} \left[ \sum_{n=1}^{N-1} 2 \left( 2R_\theta(n\Delta) - R_\theta(n\Delta + T) - R_\theta(-n\Delta + T) \right) \sum_{i=1}^{N} 1 ight]$$

$$+ 2R_\theta(0) - 2R_\theta(T) \sum_{i=1}^{N} 1$$

3.105
n = Sample Number
i = Summation Index

Figure 18 Region of summation for i and n (N=4).
Using the notation of Allen, performing the summations and noting the summand is independent of \( m \) [9],

\[
E \{ A \} = M \left[ \sum_{n=1}^{N-1} (N - n) \left( -2U(n\Delta) + U(n\Delta + T) + U(-n\Delta + T) \right) + Nu(T) \right]
\]

where \( U(T) = 2(R_{\theta}(0) - R_{\theta}(T)) \)

From equation 3.101c

\[
B = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{M} \sum_{l=1}^{M} \left( \theta(t_i - (k-1)T)\theta(t_j - (l-1)T) + \theta(t_i - kT)\theta(t_j - lT) \right)
\]

\[
- \theta(t_i - (k-1)T)\theta(t_i - lT) - \theta(t_i - kT)\theta(t_j - (l-1)T) \bigg) \bigg)
\]

Taking the expected value of \( B \)

\[
E \{ B \} = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k=1}^{M} \sum_{l=1}^{M} \left( R_{\theta}(t_i - t_j + (l-k)T) + R_{\theta}(t_i - t_j + (l-k)T) \right)
\]

\[
- R_{\theta}(t_i - t_j + (l-k+1)T) - R_{\theta}(t_i - t_j + (l-k-1)T) \bigg) \bigg)
\]

First let \( k - 1 = m, \quad i - j = n \quad \text{and} \quad t_i - t_j = n\Delta, \text{then} \)

\[
E \{ B \} = \sum_{i=1}^{N} \sum_{n=i}^{N} \sum_{k=1}^{M} \sum_{m=1}^{k-N} \left( 2R_{\theta}(n\Delta - mT) - R_{\theta}(n\Delta - (m-1)T) - R_{\theta}(n\Delta - (m+1)T) \right)
\]
Proceeding as before but for two sets of double summations,

\[ E \{ B \} = \sum_{i=n}^{N} \sum_{n=1}^{N} \left[ \sum_{m=1}^{M-1} (M-m)(4R_\theta(n\Delta-mT) - R_\theta(n\Delta-(m-1)T)) \right. \\
- R_\theta(n\Delta+(m+1)T) - R_\theta(n\Delta-(m+1)T) - R_\theta(n\Delta+(m-1)T) \\
+ M \left( 2R_\theta(n\Delta) - R_\theta(n\Delta+T) - R_\theta(n\Delta-T) \right) \right] \\
= \sum_{n=1}^{N-1} (N-n) \left[ \sum_{m=1}^{M-1} (M-m)(4R_\theta(n\Delta-mT) + 4R_\theta(n\Delta+mT)) \right. \\
- R_\theta(n\Delta-(m-1)T) - R_\theta(n\Delta-(m+1)T) \\
- R_\theta(n\Delta-(m+1)T) - R_\theta(n\Delta+(m+1)T) \\
- R_\theta(n\Delta+(m-1)T) - R_\theta(n\Delta+(m-1)T) \\
+ M \left( 4R_\theta(n\Delta) - R_\theta(n\Delta+T) - R_\theta(n\Delta-T) \right) \\
- R_\theta(n\Delta-T) - R_\theta(n\Delta+T) \right]
Combining like terms and again using the notation of Allen we find that,

\[ E \left\{ B \right\} = \sum_{n=1}^{N-1} \sum_{m=1}^{M-1} (N-n)(M-m) \left\{ \begin{align*} &-2U(n\Delta-mT) - 2U(n\Delta+mT) \\ &+ U(n\Delta-(m+1)T) + U(n\Delta+(m+1)T) \\ &+ U(n\Delta-(m+1)T) - M \left[ 2U(n\Delta) - U(n\Delta+T) - U(n\Delta-T) \right] \end{align*} \right\} \]

Substituting equations 3.106 and 3.113 into 3.101a, one can obtain the expected value of the estimate of \( \sigma^2(\Delta\theta) \) as a function of the number of pulses \( N \), the number of samples per pulse \( M \), and the correlation properties of the phase noise, \( U(T) \).
Since the signal can be assumed to be highly correlated over a pulse duration, then for the condition \( nA << T \) it can be shown that

\[
E \left\{ \sigma_N^2 (\Delta \theta) \right\} = \frac{1}{N - 1} \left[ NU(T) - \frac{1}{N} U(NT) \right]
\]

3.114

and since

\[
N \to \infty E \left\{ \sigma_N^2 (\Delta \theta) \right\} = E \left\{ \sigma_\infty^2 (\Delta \theta) \right\} = U(T)
\]

3.115

the ratio can be formed as

\[
\frac{E \left\{ \sigma_N^2 (\Delta \theta) \right\}}{E \left\{ \sigma_\infty^2 (\Delta \theta) \right\}} = \frac{1}{N - 1} \left[ N - \frac{1}{N} U(NT) \right]
\]

3.116

This is the bias function given by Allen [9] which relates the expected value of estimates made over a finite sample size with those made over an infinite number of samples.

Using the white phase and frequency noise model given in equation 3.93, Allen shows that

\[
\mu(T) = a(\mu) |T| + 2
\]

where \( a(\mu) \) is a constant for a given value of \( \mu \).

\( \mu = -1.0 \) for white noise frequency modulation,

\( \mu = -2.0 \) for white noise phase modulation.
Hence,

$$\frac{E\left\{ \sigma^2_N(\Delta \theta) \right\}}{E\left\{ \sigma^2_{\infty}(\Delta \theta) \right\}} = \frac{N}{N-1} \left[ 1 - \frac{1}{N} \right]$$

3.117

and the bias error is shown to be dependent on the number of sample pulses and the noise process. This dependency is illustrated in Figure 19 where the above ratio is plotted versus $N$ and $\mu$. Note that the bias error is negligible for $N > 20$.

Therefore, if the noise process is known, then one can easily compensate for finite sample sizes. The type of noise process can be determined by techniques developed by Allen [9] and others, but unfortunately they appear to be only marginally effective for analysis of pulsed signals.

3.4 Cancellation Ratio Limiting Factors
3.4.1 Limitation in the Cancellation Ratio Due to Frequency Noise

If one considers the limiting value of the cancellation ratio estimate due only to frequency instability from pulse-pulse, one must constrain all parameters of the adjacent pulses to be constant.

$$\hat{C} = 1 - \frac{2}{D} \int_0^{T_o} A^2(t) \cos(2\pi \Delta f t) \, dt$$

3.118
Figure 19  Illustration of the dependency of the bias on the number of pulses and noise process type.
Moreover, if one assumes \( A(t) \) to be a pulse of constant amplitude \( A \) with a duration of \( \tau \) then

\[
\hat{C} = 1 - \frac{1}{\tau} \int_0^t \cos(2\pi \Delta f t) \, dt \quad 3.119
\]

Taking the expected value of \( C \)

\[
E \{ \hat{C} \} = 1 - \frac{1}{\tau} \int_0^t E \{ \cos(2\pi \Delta f t) \} \, dt \quad 3.120
\]

where if one assumes \( \Delta f \) to be normally distributed then

\[
E \{ \cos(2\pi \Delta f t) \} = \frac{1}{(2\pi)^{1/2} \sigma(\Delta f)} \int_{-\infty}^{\infty} \cos(2\pi \Delta f t) \frac{e^{-\Delta f^2}}{\sigma^2(\Delta f)} \, d\Delta f
\]

\[
= e^{-2\pi^2 \sigma^2 (\Delta f)^2} \quad 3.121
\]

Substituting equation 3.121 into 3.120

\[
E \{ \hat{C} \} = 1 - \frac{1}{\tau} \int_0^t e^{-2\pi^2 \sigma^2 (\Delta f)^2} \, dt
\]

\[
= 1 - \frac{1}{(2\pi (\sigma(\Delta f) \tau)^2)^{1/2}} \int_0^{2\pi \sigma(\Delta f) \tau} \frac{e^{-x^2/2}}{(2\pi)^{1/2}} \, dx \quad 3.122
\]
This result is plotted in Figure 20 vs $\sigma(\Delta f)\tau^1$. If $\Delta f \tau$ is very small, equation 3.122 assumes an asymptotic form. In this case, the exponential in 3.122 can be approximated

$$e^{-x^2/2} = (1 - x^2/2).$$

Substituting this in 3.122 one finds

$$E\{\hat{C}\} = \frac{(2\pi\sigma(\Delta f)\tau)^2}{3} \quad \Delta f \tau \ll 1$$

3.123

which is quite accurate for $\Delta f \tau < 0.1$.

### 3.4.2 Limitation of the Cancellation Ratio Due to Phase Noise

The expression for the cancellation ratio where $\Delta \theta$ is the only random variable and $A(t)$ is defined as in the previous section is

$$\hat{C} = 1 - \frac{1}{\tau} \int_0^t \cos \Delta \theta \, dt = 1 - \cos \Delta \theta$$

3.124

Taking the expected value results in

$$E\{\hat{C}\} = 1 - E\{\cos \Delta \theta\}$$

3.125

---

1 By using the error function tables.
Figure 20  Cancellation ratio estimation limit vs. $\sigma(\Delta f)\tau$. 
If one assumes $\Delta \theta$ to be normally distributed then

$$E \left\{ \cos \Delta \theta \right\} = \frac{1}{\sqrt{2\pi}\sigma(\Delta \theta)} \int_{-\infty}^{\infty} \cos(\Delta \theta) e^{-\frac{(\Delta \theta)^2}{2\sigma^2(\Delta \theta)}} d\Delta \theta$$

$$= e^{-\frac{\sigma^2(\Delta \theta)}{2}}$$ \hspace{1cm} 3.126

Consequently,

$$E \left\{ \hat{\theta} \right\} = 1 - e^{-\frac{\sigma^2(\Delta \theta)}{2}}$$ \hspace{1cm} 3.127

This result is plotted in Figure 21.

### 3.4.3 Variance of the Cancellation Ratio Estimate Due to Pulse-to Pulse Frequency Variation

If the signal is not coherent, then it is likely that the pulse-to-pulse changes in frequency are the limiting factor. It will be assumed that $\Delta ft$ is small (i.e., $C$ is relatively large); and, therefore, if $\Delta f$ is the only "stray" parameter, then the small angle approximation can be used for the cosine in equation 3.119 (i.e., $\cos (2\pi \Delta ft) = 1 - (2\pi \Delta ft)^2 / 2$) and hence

$$\hat{\Delta} = \frac{(2\pi)^2 (\Delta f \tau)^2}{6}$$ \hspace{1cm} 3.128
Figure 21  Limitation of cancellation ratio vs. $\sigma(\Delta\theta)$. 
In Section 3.4 equation 3.123 it was found that for $\Delta f t < 0.1$

$$E\{\hat{e}\} = \frac{(2\pi)^2}{3} \left(\sigma(\Delta f)\tau\right)^2$$

3.129

The variance of the estimate caused by $\Delta f$ only, is therefore,

$$E\left\{\left(\hat{e} - E\{\hat{e}\}\right)^2\right\} = E\left\{\left(\frac{(2\pi)^2}{3} \left(\frac{\tau^2}{2} - \sigma^2(\Delta f)\right)\right)^2\right\}$$

$$\left(\frac{(2\pi)^2}{3} \tau^2\right)^2 E\left\{\frac{\Delta f^4}{4} + \sigma^4(\Delta f) - \sigma^2(\Delta f) \Delta f^2\right\}$$

$$\left(\frac{(2\pi)^2}{3} \tau^2\right)^2 E\left\{\frac{\Delta f^4}{4}\right\}$$

3.130

If $\Delta f$ is normally distributed and uncorrelated then

$$E\left\{\frac{\Delta f^4}{4}\right\} = \frac{3}{4} \sigma^4(\Delta f)$$

3.131

and equation 3.130 becomes

$$\text{VAR } \hat{e} = \frac{4\pi^4}{3} \left(\sigma(\Delta f)\tau\right)^4$$

3.132

If the final estimate is based on the $N$ pulse average of independent estimates, then the variance is

$$\text{VAR } \hat{e} = \frac{4\pi^4}{3N} \left(\sigma(\Delta f)\tau\right)^4$$

3.133
The ratio of the standard deviation of the estimate to the mean value is, therefore,

\[
\left( \frac{\text{VAR} \hat{C}}{E \{ \hat{C} \}} \right)^{1/2} = \frac{2 \pi \sqrt{3}}{(2\pi)^{3/2}} \frac{1}{\sqrt{N}} \frac{1}{\sqrt{N}}
\]

The 1 σ uncertainty in the estimate is given by

\[
E \{ \hat{C} \} \pm (\text{VAR} \hat{C})^{1/2} = E \{ \hat{C} \} \pm \frac{\sqrt{3}}{2\sqrt{N}} E \{ \hat{C} \}
\]

In decibels it becomes

\[
10 \log \left( E \{ \hat{C} \} \pm (\text{VAR} \hat{C})^{1/2} \right) = 10 \log \left( E \{ \hat{C} \} \left( 1 \pm \frac{\sqrt{3}}{2\sqrt{N}} \right) \right)
\]

Hence, the 1 σ measurement uncertainty bounds in the estimate of C are approximately

\[
E \{ \hat{C} \} + 10 \log \left( 1 - \frac{\sqrt{3}}{2\sqrt{N}} \right) < E \{ \hat{C} \} \leq E \{ \hat{C} \} + 10 \log \left( 1 + \frac{\sqrt{3}}{2\sqrt{N}} \right)
\]

The relative 1 σ bounds are plotted in Figure 22 vs N, the number of pulses averaged.
Figure 22 Positive and negative 1σ limits of cancellation ratio estimate due to pulse-to-pulse frequency variation.
Since the frequency deviations in the above are those associated with the emitter source, they are independent of the receiver noise; and, hence, the variance from the combined effect (equations 3.18 and 3.132) is, therefore,

\[
\text{VAR } \hat{C} = \frac{1}{N} \left\{ \frac{1}{(S/N)^2} \left( \frac{4\pi^4}{3} (\sigma f)^4 \right) \right\}
\]

3.138
CHAPTER 4  Algorithm Implementation

The COHERENCY program shall compute a phase coherency metric over a specified range of signal data. The program goes through numerous steps, however, before the phase coherency can be computed. The overall flow of signal processing operations in the program is as follows:

Optionally perform an automatic determination of a down conversion frequency to zero beat the signal. This entails computing an ensemble averaged power spectrum of all the pulses in the processing range and then finding the frequency of peak power using a curve fit and interpolation.

Then for each pulse in the processing range:

1. Convert the real input signal to analytic form via a time domain Hilbert transform.
2. Frequency down-convert the signal to near base band.
3. Compute the signal phase from the analytic signal.

Compute the coherency by:

1. Unwrapping the signal phase.
2. Fitting a straight line to the phase and removing any linear trend.
3. Computing the standard deviation of the first phase difference.

Each of the above algorithmic steps is explained more fully below.
4.1 Power Spectrum Computation

The program can perform an automatic determination of the down conversion frequency required to translate the signal to base band. Processing for this determination consists of computing the power spectrum of each pulse in the process range, ensemble averaging the spectra of all pulses, and finding the frequency of peak power. The power spectrum is computed using an in-place Fast Fourier Transform algorithm. A 2048 sample radix-2 transform is performed.

Given the complex output of the FFT, the power spectrum is computed and accumulated. For each of the N complex elements of the complex array, the square of the real part is added to the square of the imaginary part, and that sum is added to the spectrum accumulator.

FORMULA:

\[
\text{SPECTRUM ACCUMULATION} = \text{SPECTRUM ACCUMULATION} + (\text{REAL PART OF SPECTRUM})^2 + (\text{IMAGINARY PART OF SPECTRUM})^2
\]
4.1.1 Conversion Frequency Determination

The down conversion frequency FC is then selected by determining the frequency of maximum power. In order to increase the accuracy of the peak determination, an interpolation is performed by fitting a parabolic curve to the data in the vicinity of the peak. The maximum power point of the curve is then determined.

The squared term of the parabola is calculated as:

\[ A = 0.5 \times ([m-1] - 2.0 \times [m] + [m+1]) \]

The linear term of the parabola is calculated as:

\[ B = -0.5 \times [m-1] - [m+1] \]

where "m" is the bin of the frequency of maximum power.

The slope of the parabola is then:

\[ -\frac{B}{2.0 \times A} \]

The width of the frequency bins is:

\[ \text{BIN_WIDTH} = \frac{\text{SAMPLING_RATE}}{\text{BLOCK_SIZE}} \]

The IF estimate is:

\[ \text{IF} = ([m-1] + \text{DELTA}) \times \text{BIN_WIDTH} \]

The IF estimates are summed over the processing range and at completion of the computation, an average is taken based on the number of blocks that were processed.
4.2 Analytic Signal Computation

A Hilbert Transform is performed on the real input data to obtain a complex signal. Here, a time domain convolution is performed of the signal with a 15 point FIR (Finite Impulse Response) Hilbert transform. The convolution is performed between the (N+M-1)-element operand (Input Signal plus Initial Conditions) array and the M-element operator (Hilbert Transform) array. The result is a N-element complex array.

**FORMULA:**

\[
\text{COMPLEX ARRAY}(n) = \sum (\text{HILBERT TRANSFORM} \langle n - m \rangle \ast \text{INPUT SIGNAL}(m)) \\
\text{for } m = 0 \text{ to } N - 1 \\
\quad n = 0, 1, \ldots, N - 1
\]

In the above equation, indices are evaluated modulo N with the symbol \( \langle n \rangle \) representing the residue of n modulo N.

**WHERE:**

- \( N = \text{NUMBER OF INPUT SAMPLES} \)
- \( M = \text{LENGTH OF THE HILBERT TRANSFORM} \)
4.3 Frequency Conversion

A coherent frequency conversion of the signal is performed using the complex heterodyne method. First, an analytic conversion signal is generated at the conversion frequency FC. This is done by first constructing a phase ramp where the ramp increment is equal to the phase advance between samples of the signal corresponding to the down conversion frequency. A complex exponential is then computed as:

$$\exp(iX) = \cos(X) + i\sin(X)$$

Where $X = 2\pi FC \cdot \Delta T \cdot (m)$

$\Delta T$ is the sampling interval

$m$ is the sample index with respect to a fixed reference time used for all pulses treated

$m = \text{FIRST TIME_OF_ARRIVAL} - \text{REFERENCE_TOA}$

$X$ is computed modulo $2\pi$

Then since the Hilbert transform is: $A + iB$, and

The complex conjugate IF is: $C + iD$,

The real result is: $A \cdot C - B \cdot D$, and

The complex result is: $B \cdot C + A \cdot D$

The burst block is then frequency-translated to baseband by multiplying the complex signal data by the generated complex exponential.

For each successive calculation, the ramp function is incremented by $X$. 
4.4 Phase Computation

The signal phase is computed for each complex signal sample within the burst. For each of the N complex elements it divides the imaginary part by the real part. The phase is the arctangent of the result.

FORMULA:

\[
\text{PHASE ARRAY} = \arctangent \left( \frac{\text{IMAGINARY PART OF COMPLEX ARRAY}}{\text{REAL PART OF COMPLEX ARRAY}} \right)
\]
4.5 Phase Unwrapping

The modulo $2\pi$ sampled phase is unwrapped to create a smooth function. The phase unwrapping algorithm unmaps an array of phase angles over the range ($-\pi$, $\pi$). The algorithm takes an array of raw phase values and creates an array of continuous phase angles, i.e., removes the discontinuities caused by the arctangent operation, giving only the principal value of an angle.

**FORMULA:**

\[
\text{UNWRAPPED PHASE}(m) = \text{RAW PHASE}(m) + \text{COR}(m-1)
\]

Where:

\[
\text{COR}(m) = \begin{cases} 
\text{COR}(m-1) - 2 \times \pi & \text{if } \Delta > \text{Limit} \\
\text{COR}(m-1) + 2 \times \pi & \text{if } \Delta < -\text{Limit} \\
\text{COR}(m-1) & \text{if } -\text{Limit} \leq \Delta \leq \text{Limit}
\end{cases}
\]

\[
\text{COR}(0) = A(0), \text{COR}(0) = 0
\]

\[
\text{Limit} = 2 \times \pi - \text{Epsilon} \quad \text{recommended Epsilon = } \pi
\]

\[
\Delta = A(m) - A(m - 1)
\]

Also, a phase offset value is added to the phase samples as the data are unwrapped.
4.6 Fit a straight line to the phase and remove any linear trend

Processing begins by scanning the phase data to accumulate the sums used to compute the linear trend. The following summations are performed iteratively:

\[
\text{Time Of Arrival} = \text{Time Of Arrival} + \left( \text{Time Of Arrival}(i) - \text{Reference TOA} \right)
\]

\[
\text{TOA Sum} = \text{TOA Sum} + \text{Time Of Arrival}
\]

\[
\text{Phase Sum} = \text{Phase Sum} + \text{Unwrapped Phase}(i)
\]

The coefficients of a polynomial for a linear fit to the phase data are then computed by means of a least-squares-fit method.

\[
\text{Constant Term} = \text{Constant Term} + \text{Unwrapped Phase}(i) \times \text{Time Of Arrival}
\]

\[
\text{Linear Term} = \text{Linear Term} + (\text{Time Of Arrival})^2
\]

In the above steps the index "i" is varied from 1 to the number of samples.

Then, a Summation Matrix and A Summation vector are formed as follows:

\[
\text{Summation Matrix} = \begin{bmatrix}
\text{Linear Term} & \text{TOA Sum} \\
\text{TOA Sum} & \text{Number Of Samples}
\end{bmatrix}
\]

\[
\text{Summation Vector} = \begin{bmatrix}
\text{Constant Term} \\
\text{Phase Sum}
\end{bmatrix}
\]

Then the above matrices are combined to form an augmented matrix with the third column being the Summation Vector. The augmented matrix is then reduced in order to solve the set of simultaneous equations.
The linear independent equations of the least-squares-fit method are solved by Gauss-Jordan reduction of double precision matrices and vectors. The resultant output vector comprises the set of the line trend coefficients which are then used to remove the trend from the data. The phase trend is computed as follows:

\[ \text{Offset TOA} = \text{Time Of Arrival}(i) - \text{Reference TOA} \]
\[ \text{Phase Trend} = \text{Trend Coefficient}(2) + \text{Trend Coefficient}(1) \times \text{Offset TOA} \]

In the above steps, the index "i" is varied from 1 to the number of samples. Finally, the trend is removed from the unwrapped phase data.
4.7 Compute the standard deviation of the first phase difference

Processing begins by first locating the data to be processed. The phase data bounded by the processing start and end times are selected. The first sample with Time Of Arrival greater than the processing start time is found and denoted as First Sample. Then the first sample with Time Of Arrival less than the processing end time is found and denoted as Last Sample.

Then,

Previous Phase Value = Phase(First Sample)
Phase Difference = Phase(i) - Previous Phase Value
Sum Of Phase Difference = Sum Of Phase Difference
+ Phase Difference
Sum Of Phase Difference Squared =
Sum Of Phase Difference Squared + (Phase Difference)^2
Previous Phase Value = Phase(i)

The above four steps are repeated varying index i from (First Sample + 1) to Last Sample.

Then,

Number Of Samples = First Sample - Last Sample
And

Sum of Phase Difference Squared =
Sum of Phase Difference Squared / Number Of Samples
Sum of Phase Difference =

\[(\text{Sum of Phase Difference} / \text{Number Of Samples})^2\]

Finally

Standard Deviation Of First Phase Difference =

\[((\text{Sum Of First Phase Difference})^2 - \text{Sum Of First Phase Difference})^{1/2}\]
LIMITATIONS

The implementation of the algorithms presented in this paper provided satisfactory results for signal-to-noise ratios that ranged between 5dB and 50dB. However, for levels of SNR lower than 5dB, there were several problems trying to apply them to pulsed signals. Because of the short duration of the pulses (typically 1 to 10µs), the spectrum is rather wide (on the order of a few MHz). The separation of the PRF lines is typically less than 1 kHz. Thus, there may be several hundred spectral lines in the vicinity of the carrier line and all of them have nearly the same amplitude. Spectral lines other than the carrier line are affected by PRI jitter as RF instability. Hence, it is necessary to locate the carrier line if there is significant PRI jitter. The portion of the spectrum of interest extends on each side of the carrier line out to one half of the PRF. RF disturbances which occur more rapidly than half the PRF cannot be distinguished from slower ones due to aliasing. This is caused by the fact that the RF carrier is, in effect, sampled once every pulse interval.

Also, when measuring the phase spectral density stability measure for pulsed signals, it is first necessary to use high quality filtering and down conversion so that a single line of the pulsed signal spectrum can be isolated for analysis through CW techniques. If the pulsing function itself is stable, any of the lines near the center of the spectrum will be acceptable. However, if the pulse interval jitters, only the center line of the spectrum will be acceptable. This is because the pulsing instabilities contribute to the spectral width of each line in proportion to its separation from the carrier frequency. Two practical problems when measuring the phase noise spectra of pulsed microwave signals are (1) mixing down to baseband coherently without any aliasing or spectrum fold over and
(2) locating the center line of the spectrum. Once this is done, the resulting single line can be analyzed in the same way as a CW signal. Precise tuning after successive mixing operations may be necessary to accomplish this down conversion.

The question of how well the frequency stability can be determined from noisy, time limited, and/or band limited data must be answered. For this reason, it is important to specify whether or not the stability estimate is limited by the SNR and/or the measurement bandwidth. Based on the relationships among the various measures, it is clear that the available SNR and bandwidth are important no matter how the stability is measured.
CONCLUSION

This paper dealt with methods that provide the capability of signal classification by extracting information about the collected radar waveform. To this end, the devised algorithms provide information about the signal received from the radar set by extracting the signal's amplitude, phase, and frequency and subsequently deriving the short-term RF stability (Coherence). Since the software is capable of analyzing a series of pulses, or even continuous data presented to it, an inter-pulse history can be maintained which will enable the analyst to characterize and classify the signal. The provision of this significant information can be used in the identification process, especially in a military situation. The resulting waveform signatures can then be stored in a data-base which will allow for comparison and verification of friend or foe in a tactical situation.
Overview

The COHERENCY routine is the main routine for the COHERENCY program. Processing in the routine essentially consists of first calling the menu display and handling routines to get the user's set up parameters. Next, the routine calls the initialization routine to verify the user inputs. If set up errors are detected, an error message is displayed and control returned to the menu routines. After successfully completing set up, the routine performs the automatic down conversion frequency determination if requested and extracts and conditions a series of phase samples from the input signal data file. The phase data is then displayed and the user given the option of performing a series of graphic interactive functions. The program loops through this option until the user requests a return to the program's main menu. At this point, the overall program processing is restarted.

Purpose
PROGRAM COHERENCY

C Collect user entered parameters
C Perform initialization and parameter verification.

CALL COHO_INIT( L_INIT_ERROR )

C If automatic intermediate frequency calculation is selected, do the necessary processing
C for the down conversion frequency.

IF ( CENTER_FREQUENCY .EQ. 0.0 ) THEN

CALL SPECTRUM_PEAK( ERROR_FLAG )

ELSE

CENTER_FREQUENCY = IF_FREQ

IF_FREQ = CENTER_FREQUENCY

END IF

C Perform the required signal processing which consists of converting the real data to
C complex, frequency down-conversion, and phase computation and unwrapping.

CALL COHO_SP(L_COND_PHASE,L_RAW_PHASE,
1 L_TOA_ARRAY,NUM_SAMPLES,
2 NUM_ELEMENTS,ERROR_FLAG)

C Generate the Phase versus TOA display.

CALL PHASE_TOA( L_COND_PHASE,
1 L_TOA_ARRAY, L_NUM_PLOT,
2 L_SCREEN_XMIN, L_SCREEN_XMAX,
3 L_SCREEN_YMIN, L_SCREEN_YMAX)

C Generate the signal phase vs. pulse no. display.

IF ( RAW_PLOT_SELECTION(1:7) .EQ. 'ENABLED' ) THEN

L_NUM_PLOT = NUM_ELEMENTS * NUM_SAMPLES

CALL PHASE_SAMPLE(L_RAW_PHASE, L_NUM_PLOT)

END IF

C Compute and display the phase statistics.

CALL PHASE_STATS(L_COND_PHASE,L_TOA_ARRAY,NUM_ELEMENTS)

STOP
END
OVERVIEW

The COHO_INIT routine performs initialization for the COHERENCY program. This includes initializing the data input system, verifying the input data type, initializing the interrupt handling system, and setting various processing parameters. Any other parameter initialization necessary is also performed within this routine.

SUBROUTINE COHO_INIT (P_INIT_ERROR)

LOGICAL*4 P_INIT_ERROR
LOGICAL*4 L_EXIT

C Controls program execution.

INTEGER*4 L_I

C General index variable.

INTEGER*4 L_FIRST_PROC

C Used to determine how to read the input bursts.

INTEGER*4 L_TIME_CHECK_ERROR

C Flag indicating that an error occurred in TIMECHECK module.

REAL*8 L_T1

C Temporary variable holding the PROCESSING START TIME value.

REAL*8 L_T2

C Temporary variable holding the PROCESSING END TIME value.

C Verify that a file name parameter has been entered.
C Get user entered parameters from MENU common block.

C Get file header information
C Set the range for the processing start/end times.

C Load the coefficients of the optimized Hilbert transform.
C Set up memory for the real to complex data conversion.

CALL R2CFIL_INIT()

DO 110 L_I = 1, HILBERT_LENGTH - 1
    INITIAL_CONDITION(L_I) = 0.0
110 CONTINUE

C Call TIMECHECK with FILE_START_TOA, FILE_END_TOA as inputs,
C PROCESSING_START_TIME, PROCESSING_END_TIME as inputs/outputs, and
C TIME_CHECK_ERROR as output, to make sure processing time range falls within time
C range of data file.

L_T1 = FILE_START_TOA + PROCESSING_START_TIME
L_T2 = FILE_START_TOA + PROCESSING_END_TIME

CALL TIMECHECK (L_T1, L_T2,
    1    FILE_START_TOA, FILE_END_TOA,
    2    L_TIME_CHECK_ERROR)

C Set the frequency conversion reference time.

REF_TOA = PROCESSING_START_TIME

C Determine the number of samples to process and where to start processing

IF ( RAW_PLOT_SELECTION(1:7) .EQ. 'ENABLED' ) THEN
    NUM_SAMPLES = 11
    FIRST_SAMPLE = INT(SAMPLING_OFFSET*SAMPLE_RATE)-5
ELSE
    NUM_SAMPLES = 1
    FIRST_SAMPLE = INT(SAMPLING_OFFSET*SAMPLE_RATE)
END IF

IF (FIRST_SAMPLE .LT. 1) FIRST_SAMPLE = 1

C Determine how to read input bursts so that all the data needed ends up in a single buffer

    HALF_FLAG = .FALSE.
    READ_COUNT = 1
    L_EXIT = .FALSE.
    L_FIRST_PROC = FIRST_SAMPLE - FILTER_DELAY
LAST_SAMPLE = FIRST_SAMPLE + NUM_SAMPLES - 1 + FILTER_DELAY

120 IF ( L_EXIT ) GOTO 130

C First case is where all the data is in the first 2K buffer read

IF ((L_FIRST_PROC .LT. I_2K) .AND.
    (LAST_SAMPLE .LE. I_2K)) THEN
    L_EXIT = .TRUE.
ENDIF

C Case where neither is in the 2K buffer

IF ((L_FIRST_PROC .GT. I_2K) .AND.
    (LAST_SAMPLE .GT. I_2K)) THEN
    READ_COUNT = READ_COUNT + 1
    L_FIRST_PROC = L_FIRST_PROC - I_2K
    FIRST_SAMPLE = FIRST_SAMPLE - I_2K
    LAST_SAMPLE = LAST_SAMPLE - I_2K
ENDIF

C Case where the ends straddle a 2K block

IF ((L_FIRST_PROC .LE. I_2K) .AND.
    (LAST_SAMPLE .GT. I_2K)) THEN
    HALF_FLAG = .TRUE.
    READ_COUNT = READ_COUNT + 1
    L_FIRST_PROC = L_FIRST_PROC - I_1K
    FIRST_SAMPLE = FIRST_SAMPLE - I_1K
    LAST_SAMPLE = LAST_SAMPLE - I_1K
ENDIF

GOTO 120

130 CONTINUE

ENDIF

RETURN

END
OVERVIEW

NAME R2CFIL_INIT

PURPOSE
The R2CFIL_INIT routine sets up the coefficients of the Hilbert transform for subsequent use by the R2C_FILTER routine. The coefficients are for an optimized Hilbert transform. Currently, a transform length of 31 is used.

SUBROUTINE R2CFIL_INIT()

C Initialize values
C 31 point transform, fl = 0.05, error = 0.0026800

HIL_COEF(1) = -0.0041956
HIL_COEF(3) = -0.0092821
HIL_COEF(5) = -0.0188358
HIL_COEF(7) = -0.0344010
HIL_COEF(9) = -0.0595516
HIL_COEF(11) = -0.1030376
HIL_COEF(13) = -.1968314
HIL_COEF(15) = -0.6313536
HIL_COEF(17) = -HIL_COEF(15)
HIL_COEF(19) = -HIL_COEF(13)
HIL_COEF(21) = -HIL_COEF(11)
HIL_COEF(23) = -HIL_COEF(9)
HIL_COEF(25) = -HIL_COEF(7)
HIL_COEF(27) = -HIL_COEF(5)
HIL_COEF(29) = -HIL_COEF(3)
HIL_COEF(31) = -HIL_COEF(1)

HILBERT_LENGTH = 31
FILTER_DELAY = HILBERT_LENGTH / 2

RETURN
END
OVERVIEW

NAME TIMECHECK

PURPOSE

This routine compares the user input start and stop times to the start and stop times in the input data file. An error message is output if the user input time range is totally outside the data file time range or if the input start time is greater than the input stop time.

SUBROUTINE TIMECHECK

(P_PSTIME,P_PETIME,P_STIME,P_ETIME,P_STATUS)

REAL*8 P_PSTIME
C Processing start time (secs)

REAL*8 P_PETIME
C Processing end time (secs)

REAL*8 P_STIME
C Data file start time (secs)

REAL*8 P_ETIME
C Data file end time (secs)

INTEGER*4 P_STATUS
C Error return status: (1 = normal, 0 = error)

P_STATUS = 1

IF ( P_PSTIME .GT. P_PETIME ) THEN
C "Input start time greater than input end time"
  P_STATUS = 0
ENDIF

IF ( P_PETIME .EQ. P_PSTIME ) THEN
C "Input stop time equal to input start time"
\texttt{P\_STATUS = 0}
\texttt{ENDIF}
IF ( P_PSTIME .GT. P_ETIME ) THEN
  C "Input start time greater than data stop time"
  P_STATUS = 0
ENDIF

IF ( P_PETIME .LT. P_STIME ) THEN
  P_PETIME = P_STIME + P_PETIME
  C "Input stop time less than data start time"
  P_STATUS = 1
ENDIF

IF ( (P_PSTIME .LT. P_STIME) .AND. (P_STATUS .EQ. 1) ) THEN
  P_PSTIME = P_STIME
  C "Input start time less than data start time"
ENDIF

IF ( (P_PETIME .GT. P_ETIME) .AND. (P_STATUS .EQ. 1) ) THEN
  P_PETIME = P_ETIME
  C "Input stop time greater than data stop time"
ENDIF

RETURN
END
OVERVIEW

NAME SPECTRUM_PEAK

PURPOSE

The SPECTRUM_PEAK routine is used to compute the intermediate frequency (IF) of the input signal when the user requests automatic determination. The IF value is used to down-convert the signal to baseband. The power spectrum of all the pulses in the processing range are ensemble averaged. The IF value is derived by parabolic interpolation of the averaged spectral peak. The routine dynamically determines the transform size used in the computation based on the first pulse in the input. The length of the first pulse is determined. If greater than 2K, then a 2K transform is used. If less than 2K, the transform is set to the first power of 2 greater than the number of samples in the first burst. Processing in the routine continues until either the processing end time is reached or the maximum number of bursts have been processed.

SUBROUTINE SPECTRUM_PEAK( P_ERR )

INTEGER*4 P_ERR

C Flag indicating that some error has occurred and that processing should terminate.

INTEGER*4 LOG2

C Determines the transform size.

LOGICAL*4 L_EOB

C The READ_WB routine sets this logical variable to TRUE when end of burst is reached.

LOGICAL*4 L_EOD

C The READ_WB routine sets this logical variable to TRUE when end of file is reached.

LOGICAL*4 L_EXIT

C Controls program execution.

INTEGER*4 L_M

C LOG2(L_SAMPLES_READ). Power of two used to determine the transform size.
INTEGER*4 L_BIN
C The number of the power spectrum bin that contains the maximum value.

INTEGER*4 L_I, L_IND
C Loop counters.

INTEGER*4 L_NUM_BLOCK
C Holds the value of the number of processed blocks.

INTEGER*4 L_NEXT_BURST
C Flag specifying whether the reader should position to the next burst or not: 0 = FALSE, 1 = TRUE

INTEGER*4 L_BURST_COUNT
C Number of processed bursts.

INTEGER*4 L_NUM_SAMPLES
C The size of the data buffer to read from the input Burst A/D file.

INTEGER*4 L_SAMPLES_READ
C The actual number of samples read by the READ_WB routine.

INTEGER*4 L_PROCESS_BLOCK_SIZE
C Process block size which is equal to the first power of 2 greater than samples read.

REAL*4 L_ACOEF
C Coefficient of the squared term in the equation for a parabola.

REAL*4 L_BCOEF
C Coefficient of the linear term in the equation for a parabola.

REAL*4 L_BIN_WIDTH
C Width of the frequency bins.

REAL*4 L_CCOEF
C Coefficient of the constant term in the equation for a parabola.

REAL*4   L_DELTA_BIN

C Estimated peak offset.

REAL*4   L_SPECTRUM_PEAK(3)

C This array contains the peak of the accumulated power spectrum.

REAL*4   L_IQ_ARRAY(I_4K+10)

C Array containing the in-phase & quadrature data.

REAL*4   L_POWER_SPECTRUM(I_4K+10)

C This array contains the accumulated power spectrum.

REAL*4   L_SPEC_SUM

C This is the sum of the IF estimates for each block processed.

REAL*8   L_END_TIME

C Processing ends when this value is exceeded.

REAL*8   L_FIRST_TOA

C Value of the first TOA in a buffer that the READ_WB routine read.

C Initialize flags and counters.

P_ERR = 0
L_EOB = .FALSE.
L_EOD = .FALSE.
L_EXIT = .FALSE.
L_NUM_BLOCK = 0
L_NEXT_BURST = 0
L_BURST_COUNT = 0
L_SAMPLES_READ = 0
L_NUM_SAMPLES = I_BLOCK_SIZE

C Initialize the power spectrum accumulator.

DO 100 L_I = 1, L_NUM_SAMPLES
   L_POWER_SPECTRUM(L_I) = 0.0
   L_IQ_ARRAY(L_I) = 0.0
100 CONTINUE
C Position file to the processing start time and read the first 2K block of data or first burst.

\[
\text{L_FIRST_TOA} = \text{PROCESSING_START_TIME} \\
\text{CALL POSITION(L_FIRST_TOA, STATUS)} \\
\text{CALL READ_WB(L_NUM_SAMPLES,} \\
1 \quad \text{L_IQ_ARRAY,} \\
2 \quad \text{L_SAMPLES_READ,} \\
3 \quad \text{L_NEXT_BURST,} \\
4 \quad \text{L_EOB,} \\
5 \quad \text{L_EOD,} \\
6 \quad \text{L_FIRST_TOA,} \\
7 \quad \text{STATUS)}
\]

C Set process block size

\[
\text{IF (L_SAMPLES_READ .LT. I_BLOCK_SIZE) THEN} \\
\text{C SET PROCESS_BLOCK_SIZE TO THE FIRST POWER OF 2 GREATER THAN C SAMPLES_READ} \\
\text{L_M = LOG2(L_SAMPLES_READ)} \\
\text{IF (L_M .LT. 9) L_M = 9} \\
\text{L_PROCESS_BLOCK_SIZE = 2 ** L_M} \\
\text{ELSE} \\
\text{L_M = LOG2(I_BLOCK_SIZE)} \\
\text{L_PROCESS_BLOCK_SIZE = I_BLOCK_SIZE} \\
\text{ENDIF} \\
\text{L_END_TIME = L_FIRST_TOA} \\
\text{L_BIN_WIDTH = SAMPLE_RATE / L_PROCESS_BLOCK_SIZE} \\
\text{110 IF (L_EXIT .OR. (L_END_TIME .GE. PROCESSING_END_TIME)) GOTO 111}
\]

C Increment the burst count and check for completion.

\[
\text{L_NUM_BLOCK = L_NUM_BLOCK + 1} \\
\text{IF (L_EOB) THEN} \\
\text{L_BURST_COUNT = L_BURST_COUNT + 1} \\
\text{ENDIF}
\]

C Perform a real-to-complex forward FFT on the burst data and scale.
CALL RFFT(L_IQ_ARRAY, L_PROCESS_BLOCK_SIZE, L_M)

DO 120 L_I = 1, L_PROCESS_BLOCK_SIZE
    L_IQ_ARRAY(L_I) = L_IQ_ARRAY(L_I) / FLOAT(2 * L_PROCESS_BLOCK_SIZE)
120  CONTINUE

C Compute the power spectrum of burst data and add to the accumulated power spectrum.

L_IND = 1
L_IQ_ARRAY(L_PROCESS_BLOCK_SIZE) = L_IQ_ARRAY(2)
L_IQ_ARRAY(2) = 0.0

DO 130 L_I = 1, L_PROCESS_BLOCK_SIZE, 2
    L_POWER_SPECTRUM(L_IND) = L_IQ_ARRAY(L_I)**2 + L_IQ_ARRAY(L_I+1)**2
    L_IND = L_IND + 1
130  CONTINUE

C Locate the bin containing the largest value of the power spectrum and then retrieve the
C largest value, its location, and the value of the closest bin on each side of the bin
C containing the largest value.

L_BIN = 1
L_SPECTRUM_PEAK(2) = L_POWER_SPECTRUM(L_BIN)

DO 140 L_I = 1, L_PROCESS_BLOCK_SIZE/2
    IF (L_POWER_SPECTRUM(L_I) .GT. L_SPECTRUM_PEAK(2)) THEN
        L_BIN = L_I
        L_SPECTRUM_PEAK(2) = L_POWER_SPECTRUM(L_I)
        L_SPECTRUM_PEAK(1) = L_POWER_SPECTRUM(L_I - 1)
        L_SPECTRUM_PEAK(3) = L_POWER_SPECTRUM(L_I + 1)
    ENDIF
140  CONTINUE

C Perform the parabolic interpolation to refine the estimate of where the power spectrum C
peak is actually located.

C Initialize the parabola variables for the interpolation.

L_ACOEF = 0.5 * (L_SPECTRUM_PEAK(1) - 2.0 * L_SPECTRUM_PEAK(2) + L_SPECTRUM_PEAK(3))
L_BCOEF = -0.5 * (L_SPECTRUM_PEAK(1) - L_SPECTRUM_PEAK(3))
L_CCOEF = L_SPECTRUM_PEAK(2)
IF (L_ACOEF .NE. 0.0) THEN
    L_DELTA_BIN = -L_BCOEF / (2.0 * L_ACOEF)
ENDIF

C Sum the IF estimates for later averaging.

    L_SPEC_SUM = L_SPEC_SUM + (FLOAT(L_BIN - 1) + L_DELTA_BIN) * L_BIN_WIDTH

IF (L_EOD) THEN
    PROCESSING_END_TIME = L_END_TIME
    L_EXIT = .TRUE.
    P_ERR = 1
ELSE
    C Zero memory where next burst will be placed. Read next process block size of samples.

    DO 150 L_I = 1, L_PROCESS_BLOCK_SIZE
        L_IQ_ARRAY(L_I) = 0.0
    150 CONTINUE
    CALL READ_WB(L_NUM_SAMPLES,
                L_IQ_ARRAY,
                L_SAMPLES_READ,
                L_NEXT_BURST,
                L_EOB,
                L_EOD,
                L_FIRST_TOA,
                STATUS)
ENDIF

    L_END_TIME = L_FIRST_TOA + L_SAMPLES_READ/SAMPLE_RATE
GOTO 110
111 CONTINUE

C Average the sum of IF estimates for final IF.

    IF (L_NUM_BLOCK .GT. 0) THEN
        IF_FREQ = L_SPEC_SUM/L_NUM_BLOCK
    ENDIF
RETURN
END
FUNCTION LOG2(P_N)

INTEGER*4  P_N
INTEGER*4  LOG2
INTEGER*4  L_I

LOG2 = 0
L_I = 1

400  LOG2 = LOG2 + 1
     L_I = L_I * 2
     IF (L_I LT. P_N) GOTO 400
RETURN
END
SUBROUTINE RFFT(REAL_ARRAY,FFTSIZE,POW_2)
REAL*4 REAL_ARRAY(*)
INTEGER*4 FFTSIZE, POW_2

C Forward FFT:
CALL APSFAST(REAL_ARRAY,POW_2)
REAL_ARRAY(2)=REAL_ARRAY(FFTSIZE+1)
RETURN
END

SUBROUTINE APSFAST(B,M)
REAL*4 B(*), T, PI2, PII, S22, PI8, C22, P7, P7TWO
INTEGER*4 IW, N, M, N4POW, NN, INV, IT
COMMON /CONS/ PH, P7, P7TWO, C22, S22, PI2

IW = 1

PII = 4.*ATAN(1.)
PI8 = PII/8.
P7 = 1./SQRT(2.)
P7TWO = 2.*P7
C22 = COS(PI8)
S22 = SIN(PI8)
PI2 = 2.*PII
N=2**M
20 N4POW = M/2

C DO A RADIX 2 ITERATION FIRST IF ONE IS REQUIRED.
IF (M-N4POW*2) 40, 40, 30
30 NN = 2
INV = N/NN
CALL APSFR2TR(INV, B(1), B(INV+1))
GO TO 50
40 NN = 1

C PERFORM RADIX 4 ITERATIONS.
50 IF (N4POW.EQ.0) GO TO 70
DO 60 IT=1, N4POW
   NN = NN*4
   INV = N/NN
   CALL APSFR4TR(INV, NN, B(1), B(INV+1), B(2*INV+1), B(3*INV+1),
                 *
                 B(1), B(INV+1), B(2*INV+1), B(3*INV+1))
60 CONTINUE
C PERFORM IN-PLACE REORDERING.

70 CALL APSFORD1(B,M)
CALL APSFORD2(B,M)
T = B(2)
B(2) = 0.
B(N+1) = T
B(N+2) = 0.
DO 80 IT=4, N, 2
   B(IT) = -B(IT)
80 CONTINUE
RETURN
END

SUBROUTINE APSFORD1(B,M)
REAL*4 B(*), T
INTEGER*4 K, KL, N, M, J

K = 4
KL = 2
N = 2**M
DO 40 J=4, N, 2
   IF (K-J) 20, 20, 10
10 T = B(J)
   B(J) = B(K)
   B(K) = T
20 K = K - 2
   IF (K - KL) 30, 30, 40
30 K = 2*J
KL = J
40 CONTINUE
RETURN
END

SUBROUTINE APSFORD2(B,M)
REAL*4 B(*), T
INTEGER*4 L(15), N, M, K, L1, L2, L3, L4, L5, L6, L7, L8, IJ
INTEGER*4 L9, L10, L11, L12, L13, L14, L15, J1, J2, J3, J4, JI
INTEGER*4 J5, J6, J7, J8, J9, J10, J11, J12, J13, J14

EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
   (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
   (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
   (L1,L(15))

N = 2**M
L(1) = N
DO 10 K=2, M
\begin{verbatim}
L(K) = L(K-1)/2
10 CONTINUE
DO 20 K=M, 14
   L(K+1) = 2
20 CONTINUE

U = 2
DO 40 J1=2, L1, 2
   DO 40 J2=J1, L2, L1
   DO 40 J3=J2, L3, L2
   DO 40 J4=J3, L4, L3
   DO 40 J5=J4, L5, L4
   DO 40 J6=J5, L6, L5
   DO 40 J7=J6, L7, L6
   DO 40 J8=J7, L8, L7
   DO 40 J9=J8, L9, L8
   DO 40 J10=J9, L10, L9
   DO 40 J11=J10, L11, L10
   DO 40 J12=J11, L12, L11
   DO 40 J13=J12, L13, L12
   DO 40 J14=J13, L14, L13
   DO 40 J15=J14, L15, L14
   IF (IJ-JI) 30, 40, 40
30 T = B(U-1)
   B(U-1) = B(JI-1)
   B(JI-1) = T
   T = B(U)
   B(U) = B(JI)
   B(JI) = T
40 IJ = IJ + 2
RETURN
END

SUBROUTINE APSFR2TR(INV, B0, B1)
REAL*4       B0(*), B1(*), T
INTEGER*4    K, INV

DO 10 K=1, INV
   T = B0(K) + B1(K)
   B1(K) = B0(K) - B1(K)
   B0(K) = T
10 CONTINUE
RETURN
END

SUBROUTINE APSFR4TR(INV, NN, B0, B1, B2, B3, B4, B5, B6, B7)
DIMENSION L(15), B0(*), B1(*), B2(*), B3(*), B4(*), B5(*), B6(*),
       B7(*)
COMMON /CONS/ PII, P7, P7TWO, C22, S22, PI2
\end{verbatim}
EQUIVALENCE (L15,L(1)), (L14,L(2)), (L13,L(3)), (L12,L(4)),
* (L11,L(5)), (L10,L(6)), (L9,L(7)), (L8,L(8)), (L7,L(9)),
* (L6,L(10)), (L5,L(11)), (L4,L(12)), (L3,L(13)), (L2,L(14)),
* (L1,L(15))

C JTHET IS A REVERSED BINARY COUNTER, JR STEPS TWO AT A TIME TO
C LOCATE THE REAL PARTS OF INTERMEDIATE RESULTS, AND JI LOCATES
C THE IMAGINARY PART CORRESPONDING TO JR.

L(1) = NN/4
DO 40 K=2, 15
   IF (L(K-1)*2) 10, 20, 30
10   L(K-1) = 2
20   L(K) = 2
   GO TO 40
30   L(K) = L(K-1)/2
40 CONTINUE

PIOVN = PII/FLOAT(NN)
JI = 3
JL = 2
JR = 2

DO 120 J1=2, L1, 2
   DO 120 J2=J1, L2, L1
   DO 120 J3=J2, L3, L2
   DO 120 J4=J3, L4, L3
   DO 120 J5=J4, L5, L4
   DO 120 J6=J5, L6, L5
   DO 120 J7=J6, L7, L6
   DO 120 J8=J7, L8, L7
   DO 120 J9=J8, L9, L8
   DO 120 J10=J9, L10, L9
   DO 120 J11=J10, L11, L10
   DO 120 J12=J11, L12, L11
   DO 120 J13=J12, L13, L12
   DO 120 J14=J13, L14, L13
   DO 120 JTHET=J14, L15, L14
TH2 = JTHET - 2
   IF (TH2) 50, 50, 90
50   DO 60 K=1, INV
      T0 = B0(K) + B2(K)
      T1 = B1(K) + B3(K)
      B2(K) = B0(K) - B2(K)
      B3(K) = B1(K) - B3(K)
      B0(K) = T0 + T1
      B1(K) = T0 - T1
60 CONTINUE
IF (NN-4) 120, 120, 70
70  K0 = INV*4 + 1
   KL = K0 + INV - 1
   DO 80 K=K0, KL
      PR = P7*(B1(K) - B3(K))
      PI = P7*(B1(K) + B3(K))
      B3(K) = B2(K) + PI
      B1(K) = PI - B2(K)
      B2(K) = B0(K) - PR
      B0(K) = B0(K) + PR
   80 CONTINUE
   GO TO 120
90  ARG = TH2*PIOVN
   C1 = COS(ARG)
   S1 = SIN(ARG)
   C2 = C1**2 - S1**2
   S2 = C1*S1 + C1*S1
   C3 = C1*C2 - S1*S2
   S3 = C2*S1 + S2*C1
   INT4 = INV*4
   J0 = JR*INT4 + 1
   K0 = JI*INT4 + 1
   JLAST = J0 + INV - 1
   DO 100 J=J0, JLAST
      K = K0 + J - J0
      R1 = B1(J)*C1 - B5(K)*S1
      R5 = B1(J)*S1 + B5(K)*C1
      T2 = B2(J)*C2 - B6(K)*S2
      T6 = B2(J)*S2 + B6(K)*C2
      T3 = B3(J)*C3 - B7(K)*S3
      T7 = B3(J)*S3 + B7(K)*C3
      T0 = B0(J) + T2
      T4 = B4(K) + T6
      T2 = B0(J) - T2
      T6 = B4(K) - T6
      T1 = R1 + T3
      T5 = R5 + T7
      T3 = R1 - T3
      T7 = R5 - T7
      B0(J) = T0 + T1
      B7(K) = T4 + T5
      B6(K) = T0 - T1
      B1(J) = T5 - T4
      B2(J) = T2 - T7
      B5(K) = T6 + T3
B4(K) = T2 + T7
B3(J) = T3 - T6
100 CONTINUE
   JR = JR + 2
   JI = JI - 2
   IF (JI - JL) 110, 110, 120
110  JI = 2*JR - 1
    JL = JR
120 CONTINUE
RETURN
END
OVERVIEW

The COHO_SP routine reads samples of a burst from a Burst A/D file. Each block of burst digitized data is converted to complex and then frequency translated to baseband. The routine then computes the phase and returns the phase along with its associated TOA. The phase is then unwrapped, any trend determined and the trend is then removed.

SUBROUTINE COHO_SP (P_COND_PHASE,P_RAW_PHASE,
  P_TOA_ARRAY,P_NUM_SAMPLES,
  P_NUM_ELEMENTS,P_ERR)

  REAL*4 P_COND_PHASE(*)
  C Conditioned phase data with trend removed.

  REAL*4 P_RAW_PHASE(*)
  C Raw phase without trend removal

  REAL*8 P_TOA_ARRAY(*)
  C Time of arrival data.

  INTEGER*4 P_NUM_SAMPLES
  C Number of samples to process per pulse.
  C May be 1 or 11 (if phase sample display is selected)

  INTEGER*4 P_NUM_ELEMENTS
  C Number of pulses processed.

  INTEGER*4 P_ERR
  LOGICAL*4 L_END_OF_DATA
  C Flag indicating the End Of Data: 0 = FALSE, 1 = TRUE

  LOGICAL*4 L_EXIT
C Controls program execution.

    INTEGER*4  L_MODE

C Filter initialization mode:
C 0 = Zero initial conditions before filtering.
C 1 = Use the initial conditions for filtering.
    INTEGER*4  L_ERROR

C Indicates that errors occurred in the called subroutines.
    INTEGER*4  L_TOA_SAMPLE

C Index to the TOA array.
    INTEGER*4  L_I, L_J, L_K

C General index variables.
    INTEGER*4  L_BURST_COUNT

C Number of processed bursts.
    INTEGER*4  L_SAMPLES_READ

C Actual number of samples read.
    INTEGER*4  L_FIRST_SAMPLE

C First sample of data to be processed in this execution.
    REAL*4  L_LIMIT

C Limit of phase change from the previous value used to unwrap the phase.
    REAL*4  L_REAL_ARRAY(I_2K)

C Array holding the raw phase values returned by the reader.
    REAL*4  L_PHASE_ARRAY(I_2K)

C Array holding the phase values returned by the phase computation routine.
    REAL*4  L_COMPLEX_ARRAY(I_4K)

C Array holding the complex data computed by the real-to-complex filter routine.
REAL*8   L_FIRST_TOA  
C Indicates the first TOA value in the currently read buffer. 

REAL*8  L_CURRENT_TOA  
C Holds the value of the current Time Of Arrival  
L_END_OF_DATA = .FALSE.  
L_EXIT = .FALSE.  

L_I = 1  
L_J = 1  
L_MODE = 1  
L_BURST_COUNT = 0  

IF (RAW_PLOT_SELECTION(1:7) .EQ. 'ENABLED') THEN  
    L_TOA_SAMPLE = FIRST_SAMPLE + 5  
ELSE  
    L_TOA_SAMPLE = FIRST_SAMPLE  
ENDIF  

IF (HALF_FLAG) THEN  
    L_FIRST_SAMPLE = MOD(FIRST_SAMPLE, I_1K)  
ELSE  
    L_FIRST_SAMPLE = MOD(FIRST_SAMPLE, I_2K)  
ENDIF  

C Load data from first burst to be processed.  
L_CURRENT_TOA = PROCESSING_START_TIME  

C Position data file to start time  
CALL POSITION (L_CURRENT_TOA, STATUS)  

C Get data from first burst  
CALL COHO_READ (L_REAL_ARRAY, L_CURRENT_TOA,  
                1 L_SAMPLES_READ, L_ERROR, L_END_OF_DATA)  

L_FIRST_TOA = L_CURRENT_TOA  

C Process bursts until the processing end time is reached.  
100 IF (L_EXIT.OR.(L_CURRENT_TOA.GT_PROCESSING_END_TIME)) GOTO 101  
    IF (L_SAMPLES_READ.GT.1) THEN
L_BURST_COUNT = L_BURST_COUNT + 1

C Convert real burst data to complex.

CALL R2C_FILTER(L_REAL_ARRAY, L_COMPLEX_ARRAY,
 1 FIRST_SAMPLE, P_NUM_SAMPLES, L_MODE)

C Beat the burst sample to baseband and update parameters for the location of the sampling
C reference point which is the beginning of the pulse.

CALL FREQ_TRNSLT(L_COMPLEX_ARRAY, P_NUM_SAMPLES,
 1 L_CURRENT_TOA, REF_TOA,
 2 IF_FREQ, SAMPLE_RATE)

C Compute the phase of the samples. Output data is stored in the P_RAW_PHASE array.

CALL PHASE_CALC(L_COMPLEX_ARRAY, L_PHASE_ARRAY,
 1 P_NUM_SAMPLES)

DO 110 L_K = 1, P_NUM_SAMPLES
  P_RAW_PHASE(L_J) = L_PHASE_ARRAY(L_K)
  L_J = L_J + 1
110 CONTINUE

P_TOA_ARRAY(L_I) = DBLE(L_TOA_SAMPLE-1) * SAMPLE_WIDTH
  + L_FIRST_TOA
  L_I = L_I + 1

ENDIF

C Read the next burst

CALL COHO_READ (L_REAL_ARRAY, L_CURRENT_TOA,
 1 L_SAMPLES_READ, L_ERROR, L_END_OF_DATA)

L_FIRST_TOA = L_CURRENT_TOA

GOTO 100

101 CONTINUE

C If we are generating the phase vs. sample number display, then extract a single phase
C sample from each pulse to process.

IF ( RAW_PLOT_SELECTION(1:7) .EQ. 'ENABLED' ) THEN

C EXTRACT ONE PHASE SAMPLE PER BURST FROM THE P_RAW_PHASE
C ARRAY AND STORE IT IN P_COND_PHASE
DO 120 L_I = 1, L_BURST_COUNT
   L_J = (L_I - 1) * P_NUM_SAMPLES + 6
   P_COND_PHASE(L_I) = P_RAW_PHASE(L_J)
120 CONTINUE

ELSE

C MOVE THE DATA FROM P_RAW_PHASE TO P_COND_PHASE

DO 130 L_I = 1, P_NUM_SAMPLES * L_BURST_COUNT
   P_COND_PHASE(L_I) = P_RAW_PHASE(L_I)
130 CONTINUE
ENDIF

C Unwrap the phase and remove the trend from data in the P_COND_PHASE array.

L_LIMIT = I_DELTA_LIMIT
CALL PHASE_UNWRAP(P_COND_PHASE, L_BURST_COUNT, L_LIMIT)
CALL PHASE_TREND(P_COND_PHASE, P_TOA_ARRAY, L_BURST_COUNT)

P_NUM_ELEMENTS = L_BURST_COUNT
RETURN
END
OVERVIEW

The COHO_READ routine loads data for the coherency processing. It returns an array of data which contains the signal samples to be processed from a single burst. The routine also handles paging to the end of burst and checking to ensure that the desired data was available.

SUBROUTINE COHO_READ( P_REAL_ARRAY, P_CURRENT_TOA, P_SAMPLES_READ, P_ERROR, P_EOD )

REAL*4   P_REAL_ARRAY(*)
C Array holding signal data to be processed.
REAL*8   P_CURRENT_TOA
C TOA of first sample in returned data array. The first sample points to the first data point
INTEGER*4 P_SAMPLES_READ
C # of samples actually read.
INTEGER*4 P_ERROR
C Flag indicating that an error occurred in the initialization routine.
LOGICAL   P_EOD
C End of Data flag.
LOGICAL*4 L_EOB
C Flag indicating the End Of Burst: 0 = FALSE, 1 = TRUE
INTEGER*4 L_I
C General index variable.
INTEGER*4 L_NEXT_BURST
C Flag specifying whether the reader should position to the next burst or not:
C 0 = FALSE, 1 = TRUE
INTEGER*4  L_NUM_TO_READ

C Number of samples to be read.
REAL*4    L_DUMMY_ARRAY(I_4K)

C Array used to scroll through the signal data file without saving the data.
REAL*8    L_FIRST_TOA

C Indicates the first TOA value in the currently read buffer.
P_ERROR = 0
L_NEXT_BURST = 0

IF ( HALF_FLAG ) THEN
  L_NUM_TO_READ = I_1K
ELSE
  L_NUM_TO_READ = I_2K
ENDIF

C The following pages through the data in a burst to get to the data used for processing.
L_I = 1
100 IF ( (L_I .GT. READ_COUNT) .OR. (P_ERROR .NE. 0) ) GOTO 101

C Read L_NUM_TO_READ samples and store number returned. Only save data on last
C pass through loop.
    CALL READ_WB (L_NUM__TO_READ, P_REAL_ARRAY, P_SAMPLES_READ,
                L_NEXT_BURST,L_EOB, P_EOD,L_FIRST_TOA,STATUS)
    L_NUM_TO_READ = I_BLOCK_SIZE

C Save toa as current_toa
    P_CURRENT_TOA = L_FIRST_TOA

    IF ( P_EOD ) THEN
C set error to end of file
    P_ERROR = 1
ENDIF

C Check for premature end of burst
  IF ( L_EOB .AND. (L_I .NE. READ_COUNT) ) THEN
    P_ERROR = 1
  ENDIF
  L_I = L_I + 1

GOTO 100

101 CONTINUE

C Check for too little data
  IF ( LAST_SAMPLE .GT. P_SAMPLES_READ) THEN
    P_ERROR = 1
  ENDIF

C Scroll through any remaining data in the burst but don't overwrite the data previously
C read for processing
  IF ( (P_ERROR .EQ. 0) .AND. (.NOT. L_EOB) ) THEN
    IF ( L_EOB ) GOTO 111

C Read the data but don't save it
  CALL READ_WB (L_NUM_TO_READ, L_DUMMY_ARRAY,
                  P_SAMPLES_READ, L_NEXT_BURST,
                  L_EOB, P_EOD, L_FIRST_TOA, STATUS)
  GOTO 110

111 CONTINUE

ENDIF

RETURN
END
OVERVIEW

NAME R2C_FILTER

PURPOSE

The R2C_Filter routine performs real to complex conversion of data using a
time-domain Hilbert transform on the input waveform. The routine is designed so it
can process a continuous stream of data or only transform a small segment of a
larger input block of data. The routine distinguishes the two cases based on the
value of the variable FIRST_SAMPLE. If equal to one, then continuous stream
processing is assumed. If greater than one, then we assume a small block will be
processed.

When a small block of data is to be processed, the calling routine passes the routine
a block signal data of which only a small number of internal samples are to be
processed. FIRST_SAMPLE is the index of the first sample to process.
NUM_SAMPLES is the number of data samples to be processed and returned.
Ideally, FIRST_SAMPLE is greater than the filter delay and the number of samples
trailing the last sample to process is greater than or equal to the delay. For this case
the routine returns NUM_SAMPLES complex data points starting at element 1 of
the output data array with no filter delay. The setting of MODE has no effect on the
to operation for this case.

For continuous processing, the routine zero pads the first data block input such that
the samples of the first block of continuous data is delayed by the filter delay. The
first block is distinguished from later blocks by MODE = 0. In all cases, however,
the routine returns NUM_SAMPLES complex points. The number of delay
samples is equal to the Hilbert transform length minus one.

SUBROUTINE R2C_FILTER ( P_REAL_ARRAY, P_COMPLEX_ARRAY,
        P_FIRST_SAMPLE, P_NUM_SAMPLES, P_MODE )

Real to Complex globals.

        INTEGER*4   P_NUM_SAMPLES
Number of samples to process.

        REAL*4     P_REAL_ARRAY(3*P_NUM_SAMPLES)
Signal data to transform.
REAL*4   P_COMPLEX_ARRAY(6*P_NUM_SAMPLES)

C Transformed output data.
  INTEGER*4   P_FIRST_SAMPLE

C First sample to process. If it is greater than one, data is not delayed.
  INTEGER*4   P_MODE

C Filter initialization mode:
  C 0 = Zero initial conditions before filtering.
  C 1 = Use the initial conditions for filtering.

  INTEGER*4   L_PTR

C In the case where only a small number of points is processed, this variable holds a
C pointer to the data. If the input array does not have enough leading points to use as the
C initial conditions, it points to the first element. Otherwise, its value is derived based on
C the difference between the value of FIRST_SAMPLE & the delay.

  INTEGER*4   L_STEP

C Increment which is derived based on the type of Hilbert transform. For odd length
C transforms even samples are 0 so only odd multiplications are performed.

  INTEGER*4   L_NINIT

C Number of zero samples to be inserted as initial conditions

  INTEGER*4   L_N_TO_MOVE

C Number of samples to move into the working array which includes the first sample
C index and the filter delay.

  INTEGER*4   L_LAST_ELEM

C Last element to process when a small number of points from a larger block is processed.

  INTEGER*4   L_FIRST_ELEM

C First element to process when a small number of points from a larger block is processed.

  INTEGER*4   L_NUM_INIT_COND

C Number of initial conditions that are to be loaded into the working array.

  INTEGER*4   L_I, L_I1, L_I2, L_IO
C General purpose indices.

INTEGER*4   L_IDO, L_NDO

C Indices used in the required convolution operation.

INTEGER*4   L_IDON, L_IDOM

REAL*4      L_VAL

C Holds the summation of the multiplications of the working array with the Hilbert coeff.

REAL*4      L_WORK_ARRAY(4300)

C Working array used in the overlap and save operation.

C Compute the number of initial conditions.

L_NUM_INIT_COND = HILBERT_LENGTH - 1

DO 100 L_I = 1, 4300
       L_WORK_ARRAY(L_I) = 0.0
100   CONTINUE

IF ( P_FIRST_SAMPLE .EQ. -1 ) THEN

C Handle initialization for continuous processing

IF ( P_MODE .EQ. 0 ) THEN

C Zero the initial conditions for the transform.

DO 110 L_I = 1, L_NUM_INIT_COND
       INITIAL_CONDITION(L_I) = 0.0
110   CONTINUE

ENDIF

C Load the work area with the initial conditions. The initial conditions are loaded into
C every other memory location to form the real component of a complex data format.

L_IO = 1

DO 120 L_I = 1, L_NUM_INIT_COND
       L_WORK_ARRAY(L_IO) = INITIAL_CONDITION(L_I)
       L_IO = L_IO + 2
120   CONTINUE
C Move the NUM_SAMPLES real input data to the work array so that a complex vector
C with zero imaginary values is formed.

    DO 130 L_I = 1, P_NUM_SAMPLES
       L_WORK_ARRAY(L_IO) = P_REAL_ARRAY(L_I)
       L_WORK_ARRAY(L_IO+1) = 0.0
       L_IO = L_IO + 2
    130 CONTINUE

C Before filtering save the last NUM_INIT_COND samples of the input data for the next
C time the routine is called.

    DO 140 L_I = 1, L_NUM_INIT_COND
       INITIAL_CONDITION(L_I) =
       P_REAL_ARRAY(L_I + P_NUM_SAMPLES - L_NUM_INIT_COND)
    140 CONTINUE

ELSE
C Handle the case of only processing a small number of points from a larger block. Here
C we zero pad only if the input array does not have enough leading points to use as the
C initial conditions.

    IF ( P_FIRST_SAMPLE .LE. FILTER_DELAY ) THEN

C Insert zero samples as initial conditions

       L_NINIT = ( FILTER_DELAY + 1 - P_FIRST_SAMPLE ) * 2
       DO 150 L_I = 1, L_NINIT
          L_WORK_ARRAY(L_I) = 0.0
       150 CONTINUE

       L_FIRST_ELEM = L_NINIT + 1
       L_N_TO_MOVE = P_NUM_SAMPLES + P_FIRST_SAMPLE - 1 +
                      FILTER_DELAY
       L_PTR = 1
    ELSE

C Use leading data samples as initial conditions

       L_FIRST_ELEM = 1
       L_N_TO_MOVE = P_NUM_SAMPLES + FILTER_DELAY * 2
       L_PTR = P_FIRST_SAMPLE - FILTER_DELAY
    ENDIF

C Load the signal data to the working array.

       L_LAST_ELEM = L_FIRST_ELEM + 2*L_N_TO_MOVE
C Convolve the input data and initial conditions with the Hilbert transform to form the complex signal component of the data. This takes into account the time delay of the Hilbert transform. For odd length Hilbert transforms even samples are zero so only odd multiplications are performed.

\[
\begin{align*}
L_{\text{IO}} &= \text{HILBERT\_LENGTH} + 1 \\
L_{\text{I}} &= 1 \\
L_{\text{STEP}} &= \text{MOD(HILBERT\_LENGTH, 2)} + 1
\end{align*}
\]

DO 170 L\_IDON = 1, P\_NUM\_SAMPLES
  \hspace{1cm} L_{\text{VAL}} = 0.0
  \hspace{1cm} L_{\text{I}1} = L_{\text{I}}
  \hspace{1cm} L_{\text{I}2} = \text{HILBERT\_LENGTH}

DO 180 L\_IDOM = 1, HILBERT\_LENGTH, L\_STEP
  \hspace{1cm} L_{\text{VAL}} = L_{\text{VAL}} + L_{\text{WORK\_ARRAY}}(L_{\text{I}1}) \times \text{HIL\_COEF}(L_{\text{I}2})
  \hspace{1cm} L_{\text{I}1} = L_{\text{I}1} + 2*\text{L\_STEP}
  \hspace{1cm} L_{\text{I}2} = L_{\text{I}2} - \text{L\_STEP}
180 \quad \text{CONTINUE}

\[
\begin{align*}
L_{\text{WORK\_ARRAY}}(L_{\text{IO}}) &= L_{\text{VAL}} \\
L_{\text{I}} &= L_{\text{I}} + 2 \\
L_{\text{IO}} &= L_{\text{IO}} + 2
\end{align*}
\]

DO 190 L\_IDO = 1, L\_NDO
  \hspace{1cm} P\_COMPLEX\_ARRAY(L_{\text{IO}}) = L_{\text{WORK\_ARRAY}}(L_{\text{I}})
  \hspace{1cm} L_{\text{I}} = L_{\text{I}} + 1
  \hspace{1cm} L_{\text{IO}} = L_{\text{IO}} + 1
190 \quad \text{CONTINUE}

RETURN
END
SUBROUTINE FREQ_TRNSLT(P_DATA_ARRAY,P_NUM_SAMPLES,
   1 P_FIRST_TOA,P_REF_TOA, P_IF_FREQ,
   2 P_SAMPLE_RATE)

   REAL*4   P_DATA_ARRAY(*)
   C Complex data vector to process.
   INTEGER*4 P_NUM_SAMPLES
   C Number of burst A/D samples being processed.
   REAL*8   P_FIRST_TOA
   C First TOA of data in this buffer.
   REAL*8   P_REF_TOA
   C Reference TOA (seconds).
   REAL*8   P_IF_FREQ
   C Conversion frequency (HZ)
   REAL*8   P_SAMPLE_RATE
   C Sampling rate (Hz)
   INTEGER*4 L_I
   C General index variable.
REAL*8   L_VAL
C Temporary variable holding real part of the resulting complex conjugate.

REAL*8   L_RAMP
C Ramp function determined through a modulo operation.

REAL*8   L_2PI_DBLE
C Value of the constant Pi in double precision.

PARAMETER ( L_2PI_DBLE = 6.283185307179586D0 )

REAL*8   L_DELTA_PHASE
C Phase increment.

REAL*8   L_TIME_ADVANCE
C Time advance over a sampling period determined by the the difference of the current
C TOA and a reference TOA.

REAL*8   L_RAMP_START_PHASE
C Initial phase start of the currently processed burst.

REAL*8   L_COMPLEX_VECTOR(2)
C Holds the value of the complex exponential which is comprised by a cosine of the ramp
C (real) function and the negative sine of the ramp function (imaginary)

C Compute phase increment, 2*pi*Ifreq*sampling period.

L_DELTA_PHASE = (L_2PI_DBLE * P_IF_FREQ)/ P_SAMPLE_RATE

C Compute the initial phase start of this burst, 2*pi*Ifreq*(time offset from reference
C TOA), reference TOA has initial phase of zero.

L_TIME_ADVANCE = (P_FIRST_TOA - P_REF_TOA)
L_RAMP_START_PHASE = L_2PI_DBLE * P_IF_FREQ * L_TIME_ADVANCE
L_RAMP_START_PHASE = MOD (L_RAMP_START_PHASE, L_2PI_DBLE)

L_RAMP = L_RAMP_START_PHASE

DO 100 L_I = 1, 2*P_NUM_SAMPLES, 2

C Compute the complex exponential using the ramp function. This complex exponential is
C approximately equal to the complex conjugate of the burst intermediate frequency
C component. Calculate the cos and sin of the elements of the ramp vector. Store results in the complex vector. Store cosine values in the real part(1). Store sine values in the imaginary part(2).

```
L_COMPLEX VECTOR(1) = COS(L_RAMP)
L_COMPLEX VECTOR(2) = -SIN(L_RAMP)
```

C Frequency translate the burst block to baseband by multiplying the time domain data by the complex exponential. Hilbert transformed data, A+jB, complex conjugate IF, C+jD. Real result is A*C - B*D, complex result is B*C + A*D.

```
L_VAL = P_DATA_ARRAY(L_I) * L_COMPLEX VECTOR(1) -
       P_DATA_ARRAY(L_I+1) * L_COMPLEX VECTOR(2)
       +
       P_DATA_ARRAY(L_I+1) * L_COMPLEX VECTOR(1)
       P_DATA_ARRAY(L_I) = L_VAL
```

C Advance the phase for next sample.

```
L_RAMP = L_RAMP + L_DELTA_PHASE
```

100 CONTINUE

RETURN
END
OVERVIEW

NAME PHASE_CALC

PURPOSE

The PHASE_CALC routine computes the phase of a digitized pulse and returns the phase.

SUBROUTINE

PHASE_CALC(P_IQ_ARRAY,P_PHASE_ARRAY,P_NUM_SAMPLES)

INTEGER*4 P_NUM_SAMPLES

P_IQ_ARRAY(*)

REAL*4

P_PHASE_ARRAY(*)

INTEGER*4 P_PHASE_ARRAY(*)

REAL*4

L_HPI

DATA L_HPI/90.0/

C General index variables.

L_I, L_J

C Compute the phase.

DO 100 L_I = 1, P_NUM_SAMPLES*2, 2

IF (P_IQ_ARRAY(L_I) .NE. 0) THEN

P_PHASE_ARRAY(L_J) = ATAN2D(P_IQ_ARRAY(L_I+1), P_IQ_ARRAY(L_I))

ELSE

P_PHASE_ARRAY(L_J) = SIGN(L_HPI, P_IQ_ARRAY(L_I+1))

ENDIF

L_J = L_J + 1

100 CONTINUE

RETURN

END
OVERVIEW
This routine unwraps the input phase data. The operation is performed in place with
the unwrapped data returned in the input array and the original data being destroyed.

SUBROUTINE PHASE_UNWRAP
1 (P_COND_PHASE, P_PHASEBUFF_SIZE, P_LIMIT)

C Global parameter declarations.
INTEGER*4 P_PHASEBUFF_SIZE
C Actual number of elements in preprocessing module's phase buffer.
REAL*4 P_COND_PHASE(P_PHASEBUFF_SIZE)
C Array containing phase data.
REAL*4 P_LIMIT
C Unwrapping limit.
INTEGER*4 L_I
C General index variable.
REAL*4 L_COR
C Phase correction factor.
REAL*4 L_360D
C Constant set to 360 degrees.
REAL*4 L_ALIM
C Limit of phase change.
REAL*4 L_PHASE1

C Last phase value.
REAL*4 L_DELTA_PHASE

C Difference between current phase value and last phase.

L_COR = 0.0
L_360D = I_2PI * I_RAD_TO_DEG
L_ALIM = (I_2PI - P_LIMIT) * I_RAD_TO_DEG
L_PHASE1 = P_COND_PHASE(1)

DO 100 L_I = 2, P_PHASEBUFF_SIZE

C Unwrap the buffer of phase data.

L_DELTA_PHASE = P_COND_PHASE(L_I) - L_PHASE1

IF ( L_DELTA_PHASE .GT. L_ALIM ) THEN
   L_COR = L_COR - L_360D
ELSE IF ( L_DELTA_PHASE .LT. -L_ALIM ) THEN
   L_COR = L_COR + L_360D
ENDIF

L_PHASE1 = P_COND_PHASE(L_I)
P_COND_PHASE(L_I) = P_COND_PHASE(L_I) + L_COR

100 CONTINUE

RETURN
END
OVERVIEW

NAME PHASE_TREND

PURPOSE

The PHASE_TREND routine removes the linear trend from an array of phase data.

Processing begins by scanning the phase data to accumulate the sums used to compute the linear trend. The routine next computes the coefficients of a polynomial for a linear fit to the phase data. The coefficients are computed by means of a least-squares-fit method. The linear independent equations of the least-squares-fit method are solved by Gauss-Jordan reduction of double precision matrices and vectors. After the line has been computed, it is removed from the input data.

SUBROUTINE PHASE_TREND(P_COND_PHASE, P_TOA_ARRAY, L_I, L_J, L_K, L_PHASE_SUM)

INTEGER*4 P_NUM_ELEMENTS

C Number of elements in each array.

REAL*4 P_COND_PHASE(*)

C Array of phase data from which a computed trend will be removed.

REAL*8 P_TOA_ARRAY(*)

C Array of TOA data corresponding to the array of phase data.

INTEGER*4 L_I, L_J, L_K

C General index variables.

REAL*8 L_TOA

C Holds difference between TOA elements in the TOA array and the reference TOA.

REAL*8 L_TOA_SUM

C Sum of TOA's.

REAL*8 L_PHASE_SUM
C Sum of phase elements.
    REAL*8 L_PHASE_DBLE
C Converts the type of the conditioned phase elements to double precision.
    REAL*8 L_SUM_VEC(3)
C Array of trend summations.
    REAL*8 L_OFFSET_TOA
C TOA offset used to compute the phase trend.
    REAL*8 L_TREND_PHASE
C Phase trend.
    REAL*8 L_SUM_MAT(3,3)
C Matrix of trend summations.
    REAL*8 L_AUG_MAT(3,3)
C Augmented matrix.
    REAL*8 L_TREND_COEF(3)
C Trend coefficients.
    REAL*8 L_PHASE_TOA_SUM
C Sum of (phase*toa).
    REAL*8 L_PHASE_TOA_SUM2
C Sum of (phase*toa) ^ 2
    REAL*8 L_TEMP1, L_TEMP2
C Temporary storage variables.
C Initialize summation variables.
    L_TOA_SUM = 0.0
    L_PHASE_SUM = 0.0
    L_PHASE_TOA_SUM = 0.0
    L_PHASE_TOA_SUM2 = 0.0
DO 100 L_I = 1, P_NUM_ELEMENTS
   L_TOA = P_TOA_ARRAY(L_I) - REF_TOA
   L_PHASE_DBLE = DBLE(P_COND_PHASE(L_I))
   L_PHASE_SUM = L_PHASE_SUM + L_PHASE_DBLE
   L_PHASE_TOA_SUM = L_PHASE_TOA_SUM + L_PHASE_DBLE * L_TOA
   L_TOA_SUM = L_TOA_SUM + L_TOA
   L_PHASE_TOA_SUM2 = L_PHASE_TOA_SUM2 + L_TOA**2
100 CONTINUE

C Linear Trend Removal

   L_ORDER = 2

C Initialize summation matrix and summation vector.

   L_SUM_MAT(1,1) = L_PHASE_TOA_SUM2
   L_SUM_MAT(1,2) = L_TOA_SUM
   L_SUM_MAT(2,1) = L_TOA_SUM
   L_SUM_MAT(2,2) = FLOAT(P_NUM_ELEMENTS)
   L_SUM_VEC(1) = L_PHASE_TOA_SUM
   L_SUM_VEC(2) = L_PHASE_SUM

C Enter summation matrix in workspace array and enter summation vector as ORDER+1th
C column of the workspace array.

   DO 110 L_I = 1, L_ORDER
      DO 120 L_J = 1, L_ORDER
         L_AUG_MAT(L_I, L_J) = L_SUM_MAT(L_I, L_I)
      120 CONTINUE
      L_AUG_MAT(L_I, L_ORDER + 1) = L_SUM_VEC(L_I)
   110 CONTINUE

C Reduce augmented matrix AUG_MAT to solve the set of simultaneous equations.

C Set main diagonal elements to unity.

   L_TEMP1 = L_AUG_MAT(L_I, L_I)
   IF (L_TEMP1 .NE. 0) THEN
      DO 140 L_J = L_I, L_ORDER + 1
         L_AUG_MAT(L_I, L_J) = L_AUG_MAT(L_I, L_J)/L_TEMP1
      140 CONTINUE
   ENDIF
C Set elements of Ith column to zero.

DO 150 L_K = 1, L_ORDER

    IF ( L_K .NE. L_I ) THEN
        L_TEMP2 = L_AUG_MAT(L_K, L_I)

        DO 160 L_J = L_I, L_ORDER + 1
            L_AUG_MAT(L_K, L_J) = L_AUG_MAT(L_K, L_J) -
            L_TEMP2*L_AUG_MAT(L_I, L_J)
        END
    160 CONTINUE

END IF

150 CONTINUE

130 CONTINUE

C Set output vector equal to last column of augmented matrix AUG_MAT.

DO 170 L_I = 1, L_ORDER
    L_TREND_COEF(L_I) = L_AUG_MAT(L_I, L_ORDER+1)
170 CONTINUE

DO 180 L_I = 1, P_NUM_ELEMENTS

C Compute the offset TOA.

    L_OFFSET_TOA = P_TOA_ARRAY(L_I) - REF_TOA

C Remove the linear phase trend.

    L_TREND_PHASE = L_TREND_COEF(2) +
    1 * L_TREND_COEF(1) * L_OFFSET_TOA

    P_COND_PHASE(L_I) = P_COND_PHASE(L_I) - L_TREND_PHASE

180 CONTINUE

    TUNING_OFFSET = L_TREND_COEF(1) / 360.0

RETURN
END
OVERVIEW

The PHASE_TOA routine is the main calling routine for generating the Phase versus TOA display. Arrays of phase and TOA information are displayed until the requested number of points have been plotted.

SUBROUTINE PHASE_TOA(P_COND_PHASE, P_TOA_ARRAY, P_NUM_PLOT, P_XMIN, P_XMAX, P_YMIN, P_YMAX)

REAL*4 P_COND_PHASE(*)

Array of phase data that will be scaled and plotted by the phase display routine.

REAL*8 P_TOA_ARRAY(*)

Array of TOA data corresponding to the phase data in the phase array. These data are scaled, and plotted by the phase display routine.

INTEGER*4 P_NUM_PLOT

Number of phase-TOA points to plot.

REAL*4 P_XMIN

Minimum X coordinate.

REAL*4 P_XMAX

Maximum X coordinate.

REAL*4 P_YMIN

Minimum Y coordinate.

REAL*4 P_YMAX

Maximum Y coordinate.
INTEGER*4 L_I, L_J
C General index variables.
REAL*4 L_CLIST(6144)
C Holds the coordinates of the points to be plotted.
REAL*4 L_XMIN
C Min X coordinate of the active PHASE vs PULSE # display area.
REAL*4 L_XMAX
C Max X coordinate of the active PHASE vs PULSE # display area.
REAL*4 L_YMIN
C Min Y coordinate of the active PHASE vs PULSE # display area.
REAL*4 L_YMAX
C Max Y coordinate of the active PHASE vs PULSE # display area.
REAL*8 L_OFFSET
C Display offset time.
REAL*8 L_TOA_MIN
C Min value of the TOA data.
REAL*8 L_TOA_MAX
C Max value of the TOA data.
REAL*8 L_PHASE_MIN
C Min value of the phase data.
REAL*8 L_PHASE_MAX
C Max value of the phase data.
CHARACTER*20 L_XLABEL, L_YLABEL
C Titles of the trace of raw PHASE vs PULSE # display in both the X and the Y axes.
C Compute the initial display offset time from the initial display start time.

L_OFFSET = FILE_STAR_TOA

C FIND MINIMA AND MAXIMA FROM THE PHASE AND TOA ARRAYS

L_PHASE_MIN = DBLE(P_COND_PHASE(1))
L_PHASE_MAX = DBLE(P_COND_PHASE(1))
L_TOA_MIN = PROCESSING_START_TIME
L_TOA_MAX = PROCESSING_END_TIME

DO 100 L_I = 2, P_NUM_PLOT
   IF (L_PHASE_MIN .GT. P_COND_PHASE(L_I)) THEN
      L_PHASE_MIN = P_COND_PHASE(L_I)
   ENDIF
   IF (L_PHASE_MAX .LT. P_COND_PHASE(L_I)) THEN
      L_PHASE_MAX = P_COND_PHASE(L_I)
   ENDIF
100 CONTINUE

C CONVERT TOA TO MILLISECONDS

L_TOA_MIN = (L_TOA_MIN - L_OFFSET) / I_MSEC_UNITS
L_TOA_MAX = (L_TOA_MAX - L_OFFSET) / I_MSEC_UNITS

C Plot tic marks on both axes.

C Assign label names.

C Initialize the display for the initial ranges.

CALL DISPLAY_INIT(FD_TIME,
1   L_XMIN, L_XMAX,
2   L_YMIN, L_YMAX,
3   L_TOA_MIN, L_TOA_MAX,
4   L_PHASE_MIN, L_PHASE_MAX,
5   L_XLABEL, L_YLABEL,
6   L_XTIC)

C Scale and plot the phase and time of arrival data until the specified number of points have
C been plotted.

RETURN
END
The PHASE_SAMPLE routine generates the initial display of the raw phase versus sample number plot. Depending on the program set up this display may be generated in a series of pages. PHASE_SAMPLE is responsible for displaying the first page of data.

SUBROUTINE PHASE_SAMPLE(P_RAW_PHASE, P_NUM_PLOT)

REAL*4 P_RAW_PHASE(*)

INTEGER*4 P_NUM_PLOT

INTEGER*4 L_NUM_PLOT

INTEGER*4 L_NUM_PAGES

INTEGER*4 L_I, L_J, L_K

REAL*4 L_POINT

REAL*4 L_INCREMENT
C Increment between points to be plotted in the X direction.
REAL*4   L_CLIST(6144)

C Holds the coordinates of the points to be plotted.
REAL*4   L_XMIN

C Min X coordinate of the active PHASE vs PULSE # display area.
REAL*4   L_XMAX

C Max X coordinate of the active PHASE vs PULSE # display area.
REAL*4   L_YMIN

C Min Y coordinate of the active PHASE vs PULSE # display area.
REAL*4   L_YMAX

C Max Y coordinate of the active PHASE vs PULSE # display area.
REAL*8   L_PHASE_MIN

C Min value of the phase data.
REAL*8   L_PHASE_MAX

C Max value of the phase data.
REAL*8   L_LAST_PULSE

C Value of the last pulse to be plotted.
REAL*8   L_FIRST_PULSE

C Value of the first pulse to be plotted.
CHARACTER*20   L_XLABEL, L_YLABEL

C Titles of the trace of raw PHASE vs PULSE # display in both the X and the Y axes.
C See if there is more than one page of data

L_NUM_PAGES = INT( FLOAT(NUM_ELEMENTS) /
1                     FLOAT(PLOT_PAGE_SIZE) + 0.99)

IF (L_NUM_PAGES .LT. 1) THEN
    RETURN
ENDIF
C Set range of pulses to be displayed
   L_FIRST_PULSE = 1.0
   L_LAST_PULSE = PLOT_PAGE_SIZE

   IF (L_LAST_PULSE .GT. NUM_ELEMENTS) THEN
      L_LAST_PULSE = NUM_ELEMENTS
   ENDIF

   PAGE_NUM = 1
C Initialize the display viewport for the phase vs pulse # display.

C FIND THE NUMBER OF POINTS TO BE PLOTTED ON THE FIRST TRACE
C FIND THE MINIMUM AND MAXIMUM OF THE PHASE ARRAY
C Plot tic marks on both axes and truncate labels on the X axis.
C Assign label names.
C Initialize the display for the initial ranges.
   CALL DISPLAY_INIT(FD_PULSE,  
                    1 L_XMIN, L_XMAX,  
                    2 L_YMIN, L_YMAX,  
                    3 L_FIRST_PULSE, L_LAST_PULSE,  
                    4 L_PHASE_MIN, L_PHASE_MAX,  
                    5 L_XLABEL, L_YLABEL,  
                    6 L_XTIC)

C SCALE THE PHASE DATA
C PLOT THE PHASE VERSUS PULSE NUMBER DATA AS DISCRETE POINTS
C Compute the increment of the points to be plotted along the X axis.
C Plot the pulses as determined by the plot page size.
   L_K = 1

   DO 110 L_J = INT(L_FIRST_PULSE), INT(L_LAST_PULSE - L_NUM_POINTS)
   C Compute the X coordinate for the first point of each pulse to be plotted.
      L_POINT = FLOAT(L_J) - 5.0 * L_INCREMENT
   C Plot eleven points per pulse.
      DO 120 L_I = 1, 11
   C If the point to be plotted is the midpoint for a given pulse mark it with a rectangle,
   C otherwise plot it as a dot.
   C Increment phase index.
      L_K = L_K + 1
   C Increment the X coordinate for each point of the pulse.
      L_POINT = L_POINT + L_INCREMENT
   120 CONTINUE
   110 CONTINUE
RETURN
END
OVERVIEW

NAME PHASE_STATS

PURPOSE

This routine computes the standard deviation of first phase difference and the standard deviation of residual phase. Processing begins by first locating the data to be processed. The routine selects the phase data bounded by the processing start and end times. It then computes the sums and finally the desired standard deviations. Processing ends with the display of the statistics.

SUBROUTINE PHASE_STATS(P_PHASE_ARRAY, P_TOA_ARRAY, P_NUM_ELEMENTS)

REAL*4 P_PHASE_ARRAY(*)

C Array containing the phase data that will be used to update the statistical summations.

REAL*8 P_TOA_ARRAY(*)

C Array containing the time of arrival data that corresponds to the phase data.

INTEGER*4 P_NUM_ELEMENTS

C Number of elements in the phase array.

LOGICAL*4 L_EXIT

C Flag used to terminate loops

INTEGER*4 L_I, L_N

C Loop indices.

INTEGER*4 L_SAMPLE1

C First sample following the minimum TOA.

INTEGER*4 L_SAMPLE2

C First sample following the maximum TOA.
REAL*8 L_PHASE_DIFF  
C Difference between two contiguous phase values.
REAL*8 L_PREV_PHASE  
C Previous phase value; used to compute the phase difference.
REAL*8 L_SUM_PHASE_DIFF  
C Sum of First Phase differences 
REAL*8 L_SUM_PHASE_DIFF2  
C Sum of squared First Phase differences.
REAL*8 L_STD_FIRST_PHASE  
C Standard deviation of First Phase differences.

C Find the range of samples to process. 
C FIND FIRST SAMPLE WITH TOA > START_TIME AND STORE AS SAMPLE1
C FIND FIRST SAMPLE WITH TOA < END_TIME AND STORE AS SAMPLE2

L_I = 1  
L_EXIT = .FALSE.
100 IF ( (L_I .GE. P_NUM_ELEMENTS) .AND. L_EXIT ) GOTO 110
    IF (P_TOA_ARRAY(L_I) .GT. PROCESSING_START_TIME) THEN  
        L_SAMPLE1 = L_I  
        L_EXIT = .TRUE.  
    ENDIF  
    L_I = L_I + 1  
GOTO 100
110 CONTINUE
L_EXIT = .FALSE.  
L_I = P_NUM_ELEMENTS
120 IF ( (L_I .LE. 1) .AND. L_EXIT ) GOTO 130
    IF (P_TOA_ARRAY(L_I) .LT. PROCESSING_END_TIME) THEN  
        L_SAMPLE2 = L_I  
    ENDIF
L_EXIT = .TRUE.
ENDIF

L_I = L_I - 1

GOTO 120
130 CONTINUE

IF (P_NUM_ELEMENTS .EQ. 1) THEN
L_SAMPLE1 = 1
L_SAMPLE2 = 2
ENDIF

C Initialize local summation variables.

L_SUM_PHASE_DIFF = 0.0
L_SUM_PHASE_DIFF2 = 0.0

L_PREV_PHASE = DBLE(P_PHASE_ARRAY(L_SAMPLE1))

DO 140 L_I = L_SAMPLE1 + 1, L_SAMPLE2

C Update the summations.

L_PHASE_DIFF = DBLE(P_PHASE_ARRAY(L_I)) · L_PREV_PHASE
L_SUM_PHASE_DIFF = L_SUM_PHASE_DIFF + L_PHASE_DIFF
L_SUM_PHASE_DIFF2 = L_SUM_PHASE_DIFF2 + L_PHASE_DIFF**2
L_PREV_PHASE = DBLE(P_PHASE_ARRAY(L_I))

140 CONTINUE

L_N = L_SAMPLE2 - L_SAMPLE1

IF (L_N .GT. 0) THEN

C Compute and display standard deviation of first phase difference

L_SUM_PHASE_DIFF2 = L_SUM_PHASE_DIFF2 / DBLE(L_N)
L_SUM_PHASE_DIFF = (L_SUM_PHASE_DIFF / DBLE(L_N))**2

ENDIF

L_STD_FIRST_PHASE=DSQRT(L_SUM_PHASE_DIFF2-L_SUM_PHASE_DIFF)

RETURN
END
REFERENCES


