NEXRAD Range-Velocity Mitigation

FY2004 Report

S-Pol reflectivity after SSEF filter

Prepared for: WSR-88D Radar Operations Center

John Hubbert, Mike Dixon,
Greg Meymaris and Scott Ellis

National Center for Atmospheric Research

23 December 2004
# Table of Contents

Executive Summary ............................................................................................................ 1

1 Introduction ...................................................................................................................... 3

2 Evaluation of GMAP for use with SZ algorithms .......................................................... 4

2.1 Justification .................................................................................................................. 4

2.2 Overview ...................................................................................................................... 4

2.3 GMAP description. ...................................................................................................... 4

2.4 Suitability of spectral clutter filters for SZ algorithms. .............................................. 5

2.5 Testing clutter filters on reconstructed spectra. ........................................................ 5

2.6 Examples of synthetic spectra...................................................................................... 6

2.7 Creating a GMAP test environment at NCAR ............................................................ 8

2.8 Qualitative investigation of GMAP ............................................................................ 9

2.9 SSEF – an alternative spectral clutter filter .................................................................. 11

2.10 Quantitative testing on reconstructed ‘real’ spectra. ............................................... 16

2.10.1 Quantitative performance – weather power.......................................................... 16

2.10.2 Quantitative performance - velocity ..................................................................... 22

2.10.3 Quantitative performance – spectrum width......................................................... 27

2.11 Case study of GMAP – under-estimation of clutter power....................................... 32

2.12 Performance of GMAP and SSEF on bi-modal weather spectra............................. 39

2.13 Applying SSEF to SPOL data, using a simplified Radar Echo Classifier.................. 42

2.14 Conclusions ............................................................................................................... 46
Executive Summary

This document reports on the continued investigation of issues surrounding the SZ(8/64) phase coding algorithm for range-velocity ambiguity mitigation. Phase coded time series data was gathered by S-Pol, NCAR’s S-Band polarimetric research radar, using the SIGMET RVP8 receiver/processor which operates in parallel with the existing S-Pol receiver/processor. Data was collected at the Marshall Field site near Boulder, CO and in Mazatlan, Mexico during the NAME (North American Monsoon Experiment) field program.

A major milestone of this fiscal year was the delivery of the SZ-2 algorithm software with censoring on 15 June in cooperation with NSSL. The SZ-2 algorithm is designed to be used on the lowest elevation angles of the NEXRAD VCPs. See the NSSL/NCAR 15 June Interim Report to the ROC, [http://www.atd.ucar.edu/rsf/NEXRAD/index.html](http://www.atd.ucar.edu/rsf/NEXRAD/index.html), for further information on this topic. This is the culmination of many years effort by both NCAR and NSSL as well as by the ROC staff members. The algorithm will dramatically improve data quality by reducing the amount of “purple haze” currently seen in NEXRAD data. The algorithm, though tested and delivered, needs further testing and likely modification for optimum performance.

Another major accomplishment was the delivery of the 30 September Interim report to the ROC on the comparison of the SIGMET and NSSL/NCAR SZ-1 algorithms. Experimental time series data was gathered by alternating PPI scans of short PRT phase coded and long PRT non-phase-coded data. The phase coded data was replayed through the RVP8 so that both the SIGMET and NSSL/NCAR algorithm could process the same data. The long PRT processed moments were used as truth for the short PRT moments. The analysis showed that the moments produced by the two SZ algorithms were by-in-large quite comparable but there were notable differences. There were a significant number of anomalous velocities calculated by the SIGMET algorithm as evidenced by the scatter plots. The cause of the anomalous velocities is not known but is an artifact of the SIGMET algorithm. Also, no bias was found in the SIGMET weak trip velocity estimates. There was a 2 m/s bias found in simulations conducted by NSSL as reported in their annual report to the ROC, Part 6 (2002). NCAR’s full Interim Report can be viewed at [http://www.atd.ucar.edu/rsf/NEXRAD/index.html](http://www.atd.ucar.edu/rsf/NEXRAD/index.html).

Another accomplishment was the analysis and testing of the SIGMET GMAP clutter filter and the development and implementation of NCAR’s similar spectral domain clutter filter called the Spectral Stationary Echo Filter (SSEF). The clutter filters were tested by constructing time series from two separate time series: one dominated by clutter and the other dominated by weather. In this way the weather and clutter powers of the combined time series were known a priori. Results of the analysis show that that 1) GMAP is a suitable choice for use in conjunction with the SZ algorithms, 2) the GMAP could be improved and fine tuned and 3) an algorithm such as SSEF is a possible alternative to GMAP.

NCAR also continued its analysis of phase noise of the experimental data from RVP8 in S-Pol data. A new improved SIGMET transmit card was installed and data was taken during NAME using the new card. The analysis shows that the phase stability has improved but still is not as good as it should be. SIGMET has indicated that their new 72 MHz IFD and their accompanying receiver card should improve the phase stability further.

An SZ-1 censoring algorithm was also developed and tested on experimental S-Pol phase coded data. SZ-1 censoring is fundamentally different from SZ-2 censoring since there is no
accompanying long PRT scan. SZ-1 censoring relies upon various texture estimates of the velocity and an estimate of the spectral quality of the weak trip recovered signals. Analysis shows that poor quality data is effectively eliminated while preserving interesting meteorological signatures such as tornadoes, convergent/divergent weather and gust fronts.
1 Introduction

There are three main areas of this year’s work: 1) the delivery of practical SZ-2 algorithm including censoring, 2) a comparison study of the SIGMET and NSSL/NCAR SZ algorithms and 3) the analysis and testing of SIGMET’s GMAP clutter filter which is reported in this document.

The first two items are the subject of separate interim reports which were submitted to the ROC earlier during FY04.

The results of the GMAP study are presented in section 2 of this report. This GMAP study was performed using experimental data. Radar time series data were sorted into weather-only and clutter-only categories. The time series were then combined so that both the weather and ground clutter powers were known a priori. Therefore the performance of the GMAP clutter filter can be precisely evaluated. For comparison purposes, NCAR also designed and tested a spectral clutter filter that is similar to GMAP.

There are three other areas of work included in this report:

- Section 3 presents work done on the continued development of SZ-1 censoring;
- Section 4 presents work on an investigation into phase coding for dual polarization radar.
- Section 5 presents the results of continued work on the RVP8/S-Pol phase stability analysis;

The SZ-1 analysis combines and develops the areal velocity variance technique with the weak trip spectral quality technique. These were reported in NCAR’s 2003 Annual Report to the ROC. It is shown that combining these estimates should lead to improved censoring performance. The phase stability or noise analysis is also a continuation of last years work. Phase noise of the RVP8/S-Pol system was improved this year by 1) use of SIGMET’s new transmitter card and the judicious choice of system-compatible PRT values. Spectral analysis is used to identify the source and frequency of the phase noise.
2 Evaluation of GMAP for use with SZ algorithms

2.1 Justification

This work was carried out to fulfill the following tasks in the FY2004 statement of work:

1. Evaluate Sigmet’s GMAP clutter filter for compatibility with SZ-2 phase coding.
2. Determine if other spectral clutter filter methodologies would be acceptable alternatives.

2.2 Overview

This work was organized into the following tasks:

- Demonstrate GMAP qualitatively by applying it to synthetic spectra.
- Develop the Spectral Stationary Echo Filter (SSEF) in order to have a filter against which to compare GMAP.
- Apply GMAP and SSEF to spectra reconstructed from real weather and clutter spectra.
- Verify the performance of the filters by comparing the results to ‘truth’ as determined from the original spectra.
- Investigate cases in which GMAP does not work well.
- Develop a simple system which combines the SSEF filter with a simplified Radar Echo Classifier (REC) and test this system on real data.

The organization of this report reflects these tasks, which are presented in the order stated above.

2.3 GMAP description.

GMAP is the **Gaussian Model Adaptive Filter** developed by Sigmet. The following description of GMAP was provided in correspondence from Alan Siggia of Sigmet:

GMAP is a complete spectral clutter filter algorithm based on a Gaussian clutter model and the noise statistics of the spectrum. Return value is the power removed from the spectrum, and the modified spectrum itself.

The GMAP clutter filter incorporates signal modeling, clutter modeling, and rank order spectral analysis to provide a robust filter that can operate over a wide range of conditions. GMAP works from an input spectrum containing unknown amounts of clutter, signal, and noise, and returns an output spectrum with the clutter influence removed. The algorithm requires only a single tuning parameter 'Wc' (the assumed clutter width in m/sec), and can be summarized as follows:

1. Convert the input spectral values to dB and sort the entire collection of points in order of increasing power. This gives us a rank ordered power spectrum, along with indices of each point in the original DFT.

2. Analyze the noise and signal characteristics of the rank ordered spectrum to deduce the noise power and spectral signal point, i.e., the point beyond which we seem to have signal statistics rather than noise statistics. Note that this uses the same set of procedures as the whitening algorithm for Random Phase mode.
3. Create a model for windowed Gaussian clutter having a normalized spectrum width based on the input value 'Wc' and the current wavelength and PRF. The clutter model also takes account of the particular window and spectrum size being used.

4. Calculate the power ratio between the mean of the middle three points of the clutter model and the middle three points of the original spectrum. Search the clutter model outward from DC for the point at which it drops below the noise level from #2. This tells us the width that clutter of this power would occupy within this spectrum, and defines the 'clutter gap' of points to be removed around zero velocity.

5. If we found a trusted signal cut in #2 from the rank ordered spectrum, then fit a Gaussian model to the spectral points that seem to represent valid weather signal. First calculate R0&R1 from the sparse spectrum containing only those 'intelligent' points, minus whichever ones are being discarded from the clutter gap.

6. Proceed with modeling and interpolating the clutter gap region of the spectrum. The procedure is an iterative one, in which we fill the gap with modeled points, and then recompute the model parameters based on the R0&R1 from #5, combined with these new trial points. Continue until no significant signal power is added back into the clutter gap, at which point we assume that we've reconstructed whatever original signal power was lost in the clutter gap.

NOTE: The algorithm can be modified to use an input noise level rather than a deduced noise level. In this case, steps #1 and #2 are skipped, and step #5 is modified to compute R0&R1 from the entire set of spectral points, with a noise correction then applied to R0.

2.4 Suitability of spectral clutter filters for SZ algorithms.

Traditional ground clutter filters have been of the IIR (Infinite Impulse Response) variety. These filters contain feedback terms and not only affect the magnitude but importantly the phase of the filtered time series. Since phase coding depends critically on the phase relationship of the time series members, it is intuitive that IIR filters are inappropriate for use with phase coded signals, at least if one desires to recover the moments of the weak trip signal when there is clutter in the strong trip. The biases of weak trip velocities were discussed and documented in NCAR’s 2003 Annual Range-Velocity Mitigation Report to the ROC available at, http://www.atd.ucar.edu/rsf/NEXRAD/index.html. Since spectral domain clutter filters operate on the magnitude of a time series’ spectrum leaving the phase intact, spectral clutter filters are a natural choice for use with SZ phase coding.

2.5 Testing clutter filters on reconstructed spectra.

In order to verify the effectiveness of a clutter filter, it is necessary to have some measure of the truth. The method chosen for this study was to construct ‘combined’ spectra from two component spectra, one containing predominantly clutter and one containing predominantly weather. In this way, the clutter and weather powers are known a-priori. The filter to be tested is applied to the combined spectrum, and the resulting spectrum, supposedly with clutter removed, is compared with the original weather spectrum from which the combined spectrum was created.
Creating combined spectra in this way also allows control of the Clutter-to-Weather-Ratio (CWR), i.e., the ratio of power in the clutter spectrum to the power in the weather spectrum. In the study the CWR was varied from 0 to 40 dB.

Two sources were identified for the spectral components: (a) synthetic spectra, in which spectra were created from a Gaussian model upon which simulated noise was superimposed, and (b) real spectra, which were obtained from SPOL time-series data. For the real spectra, the time series data was examined for regions of clutter largely free of weather, and for regions of weather largely free of clutter. These spectra were stored in a data base, and reconstructed spectra were formed by randomly selecting one clutter spectrum and one weather spectrum.

In summary, the following 2 methods were used for generating the spectra for testing:

(a) **Synthetic spectra:**

1. Generate a simulated weather spectrum, with specified power, velocity and width.
2. Generate a simulated clutter spectrum, with specified CWR, velocity and width.
3. Combine the two spectral components.

The individual spectra were simulated as a 1024-sample Gaussian spectrum with exponentially-distributed noise. This was inverted into the time domain, 64 samples were taken from the center of the time series, a Hanning window was applied and the result was converted back into the spectral domain.

(b) **Reconstructed real spectra:**

1. Randomly select a weather spectrum from suitable SPOL data.
2. Randomly select a clutter spectrum from suitable SPOL data.
3. Adjust the clutter spectrum power to a specified CWR.
4. Randomly shift the weather spectrum on the circle to ensure a uniform distribution of velocity values.
5. Combine the two spectral components.

### 2.6 Examples of synthetic spectra

The figures which follow show some examples of synthetic spectra. The examples demonstrate the challenge of separating clutter and weather when the weather has a velocity close to 0 and a narrow spectrum. Figure 2-1 shows a spectrum with a weather velocity of 20 m/s, so the two spectral peaks are well separated and are therefore relatively easy to identify. By contrast, Figure 2-3 shows a spectrum in which the weather has a velocity close to 0 and a narrow spectrum width, which makes the task of separating the clutter from the weather much more difficult.

In the figures, the red line shows the weather spectrum, the green line shows the clutter spectrum and the blue line shows the combined spectrum.
Figure 2-1 Re-constructed clutter/weather spectra CWR 20 dB. Clutter width 0.5 m/s. Weather velocity 20 /ms, width 2.5 m/s. Clutter peak and weather peak are easily separated.

Figure 2-2 Re-constructed clutter/weather spectra CWR 20 dB. Clutter width 0.5 m/s. Weather velocity 5 /ms, width 2.5 m/s. Clutter peak and weather peak closer together, but still distinct.
2.7 Creating a GMAP test environment at NCAR

In order to test GMAP on reconstructed spectra, it was necessary to develop a test environment at NCAR in which to run GMAP. This was done by combining code from the RVP8 with NCAR code. In addition, some routines were written to simulate functions available on the RPV8 and in the Intel Performance Primitives (IPPC) library.

Figure 2-4 shows how the system was set up. The module, *libgmap.a*, combines the Sigmet and NCAR code into a single library and is then used for linking with MATLAB or C++ code. Results from code running in the test environment were verified against results from the same code running on the RVP8 using only the Sigmet environment, to make sure that all of the testbed routines were working correctly.
2.8 Qualitative investigation of GMAP

As an initial step, GMAP was run on a number of spectra, and the results were examined qualitatively to assess the filter characteristics.

In the figures which follow, the weather spectrum is shown in blue and the combined spectrum (weather plus clutter) is shown in green. The notch applied by GMAP is shown in red.

Figure 2-5 shows the result of running GMAP on a combined spectrum in which the clutter peak and weather peak are well separated. The red notch, which GMAP computed, removes most of the clutter power. Ideally it would be more aggressive.

Figure 2-6 shows the result of applying GMAP to a spectrum in which there is considerable overlap between the weather and clutter, but the two peaks are nevertheless distinct. In this case GMAP performs very well.

Figure 2-7 shows an example in which the clutter and weather have similar spectrum widths (0.5 m/s) and the weather has a velocity close to 0. The weather signal combines with the clutter spectrum making it difficult to separate the two. Nevertheless, GMAP does a good job and produces a notch of the correct depth.

Figure 2-8 shows an example in which the clutter completely dominates the weather. GMAP notches out the main peak of the clutter spectrum, but leaves a high noise floor which overestimates the weather signal. This is a common problem with clutter filters, in which residual clutter remains because of the elevated noise floor.
Figure 2-6 Testing GMAP filter on simulated spectra. CWR 20 dB. Clutter width 0.5 m/s. Weather velocity 5 m/s, width 2.5 m/s. Clutter and weather overlaps considerably. GMAP performs well.

Figure 2-7 Testing GMAP filter on simulated spectra. CWR 20 dB. Clutter width 0.5 m/s. Weather velocity 1 m/s, width 0.5 m/s. Peaks are heavily overlapping. GMAP performs well.
2.9 SSEF – an alternative spectral clutter filter

NCAR developed the Spectral Stationary Echo Filter (SSEF), for the following reasons:

- The statement of work required the consideration of alternative filters;
- It was desirable to have a filter against which to compare GMAP in order to assess its performance.

SSEF differs from GMAP primarily in the way it applies the notion of a Gaussian model. GMAP applies a Gaussian model to the clutter and then interpolates using a Gaussian fit to the weather. By contrast, SSEF applies an aggressive rectangular notch, computes a Gaussian fit to the remaining signal and adjusts the notch to this Gaussian fit. SSEF repeats this procedure twice to converge on an optimum fit.

The first step in SSEF is to identify the clutter and weather peaks, if possible. The spectrum is divided into 8 parts, centered on 0 velocity, and the power for each part is computed. The main peaks are then identified. If there are 2 main peaks, the spectrum is classified as bi-modal. Figure 2-9 shows an example of this procedure. The clutter and weather peaks are shown in blue.
Figure 2-9 SSEF initialization: check power distribution. Divide the spectrum into 8 parts. Compute power for each part. Check for bimodal spectrum. Identify clutter and weather peaks.

The next step is to apply an aggressive rectangular notch around the clutter peak, shown in red in Figure 2-10. This is the ‘first guess’ notch. A Gaussian fit is performed on the remaining spectrum, i.e., after applying the notch, and is shown as a dotted magenta line.

Figure 2-10 SSEF filter, first guess: Apply aggressive notch around the clutter peak. Fit a Gaussian to the remaining power.

The ‘second-guess’ notch is determined by the intersection of the Gaussian fit with the spectrum around the clutter peak. This notch is shown in red on Figure 2-11.
The procedure is repeated once more to produce the 'final guess' notch, as shown in Figure 2-12. In this figure you will also notice that a lower bound is applied to the Gaussian fit. The lower bound is based on an estimate of the noise floor.

Figure 2-12 through Figure 2-15 show the final SSEF guess for the spectra on which GMAP was demonstrated in the previous section. (See Figure 2-5 through Figure 2-8). SSEF performs
comparably to GMAP in most cases. It applies a more aggressive notch to the first case, producing a better result.

Figure 2-13 SSEF filter example – simulated spectra. CWR 20 dB, Clutter width 0.5 m/s. Weather velocity 5 /ms, width 2.5 m/s.

Figure 2-14 SSEF filter example – simulated spectra. CWR 20 dB, Clutter width 0.5 m/s. Weather velocity 1 m/s, width 0.5 m/s.
Figure 2-15 SSEF filter example – simulated spectra. CWR 10 dB, Low SNR.
Weather velocity 20 m/s, width 2.5 m/s
2.10 Quantitative testing on reconstructed ‘real’ spectra.

In order to examine the performance of the filters quantitatively, sets of 2000 combined spectra were reconstructed from the SPOL clutter and weather data sets, using the procedure detailed in section 2.5(b). 5 sets of spectra were used, for CWR values of 0, 10, 20, 30 and 40 dB.

2.10.1 Quantitative performance – weather power

Figure 2-16 through Figure 2-25 show the results of running GMAP and SSEF on the 2000 reconstructed spectra, computing the remaining (weather) power and plotting this vs. the ‘truth’. The truth is assumed to be the power from the weather spectrum prior to combining it with the clutter spectrum.
Figure 2-16 GMAP filtered power vs. truth, CWR 0 dB.
GMAP shows some positive and negative errors.
Large positive errors indicate GMAP is adding power to some spectra.

Figure 2-17 SSEF filtered power vs. truth, CWR 0 dB.
SSEF shows some negative errors,
indicating an over-aggressive notch in some circumstances.
Figure 2-18 GMAP filtered power vs. truth, CWR 10 dB.
GMAP making positive errors up to 20 dB above the 1:1 truth line.
This indicates GMAP is adding power to some spectra.

Figure 2-19 SSEF filtered power vs. truth, CWR 10 dB.
SSEF showing some negative errors, which indicate an over-aggressive notch.
Figure 2-20 GMAP filtered power vs. truth, CWR 20 dB.
GMAP showing large positive errors up to 30 dB above 1:1 truth line. The filter is not removing the clutter power in those cases.

Figure 2-21 SSEF filtered power vs. truth, CWR 20 dB.
SSEF producing errors evenly distributed around 1:1 line. Less scatter than GMAP.
Figure 2-22 GMAP filtered power vs. truth, CWR 30 dB.  
GMAP producing large positive errors, indicating that it is not removing the clutter power at these points.  
For smaller errors, residual clutter power is probably responsible.

Figure 2-23 SSEF filtered power vs. truth, CWR 30 dB.  
SSEF starting to have large positive errors, showing that it is also not removing clutter power at those points.  
For smaller errors, residual clutter power is probably responsible.
Figure 2-24 GMAP filtered power vs. truth, CWR 40 dB.
Large positive errors predominate.
Almost no points line on the 1:1 truth line.
Residual clutter power responsible for some of the scatter.

Figure 2-25 SSEF filtered power vs. truth, CWR 40 dB.
Large errors predominate.
However, still a considerable cluster of points near the 1:1 truth line.
Residual clutter power responsible for some of the scatter.
2.10.2 Quantitative performance - velocity

Figure 2-26 through Figure 2-35 show plots of velocity as estimated from the filtered spectra, vs. the ‘truth’ velocity from the original weather spectrum.

Figure 2-26 GMAP filtered velocity vs. truth, CWR 0 dB.
GMAP performs well.

Figure 2-27 SSEF filtered velocity vs. truth, CWR 0 dB.
SSEF shows some scatter close to the (0,0) point, i.e., where weather has velocity of 0.
Figure 2-28 GMAP filtered velocity vs. truth, CWR 10 dB.
GMAP starting to miss some significant clutter.
Hence the horizontal line through 0 m/s, in which clutter dominates the velocity.

Figure 2-29 SSEF filtered velocity vs. truth, CWR 10 dB.
Some limited scatter
SSEF shows little evidence of completely missing clutter.
Figure 2-30 GMAP filtered velocity vs. truth, CWR 20 dB.
GMAP starting to miss quite a number of significant clutter points. Horizontal line is more marked.

Figure 2-31 SSEF filtered velocity vs. truth, CWR 20 dB.
SSEF starting to miss significant clutter. Scatter is increasing.
Figure 2-32 GMAP filtered velocity vs. truth, CWR 30 dB.
GMAP missing large number of significant clutter points.
The reason for the multiple horizontal lines is not known.

Figure 2-33 SSEF filtered velocity vs. truth, CWR 30 dB.
SSEF missing more significant clutter points.
Still performing better than GMAP.
Figure 2-34 GMAP filtered velocity vs. truth, CWR 40 dB.
GMAP missing most of the clutter power. Most points appear in the horizontal features. Almost no points left on the 1:1 truth line.

Figure 2-35 SSEF filtered velocity vs. truth, CWR 40 dB.
SSEF missing a large fraction of the clutter power. However, for velocities well away from 0, for which the clutter peak and weather peak are well separated, there are still many points on the 1:1 line.
2.10.3 Quantitative performance – spectrum width

Figure 2-36 through Figure 2-45 show plots of spectrum width as estimated from the filtered spectra, vs. the ‘truth’ values from the original weather spectrum. Both filters perform reasonably well for CWR values up to 20 dB. For CWR values of 30 dB and above, the scatter is large.

Figure 2-36 GMAP filtered spectrum width vs. truth, CWR 0 dB.
Good performance. Limited scattered about the 1:1 truth line.

Figure 2-37 SSEF filtered spectrum width vs. truth, CWR 0 dB.
Good performance. Limited scattered about the 1:1 truth line.
Figure 2-38 GMAP filtered spectrum width vs. truth, CWR 10 dB. Scatter increasing, GMAP still performing well.

Figure 2-39 SSEF filtered spectrum width vs. truth, CWR 10 dB. Scatter increasing, SSEF still performing well.
Figure 2-40 GMAP filtered spectrum width vs. truth, CWR 20 dB.
Results clustered around the 1:1 truth line. Filter still performing well. Scatter increasing.
Some horizontal line features showing up at filtered width values of about 0.2 and 0.75 m/s.
The cause of these is not known.

Figure 2-41 SSEF filtered spectrum width vs. truth, CWR 20 dB.
Results clustered around the 1:1 truth line.
Filter still performing well. Scatter increasing.
Figure 2-42 GMAP filtered spectrum width vs. truth, CWR 30 dB.
1:1 truth line less evident. Scatter high. Horizontal line features show up at filtered width values of about 0.75 and 1.2 m/s. The cause of this is unknown.

Figure 2-43 SSEF filtered spectrum width vs. truth, CWR 30 dB.
1:1 truth line less evident. Scatter high. No evidence of horizontal line features.
Figure 2-44 GMAP filtered spectrum width vs. truth, CWR 40 dB.
Filter performing badly. No evidence of 1:1 truth line. Horizontal line feature at a filtered width of 1.2 m/s dominates. The cause of this is not known.

Figure 2-45 SSEF filtered spectrum width vs. truth, CWR 40 dB.
Filter performing marginally. Scatter large. 1:1 truth line still evident, but not strong. Horizontal line feature at a width of about 0.5 m/s.
2.11 Case study of GMAP – under-estimation of clutter power.

It was noticed that for some spectra, GMAP underestimates the clutter power, and applies a notch which is neither deep nor wide enough.

Figure 2-46 through Figure 2-55 show examples of running GMAP and SSEF on a case which demonstrates this effect. The underlying clutter and weather spectra are the same for all plots. The clutter-to-weather ratio (CWR) varies, from 0 to 40 dB, in increments of 10 dB.

As can be seen from the plots, for some reason GMAP significantly under-estimates the depth of the notch required to remove the clutter power, and in one case actually increases the clutter peak. It is not know why GMAP has this behavior on these spectra but not on other spectra which are similar in nature.
Figure 2-46 Applying GMAP to reconstructed spectrum, CWR 0 dB. GMAP underestimates the notch depth.

Figure 2-47 Applying SSEF to reconstructed spectrum, CWR 0 dB. SSEF over-estimates the notch depth, but this is not a serious error.
Figure 2-48 Applying GMAP to reconstructed spectrum, CWR 10 dB. GMAP underestimates required notch depth.

Figure 2-49 Applying SSEF to reconstructed spectrum, CWR 10 dB. SSEF does well on estimation of the notch depth.
Figure 2-50 Applying GMAP to reconstructed spectrum, CWR 20 dB. GMAP performs an odd reconstruction, which actually increases the clutter peak. The reason for this behavior is not known.

Figure 2-51 Applying SSEF to reconstructed spectrum, CWR 20 dB. SSEF performs OK on the notch. Residual clutter power in the remainder of the spectrum is a problem.
Figure 2-52 Applying GMAP to reconstructed spectrum, CWR 30 dB. GMAP severely underestimates clutter power. Residual clutter power in the remainder of the spectrum is also a problem.

Figure 2-53 Example of applying SSEF to reconstructed spectrum, CWR 30 dB. SSEF performs well on the notch. Residual clutter power in the remainder of the spectrum is a problem.
Figure 2-54 Applying GMAP to reconstructed spectrum, CWR 40 dB.
GMAP severely underestimates clutter power.
Residual clutter power in the remainder of the spectrum is a problem.

Figure 2-55 Applying SSEF to reconstructed spectrum, CWR 40 dB.
SSEF performs well on the notch.
Residual clutter power in the remainder of the spectrum is a problem.
Figure 2-56, below, shows a similar problem for GMAP, in which it severely under-estimates the depth of the notch required to reduce the clutter power.

![Example of GMAP error – notch too shallow](image1)

**Figure 2-56 Example of GMAP error – notch too shallow**

![SSEF applied to the same spectrum as above figure.](image2)

**Figure 2-57 SSEF applied to the same spectrum as above figure.**
2.12 Performance of GMAP and SSEF on bi-modal weather spectra.

When clutter is present and the weather spectrum itself is bi-modal, it is possible to have a combined spectrum with 3 peaks. This situation is difficult for the filters to deal with, since they both make assumptions about the Gaussian nature of the spectra. Figure 2-58 through Figure 2-63 show the results of applying GMAP and SSEF to spectra with bi-modal weather.

**Figure 2-58 Example 1 of applying GMAP to a bi-modal spectrum, CWR 20 dB.**
GMAP performs well, removes main clutter peak. Leaves the secondary weather peak.

**Figure 2-59 Example 1 of applying SSEF to a bi-modal spectrum, CWR 20 dB.**
SSEF performs well, similarly to GMAP.
Figure 2-60 Example 2 of applying GMAP to a bi-modal spectrum, CWR 20 dB. GMAP performs well, removing the main clutter peak, leaving the secondary weather peak.

Figure 2-61 Example 2 of applying SSEF to a bi-modal spectrum, CWR 20 dB. SSEF performs well, similarly to GMAP.
Figure 2-62 Example 3 of applying GMAP to a bi-modal spectrum, CWR 20 dB. GMAP performs quite well, leaving most of the weather power.

Figure 2-63 Example 3 of applying SSEF to a bi-modal spectrum, CWR 20 dB. SSEF does not perform as well as GMAP, and removes almost all of the weather peak next to the clutter peak.

The performance of clutter filters in the bi-modal case may be moot, since it is not clear that the moments estimation algorithms will perform well on a bi-modal spectrum anyway.
2.13 Applying SSEF to SPOL data, using a simplified Radar Echo Classifier.

A major problem associated with all clutter filters, including those which operate in the spectral domain, is that if the weather signal has properties similar to clutter, the weather is removed in error. This is particularly true for stratiform rain which can have a narrow spectrum width and a velocity close to zero when the weather advection vector is normal to the azimuth line from the radar to the weather.

A possible solution is to run the Radar Echo Classifier (REC) first, and use the results of the REC to identify regions of weather on which the clutter filter should not be run.

As a first step in demonstrating this procedure, a highly simplified version of the REC was developed. This simple REC used the smoothness of the reflectivity field along a beam to identify regions in which weather is present, on the assumption that weather has a smoother spatial reflectivity field than does clutter.

SSEF was coded to include this simplified REC, and the filter was then run on some long-PRT time-series data which was collected at the SPOL Marshall site.

Figure 2-64 and Figure 2-65 show the reflectivity before and after the application of SSEF. Much of the clutter is removed, though some residual clutter power remains in regions of strong clutter.

Figure 2-66 and Figure 2-67 show the plots for velocity, before and after applying SSEF. The boundary feature SE of the radar is considerably enhanced after the clutter has been removed. Folding occurs at about 8 m/s because of the long PRT.

Figure 2-68 and Figure 2-69 show the plots for spectrum width, before and after applying the filter. The clutter areas show up as having very low spectrum widths, and most of these areas are removed by the filter.
Figure 2-64 SPOL reflectivity – no filter.

Figure 2-65 SPOL reflectivity – after SSEF filter.
Figure 2-66 SPOL velocity – no filter.

Figure 2-67 SPOL velocity – after SSEF filter.
Figure 2-68 SPOL spectrum width – no filter.

Figure 2-69 SPOL spectrum width – after SSEF filter.
2.14 Conclusions

- Because it works in the spectral domain, GMAP is suitable for use with the SZ algorithms.

- On certain spectra, GMAP sometimes produces odd results. These show up as a notch which is too shallow, and sometimes even a reconstructed notch which increases the clutter peak. The reason for this behavior is not known.

- The performance of SSEF relative to GMAP suggests that GMAP is not optimal in its present form and that perhaps GMAP could be tuned or altered slightly to improve its performance.

- SSEF is a possible alternative to GMAP.
3 SZ-1 Censoring

The SZ-1 algorithm requires a different censoring algorithm than SZ-2 as described in the June, 2004 Interim Report. The SZ-1 algorithm has only the short PRT phase coded data from which to determine the trip order. The PRT is chosen such that only two overlaid weather echoes are possible. The power ratio and out-of-trip contamination cannot be accurately computed using only the short range scan. This is because the power leaking from one trip to another is unknown and contaminates the power ratio and out of trip contamination estimates. Thus, another censoring strategy must be used in the absence of the information from a long PRT scan. When the SZ-1 algorithm fails, the result is an increase in the areal variance of the velocity field (i.e., velocities become noisy). Because weather echoes typically have a much smaller areal velocity variance (i.e., they appear “smooth”), the noisy characteristics of bad radial velocity estimates can be identified and exploited for censoring. Also, estimates of the quality of the spectra of reconstructed weak trip echoes are used for censoring. If the power in the reconstructed spectra is spread to much, as a function of velocity, the echo is judged to be likely non meteorological and thus the data likely should be censored. This spectral quality estimate is used in conjunction with the areal velocity variance estimates in a fuzzy logic algorithm to determine censoring.

3.1 Method

A fuzzy logic algorithm has been developed to identify contaminated radial velocity estimates. The algorithm combines spatial information from the radial velocity field with information derived from weak trip spectra. The algorithm has been designed to avoid censoring weather phenomena that can have high areal velocity variances. The surface divergence of microbursts, the convergence of outflow boundaries, and the rotation of mesocyclones and tornadoes are examples of weather events that might have a high spatial velocity variance.

Several parameters, or feature fields, are computed from the radial velocity data and input into the fuzzy logic algorithm. A small region of data is used to compute the feature fields, which include the radial texture, azimuth texture, radial spin and azimuth spin.

The range texture is computed as,

\[
\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} (\text{range\_difference}_{i,j})^2}{M \times N},
\]

where range_difference is the gate to gate difference in range of the radial velocity, i and j are the range and azimuth indices, M and N are the number of gates in the azimuth and range direction respectively. The azimuth texture estimate is the same as the range texture estimate only computed for gate to gate differences in azimuth.

The range and azimuth spin are adopted from Steiner et al (2002) and are a measure of the number of times the slope, or first derivative, in the velocity changes sign in either the range or azimuth direction. To compute range spin, we simply count the number of times within the data region the trend in radial velocity changes from increasing in range to decreasing in range, or decreasing in range to increasing in range. Azimuth spin is computed similarly in the azimuth
direction. It is important to set a minimum threshold in the change of radial velocity to be counted as a spin change in order to erroneously avoid counting measurement noise.

The texture and spin feature fields are computed separately in range and azimuth to distinguish between the noise of invalid velocity data and weather signals that may appear noisy. Two dimensional fields would appear noisy for strong gradients, convergence, divergence or rotation. These signatures in radial velocity, however, generally result in high texture and spin in either the range or azimuth direction. However, contaminated velocity data leads to high texture and spin in both directions. It is also important to map the radial velocity onto a circle from 0 to $2\nu_N$ before computing texture and spin in order to avoid erroneous censoring when the velocity folds.

Experiments have been performed to determine the optimal size of the computation region for the texture and spin fields. Currently the computations in azimuth use 7 beams and 5 range bins and in range use 3 beams and 7 range bins. These values should be adaptable to facilitate optimization.

The spectra quality estimate is based on the assumption that weather signals should have the majority of their power concentrated around some mean velocity. In the present scenario, weak trip echoes will have eight replicas in the frequency domain when cohered to the strong trip. Upon notch filtering and recohering the weak trip, the weak trip spectra will display a central or primary “bump” with three sidebands on each side (Sachidanada and Zrnic 1999). After deconvolution, the power of the sidebands should be relocated at the location of the primary “bump”. If this is not the case, i.e., significant power remains in the sidebands, then the weak trip is considered “contaminated”. The weak trip moments are likely bad and should be censored. In order to identify contamination at the spectral level for weak trip echoes, the Nyquist interval is broken into eight equal partitions, with one bin centered on the estimated velocity. The peak power in each of the eight spectral bins is computed and sorted from largest to smallest. The average power of the three weakest bins is divided by the power in the strongest bin, which contains the estimated velocity. For relatively clean spectra, this ratio should be small and the larger the ratio gets the more out of trip leakage is present.

The five feature fields described above are combined in a fuzzy logic context to determine if the data are to be censored. First, membership functions are designed to map each feature field to a range between 0 and 1, with 1 indicating strongest agreement and 0 no agreement. The output of the membership functions is called the interest field. In our case, an interest of 1 indicates velocity data to be censored and 0 indicates valid data. The S-shaped range spin membership function is plotted in Figure 3-1. Weather echoes typically have spin values near zero and the likelihood, and thus interest value, of contamination increases as the range spin increases. Notice the membership function is not a boundary threshold but allows for varying degrees of interest between spin values of 0 and 0.15. The azimuth and range spin feature fields use the same membership function.
The membership functions for all five feature fields have shapes similar to Figure 4. The membership functions are computed as,

\[
mf(x) = \begin{cases} 
2 \left( \frac{x - x_0}{x_1 - x_0} \right)^2, & x_0 < x < \bar{x} \\
1 + (-2) \left( \frac{x - x_0}{x_1 - x_0} \right)^2, & \bar{x} < x < x_1 \\
0, & x < x_0 \\
1, & x > x_1,
\end{cases}
\]

where \(x_0\) and \(x_1\) are the values of the membership function for interests of 0 and 1 respectively and \(\bar{x}\) is the midpoint between \(x_0\) and \(x_1\). The \(x_0\) and \(x_1\) values for the 5 feature fields used are listed in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>Range spin</th>
<th>Azimuth spin</th>
<th>Range texture</th>
<th>Azimuth texture</th>
<th>Spectral ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0</td>
<td>0</td>
<td>0</td>
<td>40</td>
<td>40</td>
<td>0.15</td>
</tr>
<tr>
<td>X1</td>
<td>0.15</td>
<td>0.15</td>
<td>70</td>
<td>70</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 2: X0 and X1 values for feature fields.

The range and azimuth texture interest fields are combined into a single texture interest field using a fuzzy logic “and” operator. This is done because invalid velocity data are noisy in both
the range and azimuth directions, while certain weather phenomena appear noisy in either range or azimuth. Experiments were made using 2 different “and” operators, namely the minimum and the product of the two appropriate interest fields. It was found that the minimum operator yielded better separation of valid and invalid velocity values and is used in this study. The two spin interest fields are combined into a single spin interest field in the same way as the texture.

The three interest fields for censoring velocity (combined texture, combined spin and spectral ratio) are multiplied by a-priori weights, summed and divided by the sum of the weights to normalize the final weighted sum. Experiments with the weights showed that the best values are simply 1 for all three input fields. The final step, and hence the censoring decision, is made by thresholding the weighted sum at 0.5. If the weighted sum is greater than 0.5, the gate is censored.

The SZ-1 method has been designed in order to deal with missing data within the PPI scans. A minimum amount of data must exist within the 2-D region, otherwise the range gate in question is censored. This is done to account for any other censoring that may occur prior to the fuzzy logic algorithm.

3.2 SZ-1 censoring results

Figure 3-2 shows the SZ-1 recovered data for a time series scan collected using the RVP-8 processor on the NCAR S-Pol radar. The SZ-1 algorithm was performed offline using the Matlab IMAT software. The black azimuth rings denote the ends of the first and second trips.

![Figure 3-2 Short PRT phase coded PPI of SZ-1 recovered radial velocity (ms⁻¹).](image)
The results of applying the SZ-1 censoring algorithm to the data in Figure 3-2 are presented in Figure 3-3. The regions without significant SNR in the short PRT scan are plotted as white and the data with sufficient SNR that were censored are indicated by purple. It can be seen that the contaminated velocity data are successfully censored using the SZ-1 fuzzy logic censoring algorithm. It can also be seen that in areas of overlaid weather echoes the SZ-1 censoring method removes a small amount of data surrounding the contaminated data. This small expansion of the censored region is not surprising given the 2-D nature of the texture and spin feature fields described in section 3.2. This effect is reduced by increasing the weight of the spectral ratio feature field, which is computed at each range gate. However, small regions of contaminated data leak through using an increased spectral ratio weight.

\begin{figure}
\centerline{\includegraphics[width=\textwidth]{figure3-3.png}}
\caption{Short PRT phase coded PPI of SZ-1 recovered radial velocity (m/s) after censoring.}
\end{figure}

It is possible to apply the current WSR-88D short-range scan censoring method to the data above, because a long PRT scan was also performed. The results are presented in Figure 3-4, and can be compared to Figure 3-3 (SZ-1). Clearly SZ-1 results in a significant improvement in velocity recovery area.
Figure 3-4. Short PRT PPI of radial velocity after the current WSR-88D censoring algorithm is applied with a 5 dB threshold (default setting).

The new SZ-1 censoring algorithm was applied to a tornadic supercell case. This case was generated by applying the SZ(8/64) phase code to two non-phase coded PPIs for first and second trip. The two scans were then overlaid and separated using the SZ-1 algorithm. In this case the moments from the original time series can be used to compute the error associated with the SZ-1 algorithm, which can be used to validate the censoring. The strong rotation from the mesocyclone or tornado is a good test of the fuzzy logic to separate contamination from weather echoes that appear to have high spatial variance in the radial velocity estimate.

Figure 3-5 shows a PPI of the SZ-1 recovered radial velocity with a circle drawn around the rotation signature. Folding of the radial velocity is evident south of the rotation signature, illustrating the importance of computing the texture and spin feature fields on the 0 to $2v_N$ circle. Without taking the folded velocity values into account, the spin and texture fields would be artificially high in the folded regions, resulting in the unnecessary censoring of valid velocity estimates.
To illustrate the fuzzy logic technique, each of the feature fields for the data shown in Figure 3-5 are presented. The azimuth texture is displayed in Figure 3-6 and the range texture is shown in Figure 3-7. Notice that in the region of the rotation signature the azimuth texture exceed 200. These values result in an interest value of 1 when the membership function is applied, indicating a strong likelihood of censoring from the azimuth texture feature field. It is not surprising that the azimuth texture is high for the rotation signature given the large the beam-to-beam change in radial velocity. The range texture in this case is quite low and is less than about 20 resulting in an interest field near 0, indicating no censoring, for the range texture feature field.
Figure 3-6. PPI of the computed azimuth texture.

Figure 3-7. PPI of the computed range texture.
The results for the azimuth spin (Figure 3-8) and the range spin (Figure 3-9) are similar to those for the texture. The azimuth spin is high and has an interest of 1 in the rotation signature indicating censoring, while the range spin is zero, indicating no censoring.

Figure 3-8. PPI of the computed azimuth spin.
After the fuzzy logic censoring method is applied, the data in the rotation signature is (correctly) not censored (see Figure 3-10). This is a result of the combination of the azimuth and range texture and spin fields with the fuzzy logic and operator, which meets the requirement that censored data be noisy in both directions. The censoring results in Figure 3-10 can be compared to the error in the SZ-1 recovered velocities in Figure 3-11. Recall that the errors are the differences of the original non-phase coded velocity estimates and the SZ-1 recovered velocity estimates. It can be seen that all of the censoring regions are characterized by large errors and that the fuzzy logic censoring algorithm does not erroneously censor regions of folded velocity or the rotation signature. One weakness of the SZ-1 censoring which is evident is isolated range gates or very small patches with high errors. Perhaps when the spectral variable is added to the analysis of this data set that problem will be mitigated. If not, research into a viable solution should continue.
Figure 3-10 Short PRT phase coded PPI of SZ-1 recovered radial velocity (ms\(^{-1}\)) after censoring.

Figure 3-11. The SZ-1 radial velocity errors (ms\(^{-1}\)).
4 SZ Phase Coding For Dual Polarization

Phase coding of the transmit pulse for dual polarization radar is relatively straightforward. However, complications do arise and single polarization SZ algorithms may need to be modified accordingly. Shown in Figure 4-1 is a diagram of the $i^{th}$ transmit pulse and $i^{th}$ time series sample corresponding to a delay of $t$ after the transmit pulse.

![Diagram of the time series for the simultaneous transmit dual polarization scheme to be used for NEXRAD radars.](image)

The diagram indicates that H and V polarization pulses are transmitted simultaneously as is planned for the dual polarization upgrade of NEXRAD radars. Note in this example that the H and V pulses have distinct and separate phase codes applied, $\phi_i$ and $\psi_i$, respectively. For single transmitter radars, such as the NWS WSR-88Ds, these phases will not be set separately. For these phase codes to be different hardware would be required that could apply phase shifts to the transmit pulses down-stream from the klystron amplifier in the high power, RF frequency path of the transmitted signal. Currently there is no technology available that can do this accurately. Thus, for the NEXRAD radars, $\phi_i = \psi_i$ unless two transmitters were used.

At time $t$ after the $i^{th}$ pulse the H and V received signals are

$$R_h(i) = hh_i \phi_i + vh_i \psi_i$$

(1)

$$R_v(i) = vv_i \psi_i + hv_i \phi_i$$

(2)

Each received signal has a copolar component, $hh$ or $vv$, and a cross polar component, $vh$ or $hv$. The crosspolar components are contaminating signals. For most weather targets, the crosspolar signals will be -20 dB or greater down from the corresponding copolar signal and thus, the copolar signal can be recovered with minimal degradation due to the contaminating crosspolar signal. Exceptions to this will be regions of large wet hail where LDR (linear depolarization ratio, defined as the ratio of the crosspolar power to copolar power) can be -15 dB and in clutter where LDR can be -10 dB or larger. Thus, the stronger copolar signal will be somewhat contaminated by the crosspolar return signals that result from the simultaneous orthogonal transmitted pulse. In general, this contamination will result in a small increase in variance of the copolar moments.

However, the copolar power can also be biased. If the crosspolar power is -10dB down from the copolar signal (i.e., $LDR = -10dB$), the biases can be significant and for this case, the copolar reflectivity will be 0.4 dB high if the signal copolar and crosspolar signals are uncorrelated. This bias will be higher when the crosspolar signal is significantly correlated to the copolar signal which is typically the case for clutter or large depolarizing scatterers.
Let $P_1$ and $P_2$ be the powers of a stronger and weaker scattered signal, respectively. Let $\mu$ be the complex correlation coefficient between the two signals. It can be shown that the total power of the combined signal is

$$P_t = P_1 + P_2 + 2 \sqrt{P_1 P_2} \, \text{Re}\{\mu}\quad (3)$$

For Rayleigh scatterers, $\text{Im}\{\mu\}$ is nearly zero and the magnitude of $\mu$ is quite small (0.1 to 0.2 typically). The phase of $\mu$ is termed the phase shift upon backscatter. For clutter this backscatter phase can be most anything and thus for clutter the $\text{Im}\{\mu\}$ will in general be non zero. However, for the following analysis we assume $\mu$ is real. The total power of the combination of two uncorrelated signals (such as meteorological scatter from different resolution volumes) is simply $P_1 + P_2$ as Eq. (3) shows. But if the signals are correlated the total power can increase above this nominal value. Shown in Figure 4-2 is the total power of two combined signals as a function of the power of the weaker of the two signals with the correlation coefficient $\mu$ as a parameter.

![Figure 4-2](image)

**Figure 4-2 The total power of the addition of two correlated signals.**

Let $P_1$ be the desired copolar power while $P_2$ is the contaminating overlaid crosspolar power. $P_1$ is at 0 dB so that the vertical axis can be interpreted as the amount of bias to $P_1$ power in dB. As can be seen from the figure, as the correlation between the two signals increases the bias increases. This is not large problem for SZ phase coding and data quality for NEXRAD. However, if the copolar power is increased by a dB or two due to the crosspolar signal, then this will decrease somewhat the recovery region for weaker trip moments. Also in hail if LDR= -15 dB and the co-to-cross correlation is 0.6, then the copolar reflectivity is biased by about 1 dB.
The effect of the contaminating crosspolar signal on the copolar velocity estimates should be minimal in precipitation since 1) LDR is low and 2) the mean velocity of the crosspolar signal is usually very close to the copolar velocity and thus should not bias velocity estimates. If second trip echoes are also present, then there will be three competing echoes. For example Eq.(1) becomes

\[ R_h(i) = hh_i \Phi_i + vh_i \Psi_i + hh_{i-1} \Phi_{i-1} \]  

(4)

where the crosspolar weak second trip signal has been ignored. Fortunately, since the strong trip copolar and crosspolar signals will have similar frequency content (at least in most cases), SZ phase coding should still be effective. That is, the stronger trip signal and its accompanying crosspolar counterpart will both have similar velocities so that a SZ notch filter should effectively attenuate both when trying to estimate weak trip velocity.

**Two Transmitter Configuration**

If the two simultaneously transmitted pulses could be phase coded separately, then it would be possible to estimate the moments of the weaker crosspolar signal. Since velocity and spectrum width are estimated from the stronger copolar signal, the additional information from the width and velocity of the crosspolar signal is minimal though the velocity and spectrum width of the crosspolar signal can be different from the copolar estimated moments. The estimation of the crosspolar power would, however, yield LDR which is a measure of the precipitation particles asymmetry. It has been argued that the copolar correlation coefficient, \( \rho_{hv} \), also contains the information about particle asymmetry and thus can act as a proxy for LDR. (The additional cost of having a second transmitter for the NEXRAD radars has been deemed to costly for the additional benefit of LDR measurements.) Two transmitters would also allow for the measurement of both crosspolar signals and thus make possible for the automated \( Z_{dr} \) calibration technique proposed by Hubbert et al (2003). Two transmitters would also increase the sensitivity by 3 dB over the single transmitter configuration.
5 RVP8/S-Pol System Phase Stability Analysis

5.1 Introduction

There are three aspects of the transmitted phases that are relevant to the SZ(8/64) scheme: the burst pulse phase measurement, the pulse-to-pulse phase stability, and the accuracy/precision of the burst pulse phase angle compared to the requested angle. If the burst pulse phase measurement is poor, then the phase errors introduced when cohering effectively reduce the signal to noise ratio as well as affect the SZ(8/64) modulation code, possibly degrading weak trip recovery. The pulse-to-pulse phase stability only becomes important if the measured burst phase is not used for cohering. In this scenario, poor phase stability has deleterious effects on both spectral and pulse pair estimates of the spectrum moments. Finally, the accuracy and precision of the transmitted burst pulse phase angle with respect to the requested phase angle, can affect the recovery of the weaker trip even if the burst pulse angles are accurately measured. Of these three aspects, we focus only on the latter two.

Let the $i^{th}$ requested or theoretical transmit phase shift be denoted by $\psi_i^{r}$ and set $\phi_i^{r} = \psi_{i+1}^{r} - \psi_i^{r}$. For this report, $\phi_i^{r}$ will be referred to as the requested or theoretical transmit phase difference.

Let the $i^{th}$ transmitted phase shift be denoted by $\psi_i$, and let the transmitted phase difference be defined as $\phi_i = \psi_{i+1} - \psi_i$. In the case of SZ(8/64), $\psi_i^{r}$ is the theoretical switching code and $\phi_i^{r}$ is essentially the modulation code (the true SZ(8/64) modulation code is equal to $-\phi_{i-1}^{r}$). The measured transmitted phase shift, denoted $\hat{\psi}_i$, is obtained by taking the complex argument of the measured burst pulse I&Q which, for the RVP8, is stored in the first range gate. This will be referred to as the measured burst phase. The measured phase difference is defined as $\hat{\phi}_i = \hat{\psi}_{i+1} - \hat{\psi}_i$.

In this study, an analysis of $\hat{\psi}_i$ and $\hat{\phi}_i$ is performed on the current S-POL configuration (July 20, 2004) for both the so-called SZ(8/64) mode and the standard non-phase coding mode. For reference, an analysis is also performed on the S-POL configuration as of July 7, 2003 in SZ(8/64) mode. Two changes were made to the system between these times: first, a new transmit card was installed and second, in 2004, S-POL was run using pulse repetition frequencies (PRF) that were integer-divisible by the various clocks in the system.
Figure 5-1: Time-series of the measured phase angles $\hat{\psi}_i$, in degrees, over a 90 degree PPI scan when the radar was in standard mode (i.e. $\psi_i^\prime = 0$). The red line shows the same data after a 65 point mean filter. The filtered data are denoted $\bar{\psi}_i$.

**5.2 Standard Mode - July 20, 2004**

The PRT (pulse repetition time) for the July 20, 2004 data was 1 ms. Figure 5-1 shows a time-series plot of measured phase angles, $\hat{\psi}_i$, for raw data (blue) and after applying a 65-point mean filter (red), both in degrees, over a 90 degree PPI scan. These filtered phase angles, denoted $\bar{\psi}_i$, are used for de-trending.

Note that errors introduced from the measurement of the burst pulse ($\psi_i^\prime - \hat{\psi}_i$) as well as system transmit phase errors ($\psi_i^\prime - \psi_i$), are represented in the randomness of the time-series data, but their relative contributions are not known exactly. However, the RVP8 burst pulse measurement error is about 0.1 degrees (for klystron systems) and thus the phase error analysis presented here should be dominated by other system aspects.

In standard mode, there is a small positive large-scale trend over the 90 degree PPI scan (about 15 seconds) in addition to the short-scale variability. In order to break up the longer-scale trends of the same order as a beam (64 pulses) the data are de-trended using a 65-point mean filter. As
noted above, the filtered time-series is plotted in red in Figure 5-1 and the de-trended time-series data are displayed in Figure 5-2.

Figure 5-2: Time-series of the de-trended measured burst phase angles \((\hat{\psi}_i - \bar{\psi}_i)\), in degrees, over a 90 degree PPI scan when the radar was in standard mode (i.e. \(\psi'_i = 0\)).

Figure 5-2 shows the de-trended measured burst pulse phase angles \((\hat{\psi}_i - \bar{\psi}_i)\). The values are mostly between ±1.5 degrees.
Figure 5-3: Histogram of the de-trended measured burst phase angles ($\hat{\psi}_i - \bar{\psi}_i$), in degrees, over a 90 degree PPI scan when the radar was in standard mode (i.e. $\psi'_i = 0$).

The histogram of the de-trended measured burst phase angles, $\hat{\psi}_i - \bar{\psi}_i$, is shown in Figure 5-3. The mean of this distribution is approximately 0 and the standard deviation is approximately 0.75 degrees.
Next the statistics of the measured phase difference, $\hat{\phi}$, are examined. These statistics are perhaps more relevant since Doppler velocity is estimated from such phase differences. In Figure 5-4, the time-series of the measured phase difference ($\hat{\phi} = \hat{\phi}_i - \phi'_i$) are shown. Values mostly lie between ±1 degree with regions of larger variance. These areas of larger variance indicate the likely presence of some electronic interference signal.
Figure 5-5: Histogram of the measured phase difference, in degrees, over a 90 degree PPI scan when the radar was in standard mode (i.e. $\psi_i = 0$).

The histogram of the data in Figure 5-4 ($\hat{\phi} = \hat{\phi}_i - \hat{\phi}_f$) is shown in Figure 5-5. The mean of this distribution is approximately 0 and the standard deviation is approximately 0.4 degrees. The reason that this standard deviation is smaller than that for the measured burst phase is that there exist smaller scale trends in the measured burst phase that are being cancelled out in the measured phase difference. Perhaps the most striking aspect of the histogram is the bimodal nature of the distribution which again indicates the presences of some interference signal (i.e., leakage from power supply, clock oscillators, etc.).
The frequency of the interfering signals can be investigated by taking the FFT of the time series of measured burst phases. The magnitude of FFT of the phases is shown in Figure 5-6. Spikes in this plot indicate the presence of interfering or contaminating signals at about 60, 228, 409, and 500 Hz.
5.3 SZ(8/64) Mode - July 20, 2004

The PRT for the July 20, 2004 data is 0.8 ms. In order to compare the measured and theoretical burst phase angles, it is necessary to start the theoretical code at the same index in the sequence as the measured code. However, the requested phase angles ($\psi^r$) are not stored, and thus the easiest way to ensure proper synchronization is to compare modulation codes or, equivalently, the phase differences. Figure 5-7 compares the measured ($\hat{\phi}$) and theoretical ($\phi^r$) phase differences and shows that the two phase sequences are synchronized. The starting index of the two sequences could still be off by a multiple of 8, however, for the SZ(8/64) phase code, this would result only in a constant angle offset in the transmitted phase angles ($\psi - \psi^r = constant$).
Figure 5-8: In blue is the time-series of $\hat{\psi}_i - \psi'_i$, i.e. the difference, in degrees, between the measured and theoretical burst phase angles, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code). The red line shows the same data after a 65 point mean filter, denoted $\hat{\psi}_i - \psi'_i$.

Figure 5-8 shows the circular difference, in blue, between the theoretical and measured burst phase angles ($\hat{\psi}_i - \psi'_i$). This field will be referred to as the measured burst phase error. The red line shows the same data that has been filtered using a 65-point mean filter, which is denoted $\hat{\psi}_i - \psi'_i$. If the transmitted burst phase angle was exactly as requested and the transmitted burst was measured perfectly, then the measured burst phase error would be a flat line at 0. Having only matched the phase difference angles (i.e. their modulation codes match), this field should still be constant. However, there is some drift evident.
Figure 5-9: The time-series of $\dot{\psi}_i - \dot{\psi}'_i$, i.e. the measured burst error, in degrees, over an excerpt of a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code). The green vertical lines demarcate every 64 pulses (1 beam).

Figure 5-9 shows a subset of the unfiltered burst phase errors ($\dot{\psi}_i - \dot{\psi}'_i$) from Figure 5-8. The green lines demarcate different 64-pulse beams. There are places where within 1 beam the phase errors drift by as much as 20 degrees. This drift, however, is substantially better than that which was observed in 2003, which will be shown in the next section.
Figure 5-10: Time-series of \( \hat{\psi}_i - \hat{\psi}_i' \), i.e. the de-trended measured burst phase error, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

The de-trended (65 point circular mean filter) measured burst phase errors, \( \hat{\psi}_i - \hat{\psi}_i' \), are shown in Figure 5-10. It appears there may be minor smaller scale trends that were not removed by the de-trending.
The histogram of \((\hat{\psi}_i - \psi'_i) - (\bar{\psi}_i - \psi'_i)\), i.e. the de-trended measured burst phase errors, is shown in Figure 5-11. The mean of this distribution is approximately 0 degrees and the standard deviation is about 1 degree. This is more than the corresponding value from the data collected in standard mode (page 64) which indicates that the phase shifting mechanism is responsible for the additional phase noise.
Figure 5-12: Time-series of \( \hat{\phi}_i - \phi_i \), i.e. the difference between the measured phase difference angle and the theoretical, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

The difference between the theoretical and measured phase difference angles (\( \hat{\phi}_i - \phi_i \)) are shown in Figure 5-12. These values are largely between ±1.75 degrees.
Figure 5-13: Histogram of $\hat{\phi}_i - \phi_i^T$, i.e. the difference between the measured phase difference angle and the theoretical, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

The histogram of $\hat{\phi}_i - \phi_i^T$ is shown in Figure 5-13, i.e. the difference between the measured and theoretical phase difference angles. The mean and standard deviation of this distribution are approximately 0 and 0.67 degrees, respectively. Note that this standard deviation is less than the standard deviation of the de-trended measured phase errors (Figure 5-11). This implies that there are some short-scale correlations that are not being removed from the de-trending process. Also the standard deviation here of $\hat{\phi}_i - \phi_i^T$ is larger than the corresponding value (0.4 degrees) for the data collected in standard mode (Figure 5-5) and again this indicates the phase shifting network is responsible for additional phase noise of about 0.3 degrees.
Figure 5-14: Spectrum of $\hat{\psi}_i - \psi'_i$, i.e. the measured phase error, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

Figure 5-14 shows the spectrum of the measured burst phase errors and interfering frequency components are evident at about 60, 108 and 408 Hz. SIGMET has suggested that replacing the 36 MHz IFD with their new 72 MHz IFD and replacing the SIGMET receiver card should reduce this phase noise. Another possible source of phase noise is the S-POL transmitter and NCAR is currently investigating this.
Figure 5-15: In red is the time-series of the measured phase difference ($\hat{\phi}$) in degrees, over an excerpt from a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code). The blue line, which is mostly overlaid by the red, shows the theoretical phase difference angles ($\phi^r$).

5.4 SZ(8/64) Mode – July 7, 2003

The PRT for the July 7, 2003 data is 0.960 ms. Because this data was collected using a now-outdated phase shifter card, an in-depth analysis is mostly superfluous. This section is included in order to make a baseline comparison of the phase stability before the new phase shifter card and the use of more compatible PRF’s. Since the plots are included only for reference, only a few comments will be made.
Figure 5-16: In blue is the time-series of $\hat{\psi}_i - \psi'_i$, i.e. the measured burst phase error, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code). The red line shows the same data after a 65 point mean filter, denoted $\bar{\psi}_i - \psi'_i$.

The smoothed time-series in Figure 5-16 does not do an effective job of tracking the trends and thus the analysis of the de-trended measured burst phase errors was not performed.
Figure 5-17: The time-series of $\hat{\psi}_i - \psi_i$, i.e. the measured burst phase error, in degrees, over an excerpt of a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code). The green vertical lines demarcate every 64 pulses (1 beam).

The phase drifted as much as 150 degrees within 1 beam (64 pulses). If the data were not cohered using the measured burst phases a bias of about 0.27 m/s would result for this PRT.
Figure 5-18: Time-series of $\hat{\phi}_i - \phi'_i$, i.e. the difference between the measured phase difference angles and the theoretical, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).
Figure 5-19: Histogram of $\hat{\phi}_i - \phi_f$, i.e. the difference between the measured phase difference angles and the theoretical, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

The mean and standard deviation of this distribution are approximately 0 and 1.6 degrees, respectively.
Figure 5-20: Spectrum of $\hat{\psi}_i - \psi'_i$, i.e. the measured phase error, in degrees, over a 90 degree PPI scan when the radar was in SZ(8/64) mode (i.e. the theoretical transmit phase angles are defined by the SZ(8/64) phase code).

5.5 Conclusions

There is a marked improvement in the phase stability of the current system as compared with that of 2003. In both standard and SZ(8/64) modes, spectral analysis shows that the frequencies of the interference, that causes the measured burst phase angles to be different from the requested phase angles, are very similar.

For standard mode, the pulse-to-pulse phase stability appears to be fairly good with about a 0.4 deg standard deviation. The impact of the phase instability is minimized if the radar data are cohered using the measured burst pulse phase.

For SZ(8/64) mode, the phase stability is somewhat worse. Under the assumption that the phase measurements are accurate, in the event of non-overlaid echoes, cohering using the measured burst phase angles should lead to satisfactory recovery of the moments. In the case of overlaid echoes, the effect of having the burst phase angles different than the requested phase angles is that the out of trip echoes will be modulated by a code that is not exactly SZ(8/64). It has been shown by Sachidananda et al. (1999) that phase noise reduces the recovery region and increases the measurement error of weak trip echoes. Hubbert et al. (2003) showed that phase noise can
lead to anomalous velocity streaks in retrieved weak trip velocities. Thus, for optimum performance of the SZ algorithm, it is important to verify the noise figure of the radar including the phase shift and phase measurement networks. SIGMET has indicated that the phase stability of NCAR’s RVP8/S-Pol should be improved with the installation of the new 72 MHz IFD and receiver card. This is planned for FY2005.
6 Summary and Conclusions

A major accomplishment in FY2004 was the delivery of the SZ-2 algorithm code to the ROC on 15 June. This is the culmination of several years of range-velocity mitigation work by both NCAR, NSSL and the ROC. The SZ-2 algorithm is meant to be used at lower elevation angles where up to four range overlaid signals can occur. The details of the delivered SZ-2 algorithm can be found in the 15 June Interim Report to the ROC. Application of the SZ-2 algorithm to experimental data show that significantly less velocity data will be censored as compared to the currently used NEXRAD range-velocity mitigation algorithm. Even though a practical SZ-2 algorithm has been delivered the algorithm will need further development and “fine tuning” as more experimental data is analyzed.

Another major achievement this year was the delivery of the 30 September Interim Report to the ROC on the comparison of the SIGMET and NSSL/NCAR SZ algorithms. The purpose of the work was to verify the performance of the SIGMET and NSSL/NCAR SZ algorithms. Data gathered by S-Pol alternated between unphase coded long PRT PPIs and phase coded short PRT PPIs. The phase coded data was then played back through RVP8 so that moments data were produced by both SIGMET’s and NSSL/NCAR’s SZ algorithm. These moments were inter-compared and compared to the moments calculated from the long PRT data, which served as truth. This analysis showed that by-in-large the moments produced by the two SZ algorithms were very similar but there were notable differences. The SIGMET SZ algorithm produced anomalous velocity estimates that caused peculiar streaks in the scatter plots of estimated velocities. Much of the anomalous velocities were eliminated when regions with clutter were avoided. The cause of the velocity anomalies was attributed to the SIGMET SZ code. Also, no bias was found in the SIGMET weak trip velocity estimates. There was a 2 m/s bias found in simulations conducted by NSSL as reported in their annual report to the ROC, Part 6 (2002). The conclusion is however, that even though the SIGMET SZ algorithm performed better than expected, the NSSL/NCAR algorithm is still preferable since it can more easily be upgraded and integrated into the SZ-1 and SZ-2 algorithms which are considerably more complicated than just the SZ algorithm due to censoring, clutter filtering, etc. NCAR’s full Interim Report can be viewed at [http://www.atd.ucar.edu/rsf/NEXRAD/index.html](http://www.atd.ucar.edu/rsf/NEXRAD/index.html).

NCAR also investigated the performance of the GMAP clutter filtering algorithm using experimental data. Two sets of time series were sorted: 1) times series containing only weather echoes and 2) time series with only ground clutter echoes. Thus, times series could be combined so that the weather power and clutter power was known and the GMAP clutter filter performance could be effectively evaluated. NCAR also developed and tested a similar GMAP type spectral domain clutter filter called the Spectral Stationary Echo Filter (SSEF). The analysis confirmed that such spectral domain clutter filters that operate on the magnitude of spectra are compatible with SZ phase coding and pose no problem. C-code for the SSEF algorithm can be found in the Appendix.

On certain spectra, GMAP sometimes produces incorrect results. For example sometimes the ground filter notch is too shallow and sometimes even a reconstructed notch can increase the clutter peak. The reason for this behavior is complicated and depends on the shape of the individual spectra. Fortunately these cases are rare and use of a GMAP type clutter filter is still recommended. The performance of SSEF relative to GMAP suggests that GMAP is not optimal
in its present form and that perhaps GMAP could be tuned or altered slightly to improve its performance.

A fuzzy logic based censoring method was developed for use with the SZ-1 algorithm. The method combines information about the spatial texture of radial velocity estimates with spectral characteristics to detect and censor contaminated data. It was found that the algorithm performs well and produces results comparable to the SZ-2 censoring algorithm. The spectral ratio adds skill to the SZ-1 censoring algorithm and narrows the censored region because it is computed at each gate, and not over a small region. Computing the texture and spin feature fields both in the range and azimuth directions separately and only censoring data with high values in both directions, effectively censored contaminated data without censoring a valid tornado signature. The tornado signature is an example of a weather echo that has high velocity texture and spin in one direction (azimuth), with relatively low values in the other (range). Contaminated velocity estimates are noisy in both the azimuth and range directions, thus enabling the separation of contaminated and valid echo. Further tests should be conducted on a wide range of weather echoes, which may have high texture values, including strong velocity gradients, gust fronts/convergence lines, and strong divergence signatures such as microbursts. Many experiments have been performed to determine optimal membership functions, weights, fuzzy logic rules, and interest field combinations, however further optimization of the SZ-1 algorithm is required. One noticeable need for improvement is the censoring of contamination that has a spatial extent of only a few range gates. There are many options including existing speckle filters that can be tested.

In order to reduce the phase noise of the RVP8/S-Pol system as reported in the FY2003 Annual Report to the ROC, two steps were taken: 1) SIGMET’s new transmit card was installed and 2) long and short PRTs were chosen that were divisible by the 36MHz IFD sampling rate and the 10MHz GPS synchronization clock. Analysis of the new improved FY2004 data showed that the phase stability has improved greatly and that selection of system compatible PRTs is very important. The standard deviation of phase measurements was about 0.4 degrees when no phase coding was applied to the transmit pulses and about 0.7 degrees when phase coded pulses were used. From NEXRAD specifications for phase noise of the ASR-9 transmitter, this phase noise should be better than -54 dB. This corresponds to better than 0.1 degree of phase noise. Communications with SIGMET indicate that upgrading RVP8 with their new 72 MHz IFD and their new receiver card should improve the RVP8/S-Pol noise figure. NCAR is also investigating improvements to S-Pol’s transmitter.

Acknowledgements: We would like to thank the ROC staff, especially Darcy Saxion and Rick Rhoton, Sebastion Torres, Dusan Zrnic, Dale Sirmans and Jeff Keeler for the many technical discussions and assistance that helped make this report possible.
7 References


8 Appendix.

Appendix A: C++ code for SSEF

This appendix includes the C++ code for the ClutFilter class, which was used to develop and test SSEF.

```cpp
François Hotte, RAP, NCAR
P.O.Box 3000, Boulder, CO, 80307-3000, USA

// This class
class ClutFilter {

public:

    // constructor
    ClutFilter();

    // destructor
    ~ClutFilter();

    // perform filtering
    void run(const double *rawMag,
              int nSamples,
              double max_clutter_vel,
              double init_notch_width,
              double nyquist,
              double *filteredMag,
              int &clutterFound,
              int &notchStart,
              int &notchEnd,
              double &powerRemoved,
              double &vel,
              double &width);

protected:

private:

    void _locateWxAndClutter(const double *power,
                              int nSamples,
                              double max_clutter_vel,
                              double initNotch_width,
                              double nyquist,
                              int &notchWidth,
                              int &clutterFound,
                              int &weatherPos,
                              int &clutterPos);
```
void _fitGaussian(const double *magnitude,
    int nSamples,
    int weatherPos,
    double minMagnitude,
    double nyquist,
    double &vel,
    double &width,
    double *gaussian);

};
#endif

////////////////////////////////////////////////////////////////////////
// ClutFilter.cc
//
// Mike Dixon, RAP, NCAR, P.O.Box 3000, Boulder, CO, 80307-3000, USA
//
// March 2004
//
////////////////////////////////////////////////////////////////////////
// Perform clutter filtering
//
////////////////////////////////////////////////////////////////////////
#include <iostream>
#include <iomanip>
#include "ClutFilter.hh"

#ifndef MIN
#define MIN(a, b) ((a) < (b) ? (a) : (b))
#endif

#ifndef MAX
#define MAX(a, b) ((a) > (b) ? (a) : (b))
#endif

using namespace std;

// Constructor
ClutFilter::ClutFilter()
{
}

// destructor
ClutFilter::~ClutFilter()
{
}

////////////////////////////////////////////////////////////////////////
// perform filtering
//
// Returns power removed
void ClutFilter::run(const double *rawMag,
    int nSamples,
    double max_clutter_vel,
    double init_notch_width,
double nyquist,
double *filteredMag,
int &clutterFound,
int &notchStart,
int &notchEnd,
double &powerRemoved,
double &vel,
double &width)
{

  // initialize
  clutterFound = 0;
  notchStart = 0;
  notchEnd = 0;
  powerRemoved = 0.0;

  // compute power from magnitudes
  double rawPower[nSamples];
  const double *rm = rawMag;
  double *rp = rawPower;
  for (int ii = 0; ii < nSamples; ii++, rm++, rp++) {
    *rp = *rm * *rm;
  }

  // compute min power - for use as noise value
  double minPower = rawPower[0];
  for (int ii = 1; ii < nSamples; ii++) {
    if (minPower > rawPower[ii]) {
      minPower = rawPower[ii];
    }
  }

  // locate the weather and clutter
  int weatherPos, clutterPos;
  int notchWidth;
  _locateWxAndClutter(rawPower,
    nSamples,
    max_clutter_vel,
    init_notch_width,
    nyquist,
    notchWidth,
    clutterFound,
    weatherPos,
    clutterPos);

  // notch out the clutter, using the initial notch width
  double notched[nSamples];
  memcpy(notched, rawPower, nSamples * sizeof(double));
  for (int ii = clutterPos - notchWidth; ii <= clutterPos + notchWidth; ii++) {
    notched[(ii + nSamples) % nSamples] = minPower;
  }

  // widen the notch by one point on either side,
  // copying in the value adjacent to the notch
  notched[(clutterPos - notchWidth - 1 + nSamples) % nSamples] =
  notched[(clutterPos - notchWidth - 2 + nSamples) % nSamples];
  notched[(clutterPos - notchWidth + 1 + nSamples) % nSamples] =
  notched[(clutterPos - notchWidth + 2 + nSamples) % nSamples];
double gaussian[nSamples];
int maxSearchWidth = notchWidth * 2;
if (maxSearchWidth > nSamples / 4) {
    maxSearchWidth = nSamples / 4;
}
double matchRatio = 10.0;
double prevPower = 0.0;
int clutterLowerBound;
int clutterUpperBound;

// iterate 3 times, refining the correcting further each time
for (int iter = 0; iter < 3; iter++) {

    // fit gaussian to notched spectrum
    _fitGaussian(notched, nSamples, weatherPos, minPower, nyquist,
         vel, width, gaussian);

    // find where clutter peak drops below gaussian, and
    // create a notched spectrum using the gaussian where
    // the clutter peak was
    clutterLowerBound = clutterPos - maxSearchWidth;
    clutterUpperBound = clutterPos + maxSearchWidth;

    prevPower = rawPower[clutterPos];
    for (int ii = clutterPos - 1; ii >= clutterPos - maxSearchWidth; ii--) {
        int jj = (ii + nSamples) % nSamples;
        double power = rawPower[jj];
        double gauss = gaussian[jj];
        if (power <= gauss) {
            // power falls below gaussian fit
            clutterLowerBound = ii + 1;
            break;
        }
        if (power < gauss * matchRatio) {
            if (power > prevPower) {
                // power came close to gaussian fit and is moving away
                clutterLowerBound = ii + 1;
                break;
            }
        }
        prevPower = power;
    }

    prevPower = rawPower[clutterPos];
    for (int ii = clutterPos + 1; ii <= clutterPos + maxSearchWidth; ii++) {
        int jj = (ii + nSamples) % nSamples;
        double power = rawPower[jj];
        double gauss = gaussian[jj];
        if (power < gauss) {
            // power falls below gaussian fit
            clutterUpperBound = ii - 1;
            break;
        }
        if (power < gauss * matchRatio) {
            if (power > prevPower) {
                // power came close to gaussian fit and is moving away
                clutterUpperBound = ii - 1;
                break;
            }
        }
        prevPower = power;
    }

    // recompute notched spectrum, using gaussian to fill in notch
memcpy(notched, rawPower, nSamples * sizeof(double));
for (int ii = clutterLowerBound; ii <= clutterUpperBound; ii++) {
    int jj = (ii + nSamples) % nSamples;
    notched[jj] = gaussian[jj];
}
}

// compute the power associated with the peak at each point
powerRemoved = 0.0;
for (int ii = clutterLowerBound; ii <= clutterUpperBound; ii++) {
    int jj = (ii + nSamples) % nSamples;
    double diff = rawPower[jj] - notched[jj];
    powerRemoved += diff;
}
powerRemoved /= nSamples;

notchStart = (clutterLowerBound + nSamples) % nSamples;
notchEnd = (clutterUpperBound + nSamples) % nSamples;

// set filtered mag array
double *fm = filteredMag;
double *no = notched;
for (int ii = 0; ii < nSamples; ii++, fm++, no++) {
    *fm = sqrt(*no);
}
return;

/---------------------------------
// find weather and clutter
// Divide spectrum into 8 parts, compute peaks and means
// for each part. Check for bi-modal spectrum.

void ClutFilter::_locateWxAndClutter(const double *power,
int nSamples,
double max_clutter_vel,
double init_notch_width,
double nyquist,
int &notchWidth,
int &clutterFound,
int &weatherPos,
int &clutterPos)
{
    // initialize
    clutterFound = 0;
    weatherPos = 0;
    clutterPos = 0;

    int nHalf = nSamples / 2;
    int nClutVel =
        (int) ((max_clutter_vel / (nyquist * 2.0)) * nSamples + 0.5);
    nClutVel = MAX(nClutVel, nHalf - 1);
    nClutVel = MIN(nClutVel, 1);

    notchWidth =
        (int) ((init_notch_width / (nyquist * 2.0)) * nSamples + 0.5);
    notchWidth = MIN(notchWidth, nHalf - 1);
    notchWidth = MAX(notchWidth, 1);
// divide spectrum into 8 parts, compute power in each part
int nEighth = ((nSamples - 1) / 8) + 1;
if (nEighth < 3) {
    nEighth = 3;
}
int nSixteenth = nEighth / 2;
double blockMeans[8];
for (int ii = 0; ii < 8; ii++) {
    int jjStart = ((ii * nSamples) / 8) - nSixteenth;
    // cerr << "ii, jjStart: " << ii << ", " << jjStart << endl;
    blockMeans[ii] = 0.0;
    for (int jj = jjStart; jj < jjStart + nEighth; jj++) {
        int kk = (jj + nSamples) % nSamples;
        blockMeans[ii] += power[kk] / 8;
    }
}

// compare peak at 0 with max of other peaks
// if less than 3dB down, we have clutter
double zeroMean = blockMeans[0];
double maxOtherMean = 0.0;
for (int ii = 1; ii < 8; ii++) {
    maxOtherMean = MAX(maxOtherMean, blockMeans[ii]);
}
clutterFound = 0;
if ((zeroMean / maxOtherMean) > 0.5) {
    clutterFound = 1;
}
if (!clutterFound) {
    return;
}

// find clutter peak within velocity limits
clutterPos = 0;
double clutterPeak = 0.0;
for (int ii = -nClutVel; ii <= nClutVel; ii++) {
    double val = power[(ii + nSamples) % nSamples];
    if (val > clutterPeak) {
        clutterPeak = val;
        clutterPos = ii;
    }
}

// check for bimodal spectrum, assuming one peak at DC
// find pos of peak away from DC
double weatherMean = 0.0;
int wxMeanPos = 0;
for (int ii = 2; ii < 7; ii++) {
    if (blockMeans[ii] > weatherMean) {
        weatherMean = blockMeans[ii];
        wxMeanPos = ii;
    }
}

// check for 3dB valleys between DC and peak
// if valleys exist on both sides, then we have a bimodal spectrum
int biModal = 1;
int vallyFound = 0;
for (int ii = 1; ii < wxMeanPos; ii++) {
    if (weatherMean / blockMeans[ii] > 5.0) {
vallyFound = 1;
break;
}
}
if (!vallyFound) {
biModal = 0;
}
}
}

vallyFound = 0;
for (int ii = wxMeanPos; ii < 8; ii++) {
if (weatherMean / blockMeans[ii] > 5.0) {
vallyFound = 1;
break;
}
}
if (!vallyFound) {
biModal = 0;
}

// if bimodal, find weather peak away from clutter peak
// else find weather peak outside the notch

int iStart = 0, iEnd = 0;
if (biModal) {
iStart = ((wxMeanPos * nSamples) / 8) - nSixteenth;
iEnd = iStart + nEighth;
} else {
iStart = clutterPos + 2 * notchWidth + 1;
iEnd = iStart + nSamples - (4 * notchWidth) - 1;
}

double weatherPeak = 0.0;
for (int ii = iStart; ii < iEnd; ii++) {
int kk = (ii + nSamples) % nSamples;
if (weatherPeak < power[kk]) {
weatherPeak = power[kk];
weatherPos = kk;
}
}

// fit gaussian to spectrum

void ClutFilter::_fitGaussian(const double *power,
int nSamples,
int weatherPos,
double minPower,
double nyquist,
double &vel,
double &width,
double *gaussian)
{

// center power array on the max value

double centered[nSamples];
int kCent = nSamples / 2;
int kOffset = kCent - weatherPos;
for (int ii = 0; ii < nSamples; ii++) {
int jj = (nSamples + ii + kOffset) % nSamples;
centered[jj] = power[ii];
}

// compute mean and sdev
double sumPower = 0.0;
double sumPhase = 0.0;
double sumPhase2 = 0.0;
double *ce = centered;
for (int ii = 0; ii < nSamples; ii++, ce++) {
    double phase = (double) ii;
    double power = *ce;
    sumPower += power;
    sumPhase += power * phase;
    sumPhase2 += power * phase * phase;
}
double meanK = 0.0;
double sdevK = 0.0;
double varK = 0.0;
if (sumPower > 0.0) {
    meanK = sumPhase / sumPower;
    varK = (sumPhase2 / sumPower) - (meanK * meanK);
    if (varK > 0) {
        sdevK = sqrt(varK);
    } else {
        varK = 0.0001;
    }
}

// compute curve

double c1 = sumPower / (sqrt(2.0 * M_PI) * sdevK);
double c2 = -1.0 / (2.0 * varK);
int istart = (int) (meanK - nSamples / 2 + 0.5);
for (int ii = istart; ii < istart + nSamples; ii++) {
    double xx = ((double) ii - meanK);
    double fit = c1 * exp((xx * xx) * c2);
    if (fit < minPower) {
        fit = minPower;
    }
    int jj = (nSamples + ii - kOffset) % nSamples;
    gaussian[jj] = fit;
}

double velFac = (nyquist * 2.0) / nSamples;
vel = velFac * (meanK - (kOffset + nSamples) % nSamples);
if (vel < -nyquist) {
    vel += 2.0 * nyquist;
} else if (vel > nyquist) {
    vel -= 2.0 * nyquist;
}
width = (velFac * sdevK);

Appendix B: Range Velocity Ambiguity Mitigation Publications