FITACF description and revision

Pasha Ponomarenko and Colin Waters

1 Introduction

This work was motivated by two factors:

- Since its inception in the late 1980s the FITACF code remains pretty much a "black box" for the majority of SuperDARN data users.
- During the last decade extensive but unsuccessful attempts were made to explain large spectral width $W \geq 200$ m/s values regularly observed by SuperDARN radars at high latitudes.

At SuperDARN'04 in Saskatoon the authors initiated a discussion concerning the above topics by presenting a poster "Possible causes of large spectral width in SuperDARN echoes from high latitudes". This effort resulted in the creation of the "FITACF workgroup" charged with documenting FITACF algorithms.

We started by plotting a flow-chart, which allowed a general understanding of the tasks performed by different procedures and the connections between them. Then we focused on the procedures which are involved in estimating W. Obviously, during this process we learned about details of the FITACF algorithms and their implementation. As a result, we located several factors that increase spectral width estimates and studied their relative contributions. Several modifications to FITACF have been proposed, which considerably improved the physical validity and accuracy of FITACF output parameters.

2 Flow-chart

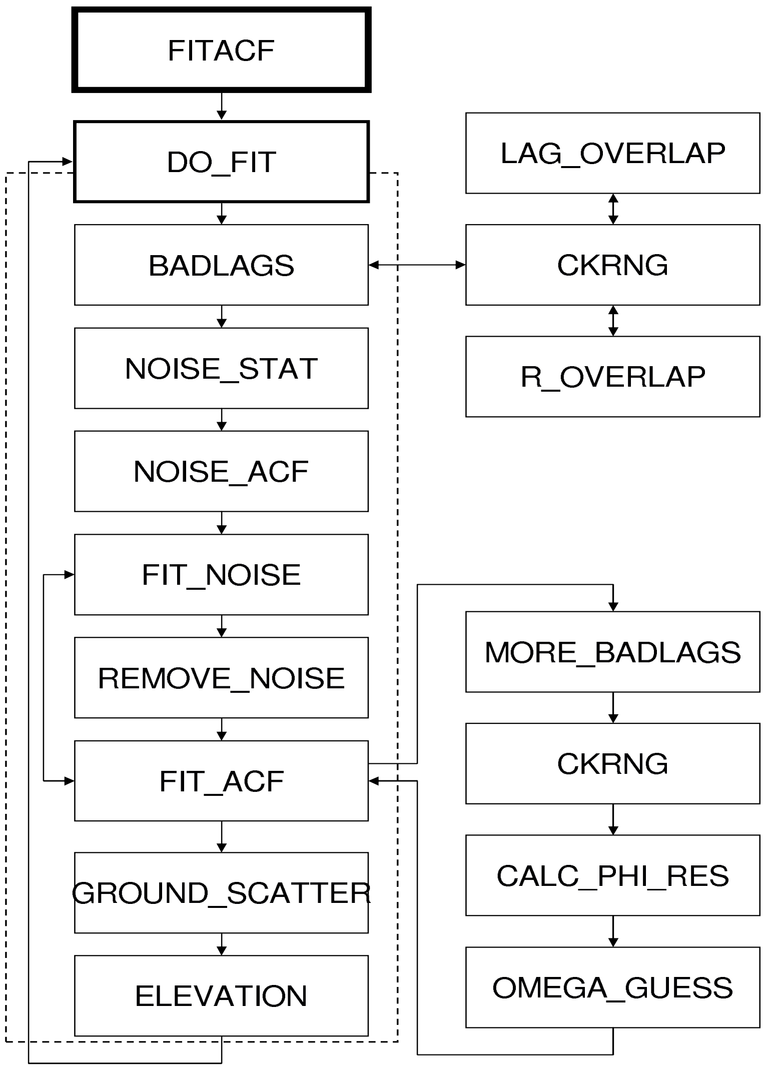
This flow-chart was constructed based on our analysis of the FITACF package (version 1.09) written in C by Rob Barnes and provided to us by Simon Shepherd.

The data are fed to FITACF in a single block read from a the DAT file. This block consists of complex ACFs measured for all range gates at a certain time in a certain beam.

The FITACF package

- creates corresponding data structures and reads data into them,
- marks ACF lags contaminated by cross-range and pulse-overlap interference
- determines reference noise parameters
- checks for the presence of coherent interference (broadcasting stations etc) and, if present, attempts to remove this component from ACFs
- marks "bad lags" based on the ACF power shape assumptions (non-increasing power)
- fits model curves to ACF power and phase to estimate output parameters and respective errors

- marks ACFs with ground/sea scatter
- calculates elevation from interferometer data (XCF)
- writes output parameters into a FIT file.



3 Confirmed and potential problems

We have singled out several factors, which either lead or may lead to incorrect or erroneous estimates of the scatter signal parameters. These include:

3.1 Removing the fluctuation level from ACF power before fitting

Location:
  FIT_ACF.C
Line number:
  229 w[k] = w[k] - P0n;

In the original FITACF, the fluctuation level P0n=$R(0)/\sqrt{N_A}$ (N_A is number of averages) is treated as a positive offset. In reality P0n is a magnitude of variation of the measured ACF power around its expected value which may both increase and decrease the measured $|R(\tau)|$. Subtracting P0n from $|R(\tau)|$ causes an overestimate of the spectral width by $\simeq 20-40\%$ for typical integration times of 3-7 s. This problem was discussed in detail in our *AG* manuscript "Spectral width of SuperDARN echoes: Measurement, use and physical interpretation" provided to the workgroup earlier. In Figure 1 we show how an experimental ACF was processed with the original FITACF (left panels) and with the same package but without subtracting the fluctuation level (right panels).

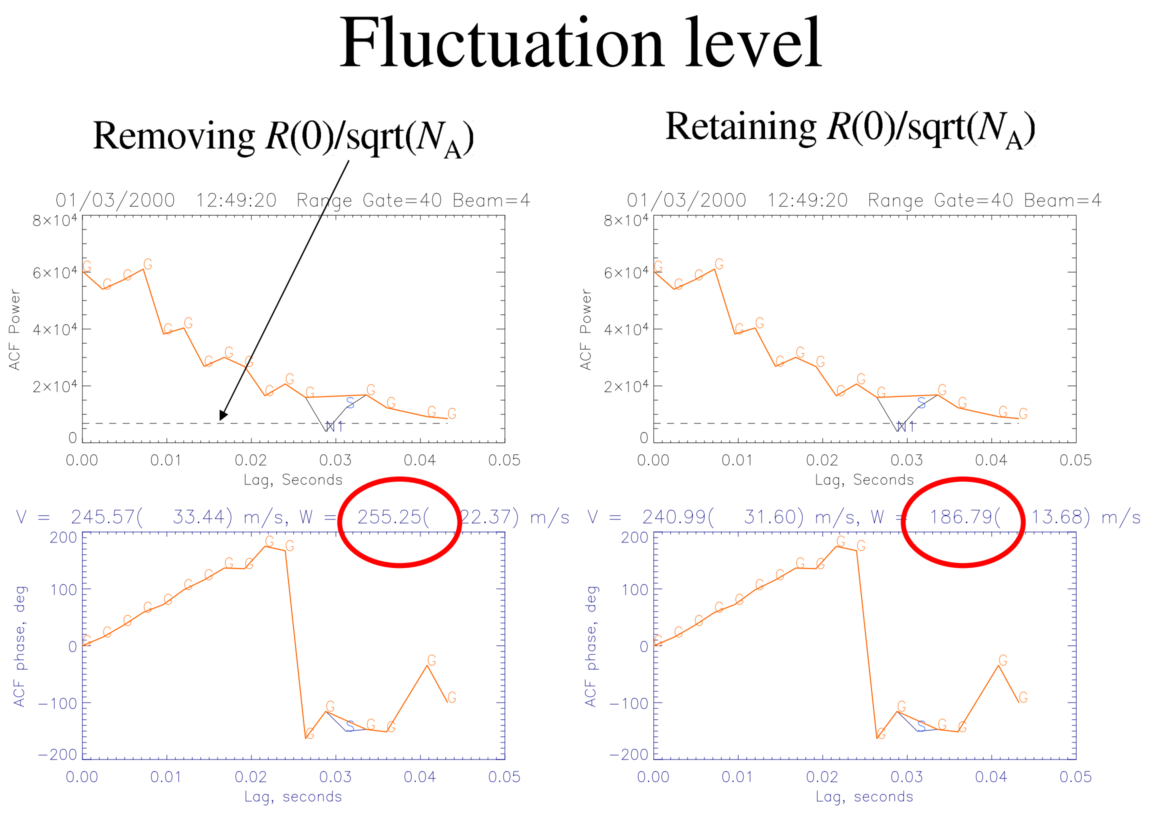


Figure 1.

Top panels show ACF power and bottom ones show ACF phase. Red lines and symbols correspond to “good” lags, black or blue solid lines and symbols correspond to “bad” lags of all sorts. Horizontal dashed line shows the fluctuation level $R(0)/\sqrt{N_A}$. Fitting results for velocity, V, and spectral width, W, are shown above the phase plots with their respective error in brackets. For this particular ACF the increase in spectral width is $\simeq 36\%$.

The same is true with regards to the “non-zero lag noise” (noise_lev = noise_pwr from NOISE_STAT.C), which is also extracted from ACF power before fitting:
Location:
FIT_ACF.C
Line number:
188 w[k] = cabs(acf[k]) - noise_lev;

Essentially, this is just a fluctuation level for a δ-correlated noise, which cannot be considered as just a positive offset.

3.2 Retaining lag 0 noise in *R(0)*

The lag 0 noise, $R_n(0)$, is a noise power at lag 0 estimated from the 10 weakest ACFs in the beam. Due to the fact that any coherent noise has been removed by the REMOVE_NOISE.C routine, $R_n(0)$ represents a δ-function noise at lag 0. Its presence causes overestimation of W, which was discussed in detail in our AG manuscript. In Figure 2 we illustrate how subtraction of $R_n(0)$ (thick black line at $\tau = 0$) affects the measured value of spectral width for ground/sea scatter ACF. In this example the presence of $\simeq$ 40% lag 0 noise causes $\simeq$ 250% increase in W.

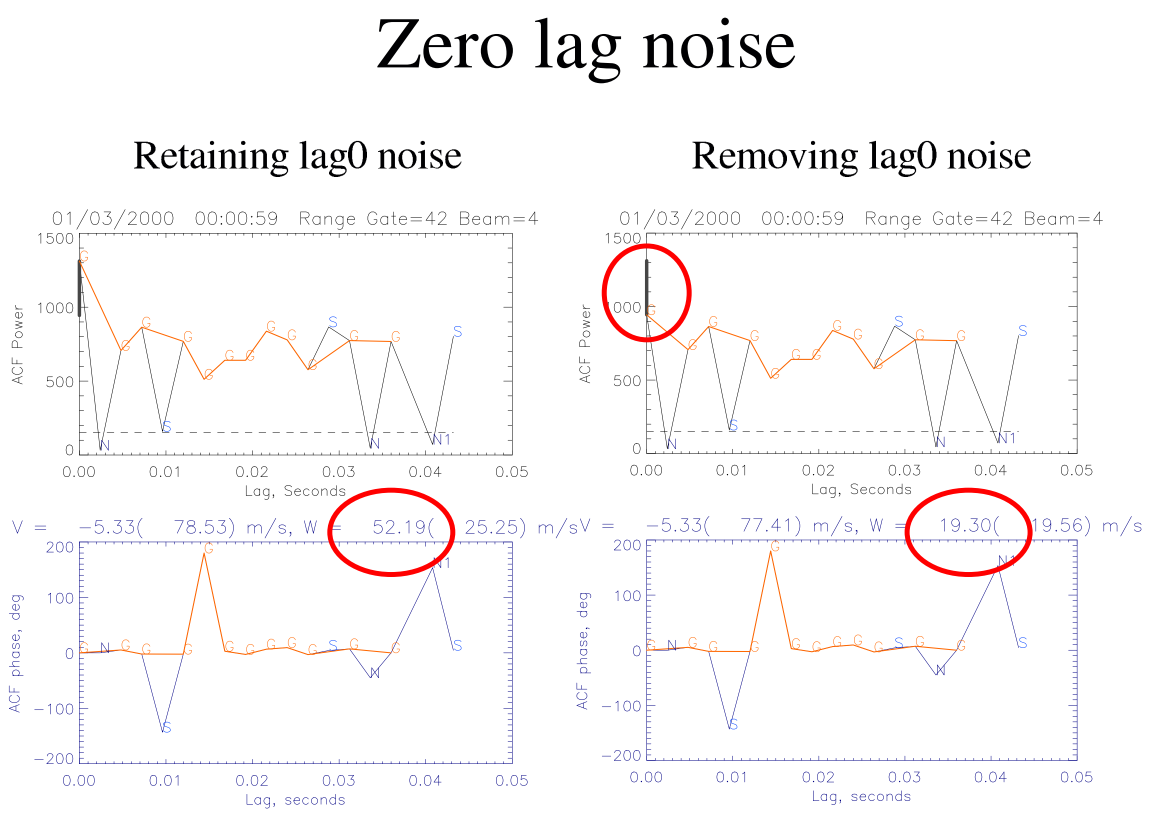


Figure 2.

3.3 Cross-range interference threshold

The very "liberal" cross-range interference (CRI) threshold implemented in FITACF allows for highly contaminated lags to be used in the power/phase fits, which affects estimates of W and its errors. These effects are compensated partially with MORE_BADLAGS routine, which applies empirical criteria to reject lags, whose power level does not correspond to a single-component ACF model. The CRI problem was also discussed in detail in our manuscript, where we proposed to change a CRI threshold from $0.3N_AR(0)$ to just $R(0)$. In Figure 3 we show a low-W ionospheric scatter ACF, which was analysed by FITACF with old (left panels) and new (right panels) threshold values. The original algorithm leaves a few definitely "bad" lags intact and marks some "good" ones as "bad" (symbol "S" shows "bad" lags identified by MORE_BADLAGS), while the new threshold provides adequate identification of the "bad" lags due to CRI (symbol "R" corresponds to lags affected by CRI).

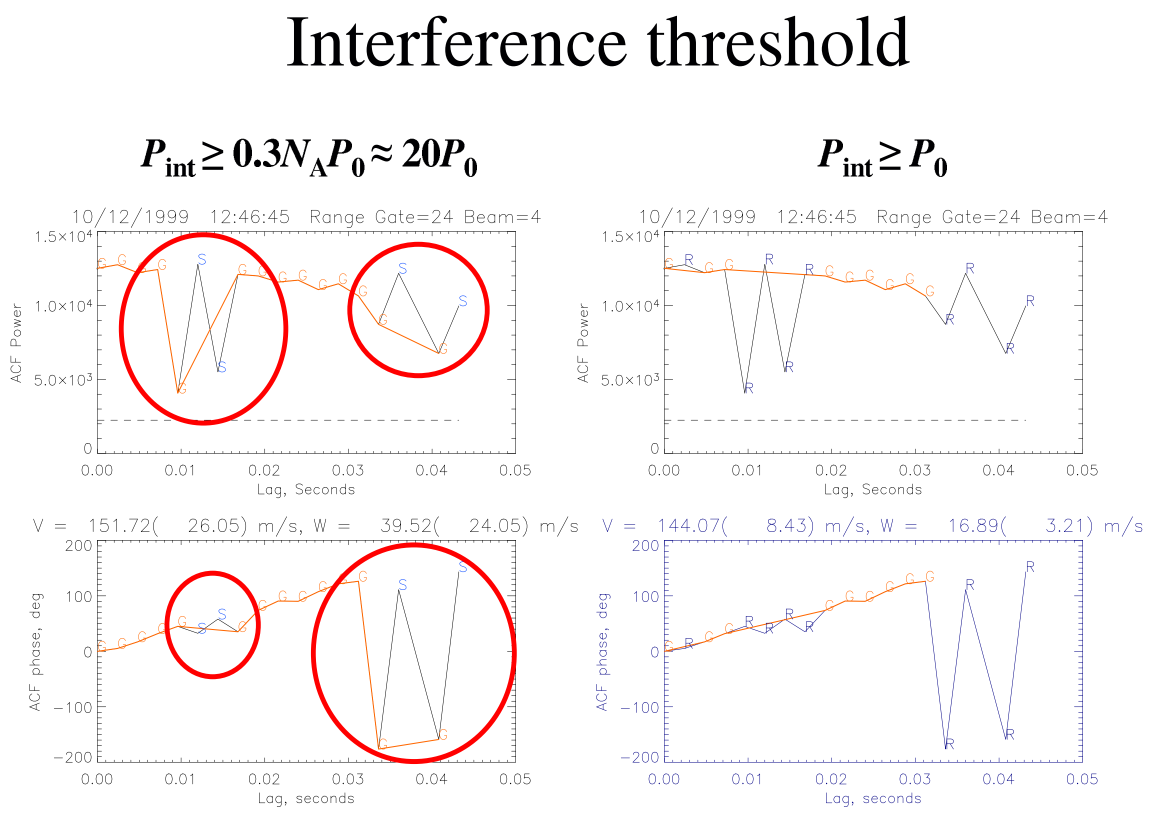


Figure 3.

3.4 Velocity error overestimate

The velocity (phase slope) estimate error is calculated as a geometrical mean from the fitting error, $\epsilon_{1\tau}$ (sdev_phi in FIT_ACF.C) and 1-tau slope estimate variation, ϵ_{fit} (omega_err_loc):
Location:

```
FIT_ACF.C
```

Line number:

```
366 /* estimate the standard deviation of the ACF phase fit to be the
367 geometric mean of the deviation of our best fit line, and
368 the variation in the 1-tau lags.
...
376 if (!xflag) ptr->sdev_phi =
377 sqrt(ptr->sdev_phi*omega_err_loc*t0);
```

The 1-tau slope variation is estimated by the OMEGA_GUESS routine via at least three valid pairs of ACF lags separated by an elementary lag (tau). If there is less than three of these pairs, the program uses an initial value for omega_err_loc=9999, which causes a gross overestimate of the geometrical mean error.

Location:

```
OMEGA_GUESS.C
74 *omega_err = 9999.; (Initial value)
...
107 } else if (nave >=3) {
...
110 *omega_err = sigma/(mpinc*1.0e-6);
111 return omega;
112 }
113 else ++tau_lim;
...
```

For illustration of this effect, in Figure 4 we present an experimental ACF which has only one valid 1-tau pair. On the left side we present fitting results for the original FITACF giving $V_{err} = 72.75$ m/s, and on the right the same but using the fitting error only producing $V_{err} = 2.29$ m/s. In this particular case due to the large velocity error a low-W ionospheric scatter ACF was mis-identified as ground/sea scatter. This is because the ground scatter flag is set according to the difference between velocity/width magnitude and velocity/width error (GROUND_SCATTER.C):

$$|W| - W_{err} < 35\text{m/s}$$

and

$$|V| - V_{err} < 30\text{m/s}.$$

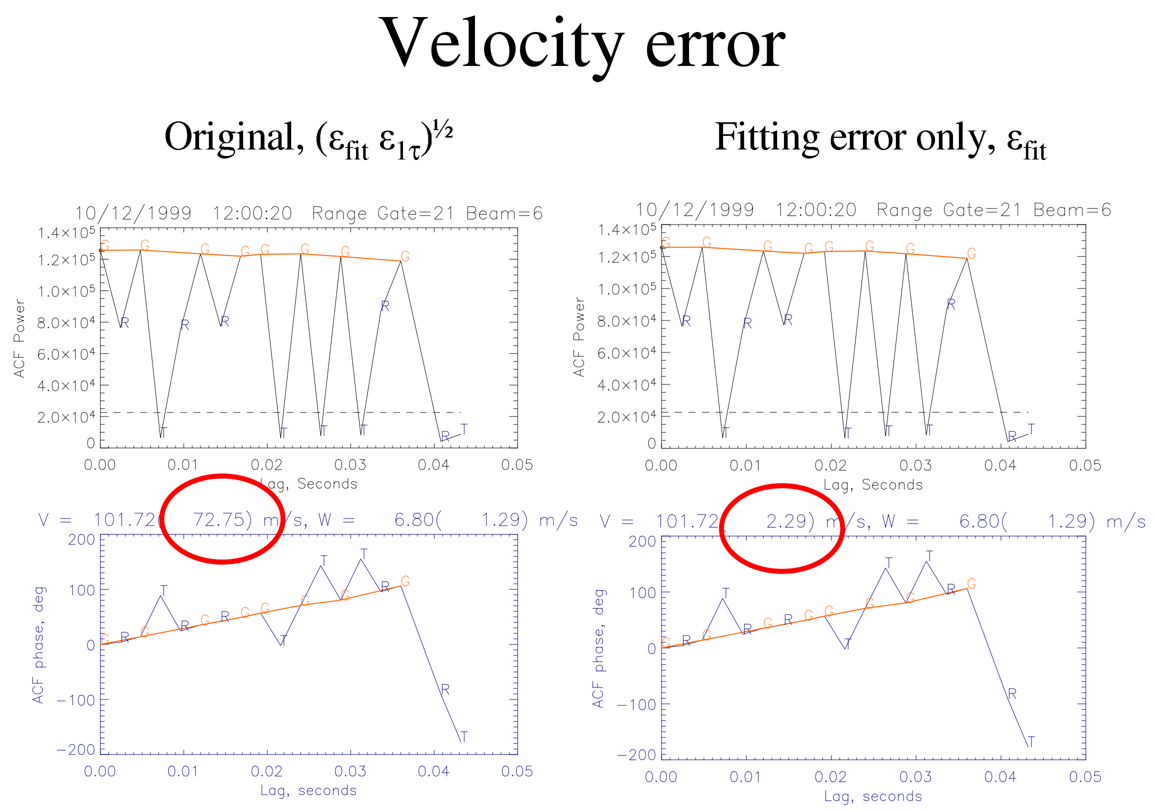


Figure 4.

3.5 Blanketing pulse length

Pulse-overlap blanketing pulse length might be longer than anticipated. For example, for TIGER it covers three range gates instead of two. Data from each SuperDARN radar should be carefully examined to ensure that transmit pulses are adequately blanked.

3.6 ACF shape routine (MORE_BADLAGS.C)

The MORE_BADLAG routine assigns bad lag labels 5, 7 and 9 according to the assumption of non-increasing ACF power (single-frequency signal). It uses purely empirical criteria (positive or negative "spikes" etc), which may not have a clear physical justification. For example, it may allow non-zero lag power to be much larger than $R(0)$. In our manuscript we have shown that the majority of "bad" lags identified by MORE_BADLAGS.C are actually caused by CRI.

3.7 Bad lags for phase fit

Low-level data below the statistical fluctuation level, while being extracted from the power fit, are still used in the phase fit. This data are heavily weighted down during the fitting process to estimate velocity, but they are still used in calculating the 2π phase flips.

3.8 Arbitrary coefficients

Arbitrary/empirical coefficients (1.6, $\sqrt{3}$, $0.3 * N_{avg}$, $OMEGA * \pi/64$) are used in FITACF without clear physical justification.

Examples (arbitrary coefficients are shown in **bold**):

1. NOISE_STAT.C

   ...

   66 #define **PLIM (1.6)** – *power limit for defining non-zero lag noise*

   60 #define **ROOT_3 1.7** – *non-zero lag power level to define if there is coherent interference*

   ...

   92 if ((acf[i][0].x > **plim**) || (acf[i][0].x <= 0.0)) continue;

   ...

   120 if ((P >= sigma * **ROOT_3**) && (sigma > 0.0)) *signal = P;

   ...

2. DO_FIT.C

   ...

   187 if (ave_noise_pwr > noise_pwr/**2.0**) { – *coherent-to-incoherent noise ratio threshold for subtracting "noise ACF"*

   188 /* run fit_acf on the noise acf and xcf */

   ...

3. RANG_BADLAGS.C

   ...

   63 #define **MIN_PWR_RATIO .3** – *cross-range interference threshold*

   ...

   118 pwr_ratio = (long) (nave * **MIN_PWR_RATIO**);

   ...

4. FIT_ACF.C

   ...

   302 while (fabs(omega_old - omega_loc) > fabs(omega_loc * **PI/64.**)) { – *accuracy of the phase fit*

   ...

3.9 Other potential problems

1. Minimum allowed number of good lags is actually 1 (!!!).
   Location:
   FIT_ACF.C
   Line number:

   390 /* POWER FITS: We now turn to the power fits. The sums have to be

```
partially redone, since we have subtracted P0n. */

/* We are now faced with the question of what to do if we don't have enough
lags left to do a fit. we can't abandon the data because the phase fit is
actually ok. we have to have at least 3 points to do the fit and estimate
an error on the fit.

If we don't have at least 3 good points, then simply set the lambda and
sigma powers both to the power_lag0 level. If there are only 2 good points
then calculate the value of sigma and lambda, but set the error estimate
to HUGE_VAL.

If we end up with only lag-0 being good, then flag the width estimate
by setting it to a large negative value.

*/
```

2. Minimum allowed signal-to-noise ratio (parameter p_0). Current value is 1 (0 dB).
3. Using 10 weakest ACF to calculate lag 0 noise power. Why it is not the “clear sky” noise?

4 Recommendations

1. Leave ACF power fluctuation level intact and during the fitting procedure just ignore those lags with power below the fluctuation level.
2. Subtract the lag zero noise from $R(0)$ before power fitting, but use the total (signal + noise) lag 0 power when calculating the fluctuation level.
3. Change cross-range interference threshold from $0.3N_A$ to 1.
4. Abandon "bad" lags labelled by MORE_BADLAGS.C as 5,7, and 9.
5. In calculating velocity estimate error use fitting error only.
6. Use the same set of "bad" lags for both power and phase. This relates to lags with power below the fluctuation level.

5 Acknowledgments

We are grateful to Kile Baker for producing FITACF tutorial available from
http://superdarn.jhuapl.edu/tutorial/fitacf.pdf
and to Simon Shepherd for providing access to original FITACF codes.