# Implementation of Median Smoothing in Radio-SkyPipe II

Dave Typinski

When attempting to remove relatively short duration noise spikes from data, the median is a better indicator of trend than the mean. I present here a method to implement median smoothing in Radio-SkyPipe II.

## Noise Reduction

Almost everyone is plagued by radio frequency interference to one degree or another. Who among us has *never* looked at a strip chart and wished there were less noise in the data? Short duration noise spikes are a common type of interference. The task at hand is to remove these from our data. One way to do it is to assume that all short duration spikes in the data are spurious, that they are statistical outliers and do not represent the actual signal from a cosmic radio source. Yes, that does involve making an assumption that may or may not turn out to be true—but, all data reduction involves making assumptions of one sort or another. That is the trade off: you get smoother data at the expense of removing statistical outliers that may actually represent a real signal. No free lunches.

## Mean versus Median

A RadioSky-Pipe (RSP) strip chart's data is simply a series of numbers and their associated time stamps. If we want to smooth the strip chart data, we have two options: using the mean value of a subset of the data, or using the median value of the same subset.

For example, suppose we have the following set of values: 1, 2, 12, 3, 1.

The mean value is commonly called the average value. That is: (1+2+12+3+1)/5 = 3.8.

To find the median value, we must first sort the series: 1, 1, 2, 3, 12. We then find the middle value in the list of sorted values; that is, element number three in our set. In our example, the median value is 2.

If these five numbers represented RSP strip chart data, we can see in Figure 1 that the median value of 2 is a much better indicator of overall trend than the mean value of 3.8. This happens because of that one data point with a value of 12: it pulls the mean value too high (according to our assumption that statistical outliers are noise, not signal), but this single data point does not alter the median value. An outlier of 12 isn't that bad, but we often see outliers several orders of magnitude greater than the bulk of the data. It is with these extreme outliers where median smoothing really outshines plain-Jane averaging.

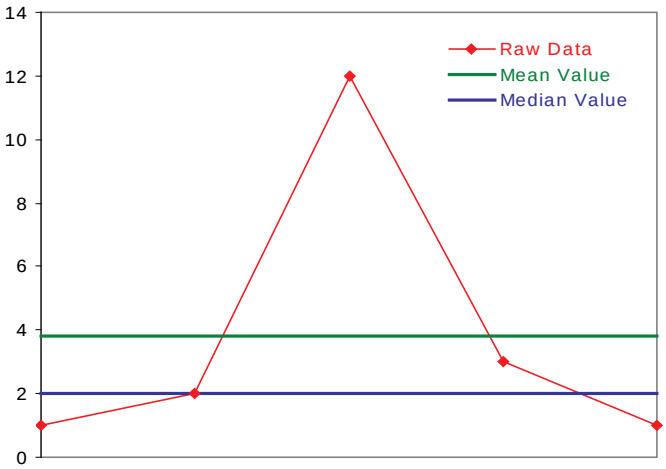


**Figure 1 – Mean value versus median value.**

## A Median Sorting Algorithm

Median smoothing can be implemented on a set of data by using a window of five consecutive data points, finding the median value, then moving the window over by one sample, and repeating the process until the whole original data set is smoothed.

To accomplish this, a simple sorting algorithm can be used to find the median of five values. RSP's equation facility is somewhat limited in terms of writing programs, so the challenge was to find a way to do this without actually moving any values, as is done in classic sorting routines, or using any programmatic loop structures.

One way to do it is to store all the intermediate sort results in variables and compare them one after the other in an appropriate sequence. I wrote the following algorithm; but, I claim no primacy—for all I know, someone invented the same thing 60 years ago. It works as show in Figure 2. The red numbers show the flow of values using the five sample values discussed above. The min(x, y), max(x, y), and Zn[x] functions are discussed in the next section.

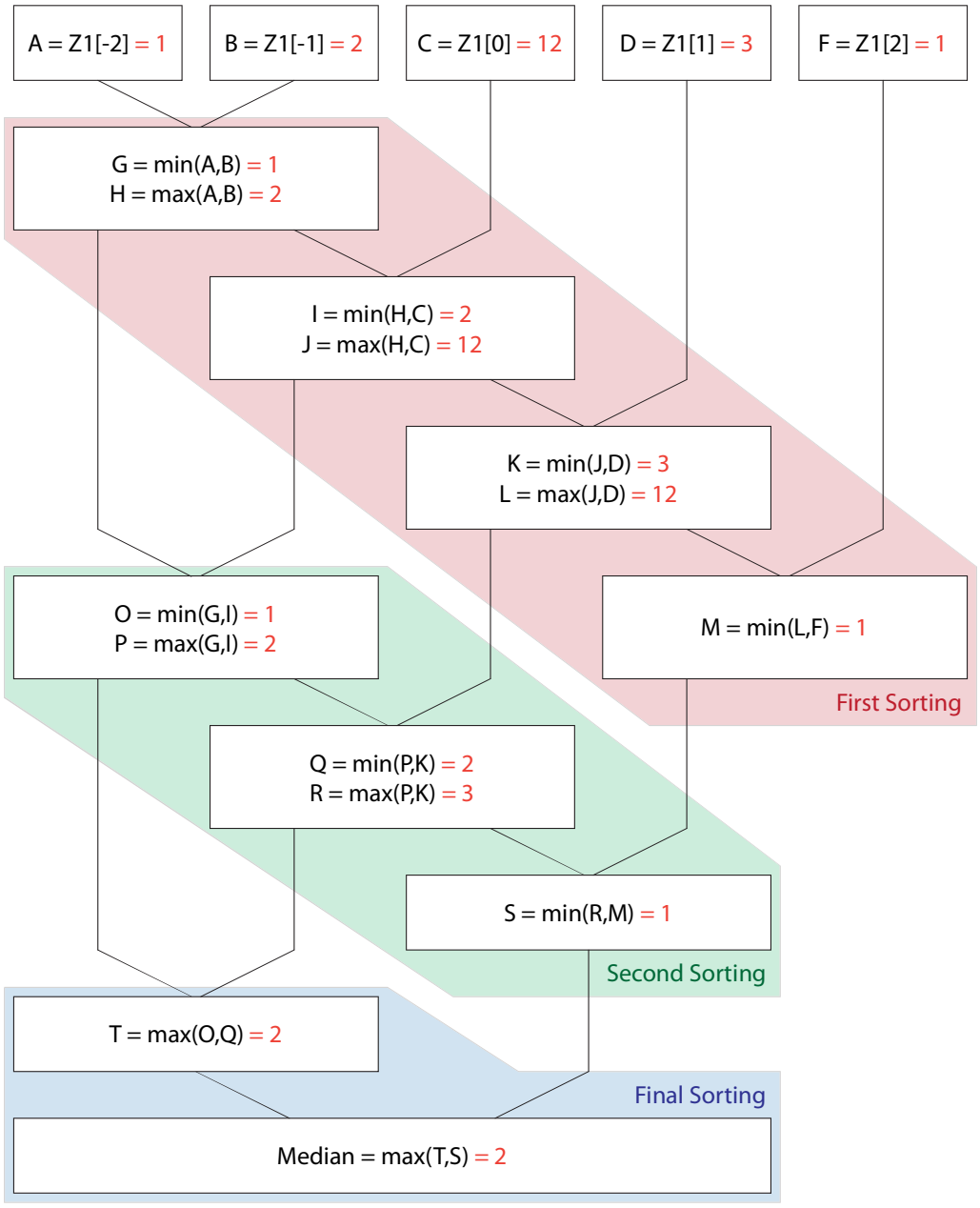


**Figure 2 – Algorithm flow chart for finding the median of five values.**

### Implementation of Median Smoothing in RSP

The algorithm described above may be applied in RSP to a channel's data by entering the equation shown below into RSP's post-processing function accessible through Tools -> Apply Equations to Data Points. The equation (which is actually a short computer program) sorts a window of five RSP data points centered on the current sample, then writes the median of the sorted list of five samples to the data in a new channel. RSP then moves the window one sample down the line, and repeats the process until the entire data file has been smoothed.

In RSP equations, the channel data is accessible via

$Zn[x]$

where

$n$ is the channel number (i.e., 1 through 8), and
$x$ is the sample offset from the current position during iteration.

For example,

Z1[-2] = two samples into the past on Channel 1
Z1[0] = current sample on Channel 1
Z1[2] = two samples into the future on Channel 1

When RSP executes the equation below, it starts at the first sample in the file and iterates across all samples. As such, this routine produces garbage on the first two and last two samples because Z1[-1] and/or Z1[-2] won't exist when the current position (the center of the five sample window) is at the first and second samples of the data file and Z1[1] and/or Z1[2] won't exist when the current position is 0 or 1 samples from the last sample in the data file.

The median smoothing equation shown below uses RSP's built-in min($x$, $y$) and max($x$, $y$) functions. Note that you cannot use capital letters for these two function names in RSP, they must be lower case.

max($x$, $y$) evaluates to the larger of $x$ or $y$
min($x$, $y$) evaluates to the smaller of $x$ or $y$

Median Smoothing RSP Equation – Human Readable Format

```
A=Z1[-2];B=Z1[-1];C=Z1[0];D=Z1[1];F=Z1[2];
G=min(A,B);H=max(A,B);
I=min(H,C);J=max(H,C);
K=min(J,D);L=max(J,D);
M=min(L,F);
O=min(G,I);P=max(G,I);
Q=min(P,K);R=max(P,K);
S=min(R,M);
T=max(O,Q);
max(T,S)
```

Median Smoothing RSP Equation – Collapsed Form
*suitable for copy-and-paste into the RSP Equation Editor*

```
A=Z1[ 2];B=Z1[ 1];C=Z1[0];D=Z1[1];F=Z1[2];G=min(A,B);H=max(A,B);I
=min(H,C);J=max(H,C);K=min(J,D);L=max(J,D);M=min(L,F);O=min(G,I);
P=max(G,I);Q=min(P,K);R=max(P,K);S=min(R,M);T=max(O,Q);max(T,S)
```

## Real World Example

Shown in Figures 3, 4, and 5 is a comparison of averaging versus median smoothing. Figure 3 shows original data. Figure 4 shows generic averaging. Figure 5 shows median smoothing. No noise removal technique is perfect. Median smoothing does a better job of cleaning up relatively short duration noise spikes than averaging does—however, this may or may not be suitable for your needs. The degree to which we smooth the data is always tempered by our desire to avoid removing real data that just happens to look like the noise we want to remove. How much smoothing is too much is a question that has no easy answer; it all depends on what information you're wanting to extract from your data.

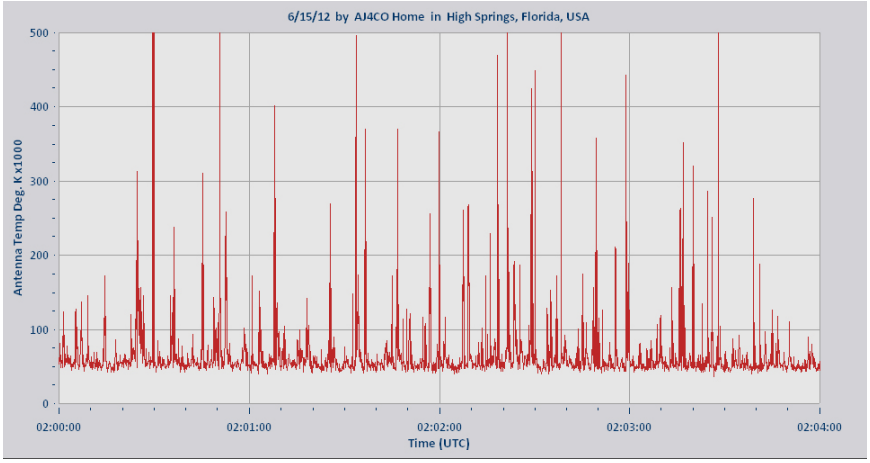


**Figure 3 – Here is some noisy real world data (10 Hz sample rate, 20 MHz galactic background with terrestrial noise).**

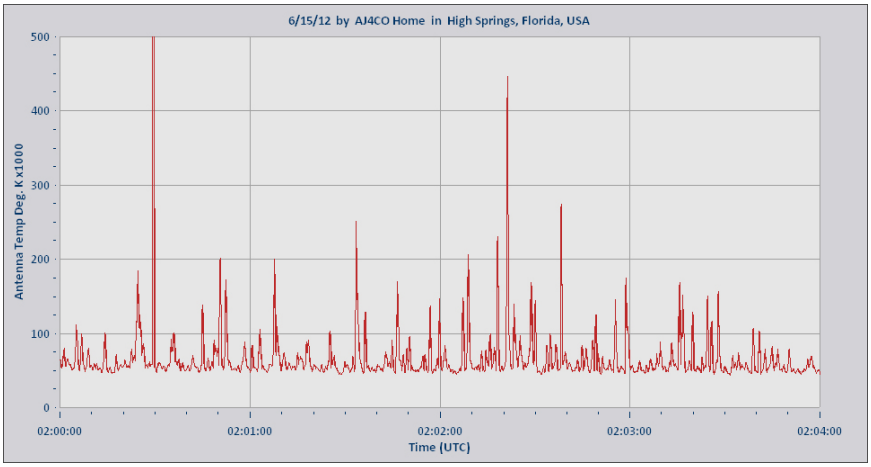


**Figure 4 – Here is the data from Figure 3 after one pass using RSP's built in Smooth by Averaging routine with a bin width of 5 samples.**

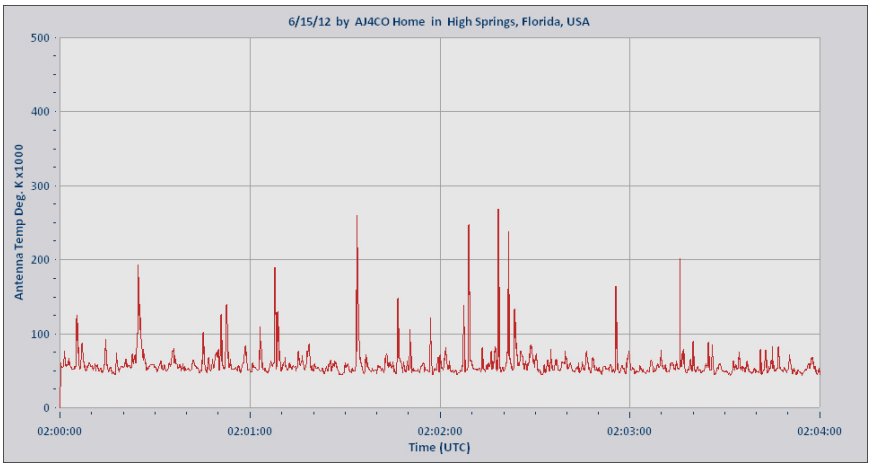


**Figure 5 – Here is the data from Figure 3 after one pass using the median smoothing algorithm described in this article.**

## Smoothing Parameters

When using median smoothing with small windows, an odd number of data points is necessary, otherwise the middle element is ambiguous—for example, in a set of six values, is value #3 or #4 the middle element? You could always interpolate, but I feel it is simpler to use an odd number of samples and have done with it. With larger windows—thousands of data points—this is less of a problem because the two middle elements (in a set with an even number of elements) are, statistically, going to be close in value compared to the range of the values in the window.

I chose a window of five data points because three seemed too small and seven or more would have required a longer algorithm. It should be noted that I did not try to write one for seven data

points. I'm neither sure a longer equation would fit in the RSP equation editor nor sure whether RSP's equation editor would have enough variable names to handle the required program. This is left as an exercise for the dear reader.

If one knows the sample rate of the data and the suspect noise spike characteristics, one can then make an educated determination of the window width required. For example, with a 20 Hz sample rate and suspected noise spike widths of less than 50 ms, it is possible that a three sample window might work. In that case, there would be two samples—ideally—at the true signal level and one sample at the noise spike level. However, if things aren't ideal, the noise spike could straddle two samples and the median value of a three sample window would not be a good indicator of the true signal. So, it is better to go with five samples in this case.

Using a window with too many samples, however, can be just as bad: we don't want to remove any spikes that are actually in the signal that we desire to see. Jovian S-bursts and short L-bursts, for example.

Note also that it is possible to run the median smoothing routine more than once, re-smoothing the already smoothed data, making it even smoother. Such multiple passes come in handy if you're trying to flatten everything to baseline while still leaving some of the signal variability intact (as opposed to finding the minimum within a given window, which completely flattens everything in the window).

I have not tested this algorithm extensively against known solar and Jovian emissions. My aim was to generate smooth traces of the galactic background over the long term, to detect the sidereal motion of the galactic plane—and this algorithm seems to work for that, depending on the severity of the noise. I do know that median smoothing with a window of five samples works quite well to de-noise the neutral hydrogen spectrum plots of the data generated by the seven meter radio telescope at Jodrell Bank.

As with all efforts to smooth data and remove unwanted noise, there is an unavoidable degree of guessing involved with determining the best set of parameters to use—and the proof of the pudding is in the eating. Try different parameters and see what works best. It is possible that median smoothing—or any other kind of data smoothing—simply won't work at all, especially if the signal you're trying to see is similar to the noise you're trying to remove; you'd end up removing the noise and the signal. We're in a tough racket: cosmic signals are just noise (no alien signals so far), interference is noise, and trying to remove one kind of noise from another is a tough nut to crack—and is often impossible.

One important thing to remember is that no matter what data sifting methods you use, they should always be described in detail in any published work based on the data.

## Further Reading


See the Radio-SkyPipe II help file for more on writing equations for RSP (under Pro Features).
**http://www.radiosky.com/skypipehelp/V2/skypipehelpindex.html**



Dave Typinski is a professional businessman and amateur scientist who has been tinkering with things electrical and mechanical since he was old enough to hold a soldering iron and a Crescent wrench. His primary scientific interests are radio astronomy, mathematics, and the history of technology. Dave is an amateur radio operator, call sign AJ4CO, and is an editor of *Radio Astronomy*. He is an active member of the Radio Jove project and operates a radio observatory that provides real time strip charts and spectrograms of interference with a little bit of solar and Jupiter mixed in.