

## Using Statistical Measures to Filter a Noisy Wave Transmission

Dean Ripley

Faculty Sponsor: Dr. Robert Hoar,  
Department of Math

The original proposal for my research was to examine voice analysis. This topic sounded very exciting and intriguing to me. The only thing I failed to consider was the fact that voice analysis is not a simple topic to research. There is underlying material that must be learned first. The first step towards this goal was to examine basic waves.

The basis of my project then changed to examining wave transmissions. Wave transmissions are common in everyday life. Being a computer scientist I use lots of wave transmissions when I use my modem. I was curious as to what methods there were to clean up transmissions over say a phone line. As anyone that has been on a phone can attest to, phone lines are never free of noise and static. So the question arises as to how a computer is able to decipher what a wave actually is that comes in with static or noise attached.

What I did was generate a wave of a certain size and this would be a "clean wave." A random amount of noise was then introduced into this wave. Afterwards, I would attempt to recover the clean wave given only the noisy data transmission. It sounds like an easy process and rather straightforward. I found it to be quite the opposite actually.

The first program I worked on, I tried to make as easy as possible. That way I would slowly learn what I was looking for, and I would not have to jump into the thick of something I knew nothing about. I generated a random wave (Graph #1), and then introduced a certain amount of noise into the wave (Graph #2). The easiest way for me to look at the wave would be to use a linear search. A linear search is basically a progression from one end to the other, stopping at every point in succession.

The first five points of the wave are stored and these are used as the "zero" or the "baseline" for the algorithm. The "baseline" for the wave is actually the standard derivation of those five points. Another value is calculated which is a cut-off point. The cut-off point is assigned as a constant amount times the standard derivation of the first five points (baseline). The wave is then traversed point by point in a linear search. Any spot in the wave where there is a jump that is larger than the cut-off value is marked as a spike location. A spike location would be defined as: a change in data such as a change in state from a zero to a one in a wave transmission. Once the end of the wave has been reached the "cleaning" process can begin. Since the locations of the spikes are known it is easy to form a wave based on these locations and the initial baseline. The two graphs on the following pages portray the cleaned wave against the noisy wave. The cleaned wave is highlighted in yellow. The difference in the two graphs is that Graph #3 uses a cut-off constant of two, while Graph #4 uses a value of one and a half.

This was my first attempt at "wave cleansing". The results I got were not that great though, there were a lot of problems I either went around or ignored using a set test case.

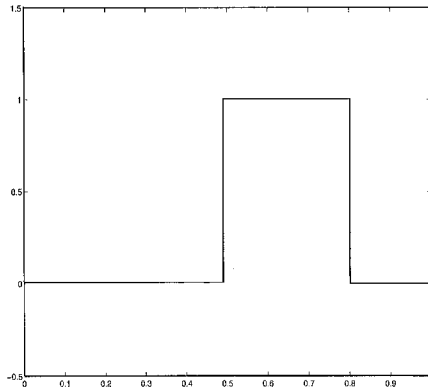
The first problem was that this approach uses a set value as means of determining spike locations. As shown on the graphs on the following pages, depending on what constant is used for the cut-off a variety of results can occur. Manipulation of this constant based on each wave is an unpractical solution. I wanted to achieve an algorithm that would work with any data set, and not have to change my algorithm to fit any particular set.

Another occurrence in waves is stray data points or blips. My first approach did not even take these points into consideration. I never had to worry about any blips as I was the one who controlled the data and also the amount of noise. This exclusion made this approach a little too unrealistic.

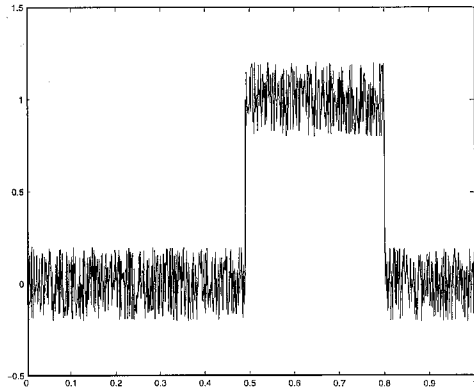
The last item I wanted to change was in the design. The algorithm I used had a linear search in it. These types of searches are very time consuming. If an extremely long wave transmission was looked at, it may take a very long time or a lot of processor power to perform the iterations. The ideal way would be to use some sort of looping structure that would hopefully increase efficiency and shorten the time involved to traverse the wave.

I started my second approach by looking at the faults of my first and hoped to improve some performance. I used a little more of my math background in this algorithm as compared to the first. The initial approach was the same in this case. I read in a random set of numbers that I made. Graph #5 and Graph #6 on the next pages refer to these initial points.

The main difference in this approach was the introduction of a key measurement from statistics called a z-score. The z-score is "The distance (in standard deviations) between a given measurement and the mean." It is calculated by taking the given value minus the mean and dividing by the standard deviation. This idea corresponded well with a paper I came across entitled "Maximum Z Scores and Outliers" written by Ronald E. Schiffler. In the paper, the question posed was "How large can the largest z-score be for a set of data?" Based on this paper, using a sample size of five data points the z-score cannot be greater than two (assuming a uniform distribution).



Graph 1

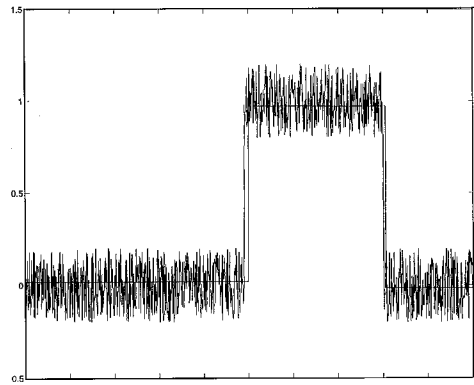


Graph 2

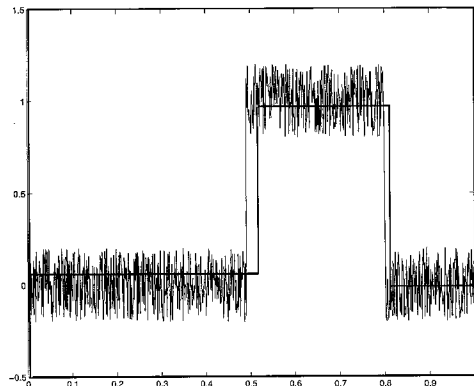
Using this idea I looked at the first five data points to again try to get a “baseline” of sorts. I needed to get a minimum value and a maximum value from these points. The minimum was the mean of the first five points minus double the standard derivative of the first five points. The maximum is the mean of the first five points plus double the standard derivative of the first five. This was the initial range of values I would use for the algorithm.

I then started to traverse the wave in another linear search starting with the sixth point. I looked at each spot in the wave, and if the point lies within the maximum and the minimum then nothing was done. The thinking here is that since it is still within my range then there has not been a spike, or the wave has not changed state. If a point was found that lay outside the range then there are two possibilities. The spot found is either a jump or a blip. A jump means that there are enough points to show that the wave has changed in signal. If there are not enough points, then the spot is classified as a blip.

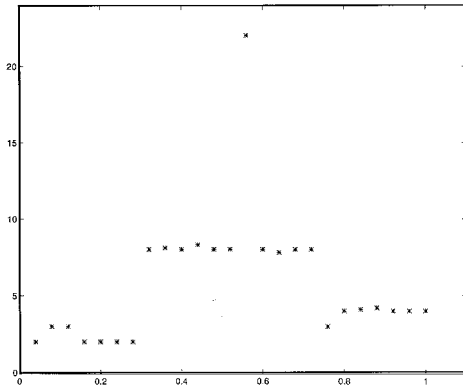
To distinguish between the two choices, the next four points are looked at. The number of these points that lie within and out of the range are saved. If there are at least three of the next four in the range, we can classify it as a jump. Those points that lie outside of the range are then classified as blips. The minimum and the maximum are then re-calculated using these new points. The only difference would be that when blips are present, their value is replaced by the mean of the wave so that they blend in. Looking at the jumps found and averaging between these jumps, the entire wave is cleaned. Graph #7 and Graph #8 on the following pages show the original points as stars, and use circles to show the cleaned up wave. Graph #7 turned out fairly reasonable, but Graph #8 had a lot of error in the end of it. The reason is in the handling of the blips. This was my second attempt to clean waves, and I thought it was a dramatic increase from the first. For starters, I took a totally different approach this time. The introduction of z-scores seemed to be a better solution to the problem, as it would provide better accuracy and conventionality. The need to spot blips was a very logical and needed course of action. As blips are a common occurrence in



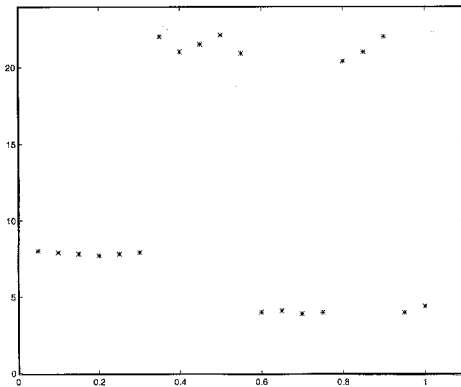
Graph 2



Graph 4



Graph 5



Graph 6

transmissions, they must be accounted for.

There were still negatives of this approach though. One of the problems that I found was actually a reason that I changed to using z-scores in the first place. The use of z-scores was added to provide flexibility. This was achieved to a point, but in the handling of blips there are obvious problems. I had to develop a hypothesis that would characterize a point as a blip rather than a jump. The problem is that how does one really know? A few blips in a data transmission does not cause too many problems, but constant large jumping of data values exposed flaws in my design. I need a better way to figure out this distinction.

To this point the search is still linear, which still brings up a concern on larger waves. I left this as it was as I felt the need to fine-tune my algorithm before changing my search pattern. If I had the time, or anyone continued this work, I would suggest creating a better way to look at blips and to look at a larger surrounding error. The looping structure would also be a nice addition. To distinguish between the two choices, the next four points are

looked at. The number of these points that lie within and out of the range are saved. If there are at least three of the next four in the range, we can classify it as a jump. Those points that lie outside of the range are then classified as blips. The minimum and the maximum are then re-calculated using these new points. The only difference would be that when blips are present, their value is replaced by the mean of the wave so that they blend in. Looking at the jumps found and averaging between these jumps, the entire wave is cleaned. Graph #7 and Graph #8 on the following pages show the original points as stars, and use circles to show the cleaned up wave. Graph #7 turned out fairly reasonable, but Graph #8 had a lot of error in the end of it. The reason is in the handling of the blips. This was my second attempt to clean waves, and I thought it was a dramatic increase from the first. For starters, I took a totally different approach this time. The introduction of z-scores seemed to be a better solution to the problem, as it would provide better accuracy and conventionality. The need to spot blips was a very logical and needed course of action. As blips are a common occurrence in transmissions, they must

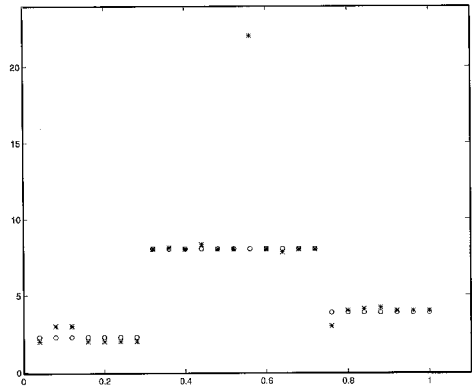
be accounted for.

There were still negatives of this approach though. One of the problems that I found was actually a reason that I changed to using z-scores in the first place. The use of z-scores was added to provide flexibility. This was achieved to a point, but in the handling of blips there are obvious problems. I had to develop a hypothesis that would characterize a point as a blip rather than a jump. The problem is that how does one really know? A few blips in a data transmission does not cause too many problems, but constant large jumping of data values exposed flaws in my design. I need a better way to figure out this distinction.

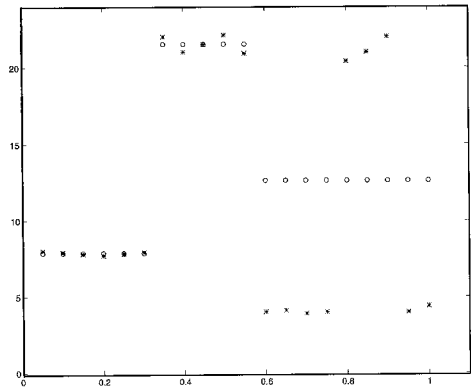
To this point the search is still linear, which still brings up a concern on larger waves. I left this as it was as I felt the need to fine-tune my algorithm before changing my search pattern. If I had the time, or anyone continued this work, I would suggest creating a better way to look at blips and to look at a larger surrounding error. The looping structure would also be a nice addition.

The last part of research done was partly for amusement and partly for curiosity. I was given some data that was actually of an image. The image given to me was a cross-section cut of someone at the heart level. Graph #9 shows this picture on the next page. The larger white area is the actual heart material. The picture is obviously very noisy as can be seen by the fuzziness. I decided to run this data through my algorithm to see what it would do. I hoped that it would sharpen up the image of the heart and make it clearer. The results I wanted would be to show the distinction in the heart material from the non-heart material. The Graph #10 on the following page shows what happened.

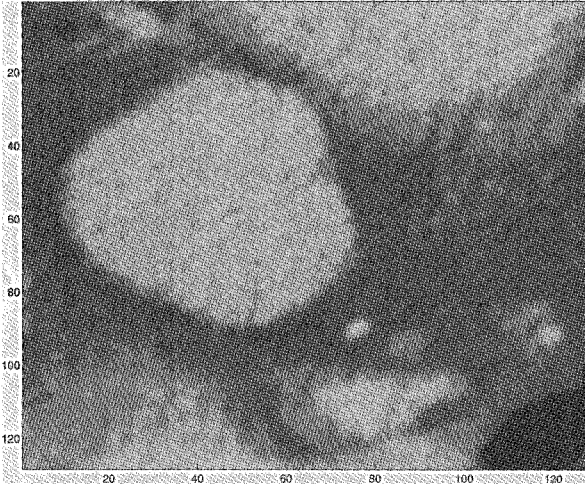
This cleaned up graph says a lot about the limitations of my coding. My program worked well upon numbers, as these are only two-dimensional. The heart is a three-dimensional image, and that is why there is the apparent hacking off of the image around the edges of the heart. My algorithm does not allow for a slope or a curvature at this point.



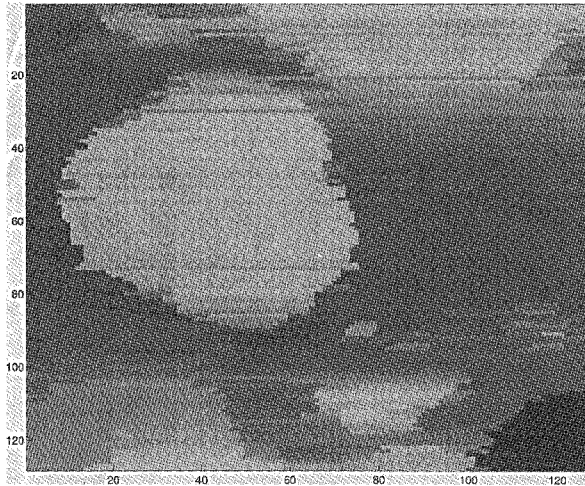
Graph 7



Graph 8



Graph 9



Graph 10

Although my research covered waves rather than voice analysis, I am extremely glad I did it. It was a fun project to work on, and I got to learn some new information and work on exciting technology. I would like to say that without the help of my adviser I would have gotten nothing done. Thanks go out to Dr. Robert Hoar of the Math Department. Anyone looking for a good advisor should look for him. In closing, I hope someone comes along and decides to continue in this area. I have laid the foundation, now someone else must build the walls!