Zachary McNulty

AMATH 342

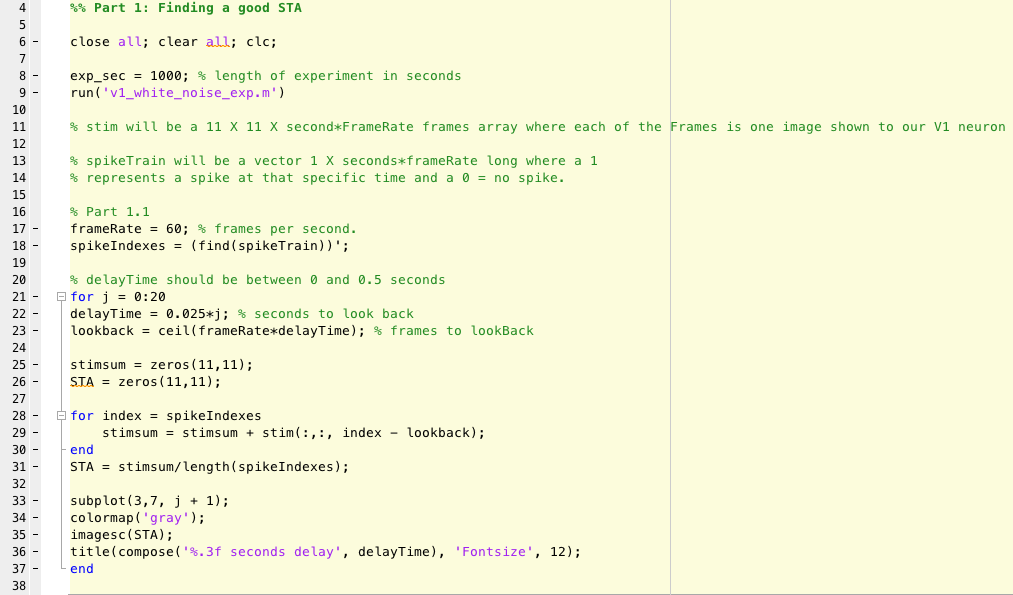
HW # 2

**Part 1: Finding A Spike Triggered Average**

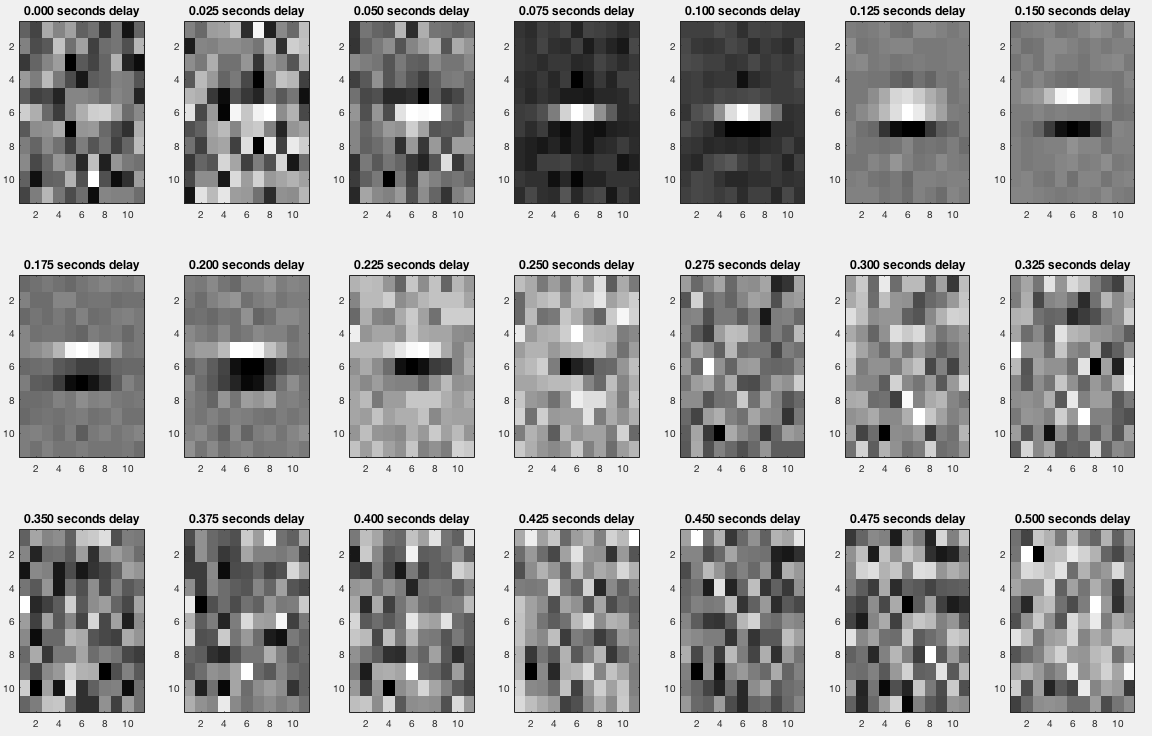
1.1)

Below is the code for determining Spike Triggered Averages for a given set of stimuli. The code starts off by running v1\_white\_noise\_exp.m, a program which simulates the activity of a specific neuron in response to a series of 11 x 11 pixel images (lines 6-10). While we have a list of all the stimuli at any given time and a list of all the time points a spike has occurred, it is still uncertain what stimuli triggered the spike at any given time because the spike might not occur immediately following the preferred stimulus. Thus, there is a delay time between the preferred stimulus and the spike, and this code works to determine that delay and the preferred stimuli.

To do so, it averages the stimuli values, the 11x11 pixel images, preceding each individual spike by a given delay time, saving this STA. Since our stimulus values are measured in frames, not seconds, I first convert between the two units (line 23). It then repeats this for a series of delay times between 0 and 0.5 seconds (lines 21-31). Lastly, in order to pick out the most discernable pattern and thus decide upon the best delay time, it graphs the STA’s for each given delay time, graphing the averaged 11x11 pixel images. I graphed these images in black and white because I felt it made it easier to determine what the extreme and intermediate stimuli values were.



Below are the STAs for a series of varying delay times. In terms of the clearest image, and thus the ideal delay time, I am looking for an image that has the least random pattern. Areas of interest should be significantly below or above the mean pixel value, while insignificant pixels should average out to the mean (grey). For these reasons, I believe the optimal delay time is around 0.150 seconds. This image suggests the neuron prefers a horizontal orientation of light/darkness, with a section of high illuminance above one of very low luminance, creating almost a gradient between the two. As we only see this pattern in the very center of the image, it also suggests the neuron might be sensitive to the spatial orientation of the bar of light as well. In a natural image, this could represent an area of high contrast, and that might be what the neuron is responding to.



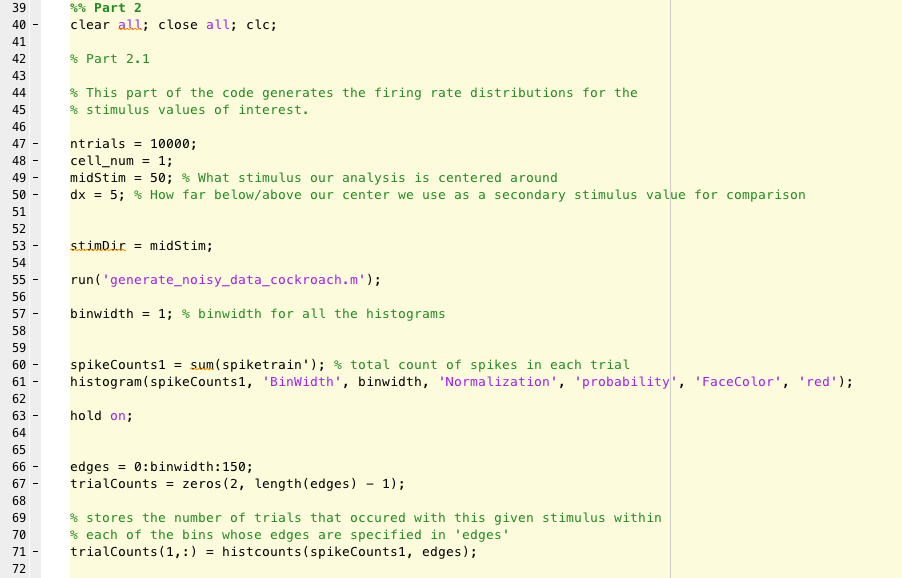
1.2)

The spike triggered average approach might be misleading if applied to neurons that react to multiple distinct types of stimuli, such as complex cells that link together the responses of several lower level simple cells. In these cases, these distinct preferred stimuli might be erroneously averaged together, creating an intermediate stimulus that the cell might not respond to. More importantly, these averages might seem to emphasize that multiple stimuli values have to occur together to get a response, but we know in actuality a complex cell can react to just a single one of these preferred stimuli as not all the simple cells it is attached to have to trigger a response to activate the complex cell.

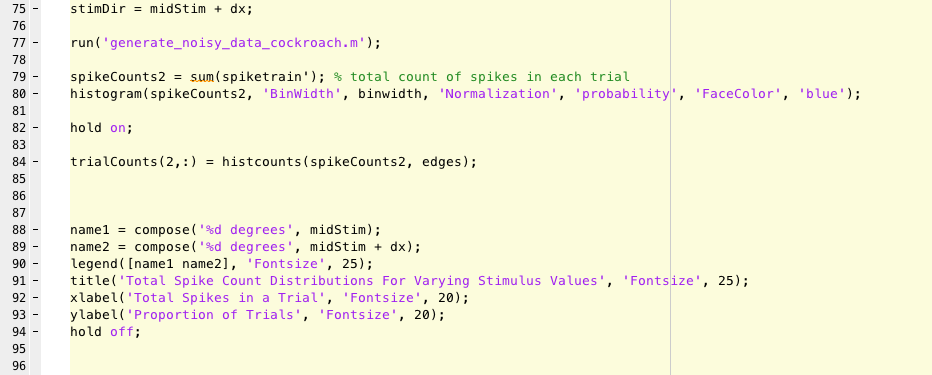
**Part 2: Maximum Likelihood Decoding**

2.1a)

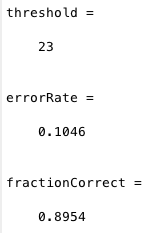
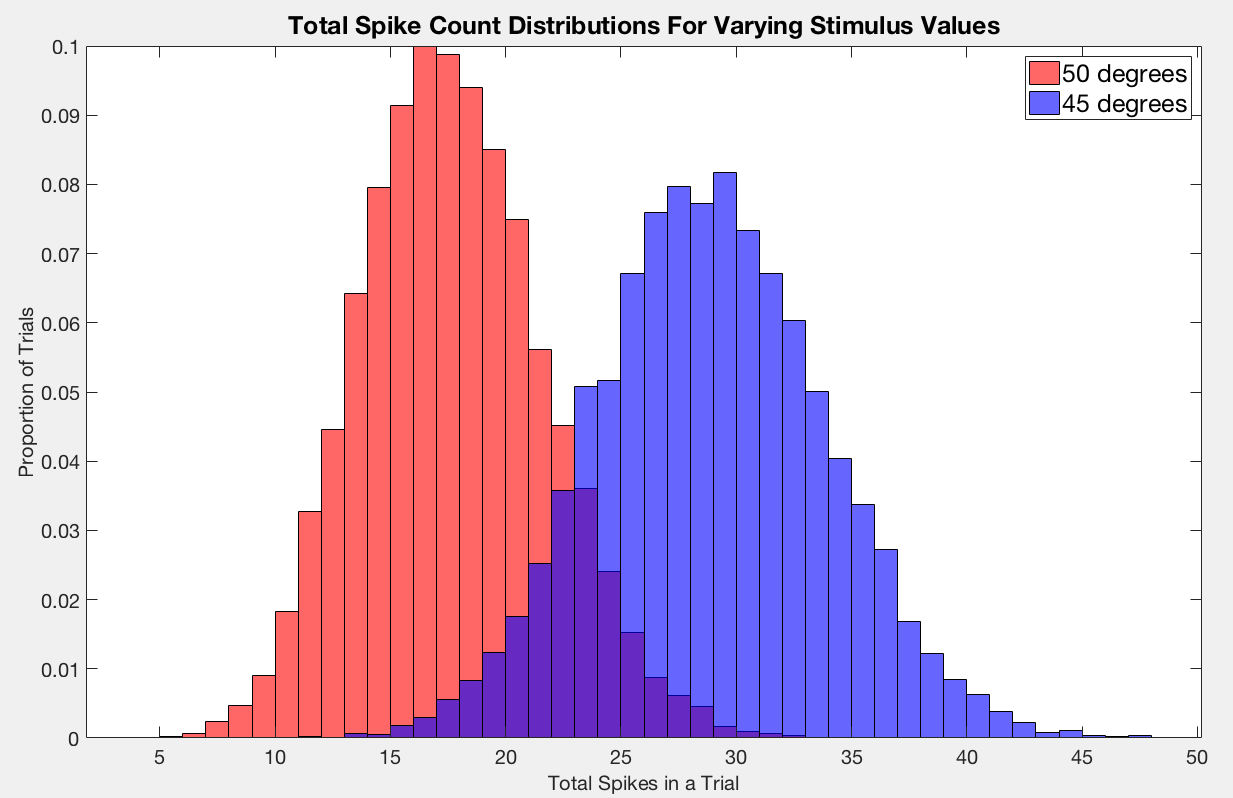
This section focuses on maximum likelihood decoding, an approach for guessing the stimulus value that produced a given set of spikes. The first section of this code generates the spikes to a given stimulus value, midStim, running many trials (lines 47-55). There is natural variation trial-to-trial in the total number of spikes per trial that the stimulus evokes, and I capture this variation by plotting the distribution in a histogram. The histogram is normalized so that the y-axis represents the proportion of trials that reached that given total spike count (x-axis). For later use, I store the counts of trials that landed in each bin of the histogram (trialCounts) and the edges of the bins (edges).



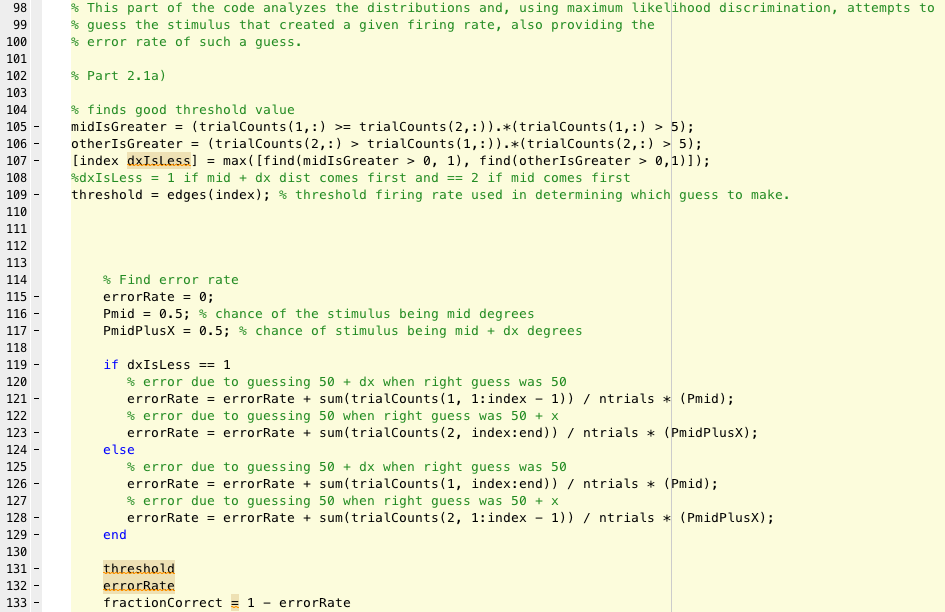
This section does the same as above, except with a slightly higher or lower stimulus value. By adjusting dx, I can see how close two stimulus values can get to each other before it becomes too difficult to differentiate between the two with any reasonable accuracy. The two histograms are plotted together to get a visual representation of the distributions of the two stimuli.



Below, we see an example of a histogram produced by the above code. As we see, the two stimulus values produce unique distributions, but the variation trial-to-trial is readily apparent in the spread of these distributions around a central mean. To use these distributions to guess the stimulus of an unknown trial, simply find the total spike count of that trial on the x-axis and see which distribution is highest at that point. However, while this guess is the most probable answer, this guess will not be correct every time due to the overlap of the graphs. While it is more likely that a trial with 25 total spikes came from a 45-degree stimulus, we see that there were also some trials with the 50-degree stimulus that had that many spikes.



In order to figure out how accurate my guess is, I will calculate the overall error rate of such a process. One way of doing this is to find a certain threshold value that separates the two distributions, and to find the chance that a trial occurs on the opposite side of this threshold than the rest of its distribution. Visually, this represents the overlapping areas of the histograms. To find this threshold, I find the index (i.e. the bin from the histogram) where one distribution stops being larger than the other, and check what Total Spike Count corresponds to this index (i.e. the edge value that bounds this bin) (lines 104-109). After that, I can use the threshold index to calculate the error by summing all trials that lie on the opposite side of this index from the rest of their main distribution. I do this for both distributions, and lastly, I normalize this error rate by dividing by the number of trials to convert it into a proportion (lines 114-133). I multiply these proportions by the chance that stimulus was presented because that given error (i.e. guessing 50 + dx degrees when 50 degrees was presented) can only occur if 50 was presented in the first place, and thus it represents a conditional probability.

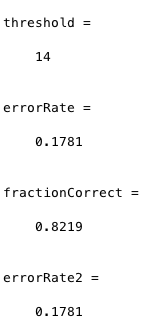
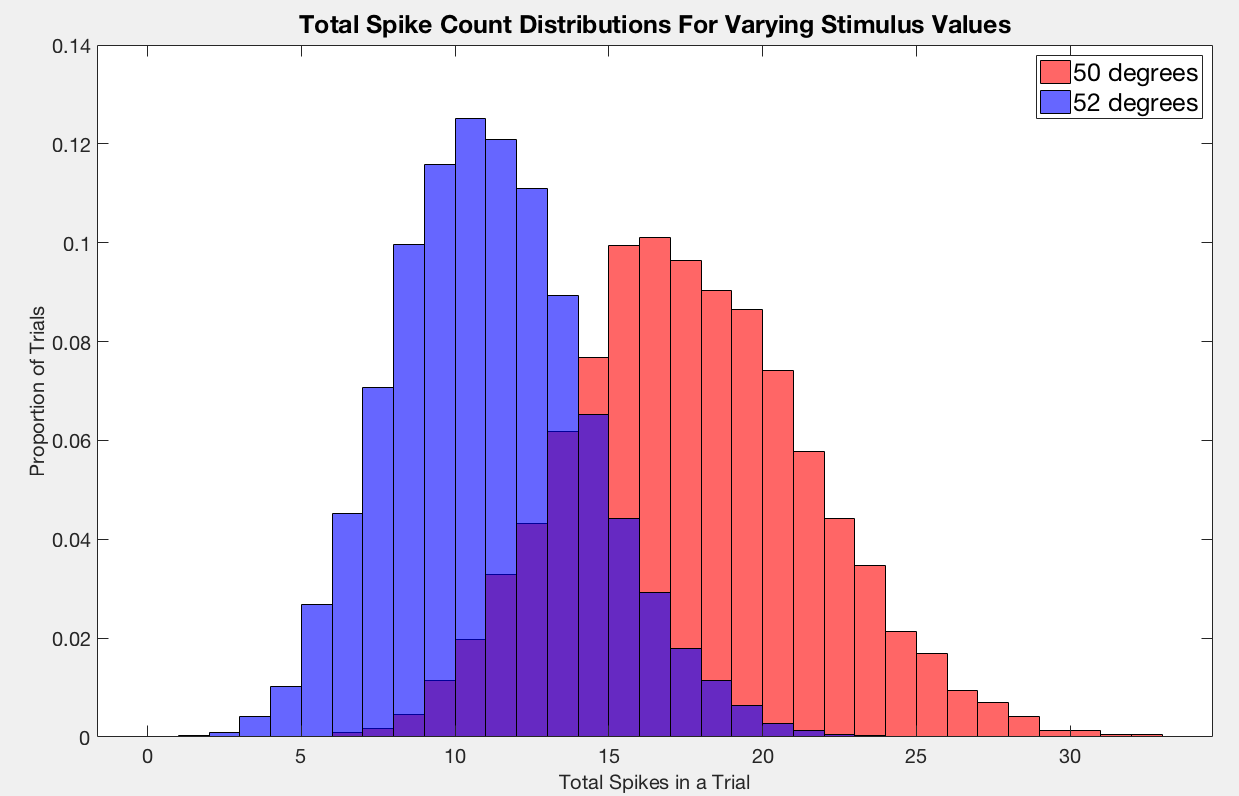


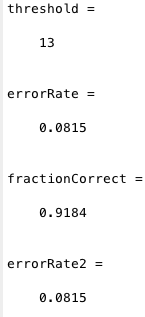
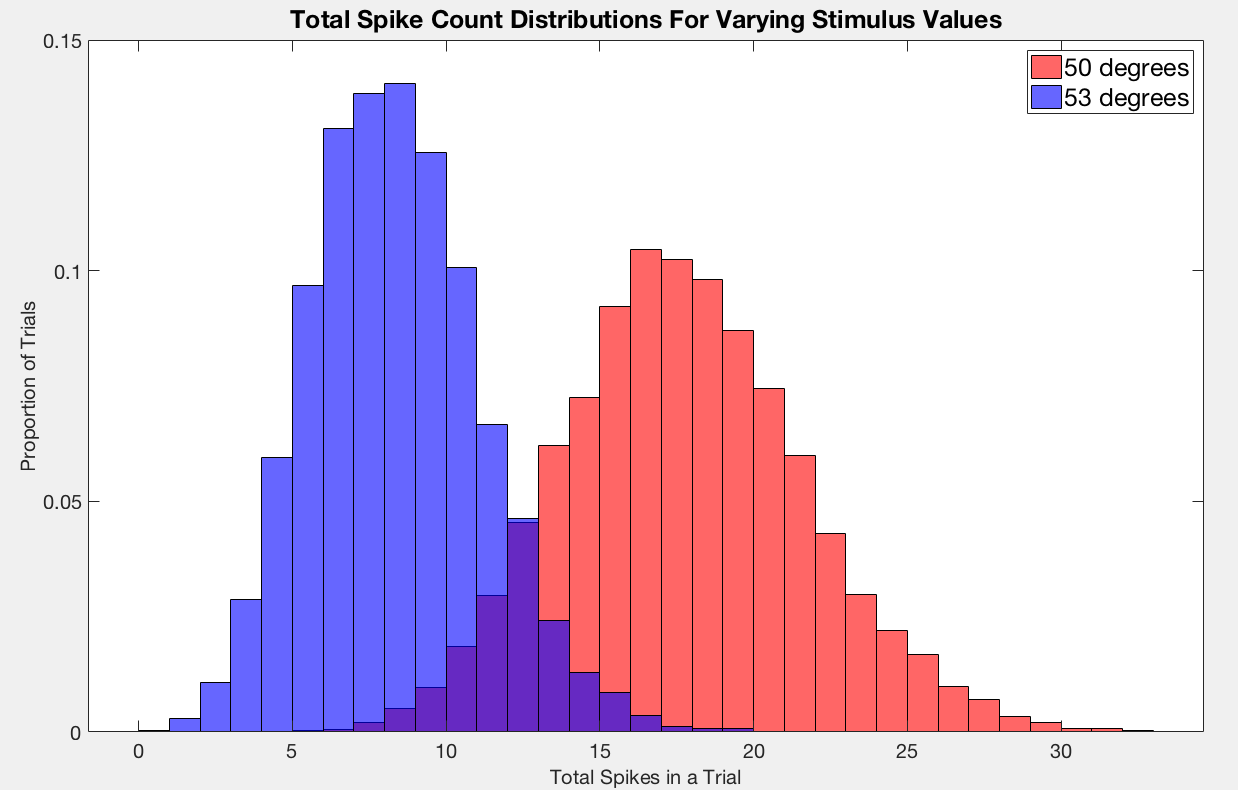
Another way of calculating the error rate is to simply calculate the chance a trial occurs in a bin of one distribution in which the other distribution has more trials that landed in that bin. First, I find all the instances/indexes where the first distribution is lower or equal to the second (line 135). I repeat the same procedure with the second distribution, only considering sections strictly less than to avoid double-counting certain sections (line 136). Next, I divide by the total number of trials in this distribution to normalize these values and convert them into proportions, and sum them all up to get the total proportion of trials in which this distribution is less than or equal to the other (line 137). Lastly, I multiply by the chance this stimulus is chosen, giving me the chance I wrongly guess the other stimulus was presented. This method might work better in cases where it is difficult to determine a single appropriate threshold value.

../../../../Screen%20Shot%202018-02-07%20at%2011.14.54%20PM.png

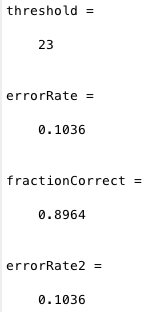
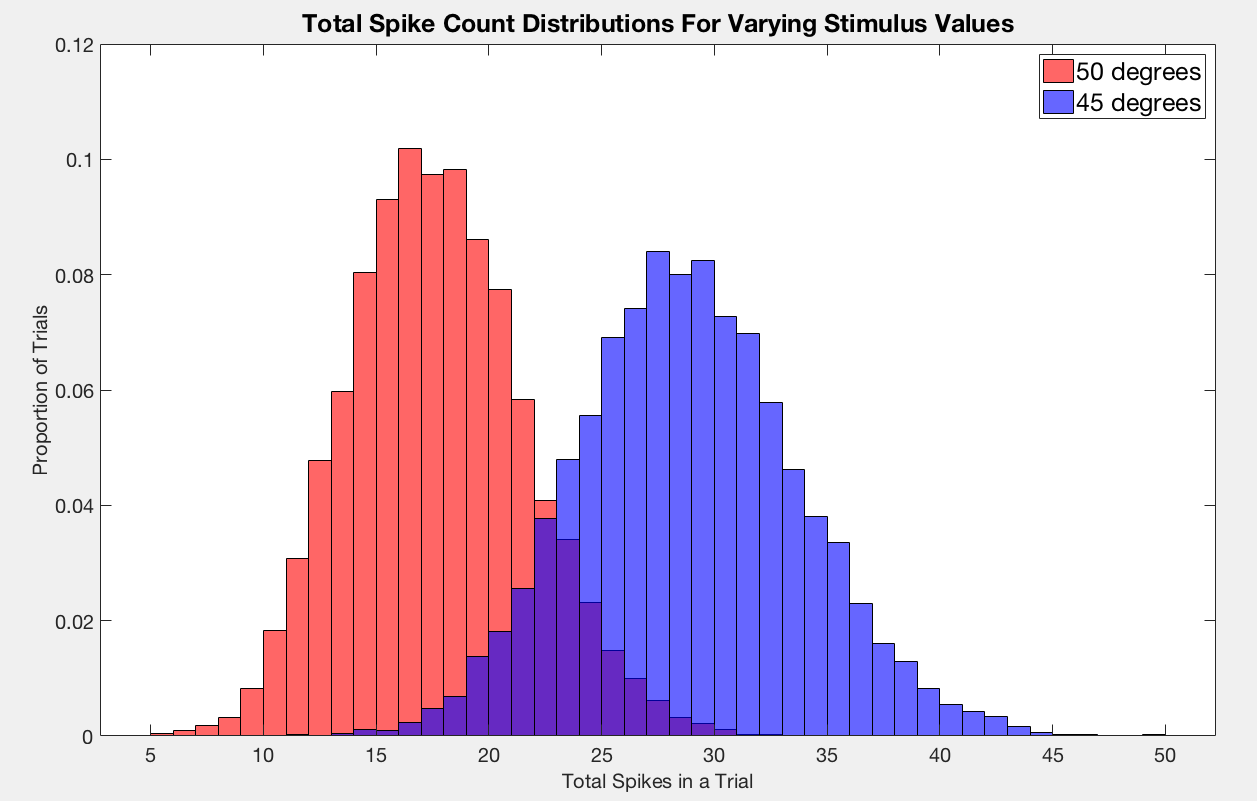
2.1b)

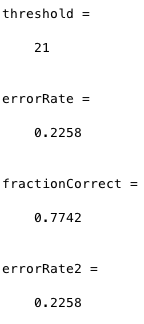
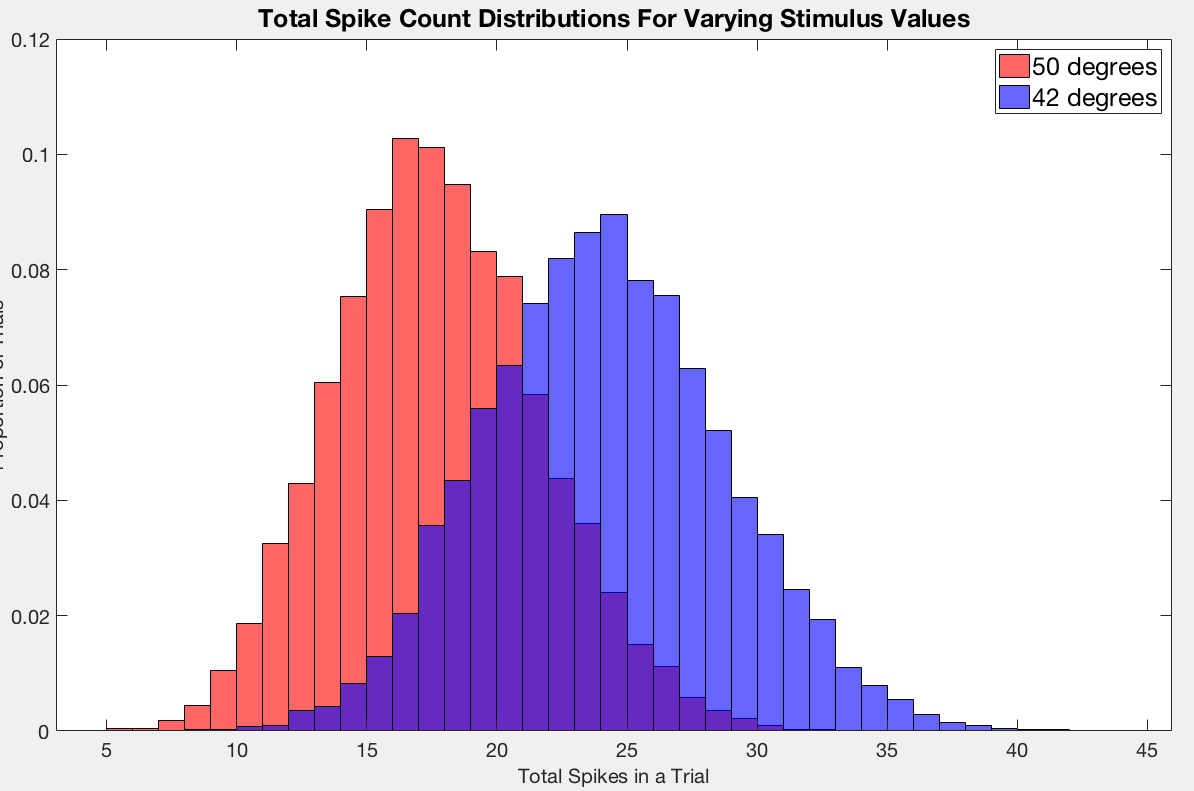
As we can see from these graphs, a stimulus value of 50 is differentiable (with an error rate of ~10%) from stimulus values somewhere around 3 degrees above 50. Less than 3 degrees above still creates significant over overlap, and thus the error rate is still greater than 10%.

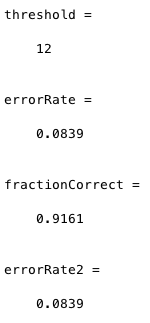
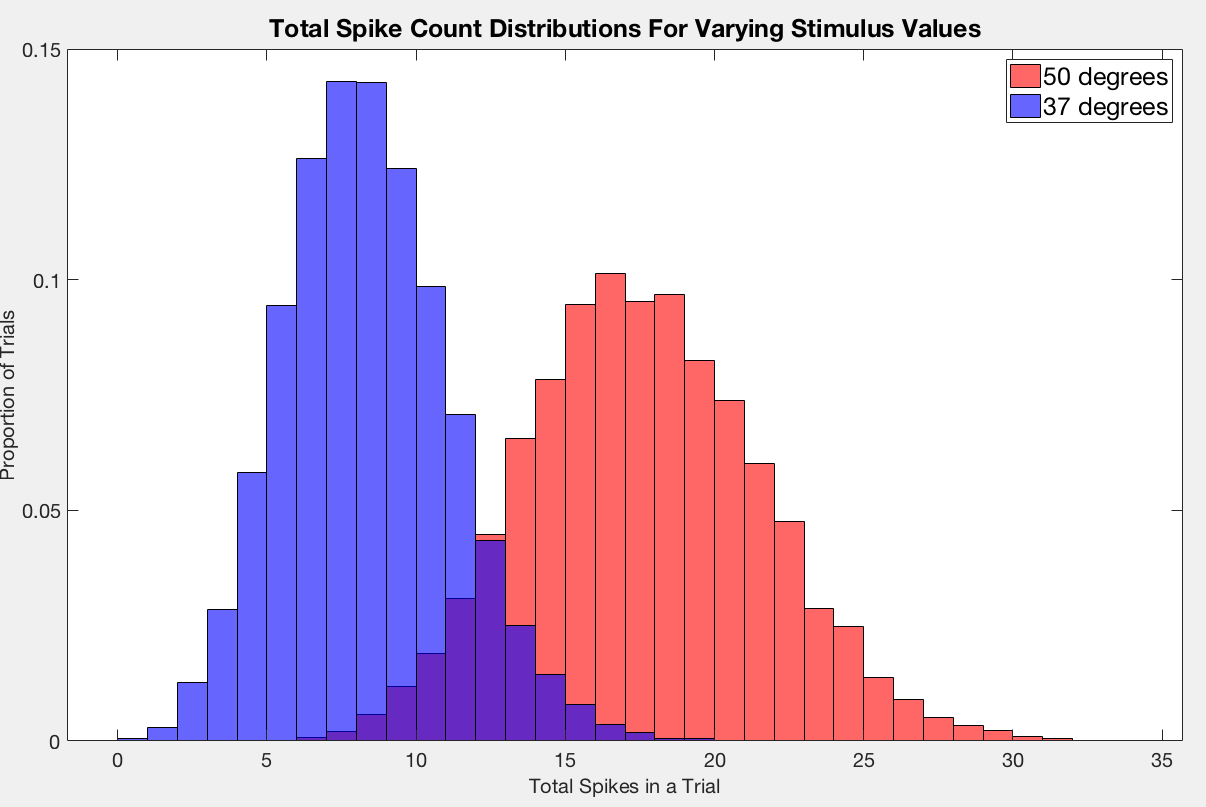




Below 50, however, there is an interesting trend. The error rate is pretty close to 10% at 5 below, but then as you drop farther below 50, the error begins to increase again! It is not until around 13 below that the error rate finally drops to around 10% again. Thinking back to the tuning curve cell 1 created in homework 1, this makes sense because the curve was symmetric around 45 degrees. As a result, a stimulus of 50 degrees is nearly identical to a stimulus of 40 degrees. This makes it difficult to differentiate a 50-degree stimulus from stimulus values close to 40 degrees.

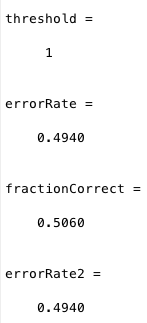
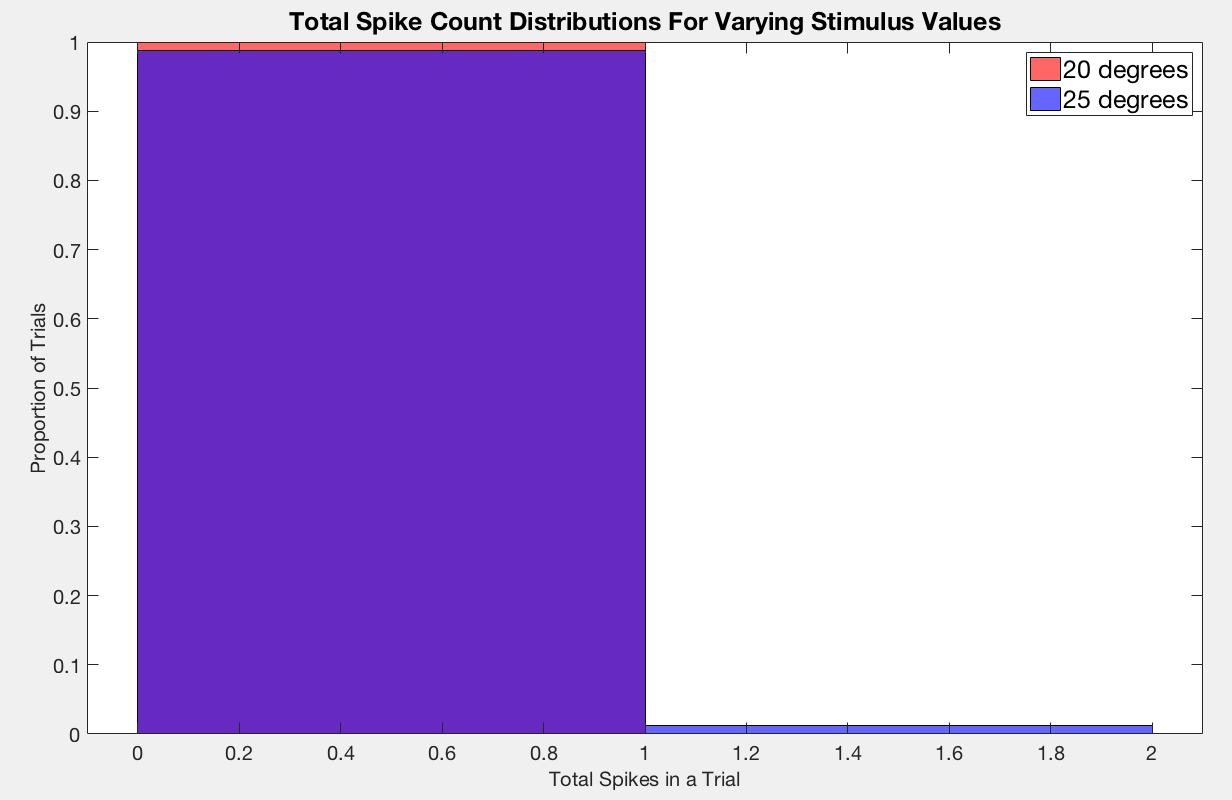


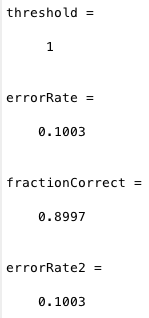
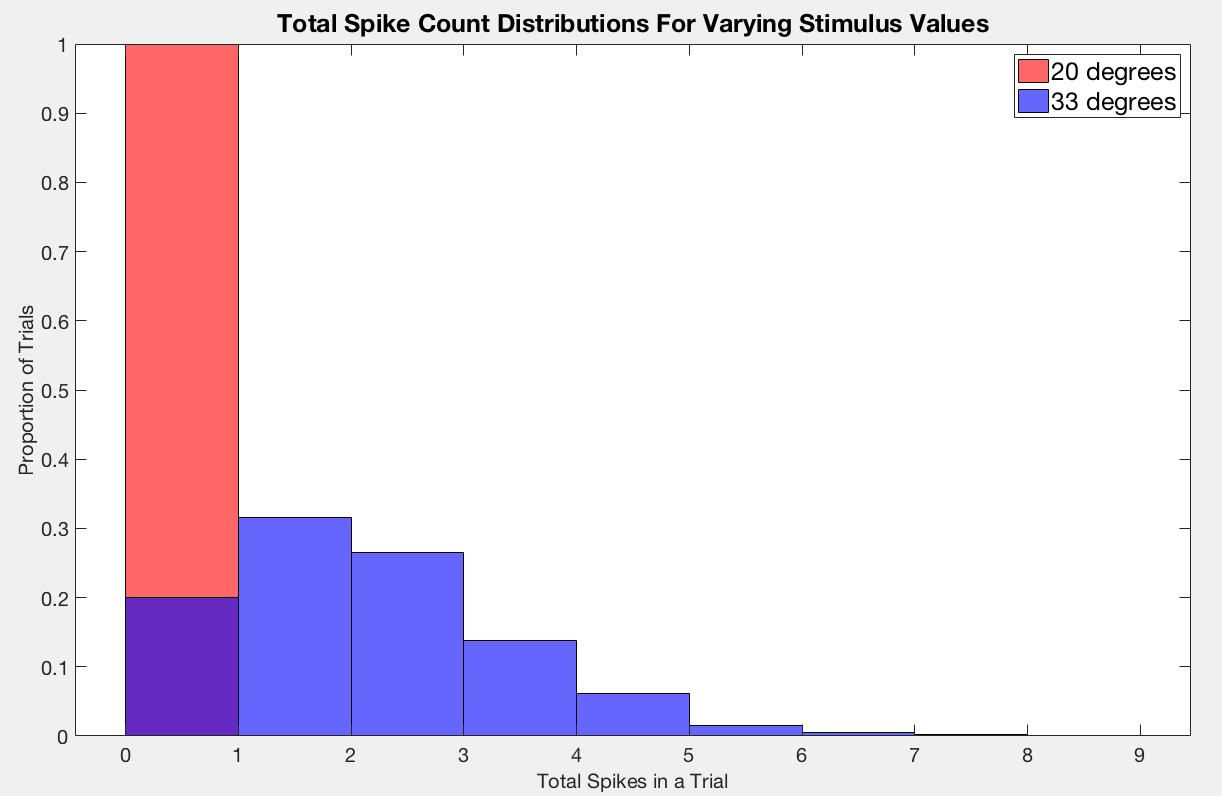




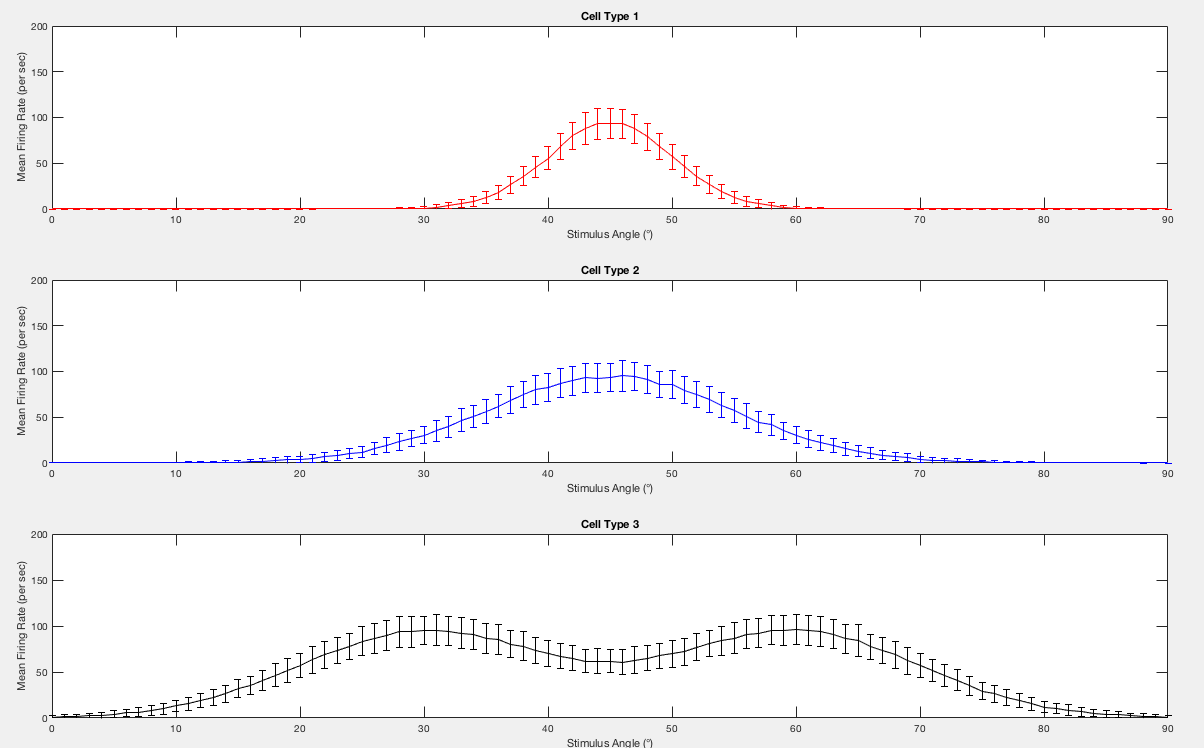
2.1c)

As we saw in homework 1, this cell shows practically no activity at stimulus values as low as 20 degrees. This is evident in the fact that nearly every trial landed in the 0 spike bin for both histograms. Thus, all nearby stimulus values are almost identical and practically impossible to differentiate. As we saw in homework 1, at values less than 20 there continues to be no activity so there will be no way to differentiate stimuli of 20 degrees from anything less than 20 with reasonable accuracy (I have included the tuning curve for Cell 1 from homework 1 as proof to this claim). When we go far enough above 20 degrees, around 13 degrees greater as seen in the second figure, we start getting significant activity and we are able to differentiate this stimulus from the lack of activity 20 degrees produces with appreciable accuracy (< 10% error).



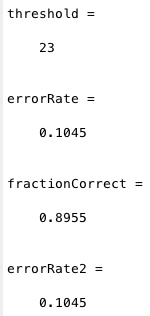
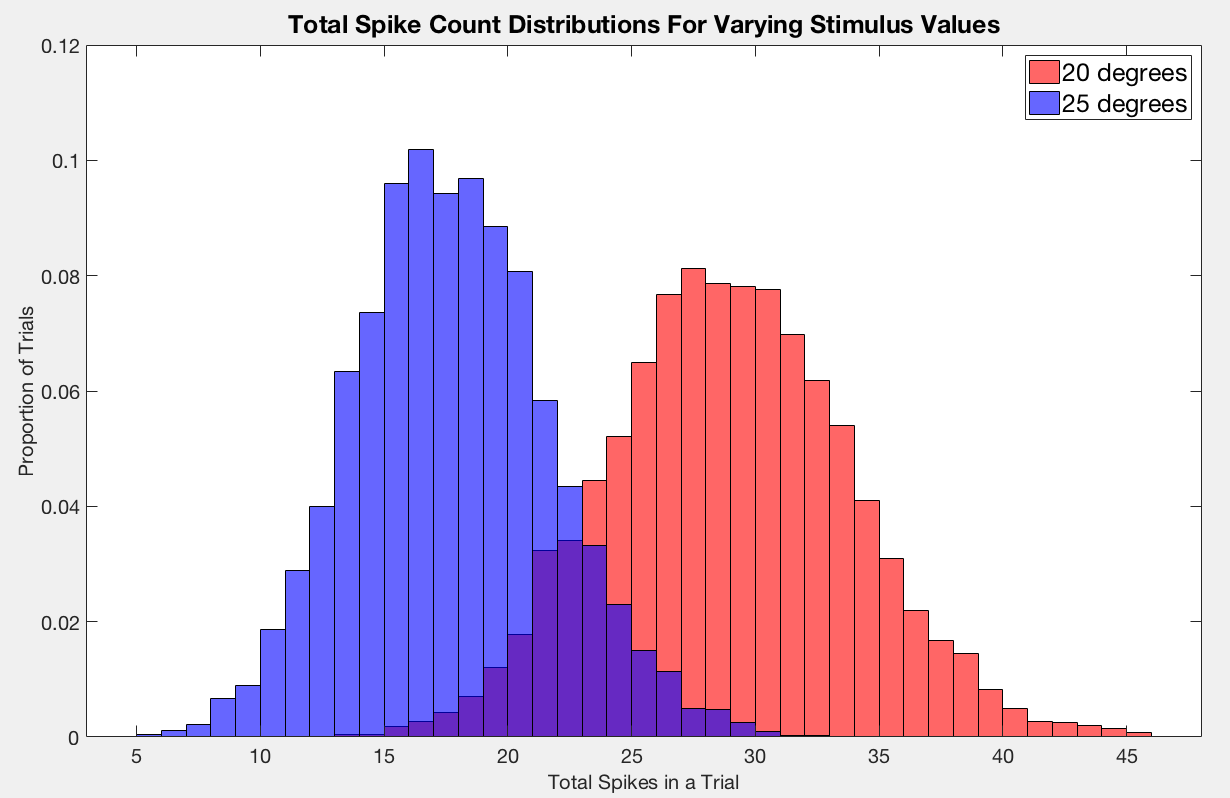


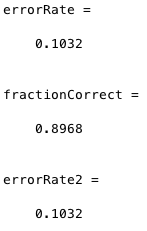
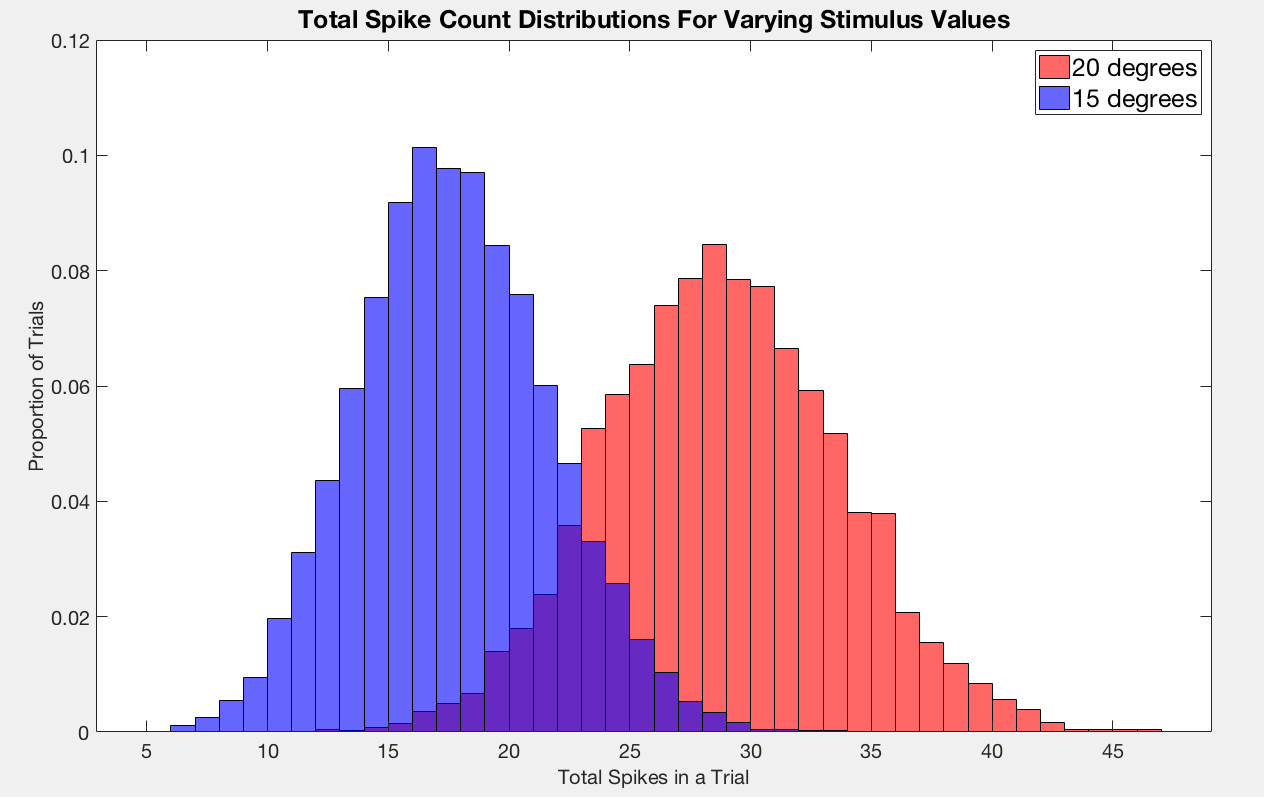
Cell 1 Tuning Curve



2.2)

In order to make it easier to differentiate between stimuli values around 20 degrees, I changed the mean (mu) of Cell 1 to be 20 rather than the 45 it was previously. Before, the main issue with trying to differentiate between stimulus values near 20 was the fact that the cell did not respond practically at all to stimulus values within that range, so there was little variation between nearby stimulus. We need some sort of variation stimuli to stimuli in order to differentiate between stimuli values. In homework 1, we saw how the cell’s responses to stimuli flat-lined quickly past the mean. The only true variation was occurring near the mean, so in order to get the variation needed to decide between two nearby stimuli values, I changed the mean to 20 degrees, so that the variation would occur around 20 degrees rather than around 45 degrees. Now the stimuli values only have to be separated by around 5 degrees to have an error rate below 10% compared to the 13-degree separation previously required. Below I have also included our tuning curve for Cell 1 from homework 1 to illustrate the lack of activity at and around 20 degrees and the fact that almost all variation is occurring close to the mean of 45 degrees.





Cell 1 Tuning Curve

