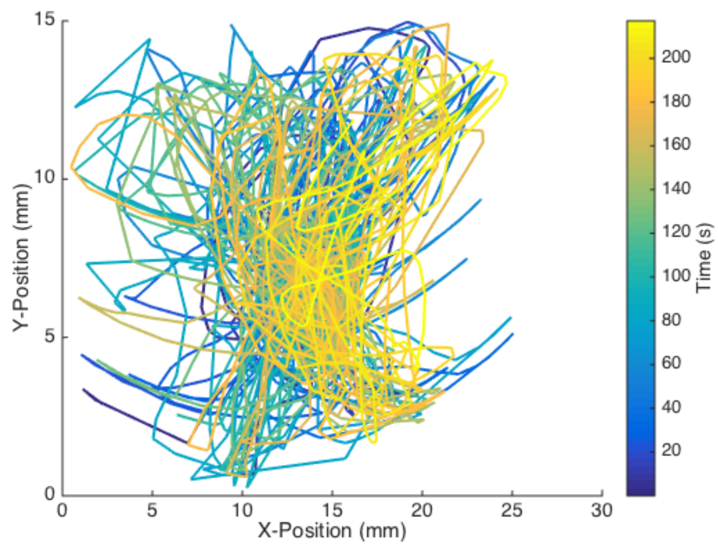
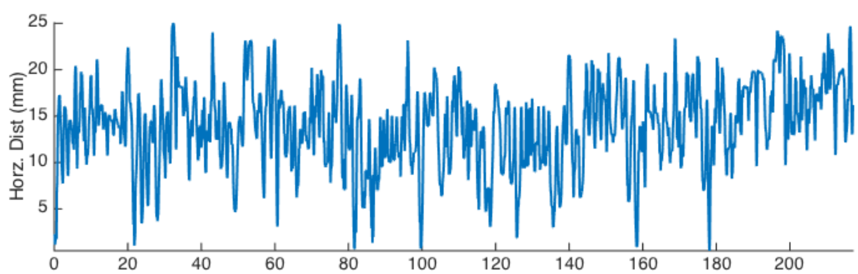
Adam Smoulder
BIOENG 1586
BCI Homework

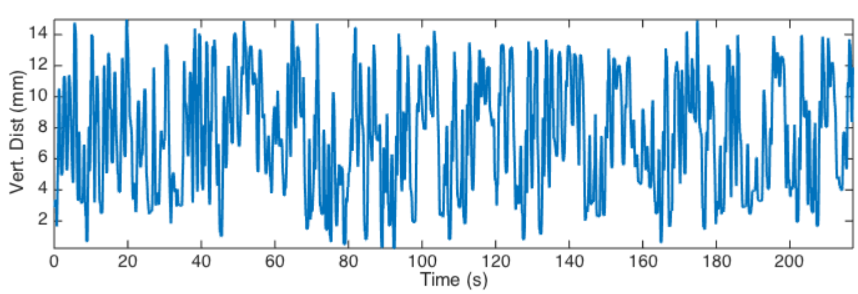
PART 1:

1. So I tried a few different ways. First, same x and y for position, but color is time:

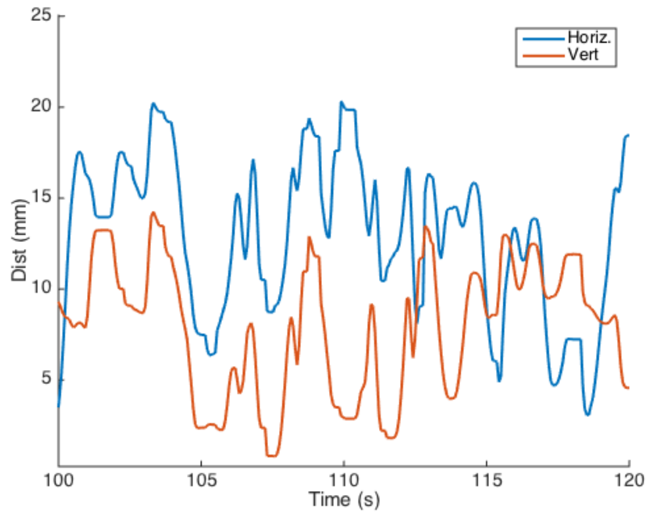


All I really get out of this is that it's ugly. Let's try doing the horiz and vert piece by piece:





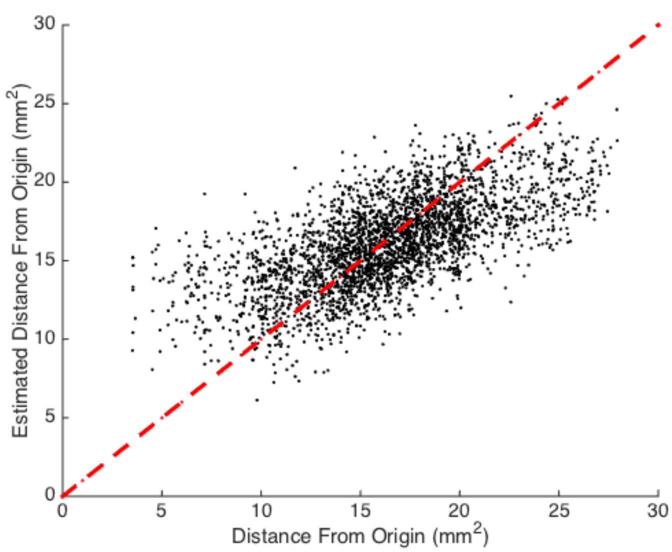
Still just seeing random crap. One last shot, let's overlay and zoom in on a smaller time period:



So there may be some relationship between when the monkey goes to higher x values that he goes towards higher y as well, though other times not so much…pretty rough to tell.

2. I did it!

3. (Thanks to Sam Waters for idea) By looking at raw distance from the origin for both the original and estimated kinematics data, we can have a single number per time-point to compare. If we plot the real data along the x-axis with the estimated data along the y-axis, we would expect a perfect filter (matching data) to be completely along the diagonal line (dashed red below):

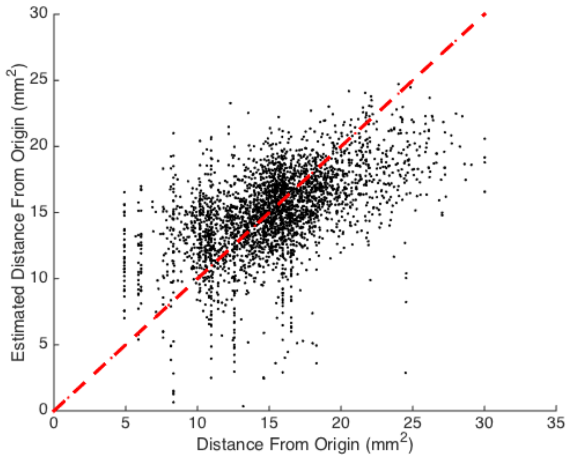


So looking at just components:

	MSEs	corrCoeffs
Total	10.708	0.65479
Horz	13.407	0.58828
Vert	3.801	0.8434

So we have a fair deal of error here (MSE), but our correlation coefficients are alright, especially the vertical prediction (r = 0.84).

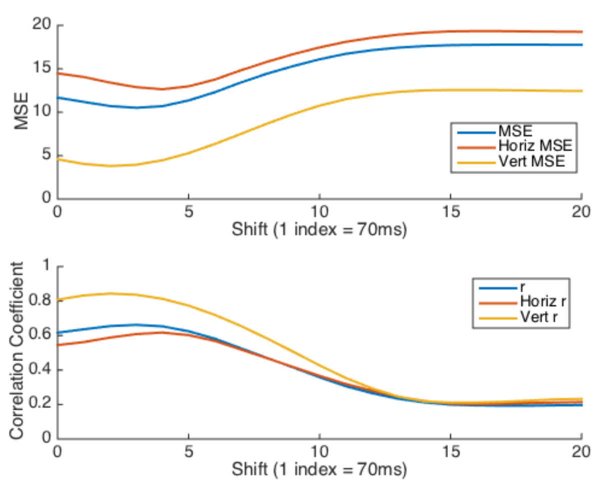
4. Same analyses as last time:



	MSE_Train	MSE_Test	r_Train	r_Test
Total	10.708	13.53	0.65479	0.53164
Horz	13.407	17.085	0.58828	0.45628
Vert	3.801	9.0582	0.8434	0.59995

So we're still kinda accurate, though we lost some. All correlation coefficients went down, and with that, all MSEs went up. Even with this, it looks like the predictions are still in the ballpark of being correctish. If our tolerance was low enough (we had wide enough spaced targets), we would probably get at least alright accuracy.

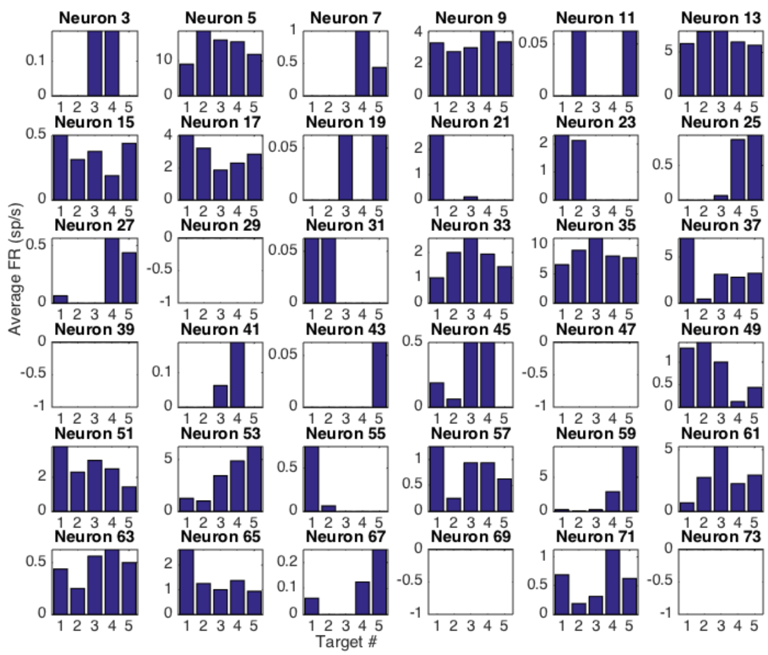
5. So I tried doing 0-20 shifts and calculated MSE and r for the total distance from origin, horizontal distance from origin, and vertical distance from origin:



Minimum MSE is located at lag of 4 points (1 pt = 70ms) for total distance, 5 for horizontal distance, and 3 for vertical distance. Greatest r-value is at the same lags for each type. I would say based on this that 280ms lag is the best (better than the 210ms proposed to test!).

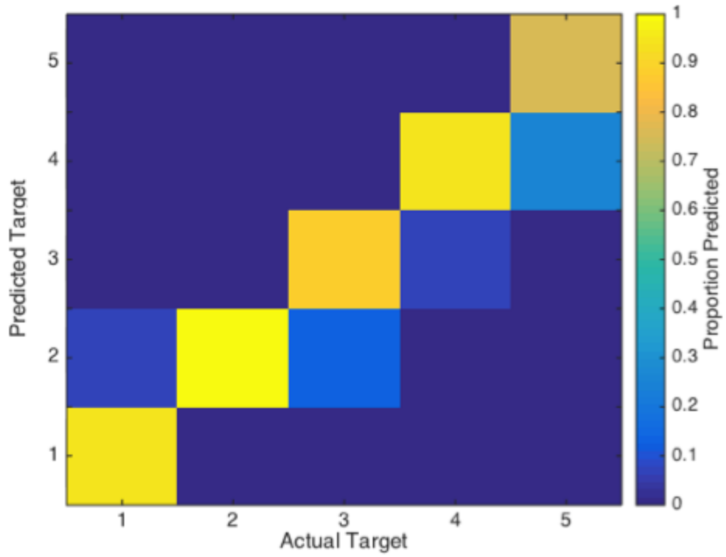
PART 2:

1. So looking at a buncha random neurons, we got:



So a lot of the neurons are tuned towards different directions. Some neurons also apparently have nothing… hm.

2-4. I had to add some small random number in so that the inactive neurons wouldn't have 0 variance, but the result was success on the held-out trial, then this overall:



The error term I added to all trials used for the training was:

```
(10^(-20))*randn(size(trainingData))
```

Overall accuracy rate was 90%! To improve it further, I'd bet more training data would be the best solution. You could also limit the number of targets further, though that limits functionality.