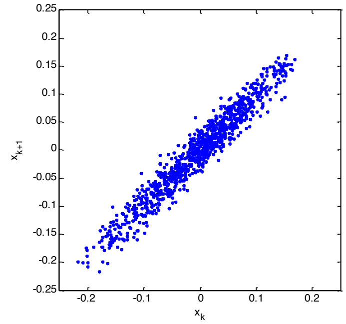
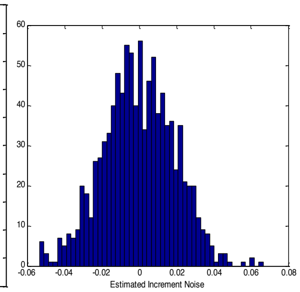
MA 568 – Statistical Analysis of Point Process Data

Solutions to Problem Set #4

```
% Question 1
load CricketData.mat
x = trainingStim(1:end-1);
y = trainingStim(2:end);
plot(x,y,'.');
A = x*y'/(x*x')
hist(y-A*x,50);
s = std(y-A*x)

>> A = 0.9708
>> s = 0.0191
```

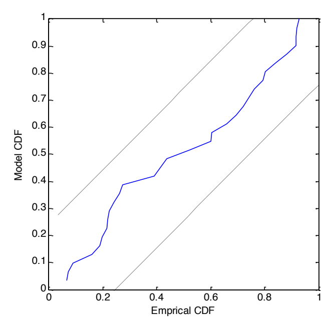





```
% Question 2
X = trainingStim;
for i = 1:10,
 Y = trainingSpikes(i,:);
 [b(i,:) dev(i) stats(i)] = glmfit(X,Y,'poisson');
 stats.p
end;
```

All of the parameters are significant at the 0.05 level.

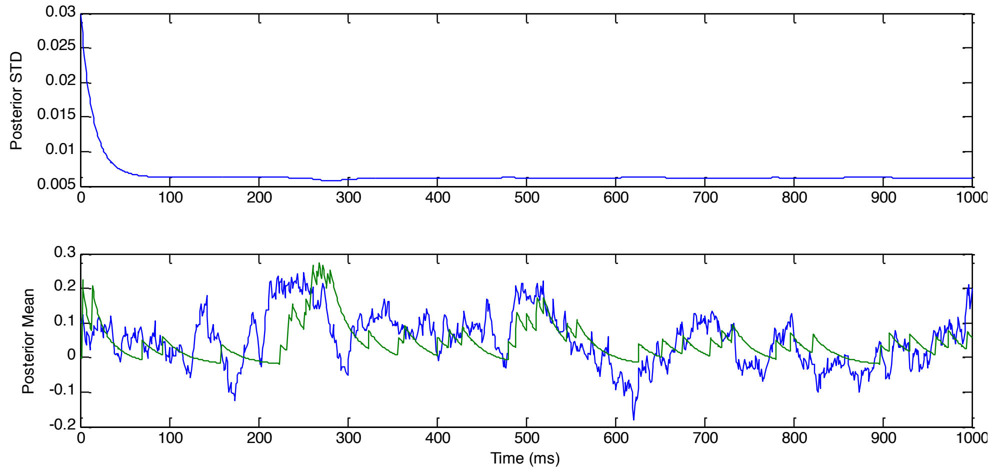
```
lambda_ML = exp([ones(size(X)); X]'*b(i,:)');
lambdaInt = cumsum(lambda_ML);
sInd = find(trainingSpikes(i,:));
Z = diff([0; lambdaInt(sInd)]);
U = expcdf(Z,1);
n = length(U);
M = [1:n]/n;
plot(sort(U),M,M,M+1.36/sqrt(n),'k:',M,M-
1.36/sqrt(n),'k:');
axis([0 1 0 1]);
```



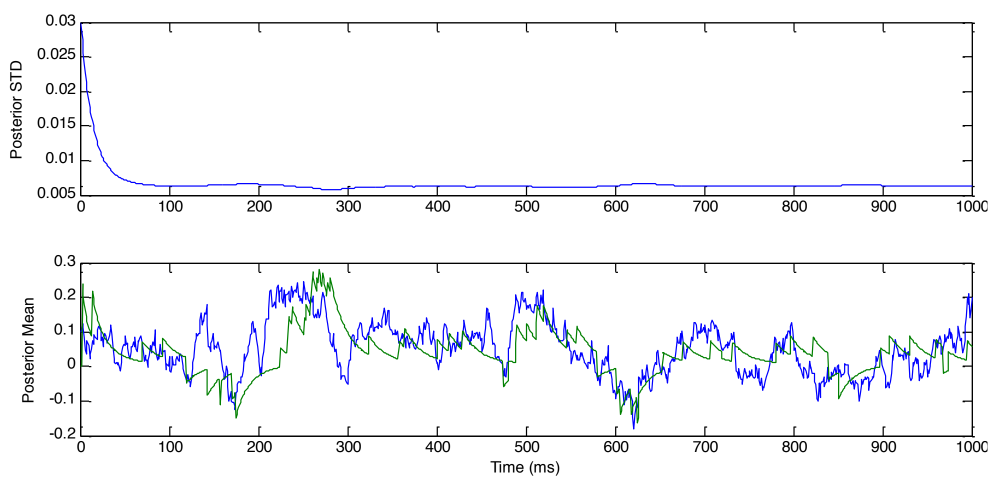
Given the relatively small amount of spiking activity available for each neuron, it is not possible to reject this simple exponential linear model.

```
% Question 3
mu(1) = 0;
sig(1) = .03;

% Decode with first neuron
for i = 2:1000,
 sig(i) = 1/(1/(A*A*sig(i-1)+s^2)+sum(b(1,2).^2.*exp(b(1,1)+b(1,2)*
   mu(i-1))));
 mu(i) = A*mu(i-1) + sig(i)*sum(b(1,2).*(testSpikes(1,i-1)-
   exp(b(1,1) +b(1,2)*mu(i-1))));
end;
```

```
plot(1:1000,sig);
plot(1:1000,testStim,1:1000,mu);
```

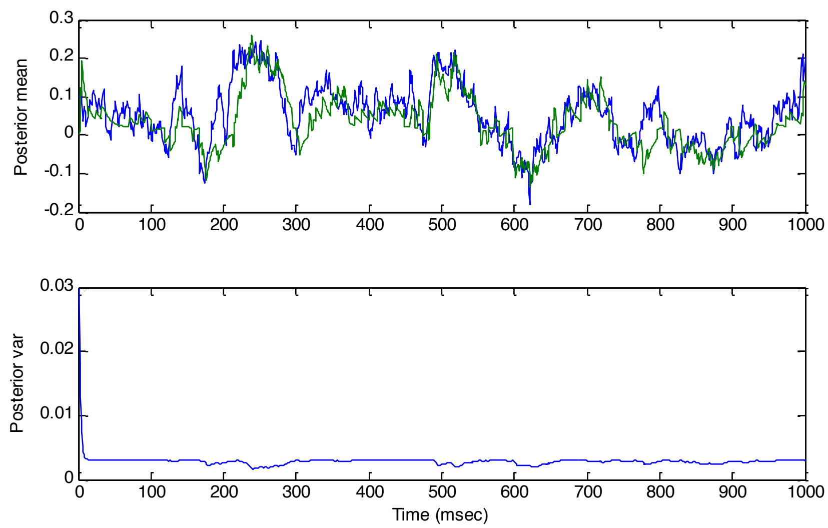



```
MSE = mean((mu-testStim).^2)

>> MSE = 0.0049

% Decode with first two neurons
for i = 2:1000,
 sig(i) = 1/(1/(A*A*sig(i-1)+s^2) + sum(b(1:2,2).*exp(b(1:2,1) +
   b(1:2,2)*mu(i-1))));
 mu(i) = A*mu(i-1) + sig(i)*sum( b(1:2,2).*(testSpikes(1:2,i-1)-
   exp(b(1:2,1)+b(1:2,2)*mu(i-1))) );
end;
plot(1:1000,sig);
plot(1:1000,testStim,1:1000,mu);
```




```
MSE = mean((mu-testStim).^2)

>> MSE = 0.0048
```

```
% Decode with all neurons
for i = 2:1000,
 sig(i) = 1/(1/(A*A*sig(i-1)+s^2) + sum(b(:,2).*exp(b(:,1) +
   b(:,2)*mu(i-1))));
 mu(i) = A*mu(i-1) + sig(i)*sum( b(:,2).*(testSpikes(:,i-1)-
   exp(b(:,1)+b(:,2)*mu(i-1))) );
end;
plot(1:1000,sig);
plot(1:1000,testStim,1:1000,mu);
```

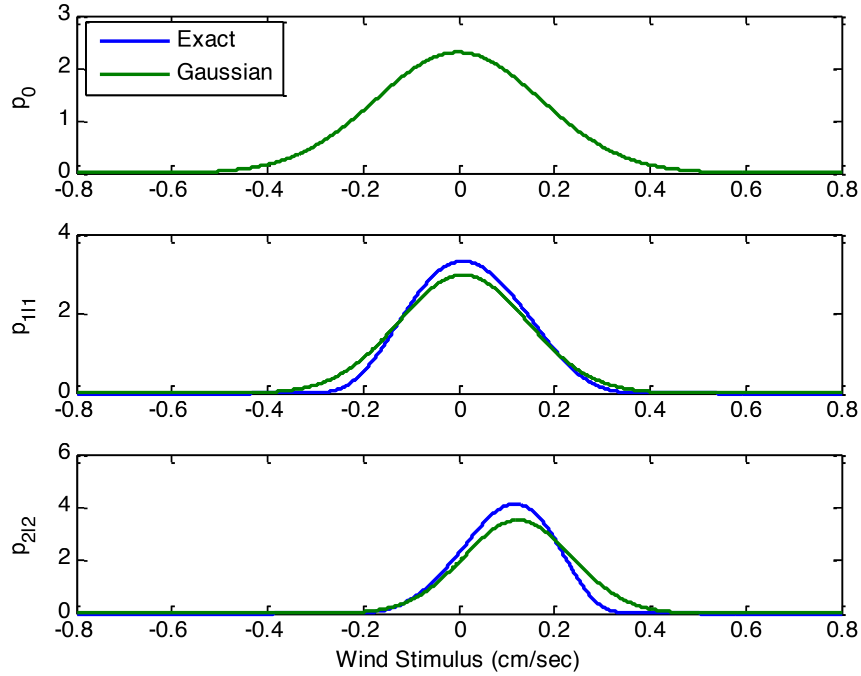



```
MSE = mean((mu-testStim).^2)

>> MSE = 0.0028
```

We are able to decode accurately with a single neuron. The mean squared error is 0.0058, which approximately corresponds to the asymptotic value of the posterior variance. The estimate uncertainty drops rapidly to this level. Adding more neurons decreases the estimation error and posterior variance to approximately 0.0028.

```
% Question 4
xrange = -1:.001:1;
p0 = normpdf(xrange,0,sqrt(.03));
p1_0 = normpdf(xrange,A*0,sqrt(A^2*.03+s^2));
plambda = exp(b(:,1)*ones(size(p1_0))+b(:,2)*xrange);
p1_1 = p1_0 .* prod(exp(diag(testSpikes(:,1))*log(plambda)-
plambda),1);
p1_1 = p1_1/sum(p1_1)*1000;

p1and2_1=exp(-(xrange'*ones(size(xrange))-A*ones(size(
xrange'))*xrange) .^2 /2/s^2) .* (ones(size(xrange'))*p1_1);
p2_1 = sum(p1and2_1,2)';
p2_2 = p2_1 .* prod(exp(diag(testSpikes(:,2))*log(plambda)-
plambda),1);
```

```
p2_2 = p2_2/sum(p2_2)*1000;

subplot(311); plot(xrange,p0);
subplot(312); plot(xrange,p1_1);
subplot(313); plot(xrange,p2_2);
```



The exact posterior distribution remains approximately Gaussian over the first two time steps. Continuing this analysis would show that the posterior distribution is approximately Gaussian at all times. Therefore the Gaussian approximation is appropriate and the approximate filter should be near optimal.