MAE288B Optimal Estimation - Homework 2, Winter 2022
— due Friday February 25, 2022.

I found a recent paper: P. Poggi, M. Muselli, G. Notton, C. Cristafari and A. Louche, "Forecasting and simulating wind speed in Corsica by using an autoregressive model," *Energy Conversion and Management,* vol. 44, pp. 3177-3196, 2003.doi:10.1016/S0196-8904(03)00108-0 This is available for download via UCSD Library once you log in.

The paper describes a model for the wind speed. This model captures the correlation over time of the wind. You are going to use such a model to describe the wind affecting the small cart that we have been using for Kalman filtering analysis in class.

Question 1 – Modeling the wind

On page 3183 in Table 2, the authors present a Gaussian model for the wind with mean and variance by month. Since we are in February, those values are mean = -0.038 ms^{-1} and standard deviation σ=1.016 ms^{-1}. The third value in the table is the correlation coefficient between the experimental and modeled distributions, which we shall not use. Note that they use three-hourly average wind speed measurement; good for wind farms but not for small carts. They also fit autoregressive (AR) models of various orders to the data from many years. MAE283A deals with some this. We shall fit a different model.

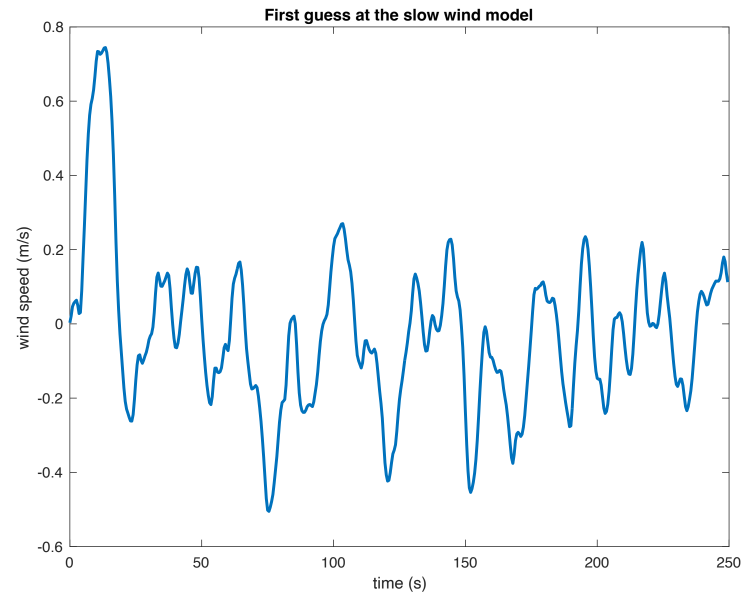
Our interest is in capturing the correlation between the wind data over time. Our description of the wind is that it comprises the following components. In addition to its mean value (which we are going to take as zero), the wind has a slowly varying value which varies between $\pm$5 ms^{-1} over about 10 seconds. You need to build a model to capture this description.

Slow component

I tried the following MATLAB commands.

```
>> [numb,denb]=butter(2,.05);
>> wi=filter(numb,denb,randn(500,1));
>> t=[0:499]'/2;
>> plot(t,wi,'LineWidth',2);shg
>> xlabel('time (s)');ylabel('wind speed (m/s)')
>> title('First guess at the slow wind model')
>> print -dpdf -bestfit firstGuess.pdf

>> std(wi)
    0.2163

>> mean(wi)
   -0.0238
```

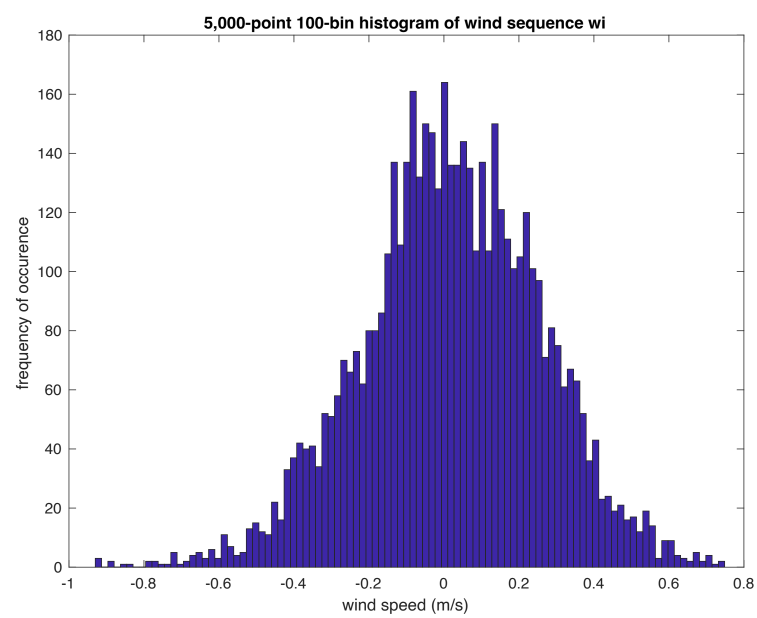
Here is what I saw.



You will note that the correlation properties are about correct, given my loose description. But the mean and variance need some adjusting.

Try the same example with, say, 5,000 data.

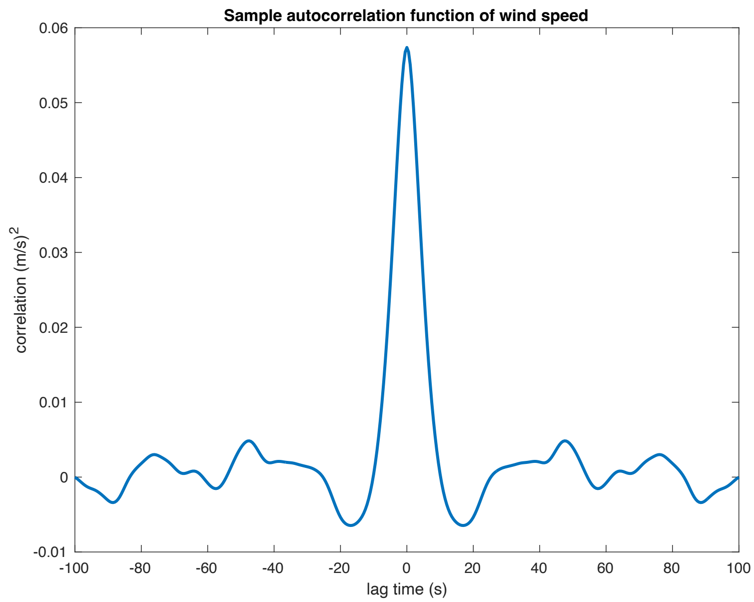
```
>> wi=filter(numb,denb,randn(5000,1));
>> t=[0:4999]'/2;
>> plot(t,wi,'LineWidth',2);shg
>> hist(wi,100);shg
>> std(wi)
    0.2388

>> mean(wi)
    0.0187
```



We see that the pdf is Gaussian with the mean and variance as computed. We cannot see the correlation from this plot though.

To look at the sample autocorrelation function try this.

```
>> plot([-200:200]/2,xcorr(wi,wi,200,'unbiased'),'Linewidth',2);shg
>> xlabel('lag time (s)');ylabel('correlation (m/s)^2')
>> title('Sample autocorrelation function of wind speed')
>> print -dpdf -bestfit firstXcorr.pdf
```



We see the strong autocorrelation about 15-20 seconds out but then dying away.

Part (i): Write down a (second order) state-space realization, $[A_s, B_s, C_s, D_s]$, of the Butterworth lowpass filter model. For your convenience, add a delay so that the state-space model has zero D_s-term. The delay does not matter since the model is being used to describe the disturbance to the cart. Alternatively, you could just use the MATLAB command `>> [As,Bs,Cs,Ds]=tf2ss(numb,denb)` and deal with the non-zero D_s.

Part (ii): Write down the Lyapunov equation for the steady-state covariance matrix of the state $x_{s,k}$ of your state equation

$$\begin{aligned} x_{s,k+1} &= A_s x_{s,k} + B_s w_{s,k}, \\ y_{s,k} &= C_s x_{s,k} + D_s v_{s,k}. \end{aligned}$$

Accordingly, determine the covariance of $y_{s,k}$

Part (iii): Using the state equation above and the whiteness of $\{w_{s,k}\}$, determine the correlation between $x_{s,k}$ and $x_{s,k-\tau}$ for some integer τ. Accordingly, write down the correlation between $y_{s,k}$ and $y_{s,k-\tau}$.

Part (iv): Choose a Q_s covariance for the process noise, $w_{s,k}$, driving your model which makes the standard deviation of the output $y_{s,k}$ 5 ms^{-1} when driven by white Gaussian $\mathcal{N}(0, Q_s)$.

Question 2 – Combining the disturbance/wind model and the cart model

The description of the cart used in class, had a white process noise affecting the cart. Replace this by the forcing driven by your wind model. You should arrive at a four-dimensional state vector with process noise affecting the input to both the disturbance model you derive in Question 1 and the cart itself. Although, feel free to take the cart's process noise variance small. Include white measurement noise on the cart's position as before.

Note, the model used in class has the wind's force as the disturbance not the wind's velocity. You will need to work out how to map velocity to force; it depends on the *windage* of your cart.

Write down this new combined linear system driven by wind and noise. Verify its observability. Explain what observability means in practical terms here.

In MATLAB (and one paper) code/write down the equations for the Kalman predictor of the four-dimensional wind-blown cart with its control input.

Use MATLAB to simulate some realistic wind data and the response of the cart to this wind. With this data, run your Kalman predictor and estimate the full four-dimensional state.

Question 3 – State-estimate feedback control

Your combined four-dimensional state system of wind and cart is obviously not reachable, because the cart input has no effect on the two wind states. Nevertheless, we can still profit from linear state variable feedback, because the full system is stabilizable.

Part (i): Use the MATLAB `dlqr` command to determine a feedback gain which minimizes the square of the cart position while not using excessive control signal. This will require choosing Q_c and R_c matrices for the control design.

Part (ii): Construct the linear state-estimate feedback controller from the Kalman predictor and the LSVF gains and your fourth-order model.

Part (iii): Simulate the closed-loop system to show that it works. Spend some time with showing the system working.

Part(iv): Fiddle with the parameters to tune the controller to your liking.

Part (v): Pat yourself on the back. Open a beer.