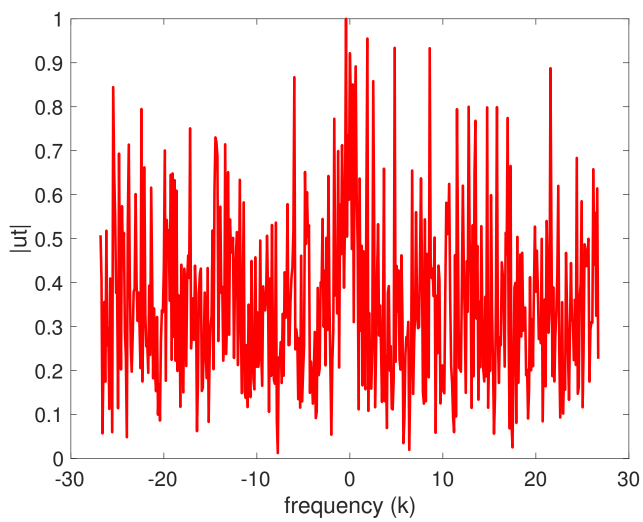
Lecture 3
Radar Detection and Averaging

Jason J. Bramburger

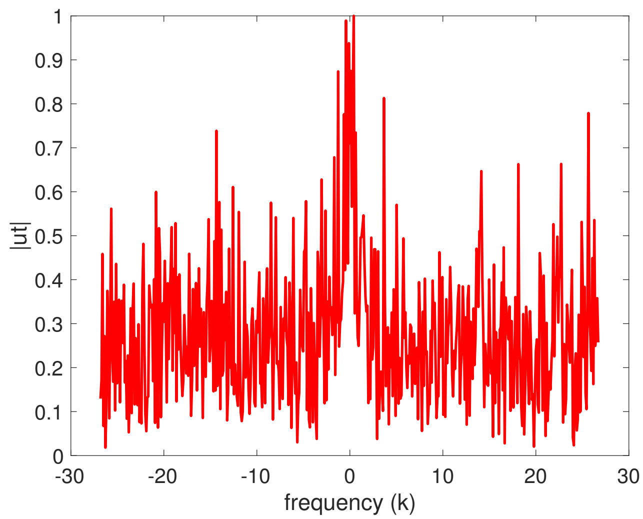
In the previous lecture we saw that filtering is an effective tool for reducing noise. Although, it fails to take advantage of two keys facts: the noise is white, and the radar will continue to produce more signal data. Recall from the last lecture that white noise was modelled by adding a normally distributed random variable with zero mean and unit variance to each Fourier component. The key here is *with zero mean*. Therefore, if you average over many realizations (in frequency space), the noise from each realization will cancel out and you will get something close to the true signal. Let's start by generating a noisy signal.

```
L = 30; % timeslot [-L,L]
n = 512; % number of Fourier modes

t2 = linspace(-L,L,n+1);
t = t2(1:n);
k = (2*pi/(2*L))*[0:(n/2-1) -n/2:-1]; %define this so we don't need to use fftshift(k) every ...
    time we plot
ks = fftshift(k);
u = sech(t);
ut = fft(u);

% Add noise
noise = 10;
utn = ut + noise*(normrnd(0,1,1,n) + 1i*normrnd(0,1,1,n));

% Plot noisy signal
figure(1)
plot(ks,fftshift(abs(utn))/max(abs(utn)),'r','Linewidth',2)
set(gca,'Fontsize',16)
xlabel('frequency (k)')
ylabel('|ut|')
```



Can you see the signal in this? Especially if you didn't know for sure that there *is* a signal in here? Let's average over two realizations to try and recover the signal.

```
ave = zeros(1,n);
for j = 1:2
   utn = ut + noise*(normrnd(0,1,1,n) + 1i*normrnd(0,1,1,n));
   ave = ave + utn;
end
ave = abs(fftshift(ave))/2;

% Plot averaged signal
figure(2)
plot(ks,ave/max(ave),'r','Linewidth',2)
set(gca,'Fontsize',16)
xlabel('frequency (k)')
ylabel('|ut|')
```

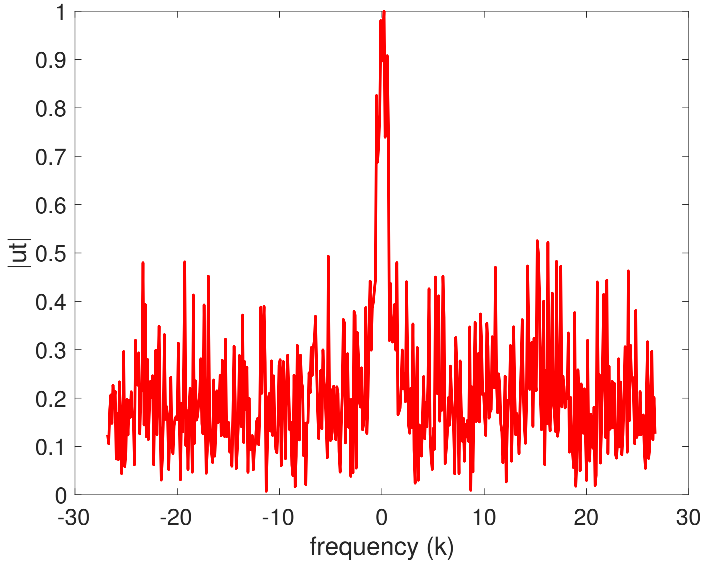


It's looking better already! Particularly, you can see the spike near $k = 0$. Let's try averaging over 5 realizations now.

```
ave = zeros(1,n);
for j = 1:5
   utn = ut + noise*(normrnd(0,1,1,n) + 1i*normrnd(0,1,1,n));
   ave = ave + utn;
end
ave = abs(fftshift(ave))/5;

% Plot averaged signal
figure(3)
plot(ks,ave/max(ave),'r','Linewidth',2)
set(gca,'Fontsize',16)
xlabel('frequency (k)')
ylabel('|ut|')
```

Things are looking good now! Let's compare over different numbers of realizations.

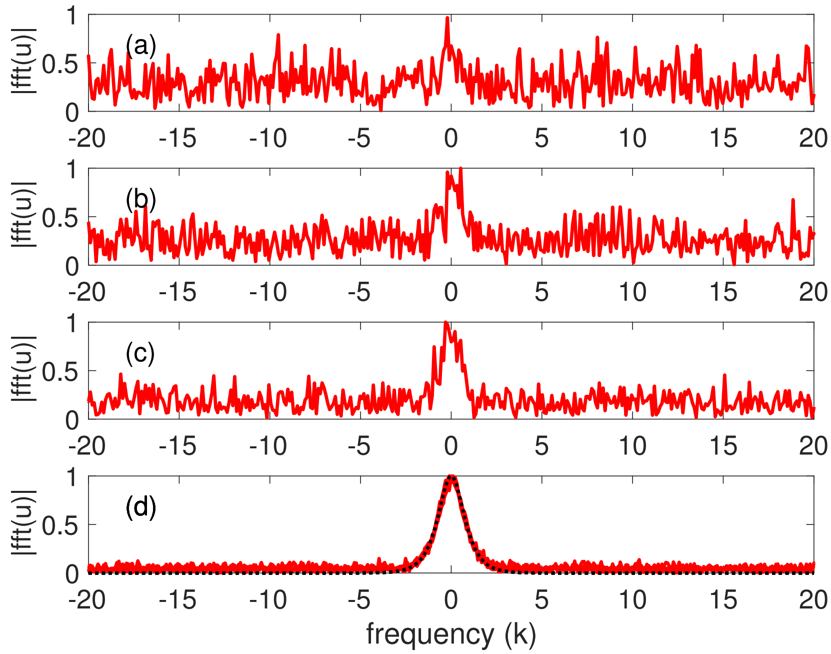
```
labels = ['(a)';'(b)';'(c)';'(d)'];
realize = [1 2 5 100];

figure(4)
for jj = 1:length(realize)
   u = sech(t);
   ave = zeros(1,n);
   ut = fft(u);

```



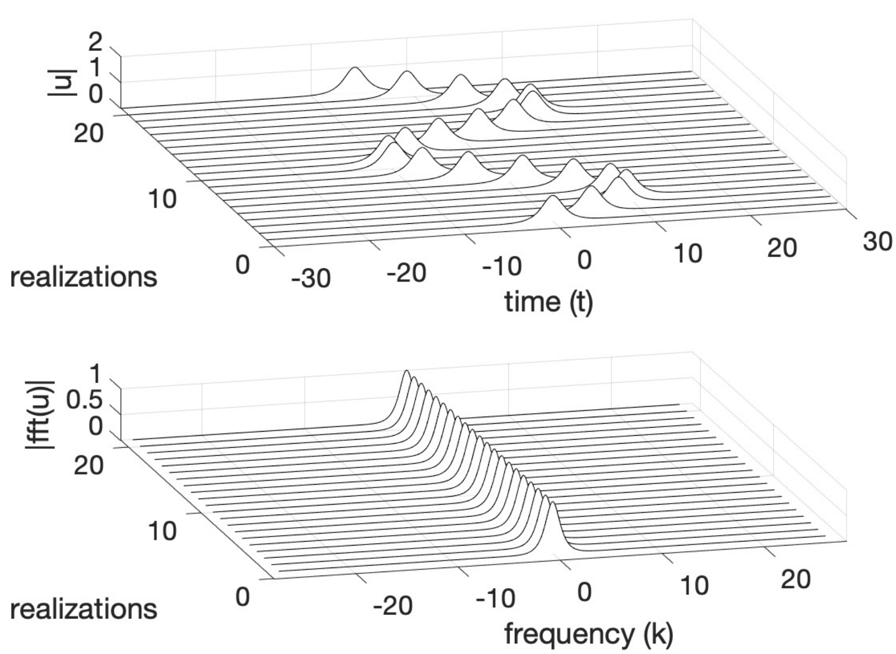

```
    % Averaging over realizations
    for j = 1:realize(jj)
       utn(j,:) = ut + noise*(normrnd(0,1,1,n) + 1i*normrnd(0,1,1,n));
       ave = ave+utn(j,:);
    end
    ave = abs(fftshift(ave))/realize(jj);

    % Plot averaged signals in one plot
    subplot(length(realize),1,jj)
    plot(ks,ave/max(ave),'r','Linewidth',2)
    set(gca,'Fontsize',16)
    axis([-20 20 0 1])
    text(-18,0.7,labels(jj,:),'Fontsize',16)
    ylabel('|fft(u)|','Fontsize',16)
end

hold on
plot(ks,abs(fftshift(ut))/max(abs(ut)),'k:','Linewidth',2)
xlabel('frequency (k)')
```

As you have probably guessed by now, the more realizations the better. With 100 realizations we get something that is very close to the idealized signal. Note that so far we have only taken the exact signal and added white noise to it on each realization. If we had radar signals measured over time, the aircraft would be moving and the signal would be at different points in time. This means that the incoming signal would maintain the same shape in the time domain, but the location of the spike will change. It turns out that this won't matter since the same signal shifted in the time domain will look the same in the frequency domain (if we are considering $|\hat{u}|$).

```
slice = 0:0.5:10;
[T,S] = meshgrid(t,slice);
[K,S] = meshgrid(k,slice);

U = sech(T - 10*sin(S)).*exp(1i*0*T);
figure(5)
subplot(2,1,1)
waterfall(T,2*S+1,U), colormap([0 0 0]), view(-15,70)
set(gca,'Fontsize',16,'Xlim',[-30 30],'Zlim',[0 2])
xlabel('time (t)'), ylabel('realizations'), zlabel('|u|')

for j = 1:length(slice)
   Ut(j,:) = fft(U(j,:));
   Kp(j,:) = fftshift(K(j,:));
```

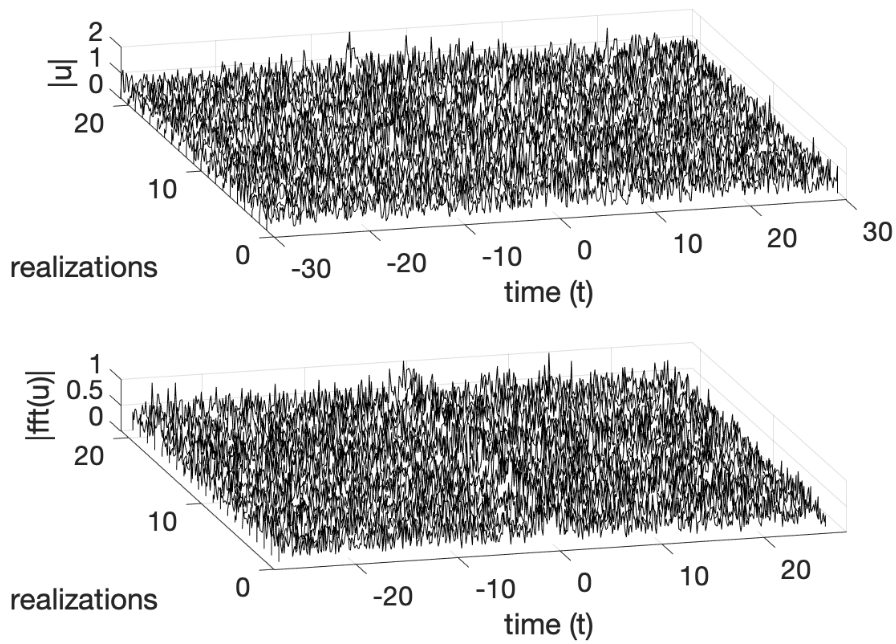



```
    Utp(j,:) = fftshift(Ut(j,:));
    Utn(j,:) = Ut(j,:) + + noise*(normrnd(0,1,1,n) + 1i*normrnd(0,1,1,n));
    Utnp(j,:) = fftshift(Utn(j,:))/max(abs(Utn(j,:)));
    Un(j,:) = ifft(Utn(j,:));
end

subplot(2,1,2)
waterfall(Kp,2*S+1,abs(Utp)/max(abs(Utp(1,:)))), colormap([0 0 0]), view(-15,70)
set(gca,'Fontsize',16,'Xlim',[-28 28])
xlabel('frequency (k)'), ylabel('realizations'), zlabel('|fft(u)|')
```



This should emphasize to you the importance that we averaged in the frequency domain instead of the time domain. If you averaged in the time, we wouldn't get anything useful. But in the frequency domain, we will be averaging the same frequency signature with noise added on top. To illustrate the difference, we can add noise to each of those realizations.

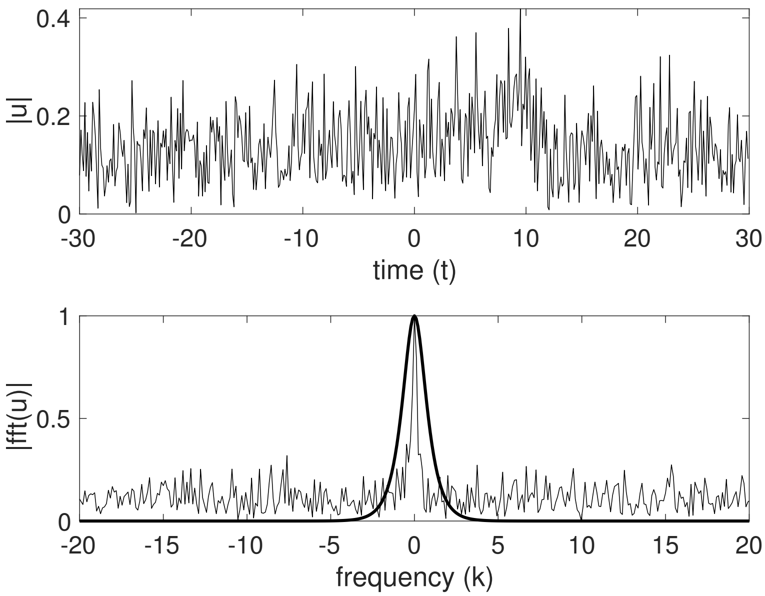
```
figure(6)
subplot(2,1,1)
waterfall(T,2*S+1,abs(Un)), colormap([0 0 0]), view(-15,70)
set(gca,'Fontsize',16,'Xlim',[-30 30],'Zlim',[0 2])
xlabel('time (t)'), ylabel('realizations'), zlabel('|u|')

subplot(2,1,2)
waterfall(Kp,2*S+1,abs(Utnp)/max(abs(Utnp(1,:)))), colormap([0 0 0]), view(-15,70)
set(gca,'Fontsize',16,'Xlim',[-28 28])
xlabel('time (t)'), ylabel('realizations'), zlabel('|fft(u)|')
```



Now, let's try to average over the time domain and the frequency domain to compare the difference.

```
Uave = zeros(1,n);
Utave = zeros(1,n);
for j = 1:length(slice)
   Uave = Uave + Un(j,:);
   Utave = Utave + Utn(j,:);
end
Uave = Uave/length(slice);
Utave = fftshift(Utave)/length(slice);

figure(7)
subplot(2,1,1)
plot(t,abs(Uave),'k')
set(gca,'Fontsize',16)
xlabel('time (t)'), ylabel('|u|')

subplot(2,1,2)
plot(t,abs(Utave)/max(abs(Utave)),'k')
hold on
plot(ks,abs(fftshift(Ut(1,:))/max(abs(Ut(1,:)))),'k','Linewidth',2)
axis([-20 20 0 1])
set(gca,'Fontsize',16)
```

```
xlabel('frequency (k)'), ylabel('|fft(u)|')
```



At the top we have the result of averaging in the time domain, while the bottom show the result of averaging in the frequency domain (as well as the idealized signal). The drawback is that through averaging we can get a good picture of the signal in the frequency domain, we lose all information about *when* the signal occurs in the time domain. This is a weakness of the Fourier transform. In the coming sections we will see some ways to get information about time.