MICHALE FEE: OK, good morning, everyone. So today, we're going to continue with our plan for developing a powerful set of tools for analyzing the temporal structure of signals, in particular a periodic structure and signals. And so this was the outline that we had for this series of three lectures.

Last time, we covered Fourier series, complex Fourier series, and the Fourier transform, discrete Fourier transform. And we started talking about the power spectrum. And in that section, we described how you can take any function and write it as a-- or any periodic function and write it as a sum of sinusoidal component. So even functions we can write down as a sum of cosines. And odd functions we can write down as a sum of sines.

Today, we're going to continue with that. We're going to talk about the convolution theorem, noise and filtering Shannon-Nyquist sampling theorem and spectral estimation. And next time, we're going to move on to spectrograms and an important idea of windowing and tapering, time bandwidth product, and some more advanced filtering methods.

So last time, I gave you this little piece of code that allows you to compute the discrete Fourier transform using this Matlab function FFT. And we talked about how in order to do this properly, you should first circularly shift. You have to take the time series. And actually, the FFT algorithm is expecting the first half of the data in the second half of that data vector. Don't ask me why. But we just do a circular shift, run the FFT. And then circular shift again to get the negative frequencies in the first half of the vector. And then you can plot the Fourier transform of your function.

This shows an example where we took a cosine as a function of time. At some frequency, here, 20 hertz. We compute the Fourier transform of that and plot that. So that's what this looks like.

Here is a cosine at 20 hertz. And you can see if you take the fast Fourier transform of that, you can see that what you see is the real part as a function of frequency. It has two peaks, one at plus 20 hertz, one at minus 20 hertz. And the imaginary part

is 0. So we have two peaks. One produces a complex exponential that goes around the unit circle like this at 20 hertz. The other peak produces a complex exponential that goes around the other way at 20 hertz. The imaginary parts cancel and leave you with a real part that goes back and forth at 20 hertz OK? So that's what those two peaks are doing.

Here is the Fourier transform of a sine wave at 20 hertz. This is phase shifted so the cosine is a symmetric function or an even function. The sine is an odd function. And you can see that in this case, the Fourier transform again has two peaks. In this case, the real part is 0. And the two peaks are in the imaginary part. The one at plus 20 hertz is minus I. And the one at minus 20 hertz is plus I.

Now, the interesting thing here, the key thing, is that when you take the Fourier transform of a function, symmetric functions, even functions, are always real. The Fourier transform of even functions is always real. The Fourier transform, if you will, the Fourier series of the even part of the function into the real part of the Fourier transform and the Fourier series of the odd part of the function into the imaginary part of the Fourier transform. OK?

Now, we introduced the idea of a power spectrum, where we just take the Fourier transform. And we take the square magnitude of that Fourier transform. And you can see that the power spectrum of the sine and cosine function is just a single peak at the frequency of the sine or cosine. OK? And you can see why that is because the sine and cosine have a peak at plus 20 hertz. For the cosine, it's real. And for the sine, it's imaginary. But the square magnitude of both of those is 1 at that frequency.

OK, any questions about that? Feel like I didn't say that quite as clearly as I could have? OK. So any questions about this?

OK, let's take a look at another function that we've been talking about, a square wave. In this case, you can see that the square wave is symmetric or even. And you can see that the Fourier transform of that is all real. The peaks are in the real part of the Fourier transform. You can see the imaginary part in red is 0 everywhere. And you can see that the Fourier transform of this has multiple peaks at intervals that are again equal to the frequency of this square wave. OK?

If you look at the power spectrum of the square wave, you can see again it's got multiple peaks at regular intervals. One thing that you often find when you look at power spectra of functions is that some of the peaks are very low amplitude or very low power. So one of the things that we often do when we're plotting power spectra is to plot not power here but log power. OK? And so we plot power in log base 10.

A difference of an order of magnitude in two peaks corresponds to a unit called a bel, b-e-l. So 1 bel corresponds to a factor of 10 difference in power. So you can see this peak here is about 1 bel lower than that peak, right? And more commonly used unit is called decibels, which are 10 decibels per bel. So decibels are given by 10 times the log base 10 of the power of the square magnitude of the Fourier transform. Does that make sense. Yes.

AUDIENCE: So [INAUDIBLE] square magnitude [INAUDIBLE] so just like [INAUDIBLE]

MICHALE FEE: No, so you take this square magnitude because-- OK, remember, last time we talked about power. So if you have an electrical signal, the power in the signal would be voltage squared over resistance. Power, when you refer to signals, is often kind of used synonymously with variance. And variance is also goes as the square of the signal.

Now, because the Fourier transform is a complex number, what we do is we don't just square it, but we take the squared magnitude. So we're measuring the distance from the origin in the complex plane. OK? Good question.

All right, any questions about this and what the meaning of decibels is? So if a signal had 10 times as much amplitude, the power would be how much larger? If you had 10 times as much amplitude, how much increased power would there be? 100 times. Which is how many bels? Log base 10 of 100 is 2. How many decibels?

AUDIENCE: 22?

MICHALE FEE: No, it's 10 decibels per bel. Deci just means a tenth of, right? Remember those units? So a factor of 10 in signal is a factor of 100 in power, which is 2 bels, which is 20 decibels. OK.

All right, now I just want to show you one important thing about Fourier transforms.

There's an interesting property about scaling in time and frequency. So if you have a signal like this that's periodic at about-- I don't know, it looks like-- OK, there it is-- about 5 hertz. If you look at the Fourier transform of that, you can see a series of peaks, because it's a periodic signal.

Now, if you take that same function and you make it go faster-- so now, it's at about 10 hertz, instead of 5 hertz, you can see that the Fourier transform is exactly the same. It's just scaled out. So the faster something moves in time, the more stretched out the frequencies are. Does that makes sense.

So if I show you any periodic function at one frequency and I show you the Fourier transform of it, you can immediately write down the Fourier transform of any scaled version of that function, because if this goes faster, if this same function but at a higher frequency, you can write down the Fourier transform just by taking this Fourier transform and stretching it out by that same factor. OK?

All right, so that was just a brief review of what we covered last time. And here's what we're going to cover today in a little more detail. So we're going to talk some more about the idea of Fourier transform pairs. These are functions where you have a function. You take the Fourier transform of it. You get a different function. If you take the Fourier transform of that, you go back to the original function. OK, so there are pairs of functions that are essentially for transforms of each other. OK? An example of that you saw here. A square wave like this has a Fourier transform that's this funny function a set of peaks. If you take the four transform of that, you would get this square wave. OK?

We're going to talk about the convolution theorem, which is a really cool theorem. Convolution in the time domain looks like multiplication in the frequency domain. Multiplication in the time domain looks like convolution in the frequency domain. And it allows you to take a set of Fourier transform pairs that you know, that we'll learn, and figure out what the Fourier transform is of any function that's either a product of those or a convolution of those kind of base functions. It's a very powerful theorem.

We're going to talk about the Fourier transform a Gaussian noise and this power spectrum of Gaussian noise. We'll talk about how to do spectral estimation. And

we'll end up on the Shannon-Nyquist theorem and zero padding. And there may be, if there's time at the end, I'll talk about a little trick for removing the line noise from signals.

OK, so let's start with Fourier transform pairs. So one of the most important functions to know the Fourier transform of is a square pulse like this. So let's just take a function of time. It's 0 everywhere, 0 everywhere. But it's 1 if the time is within the interval plus delta t over 2 to minus delta t over 2, OK, so a square pulse like that. And let's just take the case where delta t is 100 milliseconds.

The Fourier transform of a square pulse is a function called the sinc function. It looks a little bit messy. But it's basically a sine wave that is weighted so that it's big in the middle and it decreases as you move away from 0. And it decreases as 1/f.

So this is frequency along this axis. It's just imagine that you have a sine wave that gets smaller as you go away from the origin by an amount 1 over f. That's all it is.

Now, really important concept here, remember, we talked about how you can take a function of time. So once you know that the Fourier transform of a square wave, of this square wave of with 100 milliseconds, is a sinc function. You know what the Fourier transform is of a square pulse that's longer. Right? What is it? Remember, if you just take a function in time and you stretch it out, the Fourier transform just does what? It compresses. It shrinks. And if you take this pulse and you make it narrower in time, then the Fourier transform just stretches out.

So if we take that pulse and we make it narrower, 25 milliseconds, then you can see that the sinc function, it's the same sinc function, but it's just stretched out in the frequency domain. So you can see that here if it's 10 milliseconds, the width of this is 12 hertz. The full width at half max of that peak is 12 hertz. If this is 4 times narrower, then this width will be 4 times wider.

What happens if we have a pulse here that is 500 milliseconds longer? So 5 times longer, what's the width of the sinc function here going to be? It'll be five times narrower than this, so a little over 2 hertz. OK? Does that makes sense. So you should remember this Fourier transform pair, a square pulse and a sinc function.

And there's a very important concept called the time bandwidth product. You can

see that as you make the width in time narrower, the bandwidth in frequency gets bigger. And as you make the pulse in time longer, the bandwidth gets smaller. And it turns out that the product of the width in time and the width in frequency is just a constant. And for this square pulse sinc function, that constant is 1.2. So there's a limit. If you make the square pulse smaller, the sinc function gets broader.

All right, let's look at a different Fourier transform pair. It turns out that the Fourier transform of a Gaussian is just a Gaussian. So, here, this Gaussian pulse is 50 milliseconds long. The Fourier transform of that is a Gaussian pulse that's 20 hertz wide. If I make that Gaussian pulses in time narrower, then the Gaussian in frequency gets wider. And inversely, if I make the pulse in time wider, then the Gaussian in frequency space gets narrower. Yes.

AUDIENCE: I just have a question [INAUDIBLE]

MICHALE FEE: Yes.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yep. So I'm trying-- it's a little bit unclear here, but I'm measuring these widths at the half height.

OK, and so you can see that for a Gaussian, this time bandwidth product, delta f times delta t is just 1. So there's a time bandwidth product of 1.

Who here has taken any quantum mechanics? Who here has heard of the Heisenberg uncertainty principle? Yeah. This is just the Heisenberg uncertainty principle. This is where the Heisenberg uncertainty principle comes from, because wave functions are just-- you can think of wave functions as just functions in time. OK? So the spatial localization of an object is some-- the wave function is just some position of space. And the momentum of that particle can be computed as the Fourier transform of the wave function.

And so if the particle is more lo-- [AUDIO OUT] in space, then if you compute the Fourier transform of that wave function, it's more dispersed in momentum. OK, so the uncertainty in momentum is larger. So this concept of time bandwidth product in the physical world is what gives us the Heisenberg uncertainty principle. It's very cool.

Actually, before I go on, you can see that in this case, the Fourier transform of this function is the same function. Does anyone remember what a [AUDIO OUT] function is that the Fourier transform of it is another version of that same function? We talked about it last time.

AUDIENCE: Pulse train.

MICHALE FEE: Pulse train, that's right. So a train of delta functions has a Fourier transform that's just a train of delta functions. And 1 over spacing in the time domain is just equal to 1 over the spacing in the frequency domain. So that's another Fourier transform pair that you should remember. All right?

OK, convolution theorem. Imagine that we have three functions of time, y of t, like this one, y of t. We could calculate the Fourier transform of that. And that's capital Y of omega. And then we have some other function, x of t, And its Fourier transform, X of omega, and another function g of tau and its Fourier transform, capital G of omega.

So remember, we can write down the convolution of this time series, x with this kernel g as follows. So in this case, we're defining y as the convolution of g with x. So y of t equals this integral d tau g of tau x of t minus tau, integrating over all tau. So that's a convolution.

The convolution theorem tells us that the Fourier transform of y is just the product of the Fourier transform of g and the Fourier transform of x. So that, you should remember. All right?

I'm going to walk you through how you derive that. I don't expect you to be able to derive it. But the derivation is kind of cute, and I enjoyed it. So I thought I'd show you how it goes.

So here's the definition of the convolution. What we're going to do is we're going to just take the Fourier transform of y. So here's how you calculate Fourier transform of something. Capital Y of omega is just the integral over all time dt y of t e to the minus i omega t. So we're going to substitute this into here.

Now, you can see that-- OK, so the first [AUDIO OUT] is actually reverse the order of

integration. We're going to integrate over t first rather than tau. Then what we're going to do is we can move the g outside the integral over t, because it's just a function of tau. So we pull that out.

So now, we have an integral dt x of t minus tau e to the minus i omega t. And what we can do is do a little modification here. We're going to pull an e to the minus i omega tau out of here. So we have e to the minus i omega tau times e to the minus j omega t minus tau. So you can say if you multiply those two things together, you just get back to that.

Now, what we're going to do is because this is we're integrating over t, but [AUDIO OUT] a function of tau, we can pull that out of the integral. And now we have integral dt x t minus tau e to the minus i omega t minus tau. And what do you think that is? What is that? What would it be if there were no tau there? If you just cross that out and that, what would that be? Anybody know? What that's?

AUDIENCE: The Fourier transform.

MICHALE FEE: This is just the Fourier transform of x. Right? And we're integrating from minus infinity to infinity. So it does it matter if we're shifting the inside by tau? No, it doesn't change the answer. We're integrating from minus infinity to infinity. So shifting the inside of it by a small amount tau isn't going to do anything.

So that's just the Fourier transform of x. Good? And what is this? Fourier transform of g. So the Fourier transform of y is just Fourier transform of g times the Fourier transform of x. All right, that's pretty cool. Kind of cute. But it's also really powerful. OK, so let me show you what you can do with that.

First, let me just point out one thing that there is a convolution theorem that relates convolution in the time domain to multiplication in the frequency domain. You can do exactly the same derivation and show that convolution of two functions in the frequency domain is the same as multiplication in the time domain. So that's the convolution theorem.

So let's see why that's so powerful. So I just showed you the Fourier transform of a Gaussian. So what we're going to do is we're going to calculate transform of a Gaussian times a sine wave. So if you take a Gaussian, some window centered

around 0 in time-- this is a function of time now right? So there's a little Gaussian pulse in time. We're going to multiply that by this sine wave. And when you multiply those together, you get this little pulse of sine. OK? [WHISTLES] Sorry, constant frequency. Boy, that's harder to do than I thought. [WHISTLES] OK, just a little pulse of sine wave.

So what's the forehead transform of that? Well, we don't know, right? We didn't calculate it. But you can actually just figure it out very simply, because you know the Fourier transform of a Gaussian. What is that? A Gaussian.

You know the Fourier transform of a sine wave. What is that? Yeah. Hold it up for me. What does it look like? Yes, sine wave, thank you, like this. OK.

And so what do you know-- what can you tell me right away what the Fourier transform of this is? You take the Fourier transform of that and can involve it with the Fourier transform of that. So let's do that. So there's the Fourier transform of this Gaussian. If this is 200 milliseconds wide, then how wide is this?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: It's 1 over 200 milliseconds, which is what? 5 hertz, right? 1 over 0.2 is 5.

The Fourier transform of this sine wave-- and I think I made it a cosine instead of a sine. Sorry, that's why I was going like this, and you were going like this. So I made it a cosine function.

The Fourier transform of the cosine function has two peaks. This is 20 hertz. So one peak at 20, one at minus 20. And the Fourier transform of this is just the convolution of this with that. You take this Gaussian and you slide it over those two peaks. You essentially smooth this with that. Does that makes sense?

Cool. So you didn't actually have to stick this into Matlab and compute the Fourier transform of that. You can just know in your head that that's the product of a Gaussian and a sine wave, or cosine. And therefore, the Fourier transform of that is the convolution of a Gaussian with these two peaks.

And there are many, many examples of interesting and useful functions in the time domain that you can intuitively understand what their Fourier transform is just by

having this idea. It's very powerful.

Here's another example. We're going to calculate the Fourier transform of this square windowed cosine function. So it's a product of the square pulse with this cosine to give this. So what is the Fourier transform of this. What is that?

So what is the Fourier transform of that?

AUDIENCE: The sinc function.

MICHALE FEE: It's the sync function. It's that kind of wiggly, peaky thing. And the Fourier transform of that is just two peaks. And so the Fourier transform of this is just like two peaks-- yeah-- with wiggly stuff around them. That's exactly right.

All right, any questions about that? OK.

All right, change of topic, let's talk about Gaussian noise. The Fourier transform of noise and the power spectrum of noise. And we're going to eventually bring all these things back together. OK?

All right, so what is Gaussian noise? So first of all, Gaussian noise is a signal in which the value at each time is randomly sampled from a Gaussian distribution. So you can do that in Matlab. That's a very simple function. This returns a vector of length N, sampled from a normal distribution, with variance 1.

So here's what that sounds like. [STATIC NOISE] Sounds noisy, right? OK, I just wanted to show you what the autocorrelation function of this looks like, which I think we saw before. So if you look at the distribution of all the samples, it just gives you a distribution that it has the shape of a Gaussian. And the standard deviation of that Gaussian is 1.

Now, what if you plot the correlation between the value of value of this function at time t and time t plus 1? Is there any relation between the value of this function at any time t and another time t plus 1? So they're completely uncorrelated with each other. The value of time t is uncorrelated with the value of time t plus 1. So there's zero correlations between neighboring samples.

What about the correlation of the signal at time t with the signal at time t? Well,

that's perfectly correlated, obviously. So we can plot the correlation of this function with itself at different time lags. Remember, that was the autocorrelation function. And if we do that, you get a 1 at 0 lag and 0 at any other lag. So that's the autocorrelation function of Gaussian noise.

All right, now, the Fourier transform of Gaussian noise is just Gaussian noise. It's another kind of interesting Fourier transform pair. And it's a Gaussian distribution. It's a Gaussian random distribution in both the real and the imaginary part. So you can see that the blue and red-- the red here is the imaginary part-- are both just Gaussian noise. OK?

All right, now what is the power spectrum? So we can take this thing-- and, remember, when we plot the power [AUDIO OUT] just plot the square magnitude of just the positive frequencies. Why is that again? Why do we only have to plot the square magnitude of the positive frequencies?

AUDIENCE: [INAUDIBLE] Gaussian, so they're all [INAUDIBLE]

MICHALE FEE: Yep. So it turns out that the Fourier transform in a positive frequency is just the complex conjugate of the Fourier transform in the negative frequency. So the square magnitude is identical. So the power spectrum on this side is equal to the power spectrum on that side. So we just plot half of it. So that's what the power spectrum of this particular piece of signal looks like. The power spectrum of noise is very noisy.

We're going to come back, and I'm going to show you that on average, if you take many different signals, many copies of this, and calculate the power spectrum and average them all altogether, it's going to be flat. But for any given piece of noisy signal, the power spectrum is very noisy. Any questions about that? OK.

All right, so now let's turn to spectral estimation. How do we estimate the spectrum of a signal? So let's say you have a signal, S of t. And you have a bunch of short measurements of that signal. You have some signal, let's say, from the brain, like you record some local field potential or some ECG or something like that. And you want to find the spectrum of that. Let's say you're interested in measuring the theta rhythm or some alpha rhythm or some other periodic signal in the brain.

What you could do is you could have a bunch of independent measurements of that signal. Let's in this case call them four trials, a bunch of trials. What you can do is calculate the power spectrum, just like [AUDIO OUT] for each of those signals. So this is a little sample y of t, y1 of t, y2 of t. You can calculate the square of magnitude of the Fourier transform of each one of those samples. And you can estimate the spectrum of those signals by just averaging together those separate, independent estimates.

Does that make sense? So literally, we just do what we did here. We have a little bit of signal. We Fourier transform it, take the square magnitude. And now, you average together all of your different samples.

Does that makes sense? That's the simplest form of spectral estimation. It's like if you want to estimate the average height of a population of people. You take a bunch of different measurements. You randomly sample. You take a bunch of different measurements. And you average them together. That's all we're doing here. OK?

Now, you can apply that same principle to long signals. What you do is you just take that signal and you break it into short pieces. And you compute the power spectrum in each one of those windows. And again, you average them together.

Now, extracting that little piece of signal from this longer signal is essentially the same as multiplying that long signal by a square pulse. 0 everywhere, but 1 in here, 1 here. 0 everywhere else. Right? So that process of taking a long signal and extracting out one piece of it has a name. It's called windowing. Sort of like you're looking at a scene through this window, and that's all you can see.

OK, so one way to estimate the spectrum of this signal is to take the signal in this window, compute the FFT of that, take this power spectrum. And then apply this window to the next piece. Apply this window to the next piece and compute the spectrum and average them all altogether.

What's the problem with that? Why might that be a bad idea? Yeah.

AUDIENCE: Could we [INAUDIBLE].

MICHALE FEE: Good. That's a very good example. But there's sort of a general principle that we

just learned that you can apply to this problem. What happens to the Fourier transform of the signal when we multiply it by this square pulse?

AUDIENCE: Convolving.

MICHALE FEE: We're convolving the spectrum of the signal with a sinc function. And the sinc function is really ugly, right? It's got lots of wiggles. And so it turns out this process of windowing a piece of data with this square pulse actually does really horrible things to our spectral estimate.

And we're going to spend a lot of time in the next lecture addressing how you solve that problem in a principled way and make a good estimate of the signal by breaking it up into little pieces. But instead of just taking a square window we do something called tapering. So instead of multiplying this signal by square pulses, we sample the signal by applying it by little things that look like little smooth functions, like maybe a Gaussian, or other functions that we'll talk about do an even better job. OK? All right.

OK, so that process is called tapering, multiplying your data by a little [AUDIO OUT] paper that's smooth, unlike a square window. Computing spectral estimates from each one of those windowed and tapered pieces of data gives you a very good estimate of the spectra. And we're going to come back to that, how to really do that right, on Thursday. OK?

All right, let me point out why this method of spectral estimation is very powerful. So, remember, we talked about how you can see-- remember, we talked about if you have a noisy signal that has a little bit of underlying sine wave in it, we talked about in class, if you take the autocorrelation of that function, you get a delta function and then some little wiggles. So there are ways of pulling periodic signals, periodic structure out of noisy signals.

But it turns out that this method of spectral estimation [AUDIO OUT] did the most powerful way to do it. I'm just going to show you one example. This blue function here is noise, plus a little bit of a sine wave at, I think it's 10 hertz. OK, yeah. Anyway, I didn't write down the frequency.

But the blue function here is noise plus the red function. So you can see the red

function is small. And it's buried in the noise, so that you can't see it. But when you do this process of spectral estimation that we're learning about, you can see that that signal buried in that noise is now very easily visible. So using these methods, you can pull tiny signals out of noise at a very bad signal to noise ratio, where the signal is really buried in the noise. So it's a very powerful method. And we're going to spend more time talking about how to do that properly.

All right, so let me spend a little bit more time talking about the power spectrum of noise, so that we have a better sense of what that looks like. So remember, I told you if you take a sample of noise like this and you estimate the spectrum of it, you compute the power spectrum of one sample of noise, it's extremely noisy. Let's see, I'm just going to remind you what that looks like. That's the power spectrum of one sample of noise.

In order to estimate what the spectrum of noise looks like, you have to take many examples of that and average them together. And when you do that, what you find is that the power spectrum of noise is a constant. It's flat. [AUDIO OUT] Gaussian noise.

The power spectrum, really, you should think about it properly as a power spectral density. There is a certain amount of power at different frequencies in this signal. So there is some power at low frequency, some power at intermediate frequencies, some power at high frequencies. And for Gaussian noise, that power spectral density is flat. It's constant as a function of frequency. OK?

And the units here have units of variance per unit frequency-- variance per frequency. OK? Or if it were an electrical signal going through a resistor, it would be power per unit frequency. So you can see that here the value here is 0.002. The bandwidth of this signal is 500 hertz. And so the variance is the variance per unit per unit frequency times the bandwidth. And that's 1. And we started with a Gaussian noise that has variance 1, that when we calculate the power spectrum of that we can correctly read out from the power spectrum how much variance there is per unit frequency in the signal. OK?

All right, it's kind of a subtle point. I actually don't expect you to know this. I just wanted you to see it and hear it. So you know formally what it is that you're looking

at when you look at a spectral estimate of a noisy signal.

All right, let's talk about filtering in the frequency domain. So remember, we learned how to smooth a signal, how to filter a signal, either high pass or low pass, by convolving a signal with a kernel. So you remember that the kernel for a low pass was something like this. So when you convolve, that's the kernel for a low pass. And for a high pass, anybody remember what that looks like?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yep. So-- sorry, I should be a little more careful here not to mix up my axes with-- I'm going to remove that. So that's the kernel for a low-pass filter. The kernel for a high-pass filter is a delta function that reproduces the function. And then you subtract off a low-pass filtered version of the signal. OK? So that's the kernel for a high pass. OK, so this was how you filter a signal by convolving your signal with a function, with a linear kernel.

We're going to talk now about how you do filtering in the frequency domain. So if filtering in the time domain is convolving your [AUDIO OUT] with a function, what is filtering in the frequency domain going to be?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Right. It's going to be multiplying the Fourier transform of your signal times what?

AUDIENCE: The Fourier transform [INAUDIBLE]

MICHALE FEE: The Fourier transform of things like that. All right, so let's do that. So this is what we just talked about. We introduced the idea before. This was actually a neural signal that has spikes up here and local field potentials down here. And we can extract the local field potentials by smoothing this [AUDIO OUT] by low pass filtering it, by convolving it with this kernel here.

So this is what we just talked about. So if filtering in the time domain is convolving your data with a signal, then filtering in the frequency domain is multiplying the Fourier transform of a [AUDIO OUT] times the Fourier transform of the kernel. And you can see that what this does to the power spectrum is just what you would expect. The power spectrum of the filtered signal is just the power spectrum of your

original signal times the power spectrum of the kernel.

All right, so here's an example. So in blue is the original Gaussian noise. In green is the kernel that I'm smoothing it by, filtering it by. Convolving the blue with the green gives you the red signal. What kind of filter is that called again? High pass or low pass?

AUDIENCE: Low pass.

MICHALE FEE: Low pass, good. All right, so let me play you what those sound like. So here's the original Gaussian noise. [STATIC] Good. And here's the low pass Gaussian noise. [LOWER STATIC] It got rid of about the high frequency parts of the noise.

OK, so here's the power spectrum of the original signal in blue. In order to get the power spectrum of the filtered signal in red, we're going to multiply that by the magnitude squared Fourier transform of this. What do you think that looks like? So this is a little Gaussian filter in time. What is the Fourier transform of that going to look like?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: The Fourier transform of a Gaussian is a Gaussian. So the power spectrum of that signal is going to just be a Gaussian. Now, how would I plot it? It's peaked where? The Fourier transform of a Gaussian is peaked at 0. So it's going to be a Gaussian here centered at 0. We're only plotting the positive frequencies. So this, we're ignoring. And it's going to be like that, right?

So that's the Fourier transform, squared magnitude Fourier transform of that Gaussian. It's just another Gaussian. And now if we multiply this power spectrum times that power spectrum, we get the power spectrum of our filtered signal. Does that makes sense?

So convolving our original blue signal with this green Gaussian kernel smooths the signal. It gets rid of high frequencies. In the frequency domain, that's like multiplying the spectrum of the blue signal by a function that's 0 at high frequencies and 1 at low frequencies. Does that makes sense?

So filtering in the frequency domain, low filtering in the frequency domain, means

multiplying the power spectrum of your signal by a function that's low at high frequencies and big at low frequencies. So it passes the frequencies and suppresses the high frequencies. It's that simple.

Any questions about that? Well, yes--

AUDIENCE: So why is it that like-- you need to like-- of I guess when you multiply in the frequency, could you theoretically multiply by anything and that would correspond to some other type of filter. So why don't we just like throw away high frequencies? Or something like multiply by a square in the frequency domain and correspond to some different filter we don't know.

MICHALE FEE: Yeah. You can do that. You can take a signal like this, Fourier transform it, multiply it by a square window to suppress high frequencies. What is that equivalent to? What would be the corresponding temporal kernel that that would correspond to?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Good. It would be convulsing your function with a sinc function. It turns out that's-- the reason you wouldn't normally do that is that it mixes the signal across all time. The sinc function goes on to infinity. So the nice thing about this is when you smooth a signal with a Gaussian, you're not adding some of the signal here that were over here. Does that makes sense? Convolving with a sinc function kind of mixes things in time.

So normally you would smooth by functions that are kind of local in time, local in frequency, but not having sharp edges. Does that makes sense? So we're going to talk about how to smooth things in frequency with signals with kernels that are optimal for that job. That's Thursday.

What would a high-pass filter look like in the frequency domain? So high-pass filter would pass high frequencies and suppress low frequencies. Right? You've probably not heard of it, but, what would a band pass filter look like? It would just pass a band. So it'd be 0 here. It would be big somewhere in the middle and then go to 0 at higher frequencies. OK? Does that makes sense? Any questions about that? OK. Good.

If we plot this on a log plot in decibels, you can see that on a log plot, a Gaussian,

which is e to [AUDIO OUT] like f squared. On a log plot, that's minus f squared. Right? That's why on a log plot this would look like an inverted parabola. So that's the same plot here, but plotted on a log scale. Any questions about that?

I want to tell you about a cool little theorem called the Wiener-Khinchin theorem that relates the power spectrum of a signal with the autocorrelation of a signal. So in blue, that's our original Gaussian noise. In red is our smooth Gaussian noise. If you look at the correlation of neighboring time points in the blue signal, you can see they're completely uncorrelated with each other. But what about neighboring time points on the smooth signal? Are they correlated with each other?

If we look at for the red signal, y of i and y of i plus 1, what does that look like for the red signal? They become correlated with each other, right? Because each value of the smooth signal is some sum over the blue points. So neighboring points here will be similar to each other. That's what smoothness means. Neighboring time points are close to each other. So if you look at the correlation of neighboring time points in the smooth signal, it looks like this. It has a strong correlation.

So if you look at the autocorrelation of the original Gaussian noise, it has a delta function at zero. The autocorrelation of the smoothed function has some width to it. And the width to that autocorrelation function tells you the time [AUDIO OUT] this signal was smoothed. Right? OK.

Now, how does that relate to the power spectrum? So it turns out that the power spectrum of a signal, the magnitude squared of the Fourier transform, the power spectrum of the signal is just the Fourier transform of the autocorrelation. So what's the Fourier transform of a delta function? Anybody remember that?

AUDIENCE: Constant.

MICHALE FEE: It's a constant. And how about our smoothed? Our smooth signal has a power spectrum that's a Gaussian in this case. What's the transform of a Gaussian?

AUDIENCE: [INAUDIBLE]

MICHALE FEE: OK. So if you have a signal that you have some sense of what the power spectrum is you immediately know what the autocorrelation is. You just Fourier transform that

and get the autocorrelation. What's the width of this in time? How would I get that from here? How are the width in time and frequency related to each other for--

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Right. The width of this in time is just 1 over the width of [AUDIO OUT] So you have to take the full width. Does that makes sense? OK. Wiener-Khinchin theorem, very cool.

All right, let's talk about the Shannon-Nyquist theorem. Anybody heard of the Nyquist limit? Anybody heard of this?

All right, so it's a very important theorem. Basically anybody who's acquiring signals in the lab needs to know the Shannon-Nyquist theorem. It's very important.

All right, so remember that when we have discrete Fourier transforms, fast Fourier transforms, our frequencies are discretized and so is our time. But the discretization in frequency means that the function is periodic. Any signal that has discrete components and frequencies is periodic in time. Remember, we started with Fourier series. And we talked about how if you have a signal that's periodic in time, that you can write it down as a set of frequencies that are integer multiples of each other.

Now, in these signals, time is sampled discretely at regular time intervals. So what does that tell us about the spectrum, the Fourier transform?

AUDIENCE: It's periodic.

MICHALE FEE: It's also periodic. OK, so discretely sampled in frequency at regular intervals means that the signal is periodic in time. Discretely sampled in time means that the Fourier transform is periodic.

Now, we don't usually think about this. We've been taking these signals, sines and cosines and square pulses and Gaussian things, and I've been showing you discretely sampled versions of those signals. And I've been showing you the Fourier transforms of those signals. But I've only been showing you this little part of it. In fact, really be thinking that those discreetly sampled signals have a Fourier transform that's actually periodic. There's another copy of that spectrum sitting up here at 1 over the sampling rate and another copy sitting up here.

Remember, this is like a train of delta functions. The Fourier transform of that is like another train of delta functions. So there are copies of this spectrum spaced every 1 over delta t. It's kind of a strange concept.

So the separation between those copies of the spectra in the frequency domain are given by 1 over the sampling rate. Any questions about that? It's a little strange. But we'll push on because I think it's going to be more clear.

So what this says is that if you want to properly sample this signal in time, you need these [AUDIO OUT] copies of its spectrum to be far away so they don't interfere with each other. So what that means is that you need the sampling rate to be high enough-- the higher the sampling rate is, the further these spectra are in time. Delta t is very small, which means 1 over delta t is very big. The sampling rate needs to be greater than twice the bandwidth of the signal. [AUDIO OUT] bandwidth B.

So if the sampling rate is less than twice the bandwidth, what happens? That means delta t is too big. These copies of the spectrum are too close to [AUDIO OUT] and they overlap. That overlap is called aliasing-- a-l-i-a-s-i-n-g. OK?

So you can see that if you sample a signal at too low a sampling rate and you look at the spectrum of the signal, you see that it has-- like you'll see this part of the spectrum, but you'll also see this other part of the spectrum kind of contaminating the top of your Fourier transform. Does that makes sense? OK.

So let me just say it again. If your signal has some bandwidth B that in order to sample that signal properly, your sampling rate needs to be greater than twice that bandwidth, 1, 2. OK? All right, any questions about that?

Actually, there was actually recently a paper where somebody claimed-- I think I told you about this last time-- there was a paper where somebody claimed to be able to get around this limit. And they were mercilessly treated in the responses to that paper. So don't make that mistake.

Now, what's really cool is that if the sampling rate is greater than twice the bandwidth, something amazing happens. You can perfectly reconstruct the signal. Now that's an amazing claim. Right? You have a [AUDIO OUT] time. All right, it's wiggling around. What this is saying is that I can sample that signal at regular

intervals and completely ignore what's happening between those samples, have no knowledge of what's happening between those samples. And I can perfectly reconstruct the signal I'm sampling at every time point, even though I didn't look there.

So how do you do that? Basically, your sampled signal, you're regularly sampled signal, has this spectrum-- has this Fourier transform with repeated copies of the signal, repeated copies of the spectrum. So how would I recover the spectrum's original signal? Well, the spectrum of the original signal is just this piece right here.

So all I do is in the frequency domain I take that part. I keep this, and I throw away all those. In other words, I multiply my Fourier transfer sampled signal in the frequency domain by a square pulse that's 1 here and 0 everywhere else. Does that makes sense?

And when I inverse Fourier transform that I've completely recovered my original signal. What is multiplying this spectrum by the square wave in the frequency domain equivalent to in the time domain?

AUDIENCE: So I was going to ask--

MICHALE FEE: Yeah.

AUDIENCE: [INAUDIBLE]

MICHALE FEE: Yeah.

AUDIENCE: So why do you want to do that?

MICHALE FEE: Yeah, so it's amazing, right? It's cool. So let me just what it is. And then we can marvel at how that could possibly be.

Multiplying this spectrum by this square wave, throwing away all those other copies of the spectrum and keeping that one is multiplying by a square wave in the frequency domain, which is like doing what?

AUDIENCE: Convolving.

MICHALE FEE: Convolving the time domain sinc-- that regular train of samples, convolving that

with a sinc function. So let me just say that. Here, we have a function that we've regularly sampled at these intervals. If we take that function, which is a bunch of delta functions here, here, here, here, just samples, and we can evolve that with a sinc function, we perfectly reconstruct the original signal. Pretty wild.

So that's the Nyquist-Shannon theorem. What it says is that we can perfectly reconstruct the signal we've sampled as long as we sample it at a sampling rate that's greater than twice the bandwidth of the signal. OK? All right. OK, good.

So there's this cute trick called zero-padding, where you don't perfectly reconstruct the original signal, but basically you can interpolate. So you can extract the values of the original signal times between where you actually sampled it. OK? And basically the trick is as follows.

We take our sampled signal. We Fourier transform it. And what we do is we just add zeros. We pad that Fourier transform with zeros. OK? So we just take positive frequencies and the negative frequencies, and we just stick a bunch of zeros between and make it a longer vector.

And then when we inverse Fourier transform this, you can see that you have a longer array. When you inverse transform, inverse Fourier transform, what you're going to have is your original samples back, plus a bunch of samples in between that interpolate, that are measures of the original signal at the times where you didn't measure it.

So you can essentially increase the sampling rate of your signal after the fact. Pretty cool, right? Again, it requires that you've sampled at twice the bandwidth of the original signal. Yes.

AUDIENCE: Like how do you know the bandwidth of the original signal if you don't have samples?

MICHALE FEE: Good question. How might you do that?

AUDIENCE: Can you like [INAUDIBLE] different sampling lengths to get [INAUDIBLE]

MICHALE FEE: You could do that. From nearly all applications, you have a pretty good sense of what the frequencies are that you're interested in a signal. And then what you do is

you have to put a filter between your experiment and your computer that's doing the sampling that guarantees that it's suppressed all the frequencies above some point. OK?

And that kind of filter is called an anti-aliasing filter. So in that case, even if your signal had higher frequency components, the anti-aliasing filter cuts it off so that there's nothing at higher frequencies. Does that makes sense?

Let me give you an example of aliasing. Let's say I had this signal like this. And I sample it here, here, here, here, here, here. I need to do regular intervals. So you can see that if I have a sine wave that is close in frequency to the sampling rate, you can see that when I sample the signal, I'm going to see something at the wrong frequency. That's an example of aliasing. OK?

OK, so here's an example. We have a 20 hertz cosine wave. I've sampled it at 100 hertz. So I'm, you know, 5-- so what frequency would I have to sample this in order to reconstruct the cosine? I'd have to sample at least 40 hertz. Here, I'm sampling at 100. The delta t is 10 milliseconds. So those are the blue points.

And now, if I do this zero-padding trick, I Fourier transform. I do zero-padding by a factor of 4. That means if I take the Fourier transform signal and I'm now making that vector 4 times as long by filling in zeros, then I inverse Fourier transform. You can see that the red points show the interpolated values of that function after zero-padding. OK? So it can be a very useful trick.

You can also sample the signal in the time domain and then add a bunch of zeros to it before you Fourier transform. And that gives you finer samples in the frequency domain. OK?

And I think that's-- so zero-padding in the time domain gives you finer spacing in the frequency domain. And I'll show you in more detail how to do this after we talk about tapering. And it's very simple code actually. Matlab has built into it the ability to do zero-padding right in the FFT function.

OK, let's actually just stop there. I feel like we covered a lot of stuff today.