

MITOCW | 16. More on modulation/demodulation

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high-quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

GEORGE I want to actually spend a little time, not too much, talking about discrete-time Fourier transforms versus
VERGHESE: discrete-time Fourier series. You don't have any basis for comparison, but I think the way we've told the spectral content story this term is quite a bit simpler than in previous terms, but that leaves you puzzling over chunks of the notes and practice problems that refer to the discrete-time Fourier series and to periodic signals and so on. So I just wanted to give you a little insight into that, and then we'll go on to talk about modulation and demodulation some more.

OK, so we've seen that our interest is generally in signals of finite duration because practical computation has to deal with that. And so we've got signals of this form-- 0 outside of some window, and really without loss of generality, I can take it to be some window from 0 to $L - 1$. If the signal shifts in time, we know what to do with the Fourier transform. Can you hear me all right, by the way, at the back? Yeah, OK.

All right so if we've got non-zero values only over a finite range, then the computation of the discrete-time Fourier transform boils down to a simple finite computation. Now, what we'll typically do is give ourselves a little more flexibility. Since the signal is 0 outside of this interval anyway, we might sometimes allow ourselves to think of the signal as being longer, but still with zeros out here. So you might come all the way up to some $P - 1$.

And what we're saying is, this is the window of interest. Everywhere outside of this window, the signal is 0. Now, the signal can be 0 at various points inside here, as well, but what we're saying is, outside of this interval, the signal is 0. Therefore, I only need to compute this from 0 to $P - 1$, all right? And the nice thing is, it turns out that you can actually recover the time-domain signals from the samples of the DTFT through the formula on the right side.

So what we're doing is we're actually computing the DTFT just at isolated points on the axis between $-\pi$ and π , just P , capital P , points. Or you can think of them as points on the unit circle that correspond to each of those exponentials that appear in the Fourier transform definition. And we then recover the time-domain signals just from those samples, OK? And really what's driving this is the fact that the signal is 0 outside of a finite window. OK.

We'll also typically-- and if you look in the books, you'll see this, as well-- this notation often gets simplified, so $x[k]$ of ω sub k gets simplified to just x sub k . It's the k -th spectral coefficient. All right, so all that is good.

And we have this nice algorithm for computing things, which is the fast Fourier transform. So we talked about how that significantly reduces computation. Now, there are properties of these formulas that you can explore, and I have some listed here. I'm not going to go through them. They're essentially the same properties we've seen for the DTFT.

I want to focus more on this formula for reconstruction of the time signal from the spectral coefficients. By the way, in a previous writing of this formula, I had written the upper limit as $P/2$. It's actually $P/2 - 1$, so I'll fix that in the earlier slides.

OK, so what you're guaranteed is that if you apply that formula, you will recover every signal value in this window of length capital P. But what happens outside of that window? Well, if you look at this expression, is the right-hand side here periodic? You should suspect that it is because of the e to the j something n is there, right? If you look at the definition of omega sub k and look at each of these terms, it turns out that each of these will repeat periodically with period 2 pi.

Sorry, with period-- let's see. I've said it badly. This whole term will repeat when n increases by capital P, all right? So let me write it down. And why is that? Well, omega sub k is 2 pi over P. So if you increase time by capital P, you're going to increase the exponent by 2 pi, and you've got the same exponential back again. And you can do this for any integer multiple of capital P.

So what that tells you is that the expression on the right-hand side is actually going to repeat periodically outside of this interval. So it's fine to use this formula to recover the values in this window, but if you start to evaluate this formula outside of that window, you're going to start getting this whole thing repeated periodically, so you're going to get-- at this point you'll get-- and so on, OK?

So the formula doesn't know what to do except to replicate periodically. It's up to you to know that this formula is no good outside of this window. All right? There's another way to think of it, though, which is that this formula gives you a nice, compact representation for a periodic signal. So if you started off with a periodic signal, here's a way to represent it as just a sum of capital P exponentials, and that's what a Fourier series is.

So you've seen in 1803 or other places, in continuous time I imagine, that if you had a periodic signal you could represent it with a Fourier series. This is actually a Fourier series for this periodic signal. But if you know that your action of interest is all in this finite interval within one period, then you can actually use the Fourier series just to study what goes on in that one interval without worrying about what's outside.

And that's really what we've done this term, is we've kind of ignored periodic signals. We've said all the attention is in a finite interval. Within that interval, we have this Fourier representation. It's easily computed by the FFT, and everything works nicely. So just to give a concrete illustration of how we end up applying this in a particular situation that should be familiar, if I had an input going into an LTI system producing an output, and if the input was non-zero only from 0 to, let's say some n sub x, and if the unit sample response of the system was zero only from 0 to n sub h, is there a particular interval of time that you can guarantee for me will contain all the non-zero values of the output? I want you to find for me an interval outside of which the output is guaranteed to be 0. Anybody? Yeah?

AUDIENCE: [INAUDIBLE]

GEORGE VERGHESE: Yeah, good. There are many ways to think of this. One is to say, well, the input value at time 0 fires off a unit sample of duration, goes from 0 to n sub h, and then the input value at time 1 fires off a unit sample response that starts one time later, and so on. So each of these fires off a unit sample response. Well, you've got inputs extending from 0 to n x, and so you're going to have an output that extends from 0 to n x plus n h, all right?

So you're guaranteed that all the action of interest happens in this finite interval. And given that that's the case, you can actually-- whoops, what happened? You can actually do this kind of spectral representation, use the FFT, and all of that. You're going to just work on a finite interval, 0 to P minus 1, defined by that or greater than that, OK?

So this is actually one of the most frequent uses of the FFT. It's to study systems where all the action happens in a finite window and you know a priori what the length of that window is, and you can then do all your computations there. And you never look outside that window because you've already guaranteed that everything of interest happens there.

But when you read the notes, you'll find it's essentially the same story, but when you talk about Fourier series you're actually talking about the whole signal, the periodic signal, all right? One bit of notation, also, as you're reading the notes, just to go back a second here.

We've been working entirely in terms of these samples of the DTFT. When you're thinking of Fourier series, when you're thinking of this as a Fourier series, it's typical to write $X_{\omega k}$ over P as just the Fourier coefficient, $A_{\text{sub } k}$. So you'll see in the notes an $A_{\text{sub } k}$. That's just a normalized version of the Fourier transform sample, OK?

All right. That's as much as I wanted to say on this, so let's get back to talking about modulation and demodulation. If you have questions on what I talked about, you can bring them up in recitation.

All right, so just to review where we are. We've got some signal, x of n at baseband. Baseband just means that its frequency content is centered around zero. You've not done any modulation or shifting yet. You've been allotted some part of the frequency axis to do your transmission in because someone's told you, perhaps, that the medium that you're going to use can only transmit in that range, or the FCC has decreed that you're only going to use that region.

So you want to send that signal somehow in another frequency band. So modulation was a process by which we converted up to some carrier frequency, and then demodulation was what you did with the receiver to get back down. So just to look at that in a little more detail.

This is the modulation process we talked about last time. You've got a time-domain signal, your information signal. You multiply it by the cosine to get an amplitude-modulated transmitted signal. So t of n is the signal that you transmit, OK? There are other names for this. This process of multiplying a signal by a cosine at a particular frequency is referred to as heterodyning. That's a term from the earliest days of Amplitude Modulation, I think invented by Fessenden, who also invented AM, and of course it's specifically amplitude modulation for us.

All right, so just to think spectrally, we had a simplified version of this picture last time, but let's first assume that this signal has some spectrum, which is shown by a cartoon here. I'm assuming a real signal. So we know that the spectrum has a real part that's even and an imaginary part that's odd, and that's what's shown for you here on this figure, OK? So we're going to track the spectrum of the signal by tracking the real and imaginary parts separately because the spectrum is in general a complex function of frequency.

We've seen last time what happens when you multiply by the cosine. You take the spectrum, and you replicate it at the locations of the carrier. So if your carrier frequency is ω_c , here's your frequency band, going from minus π somewhere there to plus π somewhere here. You've got a plus and minus ω_c , the carrier frequency.

So what happens when you modulate is you take the spectrum and you plunk it down on plus ω_c and minus ω_c , and you scale by $1/2$, all right? So if the real part had amplitude a before, it now has amplitude $a/2$, and the imaginary part, similarly. I haven't drawn these to scale, but hopefully the labels are clear enough.

OK, so the modulation is not simple when you're thinking of what it does in the frequency domain. Now, it is not simple, but this picture is a little deceptive, perhaps, because I made an implicit assumption here. Otherwise, the picture would be a bit messier. What am I relying on here to get this simple picture? Yeah? Sorry?

AUDIENCE: [INAUDIBLE]

GEORGE VERGHESE: Centered? What's centered at zero? Oh, the spectrum here? OK, yeah, the spectrum centered at here gives me a simple picture, yeah.

AUDIENCE: [INAUDIBLE]

GEORGE VERGHESE: OK, exactly. You see, when I drew this here, you can still recognize the triangles that came from over there. But if the baseband signal had a frequency content that extended way over, then the replication that I have here would actually leak into the replication that I have there, and I get a more complicated picture, all right?

So if you want the simple picture, you actually have to limit the frequency content of your baseband signal, OK? You can see here, if the signal only extends ω_c on either side, then I'm OK. The two replications will not smear into each other, all right? So we need a limit on the frequency content.

Now, the specific limit that you have depends on the application. We'll see later when you do frequency-division multiplexing, where you're trying to put many different signals in the same general frequency band, that the restrictions might be different. But the basic idea is this, that you want any replication of your signal, if you're going to extract it later on downstream somewhere, you want the replication to not be corrupted by images of it somewhere else, or images of some other signal.

So actually, the example that I showed you last time wasn't perfect in that regard, right? Remember, this was the spectrum of our typical baseband. We had 256 samples, like this, and then 0's. And we looked at the spectral content. It was given by a sinc-like function, and this is the spectral content magnitude after modulation, and therefore it's the two replicas. I'd modulated this onto a 1,000 Hertz carrier. So this is what we saw.

And you can see here that there's funny stuff going on in here because the tails of the two replications are merging with each other, OK? So it's not perfectly symmetrical around here. And actually, these sinc-like functions decay very slowly, so even though it won't be visible to your eye, there's a considerable amount of this that's actually due to the replica out here, OK? So this case doesn't quite satisfy that band-limited condition.

If you shape your pulse a little bit more carefully, for instance, if you had more rounded edges, then you can pull in the frequency content, and you might do a better approximation to keeping the replica separate. Or you might use a higher carrier frequency. That pull them apart and have less interference, but it's certainly an issue that you need to think about. OK.

So what happens at the receiver? We already saw this briefly at the end of lecture. If what you receive is what you transmit-- in other words, if it's the signal, then extracting the $x(n)$ is easy. We said what you do is you basically do the same heterodyning again, right? You take the signal that comes in, multiply it by cosine of the carrier frequency.

That's your signal after demodulation, and a little bit of algebra shows that you actually have your original signal of interest, and then something that's your original signal modulated by a cosine at twice the carrier frequency. So now there's some hope that you can actually pull these things apart.

All right. So one question, of course, is what does the spectrum of this look like, and we'll look at that. And then the other question is, again, what constraint on the bandwidth of the signal that you originally sent from the transmitter-- what constraint is needed to recover? So let's look at the spectrum of the received signal first.

We're assuming the channel is not distorting and that we don't have noise, so what's transmitted is also what's received. So here is the spectrum of what's received. It's exactly the spectrum I showed you earlier, right? It was the baseband spectrum, but replicated at plus ω_c and minus ω_c . So this is what comes in off the channel, assuming no distortion.

And I'm going to multiply it again by a cosine at the carrier frequency. So what is it that I have to do? I take this entire spectrum, plunk down a copy centered at plus ω_c , and another copy at minus ω_c . Because my demodulation, just to remind you, my demodulation is multiplication by cosine ω_c again. It's a multiplication by cosine ω_c . Well, we know what that does in the transform domain, so here is the picture.

And the piece that we want is the center piece. So what we need to do is filter it out of what's resulted from the heterodyning. So what kind of cutoff frequency would you-- what kind of filter and what kind of cutoff frequency would you want? Any suggestions?

AUDIENCE: [INAUDIBLE]

GEORGE Sorry, I didn't hear where that came from. Yeah?

VERGHESE:

AUDIENCE: Lowpass filter?

GEORGE Lowpass filter. So, for instance, an ideal lowpass filter would be great, right? If you had a filter with a frequency-domain characteristic that was perfectly flat in some region and then cut off, let me say, at some frequency ω_0 , so something like that-- well, actually, we want a factor of 2 to compensate for the demodulation process, if we want to get exactly the same thing back. So this would be in the frequency domain. Ideal lowpass filter.

VERGHESE:

And we know how to get approximations to this, right? Because this is not really implementable. If you wanted to implement this, what kind of unit sample response would you need? A sinc function, right, but extending infinitely in both directions. But we could truncate that sinc, and we could shift it forwards in time to get a causal approximation to this filter. And the resulting frequency response will, if you plot it out, if you compute it and plot it, won't look too different from this. If I plotted the magnitude, you'll get-- it's something that's a plausible approximation to this lowpass filter.

But what cutoff frequency would you want? What ω_0 should you pick? Any suggestions? Anybody? We're trying to extract this piece. So ω_c would be a pretty safe choice, right? ω_c would be one that passed everything here, and would basically extract any signal that satisfied that initial constraint that we mentioned. So if your baseband signal originally extended from minus ω_c to plus ω_c , then a lowpass filter that extracted that would do fine without pulling in any of the replication here. So ω_c is certainly fine.

But if the signal that you transmitted at baseband actually had a narrower bandwidth than that, then you might just want to get away with a lowpass filter with a lower cutoff. Can you think of why you might want to do that? Is there anything that motivates you to use as small a bandwidth as possible? Yeah?

AUDIENCE: [INAUDIBLE]

GEORGE VERGHESE: To limit the amount of noise, right? We suppress noise in this whole story. So if you're going to build a filter like this but all the interesting action is over here, well, all of the rest of the filter is doing is letting other signals get in, especially noise, and then that's going to add to the output and make things more difficult. So you'd really like to get the smallest bandwidth that suffices to pass the signal part of what you're interested in but keep out the noise, all right?

But if you didn't know anything about the signal and its spread, or you believe that the spectrum extended really from minus ω_c to plus ω_c , then you would want to make ω not equal to the carrier frequency, right? But you've got to look at your particular situation and see what it is you're going to do.

OK, so this is the picture that we have at demodulation. You're going to take the received-- well, no, sorry. This is the modulation part. Ah, no, it's not. Sorry, this is not well-drawn. That shouldn't be x_n . That should be the received signal, OK? So the received signal comes in, gets multiplied locally by cosine, gives you the demodulated signal, and then you have the lowpass filter, so I'll change that before I post it. Simple enough?

OK. Now, there are some problems that you can run into, and doing all of this in the lab you actually see that very quickly. So let me actually put this on the board here. What we said is that our demodulated signal is going to be our received signal times cosine $\omega_c n$, right? And if we assume no distortion in the channel this, is $x_n \cos \omega_c n$.

But there's a bit of a problem here, which is that, even if you've been told what carrier frequency your center is going to use, you might not know exactly what phase. It's typically the case that you don't know what the phase is on this cosine. So you know ω_c , but you don't know exactly the phase, which means that your local carrier, your local oscillator or your local carrier multiplication here, will end up having some offset relative to the carrier used at the transmitting end, OK?

And so the question is, if we track this through, what happens through the demodulation process, so that's really what this is trying to do. So we're saying d_n is your received signal, but the local oscillator or the local cosine that you're heterodyning with at the receiver doesn't know exactly what phase was used as the transmitter, so you've got to assume that there is going to be some offset.

So this is actually what the multiplication is. And now you use a simple trig identity. It's the cosine of something times the cosine of something. That splits into this. And so what we're actually going to get from the heterodyning at the receiver is x_n times all of this, OK?

So what we're going to get is-- I should write it down here. We're going to get $0.5 x_n$. And then there's two pieces here. There is the cosine ϕ , and then there is the cosine $2\omega_c n - \phi$. OK? When you don't have any phase error, the cosine ϕ term is 1, but now it's reduced from that.

The rest of the process is the same. You're going to do some filtering to get rid of this piece, the double frequency piece, and you're going to pull out just what you're interested in. Except now it's no longer x of n itself. It's x of n multiplied by this cosine. So can you see that this could lead you into trouble? What's the worst case here? Sorry, worst case-- yeah?

AUDIENCE: [INAUDIBLE]

GEORGE
VERGHESE: Yeah, if ϕ is π over 2, then cosine ϕ is 0, and you get nothing, all right? So if you're unlucky in the offset between your local sinusoid and the sinusoid that was used at the transmitter, you could end up with nothing, OK? You can also get the negative of what was sent, and so on, so you can go through the whole set of possibilities there.

So the case of a phase error of π over 2 corresponds to looking at a signal that was transmitted on a cosine and multiplying it by a sine, OK? And you can think through that in the spectral setting, as well. Maybe you'll do some of this in recitation, or maybe you already have. But when you multiply by a sine in the spectral domain--

So if you've got-- you've got your received signal, r of n , and now you're multiplying it by sine $\omega_c n$, right? Sine $\omega_c n$, well, that's $\frac{1}{2j} e^{j\omega_c n} - \frac{1}{2j} e^{-j\omega_c n}$, right? So in the spectral domain, what happens?

Well, you've got r of n multiplied by $\frac{1}{2j}$ times this first exponential. In the spectral domain, that does a shifting and a multiplying by $\frac{1}{2j}$, and then you've got this term doing the same kind of thing. So you're going to have a shift of the spectrum of r of n in the frequency domain and a scaling by $\frac{1}{2j}$.

So if you think through what the shifting and scaling does, you see that it's a little bit more of a complicated picture of what you had over here. Well, the real part gets replicated around the $2\omega_c$ region, but flipped over. The imaginary part gets carried over intact. And then the replications around minus $2\omega_c$ -- sorry, around minus ω_c , that is-- the imaginary part gets flipped over, and the real part gets carried over directly.

Except what was real before becomes imaginary now. What was imaginary before becomes real now. So you can track through all of that. And it just comes from applying the standard DTFT results to what the spectrum of the product of an r of n and this is, OK? But the interesting thing is now, the two replications, when you sum them up, will leave you with nothing at 0 because this piece here will cancel out exactly with that piece there.

So if you think through in the spectral domain what's going on, you'll understand exactly that if you've put your signal on the cosine and you demodulate with a sine, you're going to get nothing in that lowpass region, OK? So that's just the same result but seen spectrally.

All right, so that's uncertainty between the phase of the transmitter and the phase of the receiver. Here's another thing that has a similar effect, which is an unknown delay on the channel, OK? So at the transmitting end, you've got your baseband signal multiplying the carrier. This is what's transmitted. But then you have a time delay, let's say D samples, capital D samples.

So then what's received is actually t of n minus D , and that's what's going to get multiplied by the local carrier. And I'm assuming for now that you have the phase, so we can bring them both together later. But I'm assuming now there's no phase out locally, but there's an unknown delay on the channel.

And you can see it's going to be the same kind of thing. You've got a cosine times a cosine, and the arguments are slightly different from each other. And you use the same trig identity, and what you find is the output of this process is not the input delayed, which is what you would like to get ideally. You aren't going to compensate for the delay with a causal filter, but it's also going to be scaled. And it's going to be scaled by an unknown amount that depends on that delay, all right? So it's the same kind of thing that happens.

So the question is, how do we get around this? And here's one idea that works well, and which you're actually exploring in the lab. Which is to use both the sine and the cosine, OK? So use both the sign on the cosine to demodulate. If you go completely bad on one channel because you've got the phase completely wrong with the cosine, you're going to do all right on the sine channel. If you do completely bad on the sine channel because you've got the phase wrong, then you're going to do all right on the cosine channel.

So at least one of them will work, and more typically both of them will work a little bit, and what you'll then do is combine the two outputs. OK? So you're going to have the signal coming in. There's a cosine multiplication and the sine multiplication, and then the lowpass filtering. We refer to this as the in-phase component, assuming that you were modulating on a cosine, and this is referred to as the quadrature component. So there's in-phase and quadrature.

"Quadrature" just means at right angles. So this is the I and the Q components. And if you work out what these are, assuming now that there's both a time delay and a phase offset, you can see that the in-phase component will be the signal that you want, but multiplied by cosine phi. The quadrature component will be the signal that you want, but multiplied by sine phi. And from there, it's not so hard to imagine that you could actually get back to the signal of interest.

And here's one way to do it that works fine if you've got on-off signaling. So what you would do is, here's the I. Here's the Q. And I've just represented it graphically here. This is typical to do. So here is the I component. Here is the Q component.

And you could take the root sum of squares to basically get rid of that sine theta and cosine theta term, right? So what that's going to give you is the absolute value of x of n minus D . So you can certainly get back the absolute value of what was used to modulate the carrier, and that may be all you need. If you have on-off signaling, that's all you need.

If your modulating signal never goes negative, its absolute value is the same as the signal, so this is fine. So what you will discover is that you get some signal out there, and you're looking for its length. When the length is non-zero, you say you have a 1 cent. When the length of 0, you say you have a 0 cent. In the presence of noise, of course, it won't be exactly at the origin. There might be some cloud of points there. And similarly for the 1 level.

What if you were interested in the polarity, though? So suppose it mattered to you whether the signal was positive or negative. Well, you could then just plot the point and don't take the magnitude. So you'll get something that looks a bit more like this. OK?

So what you'll have is, when a 1 is sent, perhaps you'll get that value in the absence of noise. When a minus 1 is sent-- sorry, when a 0 is sent, corresponding to minus 1, this is what you'd get. This is-- sorry, I should have said that. This is assuming bipolar signaling, right? Bipolar signaling is the case where you're interested in the sign of the signal. You use plus 1 to send a 1. You use minus 1 to send a 0, OK?

So you get some diagram like this. The only problem here is, if you've got uncertain phase and delay, you actually don't know which of these two points corresponds to the plus 1 and which corresponds to the minus 1. So there's that additional ambiguity that needs to somehow be resolved, and there are different procedures you might use.

You could, for instance, have some preamble with a sign that's agreed on and use that as a basis for figuring out which is a plus and which is a minus. And there are other ways of doing it, as well, where what's called differential coding, where basically it's not where that is, but whether it flips over to the other side or not that signals a bit.

And so what you could do is, to transmit a 1, you'll step the phase by π , and that can be detected, and to transmit a 0, you don't change the phase in the next bit slot. So if from one bit slot to the next the dot stays there, you know you've just received a 0. If from one bit slot to the next it flips over to the other side, you know you've just received a 1, OK? So even with the ambiguity, if you change the way you code at the sending end, you can actually compensate for this.

OK, now playing this game with sines and cosines can actually also be done at the transmitting end, and we haven't explored that in class, but it's something that you could think about. So we've been talking about taking the samples and multiplying them onto a cosine carrier.

You could have another bitstream whose samples you multiply onto a sine carrier. And you can just add them together and send them over the channel. At the receiving end, you multiply by cosine. Well, that'll only pull out-- you multiply that cosine and then filter, lowpass filter-- that'll only pull out the first stream in the ideal case. Multiply by a sine and filter, you'll get exactly the second stream.

So you can simultaneously send two streams on a given carrier using this scheme, this method, OK? So depending on how you make out in the lab in problem set 6-- I don't know how many simultaneous carriers you're getting, but whatever you end up with you can actually try now to transmit twice as much on each carrier by using this kind of a scheme. Could be fun.

All right, so this kind of bipolar signaling, what's called phase-shift keying-- I didn't explain that really, did I? I said-- we've said it before, but-- we've talked about bipolar or phase-shift keying. All that we mean is, if you get a signal with voltage plus 1 and minus 1 for your bit 0 on bit 1, by the time you modulate what you're going to end up doing is sending a burst of carrier here with the plus 1. And then when you come to the minus 1 region, you're going to multiply that carrier by minus 1, so you're going to suddenly step the phase, right?

So amplitude modulation with an amplifier that switches between these levels can be also thought of as a phase shifting. So you're keying between a phase of 0 degrees and a phase of 180. So this kind of scheme is used all over the place. And I actually have a slide that lists a whole bunch of schemes that you're familiar with, you see every day in all sorts of literature.

You know, 802, and Bluetooth, and Zigbee and so on. In all of these standards, there's some piece of it or some domain or some regime in which what's going on is some variant of what we've learned here. And they get fancier and more sophisticated, but you really have the key ideas here.

OK, let's now talk about putting multiple signals on a given piece of the spectrum, OK? This is exactly the situation you have in your lab now. You've got the speaker that can transmit in a certain band, and you're trying to put multiple simultaneous signals on it by using different carriers. So this is what's called Frequency-Division Multiplexing, or FDM.

And the idea is very simple. You've got three signals here and this illustration, the blue signal, red, and green. Pick a carrier frequency for each of them. Do the modulation, and then just add them on the channel.

If you've got a linear medium, then the signals will superpose, so what's received is just the sum of these, and now you can do the same kind of thing. And what we're relying on here is, again, the heterodyning principle. Whoops, sorry.

OK, so if you've got frequencies ω_r , ω_b , ω_g in the signal that you're receiving, and you multiply this with some local sinusoidal frequency, ω_0 , where will your various spectra be centered in the result? So the way to think of it is, all the sums at different frequencies here will now appear.

So you get $\omega_0 + \omega_r$. You get $\omega_0 - \omega_r$, and similarly for all of these. OK, it's the same thing that you saw with a single transmitted signal, except now it's a more elaborate constellation. It's actually this that's being transmitted.

So the receiving end, you'll pick a particular frequency to multiply the incoming signal by. The result will have pieces of the spectrum centered at each of these. So if you want to center one of these in your lowpass filter, how should you pick the local oscillator? If you want to tune in a channel, what is it that you want to do?

You want to get one of these center frequencies to sit right in the window of your lowpass filter. So what you'll end up doing is pick your local oscillator frequency to be the carrier frequency of the station you're interested in, or the signal you're interested in, OK? So it's the same idea. You have a lowpass filter, and you're using heterodyning to shift the piece of the spectrum of interest into the passband of the lowpass filter. All right?

Now, what about the bandwidth of the lowpass filter? What should it be? So now it depends on how closely spaced your carriers are, right? So for instance, if I ended up heterodyning such that my blue signal, my blue signal came into the window of interest-- and I've got the red spectrum sitting somewhere here and the other piece sitting somewhere here. OK?

I've shifted things so that this is at 0. What's this frequency now? $\omega_r - \omega_0$ -- what was it? Blue, right? So I've basically shifted these frequencies. So this is at 0, sitting in my lowpass filter. Use a different color. And I want to reject everything else.

So how should I pick that lowpass filter? Well, presumably, you want the cutoff to lie between these two frequencies. So you want half the distance to the nearest carrier, right? Half the distance to the nearest carrier frequency.

OK, so you can think through these. And notice how all our thinking has been in the spectral domain. Thinking in the frequency domain clarifies this whole thing. You really would not have been able to do what you're doing thinking entirely in the time domain.

Now, all of this comes to us, really, from-- let's see. I had that already? Yeah. All of this comes to us from a rich legacy in AM radio. We're not using this for transmission of analog signals by amplitude modulation, but it's the same principle. So these principles actually were studied from the early 1900s, actually. And the AM radio that we see around us now is actually set up exactly to do the kind of thing we're talking about.

So you've got some frequency spectrum that the FCC is allowing you to use. Different stations are given different carrier frequencies that they can operate on. They're also instructed on what bandwidth they can occupy. So basically, the carriers are 10 kilohertz apart, the way that the stations are assigned.

So if you're transmitting from your station, you'd better lowpass filter what you're sending out to 5 kilohertz before you transmit it. Because if you don't, you're going to interfere with a nearby station. Assuming there is a station in the same geographic area that's been assigned to a neighboring carrier frequency, all right? So all of these issues come in.

Another thing that actually is-- what did I do here? I think I-- I mashed together two slides. But the other thing that-- it turns out, for instance, AM radio, at nighttime, because of the way radio transmits, the signal can propagate much further. So these stations are asked to reduce their signal strength, the carrier strength, at nighttime so that they're not interfering with nearby stations. The stations that they would not interfere with during the day they could interfere with at night because propagation characteristics turn out to change.

So all of this business of your signal not interfering with your neighbor's carrier or your neighbor's portion of the spectrum, all of that ends up being important. OK, I think we've probably said as much as we want to say about the signals part of this class.

One of the things about 602 is that those of you who master it come out knowing the subject better than any of us that teach it because there's none of us that's able to teach the course right through start to finish. Well, that's not entirely true. Harry knows how to do it Chris Terman knows how to do it. The recitation instructors hang in there for the whole term, but it's very hard for one person to do that.

So I'm done. I'm going to be sitting there from lecture onwards, so thank you all for your attention, and thank you.