

SPEAKER: 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 to view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at ocw.mit.edu.

PROFESSOR: OK, last time we were talking about this thing called the theorem of irrelevance. At one level, the theorem of irrelevance is just another statement of, when you start looking at detection theory, there are things called sufficient statistics. And what the theorem of irrelevance says is you can ignore things that aren't part of a sufficient statistic. But more specifically, it says something about what those irrelevant things are. And in particular, let me repeat what we said last time but with a little less of the detail.

You start out with a signal set. Ok? You're going to transmit one element of that single-- signal set-- and you're going to turn it into some waveform-- which we'll call X of t . And X of t is going to depend on which particular signal enters the transmitter at this point. You're going to receive something where the thing that you receive has noise added to it. And also if you'll notice I've written the thing that we receive that it's including a whole lot of other stuff.

OK? In other words, the signal is constrained to j different degrees of freedom. And what you get out of the channel involves a much larger set of degrees of freedom. In other words, you're putting things in involving only a certain period of time, in a certain bandwidth, usually. And you can look at anything you want to at the output. The only thing is you can't peek at what the input was at the transmitter, because if you could there will be no sense in actually transmitting it.

OK, so the point is, we receive something which, in the degrees of freedom that we know about, Y_j is equal X_j plus the noise variables. And in the other degrees of freedom, Y_j is just equal to Z_j . And now the rules of the game are that these out of band things, all of these noise coefficients, which are not part of the ϕ_1 up to $\phi_{\text{sub capital J}}$. All of these things, the noise there, is independent of the noise in-- is

independent of the noise and the signal in what we're actually sending. Because it's independent it's irrelevant that's relatively easy to show.

But the broader way to look at this, and the way I want-- and the way I'd like to get you used to thinking about this-- is that this other stuff here can include signals sent by other users. It can include signals sent by you at other times. It can include anything else in the world. And what this theorem is saying is, if you're sending the signal in just these j degrees of freedom, then you don't have to look at anything else. So that all you have to look at is these received components, $Y_{sub j}$.

OK, it says something more than that. It says a lot more than that. These actual orthonormal functions that you use, ϕ_1 up to $\phi_{sub \text{capital } J}$, don't appear in that solution at all. OK? In other words, you really have a vector problem at this point. You have a j dimensional vector that you send. You have a j dimensional vector that you receive. You can use orthonormal functions, which are anything in the world, so long as the noise in that region that you're looking at and sending things in does not vary.

In other words, what you can send is very broadband signals. You can send very narrow band signals. You can send anything and it doesn't make any difference. What this says is when you're dealing with a white Gaussian noise channel, all signals are equivalent. OK? In other words, it doesn't make any difference what modulation system you use. They all behave the same way.

The only difference between different modulation systems comes in these second order effects, which we haven't started to look at much yet. How hard is it to retrieve carrier? How hard is it to deal with things like time synchronization? How hard is it to build the filters? How hard is it to move from passband down to baseband if you want to do your operations at baseband? All of these questions become important, but the basic question of how to deal with the Gaussian noise, it doesn't make any difference. You can use whatever system you want to, and the analysis of the noise part of the problem is exactly the same at everything. OK? And that's fairly important because it says that in fact, this signal space there we're using-- I mean

yes, we're all used to the fact that it's a vector space. L_2 is a vector space, fine-- what's important here is you're using a finite part of that vector space, and you can deal with it just as if it's finite dimensional vectors. You can deal with vectors and matrices and all of that neat stuff and you can forget about all of the analog stuff.

you look at 99 percent of the papers appear both in the information theory transactions and in the transactions of-- oh I guess it-- it's communication technology. You look at all of these things and 99 percent of the authors don't know anything about analog communication. All they've learned is how to deal with these vectors. OK?

So in fact this is the key to that. And you can now play their game. But also when their game doesn't work you can go back to looking at the analog waveforms. But you now understand what that game is that they're playing. They're assuming white Gaussian noise. They don't have to deal with the fact that the white Gaussian noise is spread over all time and all frequency. Because part of this theorem says it doesn't make any difference how the noise is characterized outside of this region that you're looking at. That's the other part of the argument that we've been dealing with all along. It doesn't matter how you model the noise anywhere outside of what you're looking at. The only thing you need is that the noise is independent outside of there.

OK? So that's-- I mean this was sort of trivial analytically, but it's really an important aspect of what's going on. OK. Let's go back to QAM or PAM. The baseband input to a white Gaussian noise channel-- we're going to model it as u of t -- we're going to look at a succession of j different signals. In other words, when we study detection, we said we're going to build the system, we're going to send one signal, and then we're going to receive that signal and we're going to tear that system down. We're not going to send more than just this one signal.

OK, now we're sending a big batch of signals. We're sending capital J of them. Where J can be as big as you please. Now we're going to look at two different alternatives. OK, one of these alternatives is sending all j signals and building a

receiver which looks at all J of them together and makes a joint decision on everything. It makes a maximum likelihood decision on this sequence of J possible inputs.

This is one of the things we've looked at. This is what happens when you have non binary detection. I mean, here you have an enormous number of things you're detecting between. And you do maximum likelihood detection on it. And the question is, do you get anything extra from that beyond what you get out of doing what we did before, just forgetting about the fact that other signals existed? Namely, which is better?

What the notes prove, and what I'm going to sort of indicate here without any attempt to prove it-- I mean it's not a hard proof, it's just-- I mean the hardest part of this is realizing what the problem is. And the problem is you can detect things in two different ways. You can detect things one signal at a time. Or you can detect them the whole sequence at a time. And you do something different in each of these cases.

OK, so we're going to assume again that these thetas are an orthonormal set. I'm going to assume that I've extended them so they span all of \mathbb{R}^2 . I'm going to let v be a sample of this output vector, v_1 to $v_{\text{sub } j}$. You see, I use the theorem of irrelevance here because I don't care about anything beyond $v_{\text{sub } j}$. Because all those other things are irrelevant. So I only look at v_1 to $v_{\text{sub } j}$.

So the little v is going to be a sample value of this random vector. And the components of the random vector, the output, are going to be the input variables plus the noise variables. And the z_j here are independent. I don't even have to assume that they're Gaussian for this argument. They can be anything at all, so long as they're independent.

OK, now if I'm doing the signal by signal detection, in other words if what I'm doing is I'm saying, "OK, I want to take this little j signal, and I want to decide on what it is just from looking at the v sample of the output." OK, I can do that. My observation is $v_{\text{sub } j}$. That's something we've talked about. I mean the fact that you might have

other information available doesn't make any difference. You still can make a detection on the basis of this one variable.

OK, so we do that. Now, we want to compare that with what happens when we make a detection for all capital J of these inputs, conditional on the whole sequence. OK, you write out the likelihood ratio. It factors. And the thing that happens when you do this-- and you have to read the notes for the details on this, and it's not hard-- what you find is that the maximum likelihood decision is exactly the same in both cases.

OK, so it doesn't make any difference whether you detect little u sub j from the observation v sub j . Or whether you detect the vector, u sub capital j , from the whole set of outputs, v sub 1 up to v sub capital J . You get the same answer in both cases.

Now, you might say, "What happens if some of these likelihood ratios come out to be right on the borderline, equal to one?" Well it doesn't make any difference because that's a zero probability thing. If you want to worry about that, you can worry about it and you get the same answer, you just have to be a little more careful.

Now here is something the notes don't say, and it's also fairly important. Here we're talking about the case where all these signals are independent of each other. Which is sort of what we want to look at in communication because for the most part what we're doing is we're taking data of some sort, we're source processing it to make the bits coming up the channel be independent of each other, and then we're going to be sending them. Well except now we want to say, well suppose that these inputs are not independent. Suppose, for example, that we pass them through an error correction encoding device before transmitting them. And the question is, what happens then?

Well the trouble is then you have dependence between u_1 and u sub j . So you can still detect each of these just by looking at the single output. And it's a maximum likelihood detection based on that observation. But if you look at the entire observation, v_1 up to v sub capital j , you get something better.

How do I know you got something better? Well I know you get something better because in this case where u_1 up to u_j are dependent on each other, these output variables, v_1 up to v_j , depends on each other. They aren't irrelevant. If you want to do the best job of maximum likelihood detection, given the observation of v_1 up to v_j , then you're going to use all those variables in your detection and you're going to get a better --

And you're going to get a better decision. In other words, a smaller error probability, then you would have gotten otherwise. OK? But the important thing that comes out of here is that this simple minded decision-- where you make a decision on u_1 based just on v_1 -- it's something you can do. You can do the maximum likelihood decision. And you know how it behaves. You can calculate what the error probability is. But you know now automatically that your error probability is going to be greater than or equal to the error probability that you would get if you base that decision on the whole set of inputs.

OK, this is an argument that, it seems hard for most people to-- that it seems hard for most people to think of right at the beginning. Which is really to say, if you're doing an optimal detection, it is optimal. In other words, anything else that you do is worse. And you can always count on that to get bound between probabilities of error that you get doing something stupid, and probabilities of error that you get doing something intelligent. And the stupid thing is not always stupid-- I mean the stupid thing is sometimes better because it's cheaper-- but the error probability is always worse there than if you did the actual optimum thing.

OK, so people in fact often do do detection where they have coded systems. They decode each received symbol separately based on the corresponding observation. They wind up with a larger error probability than they should. Then they pass this through some kind of, some kind of error correction device. And they wind up with a system that sort of performs reasonably. And you ask, "Well, would it have performed better if in fact what you did was to wait to make a final decision until you got all the data?"

And we talk a little bit about various kinds of error control later when we get into wireless. We'll see that some kinds of coding systems can behave very easily and can make use of all this extra information. And other ones can't. Algebraic kinds of schemes can't seem to make use of the extra information. And various other kinds of schemes can make use of it. And if you want to understand what's happened in the error correction field over the last five years-- unfortunately 6.451 won't be given, so you won't get all the details of this-- but the simplest one sentence statement you can say is that the world has changed from algebraic coding techniques to probabilistic decoding techniques. And the primary reason for it is you want to make use of all that extra information you get from looking at the full observation, rather than just a partial observations.

OK. Now, back to various signal sets. I put this slide up last time. I want to really talk about it this time because for each number of degrees of freedom, you can define what's called an orthogonal code. And the orthogonal codes for m equals 2 and for m equals 3 are drawn here. For m equals 2, it's something that we've seen before. Namely, if you want to send a zero, you send a one in the first compound, the first degree of freedom, a zero in the second. And if you want to send a one, you send the opposite thing. We've all seen that this isn't a very sensible thing to do when we looked at binary detection, because when you use a scheme like this of course the thing that happens is we now know that we should look at this in terms of just looking at this line along here. Because what we're really transmitting is a pilot tone, so to speak, which is half way in the middle here, which sticks right here. Plus something that varies from that.

So that when we take out this pilot tone, what we wind up with is a one dimensional system instead of a two dimensional system. Which we used to call antipodal communication-- and which everybody with any sense calls antipodal communication, even now-- but in terms of this, it's the simplest case of a simplex code. And a simplex code is simply an orthogonal code where you've taken the mean and moved it out. And as soon as you remove the mean from an orthogonal code, you get rid of one degree freedom, because one of the signals becomes dependent on the others. Which is exactly what's happened here. You've just-- if all

your signals are in one degree of freedom here, when you do the same thing down here, well you get this. And I'll talk about that later.

So, one thing you can do from an orthogonal code is go to a simplex code. The other thing you can do if you want to transmit one more bit out of this signal set is to go to a bi orthogonal code. Which says-- along with transmitting zero, one and one, zero-- you look this and you say, "Gee why don't I also put in zero minus one, minus one zero, and zero minus one down here. Which is exactly what the bi orthogonal code is. The bi orthogonal code simply says take your orthogonal code and every time you have a one, change it into a minus one and get an extra code word out of it.

What's the difference between a set of code words and a signal set? Anybody have any idea? Absolutely none. They're both exactly the same thing. And you think of it as being a code, usually, if what you're doing is thinking of generating an error correcting code and then from that error correcting code you think of using QAM or PAM or something else out beyond that. You call it a signal set if you're just doing the whole thing as one unit. What a lot of systems now do is they start out with a code, then they turn this into an orthogonal signal set.

I'll tell you in other words. A code produces bits. From the bits you group them together into sets of bits. From the sets of bits you go into a signal-- which is, for example something from one of these three possibilities here-- OK. The important thing to notice here-- and it's particularly important to think about it for a few minutes-- is because you do so many exercises. And you've done a number of them already and you will do a few more in this course where you deal with the m equals 2 case. And you can deal with this biorthogonal set here.

You can shift the biorthogonal set around by 45 degrees. In which case it looks like this. OK, so that looks like two PAM sets. It looks like a standard QAM. It looks like standard 4QAM.

This is exactly the same as this of course. When you do detection on this you do

detection by saying, when you transmit this does the noise carry you across that boundary? And then does the noise carry you across this boundary? The noise in this direction is orthogonal from the noise in this direction and therefore, finding the probability of error is very, very simple. Because because you look at two separate orthogonal kinds of noise and you can just multiply these probabilities together in the appropriate way to find out what's happening.

The important thing to have stick in your memory now is that as soon as you go to m equals three, life gets harder. In fact, if you look at this orthogonal set here and you try to find the error probability for it-- you try to find it exactly-- you can't do it like this. You can't just multiply three terms. You look at these regions in three dimensional space. If you want to visualize what they are, what do you do? What picture do you look at? You look at this picture. OK? Because this picture is just this with the mean taken away. So the error probability here is the same as the error probability here.

When I send this point the regions that I'm looking at look like this. And they're not orthogonal to each other. So to find the probability that this point gets outside of this region looks just a little bit messy. And that happens for all m bigger than two. I never knew this because well, I think this is probably a disaster that happens more to teachers than to people working in the field. Because so often I have explained to people how these two dimensional pictures work that I just get used to thinking that this is an easy problem. And in fact when you start looking at the problem for m equals three, then the problem gets much more interesting and much more useful and much more practical when n becomes three or four or five or six or seven or eight. Beyond eight it becomes a little too hard to do. But up until there, it's very easy. OK, so we're going to make use of that in a little bit.

As we said orthogonal codes and simplex codes-- if you scale the simplex code from the orthogonal code-- have exactly the same error probability. In other words, this code here where I've made the distances between the point square root of two over two, which corresponds to the distance between the points here. This and this have exactly the same error probability. So you can even, in fact, find the error

probability for this or for this, whichever you find easier. You now think it's easier to find the error probability for this. I've lead you down the primrose path, because in fact this one is easier to find the error probability for. This one, again, you can find the error probability if you want.

But the energy difference between this and this is simply the added energy that you have to use here to send the mean of these three signals. And one signal is sitting out here at 1,0,0 one is sitting at 0,1,0 one is sitting at 0,0,1. Even I can calculate the mean of those three. It's one third, one third, an one third. So you calculate the energy in one third, one third, and one third, and it's three times one ninth, or one third. Which is exactly what this says. The energy difference between orthogonal and simplex is $1 - \frac{1}{m}$. In other words that's the factor in energy that you lose by using orthogonal codes instead of using simplex codes.

Why do people ever use orthogonal codes? If it's just a pure loss in energy in doing so? Well, one reason is when n gets up to be 6 or 8, it doesn't amounts to a whole lot. And the other reason is if you look at if you look at modulating these things on to sine waves and things like that, you suddenly see that when you're using this it becomes easier to keep frequency lock and phase lock than it does when you use this. I mean you have to think about that argument a little bit to make sense out of it. But that, in fact, is why people often use orthogonal signals because, in fact, they can recover other things from it. Well because they're actually sending a mean, also. So the mean is the thing that lets them recover all these other neat things. So that sort of sometimes rules out this and it sometimes rules out that.

OK. Orthogonal and biorthogonal codes have the same energy? Well, look at them. These two signal points each have the same energy. And these two have the same energy as these two. So the average energy is the same as the energy at each point. So this and this have the same energy.

What happens to the probability of error? Well you can't evaluate it exactly. Except here is a case where just looking at the m equals 2 case gives you the right answer. I mean here to make an error you have to go across that boundary there. Here to

make an error you have to either go across that boundary or go across that boundary. The error of probability essentially goes up by a factor of two. Same thing happens here. You just get twice as many ways to make errors as you had before. And all of these ways to make errors are equally probable. All the points are equally distant from each other. So you essentially just double the number of likely ways you can make errors. And the error probability essentially goes up by two. Goes up a little-- well because you're using a union band it either goes up by a little more or a little less than two-- but it's almost two.

OK, so I want to actually find the probability of error now. If you're sending an orthogonal code. Namely, we pick an orthogonal code where we pick as many code words as we want to. m might be 64, it might be 128, whatever. And we want to try to figure out how to evaluate the probability of error for this kind of code. After you face the fact that, in fact, these lines are not orthogonal to each other.

OK, well the way you do this is you say, OK the, even though this is a slightly messy problem, it's clear from symmetry that you got the same error probability no matter which signal point you sent. Namely, every signal point is exactly the same as every other signal point. What you call the first signal depends only on which you happen to call the first orthonormal direction. Whether it's this, or this, or this. I can change it in anyway I want to and the problem is still exactly the same.

OK, so all I'm going to do is try to find the error probability when I send this signal here. In other words, when I send $1,0,0$ -- actually I'm going to send the square root of e and $0,0$ -- because I want to talk about the energy here. Why am I torturing you with this? Well 50 years ago, 55 years ago, Shannon came out with this marvelous paper which says there's something called channel capacity. And what he said channel capacity was was the minimum rate-- the, was the maximum rate at which you could transmit on a channel and still get zero error probability. Sort of the simplest and most famous case of that is where you have white Gaussian noise to deal with. You're trying to transmit through this white Gaussian noise. And you can use as much bandwidth as you want. Namely, you can spread the signals out as much as you want to.

You can sort of see from starting to look at this picture that you're going to be a little better off, for example, if you want to send one bit in these two dimensions here with orthogonal signals. If you can think of what happens down here for m equals 4, you would wind up with four orthogonal signals. And if you wind up with four orthogonal signals, you're sending two bits, so you're going to use twice as much energy for each of them.

So you can scale each of these up to be $2,0,0,0$; $0,2,0,0$; and so forth. So you're sending twice as much energy. You're filling up more bandwidth because you need more degrees of freedom to send this signal. But who cares? Because we have all the bandwidth we want. Gets more complicated. But the question is, what happens if we go to a very large set of orthogonal signals? And what we're going to find is that when we go to a very large set of orthogonal signals, we can get an error probability which goes to zero as the number of orthogonal signals gets bigger. It goes to zero very fast as we send more bits with each signal. And the place where it goes to zero is exactly channel capacity.

Now in your homework, you're going to work out a simpler version of all of this. And a simpler version is something called the union bound. And in the union bound you just assume that the probability of error when you send this is the sum of the probability of error, of making an error, to this and the possibility of making an error to this. The thing that we're going to add here-- and let me try to explain it from this picture.

I'm sending this point here. OK? And I can find the probability of error over to that point. Which is the probability of going over that threshold. I can talk about the probability of error to over here, which is the probability of going over that threshold. These are not orthogonal to each other. And in fact they have a common component to them. And the common component is what happens in this first direction here.

OK, in other words if you send $1,0,0$ and the noise, and your own noise variable clobbers you. In other words, what you receive is something in this coordinate which is way down here and sort of arbitrary everywhere else-- conditional on the noise

here being very, very large-- you're probably going to make an error. Now if you can imagine having a million orthogonal signals-- and the noise clobbering you on your own noise variable-- you're going to have a million ways to make errors. And they're all going to be kind of likely. If I go far enough down here, suppose what I receive in this coordinate is zero. Then there's a probability of one half that each one of these things is going to be greater than zero.

If I use a union bound, adding up the probabilities of each of these, I'm going to add up a million one halves. Which is 500,000. As an upper bound to a probability. And that's going to clobber my bound pretty badly. Which says the thing I want to do here is, when I'm sending this I want to condition my whole argument on what the noise is in this direction. And given what the noise is in this direction, I will then try to evaluate the error probability, conditional on this. And conditional on a received value here. In fact, the noise in the direction, w_2 , is independent of the noise in direction, w_3 , independent of the noise in direction, w_4 , and so forth. So at that point, condition on w_1 I'm dealing with m minus 1 independent random variables.

I can deal with independent random variables. You can deal with an independent random variables. Maybe some of you can integrate over these complex polycons. I can't. I don't want to. I don't want to write a program that does it. I don't want to be close to anybody who writes a program that does it. It offends me.

OK, so here we go. Where am I? OK, so the first thing I'm going to do, which I didn't tell you, because I'm going to scale the problem a little bit. I did say that here. I'm going to normalize the whole problem by calling my output W sub j instead of Y sub j . And I'm going to normalize it by multiplying Y sub j by the square root of the noise variance.

OK in other words, I'm going to scale the noise down so the noise has unit variance. And by scaling the noise down so it has unit variance, the signal will be scaled down in the same way. So the signal, now, instead of being the square root of e -- which is the energy I have available-- it's going to be the square root $2e$ divided by N_0 . Somehow this thing keeps creeping up everywhere. This $2e$ over N_0 . Well of course

it's the difference between e , which is the energy we have to send the signal, and N_0 over 2, which is the noise energy in each degree of freedom.

So it's not surprising that it's sort of a fundamental quantity. And as soon as we normalize to make the noise variance equal to one, that's what the signal is. So I'm going to call α this so I don't have to write this all the time. Because it gets kind of messy on the slides.

OK so given that I'm going to send input one, the received variable W_1 is going to be normal with a mean α which is the square root of $2e$ over N_0 and with a variance of one. All the other random variables are going to be normal random variables. Mean zero, variance one. OK? I'm going to make an error if any one of these other random variables happens to rise up and exceed W_1 .

So the thing we have here is W_1 is doing some crazy thing. I have this enormous sea of other code words in other directions. And then the question we ask is can the noise-- which is usually very small all over the place but it might rise up some place. And if it rises up someplace, we're asking what's the probability that it's going to rise up to be big enough that it's exceeds W_1 ?

So, I can at least write down what the error probability is exactly at that point. What I'm going to do is I'm going to write down the probability density for W_1 . And W_1 , remember, is a Gaussian random variable with mean α and with variance one. So we could actually write down what that density is. And for each W_1 I'm going to make an error if the union of any one of these events occurs. Namely, if any one of the $W_{sub j}$ is bigger than W_1 , bingo I'm lost. OK? So I'm going to integrate that now over all values of W_1 .

Now, as I said before if W_1 is very small, then lots of other signals look more likely than W_1 . In other words I'm going to get clobbered no matter what I do. Whereas if W_1 is large, it looks like the union bound might work here. And the union bound is what you're doing in the homework without paying any attention to W_1 . OK, so I'm going to use the union bound-- and the union bound says the probability that this union of all these different $m - 1$ events-- exceeds some value, W_1 . W_1 is just

some arbitrary constant in here. The probability of this. I'm going to write it in two ways. It's upper bounded by one. Because all probabilities are upper bounded by one. This is not a probability density, this is a probability now so it's upper bounded by one.

And it's also upper bounded by this union bound. Mainly, the union of $m - 1$ events. Each with probability. This is the tail of the Gaussian distribution evaluated at w_1 . So I have the union bound here. I have one here. Going to pick some parameter γ . I don't know what γ should be yet, but whatever I pick γ to be I still have a legitimate bound. This is less than or equal to this, and it's less than or equal to this. Just common sense dictates that I'm going to use this when this is less than one. And I'm going to use this when this is greater than one.

That's what common sense says. As soon as I start to evaluate this, common sense goes out the window because then I start to deal with setting this quantity equal to one and dealing with the inverse of the Gaussian distribution function. Which is very painful. So I'm then going to bound what this is. And in terms of the bound on this, I'm going to affect γ that way. Because all I'm interested in is an upper bound on error probability anyway. OK, so that's where we're going to go.

So this probability of error then is then exceeded by using this bound for small w_1 . I get this integral over all w_1 between minus infinity and γ . And let's just look at what this becomes. This is just the tail of the Gaussian distribution. That's the lower tail instead of the upper tail, but that doesn't make any difference. So this is exactly Q of $\alpha - \gamma$. Mainly, α is where the mean of this density is and γ is where I integrate to. So if I shift the thing down, I get something that goes from-- well you might be surprised as to why this is minus γ instead of plus γ , and it's minus γ because I'm looking at the lower tail rather than the upper tail and asking you to think this through in real time is unreasonable, but believe me if you sit down and think about it for a couple of seconds you'll realize that this integral is exactly this lowered tail of this Gaussian distribution.

The other term is a little more complicated. It's $m - 1$, which is that term there,

times Q of w_1 , which is this term here, times the Gaussian density. Now this is the Gaussian density for W_1 , which is one over square root of 2π . This is a normalized invariance, but it has a mean of α , so it's this.

OK, well next thing we have to do is either fiddle around like mad or look at this. If you remember one of the things that you did-- I think in the previous homework that you passed in-- you found the bound on Q . Which looks like this. OK. That's just the tail of the Gaussian distribution. And the tail of the Gaussian distribution is upper banded by this for W_1 greater than or equal to zero. It's upper bounded by a bunch of other things which you find in this problem. The other bounds are tighter. This is the most useful bound to the Gaussian distribution that there is. Because it works for W greater than or equal to zero. And it's exact when W_1 is equal to zero. Because you're just integrating over half of the Gaussian density. And it's convenient and easy to work with.

But what that says is the Q of W_1 , when W_1 is anything greater than or equal to zero, looks very much like a Gaussian density. So the thing that you're doing here is taking one Gaussian density and you're multiplying it by another Gaussian density. And the one Gaussian density is sitting here looking like this with some scale factor on it at zero. The other Gaussian density is out here at α -- was α , wasn't it? Or was it γ ? Can't keep my γ s and α s straight-- OK, and it looks like this. If you take the product of two Gaussian densities of the same amplitude and everything. And the same variance. What do you wind up with?

Well you can go through and complete the square. And you can sort of see from looking at this from the symmetry of it that what you're going to get is the Gaussian. Which is right there. α over 2. OK so these two things, when you multiply them, look like this. This has the same variance as these two things do. But it's centered on α over 2. If you want to take the Fourier transform and multiply the Fourier transform. I mean, you take the Fourier transform of the Gaussian and you got a Gaussian.

So here we're multiplying. Up there in Fourier transform space you're convolving.

And when you convolve a Gaussian with a Gaussian you got a Gaussian. When you multiply Gaussian by a Gaussian you got a Gaussian. When you do anything to a Gaussian, you got a Gaussian. OK? So this thing-- and if you don't believe me just actually take these two exponents and complete the square and see what you get-- so the mean is going to be $\alpha/2$.

So, here you have a term which is one tail of the Gaussian distribution centered at $\alpha - \gamma$. And here you have another one centered at $\alpha/2$. When you see these two terms, something clicks in your mind and says that sometimes this term is going to be the significant term. Sometimes this term is going to be significant term. And it depends on whether $\gamma - \alpha/2$ is greater than or less than $\alpha/2$. So you can sort of see what's going to happen right away.

Well I hope you can see what's going to happen right away, because I'm not going to torture you with any more of this. And you can look at the notes to find the details. But you now sort of see what's happening. Because you have a sum of two terms. We're trying to upper bound this. We don't much care about factors of two, or factors of square root of 2π or anything. We're trying to look at when this goes to zero. When you make m bigger and bigger. And when it doesn't go to zero.

So when we do this, the probability of error is going to be less than or equal to either of these two terms. And here are these two alternatives that we spoke about. Namely when $\alpha/2$ is less than or equal to γ , we get this. When $\alpha/2$ is greater than γ , we get this. And this involves choosing γ in the right way, so that that union bound is about equal to one at γ . OK, well that doesn't tell us anything. So we say, "OK, what we're really interested in here is, as m is getting bigger and bigger, we're spending a certain amount of energy per input bit. And that's what we're interested in as far as Shannon's theorem is concerned. How much energy do you spend to send a bit through this channel?"

And this gives you the answer to that question. You let $\log m$ equal to b . m is the size of the signal alphabet, so b is the number of bits you're sending. So E_b , namely

the energy per bit, is just the total energy in these orthogonal waveforms divided by b . So that's the energy per bit. OK, you substitute these two things into that equation and what you get is these two different terms. You got E to the minus b times this junk. And E to the minus b times this junk. What happens when m gets big? When m gets big, holding E_b fixed, so the game is we're going to keep doubling our orthogonal set, being able to transmit one more bit. And every time we transmit one more bit, we get a little more energy that we can use to transmit that one extra bit.

So we used an orthogonal set, but using a little more energy in that orthogonal set. So what this says is that the probability of error goes down exponentially with b , if either one these terms are positive. And this, and looking at the biggest of these terms tells you which one you want to look at.

So, anytime that E_b over $4N_0$ is less than or equal to $\log n$ is less than E_b over N_0 , you get this term. Any time E_b over $4N_0$ is greater than the natural log of 2, you get this term. Now when you go through the union bound in your homework, I'll give you a clue. This is the answer you ought to come up with when you're all done. Because that's what the union bound tells you.

Here, remember we did something more sophisticated the union bound because we said the union bound is lousy when you get a lot of noise on W_1 . And therefore we did something separate for that case. And what we're finding now is depending on how much energy we're using, namely depending on whether we're trying to get very, very close to channel capacity or not. If you're trying to get very close to channel capacity you've got to use this answer here. Which comes from here. And in this case, this says that the probability of error goes to zero exponentially in the number of bits you're putting into this orthogonal code, at this rate. E_b over $2N_0$ minus $\log 2$. Which is positive if E_b over $4N_0$, well--

I'm sorry. If you can, if you can trace back about three minutes, just reverse everything I said about this and about this. Somehow I wrote these inequalities in the wrong way and it sort of confused me. This is the answer you're going to get from the union bound. This is the answer that we get now because we use this

more sophisticated way of looking at it.

Sob. No. Erase what I just said in the last thirty seconds and go back to what I said before that. This is the answer you're going to get in the homework. This is the answer that we want to look at. This thing goes-- is valid-- anytime that E_b over N_0 is greater than or equal-- is greater than natural log of 2. When E_b over N_0 is equal to log 2, that's the capacity of this channel. Namely E_b equals N_0 log 2. Well better to say it E_b over N_0 equals natural log of 2 is capacity.

And anytime E_b over N_0 is greater than log 2 this term in here is positive. The error probability goes down exponentially as b gets large and eventually goes to zero. It doesn't get down as fast here as it does here. But this is where you really want to be because this is where you're transmitting with absolutely the smallest amount of energy possible. OK, so that's Shannon's formula. And at least we have caught up to 50 years behind what's going on in communication. And actually shown you something that's right there. And in fact, what we went through today is really the essence of that, of the proof of the channel capacity theorem. If you want to do it for finite bandwidth it gets much harder. But for this case, we really did it. It's all there. I mean you read the notes and in a little extra detail. But just with the grunge work left out, that's what's going on.

OK, wireless communication. That's what we want to spend the rest of the term on. We want to spend the rest of the term on that because whether you realize it or not, we sort of said everything there is to say about white Gaussian noise. And when all of that sinks in, what you're left with is the idea the white Gaussian noise. You're really dealing with just a finite vector problem. And you don't have to worry about anything else. The QAM and the PAM, all that stuff, all disappear. Doesn't matter whether you send things broadband, narrow band, whatever. It's all the same answer.

Wireless is different. Wireless is different for a couple of reasons. You're dealing with the radiation between antennas because you're dealing with the radiation between antennas rather than what's going on on a wire. I mean, what's going on a

wire, the wire is pretty much shielded from the outside world. So you send something, noise gets added, and you receive signal plus noise. There's not much fading, there's not much awkward stuff going on. Here all of the stuff goes on.

As soon as you start using wireless communication, you're allowed to drive around in your car talking on two phones at once. With your ears and eyes shielded. You can kill yourself much easier that way than you can with ordinary telephony. You can be in constant communications with almost anyone. OK, so you have motion, you have temporary locations. You have all these neat things. And if you look at what's happening in the world, the less developed parts of the world have much more mobile communication than we do. Because in fact they don't have that much wire communication. It's not that good there. So they find it's far cheaper to get a mobile phone. Than like us where we have to pay for both a wire line phone and a mobile phone.

So they have sort of the best of the two worlds there. Except their mobile phones are like our mobile phones. They only work three quarters of the time. And all of the research that's going into sending video over wireless phones, it seems that nobody's spending any time trying to increase the amount of time you can use your wireless phone from 75% to 90%. And if any of you want to make a lot of money and also do something worthwhile for the world, invent a wireless phone the works 90% of the time. And you'll clean up, believe me. And you can even send video on it later if you want to.

OK that's another thing that wireless has turned out to be very useful for. And I'm sure you all know this. It avoids mazes of wires. I mean many people in their homes and offices and everywhere are starting to use local area wireless networks just as a way of getting rid of all of these maddening wires that we have running all over the place. As soon as we have a computer and a printer and a fax machine and a blah blah blah, and a watch which is connected to our, and a toaster which is connected to our computer. Argh! Pretty soon we're going to be connected to our computers. We're going to have little things stuck in our head and stuck our neck and all over the place. So it'll be nice to have these, it'll be nice to not have wires when we're

doing that.

OK, but the new problem is that the channel, in fact, changes with time. It's very different from one time to another. And you get a lot of interference between channels. In other words, when you're dealing with wireless you cannot think of just one transmitter and one receiver anymore. That's one of the problems you want to think about. But you really have to think about what all the other transmitters are doing and what all the other receivers are doing.

It was started by Marconi in 1897. It took him about three years to get transcontinental communication. I mean we think we're so great now-- being able to have research move as quickly as it does-- but if you think of the amount of time it takes to create a new wireless system, it's a whole lot larger now than it was then. I mean he moved very fast. I mean the technology was very primitive and very simple. It was not a billion dollar business. But in fact it was very, very rapid. But what's happened since, with wireless, has been very fitful.

Businesses have started. Businesses have stopped. People have tried to do one thing. People have tried to do another thing. They name things by something different all the time. I mean one sort of amusing thing is back in the early seventies the army was trying very hard to get wireless communication in the field. And they called this packet radio. And they had all the universities in the country spending enormous amounts of time developing packet radio. Writing many papers about it. They finally got disgusted because nothing was happening. So they pulled all the funding for that.

And about five years later, when the people at DARPA and NSF and all of that forgot about this unpleasant experience, people started talking about ad hoc networks. Guess what an ad hoc network is? Same thing as packet radio. Just a new name for an old system, and suddenly the money started flowing in again. We don't know whether it'll be any better this time than it was last time. But anyway that's the way funding goes.

OK, what we're going to talk about in this class is sort of an old fashioned thing. It's

not as sexy as what all these other systems are. It's just cellular networks. It's probably because that's well understood by now, and it's because we can talk about all of these fundamental problems that occur in mobile communication just in the context of this one kind of system that, by now, is reasonably well understood. When you're doing cellular communication, you wind up with a large bunch of mobiles all communicating with one base station. OK, in other words you don't have the kind of thing you had in the packet radio network or in the ad hoc network, where you have a huge number of mobile telephones which are all communicating to each other. And where one phone has to relay things for others. You wind up with a very complicated network problem.

Here, it's in a sense, a much simpler and more sane structure. Because you're using mobile for doing the things that mobile does well. And you're using wires for the things that wires do very well. Mainly you have lots of mobiles which are moving all over the place. You have these fixed base stations which are big and expensive, and put up on hills or on buildings or on big poles or something. And you spend a lot of money on them. You have optical fibers or cables or what have you running between them or running from them to what's called a MTSO. Mobile Something Subscriber Office-- and I can never remember what those letters stand for-- Mobile Telephone Subscriber Office. All I had to do to remember that was thank this was done by telephone engineers. No, Mobile Telephone Switching Office and telephone engineers think in terms of switching and in terms of telephones. And mobile and offices just follows along.

So the way these systems work is you go from a mobile to a base station. From the base station to one of these MTSOs, which is just a big switching center. From there you're in the wired network. And from there you can either go back to a mobile or go back to a wire line telephone or go anywhere you want to. But but the point in that, and I think this is important to remember, is that cellular networks are an appendage of the wire line network. And you always have this wire line network in the middle. You probably always will. Because wire line networks have things like fiber which carries enormous amounts of data very, very cheaply. And mobile is very limited as far as capacity goes. And it's very noisy.

OK, so that lets us avoid the question of how do you do relaying. When you see pictures of this, people draw pictures of hexagon cells.

AUDIENCE: [UNINTELLIGIBLE] turn around.

PROFESSOR: Oh, when I hit this, and it uh, OK. I mean there's not much information on this picture anyway but, [LAUGHTER] OK, but people think in terms of base stations put down uniformly with nice hexagons around them. And any time a mobile within one hexagon it communicates with the base station which is at the center of that hexagon. And in reality what happens is that the base stations are spread all over the place in a very haphazard way. I shouldn't say haphazard, because people worked very hard to find places to put these base stations. Because you need to rent real estate, or buy real estate to put them in. You have to find out somehow what kind of coverage they have. And it's a very fascinating and very difficult problem.

One thing I'm going to try to convince you of in the next lecture or so is that the problems of choosing base stations are very heavily electromagnetic in nature. You really have to understand electromagnetism very well. And and you have to understand the modeling of these physical communication links very, very well in order to try to sort out where base stations should go and where they shouldn't go. The other part of the problem is the part of the problem dealing with how do you design the mobile phone itself? How do you design the base station itself? And these are questions which don't depend so much on the exact modeling of the electromagnetic channel. They only depend on very coarse characteristics of it.

And very often, when you start to study mobile, you will spend an inordinate amount of time studying all of the details of these electromagnetic channels. Which in fact are very important as far as choosing base stations are concerned. And have relatively little to do with the questions of how do you design mobiles. How do you design base stations? It has enough to do with it that you have to know something about it, but it's not central anymore.

OK, so let's look at what the problems are. As I said the cellular network is really an appendage to the wire network. The problems we're going to have to deal with is when you're outgoing from your own cell phone, there's some kind of strategy that has to be used for you to find the best base station to use. And it's a difficult question because you're trying to find a base station you can communicate with and one that's not so overcrowded that you can't talk to it. So that's one big problem. We won't talk about that much. Another is the ingoing problem. Finding a mobile. If you think about that, it's really a very tricky problem. Because I run around in my car with my cellphone turned off all the time. And I only turn it on if I want to talk to somebody. So I turn it on and somehow the whole network has to suddenly realize where I am. And you know that happens with all of these cellular networks all over the place. And every time somebody turns on their cell phone there's a lot of stuff going back and forth that says who is this guy? Does he have the right to talk? Has he paid his bill? And how do I actually find a base station for him to use?

So this is kind of, both these questions are kind of difficult. And the even worse question is if somebody's calling me and I live say, in Boston, or close to Boston, and I'm out in San Francisco and somebody calls me on my cell phone, the call gets to me. And if you just imagine a little bit what has to go on in this cellular network in order for the cellular network to realize that I'm in San Francisco instead of Boston. And then realize how to get calls to me in San Francisco. I mean there's a lot of interesting stuff going on here.

But we're not going to talk about any of that because that's really sort of an organizational question as opposed to a physical communication question, which is the kind of thing we're interested in here. OK, when you have these multiple mobiles which are sending to the same base station. People who are working on mobile communication, sort of the practical side of it, call this the reverse channel. Why they call this the reverse channel and the other one the forward channel, I don't know. Forward channel goes from the base station to the mobile. Reverse station, reverse channel goes from the mobile to the base station. And what it says is the terminology was chosen by the people designing the base stations. That's sort of clear. But if you read about this in any more technical publication, you will see this

thing being called a multi access channel. It's the multi access channel because many, many users are all trying to get into the same base station. And this one electromagnetic wave-- which is impinging on the various space station antennas-- is carrying all of that stuff all multiplexed together in some way. And it's not multiplexed together in a sensible way, because it's multiplexed together just by all of these waveforms randomly adding to each other. So information theorists call these things multi access channels.

When you're going the other way, base station to mobiles, it's called the forward channel by the telephone engineers. It's called the broadcast channel by information theorists. For those of you who think about broadcast in terms of TV and FM and all that sort of stuff, this is a little bit confusing. Because this is not the same kind of broadcast that you're usually thinking about. I mean the usual kind of broadcast is where everybody gets the same thing whether you want it or not. But that whole signal is there, and you all get the whole thing.

Here what it is, is you're sending a different message to everyone. You don't want everyone to be able to tune in and receive what anybody else is getting. You want a little privacy here. So it's really broadcasting separate messages and trying to keep them separate. While the systems, almost all of them, are now digital I think. In the sense of having a binary interface, this is the same issue we've been talking about all along. You say something is digital if there's the binary interface on it. The source is either analog or digital. Cellular communication was really designed for voice. Now all the research is concerned with how do you make it work for data also. One of the things we're going to talk about a little bit is why the problems are so very different. I mean you would think they're both the same problem. Because in both cases you're just transmitting a string of bits. That's all that's going on. But the big difference is that in voice, you can't tolerate delay in voice. In data you can tolerate delay. You can tolerate a lot of delay in data. And therefore you can do lots of things with data that you can't do with voice. If you want to have a system that deals with both voice and data, it's got to be able to get the voice through without delay. And you have to find some way of solving that problem.

OK, let me just say this quickly. Let me just see if there's anything else of content here. This is all just boiler plate stuff. Let me skip that. And skip this. The thing we're going to be concerned about here is really these physical modeling issues. And where we wind up with that is we're typically talking about bandwidths that are maybe a few megahertz wide. Maybe a few kilohertz wide. Or maybe a few megahertz. But we're talking about carrier frequencies which are usually up in the gigahertz range. And they keep varying depending on which new range of frequencies gets opened up. They started out a little below a gigahertz. They only went up to 2.4 and now they're up around five or six. And things like that. When we talk about physical modeling, we want to understand what difference it makes what carrier frequency you're at.

And it does make a difference because we'll talk about Doppler shift. And Doppler shift changes a lot as you go from one range to another. But for the most part these systems are narrow band. There's a lot of work now on wide band systems. And what's does a wide band mean? Does it mean more than a megahertz? No, it means a system where the bandwidth that you're communicating over is a significant fraction of the carrier frequency. If there is a carrier frequency. Many of these wide band systems are not even done in terms of the carrier frequency. They're just done in terms of an arbitrary waveform which takes over an enormous amount of bandwidth.

If you're dealing with the narrow band problems, white Gaussian noise is a good assumption for the noise. But now, along with the noise you have all these other effects. You have a channel, where the channel is not just a pass through wire with a little attenuation on it. I mean, remember what we've done all along. We have absolutely ignored the question of attenuation. We've just gotten rid of it and say what you send is what you receive. We've gotten rid of the problem of filtering on the channel. We've said a little bit about it, but essentially we've avoided it. Now the problem that you have is this channel that you're transmitting over really comes and goes. Sometimes it's there, sometimes it's not. So it's a time varying channel. It's a time varying channel which depends on the frequency band that we're using.

And one of the things that we have to talk about in order to come to grips with this is questions about how quickly does it change and why does it change. And how much do you have to change the frequency before you got something that looks like an independently different channel? So we have to deal with both of those and we're going to do that next time. And in trying to come to grips with these questions, the first thing we're going to do is to look at very, very idealized models of what goes on in communication. Like, we're going to look at a point source radiating outwards. We're going to look at a point source radiating outwards, hitting a barrier, and coming back.

Interesting problem to look at and you ought to read the notes about this. What happens when you're in a car and you're driving at 60 miles an hour towards the reflecting wall. And right before you hit the wall, what's the communication look like? OK, that's a very neat and very simple problem. You can't do it many times, but we will talk about that next time.