

PROFESSOR: So many of you have signed up this term. As most of you know, this course is now on an alternate year basis, but it really depends on how much interest there is in it. So this is certainly very satisfactory interest.

This morning I'm just going to go through the information sheet logistics a little, tell you a little bit about the course, and then go through a very rapid path through the first three chapters. So hang onto your seats.

But first of all, the information. One thing I did not mention in the information sheet is that due to the sponsorship of OpenCourseWare and the department, we are going to have this class on TV or videotaped for OCW in the department archives. And I hope that's OK with everybody. I think at some point we'll send out a permission sheet. Does anyone anticipate having any problem with the camera lingering on them or the back of them from time to time? I don't think this is going to be an issue. But if it is, either mention it now or come up and see me privately. As I say, I think eventually you'll have to sign permission sheets.

Our cameraman is Tom White. Let's give a hand to Tom. And he promises to be as unobtrusive as possible.

OK. There are four handouts this morning. I hope you have them all. First of all, the information sheet. Second, a little personal data sheet that I ask you to each fill out so we know who's in the class and a little bit about you. The first problem set, which is due a week from today. In general, problem sets will be due on Wednesdays and handed out on Wednesdays. And please do pass them in on time. We won't make any serious attempt to grade them if they don't come in on time. I'll mention our philosophy about problem sets in a little while. And then chapters one through three, which are all sort of warm-up chapters. And problem set one is this kind of warm-up exercises to get you into context, to get you into the mood for this course. It doesn't have to do -- well, it does. But at this point, it's more just getting up to speed with the language and the environment.

The teaching assistant is Ashish Khisti. Ashish, why don't you stand up just so everybody is sure to see you. I'm delighted to have Ashish. He was a superb student this course. Certainly understands it very well. Ashish will even be a distinguished guest lecturer for lectures two through four, because I will be on vacation during that time. So you'll get to know Ashish very well. I think he will be able to help you quite a bit.

His office hours we've tentatively scheduled for Tuesday from five to seven in the basement area of Building 32, the Stata Center. Is that going to be OK with everybody? Is there anybody who's --? It's of course set that way because the homeworks are due on Wednesday. That's all right? All right. We aim to please.

I've already mentioned Tom White.

I want to mention that we've tentatively scheduled the midterm for Wednesday, March 16, which is the Wednesday before MIT vacation week. And normally I've run it for two hours to try to minimize time pressure. For 9:30 class, that's easy. We simply start it at nine. But again, if anyone has any personal problem with that, please let me know. You may not know about that yet. But I want to let you know immediately.

Prerequisite. I've held to the policy of enforcing the prerequisite, which is 6.450, rigorously. Not so much because the content is really what's needed. This course is entitled Principles of Digital Communications two. But what it really is is a course on coding, modern coding, in a communications wrapper. Our motivation all along will be how to get to capacity on the additive white Gaussian noise channel. Within that story, we're going to develop all the principal classes of codes that you would see in any coding course. But we're always going to be measuring them by how well they do on a particular communications channel, the canonical additive white Gaussian noise channel.

But in fact, we're not going to use very much of what you had in 6.450. The point of having taken 6.450 is simply to be up to a certain level of speed and sophistication in dealing with rather mathematical concepts. As you know, 450 is rather

mathematical. Most of the Area 1 courses are rather mathematical. This one is rather mathematical. And if you simply aren't used to making logical arguments fairly quickly and in a snap, you're going to have trouble with this course.

So that's really the reason for a prerequisite. And I have found it's important to enforce it, because we always get, each year, a couple of people who probably shouldn't be here. So is there anyone here who hasn't taken a version of 450 before? Yes? What's your name?

AUDIENCE: Mesrob. I emailed you.

PROFESSOR: Oh. So you emailed me -- Mesrob? OK. And you had taken a couple of other very rigorous courses, and so I'm happy with your prerequisites.

The registrar told me that Mukul Agarwal and Daniel Wang, that's you, and Andrew Cross have not had the prerequisites. Daniel? What's the story?

AUDIENCE: I audited [INAUDIBLE].

PROFESSOR: You audited 450? OK. Perhaps we could talk after class about, in general, what courses you've taken, and why you think you can keep up it.

AUDIENCE: [INAUDIBLE].

PROFESSOR: OK. Or just let me know how you prefer to handle it. We can do that immediately after class in a corner of the classroom, if you want. And are the other two people here? Agarwal? No? Cross? So they seem to have selected themselves out. All right. Very good. And everyone else here has had 450.

All righty. There's no text. Unfortunately there's really isn't any text that comes close to covering all the material we're going to cover. Many of the individual chapters in this course are covered by various textbooks, of course, in far greater depth than I am able to cover them here. So partly what you're getting from me is a selection of a thread of what you really need to know, at least first time through. And over the years, I've developed course notes of my own, which I'm now reasonably satisfied with. At least up through the last chapters, where we'll spend a little bit more time

this year. And so I hope that that will be satisfactory.

But I do give you some supplementary texts in here, and more are coming out all the time. And part of being a graduate student is understanding where to find supplementary material if you need to. So let me know if you ever feel -- if you ever want to know, where can I read more about something, I'd be happy to give you a suggestion.

Office hours I already mentioned. Problem sets. They'll be weekly, except before the exam and except in the last week to the course. The purpose of the problem sets is for you to get practice in getting up to speed. If you don't use them that way, you're not using them correctly. I absolutely don't care if you do them together or whatever method is most effective for you to learn. These are intended for you to learn.

I don't recommend going in -- many of these of have been offered in prior years, and the answers are to be found somewhere on the net or in somebody's library. But as you've heard in every other class, I don't recommend your going about them that way. Not much weight is put on them in the grading. The TA or the grader will simply put a grade of zero, one, or two. We'll be happy to discuss any of them where you feel you didn't really get it, or don't know how to get your arms around a particular problem.

And in fact, I wouldn't put any weight on the homeworks, except that every year students say, well, if you put 0% on the problem sets, then we won't do them. And so I put 15% on the problem sets. But in fact the problem sets -- we do get an idea of how engaged you are in the course and how much you're getting it from the problem sets. That's important feedback, both generally for the class and individually for each of you. You know, there have been cases where somebody is just copying the problem set from the previous year every week, and then we can pretty well predict that they'll bomb out on the midterm, and they do.

So anyway. You've been at MIT a while. This is pretty much the standard philosophy, I think, at least in Area 1. Any questions about what I just said? No?

The midterm. Scheduled for two hours in the Wednesday, the last class before the vacation. Counts for 1/3. The final is scheduled during finals week. I don't have control over that. It will count for 1/2. Basically, the grade comes from adding up your scores on those two things. Doing a scatter chart and trying to make some intelligent guess. Usually we get a little more than half A's, a little less than half B's. But I don't have any fixed number in mind for that. It's really, I try to do it based on how you do, using pluses and minuses for decoration. And I don't know. Again, I think it's like what you see in every other course.

All right. The topics. What's going to be different about 6.451 this year from 2003, which is the last time it was offered for previous years? Not too much. Basically, this course -- originally there was a single course, which covered 6.450 and 451. It evolved, got split into two courses. And this course has pretty much become a coding course. So you are here because you're interested in coding, in particular for communications. This is a good first course in coding for, I think, other interests.

Are you Andrew Cross? Yes. What is your situation on the prerequisites?

AUDIENCE: [INAUDIBLE].

PROFESSOR: OK. Let's talk about it afterwards.

Alright. So as I said earlier, the course is pretty much stabilized at this point to be a course on coding over the additive white Gaussian channel. Those of you who are here with other interests, that's not the only place that coding is used, I think will be reasonably satisfied with the course. We're always putting it in a communication context, but we talk about all the major classes of algebraic block codes, convolutional codes, tail-biting codes, capacity approaching codes, low-density parity-check codes, lattice codes, trellis coded modulation, if you want to get ... and all of their decoding algorithms, which are just as important as the codes themselves.

So I hope that's what you're after, is to get a view of which codes have proved to be

most important over the years. My context is communication, but I believe you'll get exposed to most of the things you would need, regardless of why you happen to be interested in coding.

What is going to be different this year is I'm going to be rather ruthlessly chopping out not perfectly essential topics wherever I can through the course. And in particular, I'm going to run through chapters one through three today, assuming that -- well, one is just introduction. Chapter two is basically how you get from continuous time to discrete time on additive white Gaussian noise channels. You did that fairly well in 6.450, so I assume all I need to do is basically sketch it for you. And chapter three is proof of the channel capacity formula, which you may or may not have encountered somewhere else. But even if you haven't encountered it, you're probably reasonably willing to take it on faith. And so that allows us to get immediately into chapter four in the next lecture, where Ashish will be starting to work up from the smallest, simplest little constellations in Euclidean space, which is where the story begins.

This hopefully will give us more time at the end of the course for more on what's been happening in the last ten years or so. First of all, a much more analytical treatment of capacity approaching codes than I've been able to give in the past. The real theory. How do you design them? How do they perform? Some more details. Enough so that you could go ahead and implement them, I believe, and analyze them, and optimize them. And that's, I think, really important in this course. Not just because that's the way people are doing coding these days, but also because that's the end of the story. After 50 years, we finally did get to channel capacity, and this was the way we got there.

In addition, if we really do well, we'll be able to do at least a week, maybe two weeks, on codes for band limited channels, where you need to go beyond binary. You need to send codes, at least symbols, that have many levels on them, not binary levels, in order to send it at high spec efficiencies, many bits per second per Hertz, which is, of course, what you want for most wireless or wireline channels. You are able to send lots of bits per second per Hertz. How do we do that? Again,

hopefully getting to capacity.

And finally, I always offer the teaser, and for at least five years, I haven't been able to get there. We could talk about linear Gaussian channels. Where you get into equalization, precoding. Other topics that are very important. But I'm almost certain we won't get there.

And in addition, this subject, this kind of a scalar version of what's done to a fare-thee-well in the matrix way, in the wireless course, 6.452, which is taught in the next classroom in the next hour and a half. So I anticipate in future the whole equalization of precoding -- intellectually, it should be merged with Wireless. Because they do matrix versions of all of that. On the other hand, it forms a part of this story, which is just basically point-to-point transmission over the additive white Gaussian noise channel.

All right. I very much encourage questions and feedback, and I'm always willing to be distracted, tell stories, respond to just what's on your mind. It's much more fun for me, and probably helps you, too. Does anybody have any questions at this point, or observations? No? All right. We should just get into it.

OK. So I say I always like to be thinking of this from a communications design point of view. We are engineers. I am a communications engineer and theorist. And what's the design problem that we're really trying to solve in this course? The design problem is that somebody -- your boss, or the FCC, or somebody -- gives you a channel, which usually involves some set of frequencies and ... over some physical medium and says, OK. I want you to design a communications device, a modem, to get as much digital information over that channel as you can.

So in chapter one, we talk about a couple of channels which have been very important for the development of coding. One, for instance, the deep space channel. The deep space channel -- what's the problem? You have a highly power-limited satellite way out there. It's got a little tiny antenna. It can use whatever frequencies it wants. It's talking to a huge 140-foot dish or whatever somewhere in Pasadena or somewhere else around the world. And that's basically the channel.

And you can send any wave form you'd like, subject, really, to power limitation.

This is an extreme example of a power-limited channel. How do we characterize such a channel? Well, frequencies are unlimited, but the signal-to-noise ratio is very poor. You get a very tiny signal, and you get noise in the antenna front end, which they've done everything they can reduce it, they've cooled it to a few degrees above zero and so forth. Nonetheless, the front end noise -- in the very first stages of amplification are what is the noise. And you've basically got to transmit through that.

So you have a pure additive white Gaussian noise channel, which we we'll always write as the output of the channel Y of t is the input, X of t , plus the noise, N of t . Which is additive, independent of the signal, and white. The spectrum can be taken to be flat.

All right. And what else do we have? We have that this has some average power P , and this has some noise power spectral density which is characterized by a parameter N_0 , which is basically the noise power per Hertz. Per positive frequently Hertz, as it turns out, because these things were all defined back in the dark ages.

OK. And we also, either because of the channel specification or because ultimately, we decide to signal in a particular bandwidth, we have a bandwidth W . Again, that's just measured over the positive frequencies. So if it goes from zero to W , that's a baseband channel. If it goes from W_0 to W_0 plus W , that's a passband channel. But we're going to find out it doesn't really matter where the passband is, or the baseband.

So those are three parameters that specify an additive white Gaussian noise channel. And we can aggregate them into a signal noise ratio. A signal-to-noise ratio is the ratio of a signal power to a noise power. In this case, the signal power, by definition, is P . The noise power, it's N_0 per Hertz. We're going across W Hertz. So by the way we define N_0 , the signal-to-noise ratio is just P over $N_0 W$. That's the amount of noise power in the band.

And there's an implicit assumption here that only the noise in the band concerns

you, and it's one of the key results of 6.450 or equivalent courses that out-of-band noise is irrelevant, because you know information about what's in-band, so you may as well filter it out. And therefore we don't have to count any noise power that's outside the signal transmission band or the signal space, as we would say.

So two key things. It's easier to keep in mind the signal-to-noise ratio than both P and N_0 . By an amplifier, you could scale both P and N_0 so really all you're interested in is the ratio anyway. So we have these two key parameters for a additive white Gaussian noise channel.

All right. A more typical kind of design problem is, say, a -- well, the telephone line modem, which is something on which I've spent a good part of my career. Your boss says, OK. I want you to build me a modem that sends as much data as possible over telephone lines. How would you approach that problem? OK? This is your summer project. What would you do on the first day?

I really would like a little feedback, folks. Makes it so much more fun for me. Yeah?

AUDIENCE: Can we just tell him he's [INAUDIBLE] frequency -- we have [INAUDIBLE] frequency, so it's not like this time it would [INAUDIBLE].

PROFESSOR: All right. So you have in mind a certain channel model. How would you verify that channel model?

AUDIENCE: Well, you put the channel frequency response, you could measure it somehow. You could estimate it.

PROFESSOR: Yeah. You could take a sine wave generator and run it across the frequencies, and see how much gets through at the receiver. And you could plot some kind of a channel response, which for a particular telephone channel might go something like that.

Turns out telephone channels are designed so that they have no transmission above 4 kilohertz, because you sample them 8000 times a second inside the network. They're also designed to have no DC response.

So this might go from -- this is -- well, I'm just drawing some generic response. And dB, I'll give you my dB lecture later. 300 to 3800 Hertz might be something like that.

And of course, if you measure another channel, would it be exactly the same? No, but it might be roughly the same. I mean, all these channels are engineered to pass human voice. Alright? In particular, there are now international standards that say it's got to pass at least between here and here with pretty good fidelity.

All right. So that's the first thing you do. You develop a model of the channel. And the key things about it, first order are, it's nearly a linear channel. What you put in will nearly come out. It has a certain bandwidth, which is somewhere between 2400, we used to say, way back in the '60s. Nowadays we might say we can actually get through 3600, 3700 Hertz. But that's what's the order of the channel bandwidth is. So we have W equals 3700 Hertz.

And then, of course, we have some noise. If we didn't have any noise, how many bits per second could we get through here? Hello? Infinite, right. Send one number with infinite precision. And that would communicate the entire file that we wanted to send.

And so we have a signal-to-noise ratio. It's questionable whether the noise is totally Gaussian. You've heard a telephone channel. You know, it's some weird stuff. But this is what our kind of normal telephone channel would look like. I should say, greater than this.

Nowadays. Used to be not nearly as good... Or I shouldn't say nowadays. This was true about ten years ago. Nowadays everybody is using cell phones, and these things have all gotten terrible again. All right? So please do not design your modem for a cell phone. Let's plug a nice wired connection into the wall.

OK. So that's a first gross characterization of this channel. What would you then do on the second day of your summer project? Yeah?

AUDIENCE: [INAUDIBLE].

PROFESSOR: OK. That's a good idea. What do you have in mind in designing a modulation key?

AUDIENCE: [INAUDIBLE].

PROFESSOR: A signal constellation. All right. What signal constellation are you going to use? PAM? QAM? Yeah? What would control your choice? ISI might. This channel doesn't have a flat response.

We might -- let's say we've actually developed a model where Y of t is X of t going through a linear filter, convolved with some response h of t plus N of t . So what we transmit gets filtered. We add noise to it. And that's what we see at the output.

That's just a first order model. This is basically the way modems were designed for telephone channels for many, many years. Just with that simple model.

All right. So if we have a filter here, we might have to worry about intersymbol interference. But you're not quite back at the basic level.

Let's suppose we decide that the channel is reliable just within some particular frequency band w , and within that band, it's more or less flat. All right? When I talk about white Gaussian noise, I'm always going to mean that the channel, the noise spectral density, and the channel response are flat within the given bandwidth. All right? So I'm not going to have to worry about ISI at this level. Of course I will in my telephone line modem. Yeah?

AUDIENCE: So even if we want high speed, [INAUDIBLE] so we want a rate that would be greater than [INAUDIBLE].

PROFESSOR: I suppose. I mean, do you have enough information to say?

AUDIENCE: Well, for example, if we [INAUDIBLE] 64 kilobits per second, can we travel 3,700 [INAUDIBLE] of bandwidth that we need use the bandwidth [INAUDIBLE].

PROFESSOR: OK. Is there any chance of sending 64 kilobits through this channel?

AUDIENCE: Yes.

PROFESSOR: There is? How do you know that?

AUDIENCE: If you use [INAUDIBLE PHRASE].

PROFESSOR: All right. So that would be 32 level complement -- that would be five bits per Hertz, basically, talking in terms of spectral efficiency. Times even 4000 kilohertz, that only gives you 20 --

AUDIENCE: Then a higher level.

PROFESSOR: Then a higher level. 1024 level QAM. But what's going to limit you? I mean, why not make it a million level?

AUDIENCE: [INAUDIBLE].

PROFESSOR: The signal-to-noise ratio, all right? OK.

Well. I suggest what you do on your second day is you start thinking about, on the one hand, simple modulation schemes. And I would say 32 -bit QAM is simple in the context of this course. It has no coding. But just kind of give yourself a baseline of what might work.

But another very good thing to do is to establish an upper limit. Now, it's kind of amazing that we can do this. Up to 1948, no one would've thought that it was a reasonable to say, well, there's a certain upper limit on what we could ever transmit through this channel. The idea was if you transmitted fast, you'd make a lot of errors. If you slowed down, you'd make fewer errors. And you know, there's no hard and fast limit.

But of course, what Shannon did was to show that for mathematically well-specified channels, of which this is one, there is a certain limiting data rate called the channel capacity. And what the capacity is for an additive white Gaussian noise channel, this is the most famous formula in information theory. If we have time, I'll derive it for you. Otherwise I encourage you to look it up.

The capacity of this channel is entirely specified by these two parameters,

bandwidth and signal-to-noise ratio. And it's simply W times the binary logarithm of one plus SNR in bits per second. All right? And the Shannon proved a strong capacity theorem and a converse. The capacity theorem says, there does exist a coding scheme -- perhaps involving QAM, perhaps involving lots of other stuff -- that can get you as close as you want in rate, for any rate less than capacity, there exists a -- let's say a coding slash modulation scheme, slash modulation, slash detection, slash decoding scheme -- such that the probability of error is less than or equal to ϵ , where ϵ can be chosen as in any arbitrarily small number. All right?

So if you want a probability of error of ten to the minus five or ten to the minus ten, however you measure it -- can't be zero. Very big difference between zero and ten to the minus ten. But if you want probability of error of less than ten to the minus ten, then Shannon says there exists a scheme that can get there, as long as your rate is less than this capacity or Shannon limit.

All right. So I suggest it would be a good idea to calculate what that is for this. Now, signal-to-noise ratio is 37 dB. Then one plus SNR is about the same as SNR. \log_2 of 37 dB, factor of 2 is 3 dB, so that's about 12 and a third. This is our first example of why calculating in dB makes things very easy.

So this is about 12 and a third times whatever the bandwidth is. 3700 Hertz, and we go through that calculation, and we get something over 4,2000 bits per second. All right?

So you have no hope of going more than 42000 bits per second, if this is, in fact, an accurate channel model. I mean, I'm assuming that these two numbers are correct. If this turns out to be 50 dB, then you can go a little further. All right? Yes?

AUDIENCE: [INAUDIBLE].

PROFESSOR: Yes. And it's really only true for a perfectly flat channel that satisfies the complete mathematical definition of a white Gaussian noise channel within the band. I'm shocked to find that this is -- lots of people get through MIT Communications course

and never see a proof of this formula. This is about as fundamental as you get. That's why I include one in this course. Or perhaps you've never even seen the formula before. Certainly don't understand where it's valid and where it's not valid.

OK. So on the other hand, we might try something like 32 QAM. If we know something about that, we make it so that it fits in here. We find that we can get maybe five bits per second per Hertz. That's something called spectral efficiency, which in this course, we'll always designate by ρ . So we can get maybe 3000 Hertz without having to worry about filtering too much. So we're going to have a system that sends 3000 QAM symbols per second -- 3000 two-dimensional signals per second, each one carrying five bits. And so with that, we'll get 15000 bits per second.

So these kind of establish the alpha and omega, the baseline and the ultimate limit, of what we could do over this channel. With a simple uncoded scheme, we would check that we can, in fact, get our desired error rate at the signal-to-noise ratio that we have here. So we know, without sweating hard, we could get 15000 bits per second. Shannon says we can get 42000 bits per second.

Maybe we want to put a little coding in there. How does Shannon say you could get to this marvelous rate? He says, well, what you need to do is to choose a very, very long code. And if you choose it the right way and decode it in the right way, then I can prove to you that your probability of error will be very small.

Of course, Shannon's proof is not constructive at all. If you've ever seen it, it involves just choosing a code at random, decoding it exhaustively by simply looking at all the two the nr code words, and finding which one is closest to the received signal, and that's not very practical. So there's nothing, in that sense, practical about Shannon's theorem.

But Shannon says, the way you get there is you code. What is coding? Coding is introducing constraints into your transmission. Uncoded is where you just send an independent five bits in every symbol. And the next symbol is newly, you choose it new. So there's no dependencies between the symbols. It's just one shot

communication, again and again and again.

Coding is introducing constraints. From Shannon, it's clear you need to introduce constraints over very, very long sequences. Basically what we do is we take some alphabet of all the sequences that we could possibly send -- maybe the alphabet is all 1024 QAM sequences that we could possibly send.

And then the code, by the code, we say we weed out a lot of those sequences. So we only send a small subset of all sequences that we might have sent. As a result, we make sure that there is a big distance between the sequences that are actually in the code. The fact that there's the much bigger distance than there is between the signal points in any particular small constellation means that if we do truly exhaustive maximum likelihood decoding, it's extremely unlikely we'll confuse one sequence for another. And therefore, we can get a very low error probability.

And the whole course is about the history of people's efforts to realize what Shannon said on the particular channel that we're talking about. The additive white Gaussian noise channel, which, as you might imagine, is the canonical channel. But it's also proved to be a very good model for things like the deep space channel, the telephone line channel.

In chapter one, I talk about the history and you can get some sense for the time flow. And it really took until -- well, depending on how you count it, up to about 1995 to when people could get effectively within tenths of a dB of the Shannon limit and you could say the problem was cracked. And research has continued for a little -- the past ten years, and this has been made more practical, and they've spent more channels refined in many, many directions. But really, you could say it was a 50 year project to take this existence theorem and make it real in a practical engineering sense.

And that's the story of this course. I think it's a terrific story. So that's the way I like to package the course. How are we going to get to what Shannon said we could do? Alright, so that's the whole story.

You having a problem hearing me?

AUDIENCE: It's just the noise in the system.

PROFESSOR: OK. Well, it's probably not an additive white Gaussian noise channel.

One of the things about an additive white Gaussian noise channel is that there's no dependence in it. Time dependence in it. So a third channel of very very practical interest these days is a wireless channel, radio channel. Let's imagine it's a single user channel. Everybody in this -- everything in this course is going to be single user, point-to-point.

But nonetheless, the FCC has said we can have five Megahertz. All right? And we've got to stay within that five Megahertz up to some little slopover that won't bother anybody. And again, your design problem could be, well, how do you send as much data as possible through a five megahertz wireless channel?

If you take the Wireless course, and I encourage you to do that, or even if you listen to the last couple of lectures, I think, in 450, you know that there's a big difference in the wireless channel and it's time-varying. In particular, it has fading. So sometimes it's good and sometimes it's bad. You have outages. Not characteristic of the telephone line and the deep space channel. So that introduces -- well -- a whole lot, many more considerations.

I'm not going to talk about that in this course. For that course, it's held in the next room immediately following this one. And I think it's an excellent course.

OK. So that's the story of the course. We're going to go from day one knowing nothing about how to signal through this channel, except maybe we've taken some course that's introduced to us words like PAM and QAM. And by the end of the course, we're going to know how to go at the Shannon limit. And we're going to know a whole lot of techniques at that point. We're going to know how to code, we're going to know how to decode at least some representative examples, which I've chosen to be the ones that have been the most useful in practice. And they also tend to be the most interesting in theory. OK? Is that clear?

OK. Just an aside on the first homework set. As I said, these are just supposed to be warm-up exercises that should get you comfortable with operating in the additive white Gaussian noise channel, which, at least some discrete time, turns out to be trying to code in Euclidean space. To encode and decode in Euclidean space.

And the very first thing I do here is to give you my dB lecture. Now, how many people here would say they're comfortable using dB? Nobody. OK.

Over the years, I've tried to encourage Bob Gallager to talk more about dB in 6.450. And now I believe he does give the dB lecture, but I don't believe he gives it with very much conviction. I'm convinced the dB are a very useful thing to know about, and I think Bob had a bad experience with dB early in life is really what the problem is. Some army sergeant said somebody was three dB taller than somebody else, and he was so revolted, he never wanted to talk about dB again.

OK. dB, which stands for decibel, one tenth of a Bell -- the reason B is capitalized is it is after somebody's name, Alexander Graham Bell -- are very useful whenever you want to use logarithms. They basically are a system of logarithms. This gets all confused with ten log ten and 20 log ten and EE tests, but basically it's just a very well-designed, for humans, system of logarithms.

Think about it. What would be the most convenient system of logarithms? We're going to use this whenever we have talking about factors, or the multiplication of a bunch of numbers. First of all, you'd want to have it be based on the base ten number system, because then it's -- you know, we have a decimal system, and we want to be able to very conveniently multiply by ten, 100, 1000, and so forth. So a very natural logarithm to consider is log to the base ten.

But so if we have log to the base ten of some factor -- let's say log to the base ten of 2 is 0.3010. And in fact, you know there are lots of tables that give logs to base ten. But it's just not so easy to remember that the log of base ten of 2 is 0.3. Basically if we take the log to the base ten of the whole range from, say, one to ten, that's really all we have to know. Because what we need is one factor of ten. This goes from

zero to one.

OK. What we'd really like to do is to spread this out for human factors engineering. We'd like to take ten to the base ten, which will now map the range from one to ten into a nice interval from zero to ten. That's very easy for people to understand. And ten to the \log_{10} of 2 is 3.01 and so forth.

OK. So if that's why we say a factor of 2 is equal to 3 dB. I find it's always helpful to say "a factor of" when talking about dB, just to remind ourselves, we're always talking about a multiplicative factor. So a factor of alpha is $10 \log_{10}$ alpha dB.

Now most of you are mathematically sophisticated enough to know this is basically the same thing as saying log to the base ten to the one-tenth of alpha. So we're really using a base here, which is ten to the one-tenth point 1, which is about 1.25 something or other. So the number whose one dB, which is beta to the one, is about 1.25.

So we're just using a log to a certain base, which is very convenient. The base is ten to the one-tenth. And that's all it is, all right? Beyond that it's not very mysterious.

So I think it's very useful just to remember a little table. alpha of one is how much in dB? It's zero. Or I write it two ways. Round numbers, or in any system, it's that. 1.25 is about 1 dB. 2 is 3 dB. Let's see. 3 is about 4.8 dB. 4 is what? 4 is just 2 squared, so that has to be 6 dB. 5 is what? 5 is $10/2$, so that has to be -- ten down here is 10 dB. That's exactly 10 dB. And so 5 has to be 7 dB. And 8 is 9 dB. And that's consistent, by the way. You see then ten eighth, which is $5/4$, is 1.25, which is 1 dB.

And just, I find it useful in everyday life. Maybe that makes me an engineering nerd. But the first problem, for instance, has to do with compound interest. If you just remember dB values of things, you can do things like compound interest calculations in your head. Or I gave you a more engineering example over here. If you want to evaluate \log_2 of 37 dB, it's very easy. You just divide by 3. All right?

So anyway. I encourage you to memorize this short table. There's some more things in it. And use it day by day, and become known as a real MIT person.

Why schemes? Well, why not?

All right. The other problems on the first homework set -- one's a quite algebraic construction, using Hadamard matrices of geometrical signal sets, like biorthogonal NS orthogonal and simplex signal sets, with a easy decoding algorithm. And that's good to know about.

Next one is a QAM problem, asking you to look at various arrangements of signal points in two dimensions, and try to find the best ones for various criteria.

The last one has to do with spherical shaping of large constellations, in many dimensions, with high spectral efficiency. Many bits per second per Hertz.

OK. And that's chapter one.

Ok. Chapter two we'll do very quickly. And then I could see we're not going to get to chapter three at all. And that's fine. You can read it. Ashish, if you want to say a word about chapter three next time, that would be fine.

Chapter two is really about given a continuous time additive white Gaussian noise channel, with model $Y(t) = X(t) + N(t)$, and here we have a certain power and we have a certain bandwidth limitation W , the same things we were using before, here we have a certain power spectral density N_0 . So there are the parameters of the channel. Or I've collapsed them into two parameters, SNR and W . How can we convert this to a discrete time additive white Gaussian noise channel, which will be a channel model like this -- a sequence of symbols Y_k is equal to a sequence of transmitted symbols X_k plus a sequence of noise symbols N_k .

Where again, this is going to be IID Gaussian. That's what "white" is in discrete time. Independent Identically Distributed Gaussian random variables. This is going to have some variance S_N , it's going to have some power constraint S_X . We don't specifically see a bandwidth in here, but the bandwidth is essentially how many symbols we get per second. And I think all of you should have a sense that a

bandwidth of W translates, through the sampling theorem, or through PAM modulation, or something else, is roughly equivalent in discrete time to two W real symbols per second, or to W complex symbols per second.

Now the -- sorry, did I screw you up? Do I need to readjust? All right. Perfect.

There is a particular way we can -- well. So we're going to show that these two, the continuous time and the discrete time channel -- if we have a continuous time channel, we can get a discrete time channel like that, where the values of the parameters, signal-to-noise ratio, for instance, translate in a natural way and that they're the same. And furthermore, where we carry as much information on the discrete time channel as we do on a continuous time channel. So there's no loss of optimality or generality in doing this.

Now you've all, with a couple of exceptions, taken 6.450, so you know how to do this. One way of doing this in a fairly engineering sense is orthonormal PAM -- Pulse Amplitude Modulation. And you should know how this goes.

We're going to take in a sequence X , which I might write out as a sequence of real symbol levels X_k , perhaps chosen from a PAM alphabet. I'm not even going to specify. And what do I need to know here? This is going to be at two W symbols per second, real symbols per second. OK. That's my transmitted sequence that I want to get to the other end.

PAM modulator -- so this is PAM modulator -- is specified by a certain pulse response p of t . and I'm going to require that this be orthonormal in the following sense. That the inner product between p of T minus kT and p of T minus jT is equal to the chronic or delta. Δ_{kj} .

So that the time shifts of this single basic pulse by T , where T equals one over two w , the symbol interval, are going to be orthogonal to each other, and what's more, orthonormal. So in effect, I've got a signal space consisting of all the linear combinations of the time shifts of p of T by integer values of T , and I'm going to send something that's in that signal space simply by amplitude modulating each one

that comes along.

All right. So what I get out here is the continuous time signal X of T , sum of X_k p of T minus kT . That's what the modulator does.

Then the channel, continuous time channel -- so this is discrete time here. Now I'm into continuous time. The channel is going to add white Gaussian noise N of T with N_0 power spectral density to give me channel output Y of T is X of T plus N of T .

And now I need a receiver to get back to discrete time. The receiver will simply be a sampled matched filter, which has many properties which you should recall.

Physically what does it look like? We pass Y of T through p of T minus T . The matched filter's turnaround in time. What it's doing is performing an inner product. We then sample at T samples per second. Perfectly phased. And as a result, we get out some sequence y equal Y_k . And the purpose of this is so that Y_k is the inner product of Y of t with p of T minus kT .

And you should be aware that this is a realization -- this is a correlator-type inner product. Correlate and sample inner product. All right?

So what are some of the properties that you developed ad nauseum in 6.450? First of all, if we take the signal part of this, and we have the sampler phasing right, what do we get out for the signal part of Y_k ? We have X of T is this. If we correlate that against, let me say, p of T minus jT for all j , we're going to get zero for everything except for the desired sample. For the desired sample, we're just going to get the desired sample out. So from this, we get Y_k is equal to X_k plus the noise term, N_k . In other words, there's no intersymbol interference.

The noise term -- by taking these inner products, what are these? These are the coefficients of y_k in the signal space, under -- these are an orthonormal expansion of Y of T . Those components which lie in the signal space. Can we ignore the components that don't lie in the signal space? Yes. If it's out of white Gaussian noise by the theorem of irrelevance or whatever it was called. There is no information about the X s contained in any part of Y or T that is orthogonal to the

signal space, in the orthogonal space.

So we simply take the parts that are in the signal space. And another property is that this is white Gaussian noise. Then N_k is simply an IID jointly Gaussian sequence with variance S_N equal to N_0 over 2 if the power of spectral density here was N_0 .

And one other property of I should have mentioned here is that again by the orthonormal property, as you would think intuitively, if we sent two W symbols per second, and each one is limited to power S_X , then this will have power $2 W S_X$. The powers are the same in discrete time and continuous time.

So what do I get? Let me now show the SNR. Let me compute it in discrete time and in continuous time. The SNR in discrete time is $2 W S_X$ over S_N , which was N_0 over 2.

I'm not seeing the two's fall in the right place. What am I doing wrong?

So this is the power. Discrete time and signal-to-noise ratio is just S_X over S_N . Now S_X is P over $2W$. S_N is N_0 over 2. Now I'm happy. It's P over $W N_0$. Continuous time is p over $W N_0$.

OK. The point is, the signal-to-noise ratio is the same.

In discrete time, talking about bandwidth, the bandwidth is really $2W$ symbols per second. So that's our measure of bandwidth in discrete time. And that is approximately equal to W Hertz in continuous time.

Now, I haven't proved that to you yet, have I? How to prove that we can get a bandwidth as small as W but no smaller. Again, you've done this in 6.450, so I hope I can just remind you.

If the shifts of p of T are orthonormal, that says something about p of T star p of T , which is the correlation of p of T with itself. It means g of T has to be 1 to equal 0, and it has to be 0 at all other times, at all other integer times.

Which means the Fourier transform, power spectral density of p of T , it has to satisfy the Aliasing Theorem, or the zero ISI, the Nyquist criterion for zero ISI, which basically means the alias version of the power spectral density of this has to add up to a perfect brick wall response between $1/2T$ minus $1/2T$ and $1/2T$ in the frequency domain.

And there are simple ways to make it do that. Which is simply to have a roll-off, a real frequency response, with a roll-off that's really sharp as you like above $1/2T$. But so we can say the nominal Nyquist bandwidth equals $1/2T$, or T is $1/2W$. So it's W .

So the moral of this is that we can design orthonormal pulses. If we have a similar interval of T equals $1/2W$, we can design an orthonormal signal set that satisfies this condition for any bandwidth that's not much greater than W . All right? And it's obvious from the Aliasing theorem that we can't do it if we try to choose some bandwidth that's less than W .

So that's what justifies our saying, if we are $2M$ symbols per second, we're going to have to use at least W Hertz of bandwidth. But we don't have to use very much more than W Hertz of bandwidth if we're using orthonormal PAM as our signaling scheme.

So we call this the nominal bandwidth. In real life, there will be a little roll-off -- five percent, ten percent. And that's a fudge factor in going from discrete time to continuous time. But it's fair that we can get as close to W as we like. Certainly in the approaching Shannon limit theoretically, we would say that we can get as close to W as we need to, I should specify. All right?

So there's one other parameter that Ashish is going to be talking a lot more about in the next lectures which is called spectral efficiency. So far I haven't told you what information the sequence of symbols is carrying. But let's suppose it's carrying R bits per second. All right? We have a modem. It's a 64000 bits per second modem or whatever. R is 64000. Or whatever it is.

My objective is to make R as high as possible subject to the Shannon limit. If I am sending R bits per second across a channel which is W Hertz, in continuous time, I'm simply going to define, I'm always going to write this as ρ . And I'm going to write it simply as a rate divided by the bandwidth.

So my telephone line case, for instance. If I was sending 40000 bits per second in 3700 Hertz bandwidth, I'd be sending 12 bits per second per Hertz. That's why we say that. That's clearly a key thing. How much data can I jam in? We expect it to go linearly with the bandwidth. ρ is the measure of how much data per unit of bandwidth.

What does that translate into in discrete time? Well, I'm dividing up this rate into $2W$ symbols per second. How many bits am I sending per symbol? I'm sending R over $2W$ per symbol. So simply ρ over two bits per symbol, or since I'm always going to think of Euclidean space, I will often write that as dimension-- ρ over 2 bits per dimension.

All right. Well that's not the exact translation I like. I'd like to talk in terms of ρ . So here, ρ can be evaluated at the number of bits I'm sending per two symbols, or per two dimensions.

All right. So for the discrete time system, our major spectral efficiency is just going to be the number of bits I'm sending every two dimensions. It turns out if you get into this game, you feel that two dimensions are more fundamental than one dimension. Which may be a manifestation of the fact that complex is more fundamental than real. Really we want to talk about the amount of information we're sending per a single complex dimension, or we can say two real dimensions. Geometrically they amount to the same thing. Again, in the notes, there are discussions about how you go back and forth, and so forth.

So the bottom line here is that we can talk about spectral efficiency, which is obviously a continuous time concept. But we can talk about it in the discrete time domain. And what it's going to mean simply is the rate that we're managing to send, by particular scheme, if it's per two dimensions. For instance, 32 QAM has a

spectral efficiency of--

I didn't make myself clear at all.

AUDIENCE: Five bits per two.

PROFESSOR: Five bits per two dimensions, all right? So that's the basic spectral efficiency of a 32 PAM modulation scheme.

2 PAM, plus or minus one. What's the spectral efficiency of that if it's for two dimensions? It's a trick question. One bit per one dimension. It's two bits per two dimensions. If it's 2 PAM, it's actually the same thing as 4 QAM, because 4 QAM is just doing 2 PAM place in two successive symbols. So 2 PAM, you could send two bits per two dimensions. 4 QAM is also two bits per two dimensions Ok.

Similarly for any [UNINTELLIGIBLE], we'll be talking about codes which cover many dimensions. If you send R bits in N dimensions, then the spectral efficiency is going to be $2R$ over N . So R over N in one dimension, $2R$ over N in two dimensions.

I guess initially, this catches people up. Because it's more difficult to normalize for two dimensions than for one dimension. But come on, suck it up. You're a graduate student at MIT. You can do it for two dimensions. And I assert that it really is the more natural thing. In particular, makes this nice correspondence between discrete and noncontinuous time parameters.

So the SNR is the same. The bandwidth has different interpretations, but we're talking with W in both cases. Spectral efficiency, we can measure it as either discrete time or continuous time.

And oh, by the way. What is the channel capacity formula in terms of spectral efficiency? The channel capacity was basically the data rate has to be less than $W \log_2$ of one plus SNR. Find an equivalent way. So this is writing things in terms of bits per second, or equivalently ρ , which is R over W , has to be less than \log_2 of one plus SNR. This is in bits per second per Hertz. Or in discrete time, we have the same formula, except this becomes 50 bits per two dimensions.

And so things have been normalized so that now you only have a single parameter to worry about. What special efficiencies can you achieve? It is purely a matter of what SNR you have. Telephone line channel 37 dB, then you get spectral efficiency of 12 and one-third by Shannon.

OK. So I'm sorry not to be able to give you the very lovely derivation of this formula as a little bit of culture. I encourage you to read chapter three. You will not be held responsible for any of these first three chapters. They're all more or less just for orientation. Next time, Ashish will get into the real stuff. Any final questions?

OK. See you in two weeks.