

IntroToLinearDynamicalSystems-Lecture01

Instructor (Stephen Boyd): Yeah, I guess this means we've started. So welcome to EE263, and I guess I should say for many of you, welcome to Stanford. Well, as I said, this is EE263. I'm Stephen Boyd.

And I'll start actually just with some — I'll cover some of the mechanics of the class and then we'll start in. Today is just gonna be sort of a fun lecture, so it's not representative of the class. So by the end of the — you'll leave thinking, "Well, it was interesting, but it was kind of, like, content-free." Anyway, trust me it's not representative of the quarter.

What's that? Oh, wow. Is there finally a room so big that I'm not loud enough? How about you people in the back? Can you hear me? I can be loud. We'll try it. We're gonna go — we'll go natural until you tell me you can't hear me or something like that. Okay.

So let me just say a little bit about the mechanics of the class. I should introduce — at the moment, we have two TAs. That will grow to the end of the day to three and maybe four later. There's Jacob Mattingley over there, Yang Wang is here, and hopefully we'll have another one or two within a day or so.

So let's see. I'm trying to think what the important things are to say about the course that are just mechanics. I'll get into prerequisites and things like that soon. I do have to make one announcement. Maybe it's even legally required. I'm not sure. But Stanford is going to experiment with putting this course in particular online, open to the world. Strangely, this is gonna be after a one-quarter delay. So that means this course will go online everywhere in a quarter.

Now, that's interesting because that means if you ask — well, right now the way it works is this: If you ask a question now in the class and it's a dumb question, then of course, as you know, it's on streaming video. That means that your roommate or mates can go back and replay your dumb question multiple times and laugh, you know. But that's only 30,000 people at Stanford who have access to that. So if we go online though, of course, that's a lot more. And of course, it means it's entirely possible that you'll end up on YouTube. Maybe I will. Who knows? We'll see what happens.

So anyway, but to put you at ease over that, if the camera — I guess they were saying one of the things they have to do is they have to clear copyright and things like that before they put this on, well, open to the world. By the way, the course notes and course materials have been open to the world forever because I control that, and it's always been open to the world. And now I'm actually talking about the lectures.

So what'll happen is I think if you do ask questions like that, rest assured your face will be all fuzzed out and they'll change your voice, too. So go ahead and feel free to ask any question you like is what I'm saying there. But who knows? It's an experiment. We'll see how it works. I'm gonna forget about it. I mean, it's gonna be interesting for us.

The other courses doing this are EE261, a couple of CS courses and things like that. And so we all looked at each other and thought, what have we gotten into? We're gonna have to behave. That's gonna be very difficult. But I think the way that's gonna work is this, is they'll be edited. So if, for example, you are watching this, it's February right now. Maybe this part will all be edited out. But the way you'll know when something interesting happened in the class is ultimately my head will go like this. There'll be a little continuity glitch there, and that'll mean that something we said needed to be edited out. Okay.

So maybe I'll just jump right in and cover the course mechanics. If you can go down to the pad, that'd be great. Okay. So I'll start by just going over the course mechanics. I'll say a little bit in a very broad brush what the topics of the class are. It seems odd to not say what a linear dynamical system is, since that's what the name of the class is. Actually, what's interesting is I'll say what it is, and we won't come back to that topic for several weeks. I'll say a little bit about why you'd wanna study the material in this class, and then we'll look at some examples.

So, first of all, in course mechanics, I should say this: Everything — the course website — oh, by the way, we do not use EE class. So if you wanna go there, that's fine. You'll just find a frame that has the real website, okay? So we don't use EE class or anything like that. It's just a website where you'd expect all the course materials to be, and we try to make the website basically the most accurate source of information. After the website, I would say the next most accurate source would be the TAs; after that, only if you're desperate, me. So if I say something and the web says something different, it's more than likely that what's on the web is right. So just to let you know, what's how we do this.

We also correct typos and things like that almost in real time, and then we deny that the typos ever existed. So just to let you know, that's the policy. So you'd say, "Excuse me. There's a minus sign here." And I'll say, "Nope, no, there's not." And you can look on the web, and all you have to do is refresh. If you refresh, the minus sign will just go away, and everything will be back to right. So that's how it's gonna work. Okay.

So I should add for the reader, the PDF file for the entire reader is available online. In fact, my understanding is that the bookstore charges some idiotic fee. How much did this cost? Do you know? Thirty bucks. See, I think that's kind of ridiculous. But anyway, that's fine. Maybe we'll find another way to print it next year. But the entire reader's available as a PDF file, all the homework, just everything is available on the website. Everything will be available there.

We also fool around a little bit sometimes during the quarter to see how often people are checking the website. So we'll post things that are incomplete, links that go nowhere, just to see, like, we'll do it at 1:00 a.m. to see, like, what — anyway, we did that experiment last year, and I think we got to, like, 1:20 a.m. before somebody said, "Hey, you posted this, but the link isn't there." So we do that just for fun. Okay.

So the course requirements are gonna be weekly homework. So the homework, tentatively we're gonna be on a Friday cycle, so the first homework, which incidentally is assigned, and you'd know that by looking at the course web page, so the homework is actually assigned. I won't even say anything about it. I won't come in and say, "Oh, by the way, Homework 3's assigned." It'll just be on the web. So Homework 1, which is assigned, will be due next Friday.

So we may have to adjust that because there'll be a section for the course, sort of a problem section or whatever, which is not required, but it will also be televised. That means that we have to go to SCPD. We have to find a room that's television-ready. So we don't know exactly what day that will be, and we may have to adjust the schedule a little bit around that. So that's why I think even right now, two places on our website, we say that the homework is due Friday, and another place it mentions homework is due on Thursdays.

When we find a room for the section and announce it, then everything will be set after that. Things like TA office hours will be set. Let's see what else would be set. The section time, all that sort of stuff, will be set. Hopefully, we'll be able to do that in the next day or two. That would be great to get that ready.

For the homework, actually I would — not only are you allowed to work together, but actually I would encourage you to work in groups, in small groups on the homework. And it has to be some kind of group that makes sense for you because a lot of the homework, it's sort of — it's easy to kind of do it. It's actually — if you really wanna know if you understand it, you try to explain it to the other person or two people you're working with. If they're kind of looking at you kind of like this, that means they don't know what you're talking about. And that means — it either means you don't understand it or you haven't explained it that well, so that's a very good way.

Oh, and by the way, that means that when you're playing the other role, when someone you're working with is trying to explain it to you, don't be polite. If what they're saying doesn't make any sense to you, just say, "Interesting, but that makes no sense." So that's how I would — I'd encourage you to do that.

The homework will take a lot of — it should take a lot of time. I don't know what an average amount is, I mean, I don't know, ten hours. I don't know honestly because then you can't ask people. It's sort of a biased sample. And some people try to — they know it'll insult me if they say, "Oh, I do it in 90 minutes," or something, and others go on and say, "Oh, no, it took me 25 hours." I don't know which to believe. So it's somewhere in between those two.

It will be graded very crudely. And I think you just do the arithmetic if you multiply the number of people in the class. These homeworks, they are big, thick things. They will take you a lot of time and effort to do these. It's not gonna — it doesn't count a huge amount into the grade, but this is graduate school and grades don't really matter and all that anyway.

And they'll be graded crudely. So I think the official amount of time is something like 15 minutes. So let's just say you'd take six hours to do the homework — let's just say eight. Someone's gonna go over it for 15 minutes. Now, that might strike you as odd or something. But, I mean, you do the homework so that you can learn. That's actually I believe where you do the real learning is in the homework, so although it doesn't count a lot for the grade, it'll be looked at by a grader briefly very late at night for 15 minutes max. So but still it's very valuable just having done it or anything like that.

What that means is please don't come to us later and say, "I think, gee, the grader didn't get some subtle point in my argument here," because basically, like, I got news for you. The grader looked at this between 2:07 and 2:14 a.m. on Tuesday. So it's likely that the grader didn't, but anyway, okay.

We'll have two exams. These are both take-home exams. You may have already have heard about these. Actually, I'm just curious. How many people have heard about these? Oh, good. And what have you heard? Were you pointing down or — what was that? Oh, that was a down — I thought we had something point down. Now, wait a minute though. But then why are you here? Don't know. Okay, sort of self-destructive instinct? Okay. All right. So all right.

So actually, these are fun. These are take-home exams. They're nominally 24 hours. People rarely take over 20 to do them, but they do sometimes. So they're fun. They're now an institution. We actually even tried to change it, I think, like, last year or something like that. And students from previous years came back to protest to say that, "No, you couldn't possibly do this. It's part of the whole experience," and so on, and so forth. So anyway, so that's how we'll do that.

And we every now and then, I just ask people if they'd be interested in any other format. Like, how about 48 hours? And that, people went, "No." And I think shorter is silly, so that's how it's gonna be.

Oh, I should say that there's something very important here for scheduling. The take-home exam traditionally, and I should also say illegally, is scheduled for traditionally sort of the end of the last week of classes, okay? So there is an official exam date for this course. I forget when it is, December — I'll probably already be in Hong Kong, by the way, at that time. So we do it sort of at the end of the class. That's totally illegal according to the university, but they know where I am. They can come and get me any time. That would be good actually to have posted on the web, would be to have the — I don't know what the police unit for the registrar's office is, but that would be good actually. All right.

So what this means though, this is actually very important. I've already gotten a couple of emails from people who are making flights home or whatever, and basically as early as you wanna leave, I mean, assuming it's not in the quarter, we will work around it. If you're leaving literally, like, the week after classes finish, no problem. You're a beta tester for our final exam, okay? So we'll work around you, just to let you know that.

Okay. This is all just mechanics. Any questions about the mechanics, how it'll work? Did I forget anything? The only thing I might have forgotten is there won't be a section this week because, well, we don't have a room. So next week will be the first week there'll be a section. We don't know what day it will be or where it will be, but that will be announced on the email list, and it will also go on the website. Okay.

So now we've covered the mechanics. Let's go a little bit into — I'll say a little bit about the prerequisites for the class. That's actually important. So the first is that you should have had an exposure to linear algebra. Now, these words are actually chosen very carefully. You typically in this class, there are people who have a very — it is a very wide range of backgrounds in linear algebra, all the way from essentially none from people who said, "Oh, I took a course on multivariable calculus. I think I know what a vector is and a matrix," all the way to people who've taken multiple courses on linear algebra.

In fact, what you really need is just something like an exposure to it. So, I mean, you definitely should have seen vectors, matrices, hopefully ideas like rank, range, null space. However, since linear algebra classes are by tradition extremely boring, it is natural that you should have either hated these courses and actually suppressed the memory of those courses as much as possible. That's natural. In the first part of the course, we will be going back over this, and that means of course that painful memories will be coming back, but there's lots of us here and we're all doing it together. So everything will be fine there.

I should also say one other thing here, something different this year — actually, it started last year. This course is now offered twice in the year. It's offered fall, but it's also offered in the spring, and that actually is very important. It means that if you decide somewhere into the class, a couple of weeks in or I don't know when, couple weeks in that in fact what you'd like to do is actually take this in the spring and maybe take something like Math 103 or — I don't mean really Math 103, but if you could just look on the course catalogue and see what's involved there — if you wanna defer, just sort of take that and then take this in the spring, that's an option for you. That wasn't an option in the past, and that was a problem because people sort of didn't drop out, saying they — but having said that, a lot of people actually just do fine.

I should also make a comment to those who come in with a much stronger background. So there are people who come in with a much stronger background in linear algebra who've had maybe an entire course or whatever several years ago, so that's fine, too. Actually, there the reaction will be that somewhere around the fourth week, you might be thinking something like, "When am I actually gonna learn anything I don't already know?" Actually, trust me, you will because this is different from the class you took, I promise. And in fact, all those people actually come back to me later, not all of them, but most of them come back later and say, "You're right. I learned something." So okay.

The only other real prerequisite, and this is not even really a prerequisite that much, is at just a few tiny places in the course, we're gonna use the Laplace transform differential equations. This actually, even if you hadn't seen this, but it would be difficult for me to

believe that you haven't, again, we'll cover all the background material needed for this. So that's really the formal prerequisite. There's not a whole lot else.

Now, the course, this material — I'll talk about this soon — is used in tons and tons of areas. But in particular, you do not need to have taken a course on, like, control systems or something like that, which is one area. You actually don't even have to have taken any course on circuits and systems or a course on dynamics. So you could be — it would be fine.

We will look at examples sort of taken grossly from control systems, circuits and systems, dynamics, but we'll also look at examples from machine learning. There'll be examples from signal processing, communications, networking, all over the place. We'll take care that you don't actually need to know anything about these application areas. I mean, these are really more like little vignettes where you just kind of oversimplify it and show it here. So don't worry if you see things like that. In fact, today you will see things like that. Don't worry about it.

Whenever it matters, we'll make sure everything is — so the point is that although it's perfectly okay for you to have had a course on control systems, circuits and systems, dynamics, for that matter, machine learning, signal processing, it's absolutely not needed. And I know every year we have people in this class from, for example, economics, from all sorts of areas. Doesn't make any difference at all. Okay.

So let me say a little bit about the outline. The first chunk of the class is basically gonna be a review of linear algebra and applications. I'm gonna sort of assume that you've already had a class where somebody droned on and on about rank and range and the four fundamental subspaces and things like that. So this will actually be about modeling and applications. So it's actually where does this actually come up; where do you use this stuff? And that's actually the theme of the class.

Let's see. Then we'll talk about autonomous linear dynamical systems. I'll actually say what those are shortly. Then we'll move onto systems with inputs and outputs. And then we'll look at sort of basic quadratic control and estimation at the very end of the course. But in fact, this is really an application if you've understood the material before. Okay.

So maybe at this point, it's time for me to say what the class is about, although we're gonna — we'll drop this topic and we'll come back to it only in about three or four weeks. So what's a linear dynamical system? Well, a continuous time linear dynamical system looks like this. It's a vector differential equation, so it looks like $\dot{X} = A X + B U$. That's a matrix times a vector, X of T plus B of T times U of T . That's a vector, that's a matrix, and I'll talk a little bit about — oh, this reminds me.

On the course website, there are some extra notes. There's quick notes on matrices. You can just read them literally in 30 minutes. Please take 30 minutes and read them because, first of all, it's gonna say that these are the things I will use without ever — I will not go into them. I won't mention it. And the other thing is if the notation you saw was slightly

different, this would set the notation straight. Actually, I kind of try to use sort of a high BBC level of mathematical notation. So if you saw a notation that's substantially different, that's because what you saw was weird and strange. So maybe it was in some other weird field or something. Throughout the class, I'll make fun of other fields periodically. No, I don't think I get to do it today. So okay. [Inaudible].

Okay. So here typically, T , as the choice of symbol suggests, is gonna represent time. Of course, it need not, but it will represent time. Here, X of T , that's this vector. Actually, it's a vector function, and that's called the state. That's a vector. Sometimes colloquially, the actual entries of X will be called informally the states like that. So that would be the [inaudible] state, but that's slang, and you should know that. That's X of T . U of T is called the input or control. It depends on the field you're in. We'll talk more about that. And Y of T here is called the output.

I think this equation is sometimes called the dynamics equation, and this equation is sometimes called the measurement or readout equation. It's got all sorts of names. And then all of these matrices here have names.

You don't have to remember any of this because I'm going over this just so that you can't — otherwise, we'd go four weeks into the class and if someone asked you how's the class going, you'd say, "It's great, but we haven't even gotten to what the title of the class is." So okay.

So this matrix A of T is called the dynamics matrix. Here, B of T is sometimes called the input matrix. C is called the output or sensor matrix. And D is called the feed-through matrix. We're gonna come back and go over this again in horrendous detail when we really get to this. So you don't have to remember any of this or all that sort of stuff. Okay.

Now, this is too ugly, so it is often written in this very simple form that looks like this. It's $\dot{X} = AX + BU$, $Y = CX + DU$, just like that, where you suppress all sorts of things and have them understood.

So a linear dynamical system is nothing but a first order linear differential equation, nothing else. Now, there's other names for this. These are called state equations, dynamics equations. They have all sorts of names. It depends on the field you're in or what application area you're in.

I should also mention that A , B , C and D are traditionally — you can usually tell where someone got their PhD and so on and so forth by their choice of notation. For example, I think there was someone who taught at Stanford in Aero/Astro for a long time, and that was $\dot{X} = FX + GU$. So turns out that's because he got his PhD at MIT in 1967 and picked this up from somewhere. So you will see other conventions here. I mean, of course, it's nothing but notation, but just to warn you that you'll see that. But if you ever see someone who writes down $\dot{X} = FX + GU$, it probably means they went to MIT or something like that or took the class from someone who went to MIT. But now they'd switch to this, too, anyway, so it's complicated.

Okay. So let me mention a couple of things. Many, but actually not all, and we'll say something about that later, are time invariant. So that means that these matrices A, B, C and D, they're constant. They do not depend on time.

Now, if there's no input, that means so there's no B or D matrices, the system's called autonomous because essentially it goes by itself and has nothing to do with what U is, which is usually interpreted as some kind of input or something like that. And very often, there's no feed-through, so you get things that are very, very simple.

Now, some notation you'll hear is this: If these inputs and outputs are scalar, the system is called single input, single output, and I think the slang for that is SISO. When you have multiple ones, it's MIMO. Now, this is a bit silly, frankly. I mean, this is kind of a holdover to when it was really a big deal to have, like, two inputs and two outputs. Even other fields, like communications and things, signal processing, are now getting used to the idea that you typically process more than one signal at a time for more than one measurement or for more than one input. So, in fact, I guess wireless communications is going through the very end of what I'd call the MIMO stage of development.

So this was very hot ten years ago, and it was a big deal, and people would get all excited, and you'd say, "Wow. What's that?" And they'd go, "It's amazing. It's totally amazing. Instead of holding up, like, one antenna and looking at the radio signal coming from it, are you ready for this? We're gonna hold up two. Unbelievable. Can you believe this? Got it? Actually two. And we'll take those two signals and by processing them right, we'll increase the capacity of our cell network." Anyway, so other fields have been there, done that decades and decades ago. There are still fields that haven't, by the way, reached the MIMO stage. They will. It'll be happening soon, sooner or later. Okay. So okay.

Let me say something about discrete time. This is what we looked at so far was a continuous time. Let's look at discrete time linear dynamical system. That's nothing but a recursion. So here, instead of a derivative, it's simply an update equation. It says that the next state is obtained by multiplying the current state by a matrix A of T, and to that you add something which is related to an input. Here, the time is an integer. So it's a discrete time thing.

And sometimes the time is either called — it could be a sample; sometimes people call it a period. For example, in economics, you would talk about periods. These could be trading days, could be anything, okay, or this could be some audio signal processing, in which case these are samples at some standard rate, like 44.1 kilohertz or something like that. Now, in this case, the signals are sequences. In the continuous time case, signals are functions, and I'll say a little bit about that. So it's nothing but a first order vector recursion.

Okay. That's done. What I just explained, there's no way anybody would have gotten any idea of what they are. But at least now you cannot say I didn't tell you what a linear dynamical system was on the first day. I've done it. We're gonna come back, and we'll be going into this in horrendous detail later in the quarter.

So let me say a little bit about why would you study linear dynamical systems. Well, it turns out the applications, they come up in — nowadays, it's everywhere, absolutely everywhere. So, I mean, sort of historically, the first application was in automatic control. This was aerospace maybe in the '60s. That would be the first sort of real application. Now, at the time, it was super advanced technology, very, very fancy.

It hit signal processing, oh, let's say, mainstream signal processing, let's say, about 15, 20 years ago is when it hit signal processing. So until that time, signal processing — and indeed, I bet your undergraduate exposure to signal processing just fiddled with a scalar signal.

Actually, for how many people is that the case? Like, how many people took signal processing? Okay. And for how many of you did you ever hear of a vector signal? Cool. Really? Where? Cool. Okay. There were just a few others. Where are some of the others who heard of vector signal processing? Where? Cal Tech. Well, okay. Times are changing. That's really cool. Okay. Great.

So what you will find actually is except for undergrad signal processing, almost all signal processing now involves linear dynamical systems in one way or another or something like it, all of it. So all signal processing pretty much involves ideas from linear dynamical systems.

Communications, I'd say it hit big time about ten years ago, although in fact in communications, there was always some stuff that went back into the '40s and stuff like that. So communication's another area.

Economics and finance, it's totally basic, if you look at evolution of an economy or something like that. It comes up all the time. In finance as well, most of the models are just linear dynamical systems. Some of the notation will be different, but it's very close to what we'll be looking at.

Circuit analysis, so it may not be linear dynamical systems, but linear dynamical systems actually plays an important role, very important role in circuit analysis, circuit simulation and also circuit design, it comes up a lot.

It comes in mechanical and civil engineering. You see it in things like dynamics of structures and all sorts of other stuff.

Aeronautics, it's everywhere. It involves dynamics, control, navigation and guidance, so, for example, GPS. So things like this are all done using this, well, the stuff you will see in this class.

A few other things you won't see, which reminds me about something I didn't say. Here's a topic that maybe should be a prerequisite but actually is not, and that's actually probability and statistics. So I think that's kind of weird that we don't have it that way, but it's just it's not a prerequisite, so just to let you know.

Occasionally, I might refer to things that involve probability and statistics, but technically that is officially just a side comment. That's just for those who know what I'm talking about. Actually, if you combine the material of this course with sort of probability and statistics, now you're getting close to actually what is used in tons and tons of fields [inaudible]. By the way, it comes up in other areas, like machine learning, as well, is another one. Okay.

Now, the usefulness of the material, it scales with available computing power, which thanks to other people in EE and materials and areas like that, has been and is and will continue to scale by Moore's Law. So that's great because that's sort of the difference between this material in 1963 and this material now. It's a huge difference.

Now, you can actually — in 1963, you could do this stuff, and there were people who did, mostly militaries. Now, anyone can do it. It's widely used; it's widely fielded. And that's basically entirely due to increases in computing power. The computing power's used for analysis and design, but it's also used for implementation and actually just imbedded in real-time systems. And occasionally, I'll make some comments about that.

And I do wanna say something about how courses evolve, so especially sort of courses on mathematical-type, engineering-type stuff. So visual signal processing, I guess maybe the first class ever given was maybe at, like, MIT in, like, 1956 or something like that, and it was this super-advanced class with maybe, I don't know, four PhD students in math in the class. And sort of for a decade or two, it was this ultra-advanced course. It was ultra-high-end technology that the only people who would field this would be, like, the military and a couple of others, maybe some banks or Boeing or something like that would do it.

But DSP, as you know, that's now, like, an undergraduate topic. I mean, it's just everywhere. I mean, so that's how that went from super-advanced, advanced PhD-level class, 30 years later, again, thanks to technology, 30 years later, it's now, like, your basic — it's your second course. In fact, in some places, they're switching it and they're making it the first course in undergraduate Cs in, for example, electrical engineering, which I think is a good idea, right, because soldering skills are, well, they're useful, sure, but maybe less so than moving forward — I can solder by the way, just if you're thinking — if you're curious. Okay.

So one of the origination in history, I can say a little bit about this, so part of it, you'll trace to the 19th century, and you'll hear names, they're names of 19th century mathematicians. So linear algebra itself, that all goes back to 19th century, early 19th century even. But it sort of blends, sort of classical circuits and systems, this would be from 1920s on. This would be the stuff done at, like, Bell Labs in the '20s. So it kind of combines that with linear algebra. So that's what the modern form is.

Now, the first really — the first time you could say here is linear dynamical systems actually either being taught or being used or whatever is actually aerospace in the 1960s, and that would be the first time it was widely used. It was used for — at that time, well, I think you'd have to say it was not used for what we would call socially positive purposes.

It was used to land missiles and things like that. So that was aerospace in the 1960s, very, very fancy, fancy stuff.

But between then and now, somewhere in the '80s, it transitioned from a specialized topic, which 12 or 15 PhD students would take, to one that basically touches all fields, and not just all fields in EE, but also other fields in optimization and finance, in economics and things like that. So that was sort of the transition time. And I've already said the story of digital signal processing, DSP, is the same.

Another story, you can go back down to the pad here, is information theory. That's the same story, that information theory, as you know, was more or less created by Shannon in the late '40s, in fact also by Kolmogorov in the Soviet Union and Moscow in about the same time, maybe even a bit earlier in that case. It was created there, and the people who actually did sort of communications just fell down laughing, saying, "Is this a joke? We could never use that." Like, "Get out of here, Mr. Theorist. We actually have work to do here." So, of course, the joke was on them because you propagate Moore's Law forward a few decades, and you'd be surprised at what you can do now.

So information theory is now not just a super-advanced esoteric topic. It's a topic, like, basically everybody needs to know about, and it is widely used. I mean, it's still also a vibrant — there's a vibrant theoretical core. But it's not just a weird thing done by the 15 PhD students who make this a topic. It's something like everybody ought to know. Okay.

I'll say one more thing about this. I'll say a little bit about non-linear dynamical systems. You hear a lot about this, and I'm deeply suspicious of these people for many reasons. I'll tell you why in a minute, but — so it is absolutely the case that many systems are non-linear. And not only that, non-linear dynamical systems is a fascinating topic. I don't deny it.

So if this is the case, why should you study linear dynamical systems? Well, I'd make a couple of arguments for it. The first is that it turns out that most methods for non-linear systems for engineering things that work for non-linear systems basically are based almost entirely on ideas from linear dynamical systems. And in fact, there's this weird thing where you design things based on linear models that are not even remotely accurate.

And unfortunately — I'll explain why I say unfortunately — unfortunately, these things, more often than they should, work. See, I don't think they — I'll tell you why I think they shouldn't: because it reinforces kind of cowboy engineering is what it — it's kind of like you walk up to someone and say, "What are you doing?" You say, "I'm designing a regulator for this thing." "Is it linear?" "Oh, no, not even close." You go, "But what are you doing?" You say, "I'm designing this. I'm pretending it's linear." So every now and then, something like that should blow up. I mean, just look, I feel — maybe just singe the person who did it a little bit.

But the point is just to send a message, which is you should kind of know what you're doing or at least respect — or at least when it works, step back and say, "Wow. That's cool because it didn't have to work, but it did." But unfortunately, these things just often work even when, like, the models are way off. It's known a little bit why that's the case, but it's just a reason to study linear systems.

The other one is that actually if you're really interested in non-linear dynamical systems, then it turns out if you don't understand linear dynamical systems, I mean, this is basically — this is the big prerequisite as far as I'm concerned. And the other thing is, it's funny, you get people who — there's like a little cult following. This would be the chaos crowd and you can go to the Santa Fe — this place and people will tell you all about this stuff. I can fool any of them. Give me five minutes with any of them. I can fool any of them. They'll say, "Oh, no. Linear systems is not interesting. You can't have bifurcations. You can't have chaos. You can't have —" blah blah blah. I mean, it will take me five minutes to put together a linear dynamical system that will totally fool them. And they'll go, like, "Oh, yeah, that's chaos right there. That's chaos." But it'll just be some linear system. That's just my ideas. You'll hear about this, okay? And actually, it's really interesting stuff, but okay.

So that finishes up my kind of overview of the course. Are there any questions? Kind of be hard to have a question because I didn't say anything. But there might be a question anyway. I guess not. Okay.

What I'm gonna do now is the rest of this lecture I'm actually just gonna go over some examples. Now, this is ideas only. There'll be no details. I do not expect you to understand any of it. All of this, we will come back and do later in the class, all of it, in horrendous detail. So don't try to get everything now. Okay. So what we're gonna do is just look at — here's a specific system.

It's \dot{X} equals AX . Here, the state vector has dimension 16. And we have a scalar output, Y of T . So people on — the slang this would be a — this would be a 16 state single-output system. Now, it turns out this is a model of a lightly damped mechanical system, but it really doesn't matter what it is for now.

But if you want a physical picture of what this is, it is a mechanical system with sort of eight generalized positions, eight generalized momento or something like that, so it's some kind of flex structure. And the dynamics is what happens when you poke it; for example, if an earthquake excites it. And then the whole thing kind of wiggles around. And the output would be the specific displacement in some axis at some point in the structure. So that's what — if you want a physical picture, but you don't need one.

So here's sort of a typical output looks like this. And the first shows what it looks like on a short time scale or shorter time scale, and then this is what it looks like on a longer time scale. Now, the point about this is this looks very — it looks quite complicated. In fact, if I showed you, let's say, just the first 100 samples, just that right there, if I showed you that and said to you — I didn't tell you where it came from, but I said, "Here's a signal

I've observed. Would you mind predicting it?" Now, if I asked you to predict it one or two seconds or whatever the time unit is here in the future, you'd say, "Yeah, no problem, I'll make a prediction."

But what if I said, "Would you mind, having seen this, predicting what this'll be doing at T equals 1000," right? You would hopefully look at this and say, "Are you insane? I mean, I don't even know what that thing is. It's weird. I mean, it's some weird thing. And even it looks like it's changing character over the 50 — let's call it seconds — over the 50 seconds you've shown me. How on earth would I be able to predict what it does 1000 seconds later?" Everybody see what I'm saying here? Okay.

For sure — this is not long enough, but I could get one, and I could definitely get one somebody at the Santa Fe Institute to go for chaos here, no problem. They could say, "Oh, yeah, I'm seeing period doubling, bifurcation, it's all there." Okay.

So the point is you could make all sorts of stories up about this. You'd look at a longer time scale. You could say, "Well, it started off chaotic, but now you can see there's some kind of regularity emerging from this." And you could make all sorts of stories.

So out here, in fact, it would seem reasonable for someone to predict if you saw this what's gonna happen for another 100 seconds or so. It would seem reasonable because some kind of periodicity or something or pattern has set in. Everybody see what I'm saying? Okay.

Now, this is hardly surprising, but it turns out — and this will be familiar to you from the idea of, like, poles and things like that, that this output and indeed the output of any linear dynamical system, it can be spectrally decomposed. It can be decomposed into modes or frequencies, complex frequencies, and the exact one I just showed you, actually, we can write it out this way. There are eight of these. And if you were to add these eight signals up, you would get this one here, okay? Now, this would hardly be surprising to you. I mean, you've sort of seen this maybe, or some of you have seen this in other courses and things like that.

But there's actually — once you know that this is the case, it's a totally different story. If I were to take this signal and decompose it like that, and I now said to you, "Please make a prediction about what the signal will be doing 1,000 seconds in the future," you would probably say, "No problem," because each of these is just a damped sinusoid, and you have a few parameters to fit for each one — that's something else you'll learn in this class — you'd have a few parameters to fit, at which point you could probably quite easily extrapolate it, okay? So again, this is probably not too — it's not too shocking. Okay.

Now, let's look at input design. Input design goes like this. I'm gonna add two inputs. So I'm gonna add two inputs. Now, by the way, if you are thinking — if you're visualizing a mechanical structure, and we're gonna have two outputs, by the way, so you can imagine a mechanical structure, and I wanna put, like, let's say two piezoelectric actuators on it. I mean, just if you wanna make the time scale be a whole building, let's put two hydraulic

actuators on it or something like that. But let's make it two piezoelectric actuators on something this small, that sits here. I have two actuators, and then I measure the displacement of the thing at two other places, okay?

So I have now a two-input, two-output system. You should have a very good idea of what happens when you put a force or a displacement input onto a structure. It will start shaking because you'll excite various modes. And eventually, if it's got damping in it — it does — it will sort of come to rest. And those two points will have been displaced if the input has a constant value or something like that.

So here's the job. We wanna find an input that brings the output to some desired point, 1 minus 2 — this is totally arbitrary because you don't know what A and B is; it's probably not relevant what the output is. So you wanna bring this output here. And so your choice now — this is a generalized — this is, like, a design or a control problem. It's choose an input or action that makes something you want to happen, happen, okay? So that's what you're being asked to do.

By the way, this is quite a realistic problem. Any disc drive has stuff like this in it where you get a command for the head to move from Track 22 to Track 236 and to do it and be tracking in some very small number of milliseconds, okay? So this is quite realistic. If you change the time scale and all that kind of stuff, this is actually very realistic, okay, the problem. Okay.

So here, let's just look at the simple approach. The simple approach is this: Let's assume, if you wait for the whole thing to come to equilibrium, that means everything's constant. That means \dot{X} , which is \dot{DXDT} , is zero. So you have zero is AX plus BU static. I'm putting in a constant input, U . By the way, don't try to follow this. We're gonna go over all this later. And then the output is also constant.

You can solve these linear equations by simply putting AX on the other side, so you have AX equals BU static minus BU static. You multiply by A inverse on the left. But I won't go through the details.

But what you find is you can actually simply work out a formula for what the optimal — what the input should be, not optimal, the input. The input required is this: So let's imagine then that these are, say, piezoelectric displacements. So one of those piezoelectric actuators should pull in, and the other should push out. And by doing that, that will twist the structure in equilibrium so that whatever the — one is displaced, let's say 1 millimeter to the left, and the other is 2 millimeters to the right or whatever this means, okay? So that's the idea. Okay.

Let's just go ahead and apply that input. Let's see what happens. Well, this shows you a little bit of negative time. The U s are zero. And [inaudible] to equal zero, we simply apply this thing, so you have a structure and it's equal zero. One of the piezoelectric actuators shrinks instantly; the other one pushes out. And of course, that causes everything to start shaking in this structure, so everything shakes. And this is what the

outputs do. You can see they shake for a while, and about 1500 time units later, they are converging indeed to what they are supposed to: 1 and minus 2, okay? So there it is.

And you could now make all sorts of arguments about this. You could say, "Well, look, that's the basic dynamics of the system. I mean, come on, the thing shakes. It takes whatever, 1500 seconds or picoseconds depending on what application you're thinking of or microseconds. It takes 1500 microseconds to just get the energy out of the system, to have it dissipate, so how could it ever be any faster?" Okay?

Well, later in the class, we'll look at problems like that, and it turns out you can do a whole lot better. So here are some inputs. Here's one, and here's what you do at one actuator, and here's what you do at the other. And this one, I mean, this is really pretty bizarre. This ends up pushing at minus 0.6, but the first thing it does, it goes in the opposite direction; same for this one. Then it goes through some very complicated dance here, including some little squiggle at the end, okay? So those are the two inputs. Don't worry about how I got them, okay?

This is what happens. So what happens is the outputs kind of wiggle around and 50 seconds later, they hit their target values and they just stay there. So you get exact convergence in 50 seconds, 50 picoseconds [inaudible]. It depends on what your application is. Everybody see this?

Now, by the way, the time scale is this: I mean, maybe I need to draw where that is. That's about right here, okay? That's what happened, okay? So that input basically gets you convergence — first of all, it's exact convergence, but it gets you convergence something like 20 or 100 times faster, so something insane like this, okay? Everybody see this?

Now, notice that the input you're using is not particularly bigger. It's not bigger at all. I mean, the final input you have to put in is minus 0.63. You never really go outside that range much. I wanna say something very important about these two input signals here. No one can look at those and say, "Oh, yeah, I was just about to try that." Okay. There's no way, okay? So this is kind of what this class is about.

So, I mean, you would not have arrived at these two inputs by trial and error, okay? You would not have done this with a proto board in your garage and a box full of different resistor values, let me assure you, okay? You can get someone who's pretty good at wiggling, fiddling with dynamical systems, like a fancy pilot. And a pilot wouldn't do this either, okay?

So now I'll tell you the end of the — I mean, so this is pretty wild. By the way, things like this can be repeated in many, many fields. You can repeat this in finance. You can repeat it in lots of others. These things will beat the hell out of anything you can do, I mean, if it's set up properly, intuitively, okay? So this is kind of what this class is about, is this kind of stuff.

By the way, I will tell you this. This stuff right here, later in the class, by the middle of this class, maybe a little bit beyond the middle of this class, if I show you, you will — I wouldn't even ask you to do this on a homework problem because it would be too insulting to you because you would know — if I asked you, I'd say, "Look. Show me how to find the inputs that move the state from here to here as quickly as possible, and do it in, like 50 whatever or something like that," it would just be insulting. You'd just say, like, "Oh, please." It would be totally obvious to you.

But I want to point out right now — by the way, if there is anyone whose intuition was good enough to think that this was the right thing to do, please speak to me after class. But this is not obvious, okay? This is not intuition or anything like that. In fact, a very important skill in the class is gonna be to explain things like this. What'll happen is this will just come out, and it will be impressive. It'll be in this one — it'll be in other areas. It'll be in signal processing. It'll be in image processing. It'll be in machine learning. Very impressive things will happen.

The problem then is gonna shift not from doing this or knowing how to do this, but to explaining it to somebody. And I'll spend some time periodically through the class explaining how to do this. You can make a very good story about it here. You can look very smart. The truth is you wrote about four lines of code to do this. That's about what it is.

But you can get a lot of mileage out of it. You can say, "It took me a while before I realized you really had to start in the other direction. That's the key, you see, was starting in the other direction. Then for a while, I couldn't get this middle section right. That was tough. Then I realized finally, this one should bump down, sort of like it's going to the final value. But no, it's just to fake out the system because then you come back up here, you see." All right.

So anyway, that's gonna be very important, okay? You'll see. You'll see. That's an important skill. Maybe we'll put that on the final exam, something like that, or we'll have a section where you have to — you will have done something cool in maybe image processing or something, and then after that, you have to explain it to someone and we'll grade it on how plausible it sounds. Okay. It'll all be made up of course. You did it by writing five lines of code. Okay. All right.

Now, it turns out here, you say, "If you can get exact convergence in 50 seconds, how about, like, 20?" Well, you can do 20 as well. And now, if you go to 20, here's what's weird. You now are using a bigger input, okay? And now, by the way, I'll show you something cool. The first thing you do has now shifted again. Now, the first thing you do is you go down, which is gonna be the ultimate way you're going finally, and here you kind of go up, and anyway — so then again, you'd have to make a story about that. And you'd say, "Well, that's totally clear." Why? "Because now you're doing it faster, you see, and so —" anyway, you'd make up some plausible story about that, which the gullible would buy. Okay.

So this raises questions like, how do you do that? How do you find something like that? And how do you analyze this trade-off between the size of U and the convergence time and things like that? So you will know all of that. I mean, this will be, as I said, this will just be so obvious to you six weeks from now, it'll be insulting to even ask. Okay.

We'll look at one more example. This one is from estimation and filtering. So here it is. And this is, you know, if you have a background in signal processing, fine; if you don't, like, if you're in some other department, don't worry about it. So we have a system. It looks like this. It might be part of a communications system, for example, and it's gonna work like this. An input comes in. It's piecewise constant, and we'll make it a period of one second. But if it's a communications system, this might be nanoseconds or something or picoseconds or whatever. It depends what this is, maybe nanoseconds or something.

So an input comes in. It gets filtered or convolved with a linear filter. That's a second-order system with a step response that I'll show in just a minute. Again, if you know this material, great; if you don't, don't worry about it. We'll come back to this. So this is essentially a smooth version of this.

And now the interesting part comes. It's going to be very crudely quantized. It's gonna be quantized with a ten-bit — a three-bit quantizer, okay? There's only eight distinct levels that it knows, okay? So this thing will run at zero hertz, even though the input signal is constant. So here's a picture that just kind of shows how this works. So the input here is zero, and then it's got some value, but it's constant for each interval, which is one second long, okay?

Now, this runs through a filter with this step response, and those of you who remember undergraduate signal processing, if you took it, will know that this means that the impulse response is sort of positive and has a little wiggle here, but you'll look at this and you'll know what that means is it smoothes things on the order of a second or so, or maybe two, or maybe one or two, and it delays things a bit as well.

And so here's this signal after smoothing comes out here, and you can sort of see a bunch of things. I mean, it's zero here. We keep this zero just to make it — so you can see the beginning, and you can see this goes down. This goes down, sure, and it starts going up, but you can see that the individual levels have kind of been smeared out and lost, okay?

And if you know something about communications, you know this is basic communications. Basically, if you're transmitting, like, a signal with multiple layers, levels, and it's coming out where you're reading it nice and clean, it means you're not going fast enough. That's what it means. So it means you should crank up the rate until things start getting hard to guess. Maybe this is too hard to guess because now you've kind of smeared things out. But you can see things. It's a few positives here translates into this kind of big bump here, but if I gave you this, it's not an easy way to estimate what this is.

But it's gonna get worse because the next thing we do is we take this, and we quantize this to eight levels, so that's three bits, and that gives you this sequence here, okay? Each of this is just three bits. And now comes the problem. From this, we want to reconstruct and estimate that. That's the problem, okay? And that's a basic problem in communications. It basically says, let's look at what our receiver, our A to D or whatever it is, is sending us, and you wanna estimate sort of what was sent at the transmitter because from that, you can decode what the message being sent was, okay? So this is sort of very basic in communications or anything like that.

So the basic approach would be something like this. You ignore the quantization because it's — actually you just consider it some imperfection that is beyond your control. And what you'd do is you'd design an equalizer for this. Now, an equalizer is another convolution system, and it has the property — and what you want is you wanna kind of undo, or I guess in, for example, geophysics, this would be called deconvolution, so that would be another area. It'd have lots of names in different fields. You wanna sort of undo the effects of the filter. That's H. So you really want one so the GH is about one. Again, if you know that this means, great; if you don't, that's fine, too.

So once you find such a G, that's called an equalizer. In communications, it would be called an equalizer. And you'll simply apply that to the receive signal to approximate U. If you do that here, it will work hideously. It just simply won't work. You won't get anywhere near close to estimating these quantities. And then if you went back to someone who does traditional signal processing and say, "What's wrong?" and you'd say, "Well, come on, give me a ten-bit A to D, and we're talking. But a three-bit is not gonna do the trick." Everybody see what I'm talking about? So okay.

Well, now it turns out this problem can be posed as a basic problem you'll see in this course, again maybe around the fifth, seventh week, something like that, in which case it'll be just completely standard for you. This one could be a homework problem, I mean, just to set it all up and do it. Maybe it will be.

And if you do that, here's what you'll get. This is an expanded picture. If you make an estimation problem, again, you will write something on the order of five lines of code. You will carry out some simple numerical linear algebra computations on it. And here's how close you'll get to the signal that was transmitted. The dark one shows the actual signal, and the dotted one is our estimate.

Now, the one thing you notice here is that you get pretty close. I mean, you get amazingly close. And in fact, you might ask how well are you doing? And the answer is that the RMS error, the root mean square error, is about 0.03. Now, that's super-duper interesting because the quantization levels, each step is about 0.25 wide, okay? So that's means the quantization errors are, like, plus/minus 0.125. And we are predicting it better than, I mean, roughly speaking, better than four times the range of the quantizer.

Are you disturbed? Who's disturbed by this? Good. You should be disturbed. Now, for the rest of you, why are you not disturbed by this? This is weird. I think there's something

wrong with your intuitions or something like that. But no, I mean, come on, you gotta admit this is kind of weird, right?

I mean, I give you this measuring device that basically makes errors of plus/minus, let's say, 125 millivolts. I give you these measurements, I mean, that's very crude. And then you go off and do something, which you'll know how to do, let's say, the seventh week. And you come back, and you actually estimate what happened from those crappy measurements. You actually get something that's typically off by around 30 millivolts. You're still not disturbed. I can tell. You're just, like, totally okay with this. Well, is this okay with you? You're, like, cool, that's fine. Why? Student:

[Inaudible]. Instructor:

Oh, okay, but you already bought the whole package and everything. Right. Okay. All right. But no, yeah, you'll know how to do this, and you'll yawn, I mean, by the seventh week at things like this. Okay. But you already bought the whole package, so you believe it all. Yeah, I'm gonna fix you guys. You know what I'm gonna do? I'm gonna come in with something that's totally wrong. I will say it with a totally straight face, and we'll just see how far I can push. You better be on your toes. That's all I have to say. Okay. All right.

Well, since you're not upset by that, I mean, if it's just, like, okay, cool, fine, theory can do anything, well, we'll just move on and we'll actually start the course proper, I mean, unless there's some questions or — this would be a good time to answer any questions about — I don't know — the basics of the class or what we're gonna do and stuff like that. So okay.

Oh, I do have one question. How many people here — we are gonna use MATLAB. I'm not a huge fan of MATLAB, but we're gonna be writing, you know, the key is you're gonna write very short things. Okay. How many people here sort of have used MATLAB before at some point? Okay. So I think that's a whole lot. We are gonna be — so you'll be fine.

The stuff we're gonna be doing is gonna be very, very basic. If you ever write a script longer than 15 lines or something like that or, I mean, if the main part of it is more than 15 lines, you're probably doing something wrong. So it's not, I mean, the key in this class is that the ratio of thinking to programming we want to be very high. So if you find yourself writing five pages and deep — if you find yourself doing anything that you could actually describe as programming, you're probably approaching this the wrong way, just to let you know. That's generally defined as more than one screen of source code, I would say, so if you're on that second or third page, something's wrong. So that's part of the class.

For those of you who, I mean, I think a lot of this you can just come up to speed on yourself, but maybe we'll convince the TAs if some of you want to just sort of have a quick introduction, if some people need this. Okay.

So let's start the course proper. I should say about the course, the first couple of lectures are gonna be review. So if you have an exposure to linear algebra, you will think it's embarrassingly and insultingly elementary. But don't worry. We'll get there. It's a little bit boring, and it picks up somewhere around, you know, it'll pick up in a couple of weeks. So okay.

So we'll start with the idea of linear functions and examples. And this presumably you've seen before. So this'll just be review, and what I will do though is focus on sort of what the meaning of all of this is, which is — and what the implications are, and that's the part that is traditionally left out of a linear algebra course, a traditional one. Okay.

So let's suppose you have a system of linear equations, so this is linear equations. So you have $Y_1 = A_{11}X_1 + A_{12}X_2 + \dots + A_{1N}X_N$ up to plus A_{1N} and X_N . And then you have a bunch of these equations. In fact, we have M of these equations. So these are — everything you see here is a scalar. Well, you don't get very far, it turns out, by using complicated notation all the time. So it's actually — with matrix notation, this goes down to four ASCII characters, which is $Y = AX$.

Now, here, Y is just the vector of these Y s stacked up to make a column vector. X is the column vector of these X s, and A is this matrix, which is M by N . So that's, I mean, this you should have seen, so this is a matrix form. And you should understand — in fact, that's kind of a theme in the class. I mean, so what's actually gone on here is we're heavily using overloading here. So everyone knows what $Y = AX$ means when these are scalars. And here, we have overloaded that to the case where X is a vector and A is a matrix, okay? So that's the idea here. Okay.

Now, this you've also seen. I hope, presume, guess that you've seen the idea of a linear function in lots and lots of courses, I hope. But here's what it is: A function is linear if F , the function, which maps \mathbb{R}^N to \mathbb{R}^M — this notation means that this function accepts as argument an N vector and returns an M vector. Now, this says $F(X + Y) = F(X) + F(Y)$. When you read these things, I should add, they're very easy on the eye because it just looks super simple. It kind of looks like just the plus went in a different place. It's just not that big a deal. It just looks very casual. It even looks pretty and simple. This is actually making a very strong statement, this statement.

And in fact, I think I'll do this once, yeah, I'm gonna do it. So we'll do it once. It's actually very useful in the notation we're gonna be using to make sure you can parse what you're looking at, and in fact for everything you write, you better be sure you can parse what you're writing because a parsing error, a syntax error is a bad thing.

So let's look at this. So first, let's actually talk about the data type. So I presume you've all had a class in basic computer programming, so you might wanna think about the actual data type of these things. So let me just ask a couple of questions here.

What is X ? Is it a matrix? Is it a number? What is it? It's an N vector, okay? All right then. What is this plus? What plus is that? What plus is it?

Student:[Inaudible].

Instructor (Stephen Boyd):It's a vector addition. So plus is being overloaded here. I mean, everyone knows what plus is between real numbers, complex numbers. This is plus between vectors. That's an overloaded plus here. What is the data type of X plus Y ? It's an N vector. All right. F is a function. Now you can do the syntax check on F . F is being passed an N vector. Is that correct? Well, sure, that's what this says. It says F has been declared to be a function that accepts this argument, an N vector. It is being passed an N vector. And so what is the return value? M vector. Okay. What plus is this?

Student:[Inaudible].

Instructor (Stephen Boyd):Yeah, which vector addition?

Student:[Inaudible]. Instructor:

Precisely. That's M vector addition. Okay. And what equals is that? Student:

[Inaudible]. Instructor:

It's M vector equals. Okay. So all of this is heavily overloaded, but I'd encourage you to do this because otherwise, what happens is you start writing stuff down, very short formulas, and they look pretty and beautiful. But watch out because the point of powerful notation is precisely that you can write totally innocent-looking things down that are five lines, and they'll actually be saying a lot. I mean, that's the good and the bad news about it. Okay.

The second is that F , you can scale X either before it's operated on by F or after, and you get the same thing. So a lot of people write this this way: αX plus βY equals αF of X plus βF of Y , like that, that that should hold. And that's one of the definitions of linearity, okay?

And this is a picture, you know, basically — actually what the picture says is this: And, by the way, this is really stupid. It already had major implications. I mean, I guarantee you I can dress this up in some story where it won't look like this, but it will be about this, and it will have serious implications in that context. Okay.

So here, what it says is this: You have a vector X and a vector Y , and X ends up getting operated on and comes out over here. That's F of X . And Y gets operated on and comes out over here. And the question then is what would happen if this vector, which is the vector sum of X and Y , were passed to F . And the answer is you can find that two ways. You can either operate on this point with F , in which case it will come out over here; or you could actually take these two things and add them and come out over here. This, by the way, already has lots of implications because, for example, if F is a simulation, this says you actually can make three predictions at the cost of two simulations. Everybody

see what I'm saying here? I'm assuming a simulation is expensive, but vector addition is cheaper. So okay. But this, you should have seen. That's a linear function.

And here are some examples. Here's one. Here's a function. It takes as argument a vector X and it multiplies this X is in \mathbb{R}^N , and it multiplies by an M by N matrix, and that multiplies something that's in \mathbb{R}^M . That's an explicit function. It's parameterized by the matrix A . Turns out that's a linear function you can just check very quickly. But here's the cool part. That's actually all the linear functions. That's a generic form. It turns out if a function is linear, it has to have the form F of X equals AX , okay?

Now, when you look at that, it looks very sophisticated and complicated and mathematical. It's actually not. It's actually quite straightforward. But I think I'll say one thing about it, and then we're gonna quit.

So it's actually quite straightforward to work out what A is, and this you should think of. I think this is on your first homework, you should think of this as someone putting on the table in front of you a black box with a bunch of BNC connectors on the left, in fact specifically N on the left, and a bunch of N BNC connectors on the right. And then you're just in a lab. And the question is, how would you make a model of that box in your lab slot so that later when you're not in the lab, you can actually predict what it will do, actually for any input, and of course given that it's linear? So okay. So I think we'll quit here and then continue on Thursday.

[End of Audio]

Duration: 77 minutes