

Lecture 15 - Notes

Rahul Arya

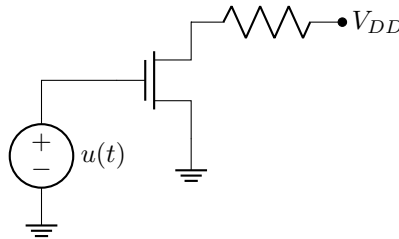
March 2019

1 Overview

So far, we have derived a number of very interesting results when considering linear systems. We know how to apply inputs to control them towards a particular state, how to determine the initial state based on observing the behavior of the system, and how to determine the state equation itself by making a few observations about the nature of the system. We also know when a system will naturally remain near an equilibrium point (by considering stability) and, if it does not, when we can apply control feedback to ensure that our system does effectively become stable.

However, all of these results rely on a fundamental assumption - that our system is in fact linear - intuitively, that its behavior is in some way “proportional” to the applied input. Most real-world systems, unfortunately, do not exhibit this desired property, especially when the inputs applied are large. As an example, consider the behavior of a simple pendulum. For small inputs, the pendulum’s response to input (like an impulse applied at the middle of the pendulum’s arc) is linear - doubling the size of the input will double the size of the pendulum’s arc¹. But for large inputs, the pendulum’s behavior will become fundamentally different - it may swing round in a full circle once, or swing up partially until the string goes slack, or something else entirely! No linear model can represent all the possible behaviors of even this very simple system.

Another, perhaps more relevant, example is that of a single transistor, as in the following circuit:



¹Not really, but this isn’t a physics class. The amplitude will actually keep growing as we pump energy into the system, but this example is not meant to be accurate, only to provide some small amount of intuition.

Notice that the behavior of the transistor (modelling it as a voltage controlled switch) changes abruptly as $u(t)$ passes a certain threshold. So even this simple and very common setup cannot be treated as a linear system.

If we can't apply it to even the simplest examples, what is the use of all the control theory that we've been studying? We will show in this lecture and in the next that we can accurately model nonlinear systems (which are hard to deal with directly) as linear systems, and then apply all the machinery of control theory to solve problems.

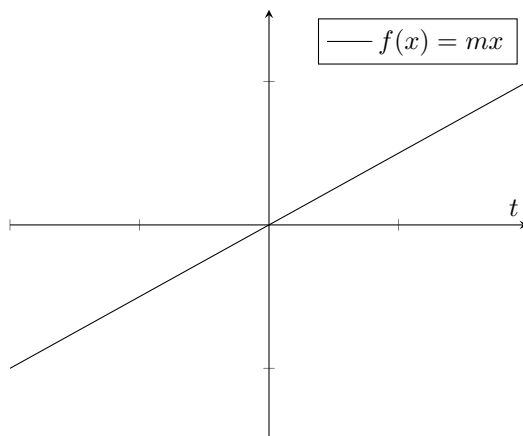
2 Linearity

First, of course, we need to construct a more formal notion of linearity - merely stating that the output is in some way "proportional" to the input is certainly not sufficient. We define a function f between vector spaces to be linear if it has the following two behaviors:

- *Scaling*: For any scalar α (typically a real or complex value) and for any element of the domain x , $f(\alpha x) = \alpha f(x)$.
- *Superposition*: For any two elements x and y in the domain, $f(x + y) = f(x) + f(y)$.

These properties apply whether the domain of f consists of scalars or vectors². An immediate consequence of scaling is that $f(0) = 0$, by considering $\alpha = 0$.

Now, we will look at an example of how these properties apply to scalar functions (functions taking a scalar input and yielding a scalar output). Consider the archetypal linear function - a straight line passing through the origin, like as follows:



²As for this purpose we can view a field of scalars as a vector space.

Why is this function linear? Observe that, for any scalars α and x ,

$$f(\alpha x) = \alpha m x = \alpha f(x),$$

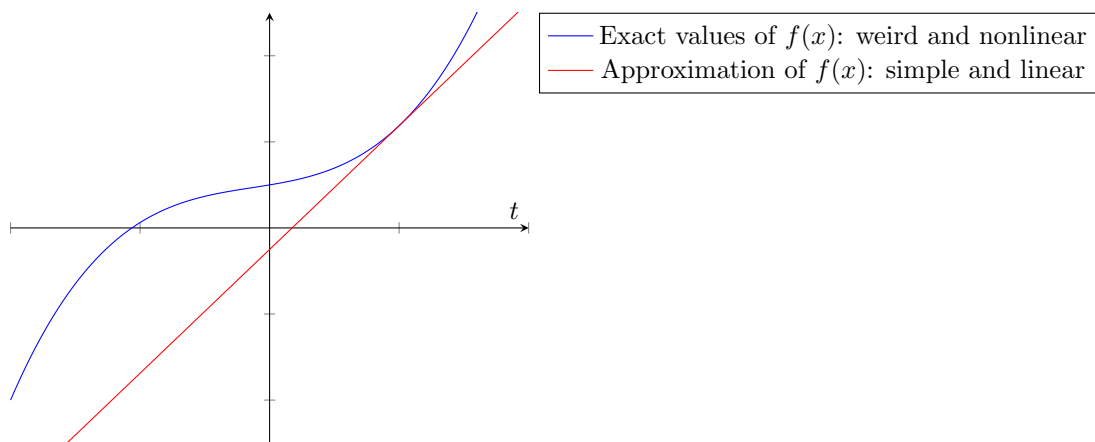
so f satisfies the scaling property. And for any two scalars x and y ,

$$f(x + y) = m(x + y) = mx + my = f(x) + f(y),$$

so f also satisfies the property of superposition. And, though this is not explicitly part of the definition of linearity, it's nice to see that $f(0) = m \cdot 0 = 0$, as we expected.

3 Linear Approximations of Scalar Functions

Now, let's look at how we can approximate a nonlinear function by a linear function. From calculus, we know that the tangent line at a point serves as a good approximation for a function, as shown:



Notice that the tangent line is only really a good approximation near the point at which it is tangent - away from that point, it tends to become increasingly worse. More formally, we call the point $(x^*, f(x^*))$ where the tangent is constructed the *expansion point* of our approximation. Clearly, at $x = x^*$, the tangent line perfectly matches the actual value of our function.

Algebraically, we may express our approximation as follows:

$$f(x) \approx f(x^*) + m(x - x^*),$$

where m is the slope of f at $x = x^*$ (so $m = \left. \frac{df}{dt} \right|_{x=x^*}$).

At this point, there's one thing that might seem a bit odd about our supposedly "linear" approximation - it's not linear! Recall that a linear function must pass

through the origin. But at $x = 0$, our approximation has the value $f(x^*) + m(0 - x^*) = f(x^*) - mx^*$, which may not equal 0.

Fortunately, we can address this through a straightforward change of coordinates. Let $\delta x = x - x^*$, and $\delta f = f(x) - f(x^*)$. We therefore may consider the linear model

$$\delta f(\delta x) = m\delta x,$$

which is indeed linear, as we desired. In some sense, you can visualize this change of coordinates as the movement of the origin to the point $(x^*, f(x^*))$, so the tangent line clearly passes through $(0, 0)$ in these new coordinates.

4 Linear Functionals

With tools for linearizing scalar functions in hand, we can turn to the problem of linearizing a scalar system with inputs. For simplicity, we will use the notation of continuous-time systems (e.g. $u(t)$ rather than $u[t]$) but the same exact principles apply.

To linearize a scalar system, we need to consider what a linear system really is. Fundamentally, a system takes an input function $u(t)$, and produces an observation function $y(t)$. Let this mapping from input functions to a scalar observation at a particular time be \mathcal{L} .

What exactly is \mathcal{L} , and what does it mean for \mathcal{L} to be linear? It is natural to view \mathcal{L} as a function. But all the functions we've seen take scalars or vectors as input and produce scalars or vectors as output. What does it mean for \mathcal{L} to take, not a scalar or vector, but another function as input?

For our purposes, we will treat \mathcal{L} not as a function, but as a new object known as a *functional*³. While functions take a finite set of values as input, functionals take an infinite set of values as input in order to produce a scalar value. More concretely, a functional \mathcal{L} taking a function $u(t)$ as input will have “access” to *all* the values $u(t)$ for all $t \in (-\infty, \infty)$, in order to compute its output $y(t)$ at a particular value of t .

³For those of you who have taken Math 54 or otherwise have a background in linear algebra, yes, a functional is no different from any other function, except that its domain is the vector space of functions, not scalars or Euclidean vectors. But for the purposes of this class, we will restrict our definition of functions to mappings between scalars or Euclidean vectors.

Also, it is true that the modern definition of a functional is *any* mapping between a vector space and a scalar field. However, a slightly older definition of functional seems to require that the domain be a vector space of functions, and that appears to be the definition used in this course.

You may also have heard of the term *linear functional* restricted to refer to a mapping from a vector space to a field of scalars, with the set of linear functions of a particular vector field collectively known as the *dual space*. Members of the dual space are certainly functionals by the modern definition of functionals, but are referred to as *linear forms* to avoid ambiguity, and to ensure consistency with our older definition of functionals.

The Wikipedia article [https://en.wikipedia.org/wiki/Functional_\(mathematics\)](https://en.wikipedia.org/wiki/Functional_(mathematics)) has a very concise summary of all these various definitions, and is absolutely worth reading.

This definition can be made more tractable through an example. Consider the functional \mathcal{L}_1 defined as follows:

$$\mathcal{L}_1\{u(t)\} = \int_{\tau-1}^{\tau} u(t) dt.$$

Here, \mathcal{L}_1 integrates its input function $u(t)$ across the interval $[\tau - 1, \tau]$. To do so, \mathcal{L}_1 needs to perform a computation based on all the infinitely many values of $u(t)$ within that interval, in order to evaluate its integral. Providing any finite number of values of $u(t)$ (such as just the value $u(\tau)$) would clearly be insufficient.

To indicate this behavior using appropriate notation, we use curly brackets ($\{$ and $\}$) in order to demarcate the inputs to \mathcal{L}_1 . Writing $\mathcal{L}_1(u(t))$ (with rounded parentheses) would imply that we pass just the value of $u(t)$ at the point t to our functional, which is meaningless. Instead, by writing $\mathcal{L}_1\{u(t)\}$, we indicate that we are passing the *function* $u(t)$ (or equivalently, the values of $u(t)$ for all values of t) to our functional, which is what we intend.

Now that we have some understanding of functionals, we can ask what it means for a functional (as opposed to a function) to be linear. Well, we know that linearity is equivalent to satisfying the properties of scaling and superposition. Expressing both of these properties analogously to how we did so for functions, we find that a linear functional \mathcal{L} has the following two behaviors:

- *Scaling*: For any scalar α and for any function $u(t)$, $\mathcal{L}\{\alpha u(t)\} = \alpha \mathcal{L}\{u(t)\}$.
- *Superposition*: For any two functions $u_1(t)$ and $u_2(t)$, $\mathcal{L}\{u_1(t) + u_2(t)\} = \mathcal{L}\{u_1(t)\} + \mathcal{L}\{u_2(t)\}$.

These definitions seem fairly straightforward.

There is one slight subtlety, however - what does it mean to add two functions together, or to multiply a function with a scalar? For the sake of completeness, we define

$$g(x) = \alpha f(x) \iff \forall x (g(x) = \alpha f(x))$$

and

$$h(x) = f(x) + g(x) \iff \forall x (h(x) = f(x) + g(x)),$$

where $f(x)$, $g(x)$, and $h(x)$ are all scalar functions, and α is an arbitrary scalar constant⁴.

⁴If you remember EE16A, you may be asking yourself - if functions can be added together or multiplied by a scalar in order to obtain new functions, then is the space of all scalar single-argument functions a vector space (since it's closed under scalar multiplication and vector addition)? The answer is yes, and it's not hard to go through all the properties of vector spaces and demonstrate that functions satisfy each one of them! But a detailed understanding of functions as vectors in a vector space beyond what we have discussed is not required at this point in the course.

5 Linearizing Nonlinear Scalar Systems

Finally, now that we understand what linear functionals are, we can see more concretely how they relate to linear systems. Consider the scalar differential equation

$$\frac{d}{dt}(x(t)) = kx(t) + u(t),$$

with the initial condition $x(0) = 0$. We know that there exists some functional \mathcal{L} that produces the value of x at some time t , given the input function $u(t)$ at all times t . Let's try to show that \mathcal{L} is linear.

One way would be to use the closed-form expression for \mathcal{L} that we saw in discussion and on the homework. But that form is hard to derive, unwieldy, and (most importantly) does not generalize well to the multidimensional vector case, or to discrete-time systems.

Instead, we will show the properties of scaling and superposition directly. Let $x_1(t) = \mathcal{L}\{u_1(t)\}$ and $x_2(t) = \mathcal{L}\{u_2(t)\}$, for two input functions $u_1(t)$ and $u_2(t)$. By definition,

$$\begin{aligned} \frac{d}{dt}(x_1(t)) &= kx_1(t) + u_1(t) \\ \frac{d}{dt}(x_2(t)) &= kx_2(t) + u_2(t) \\ \implies \frac{d}{dt}(x_1(t) + x_2(t)) &= k(x_1(t) + x_2(t)) + (u_1(t) + u_2(t)). \end{aligned}$$

Thus, $x_1(t) + x_2(t)$ satisfies the differential equation with input $u_1(t) + u_2(t)$. Since $x_1(0) + x_2(0) = 0$, it also satisfies our initial condition. Therefore, by definition,

$$\mathcal{L}\{u_1(t) + u_2(t)\} = x_1(t) + x_2(t) = \mathcal{L}\{u_1(t)\} + \mathcal{L}\{u_2(t)\},$$

so we have proven the property of superposition.

Similarly, for an arbitrary scalar α , letting $x(t) = \mathcal{L}\{u(t)\}$ for some input function $u(t)$, from the definition we have

$$\begin{aligned} \frac{d}{dt}(x(t)) &= kx(t) + u(t) \\ \implies \frac{d}{dt}(\alpha x(t)) &= k(\alpha x(t)) + \alpha u(t). \end{aligned}$$

Thus, $\alpha x(t)$ satisfies the differential equation with input $u(t)$. Since $\alpha x(0) = 0$, it also satisfies our initial condition. Therefore, by definition,

$$\mathcal{L}\{\alpha u(t)\} = \alpha x(t) = \alpha \mathcal{L}\{u(t)\},$$

so we have proven the property of scaling. As \mathcal{L} satisfies both the properties of superposition and of scaling, it is a linear functional.

Now, we can apply what we've learned in order to linearize a nonlinear scalar system. One form (though it is not the most general form) of a continuous scalar system is

$$\frac{dx}{dt} = f(x) + bu(t),$$

where b is a scalar constant, $x(t)$ is a scalar state that varies with time, $u(t)$ is our input, and $f(x)$ is a nonlinear function that we know to describe the dynamics of our system.

Since we see a nonlinear scalar function in this expression, we can try to use our techniques for linearizing a scalar function. To do so, we need to pick an expansion point x^* , near which we can accurately approximate $f(x)$. But how to find this expansion point?

One way would be to just pick an expansion point directly. For instance, if we're trying to stabilize our system around a particular point, then it would make sense to choose that point as the expansion point. This is often done when we are trying to control our system, and so can choose our inputs appropriately.

However, in our study of control, it is frequently the case that we are merely trying to determine the behavior of a system responding to inputs out of our control. For instance, we saw in previous lectures that circuit analysis in the time domain can be thought of as determining the behavior of a system (representing a circuit) responding to inputs. Clearly, in such a scenario, we cannot just arbitrarily pick an expansion point, and assert that our system will remain near that expansion point as inputs are applied.

Instead, the key idea will be to recognize that our input function is certainly known to us, and that it will typically remain around a certain value u^* . This u^* is known as a *fixed point*, *equilibrium point*, or, in circuit analysis, as a *DC operating point*. The last of these terms is particularly illustrative, as often our circuit inputs will be sinusoidal functions oscillating around a fixed DC offset.

Now that we know this DC operating point, we can solve for our expansion point x^* , by considering the value of the state that will remain stable when our DC input u^* is applied. Algebraically, we can solve the equation

$$0 = \frac{dx}{dt} = f(x^*) + bu^* \implies f(x^*) = -bu^*$$

for x^* .

Intuitively speaking, we should expect that when we vary our input slightly around u^* , that our state x should remain close to x^* even after a long period of time. This notion sounds very similar to stability, and we will discuss these connections more thoroughly in the future. Assuming for now that this assumption of stability is true, we may construct the approximation to our system:

$$\frac{dx}{dt} \approx f(x^*) + m(x - x^*) + b(u^* + u(t) - u^*),$$

where m is the slope of f at x^* , as defined previously. Simplifying using the construction of our DC operating point, we obtain

$$\begin{aligned}\frac{dx}{dt} &\approx f(x^*) + m(x - x^*) + b(u^* + u(t) - u^*) \\ &= (f(x^*) + bu^*) + m(x - x^*) + b(u(t) - u^*) \\ &= m(x - x^*) + b(u(t) - u^*).\end{aligned}$$

Since $\frac{dx}{dt} = \frac{d}{dt}(x(t) - x^*)$, and defining δu in a similar manner to our previous definition of δx , we finally obtain the linearization

$$\frac{d(\delta x)}{dt} = m(\delta x) + b(\delta u),$$

which is valid when our input u and state x remain close to u^* and x^* , respectively.

6 Example

Finally, we can look at a simple example. Consider the scalar nonlinear differential equation

$$\frac{dx}{dt} = -x^3 + u(t).$$

In our terminology, we see that $f(x) = -x^3$ and $b = 1$. Imagine that we are told that our input $u(t)$ will remain around $u^* = 1$. We can compute our DC operating point x^* by solving the equation

$$\begin{aligned}f(x^*) &= -bu^* \\ \implies -(x^*)^3 &= 1 \\ \implies x^* &= -1.\end{aligned}$$

It is interesting to note at this stage that, although it wasn't too hard in our example, finding x^* is typically the hardest part of linearizing a system. All the other steps can be done fairly mechanically, but solving a nonlinear equation, possibly with multiple unknowns, can get very difficult.

Regardless, now that we have x^* , we can find the slope m of $f(x)$ near x^* , to be

$$m = \left. \frac{df}{dx} \right|_{x=x^*} = -3(x^*)^2 = -3.$$

Thus, using the definitions of δx and δu , we find that

$$\frac{d(\delta x)}{dt} = -3\delta x(t) + \delta u(t).$$

We know how to solve for $\delta x(t)$ as a function of $\delta u(t)$, so we can ultimately obtain an approximation for $x(t)$ as a function of $u(t)$, so long as we remain near our DC approximation.

Next time, we will see how to generalize these techniques to state equations with vector (rather than scalar) states, in order to linearize arbitrary systems.