

CHAPTER 10

Linear Time-Invariant (LTI) Models for Communication Channels

In order to preview what this chapter is about, it will be helpful first to look back briefly at the territory we have covered. The previous chapters have made the case for a digital (versus analog) communication paradigm, and have exposed us to communication at the level of *bits* or, more generally, at the level of the symbols that encode messages.

We showed, in Chapters 2 and 3, how to obtain compressed or non-redundant representations of a discrete set of messages through source coding, which produced codewords that reflected the inherent information content or entropy of the source. In Chapter 4 we examined how the source transmitter might map a bit sequence to clocked *signals* that are suited for transmission on a physical channel (for example, as voltage levels). Chapter 5 considered the challenges of “demapping” back from the received noise-corrupted signal to the underlying bit stream, assuming that the channel introduced no deterministic distortion, only additive white noise on the discrete-time samples of the received signal. This led us to the binary symmetric channel (BSC) abstraction for the channel, with some associated probability of corrupting an individual bit on passage through the channel, independently of what happens to every other bit in the sequence.

We then studied, in Chapters 6, 7, and 8, how to re-introduce redundancy, but in a controlled way using parity bits. This resulted in error-detecting and error-correcting codes, or channel codes, that provide some level of protection against the bit-errors introduced by the BSC. That line of development culminated, in Chapter 9, with an outline of Shannon’s fundamental and seminal results on the maximum attainable rate of information transmission through the BSC.

The present chapter begins the process — continued through several subsequent chapters — of representing, modeling, analyzing and exploiting the characteristics of the physical transmission channel. This is the channel seen between the signal transmitted from the source and the signal captured at the receiver. Referring back to the upper portion of Figure 4-8 in Chapter 4, our intent is to study in more detail the key portion of the communication channel represented by the connection between “Mapper + Xmit samples” at the source side, and “Recv samples + Demapper” at the receiver side. These chapters reference individual slides from the posted lectures as “Slide x.y”, for Slide y of Lecture x.

■ 10.1 Getting the Message Across

■ 10.1.1 The Baseband Signal

In Slide 10.2 we see an expanded version of what might come between the channel coding operation at the source and the channel decoding operation at the receiver (as described in the upper portion of Figure 4.8). At the source, the first stage is to convert the input *bit* stream to a digitized and discrete-time (DT) *signal*, represented by samples produced at a certain sample rate f_s samples/s. We denote this signal by $x[n]$, where n is the integer-valued discrete-time index, ranging in the most general case from $-\infty$ to $+\infty$.

In the simplest case, which we will continue to use for illustration, each bit is represented by a signal level held for a certain number of samples, for instance a voltage level of $V_0 = 0$ held for 8 samples to indicate a 0 bit, and a voltage level of $V_1 = 1$ held for 8 samples to indicate a 1 bit, as in Slide 10.4. (The sample clock in this case operates at 8 times the rate of the bit clock, so the bit rate is $f_s/8$ bits/s.) Such a signal is usually referred to as the **baseband** signal.

■ 10.1.2 Modulation

The DT baseband signal shown in Slide 10.4 is typically not ready to be transmitted on the physical transmission channel. For one thing, such channels typically operate in continuous-time (CT) analog fashion, so at the very least one needs a digital-to-analog converter (DAC) to produce a continuous-time signal that can be applied to the channel. The DAC is usually a simple *zero-order hold*, which maintains or holds the most recent sample value for a time interval of $1/f_s$. With this, the DT “rectangular-wave” in Slide 10.4 becomes a CT rectangular wave, each bit now corresponding to a signal value that is held for $8/f_s$ seconds.

Conversion to an analog CT signal will not suffice in general, because the physical channel is usually not well suited to the transmission of rectangular waves of this sort. For instance, a speech signal from a microphone may, after appropriate coding for digital transmission, result in 64 kilobits of data per second (a consequence of sampling the microphone waveform at 8 kHz and 8-bit resolution), but a rectangular wave switching between two levels at this rate is not well suited to direct radio transmission. This is because efficient projection of wave energy requires antennas of dimension comparable with the wavelength of the signal, typically a quarter wavelength in the case of a tower antenna. At 32 kHz, corresponding to the waveform associated with alternating 1’s and 0’s in the coded microphone output, and with the electromagnetic waves propagating at 3×10^8 meters/s (the speed of light), a quarter-wavelength antenna would be $3 \times 10^8 / (4 \times 32 \times 10^3) = 2344$ meters long!

Even if we could arrange for such direct transmission of the baseband signal (after digital-to-analog conversion), there would be issues related to interference between this transmission and everyone else’s. Regulatory organizations such as the Federal Communications Commission (FCC) impose constraints on transmissions, which further restrict what sort of signal can be applied to a physical channel.

In order to match the baseband signal to the physical and regulatory specifications of a transmission channel, one typically has to go through a **modulation** process. This converts the digitized samples to a form that is better suited to transmission on the available chan-

nel. Consider, for example, the case of direct audio transmission from the speaker on your laptop to the microphone on your laptop, Slide 10.10. The speaker does not respond effectively to the piecewise-constant voltages that arise from our baseband signal. It is instead designed to respond to *oscillatory* voltages at frequencies in the appropriate range, producing and projecting a wave of oscillatory acoustic pressure. Excitation by a sinusoidal wave produces a pure acoustic tone. With a speaker aperture dimension in the range of 5 cm or 0.05 meters, and a sound speed of around 340 meters/s, we anticipate effective projection of tones with frequencies in the low kilohertz range, which is indeed in ! (the high end of) the audible range.

A simple way to accomplish the desired modulation in the acoustic wave example above is to apply — at the output of the digital-to-analog converter, which feeds the loudspeaker — a voltage $V_0 \cos(2\pi f_c t)$ for some duration of time to signal a 0 bit, and a voltage of the form $V_1 \cos(2\pi f_c t)$ for the same duration of time to signal a 1 bit¹. Here $\cos(2\pi f_c t)$ is referred to as the *carrier signal* and f_c is the *carrier frequency*, chosen to be appropriately matched to the channel characteristics. This particular way of imprinting the baseband signal on a carrier by varying its amplitude is referred to as *amplitude modulation* (AM), which we will study in more detail in a later chapter. The choice $V_0 = 0$ and $V_1 = 1$ is also referred to as *on-off keying*, with a burst of pure tone (“on”) signaling a 1 bit, and an interval of silence (“off”) signaling a 0.

One could also choose $V_0 = -1$ and $V_1 = +1$, which would result in a sinusoidal voltage that switches phase by $\pi/2$ each time the bit stream goes from 0 to 1 or from 1 to 0. This may be referred to as *polar keying* (particularly when it is thought of as an instance of amplitude modulation), but is more commonly termed *binary phase-shift keying* (BPSK). Yet another modulation possibility for this acoustic example is *frequency modulation* (FM), where a tone burst of a particular frequency in the neighborhood of f_c is used to signal a 0 bit, and a tone burst at another frequency to signal a 1 bit.

■ 10.1.3 Demodulation

We shall have more to say about demodulation later, so for now it suffices to think of it as a process that is inverse to modulation, aimed extracting the baseband signal from the received signal, Slide 10.11. While part of this process could act directly on the received CT analog signal, the block diagram on Slide 10.2 shows it all happening in DT, following conversion of the received signal using an analog-to-digital converter (ADC). The block diagram also indicates that a filtering step may be involved, to separate the channel noise as much as possible from the signal component of the received signal, as well as to compensate for deterministic distortion introduced by the channel. These ideas will be explored further in later chapters.

■ 10.1.4 The Baseband Channel

The end result of the demodulation step and any associated filtering is a DT signal $y[n]$, comprising samples arriving at the rate f_s used for transmission at the source. We assume issues of clock synchronization are taken care of separately.

¹A zero-order-hold DAC will produce only an approximation of a pure sinusoid, but if the sample rate f_s is sufficiently high, the speaker may not sense the difference.

In the ideal case of no distortion, no noise on the channel, and insignificant propagation delay, $y[n]$ would exactly equal the modulating baseband signal $x[n]$ used at the source, for all n . If there is a fixed and known propagation delay on the channel, it can be convenient to simply start the clock at the receiver that much later than the clock at the sender. If this is done, then again we would find that in the absence of distortion and random noise, we get $y[n] = x[n]$ for all n .

More realistically, the channel does distort the baseband signal, so the output DT signal may look (in the noise-free case) as the lower waveform in Slide 10.9. Our objective in what follows is to develop and analyze an important class of models, namely linear and time-invariant (LTI) models, that are quite effective in accounting for such distortion, in a vast variety of settings. The models would be used to represent the end-to-end behavior of what might be called the *baseband channel*, whose input is $x[n]$ and output is $y[n]$, as in Slide 10.8.

■ 10.2 Linear Time-Invariant (LTI) Models

■ 10.2.1 Baseband Channel Model

Our baseband channel model, as represented in the block diagram in Slide 10.13, takes the DT sequence or signal $x[.]$ as input and produces the sequence or signal $y[.]$ as output. We will often use the notation $x[.]$ — or even simply just x — to indicate the entire DT signal or function, see Slide 10.14. Another way to point to the entire signal, though more cumbersome, is by referring to “ $x[n]$ for $-\infty < n < \infty$ ”; this often gets abbreviated to just “the signal $x[n]$ ”, at the risk of being misinterpreted as referring to just the value at a single time n .

The block diagram in Slide 10.13 shows $x[n]$ at the input and $y[n]$ at the output, but that is only to indicate that this is a snapshot of the system at time n , so indeed we see $x[n]$ at the input and $y[n]$ at the output of the system. What the diagram should **not** be interpreted as indicating is that the value of the output signal $y[.]$ at time n depends exclusively on the value of the input signal *at that same time* n . In general, the value of the output $y[.]$ at time n , namely $y[n]$ could depend on the values of the input $x[.]$ at *all* times. We are most often interested in **causal** models, however, and those are characterized by $y[n]$ only depending on *past and present* values of $x[.]$, i.e., $x[k]$ for $k \leq n$.

■ 10.2.2 Unit Sample Response $h[n]$ and Unit Step Response $s[n]$

There are two particular signals that will be very useful in our description and study of LTI channel models. The **unit step** signal or function $u[n]$ is defined in Slide 10.15; it takes the value 0 for negative values of its argument, and 1 everywhere else. Thus $u[1 - n]$, for example, is 0 for $n > 1$ and 1 elsewhere.

The **unit sample** signal or function $\delta[n]$ is defined in Slide 10.16; it takes the value 1 when its argument is 0, and 0 everywhere else. Thus $\delta[n - 3]$ is 1 where $n = 3$ and 0 everywhere else. One can also deduce easily, see Slide 10.17, that

$$\delta[n] = u[n] - u[n - 1], \quad (10.1)$$

where addition and subtraction of signals such as $u[n]$ and $u[n - 1]$ are defined “point-

wise”, i.e., by adding or subtracting the values at each time instant. Similarly, the multiplication of a signal by a scalar constant is defined as pointwise multiplication by this scalar, so for instance $2u[n - 3]$ has the value 0 for $n < 3$, and the value 2 everywhere else.

The response $y[n]$ of the system in Slide 10.13 when its input $x[n]$ is the unit sample signal $\delta[n]$ is referred to as the **unit sample response** (or sometimes the pulse response or impulse response, though the latter term is perhaps better used only in the CT case), Slide 10.18. We denote the output in this special case by $h[n]$. Similarly, the response to the unit step signal $u[n]$ is referred to as the **unit step response**, and denoted by $s[n]$.

A particularly valuable use of the unit step function, as we shall see later, is in representing a rectangular-wave signal as an alternating sum of delayed unit step functions. This is illustrated in Slide 10.19. We shall return to this decomposition.

■ 10.2.3 Time-Invariance

Consider a DT system such as that in Slide 10.13, with input signal $x[.]$ and output signal $y[.]$, so $x[n]$ and $y[n]$ are the values seen at the input and output at time n . The system is said to be *time-invariant*, see Slide 10.20, if shifting the input signal $x[.]$ in time by an arbitrary positive or negative integer D to get a new input signal² $x_D[n] = x[n - D]$ produces a corresponding output signal $y_D[n]$ that is just $y[n - D]$, i.e., is the result of simply applying the same shift D to the response $y[.]$ that was obtained with the original unshifted input signal.

The key to recognizing time-invariance in a given system description is to ask whether the rules or equations by which the input values $x[.]$ are combined, to create the output $y[n]$, involve knowing the value of n itself (or something equivalent to knowing n), or just time *differences* from the time n . If only the time differences from n are needed, the system is time-invariant. In this case, the same behavior occurs whether the system is run yesterday or today, in the following sense: if yesterday’s inputs are applied today instead, then the output today is what we would have obtained yesterday, just occurring a day later.

Another operational way to recognize time-invariance is to ask whether shifting the pair of signals $x[.]$ and $y[.]$ by the arbitrary but identical amount D results in new signals $x_D[.]$ and $y_D[.]$ that still satisfy the equations defining the system. More generally, a set of signals that jointly satisfies the equations defining a system, such as $x[.]$ and $y[.]$ in our input-output example, is referred to as a *behavior* of the system. And what time-invariance requires is that time-shifting any behavior of the system by an arbitrary amount D results in a set of signals that is still a behavior of the system.

Consider a few examples. A system defined by the relation

$$y[n] = 0.5y[n - 1] + 3x[n] + x[n - 1] \quad \text{for all } n \quad (10.2)$$

is time-invariant, because to construct $y[.]$ at any time instant n , we only need values of $y[.]$ and $x[.]$ at the same time step and one time step back, no matter what n is — so we don’t need to know n itself. To see this more concretely, note that the above relation holds for all n , so we can write

$$y[n - D] = 0.5y[(n - D) - 1] + 3x[n - D] + x[(n - D) - 1] \quad \text{for all } n$$

²Delayed by D from the original input, if $D > 0$; or advanced by $-D$ if $D < 0$.

or

$$y_D[n] = 0.5y_D[n-1] + 3x_D[n] + x_D[n-1] \quad \text{for all } n .$$

In other words, the time-shifted input and output signals, $x_D[.]$ and $y_D[.]$ respectively, also satisfy the equation that defines the system.

The system defined by

$$y[n] = n^3x[n] \quad \text{for all } n \tag{10.3}$$

is *not* time-invariant, because the value of n is crucial to defining the output at time n . A little more subtle is the system defined by

$$y[n] = x[0] + x[n] \quad \text{for all } n . \tag{10.4}$$

This again is *not* time-invariant, because the signal value at the absolute time 0 is needed, rather than a signal value that is offset from n by an amount that doesn't depend on n . We have $y_D[n] = x[0] + x_D[n]$ rather than what would be needed for time-invariance, namely $y_D[n] = x_D[0] + x_D[n]$.

■ 10.2.4 Linearity

Before defining the concept of linearity, it is useful to recall two operations on signals or time-functions that were defined in connection with Equation (10.1) above, namely (i) addition of signals, and (ii) scalar multiplication of a signal by a constant. These operations were defined as pointwise (i.e., occurring at each time-step), and were quite natural definitions. (They are completely analogous to vector addition and scalar multiplication of vectors, the only difference being that instead of the finite array of numbers that we think of for a vector, we have an infinite array, corresponding to a signal that can take values for all integer n .)

With these operations in hand, one can talk of *weighted linear combinations* of signals. Thus, if $x_1[.]$ and $x_2[.]$ are two possible input signals to a system such as that in Slide 10.13, for instance the signals associated with experiments numbered 1 and 2, then we can consider an experiment 3 in which the input $x_3[.]$ a weighted linear combination of the inputs in the other two experiments:

$$x_3[n] = a_1x_1[n] + a_2x_2[n] \quad \text{for all } n ,$$

where a_1 and a_2 are scalar constants.

The system is now termed *linear* if the response to this weighted linear combination of the two signals is the *same weighted combination of responses* to the two signals, for all possible choices of $x_1[.]$, $x_2[.]$ a_1 and a_2 , i.e., if

$$y_3[n] = a_1y_1[n] + a_2y_2[n] \quad \text{for all } n ,$$

where $y_i[.]$ denotes the response of the system to input $x_i[.]$ for $i = 1, 2, 3$.

This relationship is represented in Slide 10.21. If this property holds, we say that the results of any two experiments can be *superposed* to yield the results of other experiments; a linear system is said to have the **superposition property**. (In terms of the notion of behaviors defined earlier, what linearity requires is that weighted linear combinations, or

superpositions, of behaviors are again behaviors of the system.)

We can revisit the examples introduced in Equations (10.2), (10.3), (10.4) to apply this definition, and recognize that all three systems are linear. The following are examples of systems that are *not* linear:

$$\begin{aligned}y[n] &= x[n] + 3 ; \\y[n] &= x[n] + x^2[n - 1] ; \\y[n] &= \cos\left(\frac{x^2[n]}{x^2[n] + 1}\right) .\end{aligned}$$

All three examples are time-invariant.

■ 10.2.5 Linear, Time-Invariant (LTI) Models

Models that are both linear and time-invariant, or *LTI models*, are hugely important in engineering and other domains. We will mention some of the reasons in the next chapter. After developing insights into their behavior and tools for their analysis in the next few lectures, we will return to apply what we have learned, to better understand signal transmission on physical channels.

The class demo that you saw, Slide 10.22, involved generating tone bursts at the loudspeaker of a laptop computer, and detecting the response at a microphone. The input $x[n]$ in this case was a baseband signal of the form in Slide 10.9, but alternating regularly between high and low values. This was converted through a modulation process into the tone bursts that you heard. The signal received at the microphone was then demodulated to reconstruct an estimate $y[n]$ of the baseband input, which you saw displayed.

With the microphone in a fixed position, we saw responses that had some consistency from one transition to the next (between tone and no-tone), despite the presence of random fluctuations riding on top of things. The deterministic or repeatable part of the response $y[n]$ did indeed show distortion, i.e., deviation from $x[n]$, though more “real-world” than what is shown in the synthetic example on Slide 10.9. However, when the microphone was very close to the speaker, the distortion was minimal.

There were features of the system response in this class demo to suggest that it may not be unreasonable to model the baseband acoustic channel as LTI. Time-invariance (at least over the time-horizon of the demo!) was suggested by the repeatability of the transient responses to the various transitions. Linearity was suggested by the fact that downward transients looked like reflections of the upward transients, and also by the appropriate scaling of the response when the input was scaled.

■ Problems and Exercises

Please see

<http://mit.edu/6.02/www/currentsemester/handouts/tutprobs/lti.html>
for online problems based on this chapter.