Introduction to Signals and Systems, Part V: Lecture Summary

So far we have only looked at examples of *non-recursive* difference equations; that is, difference equations where the output is only dependent on time-delayed values of the input signals $x[n]$, $x[n-1]$,..., $x[n-M+1]$:

$$y[n] = \sum_{k=0}^{M} b_k x[n-k] \tag{1}$$

This class of LTI systems have the desirable property that as long as the input signal and filter coefficients are bounded,

$$|x[n]| < \infty,\ \forall n, \tag{2}$$

$$|b_k| < \infty,\ \forall k, \tag{3}$$

the output signal is bounded as well, such that,

$$|y[n]| < \infty,\ \forall n. \tag{4}$$

This property is known as *BIBO (bounded-input, bounded-output) stability. Recursive* difference equations,

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k] \tag{5}$$

do not necessarily have this property, as illustrated with the following simple difference equation,

$$y[n] = 2y[n-1] + x[n], \tag{6}$$

initial condition $y[-1] = 0$, and input signal,

$$x[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{7}$$

Figure 1 below plots the output $y[n]$ for $n \in \{0, 1, \ldots, 10\}$. Note that despite bounded filter coefficients and a single nonzero input at time index $n = 0$, the output $y[n]$ grows without bound.

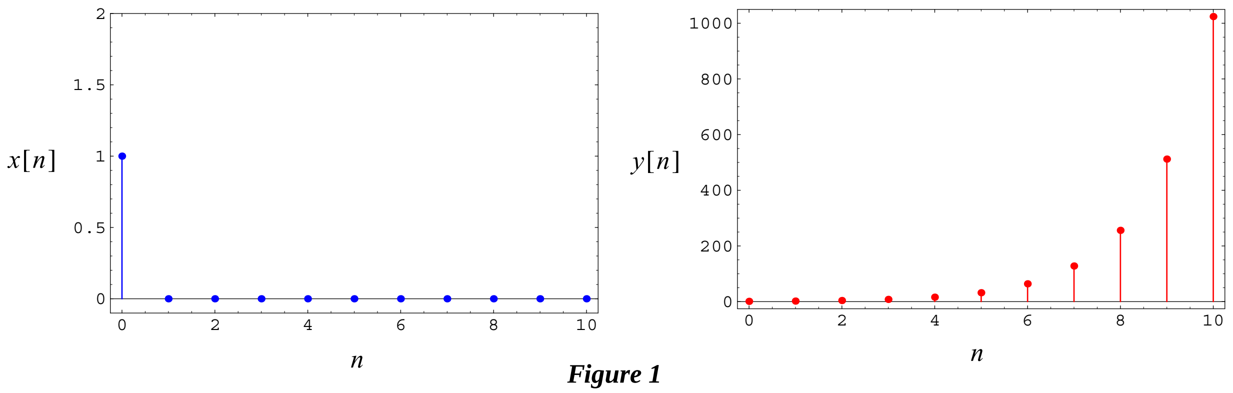

Figure 1

The output of difference equation (6) feeds back on itself to cause this unbounded result. While *feedback* can be a very desirable property of systems, especially in the field of control, the wrong kind of feedback can lead to unbounded or *unstable* system outputs. One example of "bad" feedback that we are all familiar with occurs when a microphone is placed to close to its own amplifier. Low-volume sounds near the microphone are boosted by the amplifier, whose output then feeds back into the microphone, which then gets further amplified until this vicious cycle results in the very loud familiar screeching sound that we have all heard before.

Because *some* recursive difference equations can result in undesirable system responses, does not mean that *all* recursive systems are undesirable. For example, just as we were able to derive non-recursive filters with desirable properties (such as low-pass filtering), we can do the same for recursive filters. Consider, for example, the simple recursive difference equation below:

$$y[n] = a_1 y[n-1] + b_0 x[n] \tag{8}$$

In equation (6), we have already seen how a bad choice of filter coefficients ($a_1 = 2$, $b_0 = 1$) can result in an unstable system response; suppose, however, that we choose the following filter coefficients instead:

$$a_1 = \alpha,\ b_0 = (1-\alpha),\ 0 < \alpha < 1 \tag{9}$$

such that equation (8) becomes:

$$y[n] = \alpha y[n-1] + (1-\alpha)x[n]. \tag{10}$$

For values of α close to 1, the system equation (10) becomes a simple recursive low-pass filter. In Figure 2, we plot the filtered output signal $y[n]$, the magnitude frequency spectrums $|X(f)|$ and $|Y(f)|$, and the magnitude frequency response $|H(f)|$ for a sample input signal $x[n]$ and $\alpha = 0.1$, $\alpha = 0.5$ and $\alpha = 0.9$. (See *Mathematica* notebook, sections "Linear, recursive difference equation example" for these examples.) As in previous examples, the input signal $x[n]$ is a discrete-time signal sampled at $f_s = 500$ Hz from the continuous-time signal,

$$x(t) = \sin(2\pi \cdot 4t) + noise \tag{11}$$

where, for each sample, the "noise" component of the signal consists of a uniformly distributed random number in the interval $[-1/2, 1/2]$. Note that we are able to achieve low-pass filtering with many fewer coefficients in the recursive case compared to the non-recursive case. In fact, it is true in general that for similar performance characteristics, recursive filters can be designed with fewer coefficients than non-recursive filters. Either way, filter design largely boils down to picking appropriate filter coefficients for both the recursive and non-recursive case.

As we have seen, however, an important additional consideration in designing recursive systems is that the output be BIBO-stable. Note that for a recursive LTI difference equatiaon,

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k] \tag{12}$$

BIBO (bounded-input, bounded-output) stability means that when the input signal and filter coefficients are bounded,

$$|x[n]| < \infty,\ \forall n, \tag{13}$$

$$|a_l| < \infty,\ \forall l, \tag{14}$$

$$|b_k| < \infty,\ \forall k, \tag{15}$$

the output signal is bounded as well, such that,

$$|y[n]| < \infty,\ \forall n. \tag{16}$$

In this class, we will develop an analysis tool, known as the *z-transform,* that will allow us to determine the stability of a recursive system without having to test the system response for every possible input signal. In fact, the z-transform will allow us to make predictions about the system response beyond stability; for example, we will be able to predict the exact time-domain system output for certain specific input signals.To illustrate some basic ideas, we now look at three different cases: (1) an *unstable* system, (2) a *marginally stable* system and (3) a *stable* system. Difference equations for these three systems are given in equations (17), (18) and (19), respectively:

$$y_1[n] = (-1/4)y_1[n-1] + (1/3)y_1[n-2] - (1/2)y_1[n-3] + x[n] \text{ (unstable)} \tag{17}$$

$$y_2[n] = (-1/2)y_2[n-1] + (1/2)y_2[n-2] + x[n] \text{ (marginally stable)} \tag{18}$$

$$y_3[n] = (-1/3)y_3[n-1] + (1/2)y_3[n-2] + x[n] \text{ (stable)} \tag{19}$$

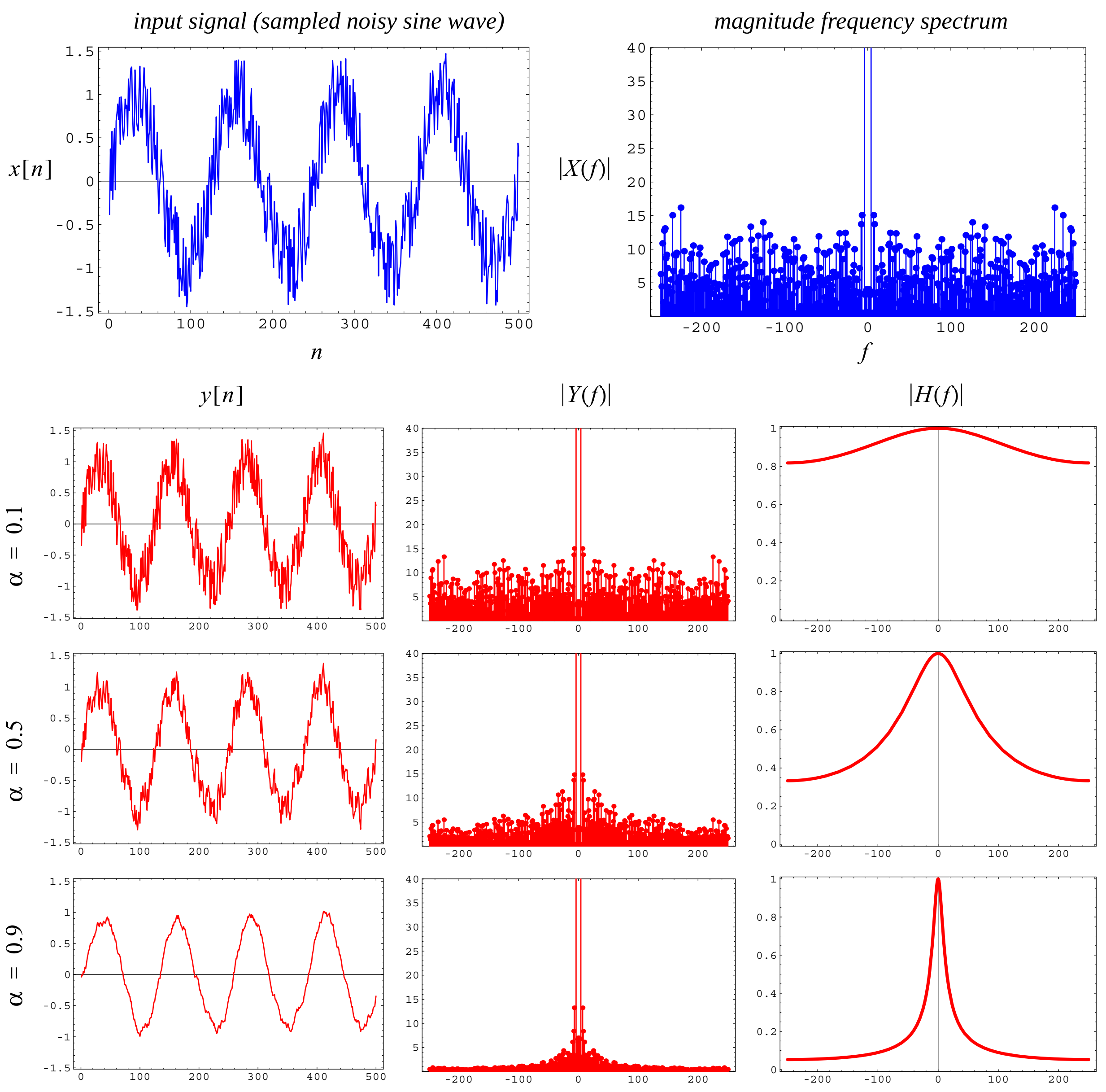

Figure 2: Recursive, low-pass filtering examples

For each of these systems, we let the input sequence be,

$$x[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \tag{20}$$

which is known as a *discrete-time unit impulse*, and assume that,

$$y_p[n] = 0\,,\ n < 0\,,\ p \in \{1, 2, 3\}\,. \tag{21}$$

Figure 3 illustrates the system output for input (20) for the three difference equations above; these system responses are known as the *impulse response* of the system (i.e., the response to an impulse input). Note that for the unstable system, the output grows without bound; for the marginally stable system, the output never goes back to zero; while for the stable system, the output decays to zero over time.

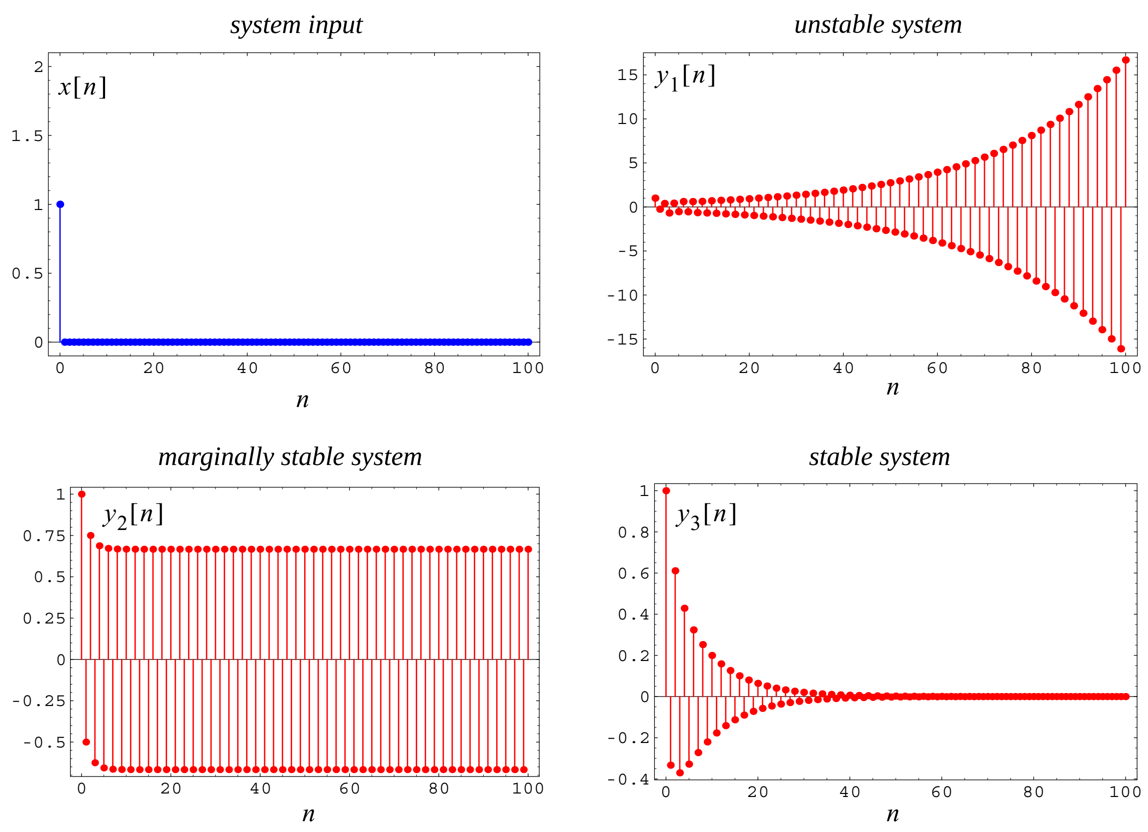

Figure 3

Now, just looking at the difference equations above, it is not obvious why the three systems should behave so radically different. However, if we transform these systems using the z-transform, to be studied later in this course, we will be able to make predictions about a system's stability without explicitly simulating the system response for a number of different inputs. For these three systems, the z-transforms are given by[1]:

$$H_1(z) = \frac{z^3}{z^3 + (1/4)z^2 - (1/3)z + (1/2)} \text{ (unstable)} \tag{22}$$

$$H_2(z) = \frac{z^2}{z^2 + (1/2)z - (1/2)} \text{ (marginally stable)} \tag{23}$$

$$H_3(z) = \frac{z^2}{z^2 + (1/3)z - (1/2)} \text{ (stable system)} \tag{24}$$

As it turns out, the roots of the denominators of the z-transforms above govern the stability of each system. For causal LTI systems[2], if the roots lie *inside* the unit circle of the complex plane (i.e. horizontal real axis, imaginary vertical axis), the system will be stable; if one or more roots lies *on* the unit circle, the system will be marginally stable; and, finally, if one or more roots lies *outside* the unit circle, the system will be unstable. In Figure 4, we plot the roots of the denominators in equations (22) through (24) in the complex plane. We can now readily observe the stability (or lack thereof) of each system. System #1 has one root outside the unit circle and is therefore unstable; system #2 has one root on the unit circle and is therefore marginally stable; and system #3 has all its roots inside the unit circle and

1. *The observant reader will notice a very close relationship between the coefficients of the difference equations and the coefficients of the numerator and denominator polynomials in the z-transforms.*
2. *"Causal" means that the output is dependent only on previous system outputs, and current and/or previous system inputs.*

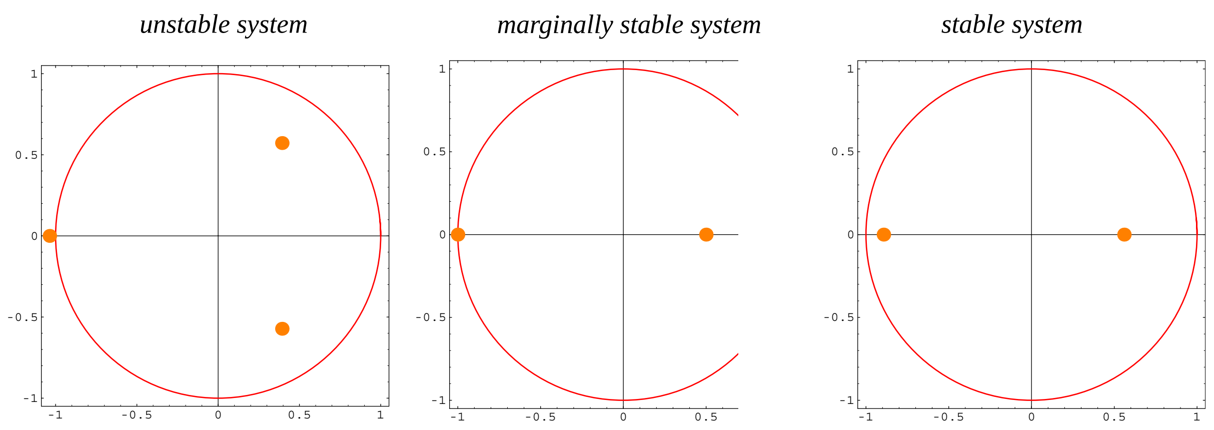

Figure 4: root locations of z-transform denominators

is therefore stable. As such, the z -transform is a very powerful analysis tool in our study of recursive difference equations.

In fact, the z -transform will allow us to find closed-form solutions for the impulse response of a system. For example, the system response of the marginally stable system is given by,

$$y[n] = \frac{1}{3}[2(-1)^n + 2^{-n}] \text{, } n \geq 0 \text{.} \tag{25}$$

We continued our lecture discussion by looking at an example of a simple *nonlinear* recursive difference equation. Nonlinear difference equations cannot be expressed in the form or equation (12) and can often behave in strange and complicated ways; as such, they are much more difficult to analyze than LTI difference equations. The example we considered was given by,

$$y[n] = Ry[n-1](1-y[n-1]) \text{, } y[-1] = 1/2 \text{, } R > 0 \text{.} \tag{26}$$

Figure 5 plots the output for different values of R; note how even this simple nonlinear system can produce remarkably complex output patterns, for example, when $R = 3.7$. (See *Mathematica* notebook, section “Nonlinear, recursive difference equation” for this example.) Nonlinear systems simply do not lend themselves to the same kind of (relatively) straightforward analysis as LTI systems.

This lecture concluded our qualitative overview of the material in this class. We ended the lecture with a slide presentation of where we will go from here (see accompanying lecture slides).

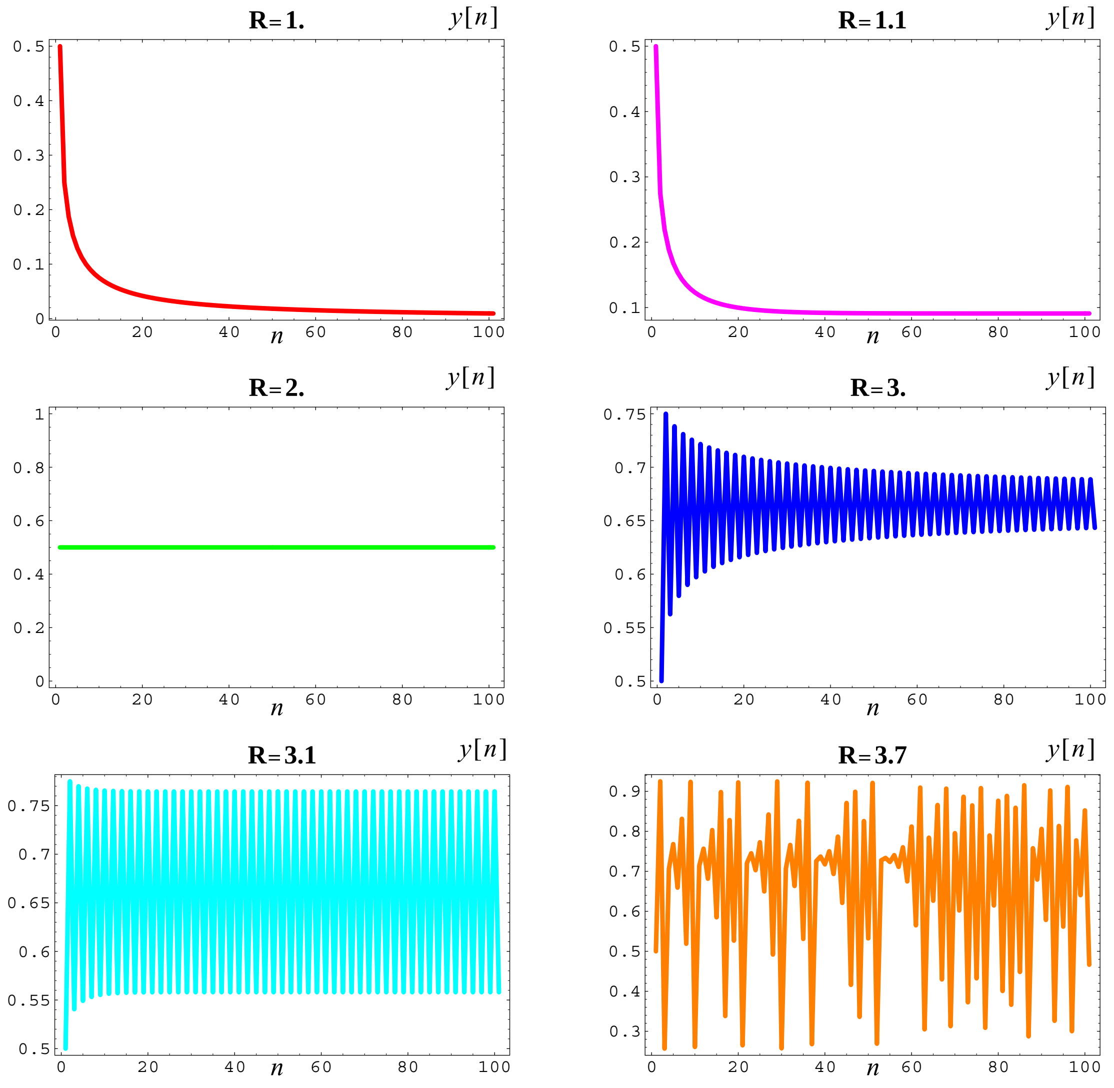

Figure 5: output for different values of R