Lecture 6

Stability

Introduction

Stability is an essential property of a practical digital system. An unstable system will (almost always) "blow up" and become useless (or worse). We experimented with this is Project 1 when we implemented an IIR notch filter. We saw that small changes in one parameter caused it change from a useful system to a noise source. Here we want to learn how we can ensure that any filters we design are stable.

BIBO stability

A useful way to define stability for a discrete system is to require that a *bounded input* always produces a *bounded output*. This is called *bounded-input, bounded-output* (*BIBO) stability*. Henceforth we will refer to this as simply "stability." It follows that a system is *unstable* if there exists a bounded input that produces an unbounded output, that is, the output "blows up" for some bounded input.

Formally, if a system is described by the transformation

$$y(n)=T\{x(n)\} \tag{1}$$

then it is BIBO stable if and only if for any bounded input

$$|x(n)|\le M_x<\infty \tag{2}$$

the output is bounded

$$|y(n)|<M_y<\infty \tag{3}$$

where M_x, M_y are finite positive numbers.

LTI system stability

Any LTI system can be described by a convolution

$$y(n)=\sum_{k=-\infty}^{\infty} h(k)x(n-k) \tag{4}$$

If $|x(n)|\le M_x<\infty$ then

$$|y(n)|\le \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)|\le M_x \sum_{k=-\infty}^{\infty} |h(k)|$$

and $|y(n)|<M_y<\infty$ if

$$\sum_{k=-\infty}^{\infty} |h(k)|<\infty \tag{5}$$

This is a *sufficient* condition for stability. We also want to show that it is a *necessary* condition. Suppose (5) is not true, so

$$\sum_{k=-\infty}^{\infty} |h(k)| = \infty \tag{6}$$

The signal defined as

$$x(-k) = \begin{cases} 1 \ , \ h(k) > 0 \\ -1 \ , \ h(k) < 0 \\ 0 \ , \ h(k) = 0 \end{cases}$$

is bounded, $|x(n)| \le 1$, and

$$y(0) = \sum_{k=-\infty}^{\infty} h(k)\, x(-k) = \sum_{k=-\infty}^{\infty} |h(k)| = \infty \tag{7}$$

The output is not bounded and the system is unstable. So, we see that (5) is also a necessary condition for stability. Therefore, a linear time-invariant system is stable (in the BIBO sense) if and only if its impulse response is *absolutely summable* (condition (5)).

FIR filter stability

An FIR filter has the form

$$y(n) = \sum_{k=0}^{M} b_k\, x(n-k) \tag{8}$$

with transfer function

$$H(z) = b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M} \tag{9}$$

The impulse response is

$$h(n) = b_n \tag{10}$$

and condition (5) is

$$\sum_{k=0}^{M} |b_k| < \infty \tag{11}$$

FIR filters are *manifestly stable*, except for the obvious, and uninteresting case where one or more of the coefficients is infinite. Stability is not an issue for FIR filters.

IIR filter stability

An IIR filter has the form

$$y(n) = \sum_{k=0}^{M} b_k\, x(n-k) - \sum_{k=1}^{N} a_k\, y(n-k) \tag{12}$$

with transfer function

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_N z^{-N}} \tag{13}$$

It’s easy to find an unstable IIR filter. For example

$$y(n)=x(n)+2y(n-1) \tag{14}$$

The bounded input $x(n)=\delta(n)$ produces the unbounded output $y(n)=2^n u_s(n)$. In contrast with FIR filters, for IIR filters stability *is* an important issue to consider.

Exercise 1: Show that system (14) with input $x(n)=\delta(n)$ produces output $y(n)=2^n u_s(n)$ (assume $y(n)=0$ for all $n<0$).

If a causal system's transfer function (13) has only simple poles, p_i, then we know that the impulse response has the form

$$h(n)=\sum_i A_i p_i^n u_s(n) \tag{15}$$

If all poles satisfy $|p_i|<1$ then $h(n)$ will exponentially decay as $n\to\infty$ and (5) will be satisfied. If any pole has $|p_i|\geq 1$ then (5) will not be satisfied, and the system will be unstable.

For multiple poles the response will have terms of the form $n\,p_i^n$, $n^2\,p_i^n$ and so on. It is still true that if $|p_i|<1$ these term will decay sufficient fast as $n\to\infty$ so that (5) will be satisfied. Therefore:

A causal system with transfer function $H(z)$ is BIBO stable if and only if all poles of $H(z)$ fall inside the unit circle $|z|=1$, that is, $|p_i|<1$ for all poles.

IIR notch filter

In Project 1 we observed that the notch filter

$$H(e^{j\omega})=\frac{1-2\cos(\omega_b)z^{-1}+z^{-2}}{1-2r\cos(\omega_b)z^{-1}+r^2z^{-2}} \tag{16}$$

is unstable for $r\geq 1$. Let's examine this in light of our BIBO stability condition. Multiplying numerator and denominator by z^2, the poles of H are the solutions of the quadratic equation

$$z^2-2r\cos(\omega_b)z+r^2=0 \tag{17}$$

These solutions are

$$p=r\cos(\omega_b)\pm\sqrt{r^2\cos^2(\omega_b)-r^2}=r\left[\cos(\omega_b)\pm j\sqrt{1-\cos^2(\omega_b)}\right] \tag{18}$$

Since $|p|=r$ the stability condition reduces to $r<1$.

Can "problem" poles be canceled?

Suppose a system with transfer function $H_1(z)$ is unstable due to a simple pole p_1 with $|p_1|\geq 1$. Let a second system have transfer function $H_2(z)$ with a zero at $z=p_1$, that is, $H_2(p_1)=0$. Suppose we cascade these systems (Fig. 1) to obtain a transfer function $H(z)=H_1(z)H_2(z)$. In principle the zero of $H_2(z)$ will cancel the problem pole of $H_1(z)$ and the cascaded system transfer function $H(z)$ will be stable. This is true in theory, but problematic in practice.

First of all, even if the cancellation is ideal, the output of the first system, call it $v(t)$, will still "blow up" only to be reduced to a finite level by the second system. No real system, analog or

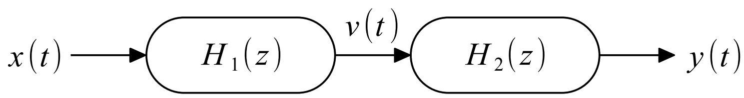

Fig. 1: Cascading two filters in an attempt to cancel a problematic pole in one of them.

digital, can properly represent an arbitrarily large signal. Therefore, the cascaded system will ultimately fail.

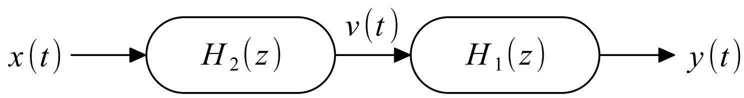

Fig. 2: Cascade with order of filters reversed.

Reversing the order of the filters would, in principle, solve this problem. Ideally, the components in $x(t)$ that could excite the problem pole of $H_1(z)$ will be zeroed out by $H_2(Z)$ and not be present in $v(t)$. To actually achieve this in practice requires *perfect* cancellation. However, our filters are implemented with finite precision, and perfect cancellation is generally not possible.

The following Scilab code applies input $x(n)=\delta(n)$ to a filter with transfer function

$$1-(1.1-10^{-9})z^{-1}$$

to produce output $v(n)$, and then applies that as input to a filter with transfer function

$$\frac{1}{1-1.1z^{-1}}$$

```
Ns = 220;
x = zeros(1,Ns);
x(1) = 1; //x is the delta function
b = [1,-1.1+1e-9];
a = [1];
v = filter(b,a,x);
b = [1];
a = [1,-1.1];
y = filter(b,a,v);
plot(y,'r.');
```

The total system transfer function is

$$H(z)=\frac{1-(1.1-10^{-9})z^{-1}}{1-1.1z^{-1}}$$

which is *almost* 1 for any value of *z*, but not exactly. The impulse response is shown in Fig. 3. Ideally this would be $y(n)=\delta(n)$ but we see that eventually $y(n)$ starts to blow up exponentially.

This same effect can arise due to round-off error when filter coefficients are represented with finite precision in a DSP processor.

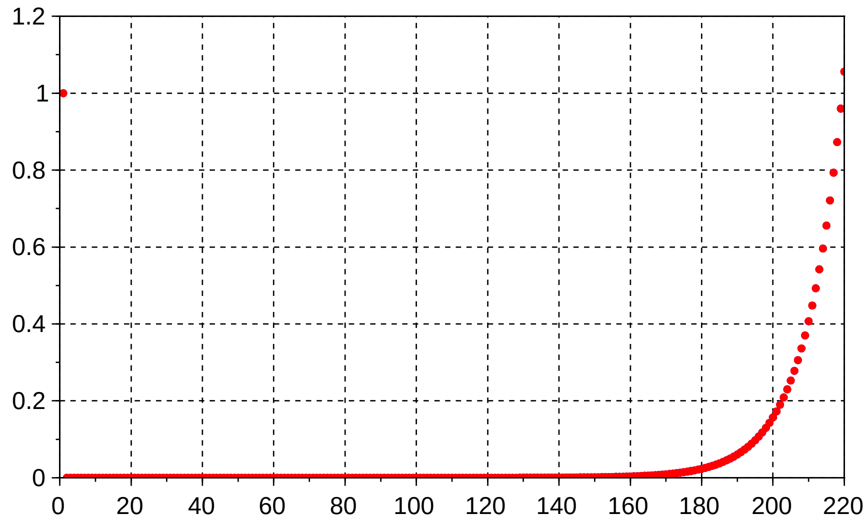

Fig. 3: Impulse response of system described by above Scilab code.