Lec 3 F01: 8/24/01, Fri.

Last time, we had a grand overview of the course, and started looking at discrete-time signals.

1. DT Signals.

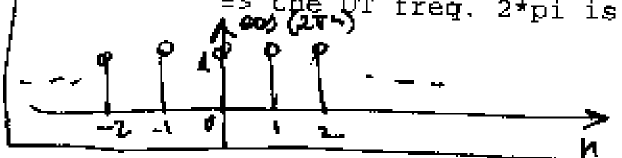
We introduced some notation and definitions, and saw that ~~it~~ was important to regard a signal as a rule, or an algorithm, for producing one number, given another number. We also looked at various representations of signals. One important point of view was to consider an N-point signal as a point, or a vector, in an N-dimensional vector space. Then we looked at some properties of signals, and some special signals, and finished by talking about some properties specific to DT sinusoids.

1.1.6. PECULIAR PROPERTIES OF DT SINUSOIDS

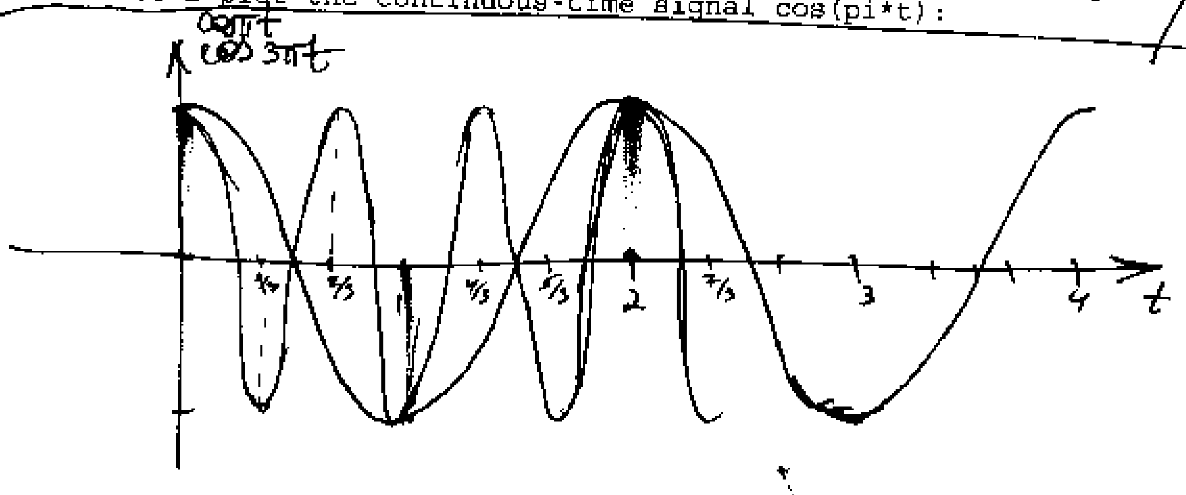
(a) the highest frequency is pi;
(b) adding 2*pi to the frequency does not change the signal;
(c) DT sinusoids are not necessarily periodic!

We looked at cos(2*pi*n), to find out whether its frequency was larger than pi. Since n is integer,

cos(2*pi*n) = 1 = cos(0*pi*n)
=> the DT freq. 2*pi is the same as DT frequency 0.



Let's do another example very carefully, to see what's happening here. Let's plot the continuous-time signal cos(pi*t):



An important thing to notice is that these signals are the same at integer points. So, if we sample either of these signals at integer points, we'll get the same signal:

cos(pi*0) = cos(3*pi*0)
cos(pi*1) = cos(3*pi*1)
cos(pi*2) = cos(3*pi*2) => cos(pi*n) = cos(3*pi*n)
.
.

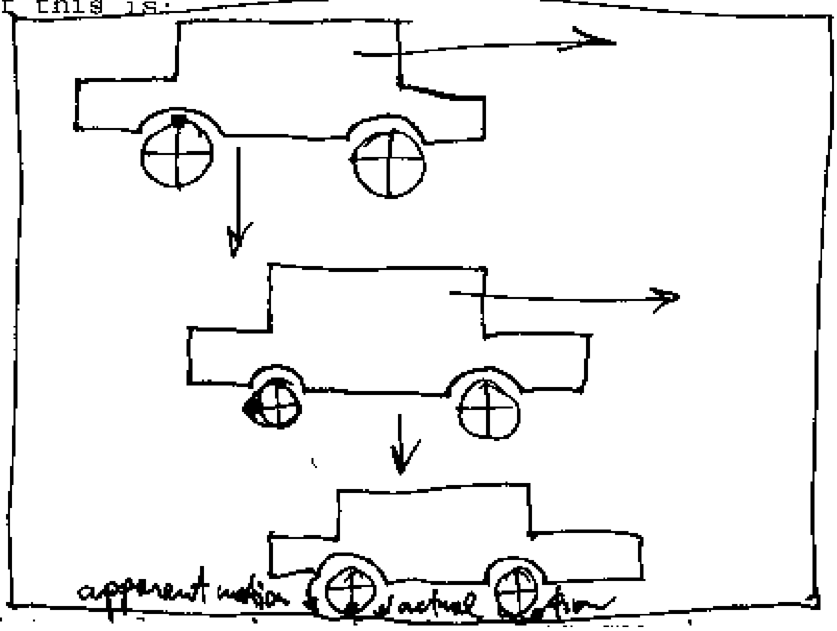
In Homework problem 2, you've encountered a similar situation, when discrete samples obtained from one continuous-time cosine could give rise to another continuous-time cosine.

More generally,

$$\boxed{\begin{aligned} &\cos(\omega n+\varphi) = \cos((\omega+2\pi)n+\varphi)\ ! \\ &\cos((\omega+2\pi)n+\varphi) = \cos[(\omega n+\varphi)+2\pi n] = \\ &= \cos(\omega n+\varphi)\underbrace{\cos(2\pi n)}_{1} - \sin(\omega n+\varphi)\underbrace{\sin(2\pi n)}_{0} = \cos(\omega n+\varphi) \end{aligned}}$$

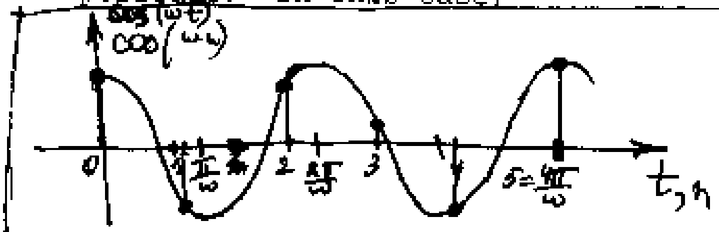
So, let me repeat: even though these two continuous-time signals are different, their sampling at integer points is the same. The red continuous-time signal oscillates faster, but it all happens in between sampling instants. The sampling points do not see this activity. This is why I said last time that two different continuous-time frequencies can appear to be the same discrete-time frequency. This phenomenon is called ALIASING.

You have all encountered aliasing when watching a movie. You must have noticed that sometimes a car moves in one direction, but its wheels seem to be rotating in the opposite direction. A simplistic picture of this is:



[SPEECH ALIASING DEMO]

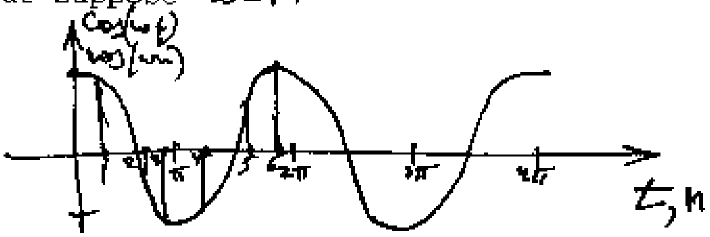
Let's now look at the last property--namely, that the DT sinusoids are not necessarily periodic.
Suppose you are sampling a CT sinusoid. If your first sample is at ~~zero~~ one, then, in order for your DT sinusoid to be periodic, it has to have a value of ~~zero~~ one again some time in the future. So, you're sampling, sampling, sampling, and at some point you hit ~~zero~~ one again. From then on, you'll start repeating. So, then your DT sinusoid is periodic. In this case,



$5 = \frac{4\pi}{\omega} \Rightarrow \omega = \frac{4\pi}{5}$

(HW2 prob 1)

But suppose $\omega = 1$:



Then there is no way a sampling point can be at another zero-crossing of this signal: after the first one, there will never be another zero crossing at an integer. So, this zero will never repeat.
What should happen in order for these two to catch up with each other?

You may remember that the next item on our agenda was to consider systems. The concept of a system is very similar to that of a signal--that is, IF you interpret a signal as a rule for transforming a number into another number.

1.2. Systems.

1.2.1. WHAT IS A SYSTEM?

Let me re-draw some of the pictures I drew last time, when we talked about signals.

We said that a signal is a RULE for producing a number in its range, given a number from its domain. So, each point in the domain is a single number.

```
    ________                     ________
   /        \                   /        \
  / Domain   \        f        / Range    \
 |   of s,    |    ------->   |   of f,    |
  \   D      /                 \   R      /
   \________/                   \________/
```

A system takes a signal as an input, and produces another signal as an output.

```
      /‾‾‾‾‾‾‾‾‾\                        /‾‾‾‾‾‾‾‾‾\
     /  Domain    \                     /  Range     \
    /  of S, D,    \        S          /  of S, R,    \
   |   a set of     |   -------->     |   a set of     |
    \  signals     /                   \  signals     /
     \            /                     \            /
      \__________/                       \__________/

DT system: DT signal -> DT signal;
CT system: CT signal -> CT signal.
```

So, while a signal assigns to every number in its domain a number in its range, a system assigns to every signal in its domain a signal in its range.

```
input signal x                                 output (or response) signal y=S[x]
         ------->| system S |  ------->
```

Remember, last time we had an example of a function which was the "division by three".

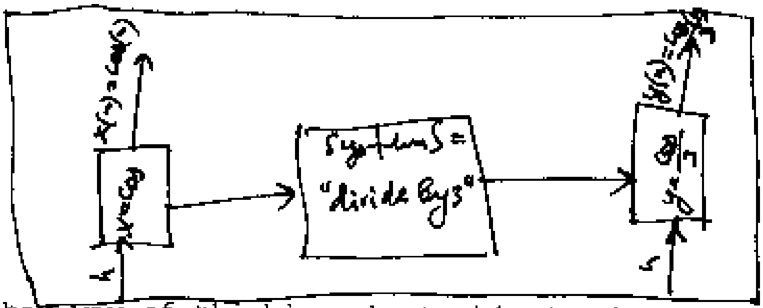
```
Example.

  argument n, an integer   | function           |  value f(n)=n/3, a real number
               ------->    | f = "divide by 3"  |  ------->
```

Well, we can also have a system which divides by three:

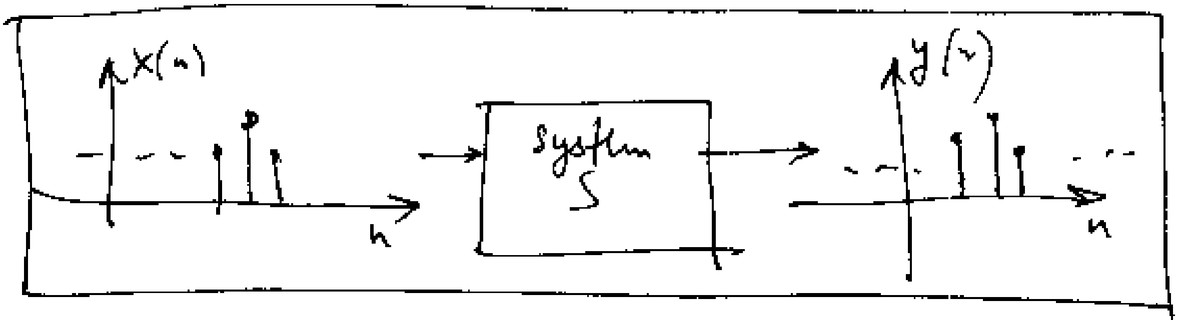
```
  input x, a DT signal   | system             |  output y = S[x], a DT signal
             ------->    | S = "divide by 3"  |  ------->

             The relationship between x and y:
             y(n) = x(n)/3 for all n between -infinity and infinity.
```

So this "for all n" is important. The crucial and most difficult thing here is to realize that this whole signal is fed into this box, not just one number, and a whole signal comes out.

For example, let's suppose that the input signal here x(n) = cos(n). Then the output is a different signal, y(n) = cos(n)/3. In other words, x is a rule for transforming a single number into another number; this system says I'm changing this rule into y:



Another way of thinking about this is that the whole graph of x is fed into S, and it produces the whole graph of y:



To emphasize that the input of S is the whole signal x, we will be using notation S[x], rather than S[x(n)].

Notation:

S[x] is preferable to S[x(n)]

S[x(n), $-\infty<n<\infty$]

This notation is also acceptable, provided that you keep in mind that this is not just x(n) for a particular number n, but rather the whole function x(n), for all values of n. Then, once the system's response is known, it can be evaluated at a particular n:

S[x](n) is synonymous with y(n), where y is the response of system S to input x.

Very often, systems are specified by input-output relationships. As we saw above,

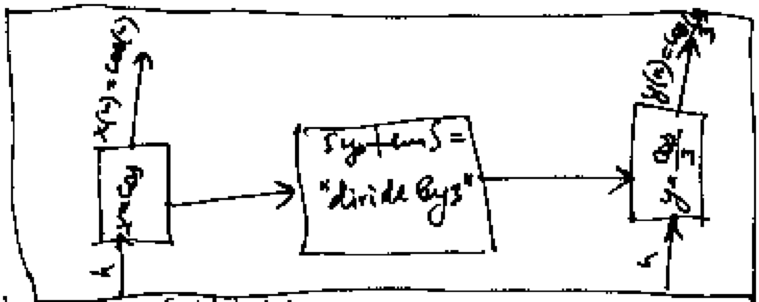
y(n) = x(n)/3, $-\infty<n<\infty$ specifies a system.

****

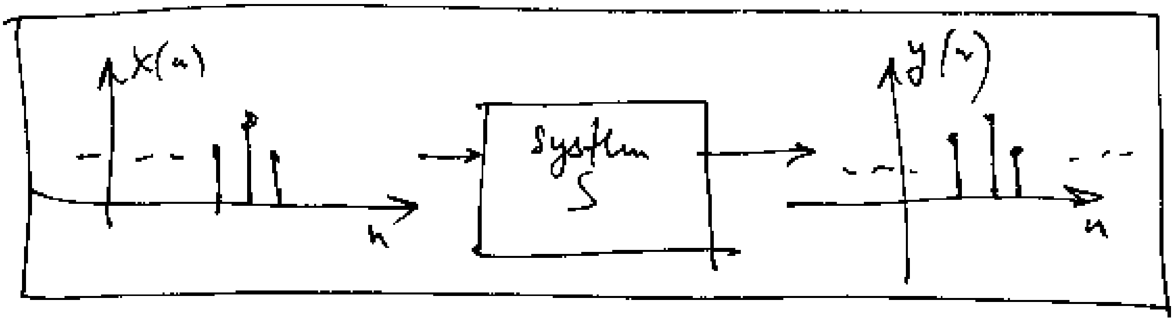
A generalization of this mapping idea is the so-called behavioral view of what a system is and does:

Behavioral view: a system is anything that IMPOSES CONSTRAINTS on a designated set of signals, where the signals are not necessarily labeled a inputs or outputs.

Any combintaion of signals that satisfies the constraints is termed a BEHAVIOR of the system.



Another way of thinking about this is that the whole graph of x is fed into S, and it produces the whole graph of y:



To emphasize that the input of S is the whole signal x, we will be using notation S[x], rather than S[x(n)].

Notation:
S[x] is preferable to S[x(n)]

S{x(n), $-\infty<n<\infty$}

This notation is also acceptable, provided that you keep in mind that this is not just x(n) for a particular number n, but rather the whole function x(n), for all values of n. Then, once the system's response is known, it can be evaluated at a particular *n*:

S[x](n) is synonymous with y(n), where y is the response of system S to input x.

Very often, systems are specified by input-output relationships. As we saw above,

y(n) = x(n)/3, $-\infty<n<\infty$ specifies a system.

****

A generalization of this mapping idea is the so-called behavioral view of what a system is and does:

Behavioral view: a system is anything that IMPOSES CONSTRAINTS on a designated set of signals, where the signals are not necessarily labeled a inputs or outputs.

Any combintaion of signals that satisfies the constraints is termed a BEHAVIOR of the system.

Ex.

Hw 2, Prob 4

Constraints: KCL, KVL, and the component equations.
HW 2, Prob 4: population dynamics. (No input, dynamical system which describes evolution of a population.)

****

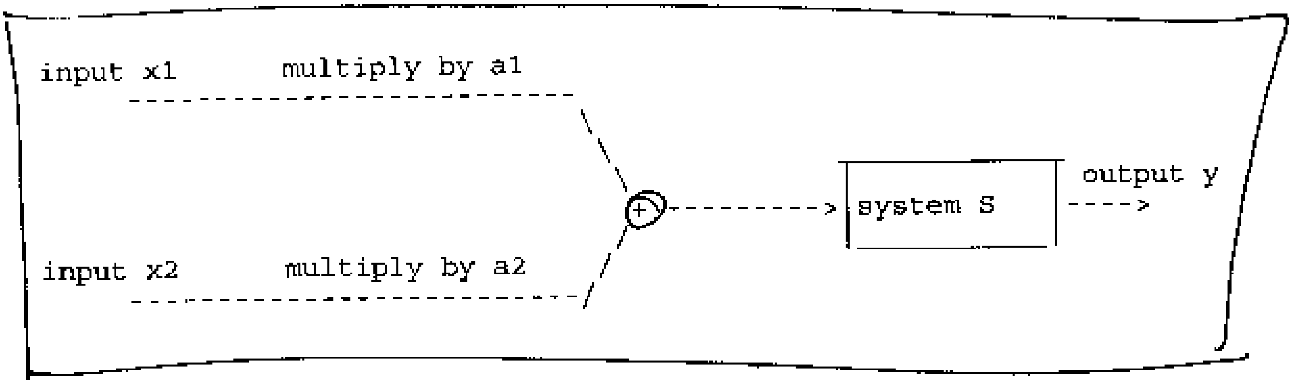
1.2 PROPERTIES OF SYSTEMS.

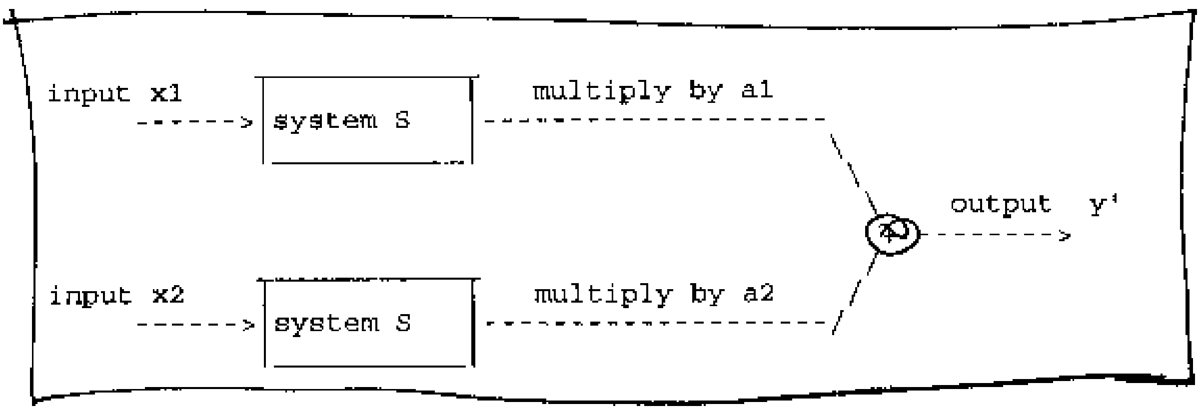
(a) Linearity. A system S is linear if, for any two input signals x1 and x2 and for any two numbers a1 and a2, it satisfies:

S[a1x1+a2x2] = a1S[x1] + a2S[x2]

That is, if the response to any linear combination of any two inputs is the same linear combination of the responses to these two inputs, then the system is linear.

Is there a neat picture with explains this horrendous formula? Of course there is.





This property is very important, because, if a system is linear, it says that we can compute the response to a complicated signal as the sum of responses to simpler signals.

****

Behavioral view: arbitrary linear combinations of behaviors are always behaviors

****

Examples.

1) y(n) = $\sum_{k=0}^{n} x(k)$

Input $x_1 \Rightarrow$ output $y_1(n) = \sum_{k=0}^{n} x_1(k)$

" $x_2 \Rightarrow$ " $y_2(n) = \sum_{k=0}^{n} x_2(k)$

Input $a_1x_1 + a_2x_2 \Rightarrow$ output $y_3 = \sum_{k=0}^{n} (a_1x_1(k) + a_2x_2(k))$

$= a_1 \underbrace{\sum_{k=0}^{n} x_1(k)}_{y_1(n)} + a_2 \underbrace{\sum_{k=0}^{n} x_2(k)}_{y_2(n)} = a_1y_1(n) + a_2y_2(n)$

2) y(n) = 2x(n)+3

input $x_1 = 1 \Rightarrow$ output $y_1(n) = 2 \cdot 1 + 3 = 5$

" $x_1 = 1 \Rightarrow$ " $y_2(n) = 5$

input $x_1 + x_2 = 2 \Rightarrow$ output $y_3(n) = 2 \cdot 2 + 3 = 7$

$\Rightarrow$ nonlinear! $\neq y_1(n) + y_2(n)$

3) q(n) = 3q(n-1)

If q_1, q_2 are behaviors, then

clearly $a_1q_1(n) + a_2q_2(n) = 3a_1q_1(n-1) + 3a_2q_2(n-1) \Rightarrow a_1q_1 + a_2q_2$ is also a behavior

Note that, in order to prove that a system is linear, we need to prove this statement for every possible pair of inputs, or for every possible pair of behaviors. However, in order to show that a system is nonlinear, one counterexample is enough.