ENGR 105: Feedback Control Design
Winter 2013

Lecture 6 - Block Diagrams

Friday, January 18, 2013

Today's Objectives

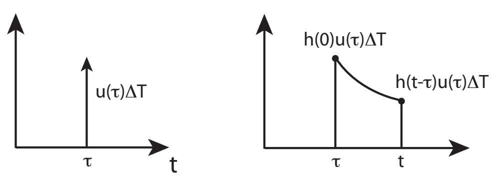
1. revisit the convolution from Lecture 4
2. learn to manipulate block diagrams

Reading: FPE Sections 3.1.1, 3.2

1 Revisiting convolution

In Lecture 4, we developed the convolution of the functions $u(t)$ and $h(t)$. Let's revisit:

Think of $u(t)$ as a train of impulses. What is the response to these impulses?



From the *principle of superposition*, we get the following: If a system has an input that can expressed as a sum of signals (e.g., as a sum of $u(t)$ values), then the response of the system can be expressed as the sum of the individual responses to the respective signals. But we have to look at the response at a later time (time $t - \tau$) in order to add up the effects of all the previous inputs (impulses) at that time t. We do this for all impulses which are discretely separated by a length of time ΔT. Note that this only works for *linear, time-invariant (LTI)* systems.

At any time t, the output y us a result of all past impulses, so:

$$y(t) = [h(t)u(0) + h(t - \Delta T)u(\Delta T) + h(t - 2\Delta T)u(2\Delta T) + \dots]\Delta T$$

$$y(t) = \sum_{k=0}^{\infty} h(t - k\Delta T)u(k\Delta T)\Delta T$$

As $\Delta T \to 0$: $y(t) = \int_0^\infty h(t - \tau)u(\tau)d\tau = u(t) * h(t)$, where $*$ is the symbol for convolution.

Question: Is the convolution commutative? Yes, $a * b = b * a$
It is also associative: $a * (b * c) = (a * b) * c$
And distributive: $a * (b + c) = (a * b) + (a * c)$

---

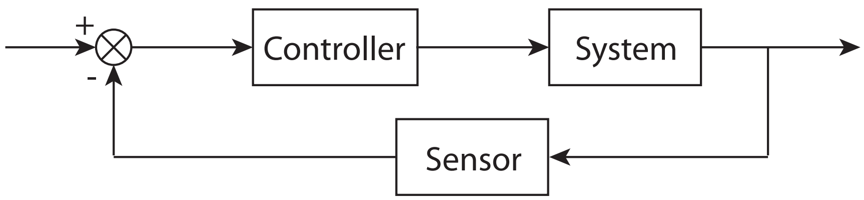
2 Block diagrams

The transfer function can be used to tell us a lot about the system, including:

- The impulse response
- Stability (through pole locations)
- The response to general inputs
- Steady-state value (if poles are stable)
- The frequency response (next lecture we will look at sinusoidal inputs)

The goal of feedback control is to shape the response of the system by closing a feedback loop, which changes the open-loop transfer function into the closed-loop transfer function.

To do this, we need to be comfortable working with systems in a form that looks like:

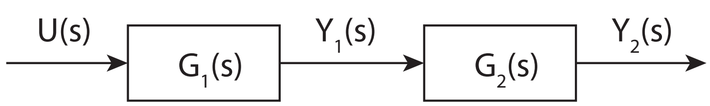


This is a block diagram

There are some simple rules of block diagram algebra that make it easy to manipulate systems into a form we want.

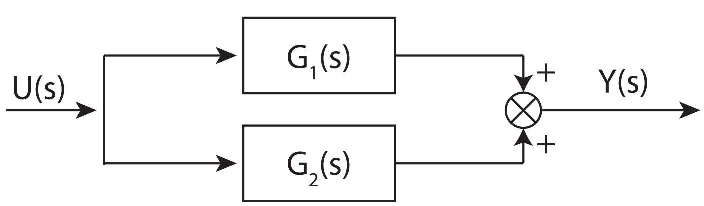
Basic Connections

Cascade

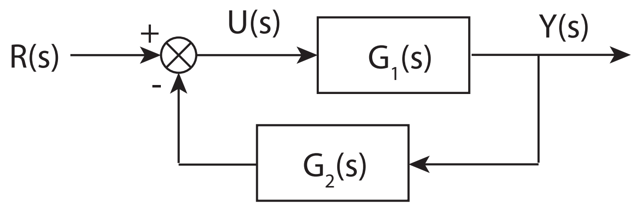


$$Y_2(s) = G_2(s)Y_1(s) = G_1(s)G_2(s)U(s)$$

Sum



$$Y(s) = [G_1(s) + G_2(s)]\,U(s)$$



Feedback

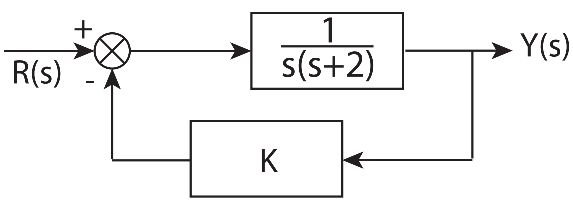
$$Y(s) = G_1(s)U(s)$$

$$= G_1(s)\left[R(s) - G_2(s)Y(s)\right]$$

$$\Rightarrow Y(s)\left[1 + G_1(s)G_2(s)\right] = G_1(s)R(s)$$

$$\frac{Y(s)}{R(s)} = \frac{G_1(s)}{1+G_1(s)G_2(s)}$$

Example



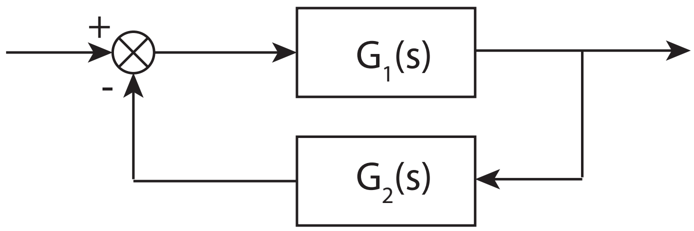
This could be a DC motor with position feedback.

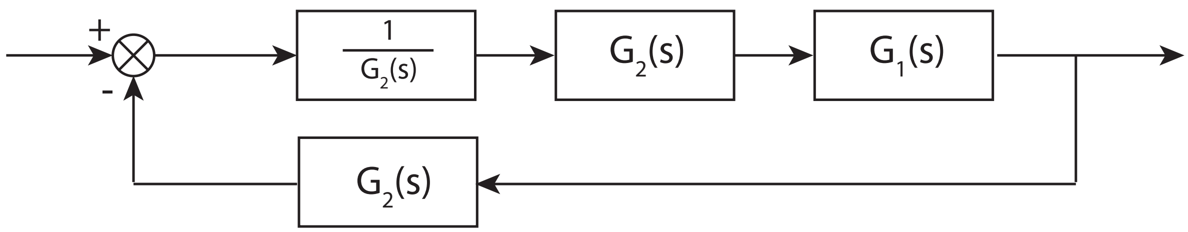
$$\frac{Y(s)}{R(s)} = \frac{\frac{1}{s(s+2)}}{1 + \frac{K}{s(s+2)}} = \frac{\frac{1}{s(s+2)}}{\frac{s(s+2)}{s(s+2)} + \frac{K}{s(s+2)}} = \frac{1}{s^2 + 2s + K}$$

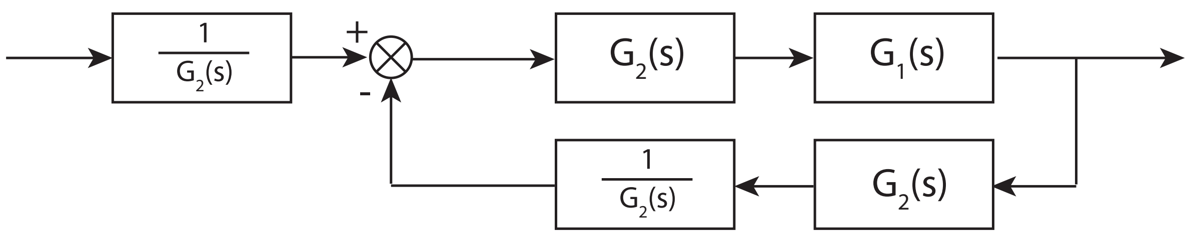
The response to a change in reference input is stable instead of continuing to integrate as in open loop. (Compare to example in Lecture 5.)

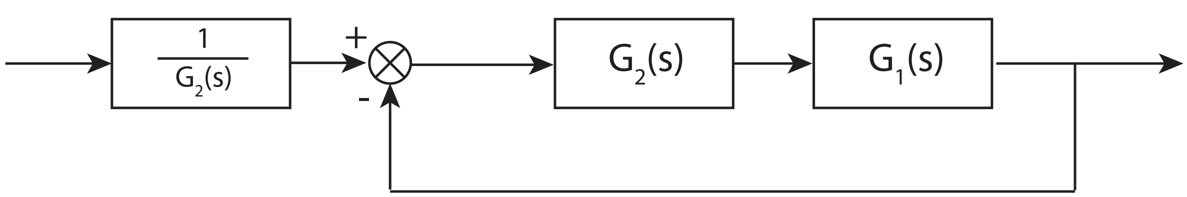
Unity feedback

It is often helpful to rearrange the system to have a unity feedback loop (no blocks in the feedback path). This can be accomplished with some manipulation of the diagram.









We can move the blocks into the feed forward or feedback paths to simplify analysis.