Imperial College London

Lecture 16

Introduction to Feedback Control – part 1

Prof Peter YK Cheung

Dyson School of Design Engineering

URL: www.ee.ic.ac.uk/pcheung/teaching/DE2_EE/
E-mail: p.cheung@imperial.ac.uk

In this lecture, we will start the final topic for this module on feedback control. This is required in order for you to design a system that behaves in the way you want without being absolutely accurate in knowing the system characteristics.

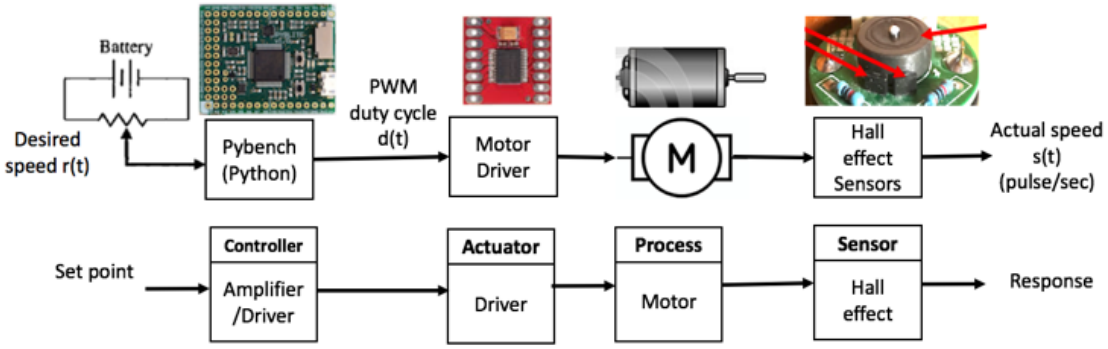
The concept of feedback is general. It is useful not only in control, but in many other situations and environments. For example, you remember last year's Electronics 1 module, where you used an op amp to amplify the ultrasonic signal. We used feedback to fix the gain of the amplifier by choosing the resistor values. The fundamental principle used there is **feedback** – it is applied in the op amp circuit to fix the gain of the amplifier, no matter what the op amp characteristic is.

What is control engineering? (a video)

Before I start this lecture, let us watch a short video explaining what is control engineering:

You can find this video on the link:
https://www.youtube.com/watch?v=Im88eVfkeBo&t=44s

Driving the DC motors – Open-loop control



- Driving the DC motors using Pybench in Lab 4 is known as "**open-loop control**"
- Potentiometer set the required speed (as voltage value)
- The Pybench board running Python produces control signals including direction (A1, A2) and PWM duty cycle. It acts as the **controller**
- The TB6612 H-bridge chip drives the motors – it is the **actuator**
- The motor is the thing being controlled – we call this "the **process**" or "the **plant**"
- The Hall effect **sensors** detect the speed and direction of the motor
- Problem: error in the desired speed setting vs the actual speed you get

PYKC 27 Feb 2020 DE2.3 – Electronics 2 Lecture 16 Slide 3

You have in the past been applying control to the DC motors, but **without feedback**. The potentiometer provide a desired motor speed. This is also known as the "**set point**". Here are what we did in Lab 4:

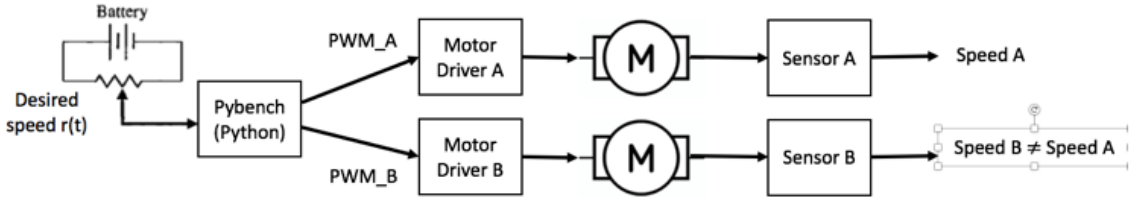
1. The Pybench board runs a program in Python in order to drive the motor through a PWM signal. The microcontroller running the Python program is the **controller**.
2. Then we use the TB6612 H-bridge chip to drive the motor. This is the **actuator**.
3. The output of the driver chip drives the motor directly.
4. Finally, we used the Hall effect sensors to measure the speed of the motor to provide an actual speed. This is the **sensor**.

The motor is what we want to control. In control terminology, the motor is known as the "**system**", the "**process**" or the "**plant**".

This system is subjected to "**open-loop**" control because the drive signal is independent of the output – there is no looping back of information to the drive input.

Finally, in our system, we are trying to control the speed of the motor. Therefore motor speed is the **control variable**.

Problem 1: Uncertainty in system characteristic



- There are many problems with open-loop control.
- First, the two motor may not respond in the same way to the drive input signal PWM_A and PWM_B. (For example, the two gear boxes may present different resistance to the motor, and the magnet inside the motors may have different strength.)
- The consequence is that the two motors are not balanced and the Segway will not go in a straight line.
- This is an example of the variation and uncertainty in the system characteristic. In this case, the steady-state behaviour of each motor may be different. It results in the actual speed of the two motors being different.
- One could use different gains to drive PWM_A and PWM_B to compensate for the difference in system characteristic. But this does not solve all the problems.

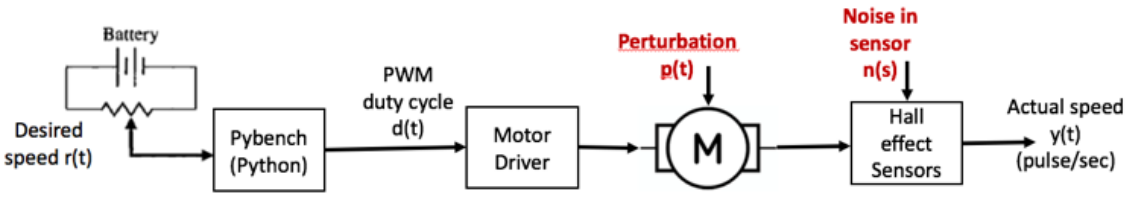
PYKC 27 Feb 2020 DE2.3 – Electronics 2 Lecture 16 Slide 4

Such a system has problems. As you have found out during the lab sessions, the two motors may be driven by the same PWM values, but the speed of the motors may be very different.

Open-loop control relies on **known system behaviour**. Any change in system behaviour (i.e. the process) will result in error in the outputs of the control variables.

However, open-loop control is not always bad. If the system characteristics is well defined and is not changing over time or under different operating conditions, open-loop control is easy to implement and will not subject to a major problem inherent in **feedback** or **closed-loop control**, which is the possibility of **instability**. Open-loop control will not cause instability, while feedback control could.

Problem 2: Disturbance and Noise



- Two other major problems exist:
  1. **Perturbation** – the motor may go on uneven surface or there may be some obstacles in the way;
  2. **Sensor noise** - The Hall effect sensors may not produce perfectly even pulses, the magnetic poles in the cylindrical magnet may not be evenly spaced.
- These two other factors will DIRECTLY affect the response of the system (i.e. the speed of the motor).
- Open-loop control cannot mitigate against these problems in any control systems.
- We need to use **feedback**, or **closed-loop control** in order mitigate these problems.

PYKC 27 Feb 2020 DE2.3 – Electronics 2 Lecture 16 Slide 5

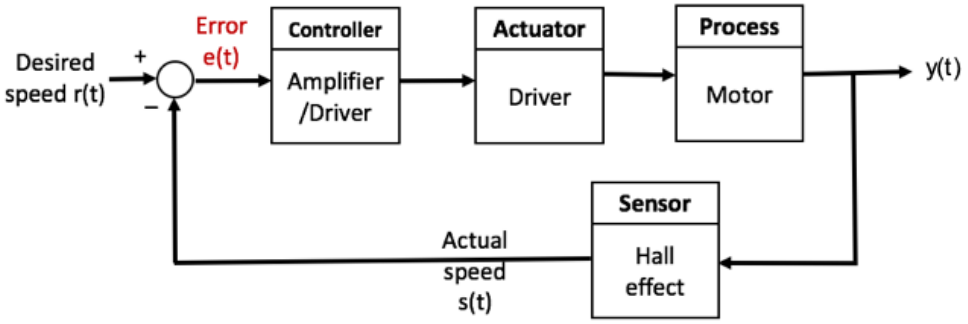
There are two further major problems with open-loop control.

The system (i.e. the motor) may experience disturbances or **perturbations**. Furthermore, the sensor may have **noise** associated with it.

These two factors p(t) and n(t) will affect the output behaviour.

A **closed-loop system** has the potential of mitigating against these undesirable factors.

Closed-loop control with feedback



- In a **closed-loop control system**, we use a **sensor** to detect the parameter that we wish to control. This parameter is also known as the "**control variable**".
- We obtain the **error signal** e(t) by subtracting the actual parameter from the desired parameter (called the "**set-point**").
- The **controller** then produces a **drive signal** to the actuator and to the plant depending on this error signal.

Here is a generic **closed-loop control system** employing **feedback**.

The **control variable** is detected using the **sensor**, producing the sensed value s(t). The sensed value is compared with the set point r(t) (i.e. the reference value) to produce the error signal e(t).

The error signal e(t) is used to as input to the **controller** (such as an amplifier) in order to provide the signal for the actuator which drives the system.

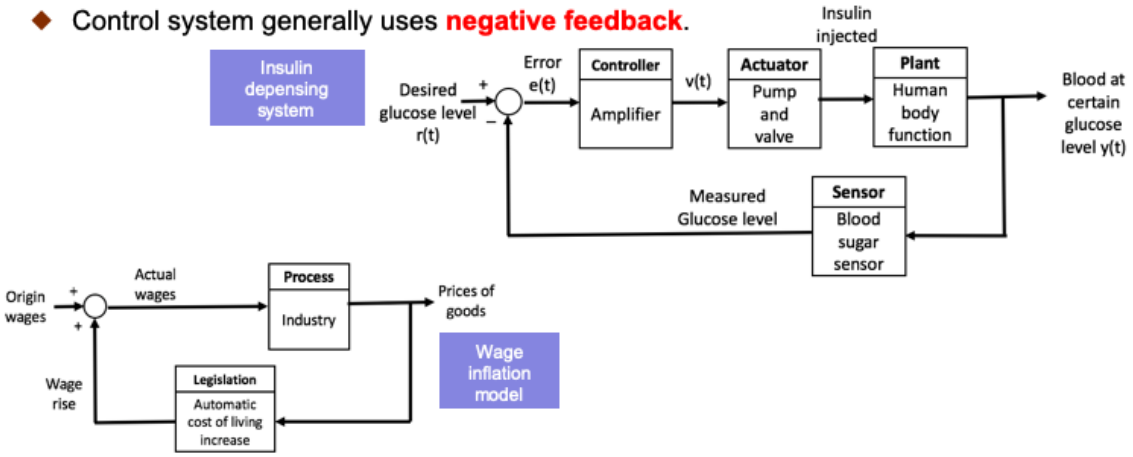
The key to this feedback control system is that the system is driven by the controller that responses to the difference between what is desired r(t) and what actually happens at the output y(t) measured by the sensor.

The purpose of this loop back (closed-loop feedback) is that by choosing or designing a proper controller, we can:

1. Provide **regulation function**, i.e. make the control variable tracks the set point.
2. Make the closed-loop system **immune to variations** in the process (e.g. motor characteristics).
3. Make the closed-loop system **less prone to disturbances** or perturbations and reduce the effect of noise.
4. **Change the system dynamic** behaviour such as its step response to what you want.
5. **Make** an inherently unstable system (such as a two-wheel vehicle) **stable**.

Negative vs Positive feedback

- Negative example: sensor of the control variable is SUBTRACTED from the desired parameter. Here is a control system for dispensing insulin to a diabetic patient.
- Control system generally uses **negative feedback**.



- A system could have positive feedback. Here is a model for wage inflation. Such a system will have its control parameter ever-increasing. Such a system is **not stable**, meaning that it never reaches a stable final value.

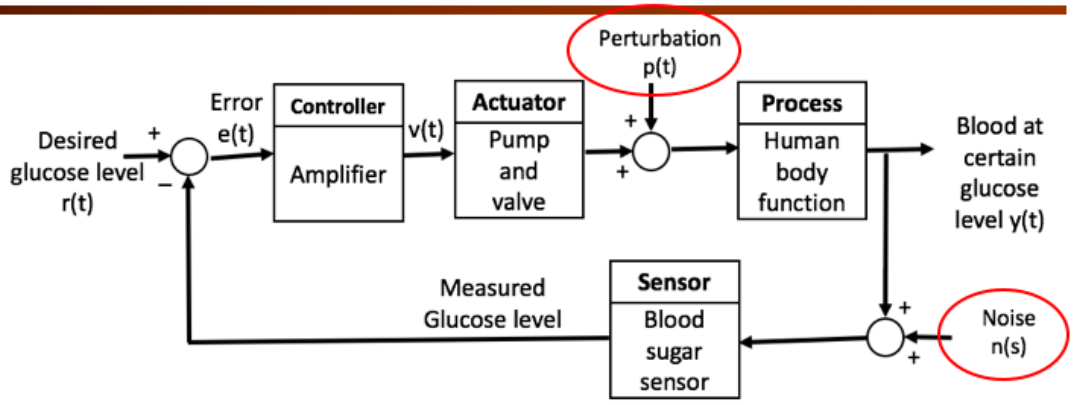
Note that we usually use **negative feedback** in order to control a system. We always subtract the output variable (or some form of feedback value) from the set point, not add.

If you add instead of subtract, the system will go unstable or goes to infinity – it would not settle to a final value. This is called **positive feedback.**

Show here is a **positive feedback** system modelling wage inflation – that's why in a normal economy, the wages always increase and will not stay at a steady-state level.

We only consider **negative feedback** in this module.

Closed-loop system with disturbance & sensor noise



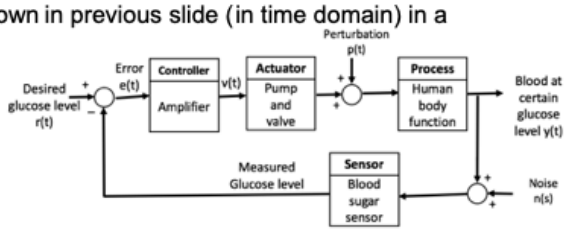
- Again all systems are not ideal and there can be **perturbation** and sensor **noise**.
- These are added to the insulin dispensing system which is under closed-loop control

We must now add two other real-life factors into our closed-loop system model.

Here we add the **perturbation** or disturbance **p(t)** just before the process. Finally we add the **noise** signal **n(t)** injected into the sensor. These will have impact to the final output y(t) (as we will see later).

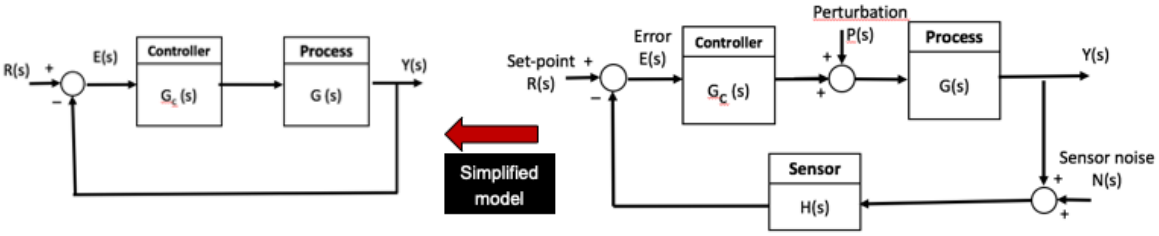
Block diagram model of a closed-loop system

- We can represent a **closed-loop system** shown in previous slide (in time domain) in a mathematical form in the Laplace domain.
- G(s) is the **transfer function** of the system we wish to control.
- $G_C(s)$ is the **controller** that we design in s-domain.
- H(s) is the **sensor** characteristic.



- R(s) is the **desired** parameter (e.g. a dc value, a step function or a ramp function).
- Y(s) is the actual **output variable** under control.
- We can simplify the system by assuming that H(s) = 1, and both perturbation and sensor noise are neglected for now (i.e. assumed to be zero).

Laplace Transform



So far, we have drawn our system as blocks of functions. In order to do analysis on such a system, we must model it mathematically. That's why you need to learn modeling of systems earlier.

The technique we use to analyse system in order to design a controller for it in a closed-loop fashion is through **Laplace transform**.

Here we assume that the actuator and the process (or system, or plant) together has a **system transfer function** in the Laplace domain of G(s).

The **sensor** in the feedback path may have a transfer function of H(s).

Our goal is to **design a controller** $G_C(s)$ which, when we close the loop, will make the system behaves in the way we want.

We often simplify the system by assuming that it has no disturbance, and the sensor has no noise (P(s) = N(s) = 0), and finally H(s) =1. Then we get the simplified model as shown. In this model, we basically have the process G(s) and the controller $G_C(s)$ which we need to design, and connected in a negative feedback configuration.

A video on open- & closed- loop systems

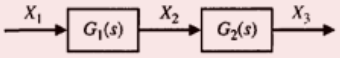
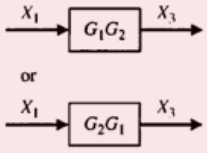
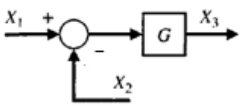
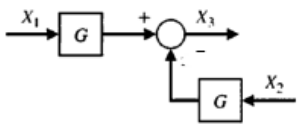
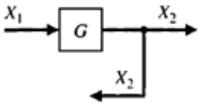
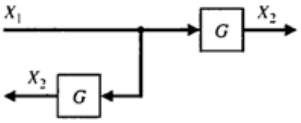
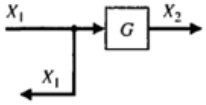
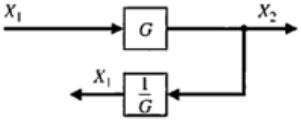
Before I move on, here is good YouTube video comparing the open-loop and the closed-loop system.

You can find this video on the link:
https://youtu.be/zjDM1qZaJ6Y

(See course webpage.)

Block diagram transformations (1)

- Here are some useful transformation in s-domain that helps with complexity reduction:

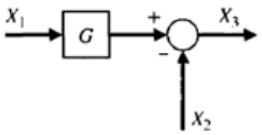
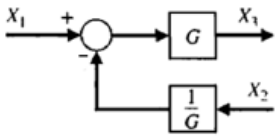
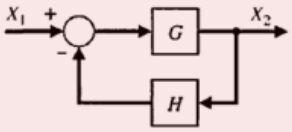
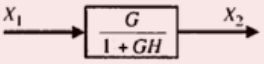
Transformation	Original Diagram	Equivalent Diagram
1. Combining blocks in cascade		
2. Moving a summing point behind a block		
3. Moving a pickoff point ahead of a block		
4. Moving a pickoff point behind a block		

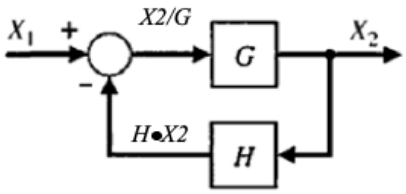
Before we move on, let use consider how to take the block diagram in the Laplace domain and perform some transformations in order to achieve a simplified diagram and to derive the system level transfer function.

There are 6 transformations. However, transforms 1 and 6, highlighted in red, are the IMPORTANT ONES.

Here is shown transforms 1 to 4. They are pretty obvious.

Block diagram transformations (2)

Transformation	Original Diagram	Equivalent Diagram
5. Moving a summing point ahead of a block		
6. Eliminating a feedback loop		



$$X_1 - H \times X_2 = \frac{X_2}{G}$$
$$\Rightarrow X_1 = \frac{X_2}{G} + H \times X_2$$
$$\Rightarrow GX_1 = (1 + GH)X_2$$
$$\Rightarrow X_2 = \left(\frac{G}{1 + GH}\right)X_1$$

The most important transform is number 6 – transforming a generic feedback loop into a transfer function.

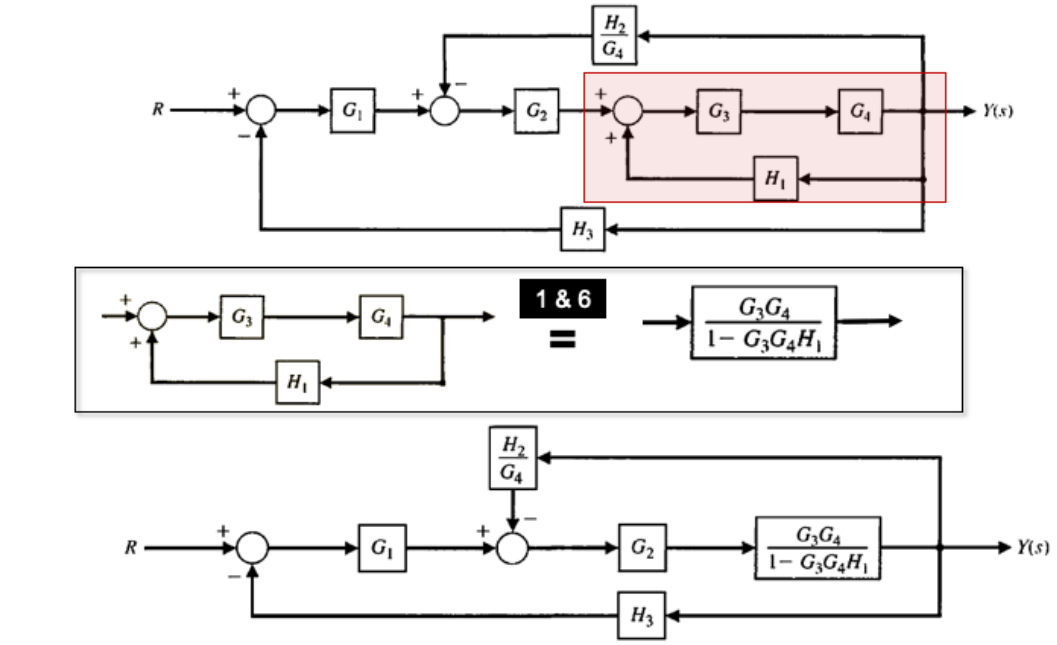
What I have shown here is the derivation of this transformation. Make sure that you are able to do the derivation yourself.

The transfer function for a feedback system with forward gain G(s) and feedback gain H(s) is:

$$\frac{G(s)}{1 + G(s)H(s)}$$

This equation is one of the very few that are worth memorizing!

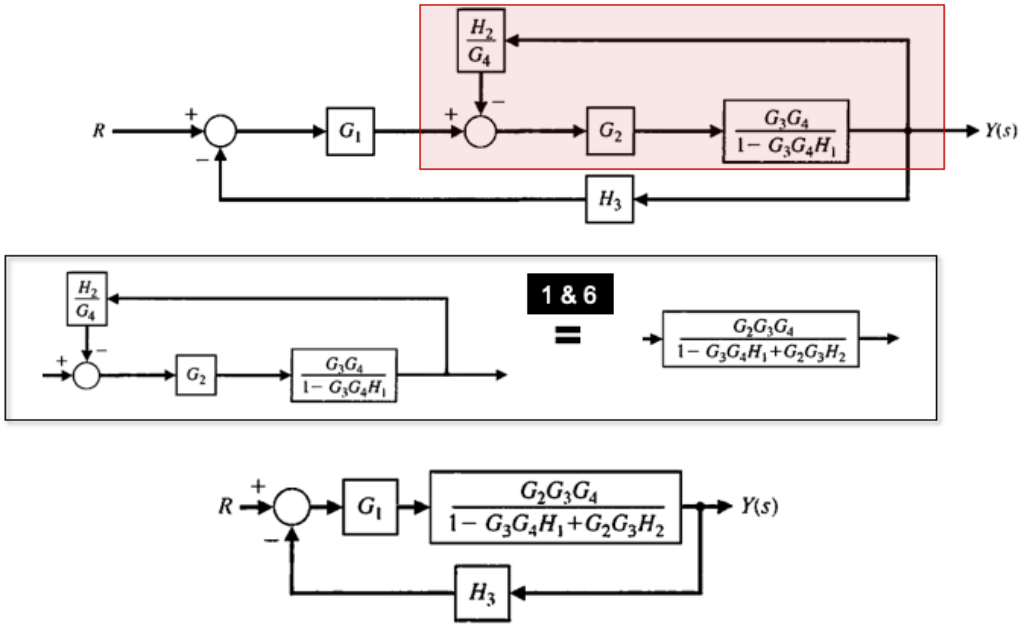
Example of system reduction by transformation (1)



Now let me go through step-by-step how a complex system block diagram can be reduced to a transfer function in the s-domain using an example.

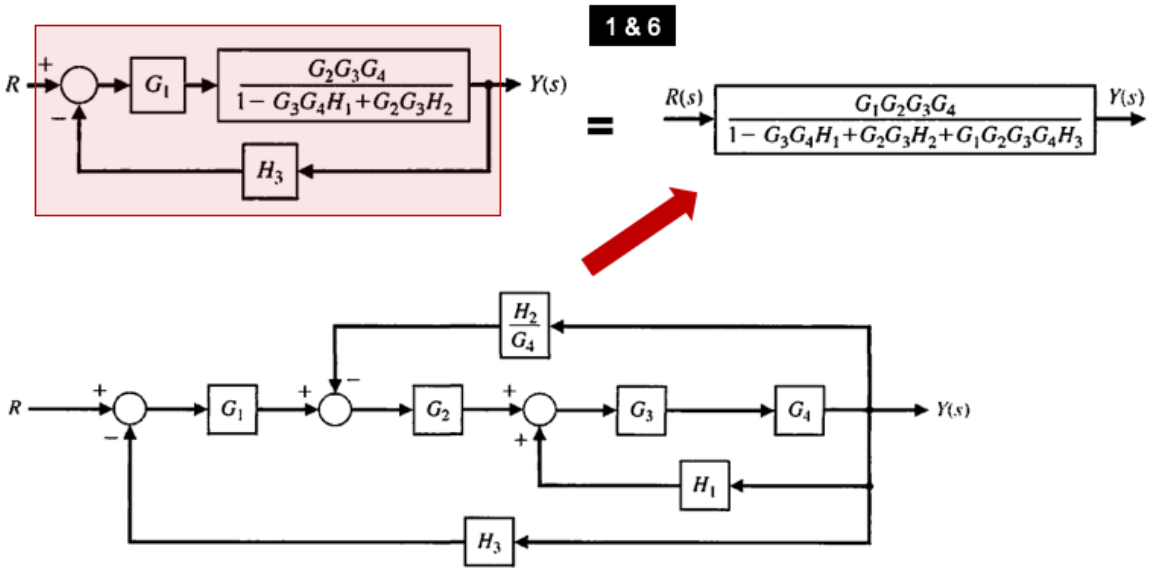
Make sure that you can follow each step.

Example of system reduction by transformation (2)



Keep following the steps.

Example of system reduction by transformation (3)



Here we show how the complex block diagram is reduced to the expression shown. Each term in the expression Gx and Hx is a transfer function block in the original diagram.

In practice we will not be dealing with any system that is as complex as the one shown here. Nevertheless, it is worth learning the technique to do such simplification for ANY system.