Imperial College London

Lecture 15

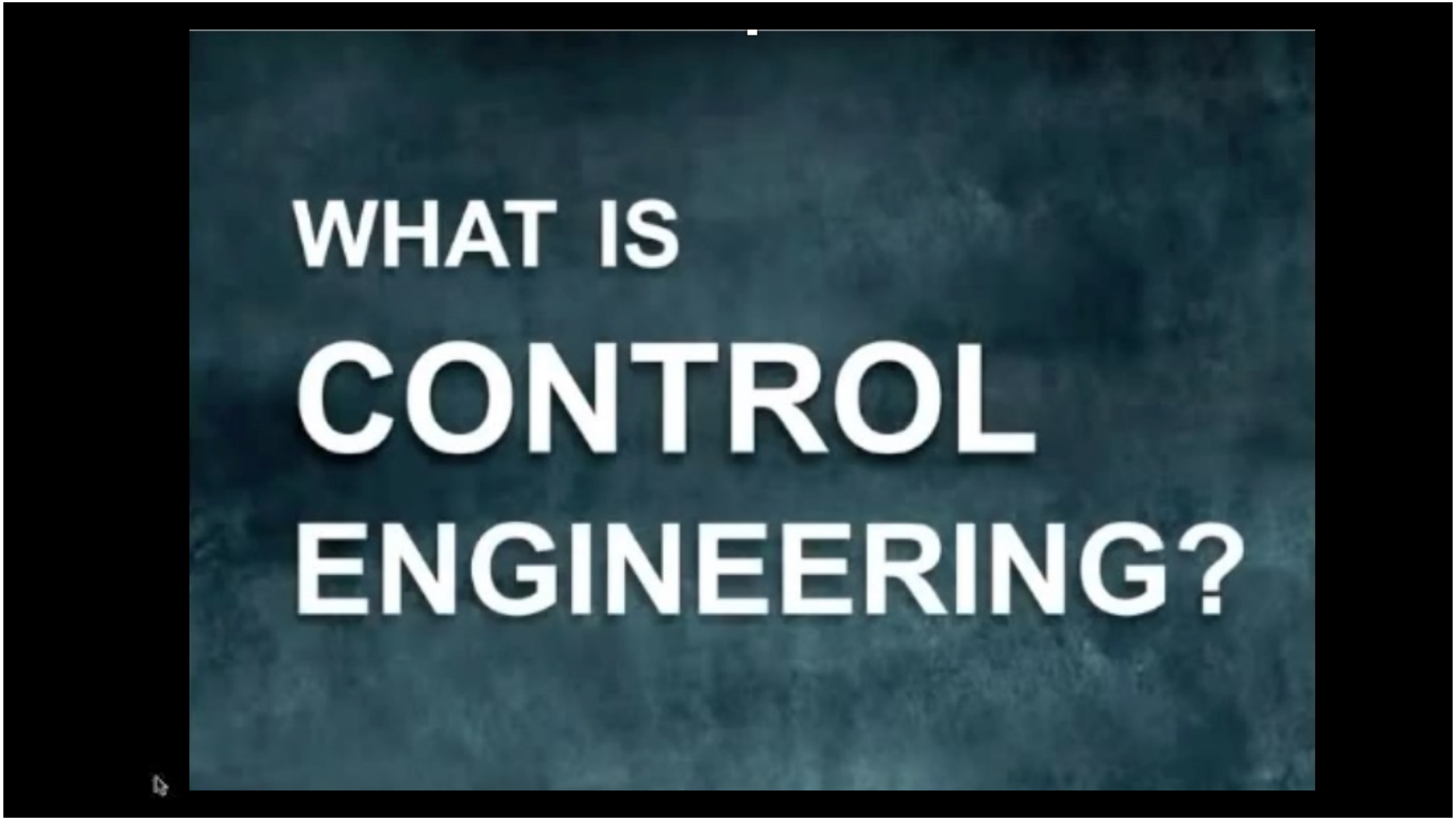
Introduction to Feedback Control

Prof Peter YK Cheung

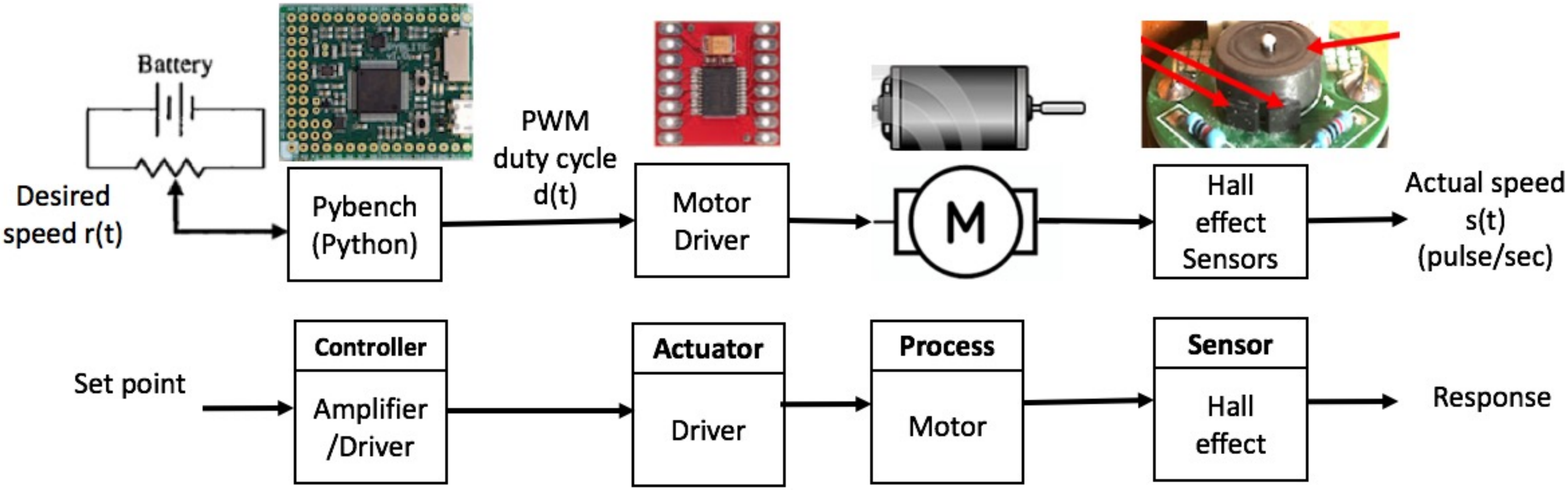
Dyson School of Design Engineering

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

What is control engineering? (a video)

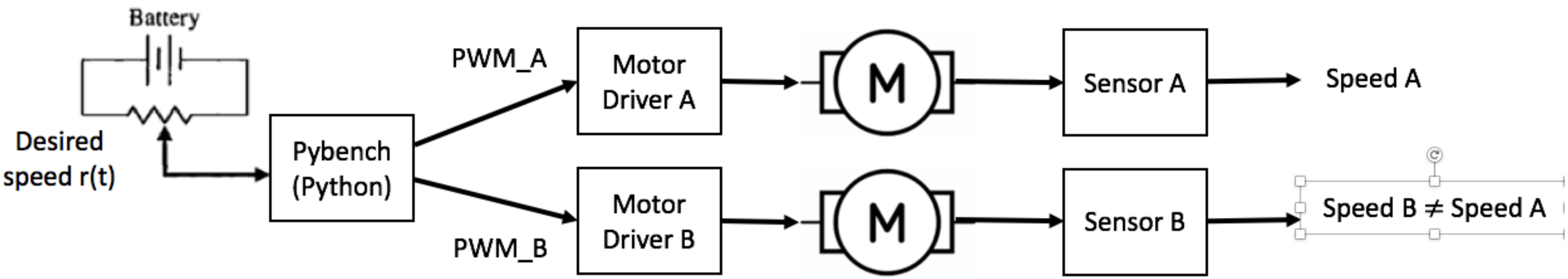


Driving the DC motors – Open-loop control



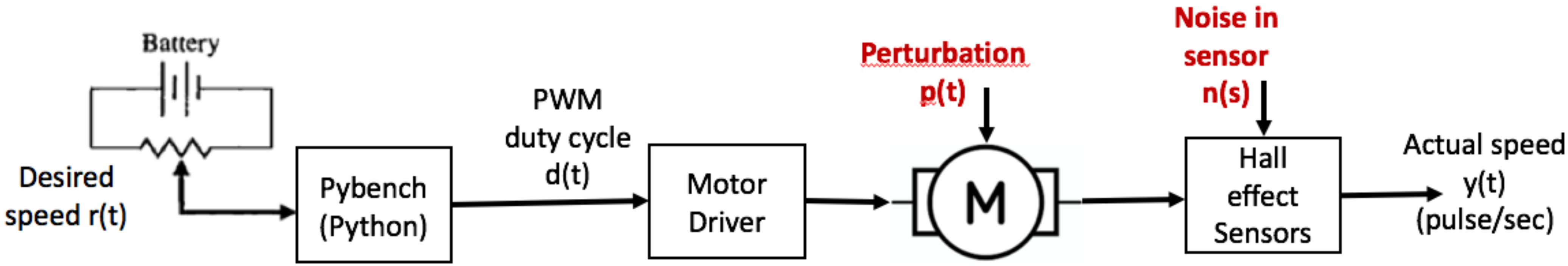
- Driving the DC motors using Pybench in Lab 5 is known as "**open-loop control**"
- Potentiometer set the required speed (as voltage value)
- The Pybench board running Python produces control signals including direction (A1, A2) and PWM duty cycle. It acts as the **controller**
- The TB6612 H-bridge 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

Problem 1: Uncertainty in system characteristic



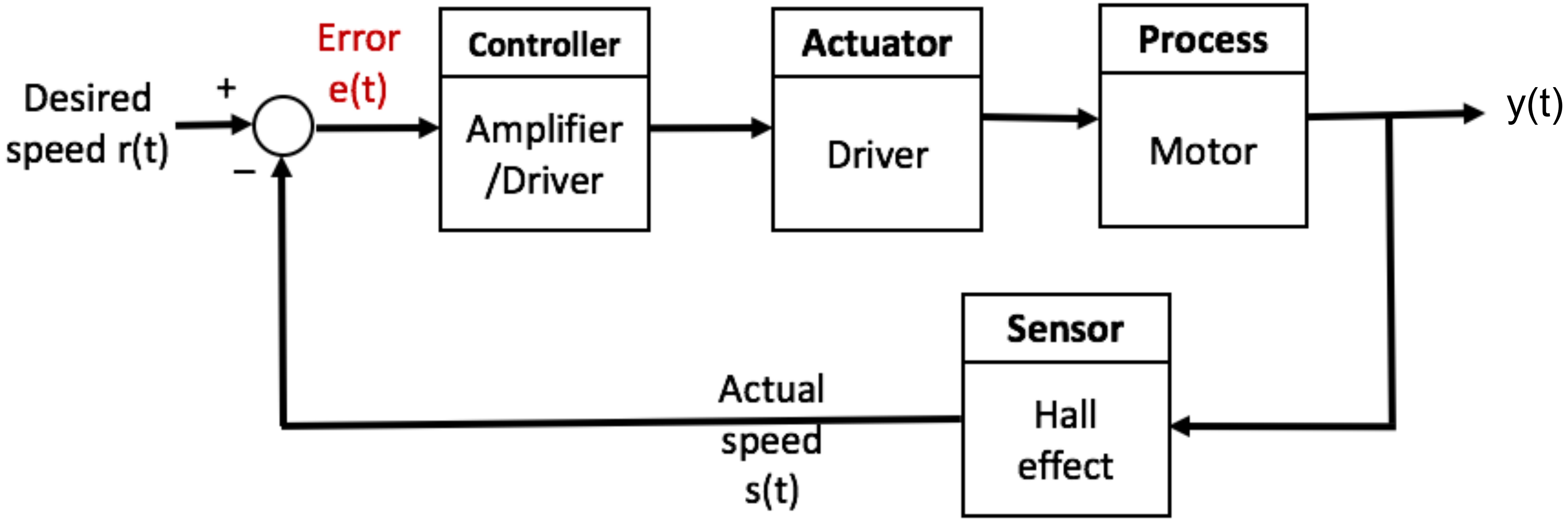
- There are many problems with open-loop control.
- First, the two motor may not respond in the same way to the drive input signal PWM_A and PWM_B. (For example, the two gear boxes may present different resistance to the motor, and the magnet inside the motors may have different strength.)
- The consequence 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.

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.

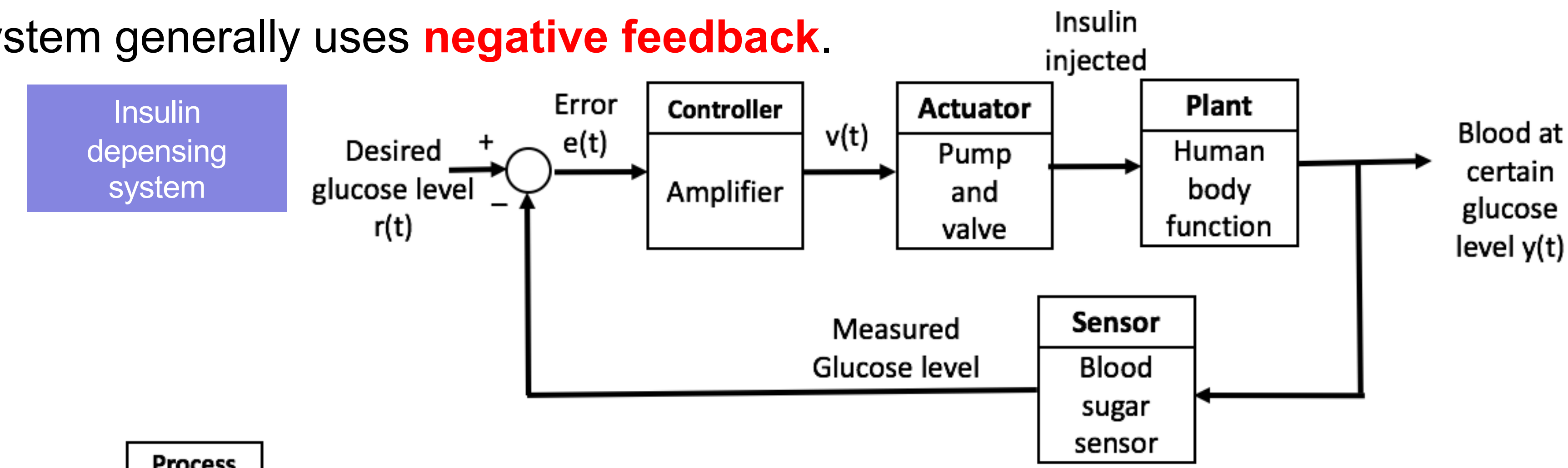
Closed-loop control with feedback

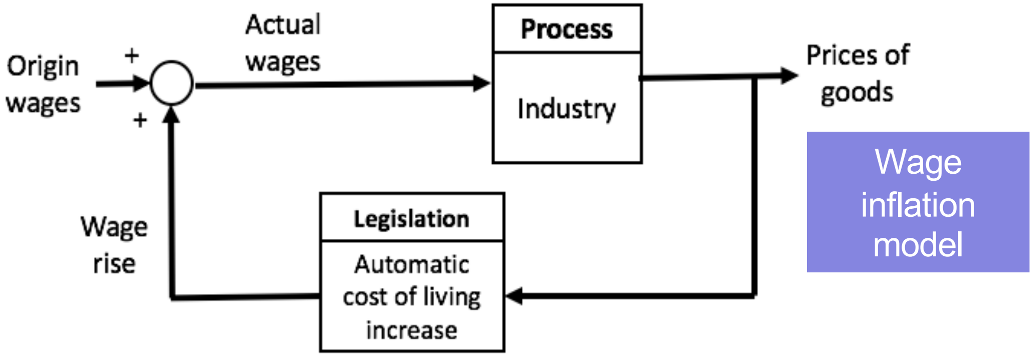


- In a **closed-loop control system**, we use a **sensor** to detect the parameter that we wish to control. This parameter is also known as the "**control variable**".
- We obtain the **error signal** e(t) by subtracting the actual parameter from the desired parameter (called the "**set-point**").
-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

Negative vs Positive feedback

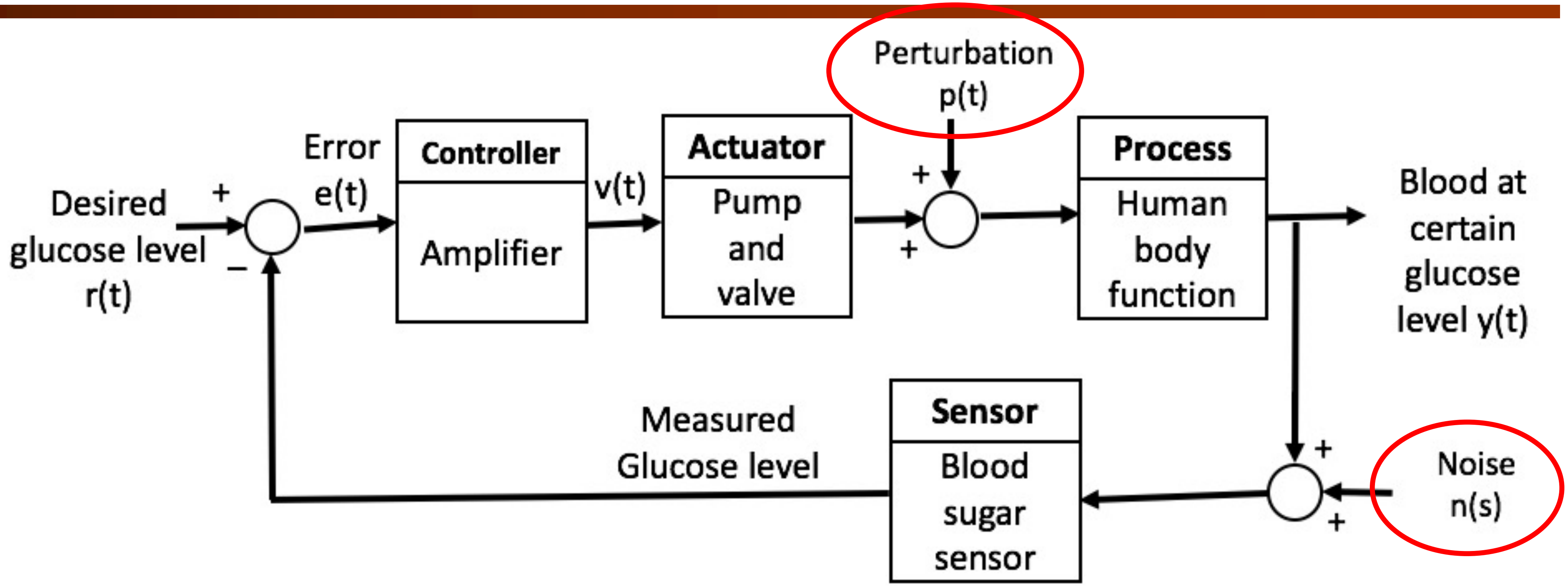
- **Negative feedback** example: sensor of the control variable is SUBTRACTED from the desired parameter. Here is a control system for dispensing insulin to a diabetic patient.
- Control system generally uses **negative feedback**.





- 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.

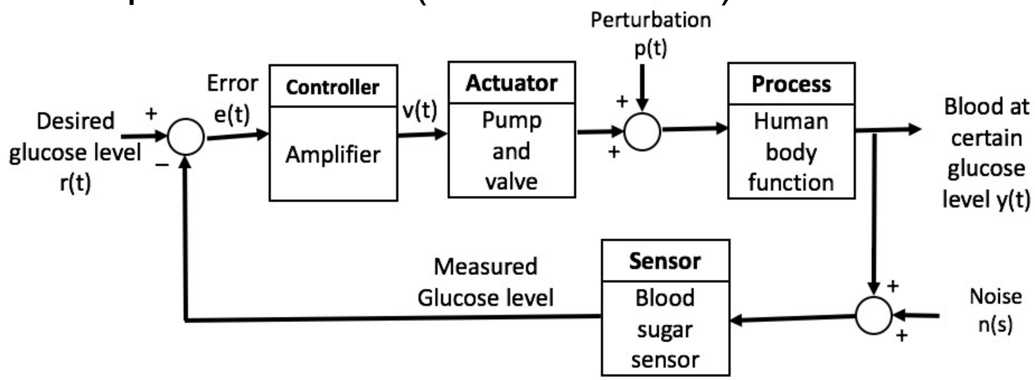
Closed-loop system with disturbance & sensor noise



- 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

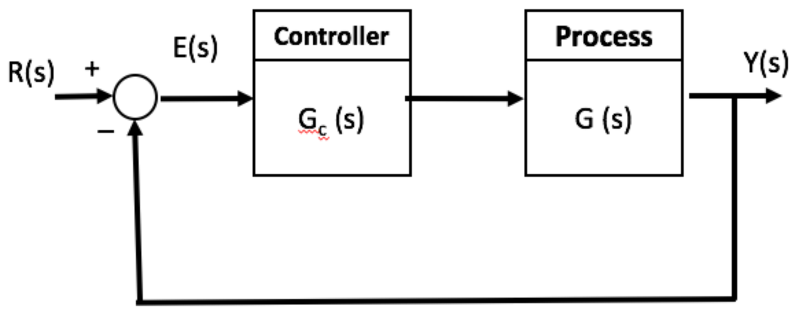
Block diagram model of a closed-loop system

- 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.

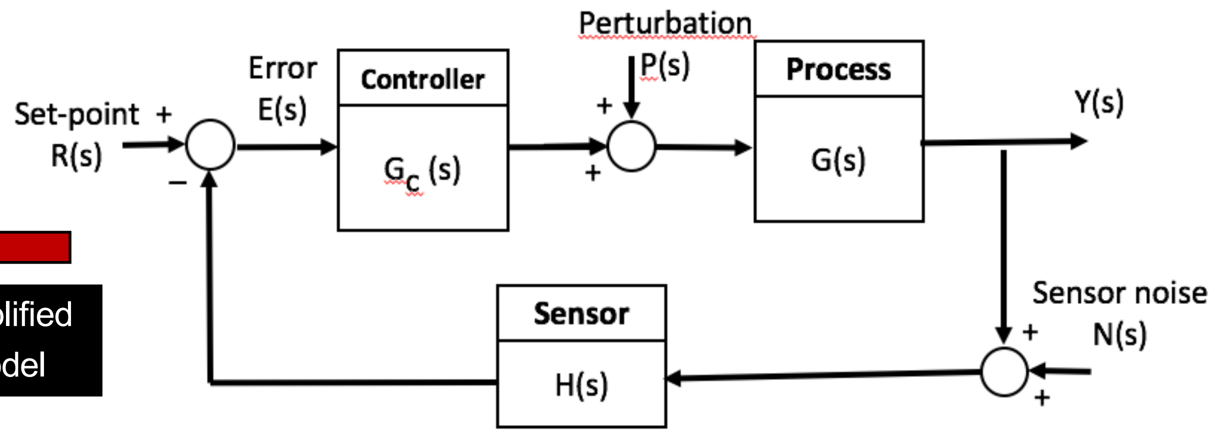


- R(s) is the **desired** parameter (e.g. a dc value, a step function or a ramp function).
- Y(s) is the actual **output variable** under control.
- We can simplify the system by assuming that H(s) = 1, and both perturbation and sensor noise are neglected for now (i.e. assumed to be zero).

Laplace Transform



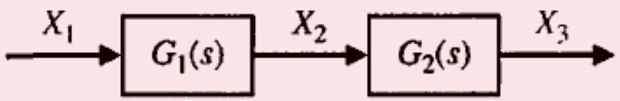
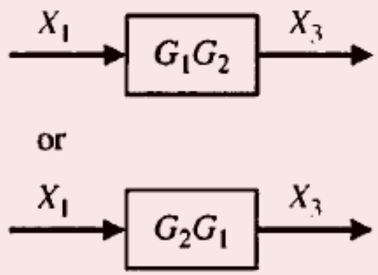
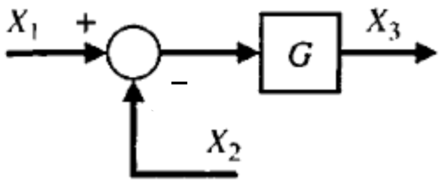
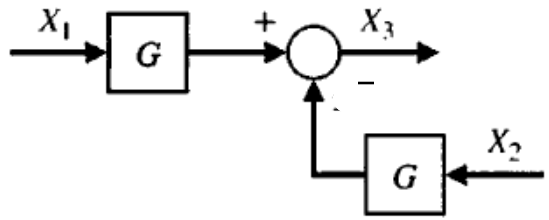
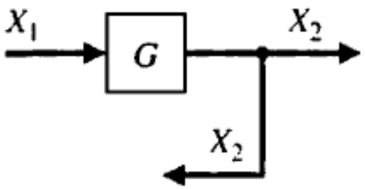
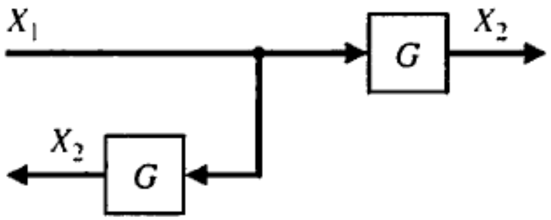
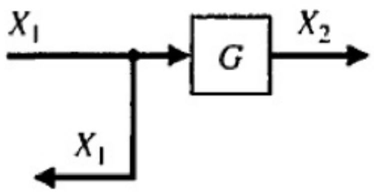
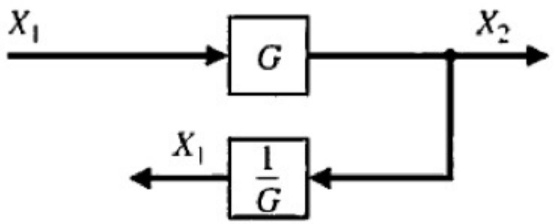
Simplified model



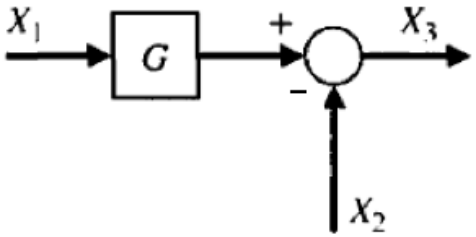
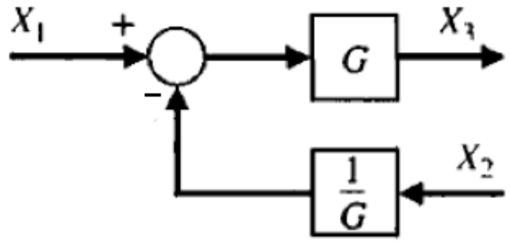
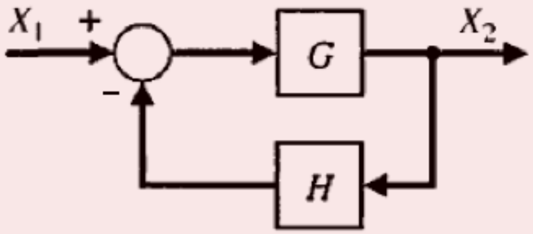
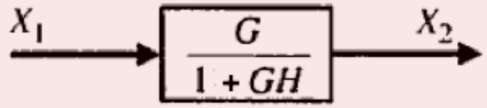
A video on open- & closed- loop systems

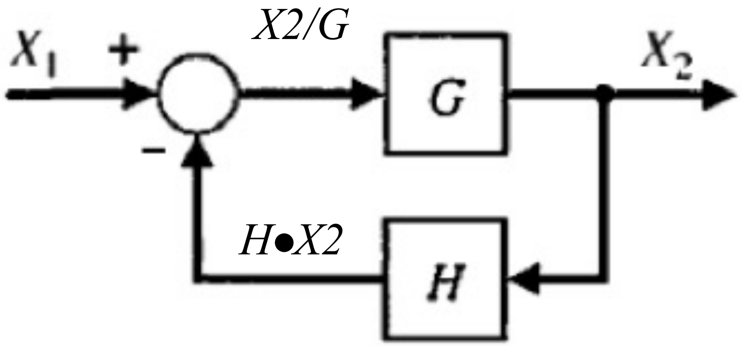
Block diagram transformations (1)

- 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		

Block diagram transformations (2)

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



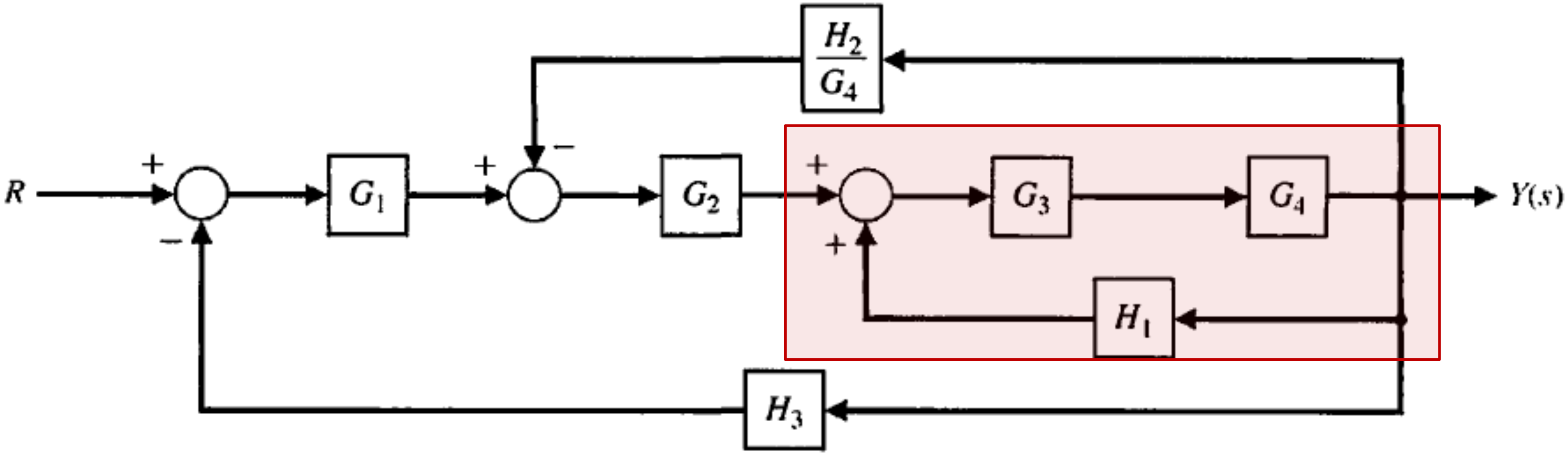
$$X_1 - H \times X_2 = \frac{X_2}{G}$$

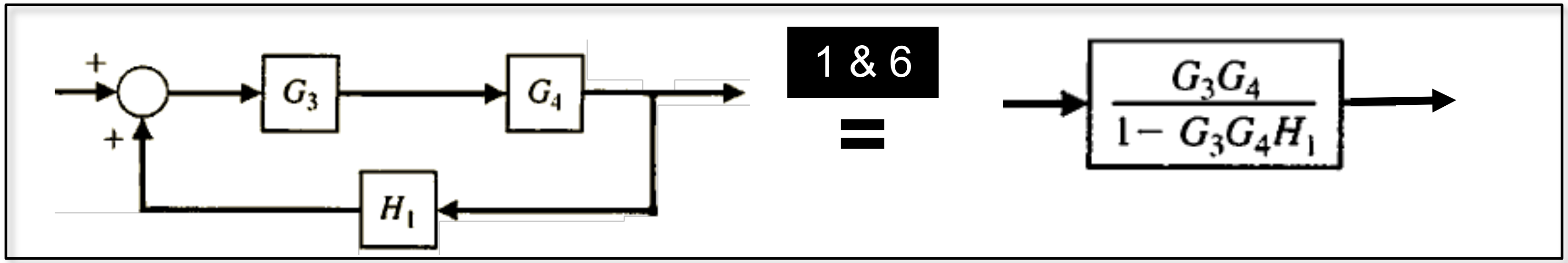
$$\Rightarrow X_1 = \frac{X_2}{G} + H \times X_2$$

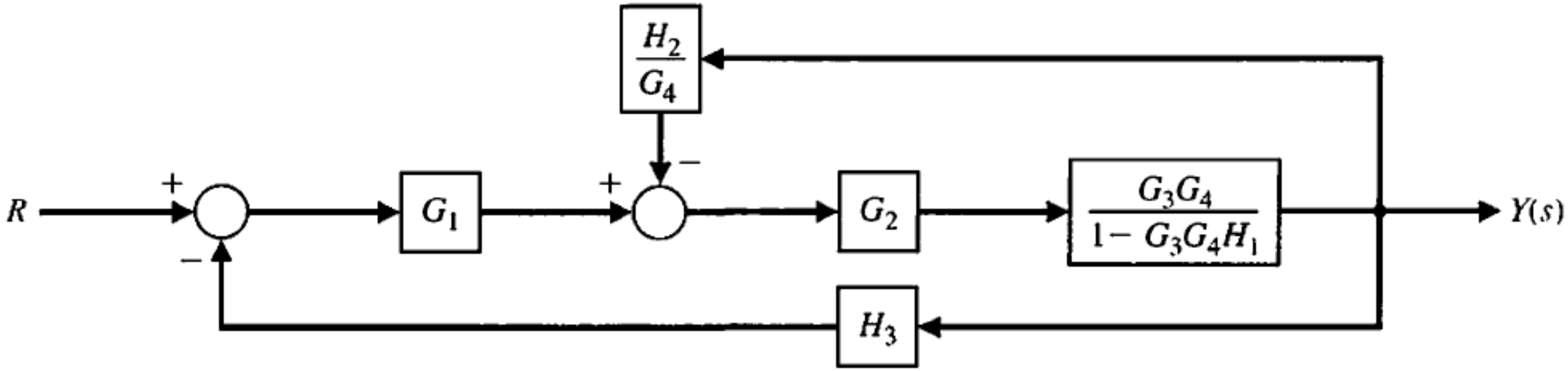
$$\Rightarrow GX_1 = (1 + GH)X_2$$

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

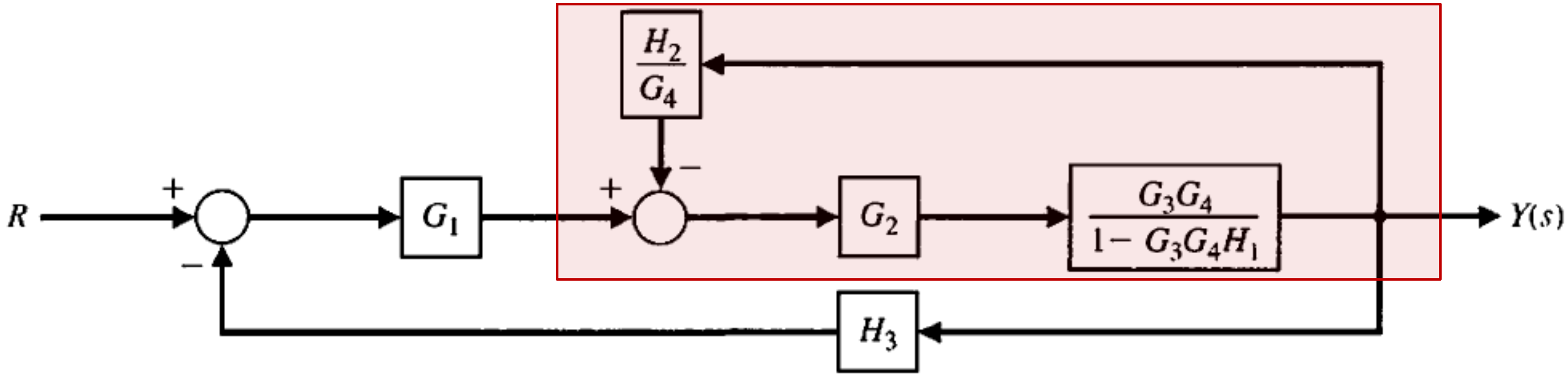
Example of system reduction by transformation (1)

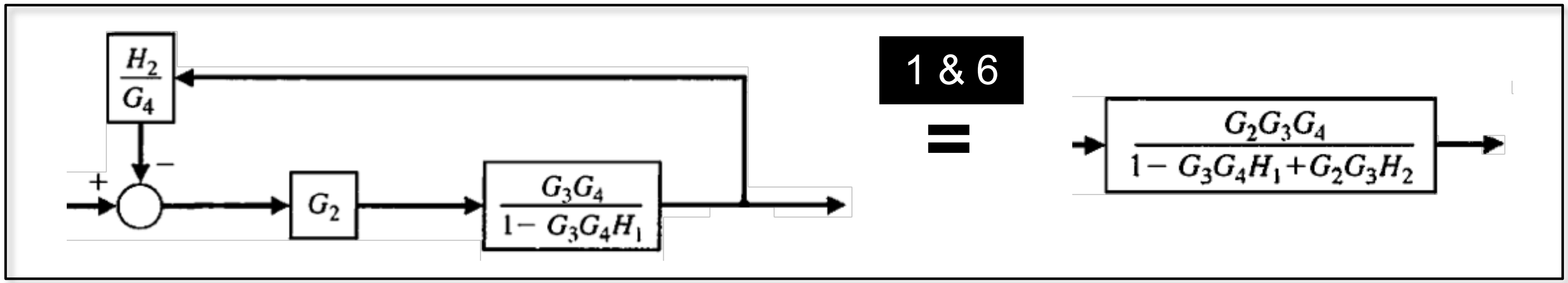


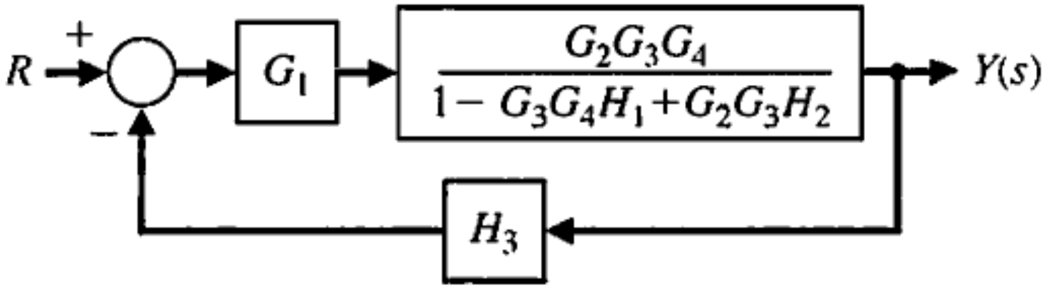




Example of system reduction by transformation (2)

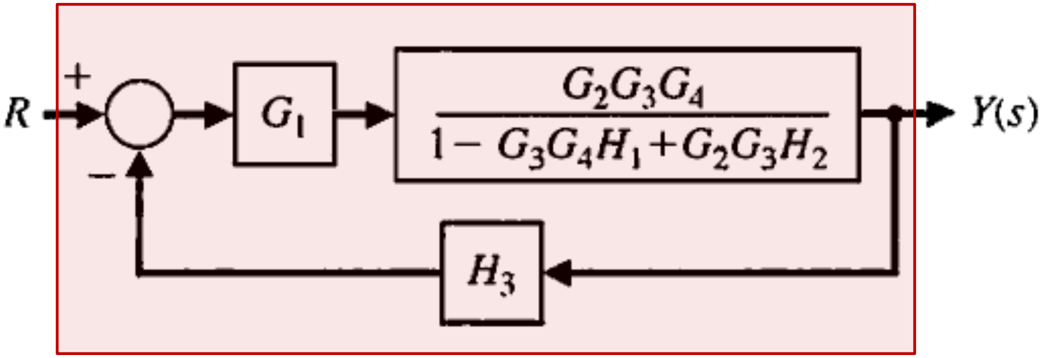




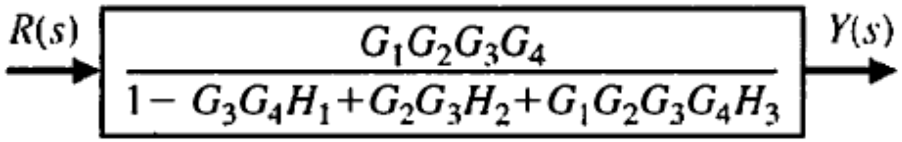


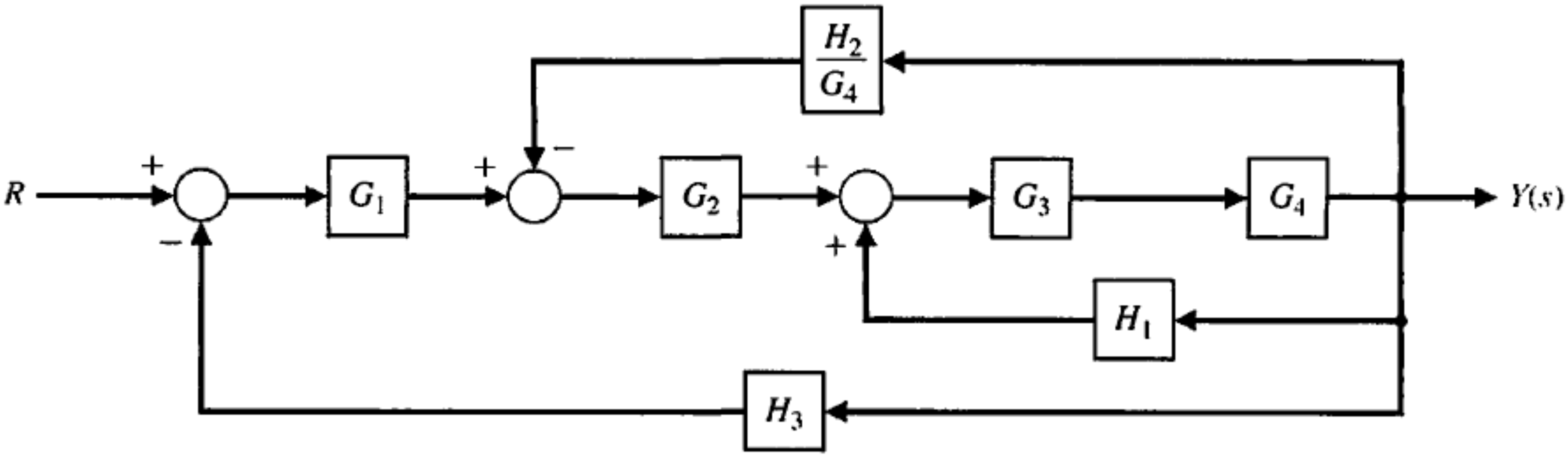
Example of system reduction by transformation (3)

1 & 6



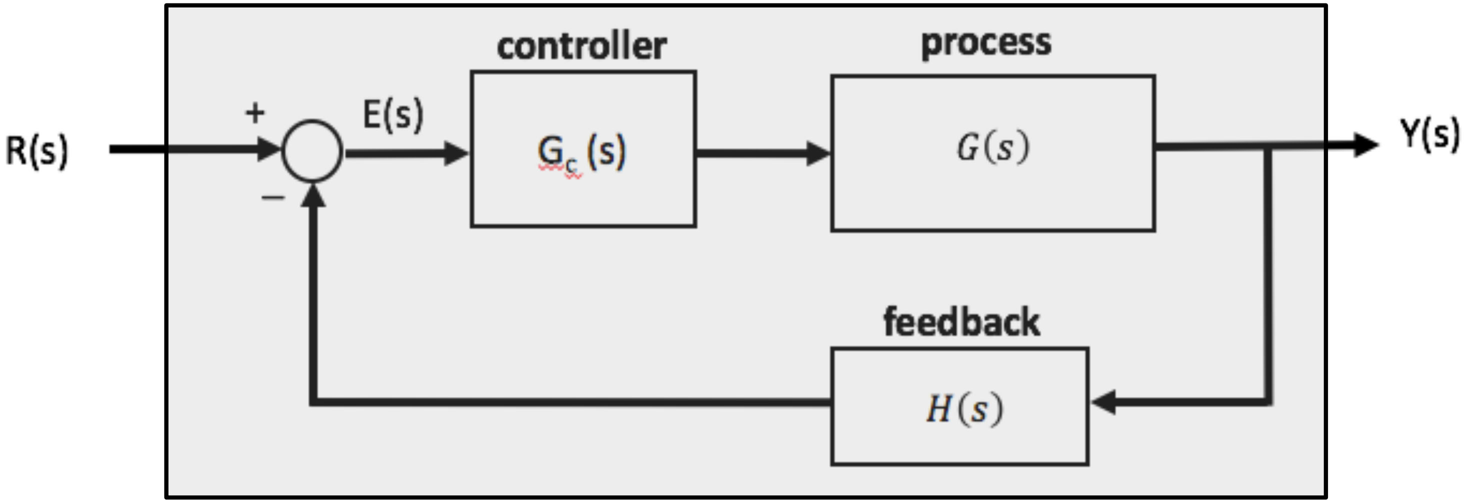
=





A generic closed-loop control system

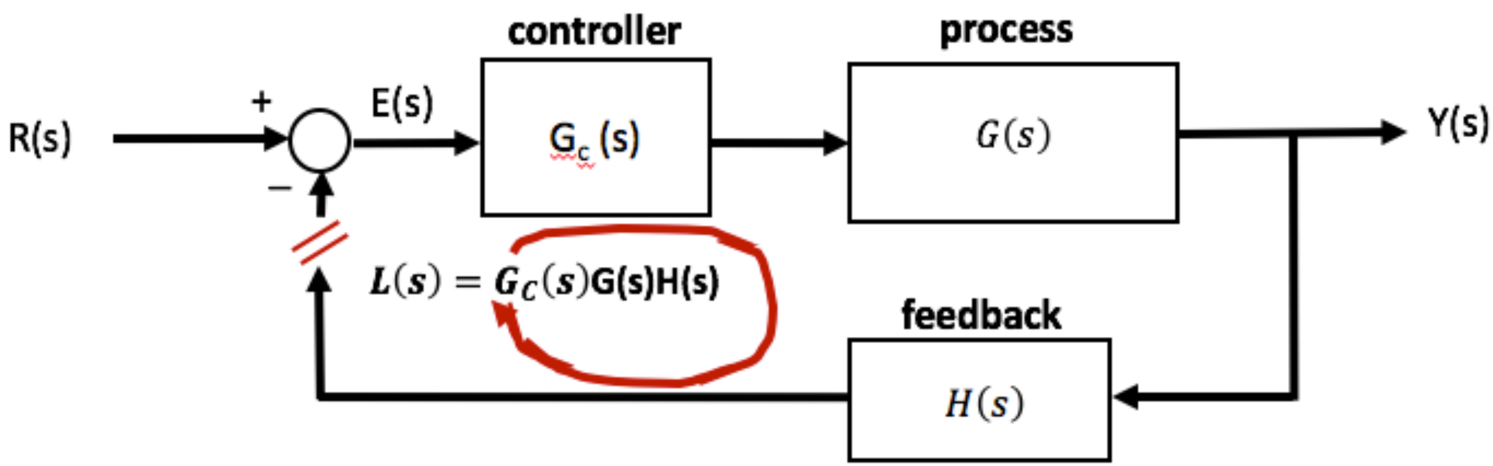
- Let us now consider a generic close loop system such as the motor or insulin pump control as shown here.



- The transfer function of the closed-loop control system from input R(s) to output Y(s) is (applying transforms 1 & 6):

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)H(s)}$$

The concept of loop gain L(s)

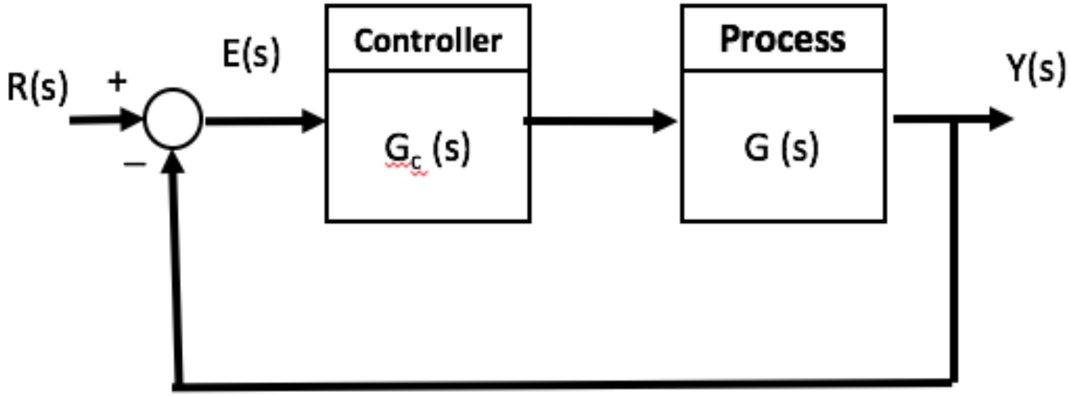


- From the previous slide, we have the transfer function of a close-loop system as:

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)H(s)} = \frac{G_c(s)G(s)}{1 + L(s)}$$

- The quantity: $L(s) = G_c(s)G(s)H(s)$ is known as **loop gain** of the system.
- It is the transfer function (gain) if you break the feedback loop at the point of feedback, and calculate the gain around the loop as shown.
- This quantity turns out to be most important in a feedback system because it affects many characteristics and behaviour in such a system.
- We will consider why such a closed-loop system with feedback is beneficial in the next Lecture.

Feedback makes system insensitive to G(s)



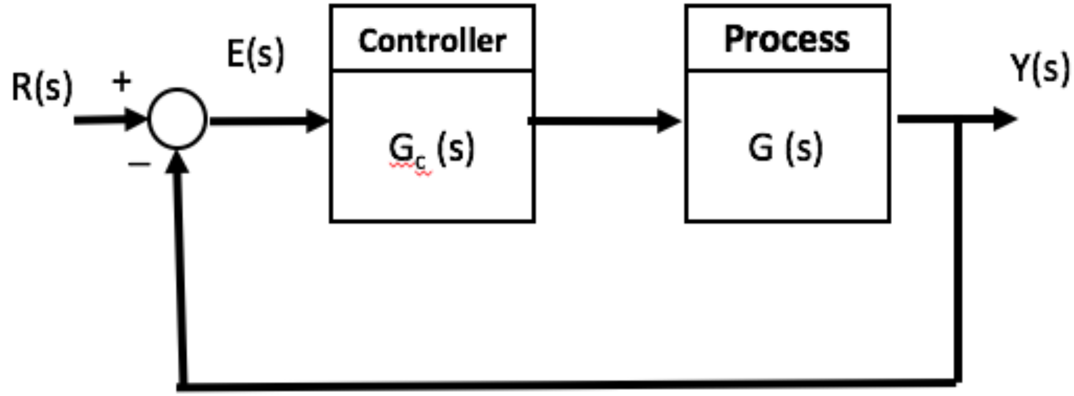
- Let us now assume that H(s) = 1 to simplify things.
- We have seen from the last lecture that the transfer function of this closed-loop system is:

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)} = \frac{L(s)}{1 + L(s)}$$

- If $L(s) = G_c(s)G(s) \gg 1$ then this term approaches 1!!
- In other words, the actual output Y(s) (e.g. motor speed) will track the desired input R(s) independent of G(s), our system behaviour:

$$\frac{Y(s)}{R(s)} \approx 1 \quad \text{if} \quad G_c(s)G(s) \gg 1$$

Feedback yields small steady-state error e(t)



- Let us suppose the input to the system is a step at t=0 with a magnitude of A: $r(t) = Au(t)$.
- Then $R(s) = A\frac{1}{s}$ (because Laplace transform of u(t) is 1/s)
- We know that in this system, y(t) will track r(t) from the previous two slides. The question is:

  "After transient has died down, what is error e(t)?"
- To calculate this steady-state error, we need to use the **final-value theorem**, which states:

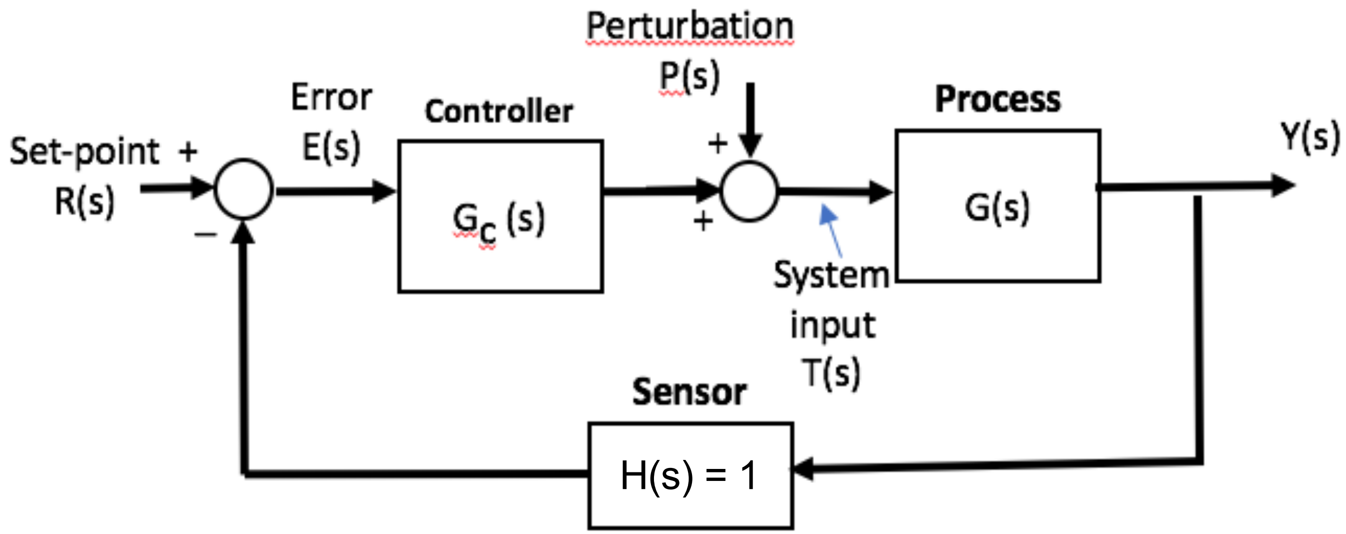
$$\lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s)$$

- Therefore,

$$\lim_{t \to \infty} e(t) = \lim_{s \to 0} sE(s) = \lim_{s \to 0} s\frac{1}{1 + L(s)} A\frac{1}{s} = \frac{A}{1 + L(0)}$$

- So the steady-state error is reduced by a factor of (1 + L(0))

Feedback reduces impact of perturbations



- Let us put back the perturbation p(t) to the system.
- Assume R(s) = 0, and the effect of perturbation P(s) on output Y(s) can be found by considering the expression for T(s) at the input to our system under control:

$$T(s) = P(s) - T(s)G(s)G_C(s)$$

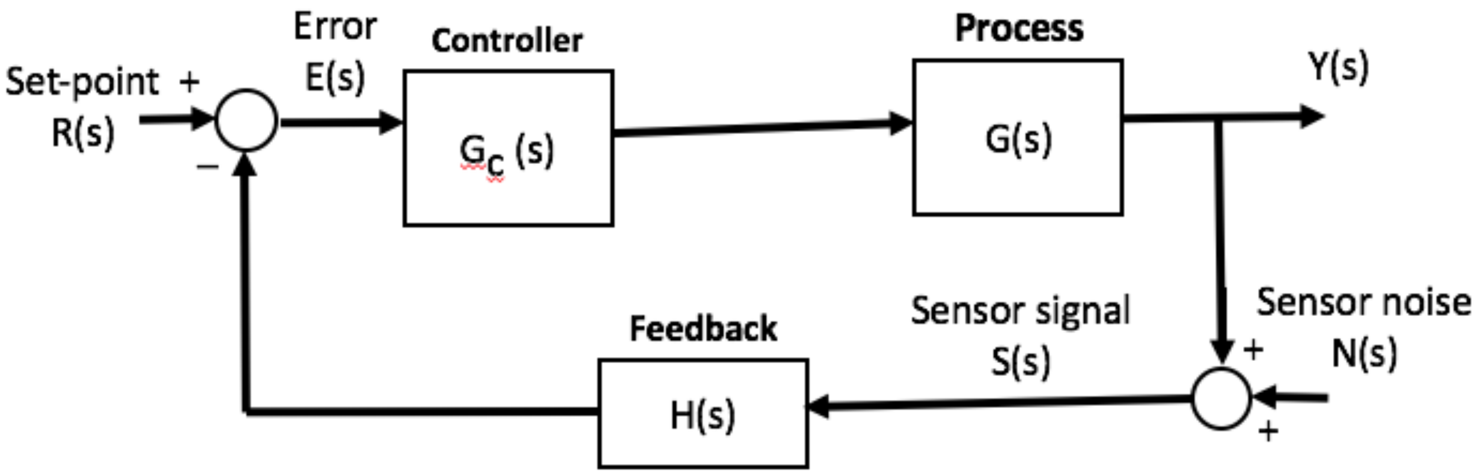
- In open-loop, $Y(s) = G(s)P(s)$

$$\Rightarrow T(s) = \frac{1}{1 + L(s)} P(s) = \frac{Y(s)}{G(s)}$$

$$\Rightarrow Y(s) = \frac{G(s)}{1 + L(s)} P(s)$$

- In closed-loop, the disturbance is reduced by the factor: $\frac{1}{1 + L(s)}$

Feedback introduces problem with sensor noise



- Let us put back the sensor noise n(t) to the system.
- Assume R(s) = 0, and the effect of N(s) on Y(s) can be found by considering the expression for S(s), the senor signal in the feedback path:

$$S(s) = N(s) - H(s)G_C(s)G(s)S(s)$$

- In open-loop, sensor is not an issue.

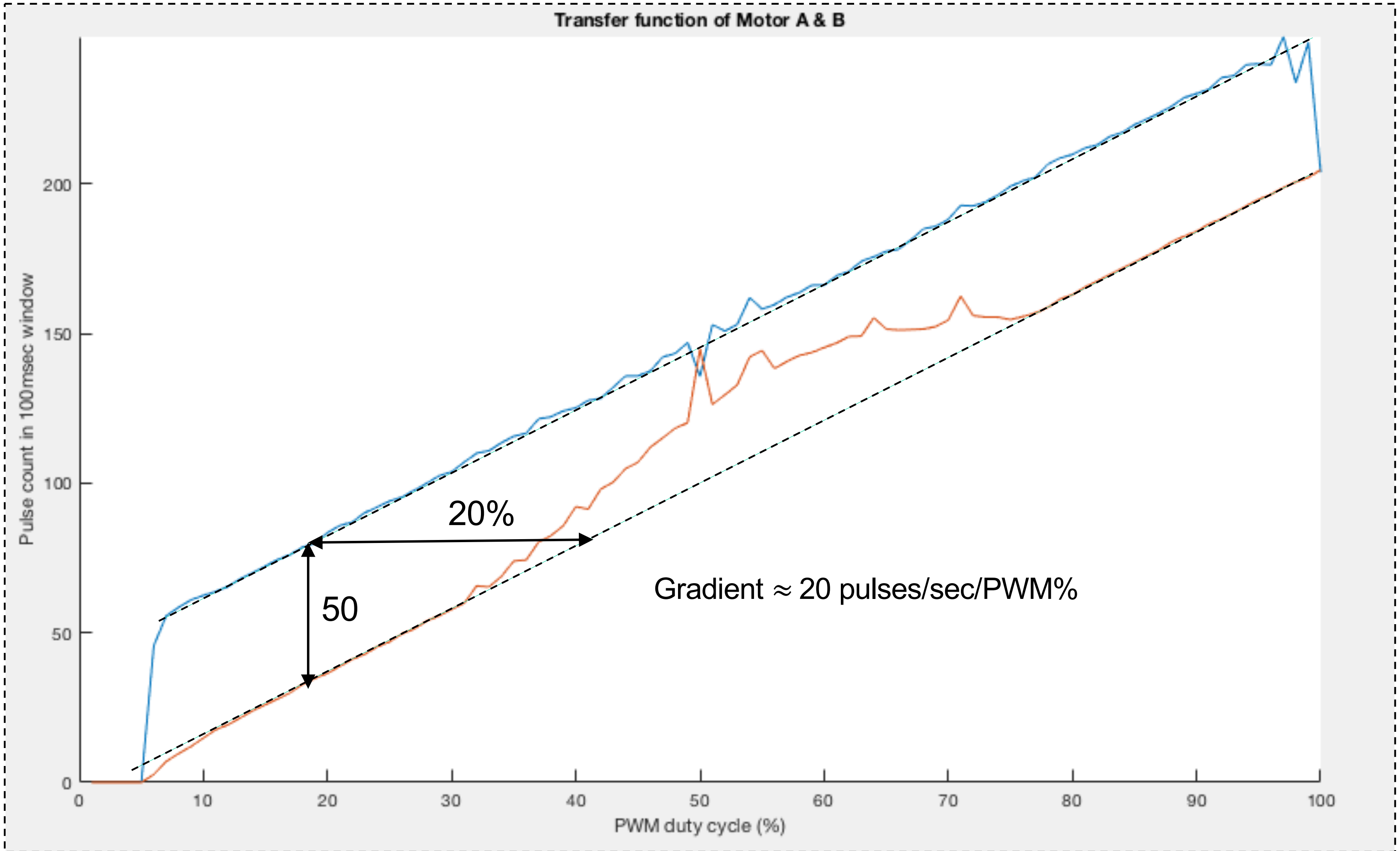
$$\Rightarrow S(s) = \frac{1}{1+L(s)}N(s)$$

$$\Rightarrow Y(s) = -L(s)S(s) = -\frac{L(s)}{1+L(s)}N(s)$$

- In closed-loop, we want L(s) to be small in order to have good attenuation of the sensor noise.
- This is in contradiction to the previous two properties. (We will consider this in more details later.)

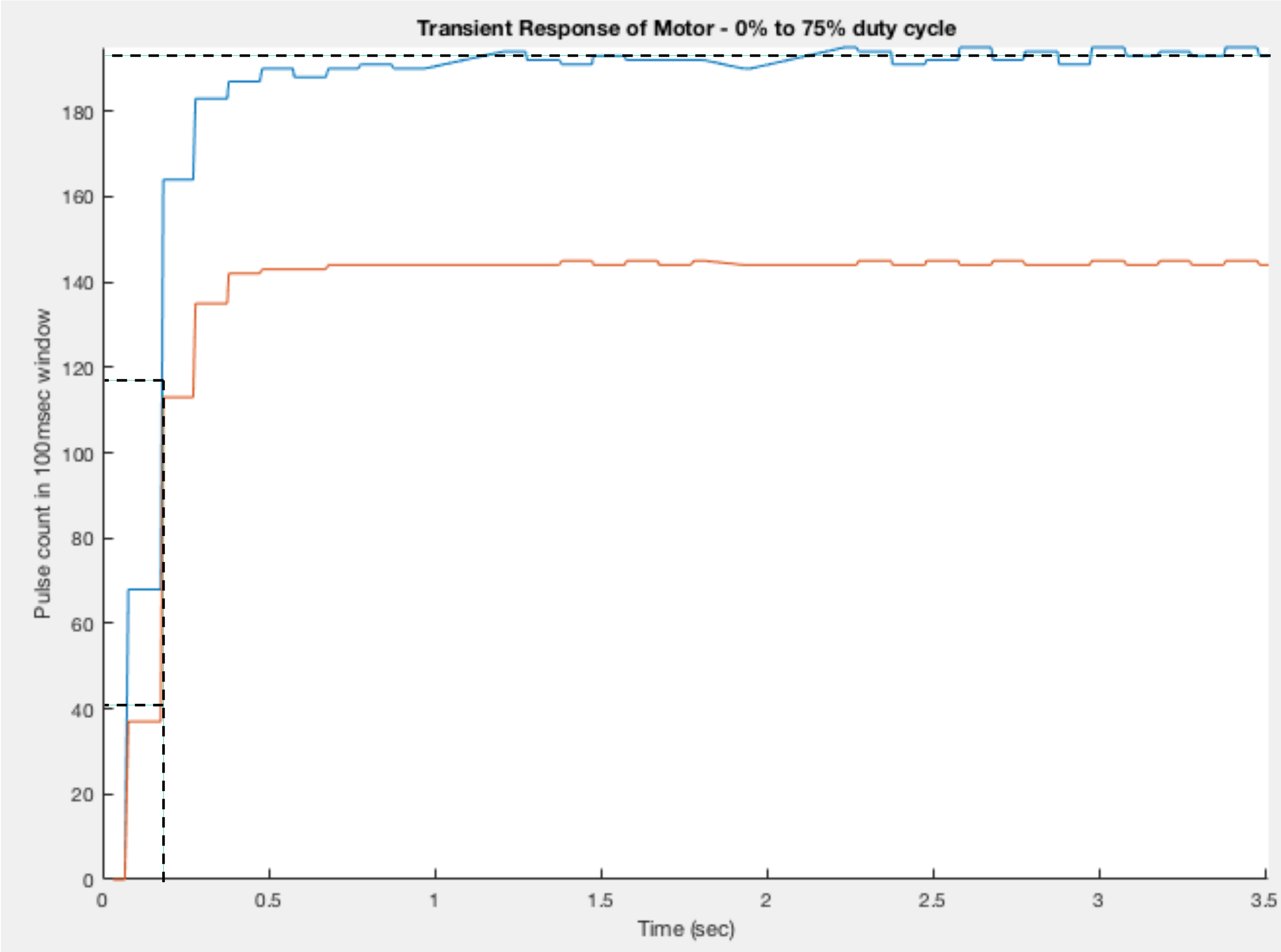
Practical process - Our DC Motors

- The two DC motors we use on the Segway may have very different characteristics.
- Here are plots of motor speed (in number of pulses per 100msec) vs PWM duty cycle for two typical motors:



Step response of the motor

- Here is the plot of the step response of two typical motors.
- The time constant (time it takes to reach 63% of final speed) is around 0.2sec.



Model of the motor – G(s)

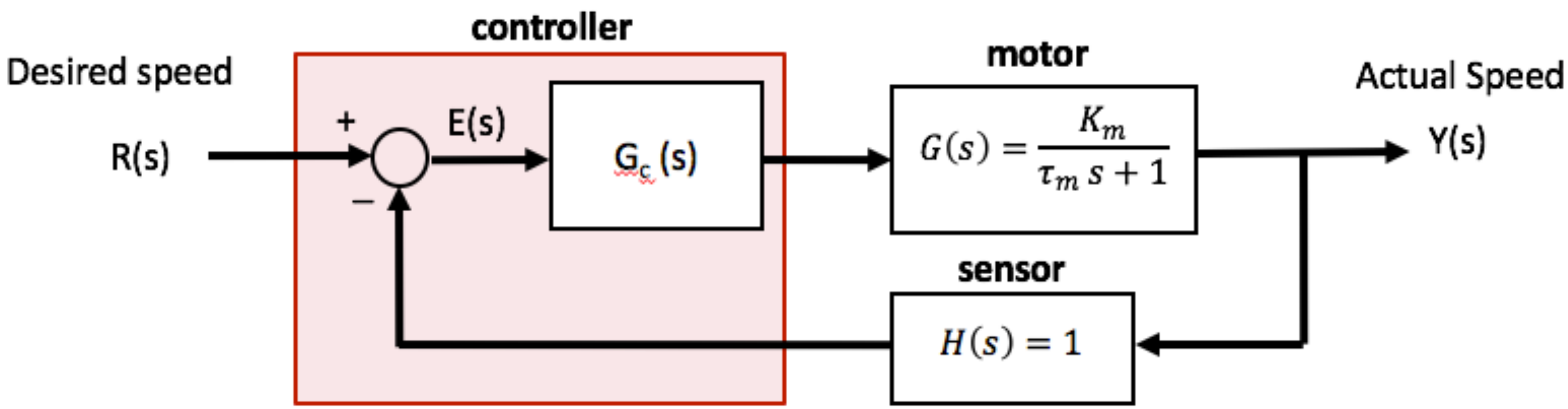
- We can model the motor as having a transfer function:

$$G(s) = \frac{K_m}{\tau_m s + 1}$$

- K_m is the dc gain, which is the gradient of the plot in slide 6 (i.e. the gain of the system when s = 0, or steady-state). Therefore K_m = 20 pulses/sec/PWM%
- τ_m is the time constant of the motor, which is estimated to be around 0.2sec in slide 7.
- Therefore:

$$G(s) = \frac{20}{0.2s + 1}$$

- Assuming H(s) = 1, we now put this motor in a feedback loop with a controller $G_c(s)$.



Proportional feedback

- Let us start with a simple controller with $G_c(s) = K_p$, where K_p is a constant.
- From transforms 1 & 6, we get:

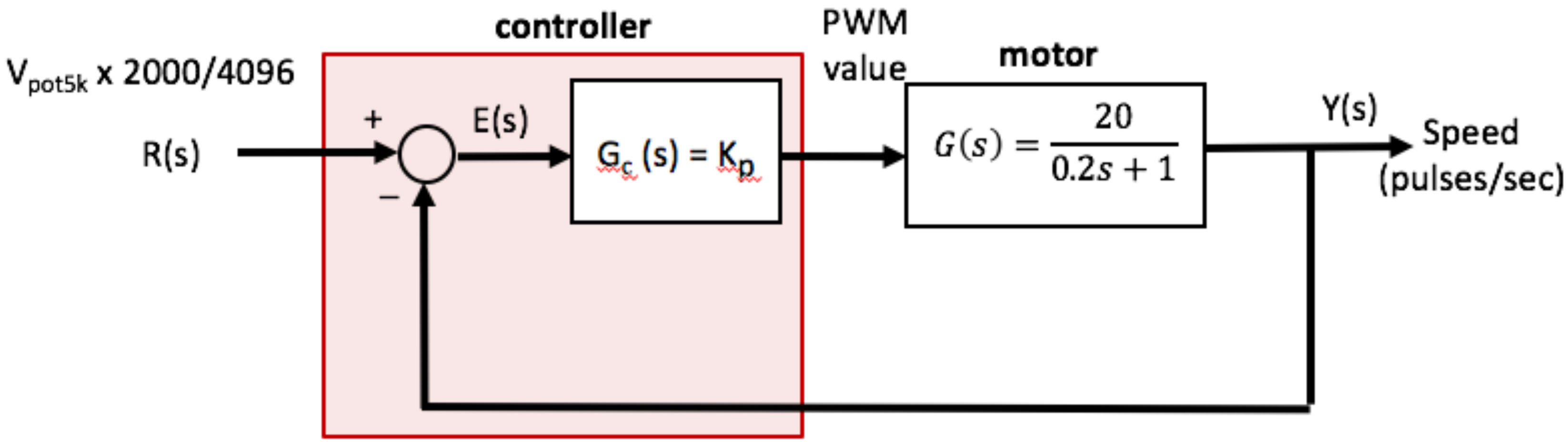
$$\frac{Y(s)}{R(s)} = \frac{L(s)}{1 + L(s)} = \frac{K_p \frac{20}{0.2s + 1}}{1 + K_p \frac{20}{0.2s + 1}}$$

- Therefore the closed-loop transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{20K_p}{1 + 20K_p + 0.2s} = \frac{20K_p/(1 + 20K_p)}{1 + \left(\frac{0.2}{1 + 20K_p}\right)s} = \frac{K_C}{1 + \tau_c s}$$

$$K_C = \frac{20K_p}{1 + 20K_p}$$

$$\tau_c = \left(\frac{0.2}{1 + 20K_p}\right)$$



How are things improved with proportional feedback?

- For our system, loop gain is L(s) = 20Kp for s=0. Assuming Kp = 5, we get a steady-state gain of:

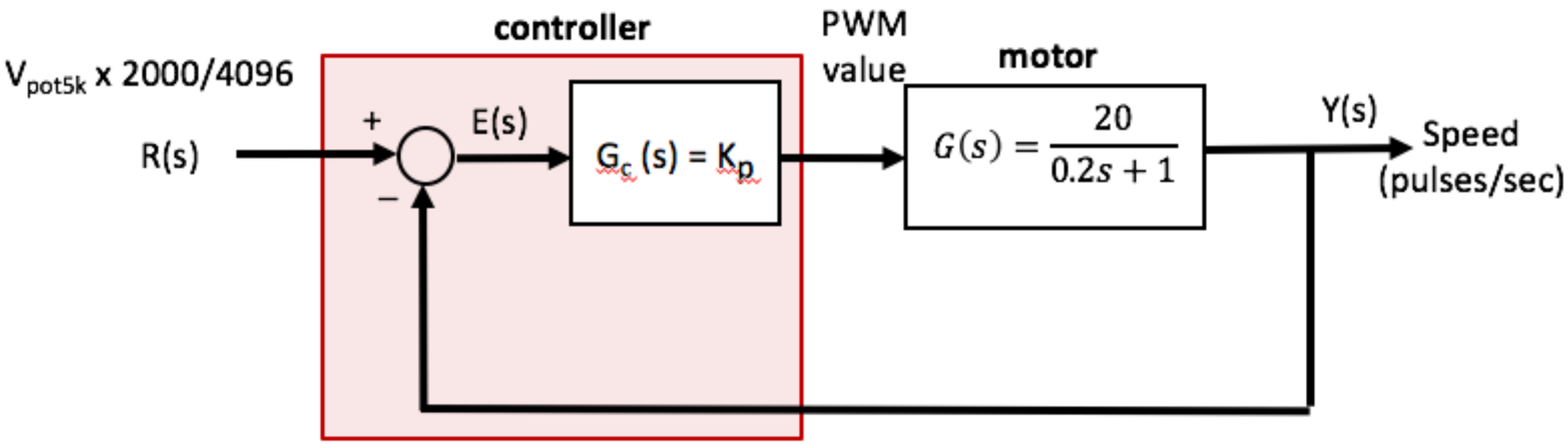
$$\left.\frac{Y(s)}{R(s)}\right|_{s=0} = \left.\frac{L(s)}{1+L(s)}\right|_{s=0} = \frac{20K_p}{1+20K_p} = \frac{100}{101} = 0.99$$

- The steady-state error for a step input of magnitude A (i.e. A * u(t) is:

$$\left.E(s)\right|_{s=0} = \left.\frac{1}{1+L(s)}\right|_{s=0} A = \frac{1}{1+L(0)}A = 0.01A$$

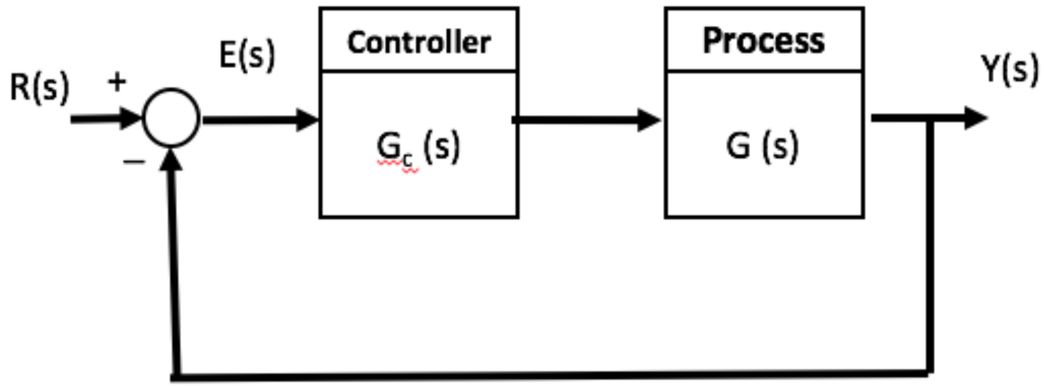
- Perturbation is also reduced by this factor (see slide 6):

$$Y(s) = 0.01P(s)$$



Three Big Ideas

1. Closed-loop negative feedback system has the general form (with example):



2. Adding the controller $G_C(s)$ and closing the loop changes the system transfer function from G(s) to:

$$\frac{\mathrm{Y(s)}}{\mathrm{R(s)}} = \frac{L(s)}{1 + L(s)}, \qquad \text{where } L(s) = G_c(s)G(s)$$

3. A closed-loop system reduces steady-state errors and impact of perturbation by a factor of (1 + L(s)), where L(s) is the loop gain.