Introduction to Automatic Control

2 - Feedback Control

University of BRISTOL

Contents

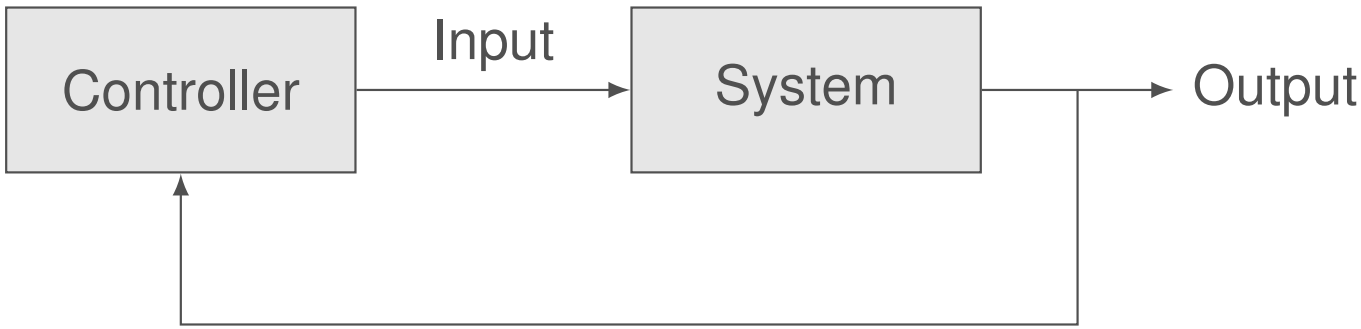
1. PID Control

2. Block Diagrams
   - Drawing Block Diagrams
   - Block Diagram Manipulation: Graphical Approach
   - Algebraic Approach

Feedback Control

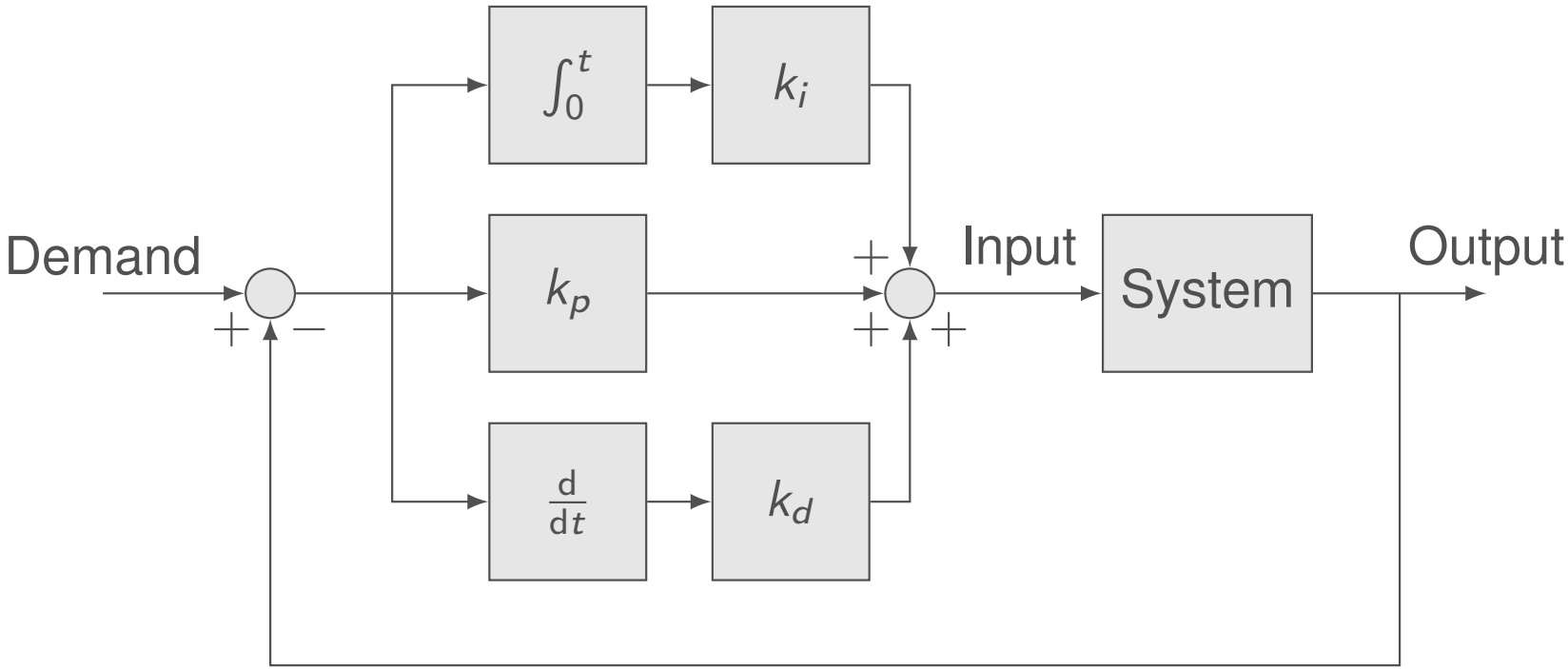
- **Open loop system:** input is *independent* of the output:

- **Closed Loop System:** Controller determines inputs based on outputs.



PID Control

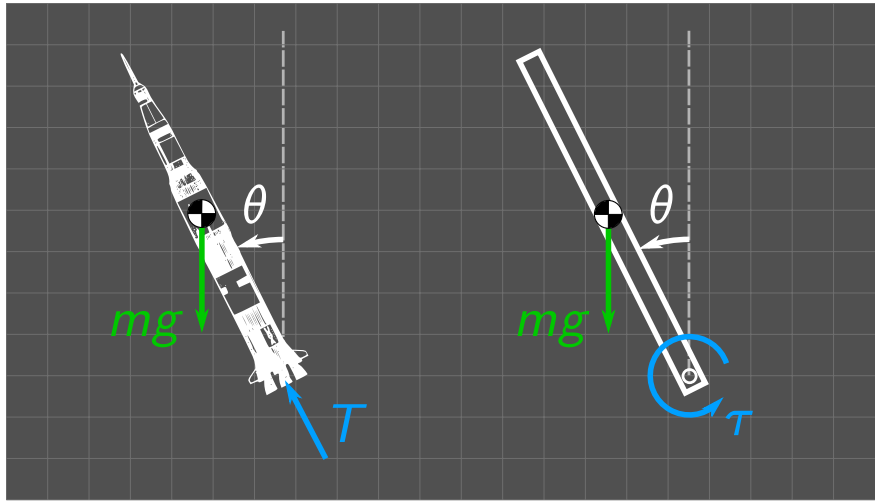
- **Proportional Integral Derivative (PID):** The most common control method *by far*.



- In this lecture we will develop some intuition for how PID control works.

Example System

- **Inverted Pendulum.**
- Dynamically *similar* to rocket orientation control problem.
  - Rocket: control direction and magnitude of *thrust* T.
  - Inverted pendulum: control *torque* τ .



- Objective to hold pendulum in vertical position.
- Inherently **unstable**.

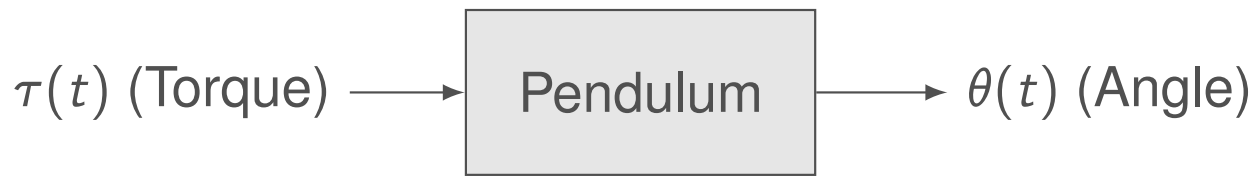
Open Loop System

- System Dynamics
  - Pendulum modelled as point mass
  - No damping (friction)

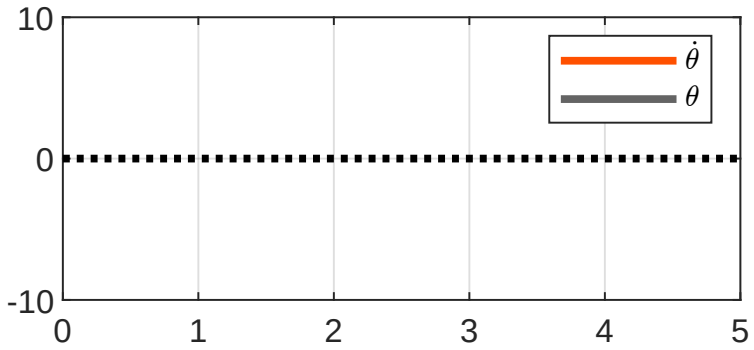
$$m\left(\frac{l}{2}\right)^2 \ddot{\theta} = \frac{mgl}{2}\sin\theta + \tau$$

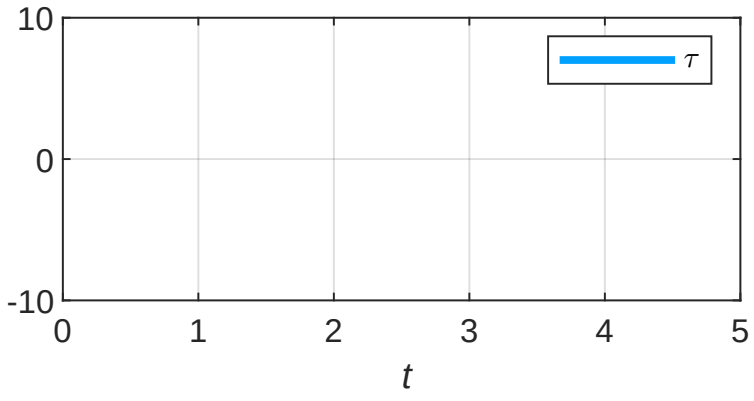
- **Nonlinear:** PID control *very* effective in practice for even nonlinear systems

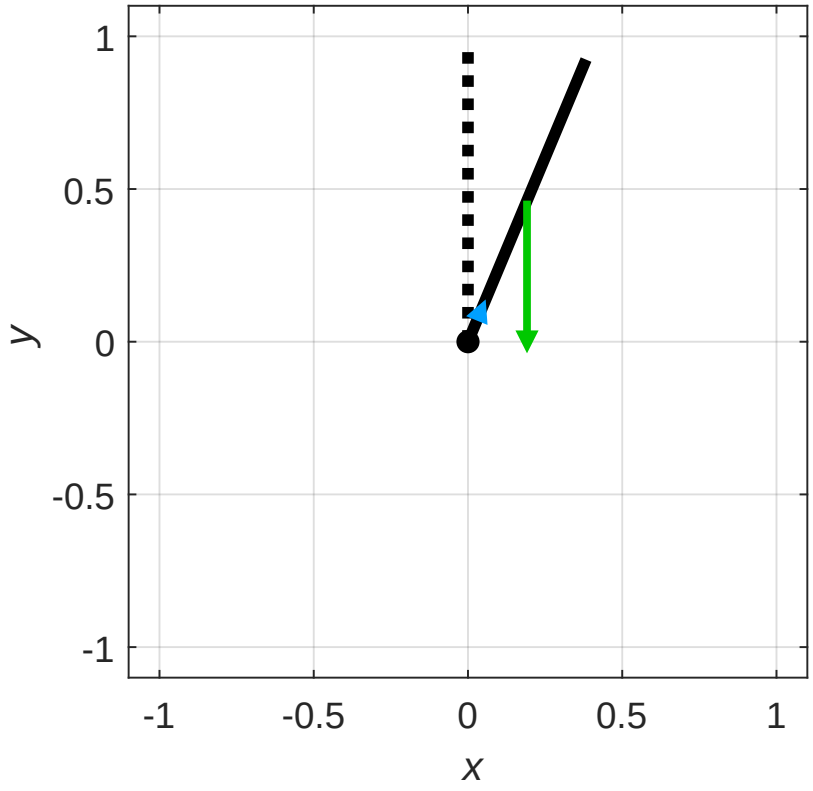
- Open loop 'Block diagram':

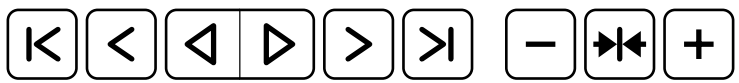


Open Loop Simulation



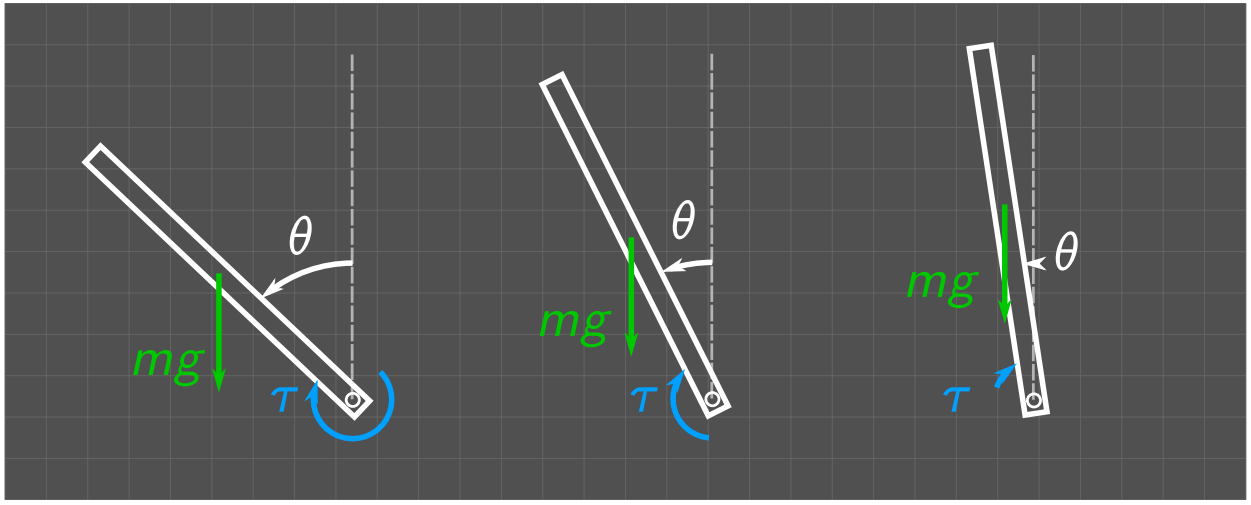






How Can We Invert the Pendulum?

- A possible solution is to apply a larger **restoring** torque as the pendulum deviates from $\theta = 0$



- Control law is

$$\tau(t) = -k_p \theta(t)$$

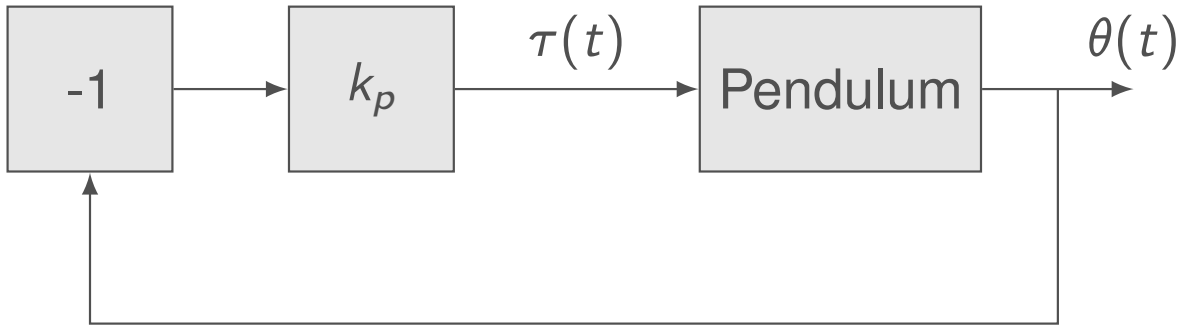
- This is known as **proportional control**
  - k_p is the **proportional gain**.

Proportional Control Block Diagram

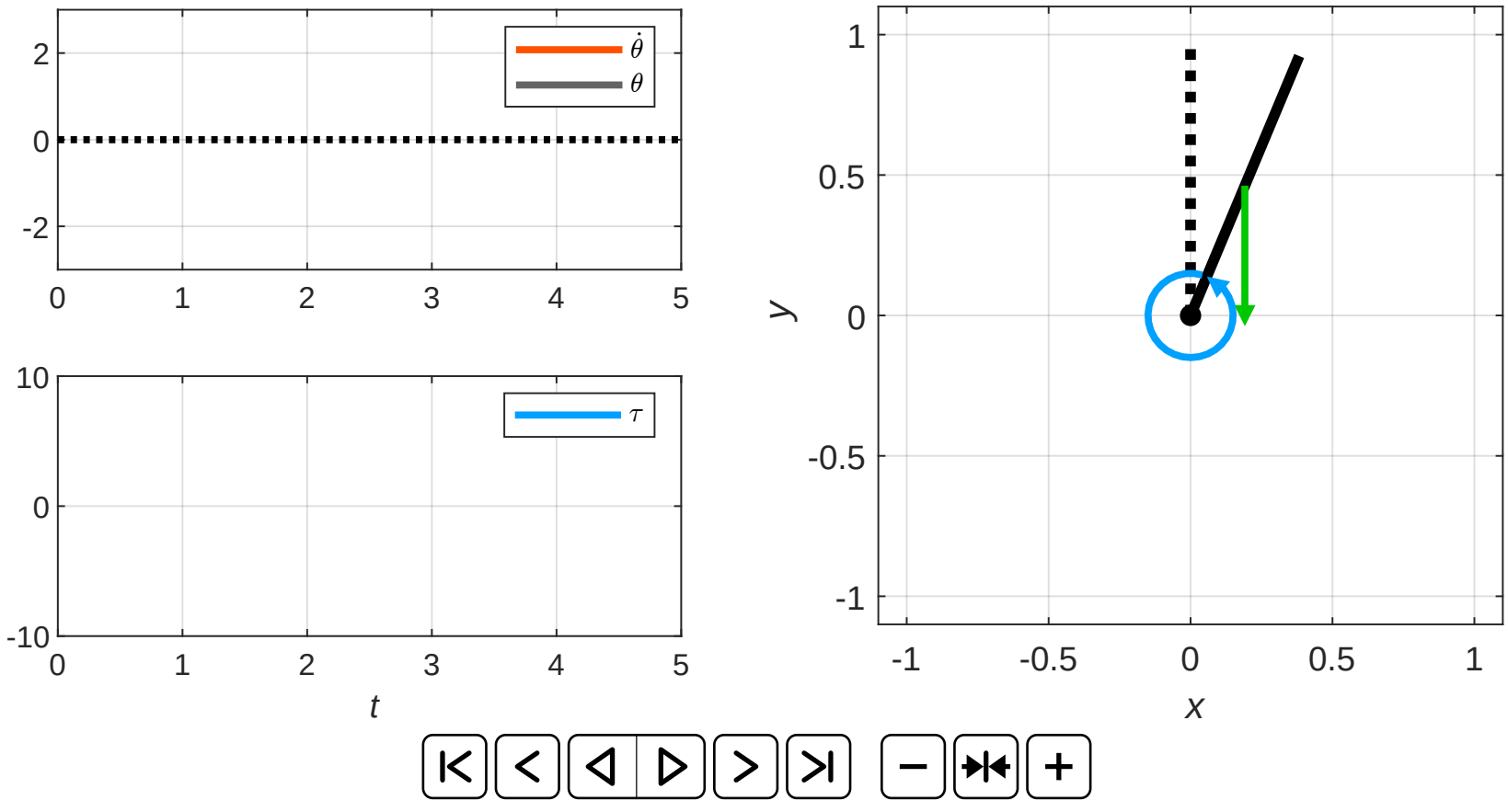
- Proportional controller:

$$\tau(t) = -k_p \theta(t)$$

- Block diagram:

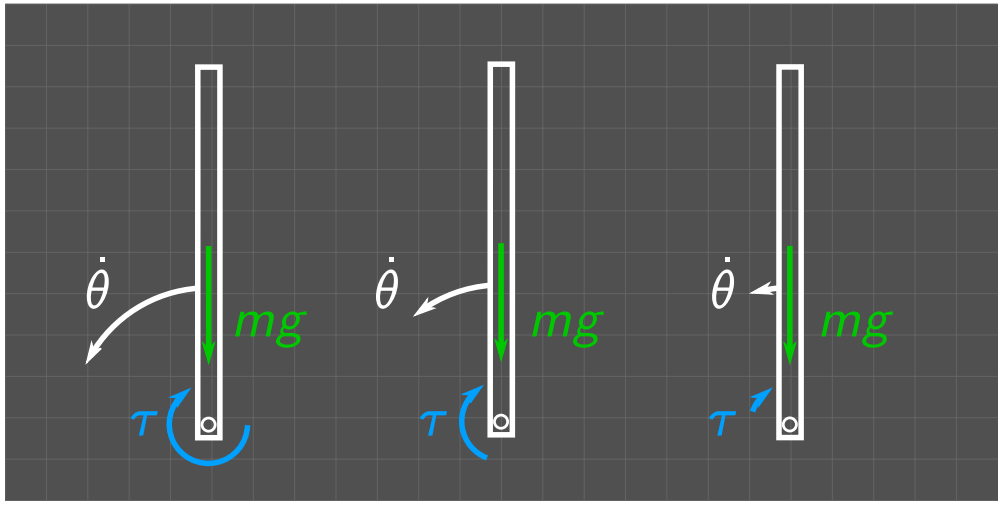


Proportional Simulation



How Can We Ensure $\theta(t) \to 0$?

- A possible solution is to use torque to counteract **velocity**.
  - 'Predicts' future errors.



- Control law is

$$\tau(t) = -k_p\theta(t) - k_d\dot{\theta}(t)$$

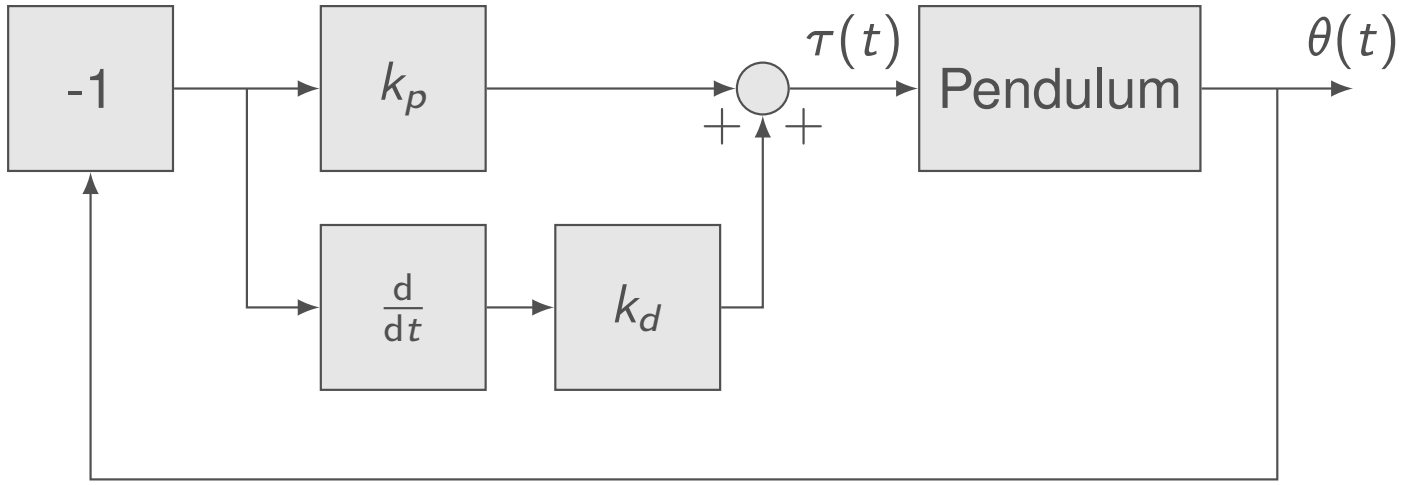
- This is known as **proportional derivative (PD) control**
  - k_d is the **derivative gain**.

PD Control Block Diagram

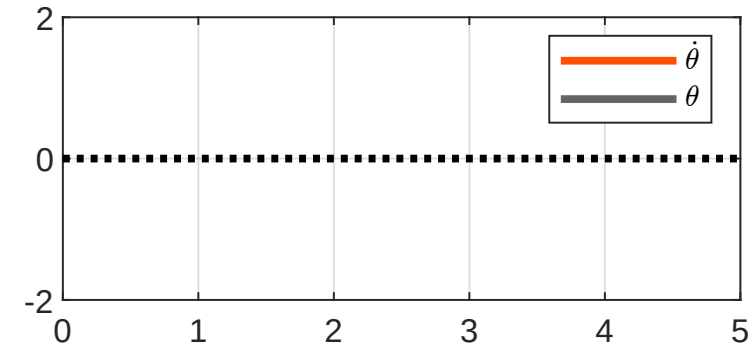
- Proportional derivative controller:

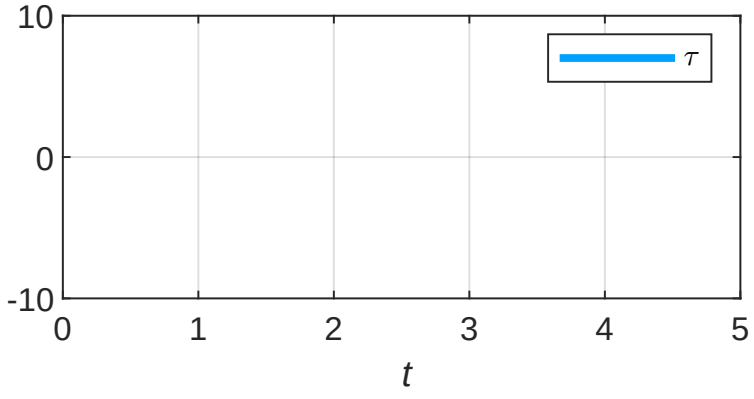
$$\tau(t) = -k_p\theta(t) - k_d\dot{\theta}(t)$$

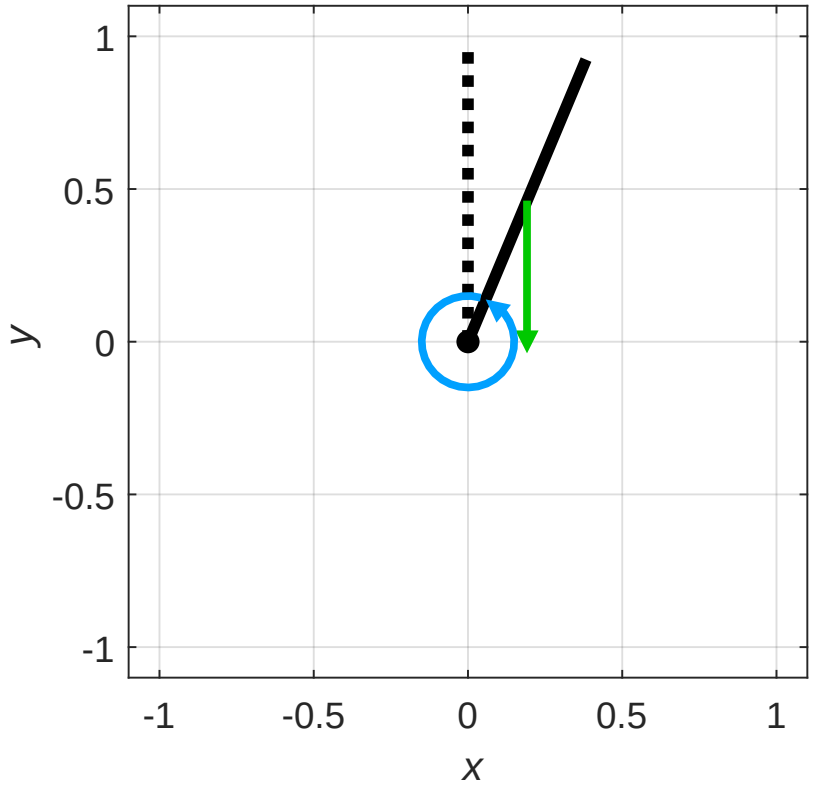
- Block diagram:

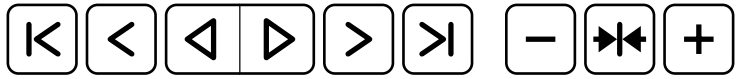


Derivative Simulation









Physical Interpretation

- Mass-spring-damper:

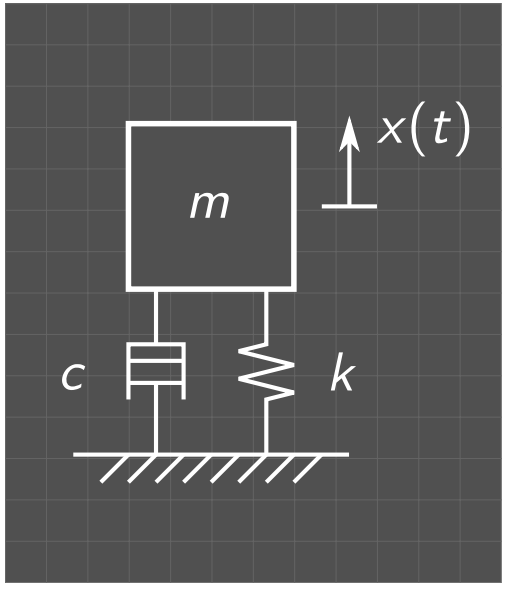
$$m\ddot{x}(t) = \sum_i f_i(t) = -kx(t) - c\dot{x}(t)$$

- PD control:

$$\text{input} = -k_p x(t) - k_d \dot{x}(t)$$

- Proportional gain acts as **stiffness** relative to desired state.

- Derivative gain acts as **damping**.

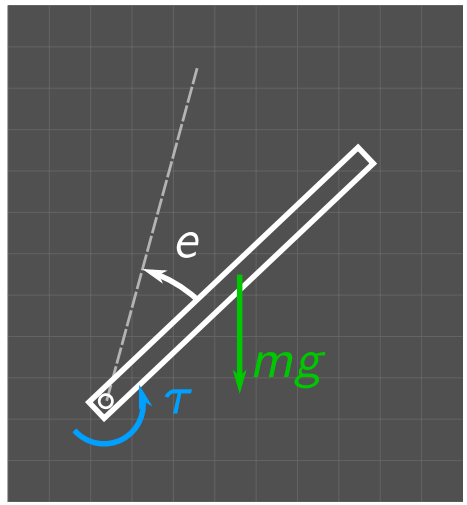


What if we Require Nonzero θ?

- A possible solution is to use **error** in our controller.

$$e(t) = r(t) - \theta(t)$$

- Now want to control so that $e(t) \to 0$.



- Control law is

$$\tau(t) = k_p e(t) + k_d \dot{e}(t)$$

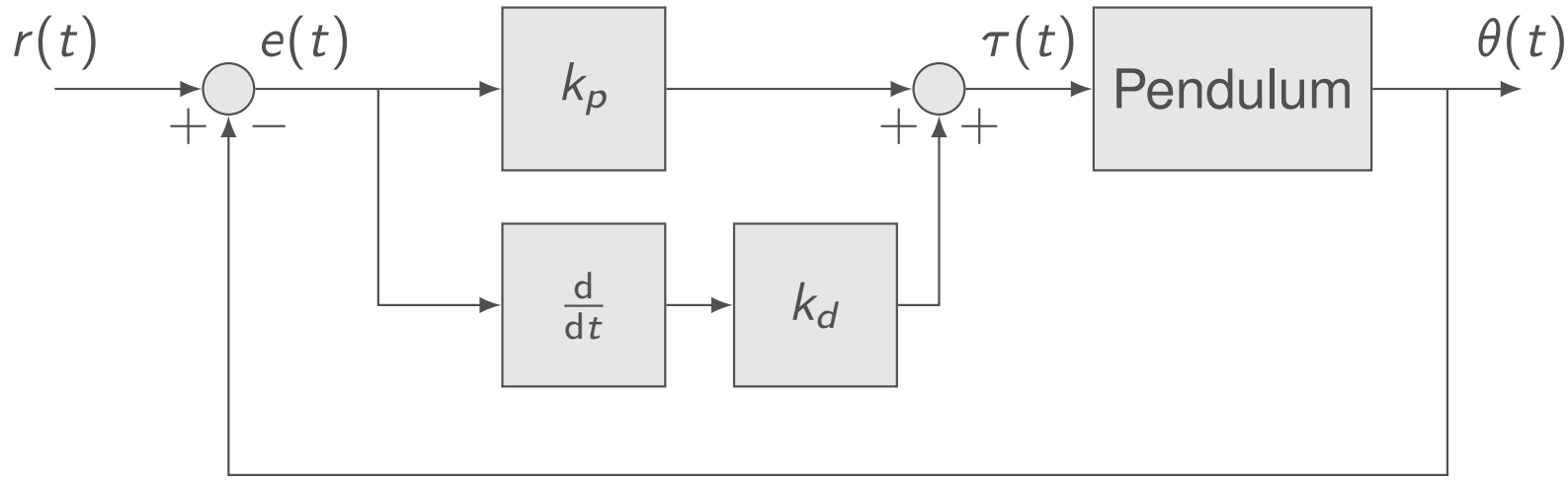
  - When $r(t) = 0$ this reverts to our previous controller.

Reference Block Diagram

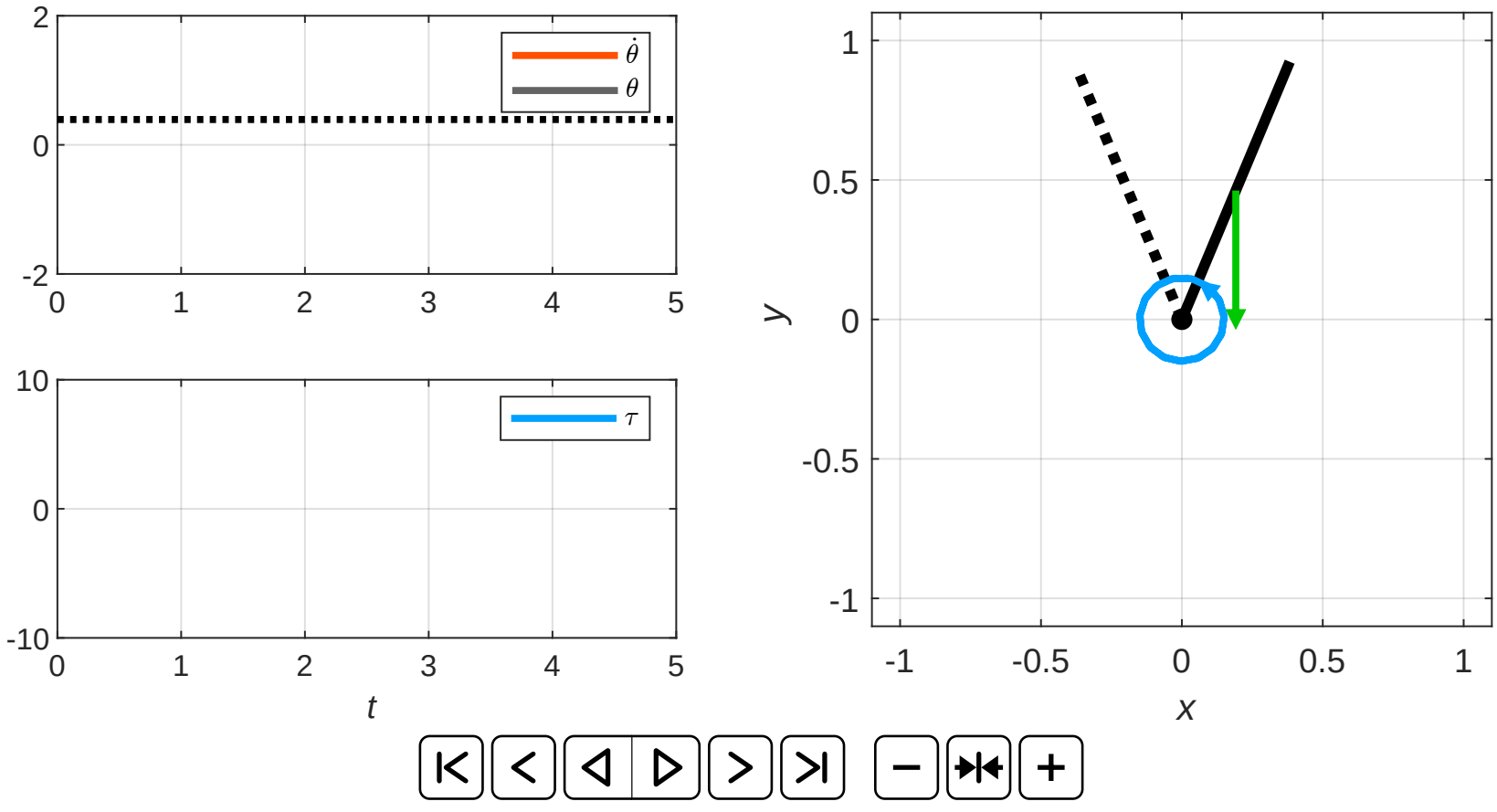
- Proportional derivative error controller:

$$e(t) = r(t) - \theta(t)$$
$$\tau(t) = k_p e(t) + k_d \dot{e}(t)$$

- Block diagram:

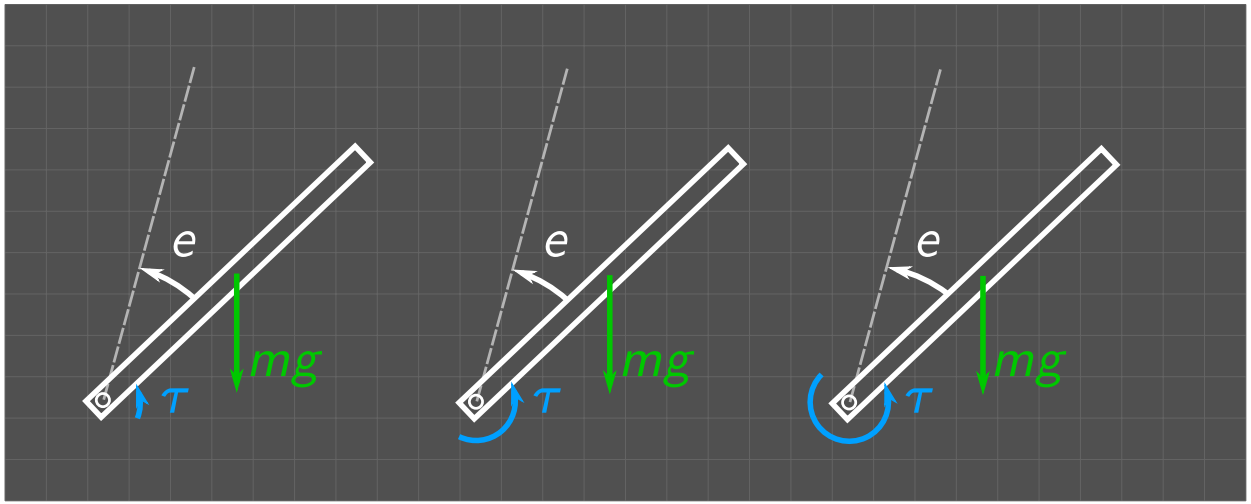


Reference Simulation



How Can We Ensure $e(t) \to 0$?

- Possible solutions:
  - Increase the proportional gain.
  - Apply a larger restoring torque as the **integral** of the error increases.



- Control law is

$$\tau(t) = k_p e(t) + k_d \dot{e}(t) + k_i \int_0^t e(\hat{t})\, \mathrm{d}\hat{t}$$

- This is known as **proportional integral derivative control** (PID)
  - k_i is the **integral gain**.

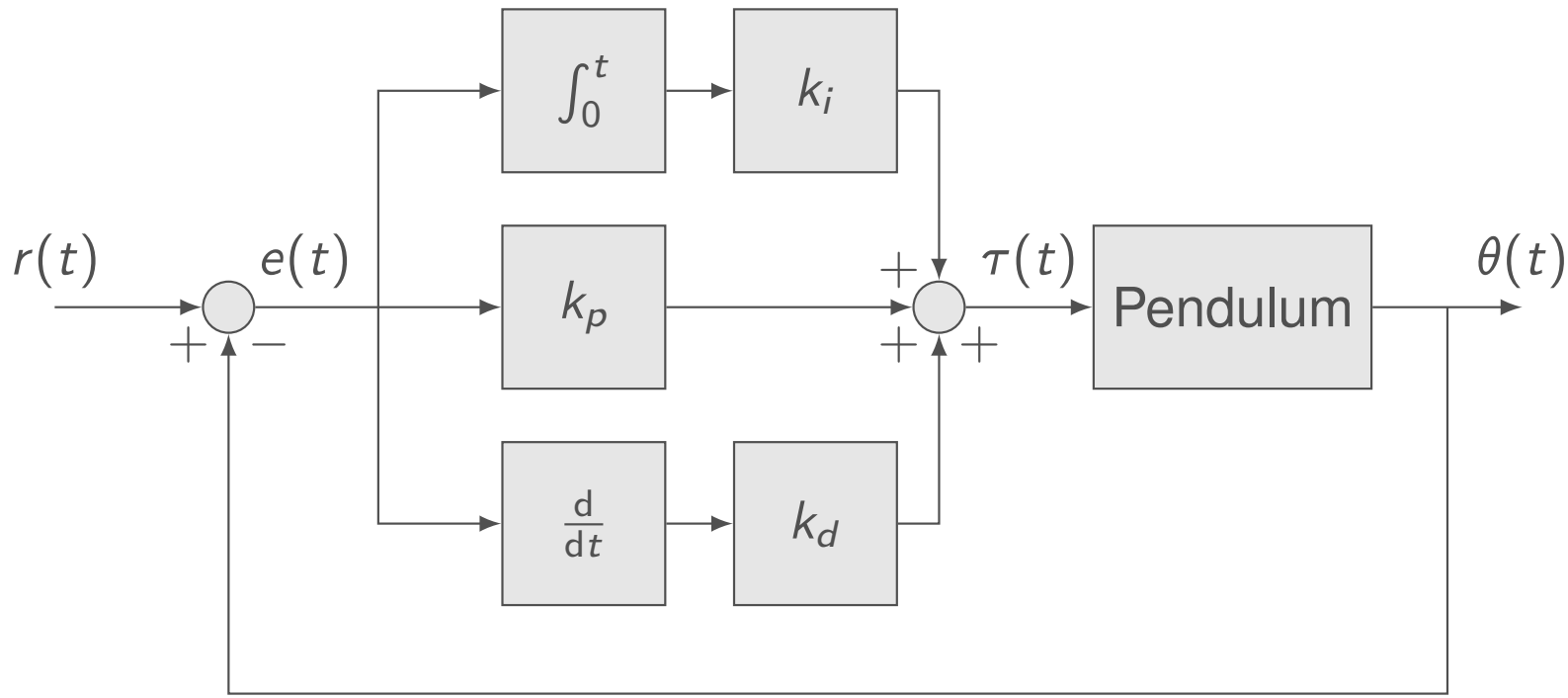
PID Block Diagram

- Proportional integral derivative (PID) controller:

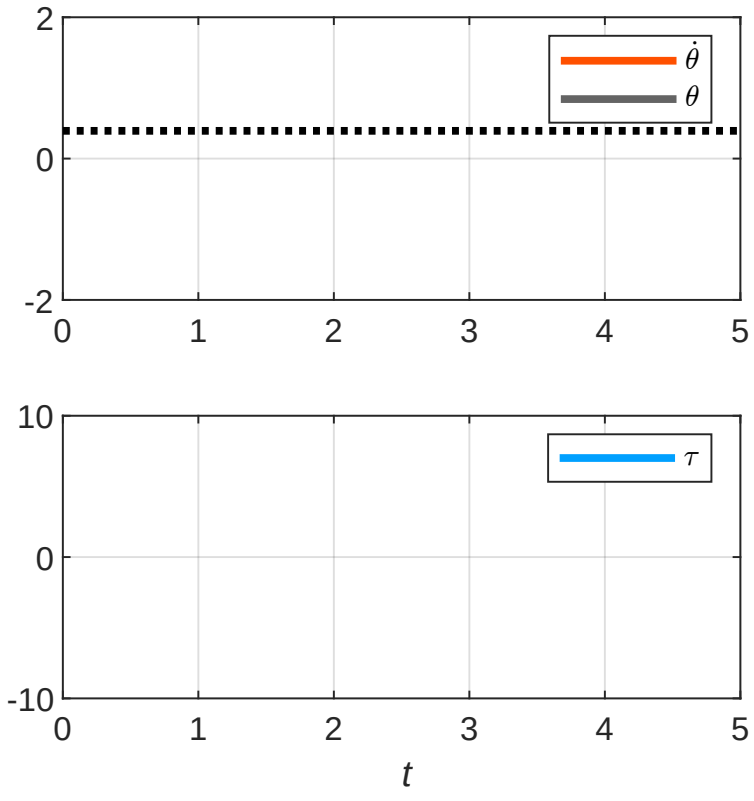
$$e(t) = r(t) - \theta(t)$$

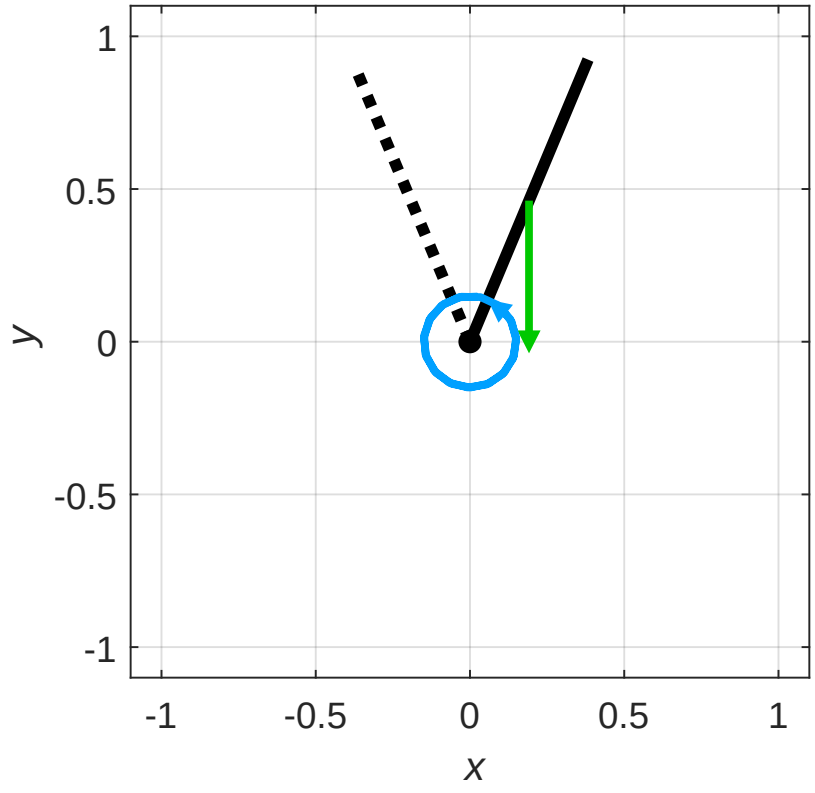
$$\tau(t) = k_p e(t) + k_d \dot{e}(t) + k_i \int_0^t e(\hat{t})\, \mathrm{d}\hat{t}$$

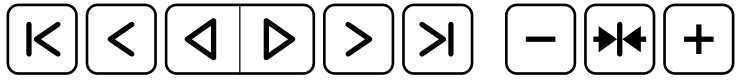
- Block diagram:



Integral Simulation







That's it!

+ Simple concepts
  - Control based on **error** between current and desired states.
  - **Proportional** gain provides 'stiffness'.
  - **Derivative** gain provides 'damping'.
  - **Integral** gain provides disturbance rejection.

+ Can be effectively tuned using trial and error.

+ Can be used to effectively control complex, nonlinear systems.

"PID works embarrassingly well."

Prof. Arthur Richards, 2021

Manual Tuning

- The following tuning approach is an extremely effective ‘rule of thumb’:
  1. Set $k_d = k_i = 0$
  2. Increase k_p until the **step response**[1] reaches the limit of **stability**
  3. Reduce k_p to achieve **quarter amplitude decay**.
  4. Tune k_i and k_d to achieve satisfactory step response.

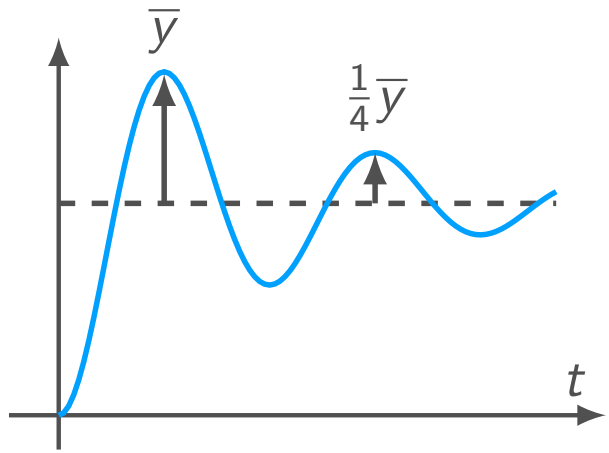


Figure: Quarter Amplitude Decay

[1] The system’s response to the demand function $r(t) = 1$ for $t \geq 0$, $r(t) = 0$ for $t < 0$.

Ziegler-Nichols Rules

- The **Ziegler-Nichols** tuning method is a set of **rules** for determining the controller gains.
  - Developed in 1940's for engineering practitioners when control system design wasn't widely understood[2].

- Define
  - k_u as the gain k_p at the point of instability.
  - p_u as the the period of oscillations at the point of instability.

- The rules are then[3]:

Controller type	k_p	k_i	k_d
P	$0.5k_u$		
PI	$0.45k_u$	$\frac{0.54k_u}{p_u}$	
PID	$0.6k_u$	$\frac{1.2k_u}{p_u}$	$\frac{0.6k_u p_u}{8}$

[2] **Ziegler1942OptimumSF**.

[3] Note that these rules don't always work.

Why Not Stop There?

- Sometimes heuristic/rule-based tuning doesn't work
  - Need to understand *how* to achieve certain behaviours.
- Need to understand limitations of automatic control
  - Can't reject both sensor noise and disturbances simultaneously.
- Heuristic tuning doesn't cut it in safety critical applications.
  - Need to understand *safety margins*.
- And more...
- **From here on out:** we will learn mathematical approaches to control system analysis and design.

You can now complete problem 1

Contents

1. PID Control

2. Block Diagrams
   - Drawing Block Diagrams
   - Block Diagram Manipulation: Graphical Approach
   - Algebraic Approach

Motivation

- We have previously discussed how the output of a single LTI system is given by

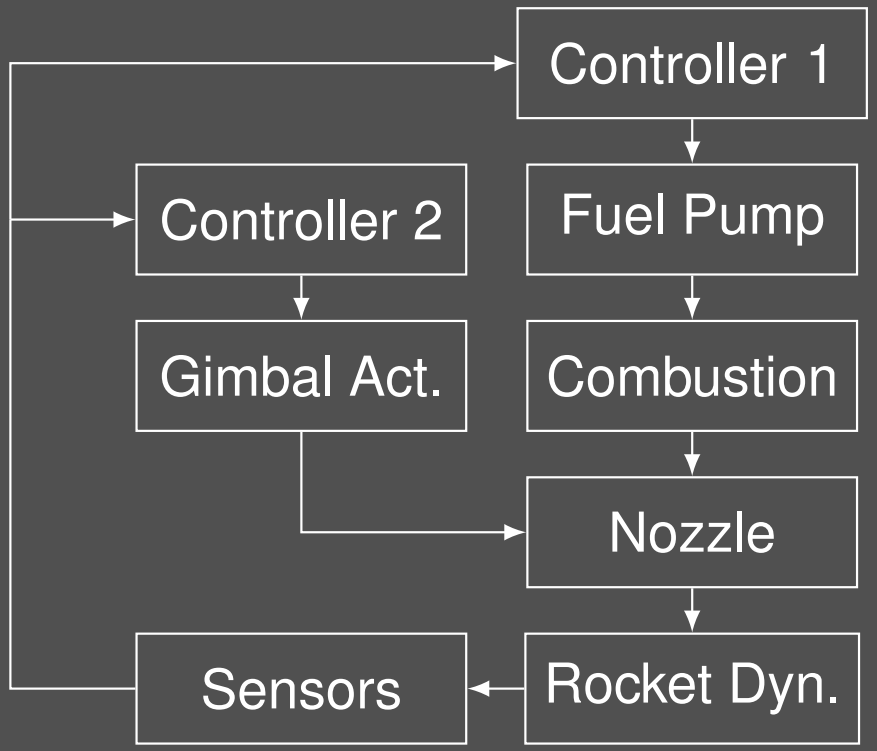
$$x(t) = \int_{-\infty}^{\infty} g(\tau)u(t-\tau)\mathrm{d}\tau,$$

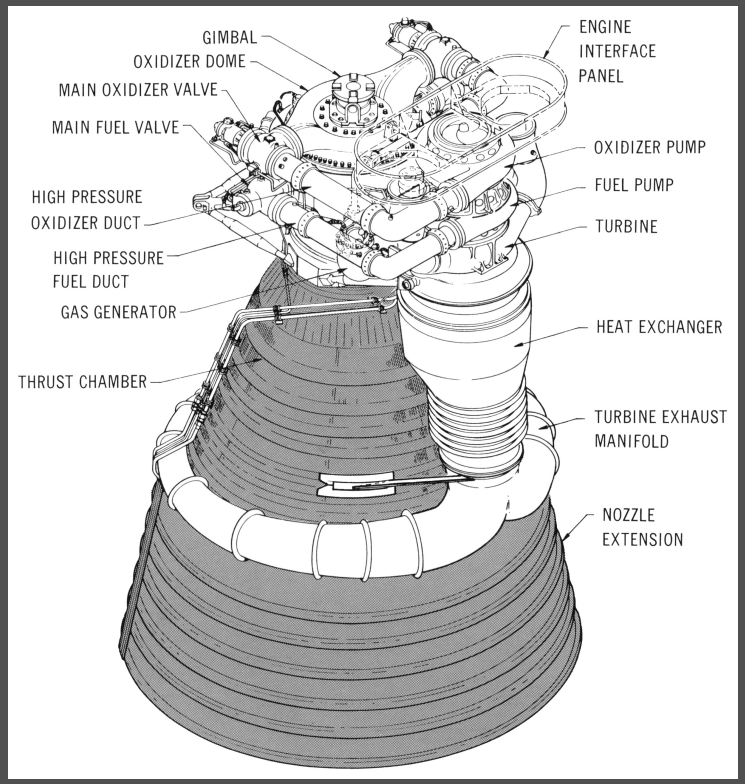
- We are often interested in modelling **multiple** systems that **interact** with each other.
  - Time domain solution not so simple in this case.

- **Block Diagrams** are a useful tool for visually representing these interactions.

- In the s-domain, **simple algebraic rules** can be used to determine the dynamics of arbitrary input-output pairs.
  - In this lecture we will learn how to manipulate and reduce block diagrams.
  - This will help us analyze the behaviour of complex, **closed-loop** systems.

Rocket Control





- How does sensor noise impulse affect fuel pump speed?
- How does a step change in rocket mass affect combustion process?
- ...

Contents

1. PID Control

2. Block Diagrams
   - Drawing Block Diagrams
   - Block Diagram Manipulation: Graphical Approach
   - Algebraic Approach

Transfer Functions

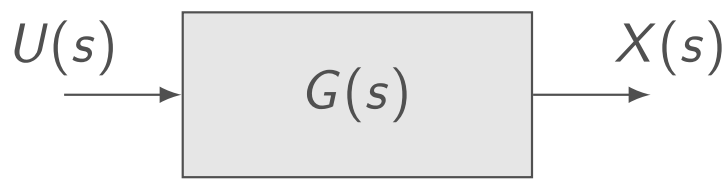
- We will always consider block diagrams (from here) in the s-domain.
  - Simple algebraic rules apply.

- We have previously discussed how the output of a **LTI system** is given in the s-domain by

$$X(s) = G(s)U(s)$$

  where $G(s)$ is the **Transfer Function**.

- This can be represented by the (s-domain) **block diagram**

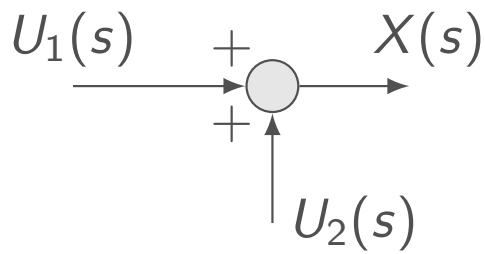


Summation of Signals

- **Addition:** Laplace transform is a **linear operator**, so summation of signals in the time domain is also summation in the s-domain:

$$x(t) = u_1(t) + u_2(t) \quad \Longleftrightarrow \quad X(s) = U_1(s) + U_2(s)$$

- Graphical representation:



System Composition

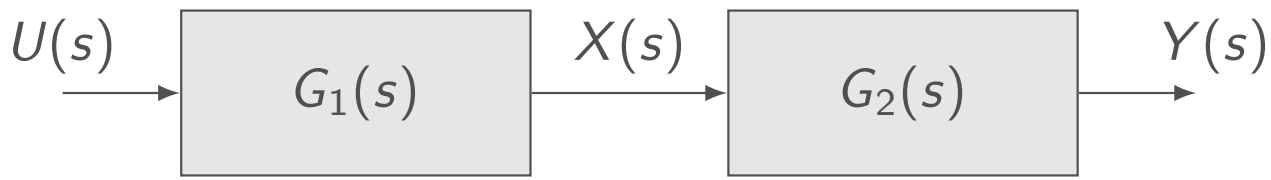
- **Composition:** Similarly for composition of systems (output of one system is input to another):

$$x(t) = \int_{-\infty}^{\infty} g_1(\tau)u(t-\tau)\mathrm{d}\tau, \quad y(t) = \int_{-\infty}^{\infty} g_2(\tau)x(t-\tau)\mathrm{d}\tau$$

$$\Longleftrightarrow \quad X(s) = G_1(s)U(s), \quad Y(s) = G_2(s)X(s)$$

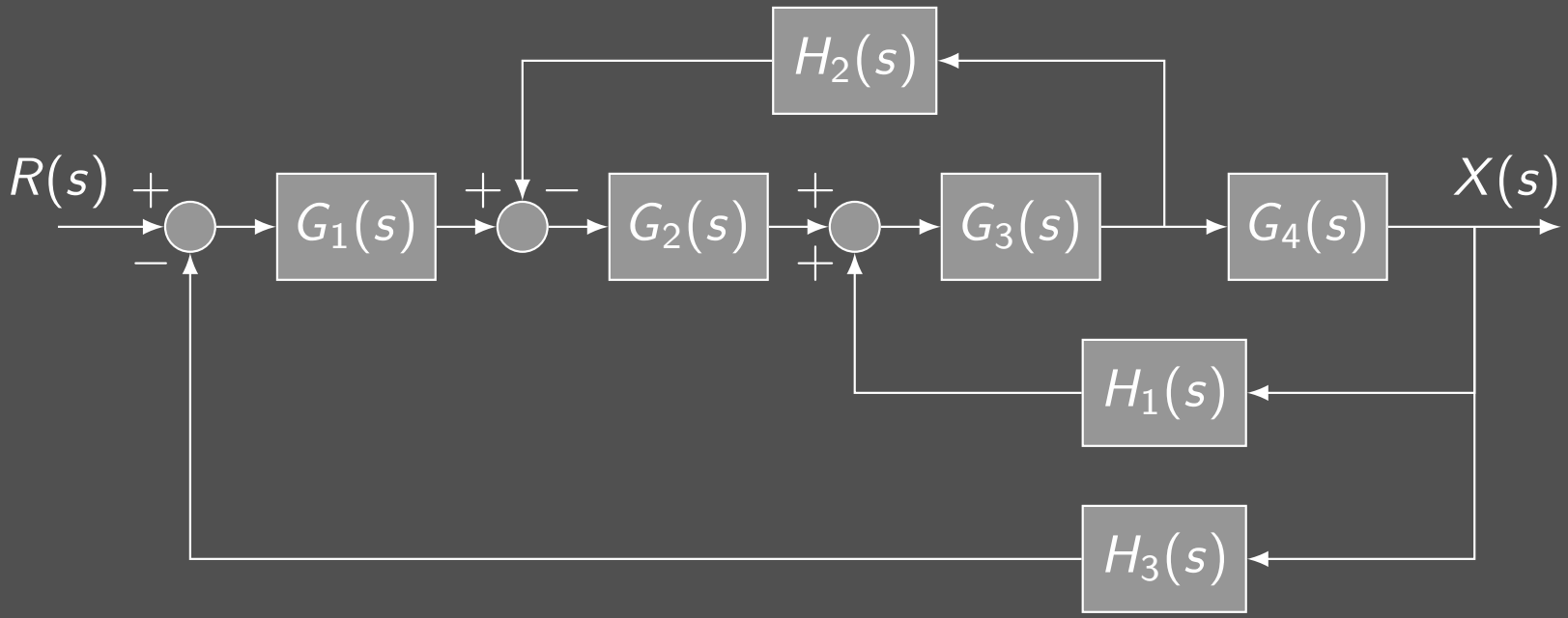
$$\Longleftrightarrow \quad Y(s) = G_1(s)G_2(s)U(s)$$

- Graphical representation:



Example

- These simple rules can be used to generate a graphical representation of many interacting systems:



- How can we determine the transfer function $\frac{X(s)}{R(s)}$?

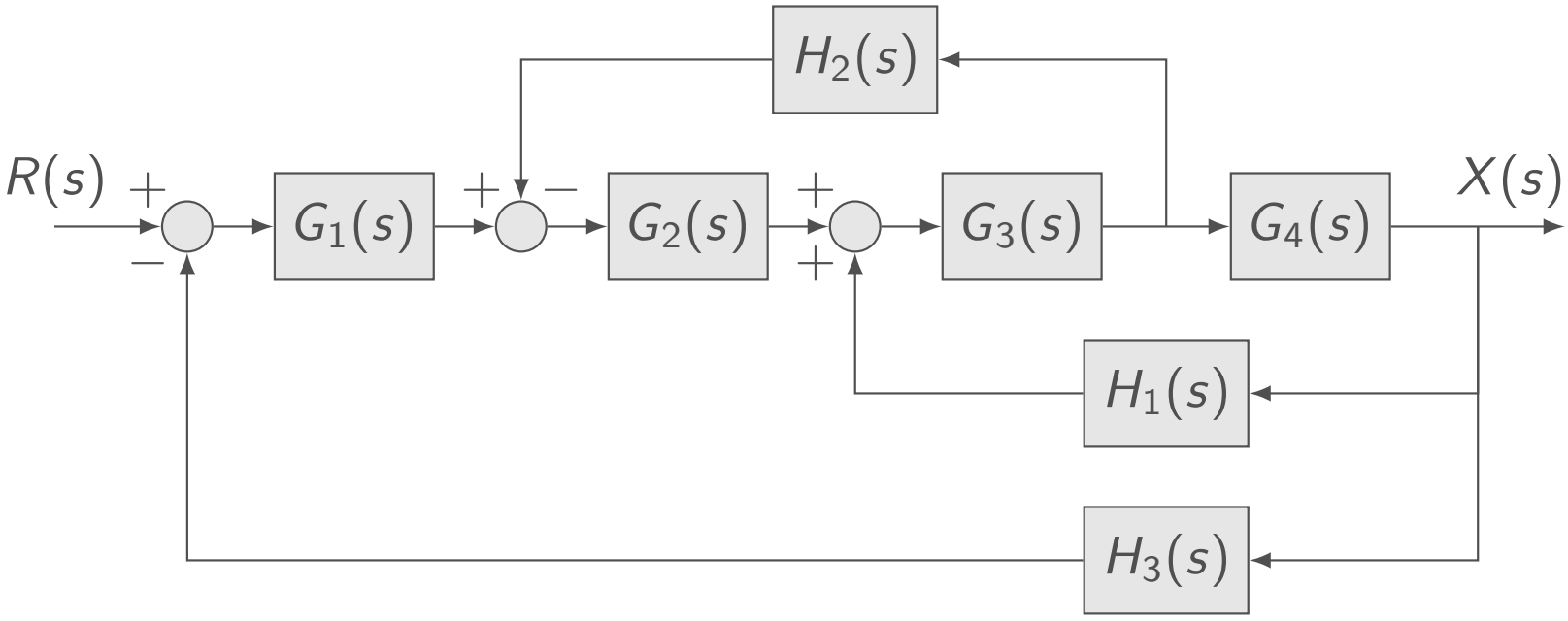
Contents

1. PID Control

2. Block Diagrams
   - Drawing Block Diagrams
   - Block Diagram Manipulation: Graphical Approach
   - Algebraic Approach

Condensing Block Diagrams

- Generally interested in how the system behaves as a whole:



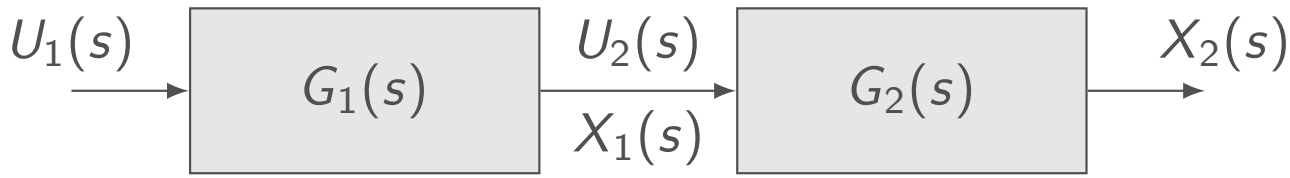
- How do we the transfer function $\frac{X(s)}{R(s)}$?
  - Graphical (rule-based) method
  - Algebraic method

Rule 1: System Composition

- Consider two systems

$$X_1(s) = G_1(s)U_1(s) \quad X_2(s) = G_2(s)U_2(s)$$

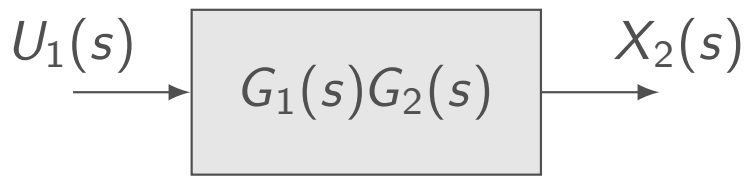
- The case where the input to system 2 is the output of system 1 (i.e. $U_2 = X_1$) can be represented by the block diagram



- In this case output 2 is given w.r.t input 1 by

$$X_2(s) = G_1(s)G_2(s)U_1(s)$$

which can be represented by the simplified block diagram

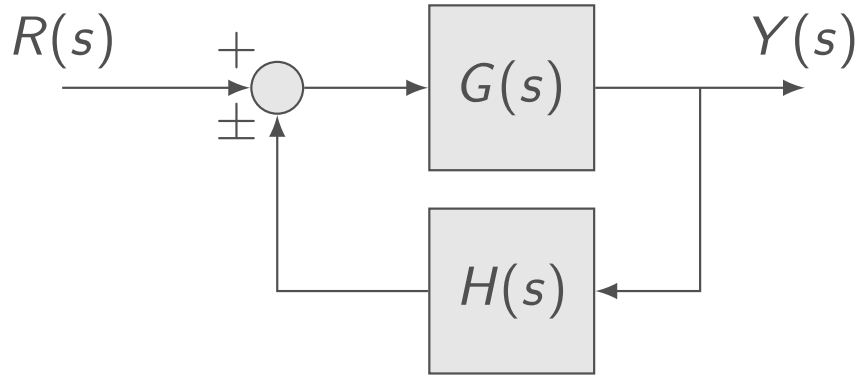


Rule 2: Feedback

- Consider the systems

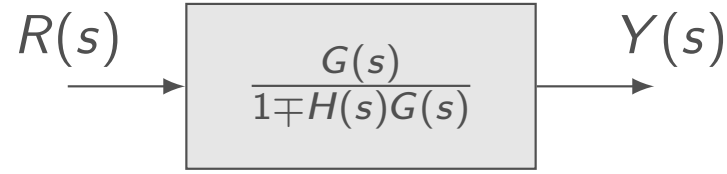
$$Y(s) = G(s)U(s) \quad X(s) = H(s)Y(s)$$

- The block diagram for **feedback** $U(s) = R(s) \pm X(s)$ is:



- The **closed-loop** system dynamics are given by[4]

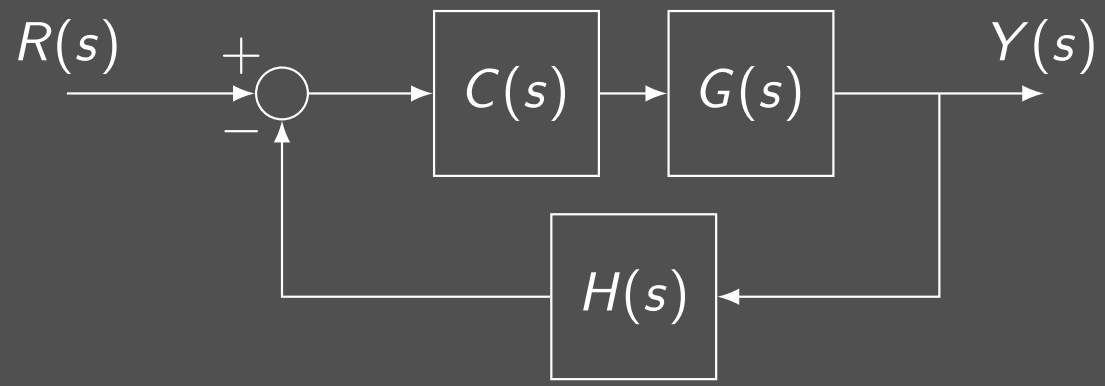
$$Y(s) = \frac{G(s)}{1 \mp H(s)G(s)} R(s)$$



[4] can you show this?

Cascade Controller

- The **closed-loop** transfer function for the cascade controlled system:

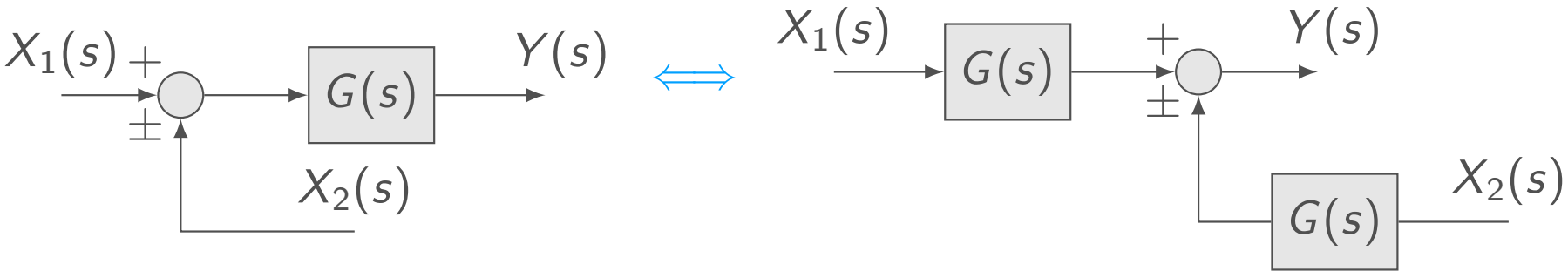


is therefore

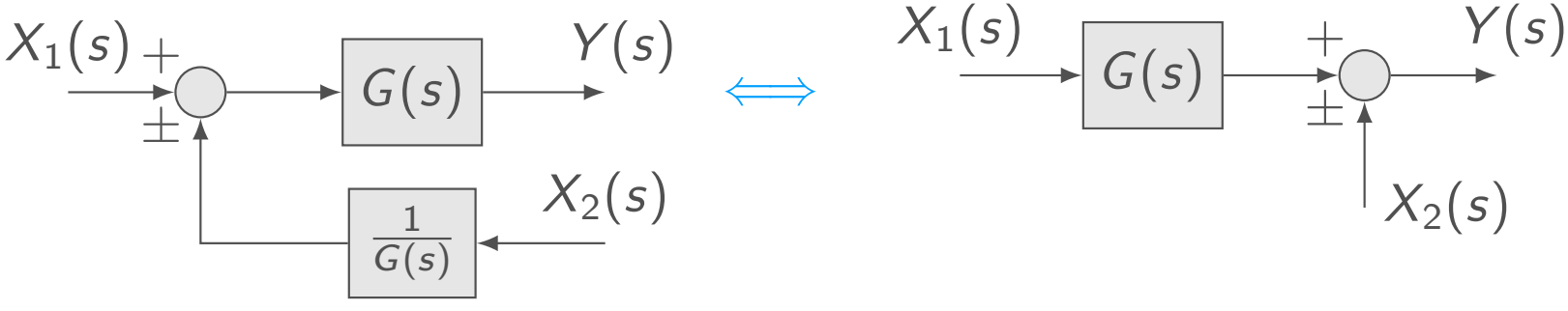
$$\frac{Y(s)}{R(s)} = \frac{C(s)G(s)}{1 + H(s)C(s)G(s)}$$

More Manipulation Rules 1

- **Rule 3:**

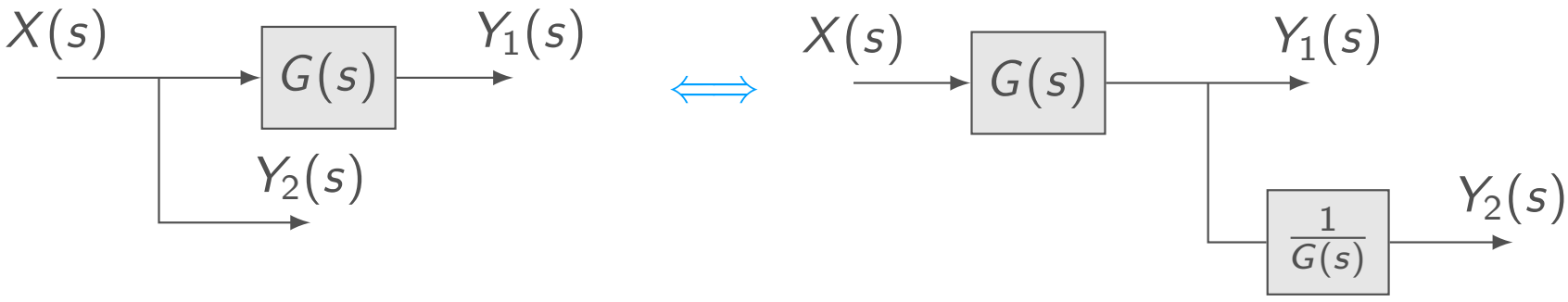


- **Rule 4:**

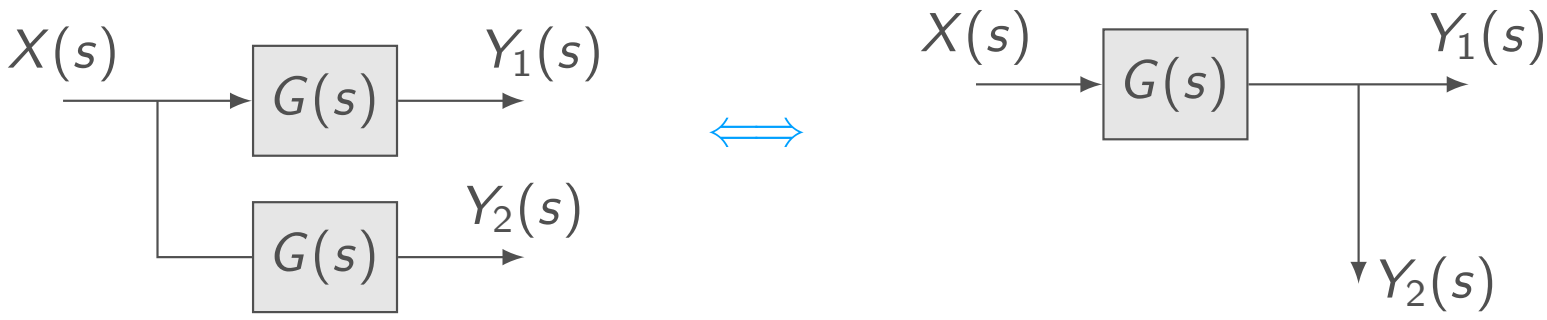


More Manipulation Rules 2

- **Rule 5:**



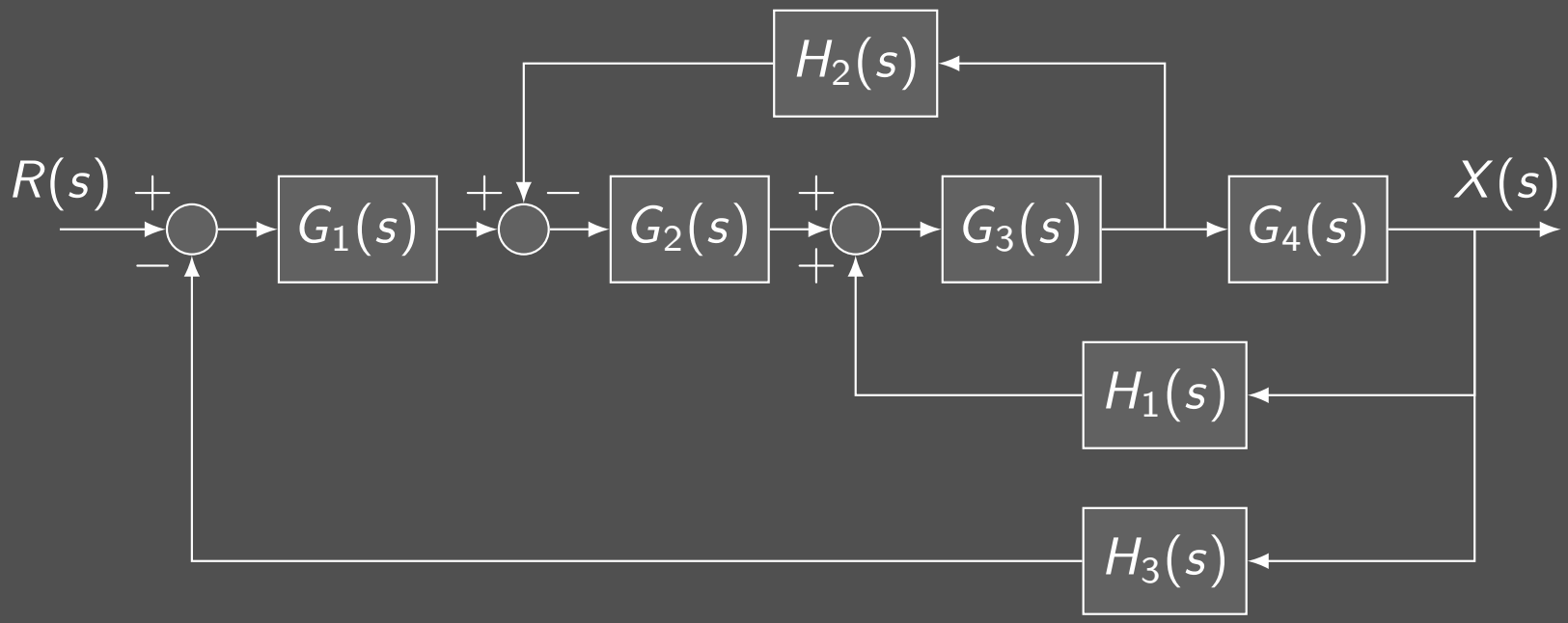
- **Rule 6:**



Example

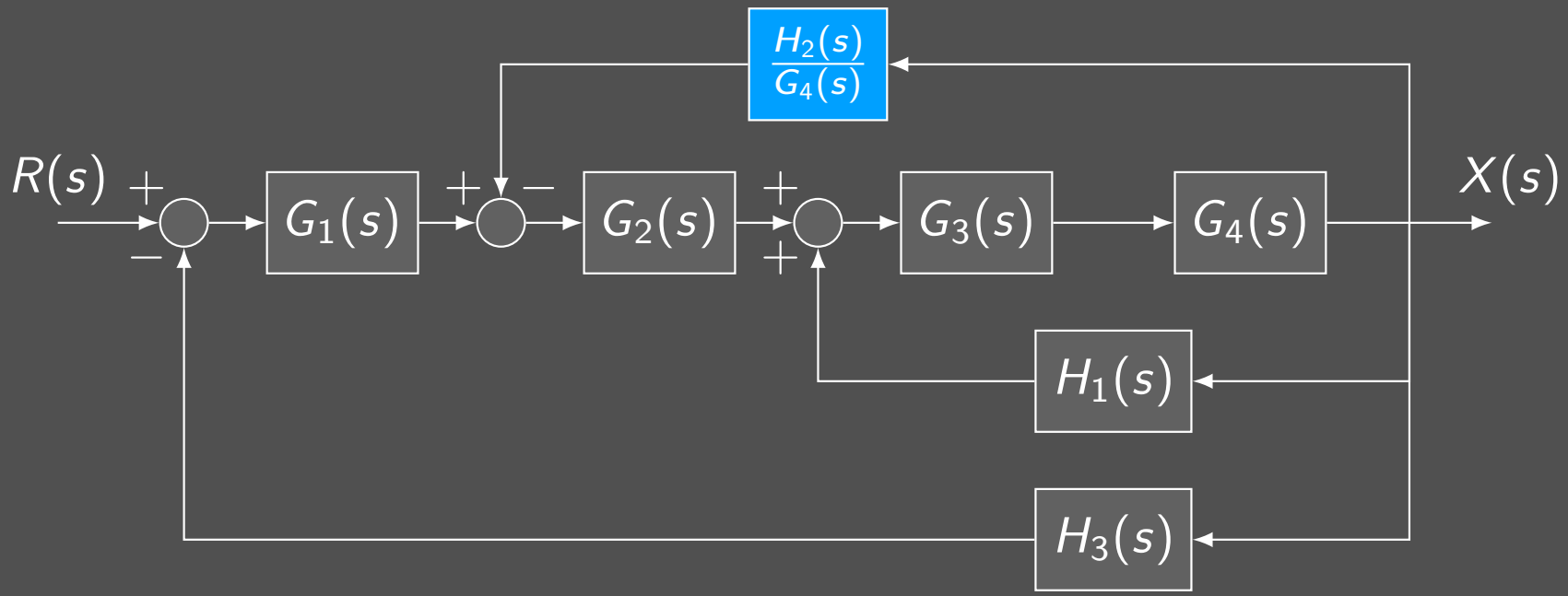
- These rules can be used to condense down more complex block diagrams to determine the transfer functions between arbitrary inputs and outputs.

Example: Determine the transfer function $\frac{X(s)}{R(s)}$.



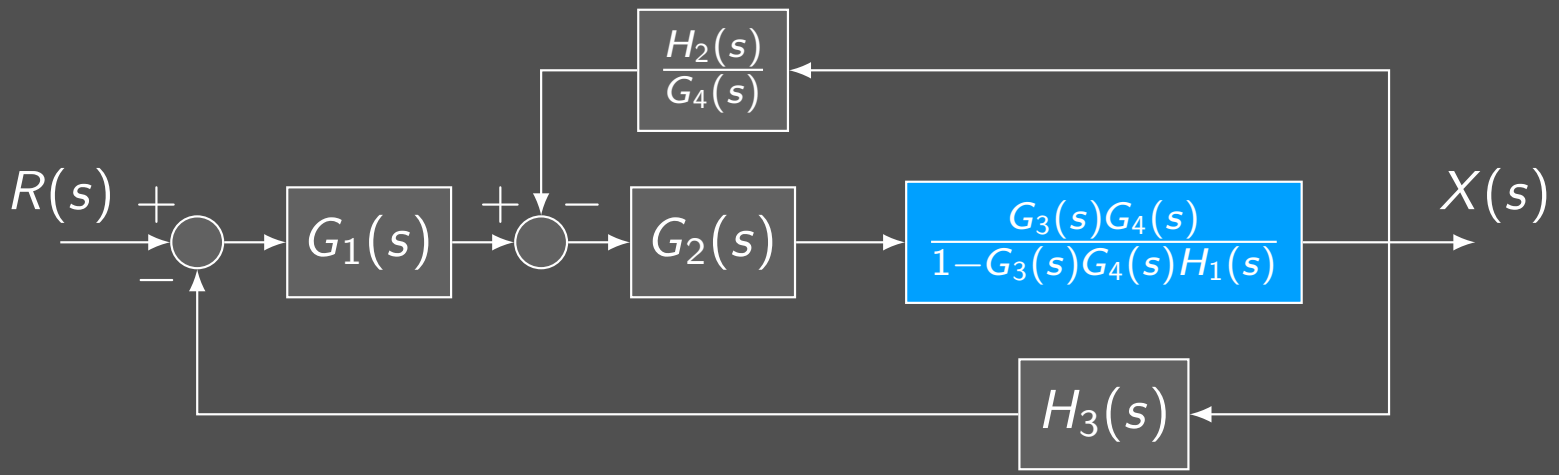
Step 1

Rule 4 to move input of $H_2(s)$ to $X(s)$:



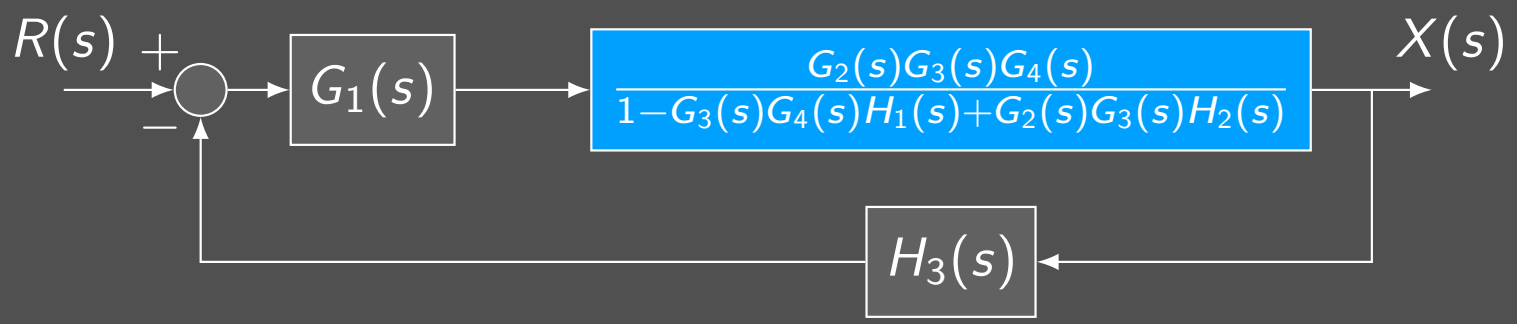
Step 2

Rules 1 and 2 to move remove $H_1(s)$ feedback loop:

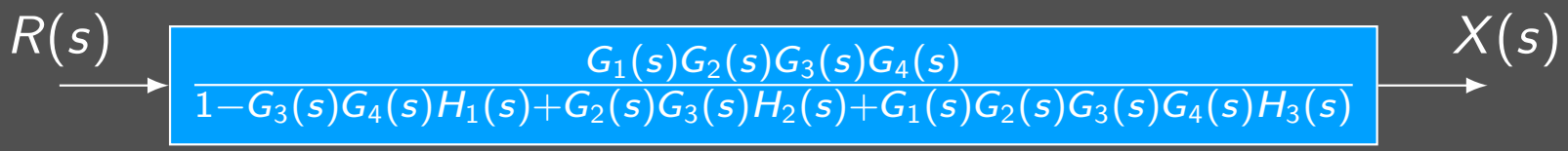


Steps 3 & 4

Rules 1 and 2 to move remove $\frac{H_2(s)}{G_4(s)}$ feedback loop:



Rules 1 and 2 to remove $H_3(s)$ feedback loop:



Contents

1. PID Control

2. Block Diagrams
   - Drawing Block Diagrams
   - Block Diagram Manipulation: Graphical Approach
   - Algebraic Approach

Algebraic Method

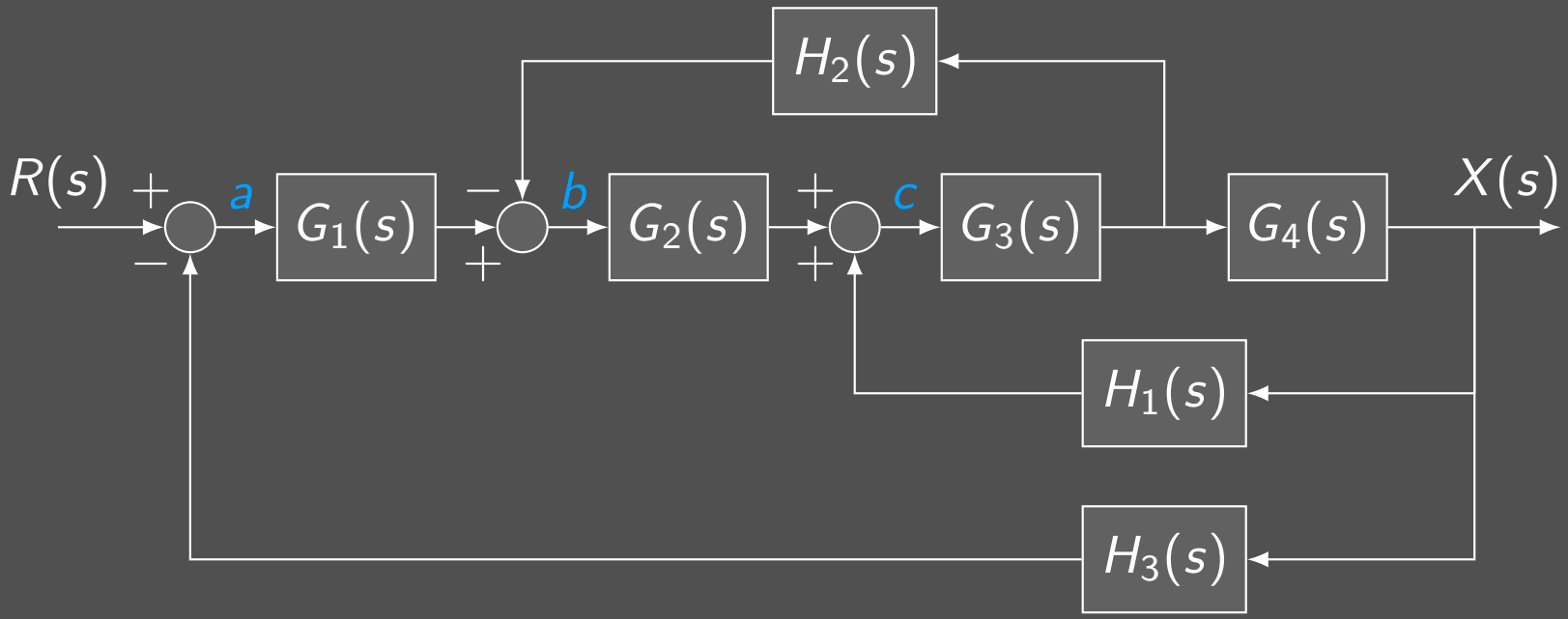
- Block diagrams can alternatively be condensed **algebraically**.

General Method:

1. Create a **dummy variable** after each summing junction.
2. Work **from output to input** to determine an expression for each variable in terms of the other variables.
3. Work **bottom-to-top** to combine your expressions together.
4. Rearrange terms.

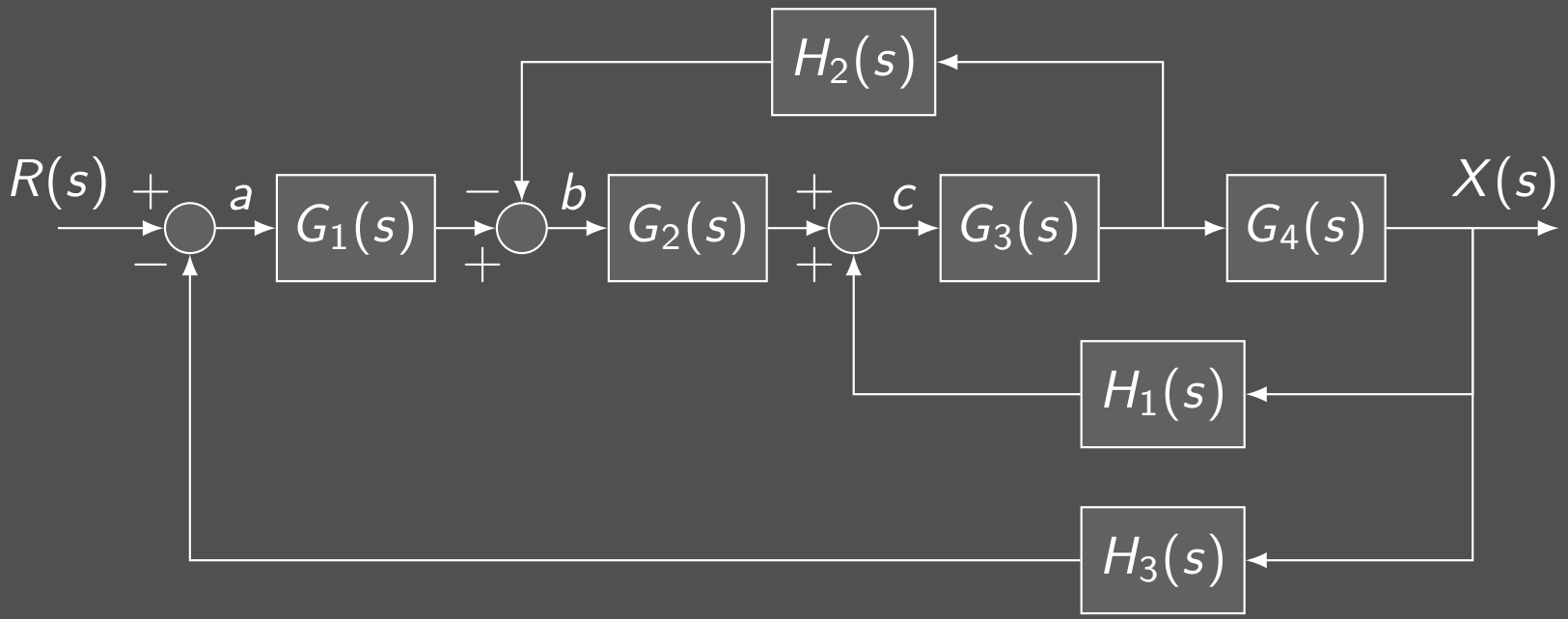
Step 1

Dummy Variables:



Step 2

Determine expressions (output-to-input):



(1) $X = G_3 G_4 c$

(2) $c = G_2 b + H_1 X$

(3) $b = G_1 a - H_2 G_3 c$

(4) $a = R - H_3 X$

Step 3

(1) $X = G_3 G_4 c$
(2) $c = G_2 b + H_1 X$
(3) $b = G_1 a - H_2 G_3 c$
(4) $a = R - H_3 X$

Combine expressions (bottom-to-top):[5]

- (4) into (3) gives

$$b = G_1 R - G_1 H_3 X - H_2 G_3 c \tag{5}$$

- (5) into (2) gives

$$c = G_2 G_1 R - G_2 G_1 H_3 X - G_2 H_2 G_3 c + H_1 X \tag{6}$$

- Rearrange for c, then (6) into (1) gives

$$X = G_3 G_4 \frac{G_2 G_1 R + (H_1 - G_2 G_1 H_3) X}{1 + G_2 H_2 G_3}$$

[5] Doesn't always work quite this 'nicely'; sometimes you need to work out how things fit together.

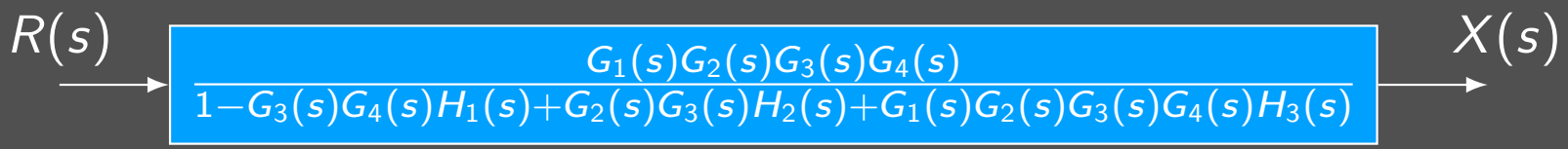
Step 4

$$X = G_3 G_4 \frac{G_2 G_1 R + (H_1 - G_2 G_1 H_3) X}{1 + G_2 H_2 G_3}$$

Rearrange terms:

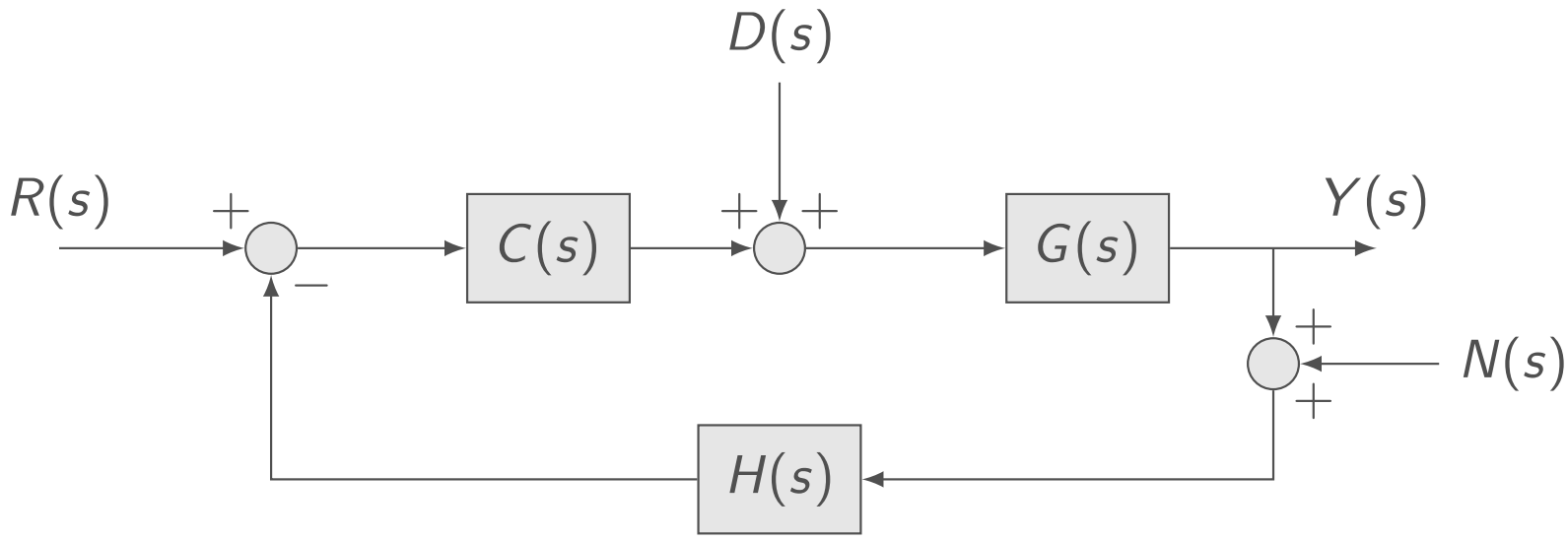
$$(1 - H_1 G_3 G_4 + G_2 H_2 G_3 + G_1 G_2 G_3 G_4 H_3)\, X = G_1 G_2 G_3 G_4 R$$

Same result!:



Multiple Inputs

- If a system has multiple inputs, an output is the sum of responses to each (**superposition**):



- Output given by[6]

$$Y(s) = \frac{C(s)G(s)}{1 + C(s)G(s)H(s)} R(s) + \frac{G(s)}{1 + C(s)G(s)H(s)} D(s) - \frac{C(s)G(s)H(s)}{1 + C(s)G(s)H(s)} N(s)$$

[6] One of the problems from this week's problem sheet.

Sanity Checks

- The system

$$X(s) = G(s)R(s) \quad \Longleftrightarrow \quad X(s) = \frac{N(s)}{D(s)}R(s)$$

  is equivalent to

$$D(s)X(s) = N(s)U(s)$$

- $D(s)$ determines the **system dynamics**, and should be the same for all transfer functions for a given output.
- $N(s)$ determines the **input dynamics**, and should be equal to the direct input/output path (**if a unique path exists**).

Conclusion

- We have introduced the most common controller structure: PID.
  - Conceptual understanding of each controller gain.
  - Heuristic methods for tuning.

- Also introduced block diagrams.
  - Method for analyzing interactions between multiple LTI invariant systems.
  - Graphical method for consolidation.
  - Algebraic method for consolidation.

- Next week: **Stability Analysis**

You can now complete problem 2