Lecture 13: Revision

Daniel Quevedo and Arman Siahvashi

27 May 2025

The University of Sydney

Table of contents

State Space vs Transfer Function Methods

State Space methods - as seen in this UoS

- ☺ Precise specification of all poles, general methods, computer implementation extends to MIMO & nonlinear systems
- ☹ How to choose poles? Difficult to assess robustness, etc, from matrices

Frequency-Domain methods – as seen in this UoS

- ☺ Visual representation, easy to see robustness, intuitive design procedures, infinite-dimensional systems (e.e., time-delays)
- ☹ Less precise specification, ad-hoc tuning methods, limited (mostly) to SISO LTI systems.

First and Second Order Systems

Understanding LTI Systems

A Linear Time-Invariant (LTI) system can be described by a simple first-order equation:

$$\dot{x}(t) + ax(t) = bu(t) \tag{1}$$

Where:

- $\dot{x}(t)$: Rate of change of the system's state (e.g., water level).
- $x(t)$: The current state of the system (e.g., water level in the tank).
- $u(t)$: The input signal (e.g., water flow into the tank).
- a: System parameter (e.g., leak rate).
- b: Input gain (e.g., efficiency).

These systems are called **time-invariant** because a and b do not change with time.

Solutions for LTI Systems - Overview

An LTI system **responds** differently depending on the input $u(t)$. Throughout this course, we will explore how different types of inputs affect the system's behavior:

- **No Input (Homogeneous Solution):** $u(t) = 0$
  - The system evolves based solely on initial conditions.
  - Solution: $x(t) = Ce^{-at}$
- **Step Input:** $u(t) = m$
  - The system reacts to a constant input.
  - Solution: $x(t) = [x(0) - \frac{m}{a}]e^{-at} + \frac{m}{a}$
- **Exponential Input:** $u(t) = Ae^{st}$
  - The system's response involves both transient and steady-state behavior.
  - Solution: $x(t) = \frac{A}{s+a}e^{st} + x_h(t)$

Key Insight: LTI systems can respond predictably to different inputs, helping us design controllers for desired outcomes.

Solving First-Order LTI Systems I: Homogeneous Solutions

To begin, let $u(t) \equiv 0$:

First-order homogeneous differential equation:

$$\dot{x}(t) + ax(t) = 0$$

Real-world examples:

- A cooling object **naturally** losing heat (with no input like fan or AC)
- A car slowing down when releasing the gas pedal (e.g. no break)
- A capacitor discharging in an electrical circuit

Assumed solution:

$$x(t) = Ce^{\lambda t}$$

where C is a constant and λ is to be determined. (see Lecture Notes 02)

Solving First-Order LTI Systems I: Homogeneous Solutions

Finding λ: Substituting $x(t) = Ce^{\lambda t}$ into the equation, we get:

$$C\lambda e^{\lambda t} + aCe^{\lambda t} = 0$$

Factoring out $Ce^{\lambda t}$, we obtain: $\lambda = -a$

Thus, the general solution is:

$$x(t) = Ce^{-at}$$

System Behavior:

- If $a > 0 \rightarrow$ **Stable system** (exponential decay to zero).
- If $a < 0 \rightarrow$ **Unstable system** (exponential growth).
- If $a = 0 \rightarrow$ **System remains constant**.

Key Takeaway: Homogeneous solutions describe natural decay or growth in systems. Understanding stability helps in **control system design**!

Second order ODE: From Mass-Spring to Suspension Systems

The equation

$$\ddot{x}(t) + a\dot{x}(t) + bx(t) = cu(t) \tag{2}$$

with a, b, c being constants is a second-order linear differential equation. For most of this lecture we will take, for simplicity, $c = 1$.

Example: The simple linear mass-spring-damper setup has a differential equation:

$$m\ddot{q} + c_0\dot{q} + k_0 q = u.$$

which can be put in the above form by dividing through by m.

Homogeneous Solutions

Let us try a solution $x_h(t) = e^{\lambda t}$ for the homogeneous equation

$$\ddot{x}_h(t) + a\dot{x}_h(t) + bx_h(t) = 0$$

Then we obtain: (Q: How?) *See Wk2 Supplementary Info. and Lecture Note 2 and 3 + Wk 4 videos on Canvas*

$$\lambda^2 e^{\lambda t} + a\lambda e^{\lambda t} + be^{\lambda t} = 0$$

- Since $e^{\lambda t}$ is never zero, this yields

$$\lambda^2 + a\lambda + b = 0$$

This equation is the so-called **characteristic equation** of the differential equation.

Homogeneous Solutions: characteristic equation

From

$$\ddot{x}_h(t) + a\dot{x}_h(t) + bx_h(t) = 0$$

we get $\lambda^2 + a\lambda + b = 0$ This is a quadratic equation. The possible values for λ are given by

$$\lambda = \frac{-a \pm \sqrt{a^2 - 4b}}{2}$$

which could have real or complex roots (called **poles** of the system):

- If $a^2 - 4b \geq 0$, we have two real poles.
- If $a^2 - 4b < 0$, we have a pair of complex poles.

Stability conditions: Real poles

- **Two distinct real roots** $\lambda_1, \lambda_2 \in \mathbb{R}$ $(a^2 - 4b > 0)$.

$$x_h(t) = C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t},\ C_1, C_2 \in \mathbb{R}.$$

- **Two equal roots** $\lambda_1 = \lambda_2 = \lambda_* \in \mathbb{R}$ $(a^2 - 4b = 0)$.

$$x_h(t) = C_1 e^{\lambda_* t} + C_2 t e^{\lambda_* t},\ C_1, C_2 \in \mathbb{R}.$$

To ensure stability, we need the two real poles, distinct or equal, to be negative.

Second Order Systems (Complex Poles)

Consider the second-order system

$$\ddot{x}(t) + a\dot{x}(t) + bx(t) = cu(t) \tag{3}$$

with $a > 0$, $b > 0$, and a unit step input $u(t) = \mathbf{1}(t)$

- Assume the characteristic equation admits two stable complex poles $\lambda_1 = -\sigma + j\omega_d$ and $\lambda_2 = -\sigma - j\omega_d$ with $j = \sqrt{-1}$, $\sigma > 0$, and $\omega_d > 0$.
- The relationship between parameters and pole locations is

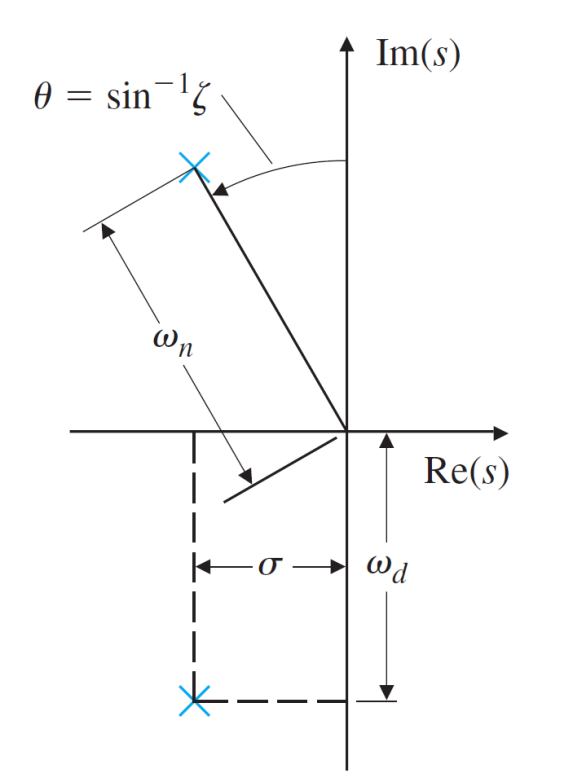
$$\sigma = \frac{a}{2}, \ \omega_d = \frac{\sqrt{4b - a^2}}{2}$$

- Define $\omega_n = \sqrt{\sigma^2 + \omega_d^2} = \sqrt{b}, \ \zeta = \sigma/\omega_n = a/2\omega_n$
- The system equation becomes

$$\ddot{x}(t) + 2\zeta\omega_n\dot{x}(t) + \omega_n^2 x(t) = c$$

Response Specifications

Graphical Representation of Complex Poles (Complex Plane)



Importance of the Complex Plane:

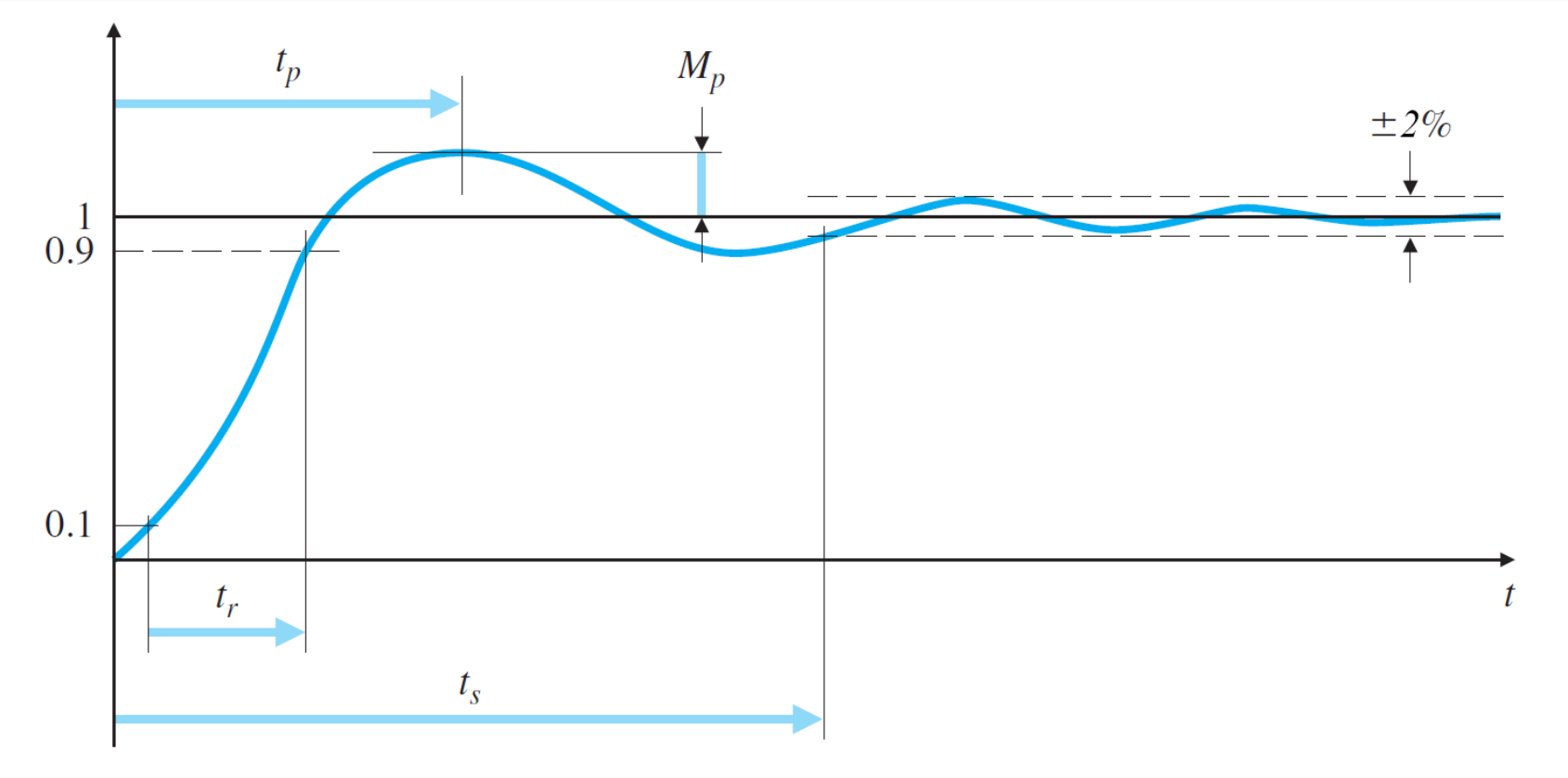
- Visualizes system stability and transient response.
- Helps in analyzing and optimizing system performance.
- Essential for effective controller design.
- Left Half-Plane (stable) and Right Half-Plane (unstable)

Symbol Summary:

Symbol	Meaning and Effect
σ	Decay rate (Real part), determines stability/speed.
ω_d	Damped oscillation frequency, defines oscillation.
ω_n	Natural frequency, affects response speed.
ζ	Damping ratio, influences overshoot/stability.

Specification

There are several important specifications, including rise time (t_r), peak time (t_p), settling time (t_s), and overshoot (M_p). **These specifications arise from real-world needs.**



Response Specifications Overview

Specification	Meaning/Application	Equation
Overshoot (M_p)	Maximum amount the response exceeds the steady-state value, indicating overshoot tendency.	$M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}}$
Time to Peak (t_p)	Time to reach the first peak of the response, showing how quickly the system responds.	$t_p = \frac{\pi}{\omega_d}$
Settling Time (t_s)	Time for the response to stay within a small percentage (e.g., 2%) of the steady-state value.	$t_s \approx \frac{4}{\sigma}$ or $t_s \approx \frac{4}{\zeta\omega_n}$
Rise Time (t_r)	Time for the response to rise from 10% to 90% of its final value, measuring response speed.	$t_r \approx \frac{1.8}{\omega_n}$

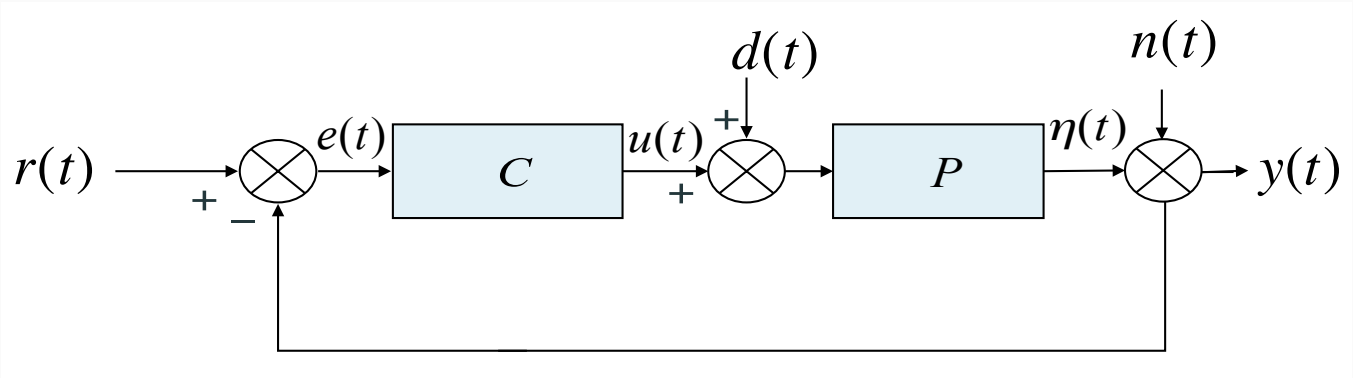
Homework: Find out how these equations were derived (Hint: Lecture Note 5, and Wk5 Supplementary Information)

Objective: Controller Design for a Second-Order System

Suppose we have a second-order dynamical system

$$\ddot{x}(t) + \dot{x}(t) = u(t) \tag{4}$$

and the goal is to design a feedback controller so that the response $y(t)$ can track a constant reference $r\mathbf{1}(t)$.



PD Control

We can also design a controller as certain combination of a proportion of the error itself (P-control) and the derivative of the error (D-control). This is called PD control described by

$$u(t) = K_P e(t) + K_D \dot{e}(t) = K_P(r - x(t)) + K_D(\dot{r}(t) - \dot{x}(t))$$

The closed-loop system becomes

$$\ddot{x}(t) + (1 + K_D)\dot{x}(t) + K_P x(t) = K_P r(t) + K_D \dot{r}(t) \tag{5}$$

By selecting different values of K_D and K_P,you can position the closed-loop poles anywhere in the left half-plane (where the system is stable). This flexibility allows you to achieve virtually any desired transient behavior—such as specific damping, rise time, or settling time—meaning, at least in theory, any response specification can be met.

Rate Feedback

The closed-loop transfer function under PD control is

$$G(s) = \frac{K_P + K_D s}{s^2 + (1 + K_D)s + K_P}$$

- There is now a *zero* in the system, i.e. a value for s at which $G(s) = 0$. In particular, the zero is at $s = -K_P/K_D$. This will affect the system response.

Sometimes PD control refers to the controller of the following form

$$u(t) = K_P(r - x(t)) - K_D\dot{x}(t),$$

i.e. the reference signal is not differentiated. This is also sometimes called rate feedback because it uses the time derivative of the output (e.g., velocity if x is position) rather than differentiating the entire error. It helps control the system's speed of response and avoids issues with noisy or discontinuous references.

Example

Consider the following second-order system $\ddot{x} + 3\dot{x} + 2x = u$

- What are the poles of the system?
- Is the system stable or not?
- Let $r(t)$ be a constant reference. Design a PD controller

$$u(t) = K_p(r - x) - K_d\dot{x}$$

so that the system response to a **step input has a settling time around 2 sec and an overshoot of about 5%.**

Solution

- $\ddot{x} + 3\dot{x} + 2x = u \implies G(s) = \frac{1}{s^2+3s+2}$
- **Characteristic Equation:** $s^2 + 3s + 2 = 0 \Rightarrow s = -1, -2$ (stable)
- **Controller form:** $u(t) = K_p\big(r - x\big) - K_d\dot{x}$, so the closed-loop transfer function is $G(s) = \frac{K_p}{s^2+(3+K_d)s+K_p+2}$
- **Specs for unit step**
  - Settling time: $T_s \approx 2\,\text{s}$
  - Percent overshoot: $M_p \approx 5\%$
- $M_p = e^{-\pi\zeta/\sqrt{1-\zeta^2}} = 0.05 \Rightarrow \boxed{\zeta \approx 0.7}$
- $T_s \approx \frac{4}{\zeta\omega_n} = 2\,\text{s} \Rightarrow \boxed{\omega_n \approx 2.86\,\text{rad/s}}$
- Desired characteristic poly: $s^2 + 2\zeta\omega_n s + \omega_n^2 = s^2 + 4s + 8.17$
- $3 + K_d = 4 \Rightarrow \boxed{K_d = 1}$
- $2 + K_p = 8.17 \Rightarrow \boxed{K_p \approx 6}$

State Space Methods

State-Space Models

- Linear ODEs can be written in state-space form:

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t)$$
$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t),$$

  where $\mathbf{x} \in \mathbb{R}^n$ is the **state**.
- To **predict** the system states and outputs, we only require knowledge of the **current** state and the future inputs (control variables!).
- The system state satisfies:

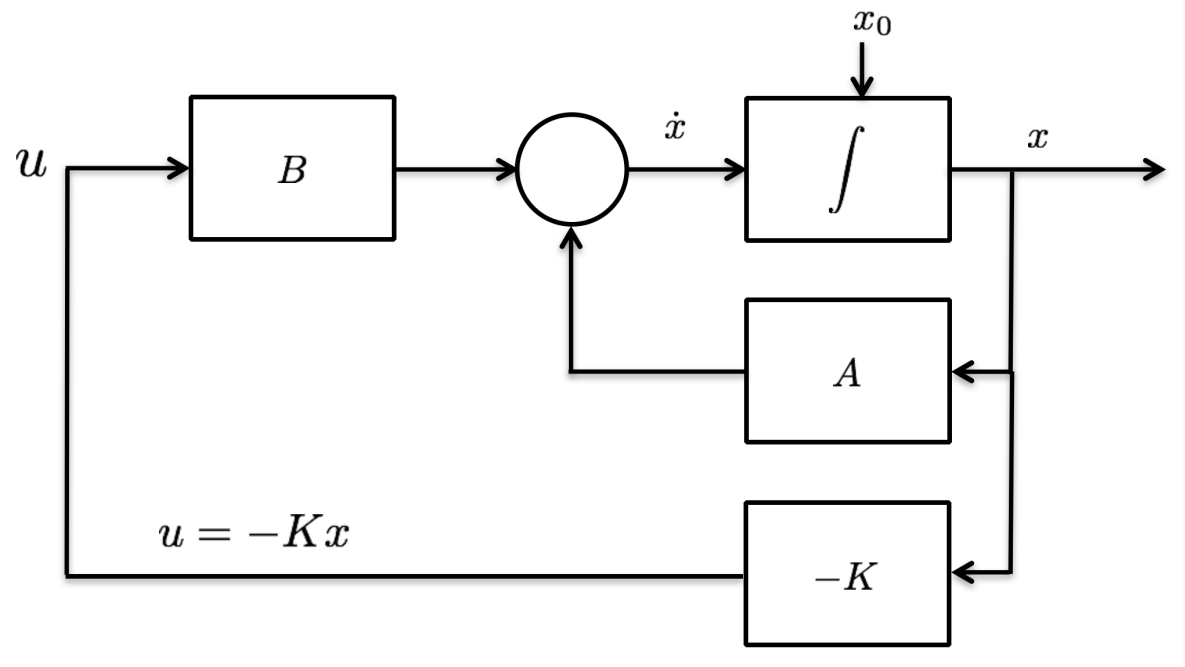
$$\mathbf{x}(t) = e^{At}\mathbf{x}_0 + \int_{\tau=0}^{t} e^{A(t-\tau)}B\mathbf{u}(\tau)d\tau.$$

- We have

$$G(s) = C(sI - A)^{-1}B + D$$

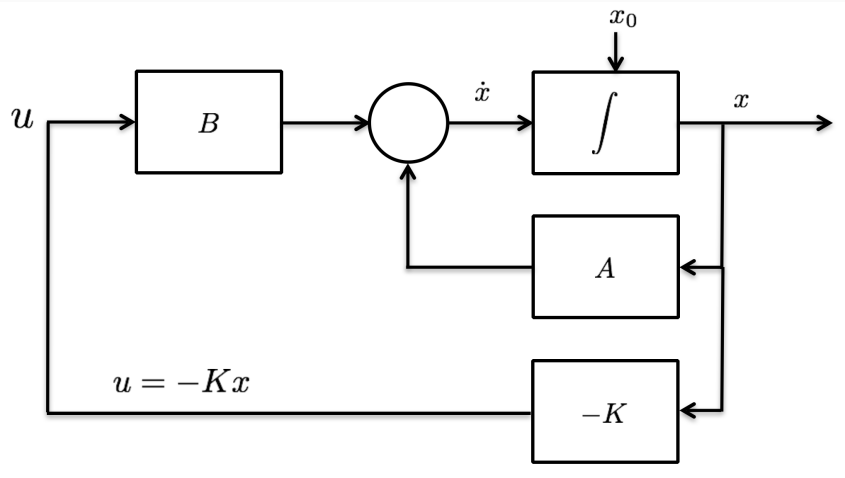
- The system is stable iff all eigenvalues of A, i.e., the poles of $G(s)$ have negative real parts.
- **Coordinate Invariance**: $\mathbf{z}(t) = T\mathbf{x}(t)$.

Control of LTI Systems using State Feedback



- Feedback allows us to **design** the dynamics of a system.
- We have learned how to synthesise linear state feedback laws $u(t) = -K\mathbf{x}(t)$ using **coefficient matching**.
- A linear dynamical system is **reachable** iff **any** final state $\mathbf{x}(T)$ can be reached from **any** initial state $\mathbf{x}(0)$ in finite time T.
- Reachable systems can be transferred into **reachable canonical form**. This simplifies the pole placement procedure.

Controller design via Pole Placement



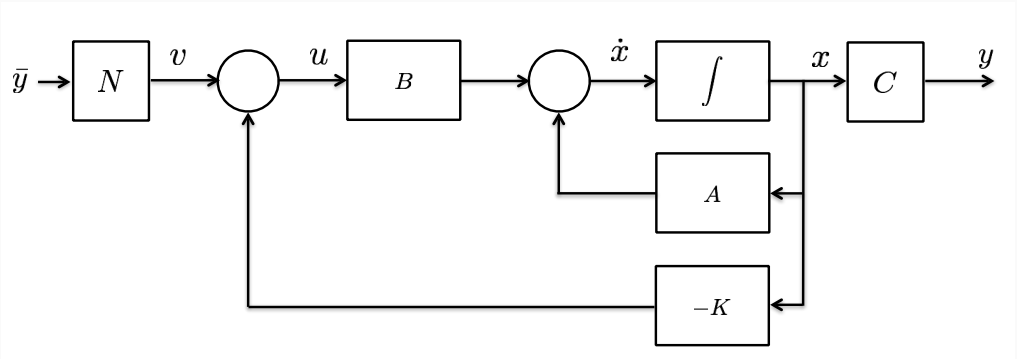
How should we choose the poles? A typical method is:

1. Choose second-order dominant poles based on desired transient response (e.g., step-response specifications)[1].
2. Place the remaining poles so that they are "less dominant":
   - Settling time 5-10 times faster (real part σ of poles 5-10 times further left).
   - Low overshoot (large damping ratio ζ).

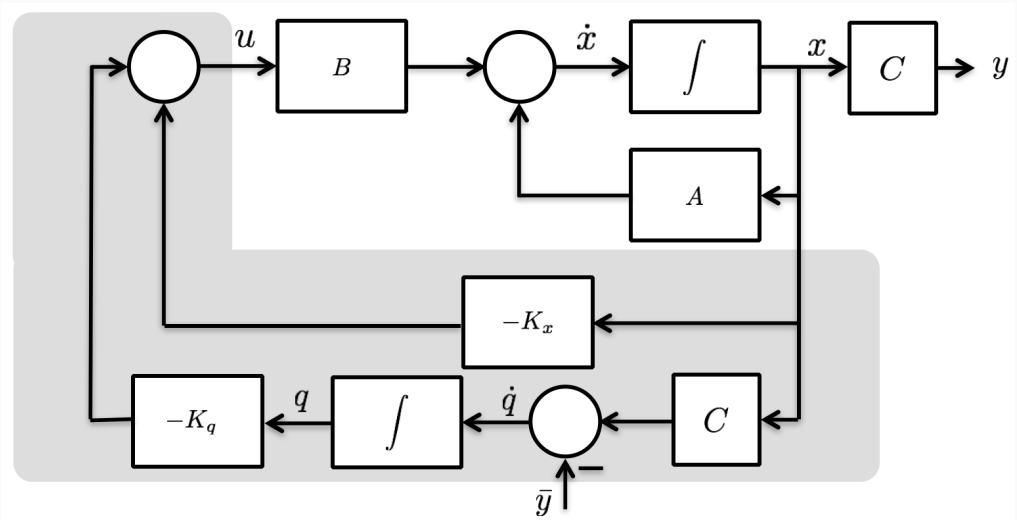
[1] Also taking into account practical issues, such as model uncertainties or noise.

Regulation to a Non-zero Set point

1. **Reference feedforward**



2. **Integral Control**

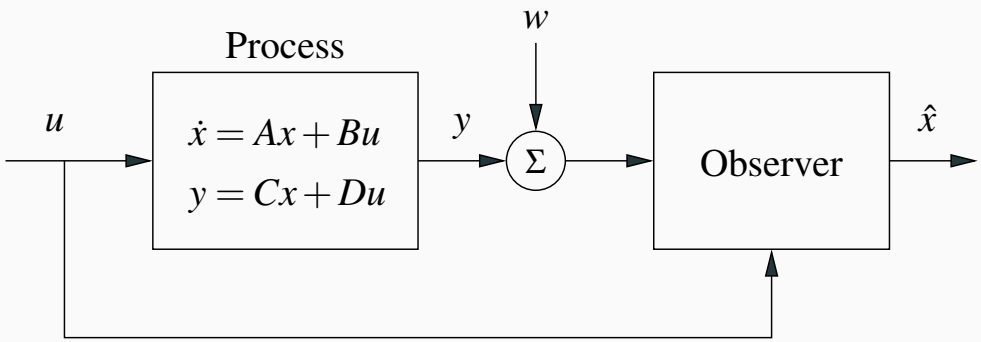


- Integral control action overcomes (some) steady-state errors!
- We require closed-loop stability, but no exact system model.

Observability

- To implement $\mathbf{u}(t) = -K\mathbf{x}(t)$ in practice, the control system needs measurements of $\mathbf{x}(t)$. Often this is not possible, or too expensive.
- Can we use state-feedback control ideas without using $\mathbf{x}(t)$?

Definition: A linear system $\dot{\mathbf{x}} = A\mathbf{x}(t) + B\mathbf{u}(t)$, $\mathbf{y}(t) = C\mathbf{x}(t)$ is **observable**, if and only if for any finite time $T > 0$ it is possible to determine the state $\mathbf{x}(T)$ from measurements of $\mathbf{y}(t)$ and $\mathbf{u}(t)$ on the interval $t \in [0, T]$.



Observability depends on the system dynamics and the choice of sensors. The latter determine the matrix C.

State Observers

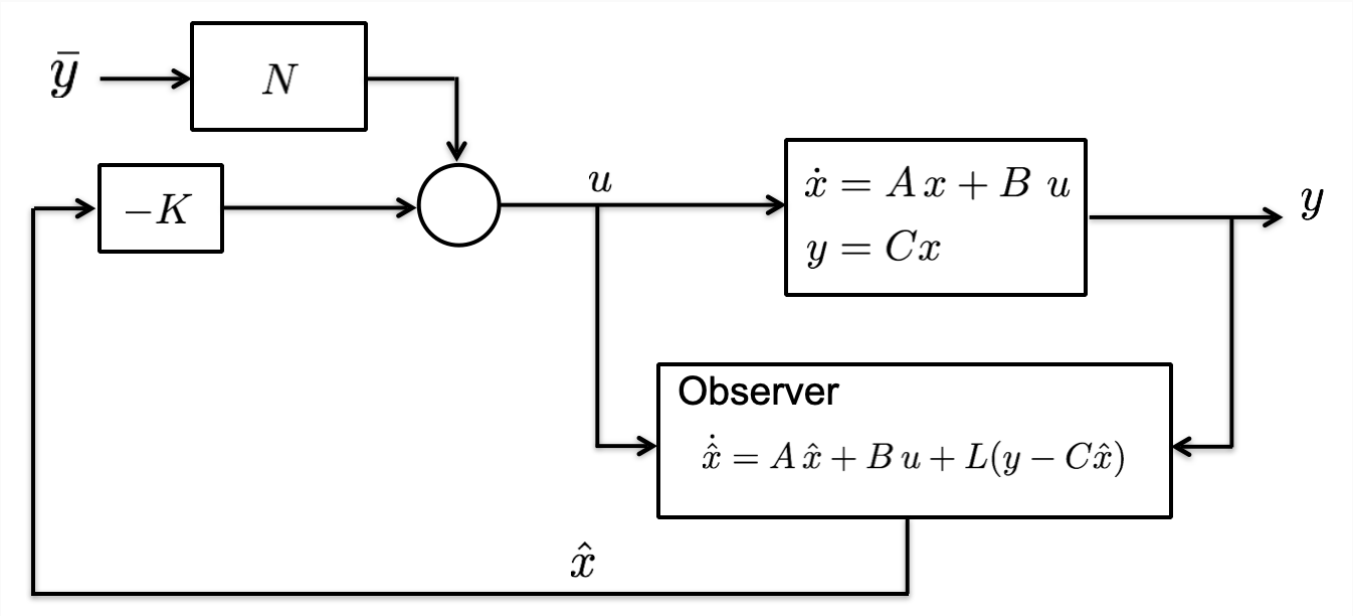
- A **state observer** is a dynamical system that uses measurements of $\mathbf{y}(t)$ and $\mathbf{u}(t)$ to compute a state estimate $\hat{\mathbf{x}}(t)$.
- A common structure is the "predictor + corrector" form:

$$\frac{d}{dt}\hat{x}(t) = \underbrace{A\hat{x}(t) + Bu(t)}_{\text{Model prediction}} + \underbrace{L(y(t) - C\hat{x}(t))}_{\text{Measurement correction}}$$

- The gain L serves to balance between:
  1. predictions based on the model and the current estimate $\hat{x}(t)$
  2. new information arriving in the measurement $y(t)$.
- The following are **equivalent**:
  1. The system (A, C) is **observable**.
  2. The **observability matrix** W_o has full rank.
  3. **Pole placement**: For any desired set of n eigenvalues, there exists a **gain matrix** L such that $A - LC$ has those eigenvalues.
- Observers can be designed using coefficient matching or duality.

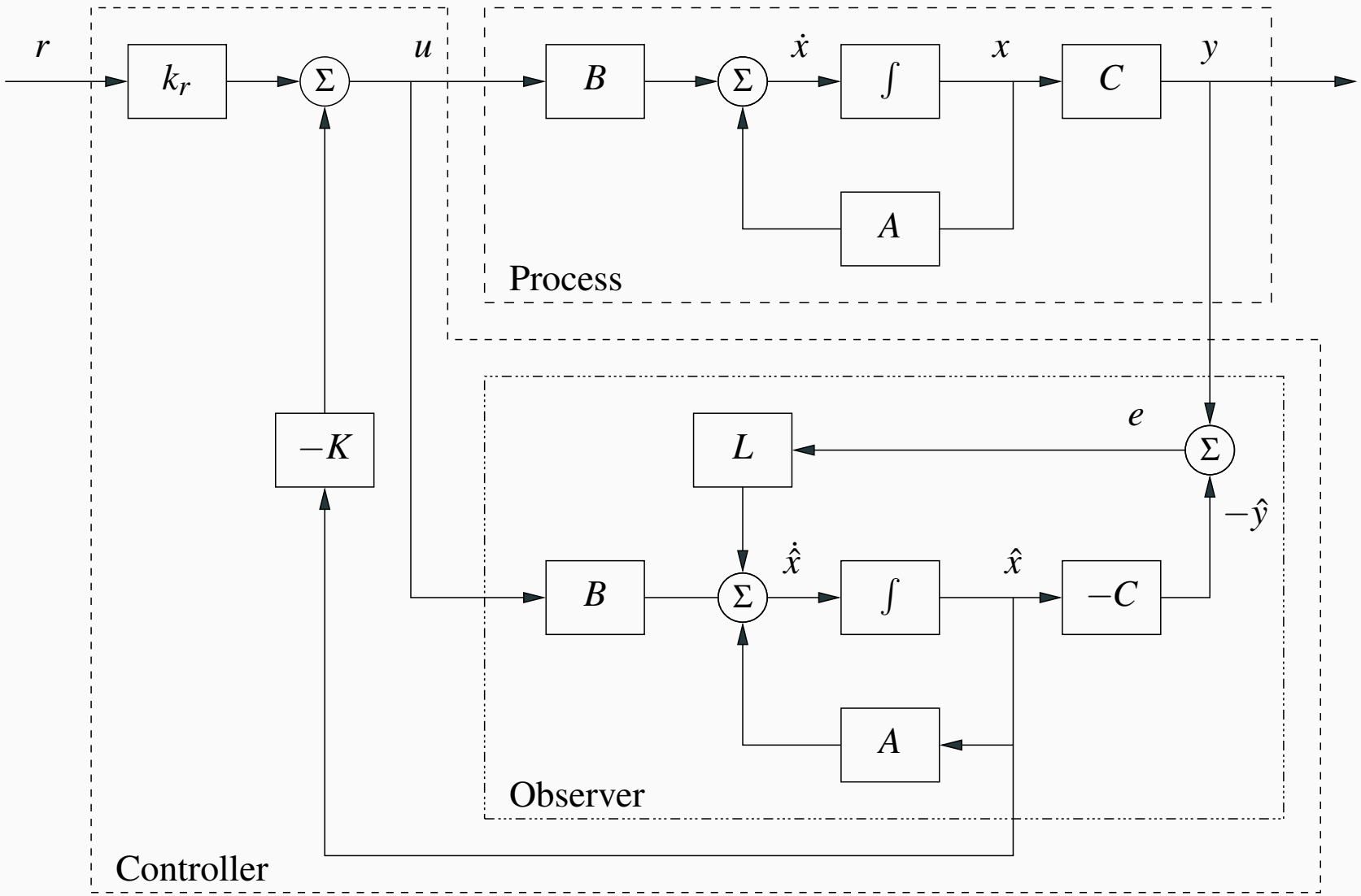
Output feedback Control

The idea is to combine state estimation with state feedback by simply replacing the state $\mathbf{x}(t)$ by its estimate $\hat{\mathbf{x}}(t)$.



$$\frac{d}{dt}\begin{bmatrix}\mathbf{x}(t)\\ \tilde{\mathbf{x}}(t)\end{bmatrix} = \begin{bmatrix}A-BK & BK\\ \mathbf{0} & A-LC\end{bmatrix}\begin{bmatrix}\mathbf{x}(t)\\ \tilde{\mathbf{x}}(t)\end{bmatrix} + \begin{bmatrix}BN\\ \mathbf{0}\end{bmatrix}\bar{y}.$$

Structure


r
k_r
Σ
u
B
Σ
$\dot{x}$
∫
x
C
y
A
Process
−K
L
e
Σ
$-\hat{y}$
$\dot{\hat{x}}$
$\hat{x}$
B
Σ
∫
−C
A
Observer
Controller

Transfer Functions and Frequency-Domain

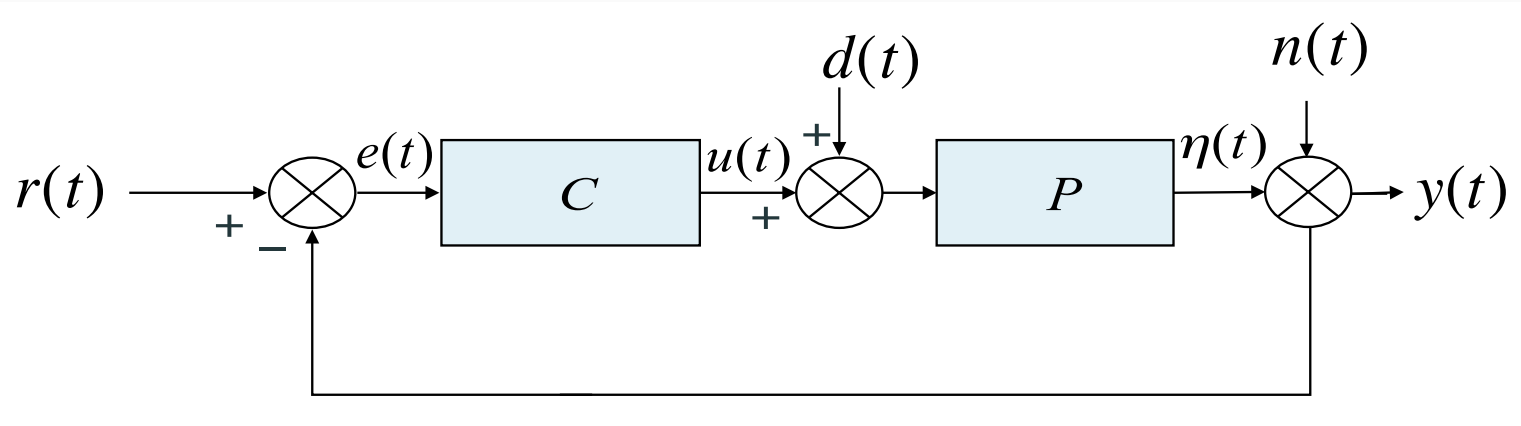
Transfer functions from ODE Models

Consider ODEs written as

$$\frac{d^n y}{dt^n} + a_1 \frac{d^{n-1} y}{dt^{n-1}} + ... + a_n y = b_0 \frac{d^m u}{dt^m} + b_1 \frac{d^{m-1} u}{dt^{m-1}} + ... + b_m u.$$

- In this case, $y(t) = G(s)u(t)$, where $G(s)$ is given by the rational function
$$G(s) = \frac{b_0 s^m + b_1 s^{m-1} + ... + b_m}{s^n + a_1 s^{n-1} + ... + a_n}.$$
- Some common examples include:
- **Integrator**, $\dot{y} = u$: $G(s) = \frac{1}{s}$.
- **Differentiator**, $y = \dot{u}$: $G(s) = s$.
- **First-order system**, $\dot{y} + ay = bu$: $G(s) = \frac{b}{s+a}$
- **Second-order system**, $\ddot{y} + 2\zeta\omega_0 \dot{y} + \omega_0^2 y = \omega_0^2 u$:
$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}.$$

Block Diagrams



- **If a system is stable**, then in steady state the response due to initial conditions will have vanished and we simply have

$$y(t) = G(s)u(t).$$

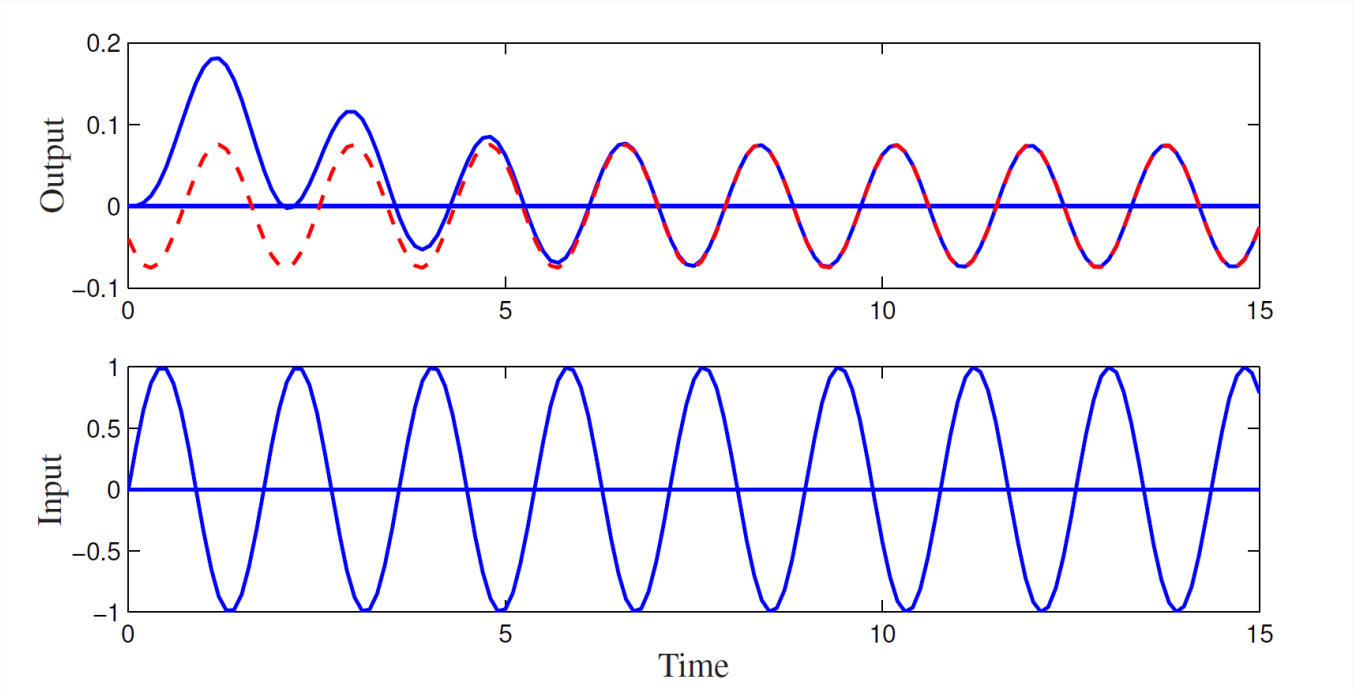
- Block diagrams allow us to examine **interconnections of systems**.
- A block diagram may include a number of input-output pairs, and therefore **several transfer functions**.
- These can be found using block diagram algebra.
- For example, we have

$$r(t) \to e(t): \qquad G_{er}(s) = \frac{1}{1 + P(s)C(s)}.$$

Frequency Response of LTI Systems

A sine wave input to an LTI system with transfer function $G(s)$ forces a sine wave output with the **same frequency**.
The output is **amplified** by $|G(j\omega)|$ and **phase-shifted** by $\phi(\omega)$.

$$u(t) = A\sin(\omega t) \longrightarrow y(t) = |G(j\omega)|A\sin(\omega t + \phi(\omega)).$$



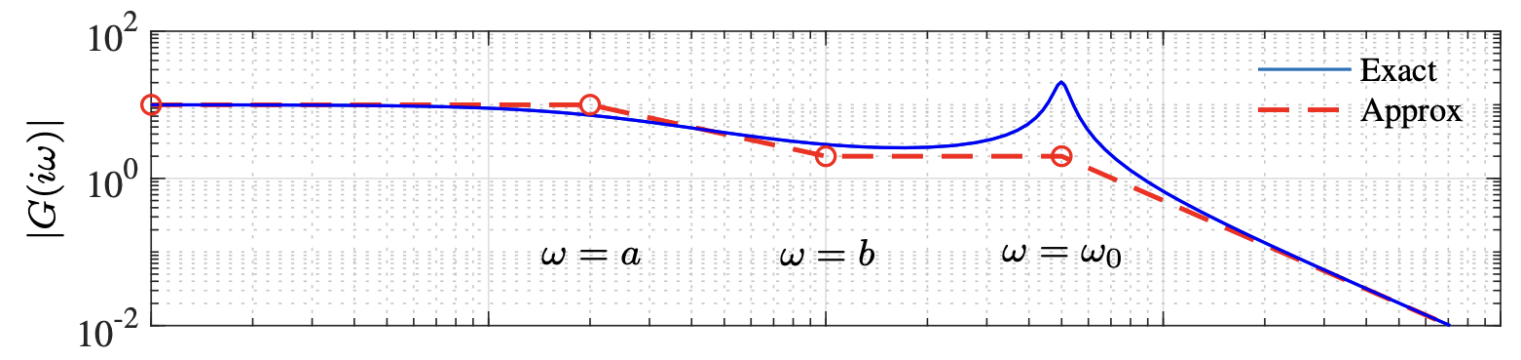
At times one can determine $G(s)$ by measuring the frequency response.

Bode Plots

A Bode plot visualises the range of sinusoidal responses for all frequencies. It contains two subplots:

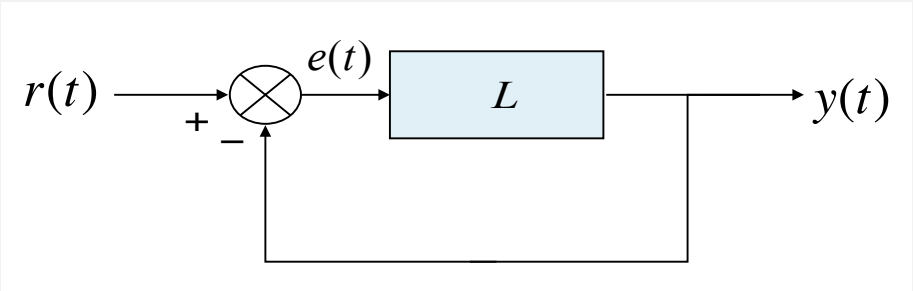
1. A plot of **magnitude** $|G(j\omega)|$ vs frequency ω.
2. A plot of **phase angle** $\angle G(j\omega)$ vs frequency ω.

- Bode plots of high-order transfer functions can be built from plots of basic terms.
- This can be done by hand using approximations, or using computers.



Closed-Loop Stability from the Open-loop Transfer Function

Open-loop transfer function: $L(s) = P(s)C(s)$

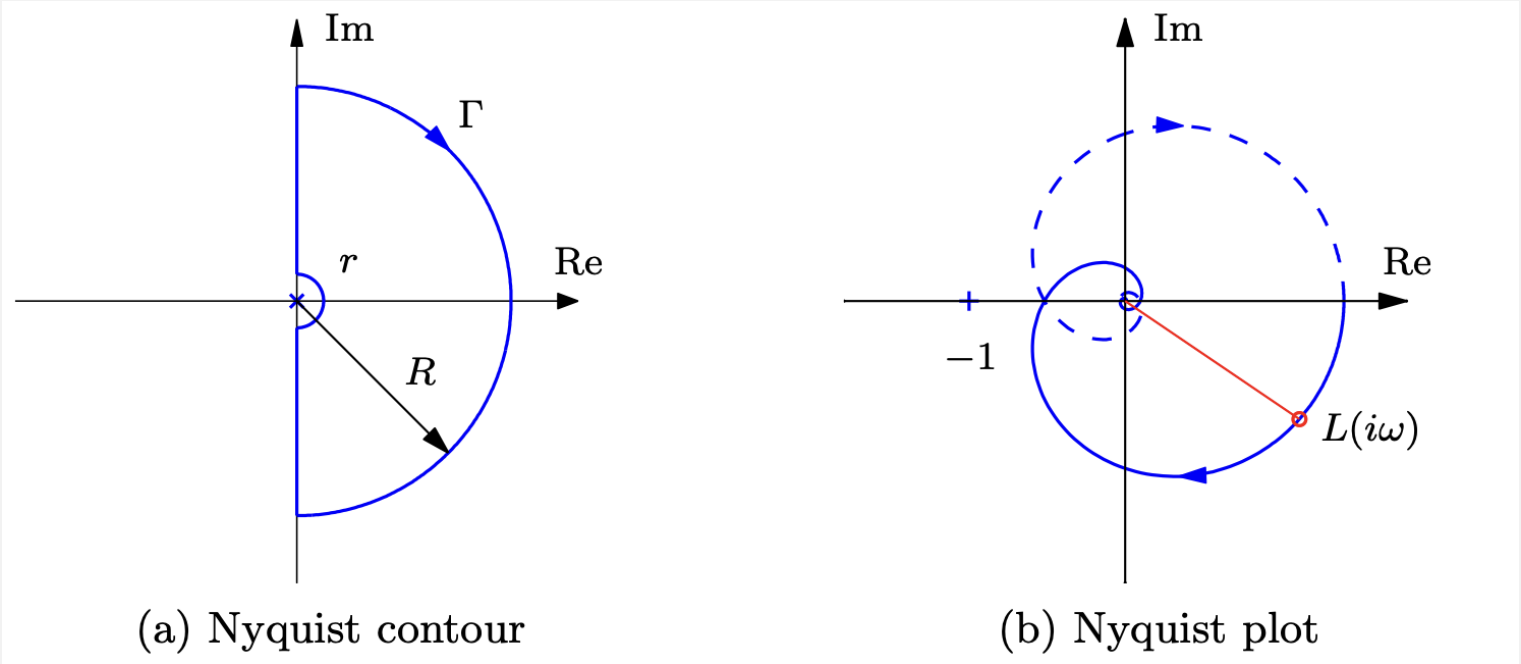


- (Some) closed-loop transfer functions are given by:

$$G_{yr}(s) = \frac{L(s)}{1 + L(s)}, \quad G_{er}(s) = \frac{1}{1 + L(s)}.$$

- The closed loop poles are the values s, such that $L(s) = -1$.
- A plot of the **open-loop** frequency response $L(j\omega)$ allows us to precisely state how many **closed-loop** poles are located in the right half plane!

Nyquist Plots



(a) Nyquist contour (b) Nyquist plot

- Plot $L(j\omega)$ on the complex plane, for all ω from $-\infty$ to ∞.
- Matlab command `nyquist` can be used to produce a Nyquist plot:

$$L(s) = 1.4\frac{e^{-s}}{(s+1)^2}$$

```
s = tf("s"); L = 1.4*exp(-s)*tf([1],[1 2 1]), nyquist(L)
```

Nyquist Theorem

Theorem
Consider the loop transfer function $L(s)$ and the closed-loop transfer function

$$G_{yr}(s) = \frac{L(s)}{1 + L(s)}.$$

Suppose that:

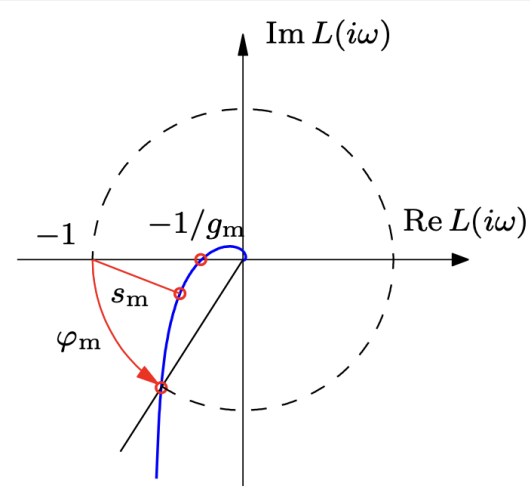
1. *$L(s)$ has P poles in the right-half-plane.*
2. *For the Nyquist plot of $L(s)$, the net number of clockwise encirclement of the point $s = -1$ is N.*

Then $G_{yr}(s)$ has $N + P$ poles in the right half-plane.

In other words, for stability the Nyquist plot should have $N = -P$ clockwise encirclements, i.e., exactly P anti-clockwise encirclements.

Robustness Margins

- An important property of feedback systems is their ability to **be robust to variability and model uncertainty**.
- If there is model uncertainty, then the true (but **unknown**) open loop transfer function is different to its model.
- Assuming that the nominal system (the model of $L(s)$) is stable, stability margins quantify **distance to instability**.
- Commonly used **Stability margins** can be read out from Bode or Nyquist plots:



1. Gain Margin: g_m
2. Phase Margin: ϕ_m
3. Stability Margin: s_m

Sensitivity Functions and Loop Shaping (Feedback Limitations)

What Limits Feedback Control? Overview

Not all control problems can be solved by clever design or tuning. Some systems are fundamentally hard to control, no matter which method you use.

In this lecture, we'll look at **4 theoretical limitations** that explain why:

1. **The Identity:** $S + T = 1$
   You cannot reduce both disturbance and noise sensitivity at the same time.
2. **Bode's Gain–Phase Relation**
   Steeper gain roll-off leads to more phase lag, hurting stability.
3. **Interpolation Constraints**
   You can't cancel unstable poles or zeros, you're stuck with them.
4. **Bode's Sensitivity Integral**
   If you push sensitivity down at some frequencies, it must pop up elsewhere.

These aren't design flaws; they're unavoidable trade-offs in feedback control.

1-First Limitation: Complementary Sensitivity Function T

Goal: We want the system to track well, reject disturbances, and ignore sensor noise.The tracking error is: $r - \eta = S \cdot r - S \cdot P \cdot d + T \cdot n$

- $S = \frac{1}{1+PC}$ is the **sensitivity function**. (e.g. model changes, tracking error, and disturbances)
- $T = \frac{PC}{1+PC}$ is the **complementary sensitivity function**. (handles what S(s) doesn't e.g. reference tracking, and sensitivity to noise)
- So: $S + T = 1$ (each affect different parts of performance and work together to balance different goals)

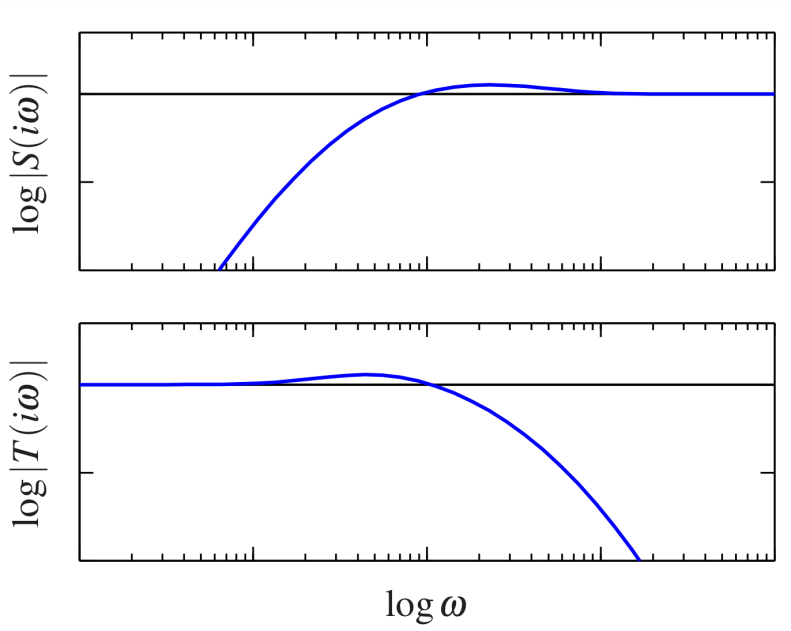
Key point: You cannot make both $|S(j\omega)|$ and $|T(j\omega)|$ small at the same time for all frequencies. Like a seesaw: one goes down, the other goes up.

- Make $|S(j\omega)|$ small at low frequencies $\rightarrow$ better tracking and disturbance rejection.
- Make $|T(j\omega)|$ small at high frequencies $\rightarrow$ better noise rejection.

You must choose where to place your trust, in disturbance rejection or in measurement accuracy.

Shaping S and T

- For many real systems, reference signals and disturbances have fairly low frequency content (i.e. change infrequently or slowly), while measurement noise tends to be high-frequency.
- For such systems, it makes sense to have $|S(j\omega)|$ small at **low frequencies** (to reject disturbances and track the reference well, and $|T(j\omega)|$ small at **high frequencies** (to suppress sensor noise).



This can be achieved by having $L(j\omega) \gg 1$ at **low frequencies**, and $L(j\omega) \ll 1$ at **high frequencies.**

2-Second Limitation: Gain Roll-off vs. Stability

We often want to make gain drop fast to block high-frequency noise. "Gain drop-off" refers to how the magnitude of the loop gain $|L(j\omega)|$ decreases as the frequency ω increases.

But Bode's rule says:

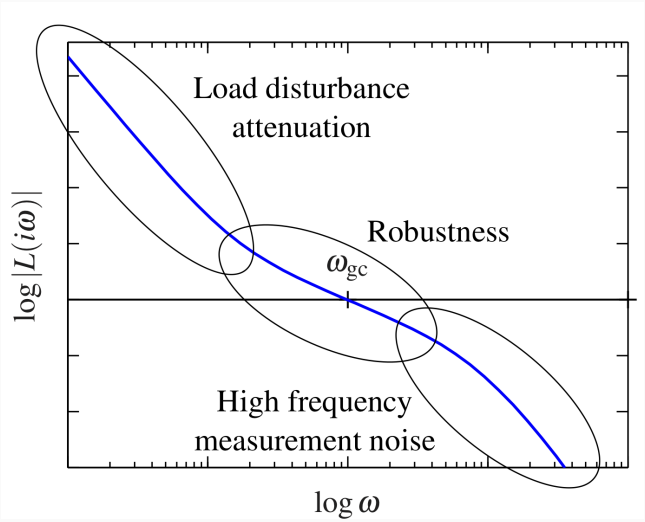
- Faster gain drop $\Rightarrow$ more phase lag.
- Too much phase lag $\Rightarrow$ lower phase margin.
- Low phase margin $\Rightarrow$ risk of instability.

So: Don't make the gain drop too sharply near crossover frequency!

Why does this matter?

- The **crossover frequency** ω_{gc} is where $|L(j\omega_{gc})| = 1$.
- This is the point where the system's **speed** and **stability** are **most sensitive.**
- If you drop gain too fast just before or after this point, phase lag increases rapidly.

Desired Loop Shape



The loop transfer function $L = PC$ should be shaped carefully to balance performance and stability:

- **High gain at low frequencies** $\rightarrow$ for good reference tracking and disturbance rejection.
  (Because $S = \frac{1}{1+L} \approx 0$ when $L \gg 1$)
- **Low gain at high frequencies** $\rightarrow$ to filter out measurement noise.
  (Because $T = \frac{L}{1+L} \approx 0$ when $L \ll 1$)
- **Gentle slope near crossover** (ω_{gc}) $\rightarrow$ to keep phase margin and avoid instability.(Bode's relation: $\angle L(j\omega_{gc}) \approx 90^\circ \times \frac{d\log|L(j\omega)|}{d\log\omega}$)

3-Third Limitation: Interpolation Constraints

- No matter how clever your controller is, the sensitivity function $S(s)$ has to behave a certain way if there are RHP poles or zeros.

For internal stability, the closed-loop sensitivity $S(s)$ must satisfy:

1. $S(s)$ is finite for all s with $\text{Re}(s) \geq 0$, **You can't have infinite gain in the RHP (that would be unstable)**
2. $S(s) = 1$ at any RHP zero of $L(s)$, **That means feedback has no effect at those frequencies.**
3. $S(s) = 0$ at any RHP pole of $L(s)$, **The system is trying to force the output to 0 at that unstable frequency**

These are called **interpolation constraints** because they "pin down" the shape of $S(s)$ at certain points.**Your controller has to "interpolate" between these fixed values**, it has no choice. We cannot cancel RHP poles or zeros using feedback — we are stuck with them and must design around their effects.

4-Fourth Limitation: Bode's Sensitivity Integral

- The sensitivity function $S(j\omega)$ tells us how much the system output reacts to disturbances or model uncertainty at frequency ω.
- Can we make $|S(j\omega)|$ small at **all frequencies** for perfect disturbance rejection?
- **Answer: No.** There is a fundamental trade-off:

$$\int_0^\infty \log|S(j\omega)|\, d\omega = \pi \sum_k p_k$$

  where p_k are the unstable poles of the open-loop system $L(s) = PC$. This equation tells us that if we add up the 'log size' of the sensitivity function across all frequencies, the total area is fixed.
- This means:
  - If $|S(j\omega)|$ is small in one region (good performance),
  - Then it must be **larger elsewhere** — you can't win everywhere!
- This is known as the **"waterbed effect"**: push sensitivity down in one spot, and it pops up in another.

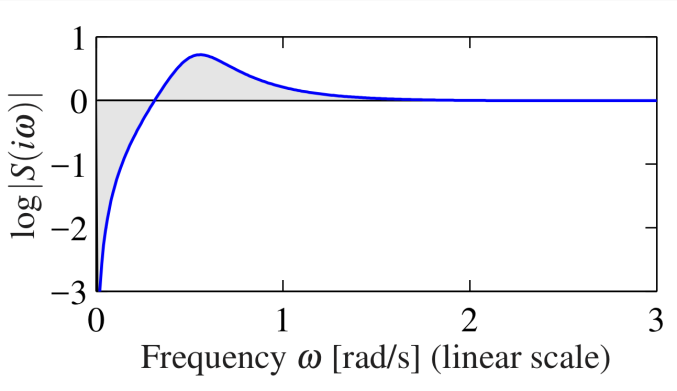
Key idea: Feedback can't eliminate all sensitivity. You must decide **where** to perform well, and where to compromise.

Interpretation: Waterbed Effect

- Bode's sensitivity integral says that for a **stable system**:

$$\int_0^\infty \log|S(j\omega)|\, d\omega = 0$$

- If $\log|S(j\omega)| < 0$ at some frequencies i.e. $|S(j\omega)| < 1$ (good disturbance rejection), then we must have $\log|S(j\omega)| > 0$, i.e. $|S(j\omega)| > 1$ for others.

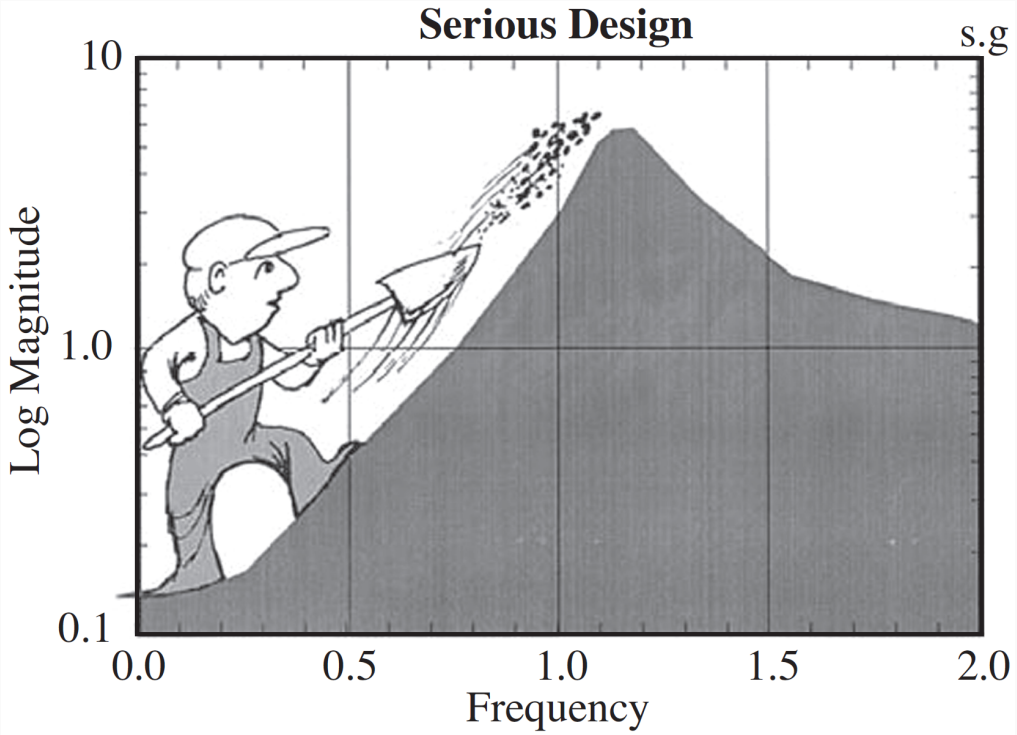


- The blue curve shows $\log|S(j\omega)|$.
- Below zero: sensitivity is low (pushed down) $\rightarrow$ good.
- Above zero: sensitivity is high (pops up) $\rightarrow$ bad.

You can't make sensitivity low everywhere — just like a waterbed, if you press down in one place, it rises in another.

Conservation Law

Dr. Gunter Stein interpreted this as a conservation law: **"conservation of dirt"**. The designer reduces sensitivity in some frequency band, but is always adding sensitivity in another. There is a risk in auto-tuning control design that you don't know where the dirt is being piled up.



The Final Exam

Basic Information

- The Final Exam will be held on Friday, 20 June.
- Assessment **of** learning.
- Worth 45% of your total mark.
- Individual, in person, no computer or generative AI use allowed.
- Programmable calculators are allowed, but not computers or phones!
- Students may bring a personal double-sided A4 size cheat sheet.
- This is a **Hurdle task**. Passing the exam is a **necessary, but not sufficient** condition for passing the unit.
- Past years' exam papers will be released this week.
- Lectures, lecture notes, weekly worked examples, problem sets, and design projects will cover all technical skills and knowledge needed for the exam.

Exam Structure

10 min reading + 120 min writing

There will be a total of 9 questions in the following four sections:

- **Section 1**: First and Second Order Systems (25 marks).
- **Section 2**: State Space Control Design (35 marks).
- **Section 3**: Real-world Dynamics and Control (15 marks).
- **Section 4**: Frequency Domain Control Design (25 marks).

These questions are to some extent similar (but not identical!) in format to your weekly worked examples and problem set questions.

Special Instructions

- ALL matrices involved are at most three by three; all numbers involved are basic which can be easily handled by a simple calculator.
- Write down all your working (intermediate steps) in the **appropriate spaces (boxes)** provided for each question. We mark the working, not just the final result for each question.
- Extra working-out spaces will be provided in the same exam paper but they will NOT be marked.

Exam References

You will have seen or worked on similar (but not identical) problems in lectures, worked examples, problem sets, and lab report questions.

- Lecture slides/notes, Problem sets, Worked examples
- Textbook
- Past exam papers: All exam papers (without solutions) will be released on Canvas this week.
- Note that exams before 2017 were designed for a different course curriculum, and exams between 2020 and 2022 were online open-book exams.

Support for Exam Preparation

- All Week 13 timetabled Practicals and Lab sessions have only one goal:

  > Clarify doubts and build a solid understanding of solutions to the tutorial questions and problem sets. We hope that this week's activities will help you to be on solid grounds to show your learning during the final exam.

- In the combined three hours, you may ask your tutor any question about any of the worked examples and problem-setting questions we developed for you.
- Your tutor will try to provide the best explanation to support your understanding.
- Your tutor may not be able to answer some questions.

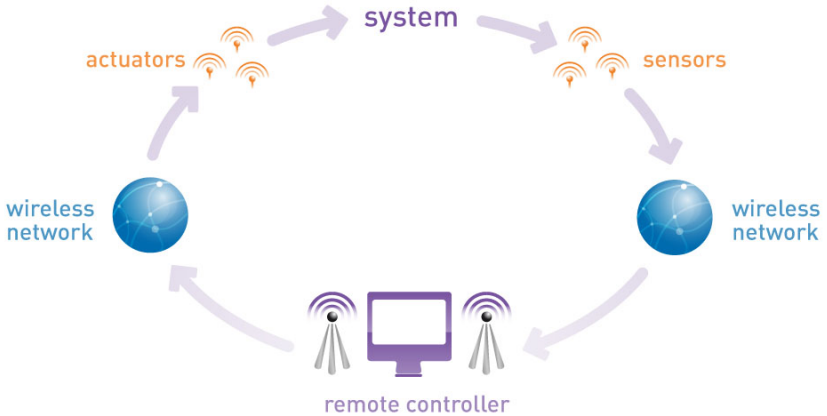
Feedback Systems for the Future

Control System Methodologies Beyond This Course

Our course solely covered systems theory and feedback design for **continuous-time deterministic LTI systems**.
Many additional exciting topics are worth exploring:

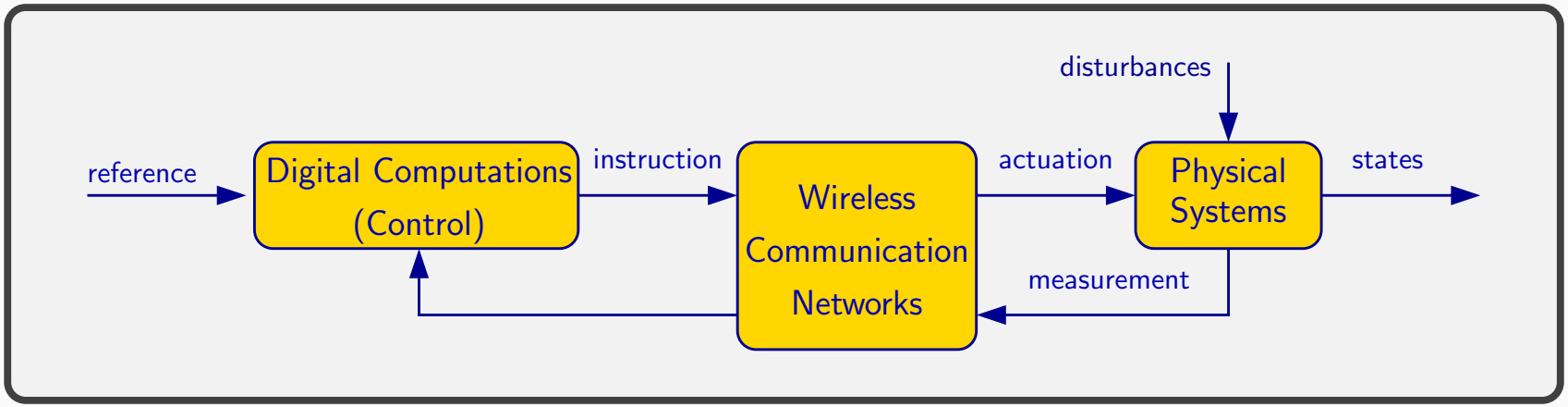
- **Optimal Filtering and Control**. Seek the best observer that estimates dynamical system state using noise statistics (e.g., the Kalman filter); Find the best controller that maximizes a performance metric (e.g., LQR methods, or model predictive controllers).
- **Robust and Adaptive Control**. Control with guaranteed performance for a range of plant parameters and disturbances; Controllers that adaptively adjust their strategies by learning the system parameters from the past input and output.
- **Nonlinear System Control**. Design intrinsically nonlinear controllers for nonlinear systems.

Feedback Systems for the Future of Our Society



- How do we control systems, whose sensors, controllers and actuators are **scattered in physically separated spaces**?
- How do we control systems with **humans in the loop** or human decision makers?

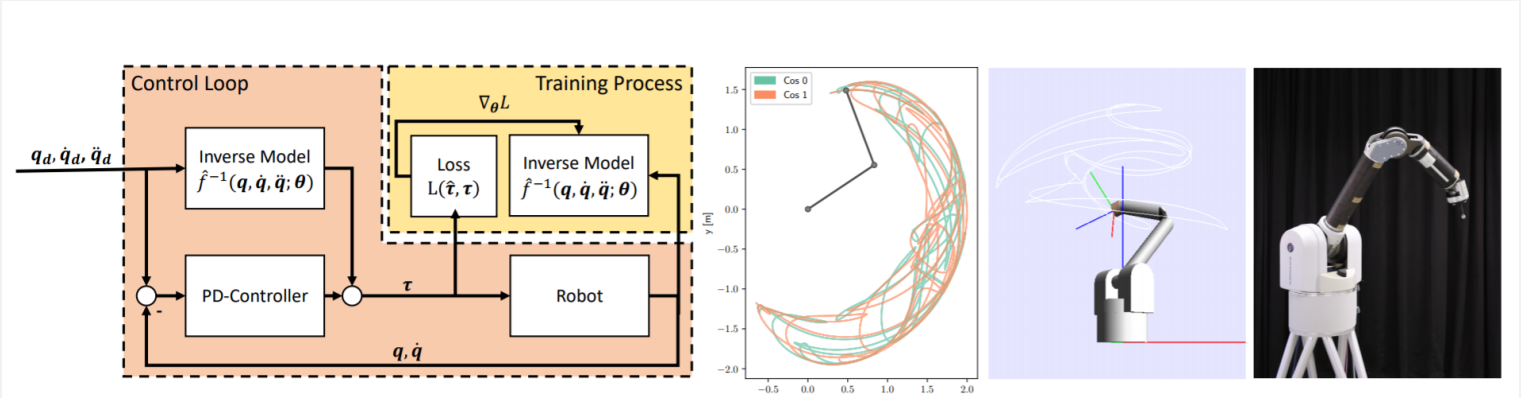
Networked Cyberphysical Systems (CPSs)



The use of networks for closed loop operation raises significant challenges:

1. **How do we deal with limited communication resources?**
   - wireless communication networks introduce random delays and data-dropouts
   - communication resources need to be shared
2. **How do we deal with security issues?**
   Due to feedback actuations, attacks may lead to performance degradation of the physical system and system failure.

Control and Machine Learning



Our Unit of Study has been based on **models** for dynamical systems.

Machine learning has been reshaping various areas of engineering. Data-driven, (partly) model-free control methodologies are being rapidly developed.

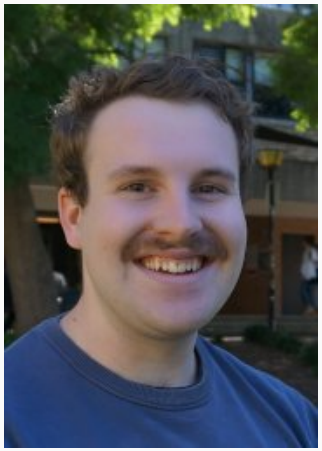
Our sincere thanks go to everyone in our class.

Without your support, feedback, advice, and hard work it is not possible for our course to be where we are now today.

Thank you!

Special Thanks

Johnny Cheng



Jack Naylor

Dechuan Liu

Raghav Mishra

Michael Somerfield

Darryl Tseng

Yurui Zhang

USS Survey

Unit of Study Survey is now open! It takes only 5 mins to complete the survey.

https://student-surveys.sydney.edu.au/students/

We value your feedback. The course will become better as a direct result of your inputs.