Control Systems I

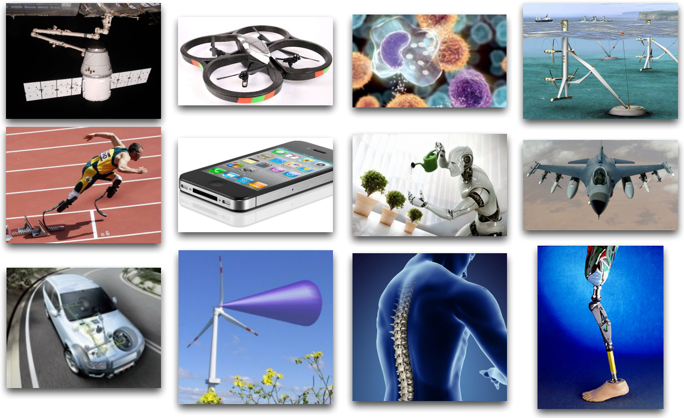
Introduction

Colin Jones

Laboratoire d'Automatique

Control Systems I

Prof. Colin Jones



Make things that **change with time** do what we want them to do

---

Make things that **change with time** do what we want them to do

Most engineered systems **require controllers to function**

Controllers can provide **optimal performance**

Analysis and understanding of dynamic systems

A controller is anything that **senses** the environment, takes **decisions**, and **modifies the environment** in order meet some **objective**.

Robot Quadrotors Perform James Bond Theme

GRASP Lab, University of Pennsylvania

Components of a Control System

Sensor Measure the world

Actuator Effect the world

System The object we're trying to control

Controller Takes decisions based on

- Measurements
- Knowledge of how the system works

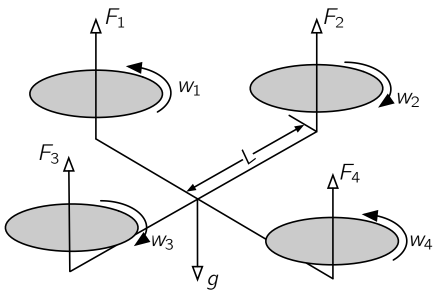
A controller is anything that **senses** the environment, takes **decisions**, and **modifies the environment** in order meet some **objective**.

Note: Controller doesn't have to be a 'computer', or an electronic circuit

Example: Autonomous Quadrocopter flight

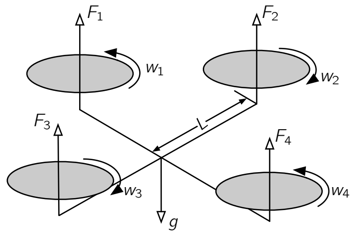
- Highly agile due to fast rotational dynamics
- High thrust-to-weight ratio allows for large translational accelerations
- Motion control by altering rotation rate and/or pitch of the rotors
- High thrust motors enable high performance control

How a Quad Works



- We can set the speed of the propellers (our inputs)
- Our goal is to control the pitch, roll and altitude

How a Quad Works



Force is quadratic in propeller speed:

$$F_i(t) = k_F w_i(t)^2$$

Moment is quadratic in prop speed:

$$M_i(t) = k_M w_i(t)^2$$

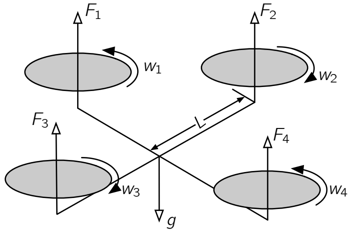
Vertical force: $F(t) = F_1(t) + F_2(t) + F_3(t) + F_4(t)$

Roll moment: $M_\alpha(t) = L(F_1(t) - F_4(t))$

Pitch moment: $M_\beta(t) = L(F_2(t) - F_3(t))$

Rotation: $M_\gamma(t) = M_1(t) + M_2(t) + M_3(t) + M_4(t)$

How a Quad Works



Force is quadratic in propeller speed:

$$F_i(t) = k_F w_i(t)^2$$

Moment is quadratic in prop speed:

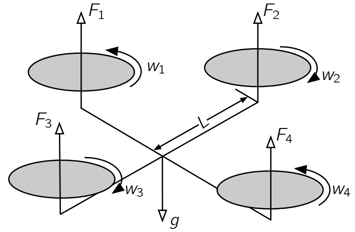
$$M_i(t) = k_M w_i(t)^2$$

Vertical force:
Roll moment:
Pitch moment:
Rotation:

$$\begin{pmatrix} F(t) \\ M_\alpha(t) \\ M_\beta(t) \\ M_\gamma(t) \end{pmatrix} = \begin{bmatrix} k_F & k_F & k_F & k_F \\ Lk_F & 0 & 0 & -Lk_F \\ 0 & Lk_F & -Lk_F & 0 \\ k_M & k_M & k_M & k_M \end{bmatrix} \begin{pmatrix} w_1(t)^2 \\ w_2(t)^2 \\ w_3(t)^2 \\ w_4(t)^2 \end{pmatrix}$$

- We have four degrees of freedom and four forces / moments
- Can set the forces / moments as we like - these are our inputs

Quad Control

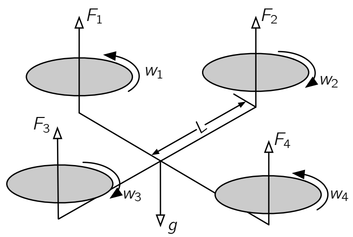


Altitude: $m\ddot{z}(t) = \underbrace{-mg}_{\text{Gravity}} + \underbrace{F(t)}_{\text{Thrust of propellers}}$

Hold altitude at z_c: $F(t) = K(z_c - z(t))$

Resulting system: $m\ddot{z}(t) = -mg + K(z_c - z(t))$

Quad Control



Roll and pitch:

$$I_\alpha \ddot{\alpha}(t) = M_\alpha(t)$$

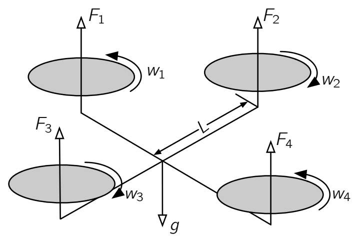
Hold attitude at α_c, β_c:

$$M_\alpha(t) = K_\alpha(\alpha_c - \alpha(t))$$

Resulting system

$$I_\alpha \ddot{\alpha}(t) = K_\alpha(\alpha_c - \alpha(t))$$

Quad Control



Yaw:

$$I_\gamma \ddot{\gamma}(t) = M_\gamma(t)$$

Keep yaw at zero:

$$M_\gamma(t) = -K_\gamma \gamma(t) - D_\gamma \dot{\gamma}(t)$$

Resulting system

$$I_\gamma \ddot{\gamma}(t) = -K_\gamma \gamma(t) - D_\gamma \dot{\gamma}(t)$$

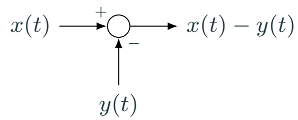
Example: Autonomous Quadrocopter flight

Demo movie

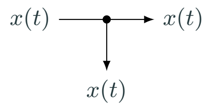
Lexus & Kmel robotics

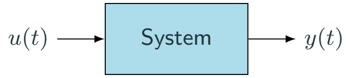
Block Diagrams - Basic Building Blocks

Summation



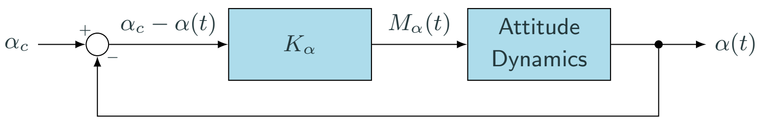
Bifurcation





Enforces a dynamic constraint between the output $y(t)$ and the input $u(t)$
e.g. $\ddot{y}(t) + \alpha\dot{y}(t) - \ddot{u}(t) + u(t) = 0$

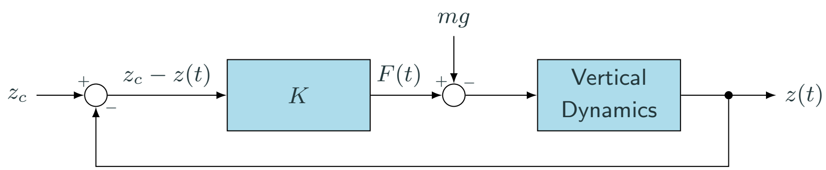
Block Diagram of Attitude Controllers



- Reference α_c
- Error $\alpha_c - \alpha(t)$
- Input $M_\alpha(t)$
- Output $\alpha(t)$
- Controller K_α
- System $I_\alpha\ddot{\alpha}(t) = M_\alpha(t)$

Goal: **Track** reference α_c

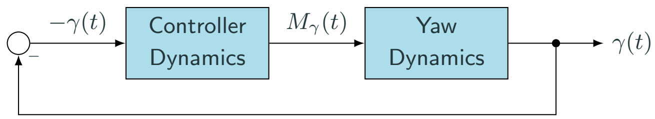
Block Diagram of Altitude Controller



- Disturbance g

Goal: **Track** reference z_c and **reject** disturbance mg

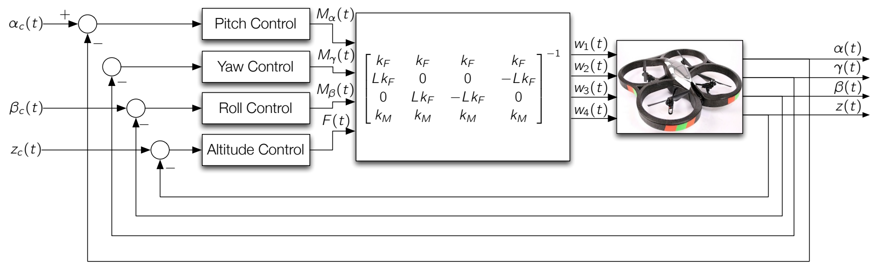
Block Diagram of Yaw Controller



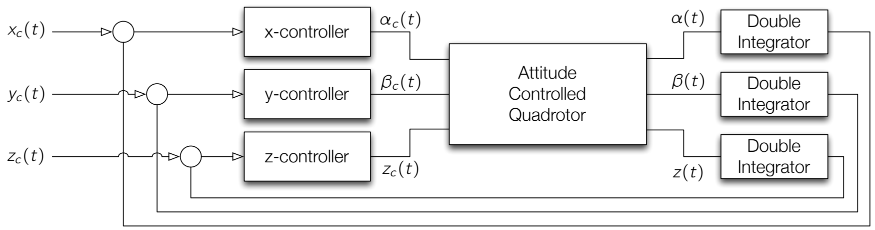
- Controller dynamics: $M_\gamma(t) = -K_\gamma \gamma(t) - D_\gamma \dot{\gamma}(t)$

Goal: **Regulate** the yaw

Cascade Control



Cascade Control



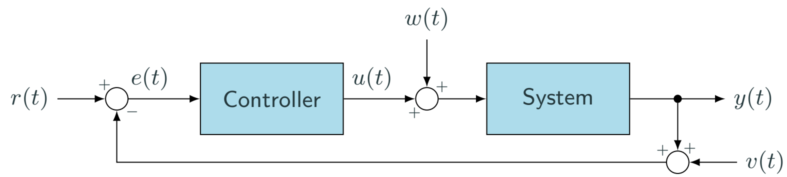
Possibly lots more loops

- Collision avoidance
- Trajectory planning
- Mission planning
- etc

Why?

- Inner loops make the system **predictable** and **simple**
- Conceptually simpler

Canonical Block Diagram

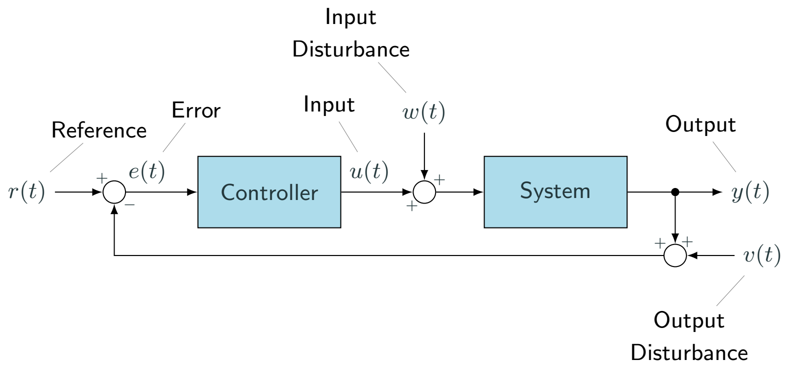


Goal: Make $y(t) = r(t)$, no matter what $w(t)$, or $v(t)$ are

If $r(t)$ is...

- zero, we're doing **regulation**
- time-varying, we're doing **servoing / tracking**

Canonical Block Diagram



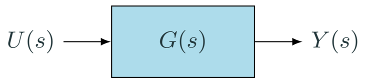
Goal: Make $y(t) = r(t)$, no matter what $w(t)$, or $v(t)$ are

If $r(t)$ is...

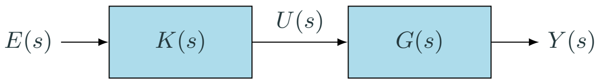
- zero, we're doing **regulation**
- time-varying, we're doing **servoing** / **tracking**

Nomenclature

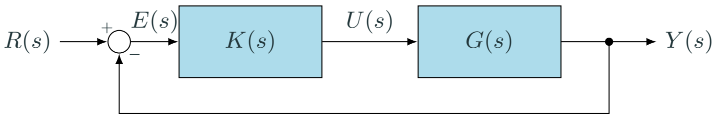
The **system**:



The **open-loop system** or **loop gain**:



The **closed-loop system**:



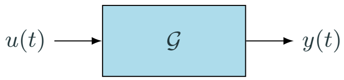
Quick Review of Systèmes Dynamique

More complete review on Moodle

What is a System?

A dynamic system transforms an input signal $u(t)$ into an output signal $y(t)$.

$$y = \mathcal{G}(u)$$



We care about LTI systems

Linear $\mathcal{G}(au_1 + bu_2) = a\mathcal{G}(u_1) + b\mathcal{G}(u_2)$

Causal $u(t) = 0$ for $t < 0$ implies $y(t) = 0$ for $t < 0$

Time-invariant $y(t) = \mathcal{G}(u(t))$ implies that $\mathcal{G}(u(t + T)) = y(t + T)$

Why are these types of systems important?

1. We can predict their behaviour from data easily

1. We can predict their behaviour from data easily

Impulse Response

The impulse response $g(t)$ is defined as the output of the system in response to a dirac delta function at time $t = 0$:

$$g(t) := \mathcal{G}(\delta(t))$$

1. We can predict their behaviour from data easily

Impulse Response

The impulse response $g(t)$ is defined as the output of the system in response to a dirac delta function at time $t = 0$:

$$g(t) := \mathcal{G}(\delta(t))$$

Theorem : Response of an LTI System

The output of an LTI system in response to an input signal $u(t)$ is

$$\mathcal{G}(u) = g * u$$

where $y = g * u$ if

$$y(t) = \int_0^t u(\tau)g(t-\tau)d\tau$$

1. We can predict their behaviour from data easily

Impulse Response

The impulse response $g(t)$ is defined as the output of the system in response to a dirac delta function at time $t = 0$:

$$g(t) := \mathcal{G}(\delta(t))$$

Theorem : Response of an LTI System

The output of an LTI system in response to an input signal $u(t)$ is

$$\mathcal{G}(u) = g * u$$

where $y = g * u$ if

$$y(t) = \int_0^t u(\tau)g(t-\tau)d\tau$$

If $\mathcal{G}$ is an LTI system, then the impulse response completely characterizes it.

1. We can predict their behaviour from data easily

Impulse Response

The impulse response $g(t)$ is defined as the output of the system in response to a dirac delta function at time $t = 0$:

$$g(t) := \mathcal{G}(\delta(t))$$

Theorem : Response of an LTI System

The output of an LTI system in response to an input signal $u(t)$ is

$$\mathcal{G}(u) = g * u$$

where $y = g * u$ if

$$y(t) = \int_0^t u(\tau)g(t-\tau)d\tau$$

If $\mathcal{G}$ is an LTI system, then the impulse response completely characterizes it.

Key limitation: Most systems have an infinitely-long impulse response.

Why are these types of systems important?

1. We can predict their behaviour from data easily
2. We can store and manipulate complex systems

2. We can store and manipulate complex systems

Transfer Function

The **transfer function** of a system is the Laplace transform of its impulse response.

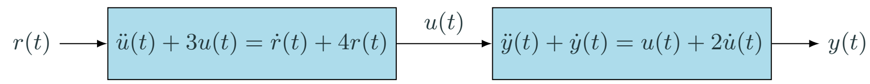
$$\mathcal{L}\{g(t)\} = G(s)$$

For LTI systems $G(s)$ is a rational polynomial function

The point: Convolution becomes multiplication

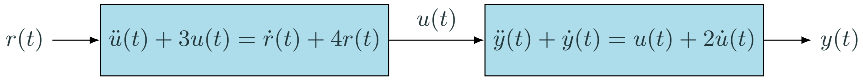
$$y = g * u \qquad \Leftrightarrow \qquad Y(s) = G(s)U(s)$$

Manipulation of Simple Block Diagrams



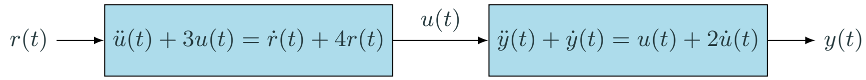
If we're given the reference function $r(t)$, what is $y(t)$?

Manipulation of Simple Block Diagrams



$$\ddot{u}(t) + 3u(t) = \dot{r}(t) + 4r(t) \quad \Rightarrow \quad s^2U(s) + 3U(s) = sR(s) + 4R(s)$$
$$\ddot{y}(t) + \dot{y}(t) = u(t) + 2\dot{u}(t) \quad \Rightarrow \quad s^2Y(s) + sY(s) = U(s) + 2sU(s)$$

Manipulation of Simple Block Diagrams

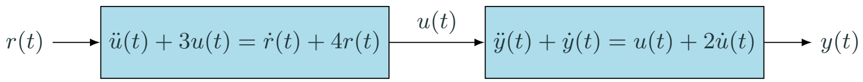


$$\ddot{u}(t) + 3u(t) = \dot{r}(t) + 4r(t) \quad \Rightarrow \quad s^2U(s) + 3U(s) = sR(s) + 4R(s)$$
$$\ddot{y}(t) + \dot{y}(t) = u(t) + 2\dot{u}(t) \quad \Rightarrow \quad s^2Y(s) + sY(s) = U(s) + 2sU(s)$$

Re-arranging gives:

$$U(s) = \frac{s+4}{s^2+3}R(s) \qquad Y(s) = \frac{1+2s}{s^2+s}U(s)$$

Manipulation of Simple Block Diagrams



$$\ddot{u}(t) + 3u(t) = \dot{r}(t) + 4r(t) \quad \Rightarrow \quad s^2U(s) + 3U(s) = sR(s) + 4R(s)$$
$$\ddot{y}(t) + \dot{y}(t) = u(t) + 2\dot{u}(t) \quad \Rightarrow \quad s^2Y(s) + sY(s) = U(s) + 2sU(s)$$

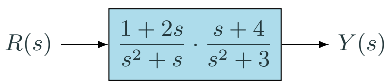
Re-arranging gives:

$$U(s) = \frac{s+4}{s^2+3}R(s) \qquad Y(s) = \frac{1+2s}{s^2+s}U(s)$$

... and we can compute the impact of $r(t)$ on $y(t)$

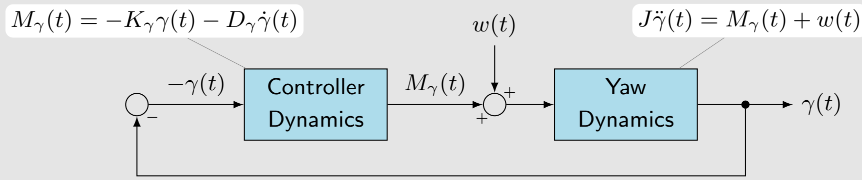
$$Y(s) = \frac{1+2s}{s^2+s} \cdot \frac{s+4}{s^2+3}R(s)$$

Series connection of blocks (convolution) becomes multiplication!



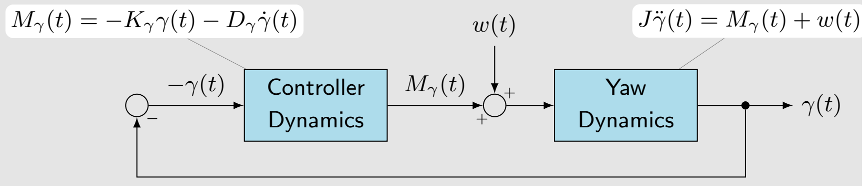
Example: System Response

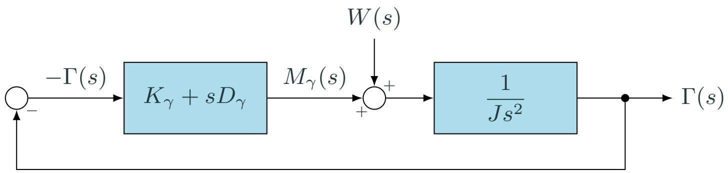
Compute response to a impulsive disturbance acting on the yaw system



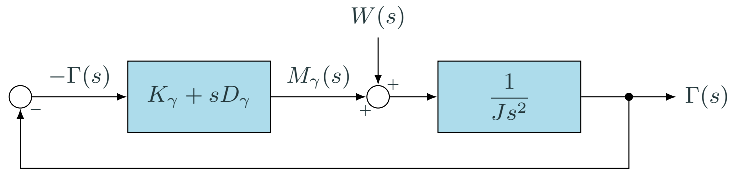
Example: System Response

Compute response to a impulsive disturbance acting on the yaw system

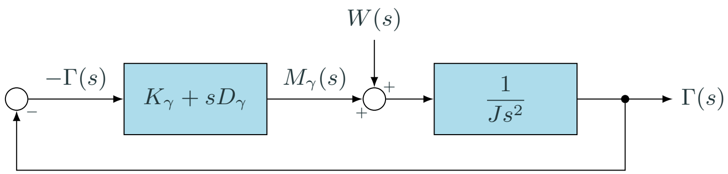




Example: System Response



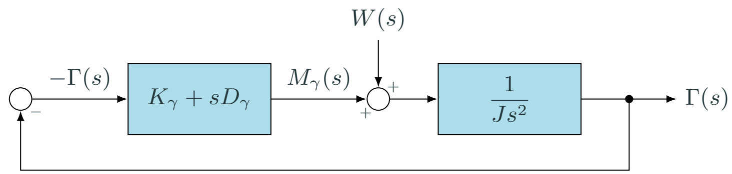
Example: System Response



Start at the output and work backwards against the arrows

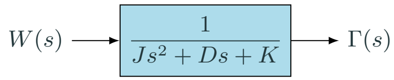
$$\Gamma = \frac{1}{Js^2}(W - (K_\gamma + sD_\gamma)\Gamma)$$
$$(Js^2 + sD_\gamma + K_\gamma)\Gamma = W$$

Example: System Response



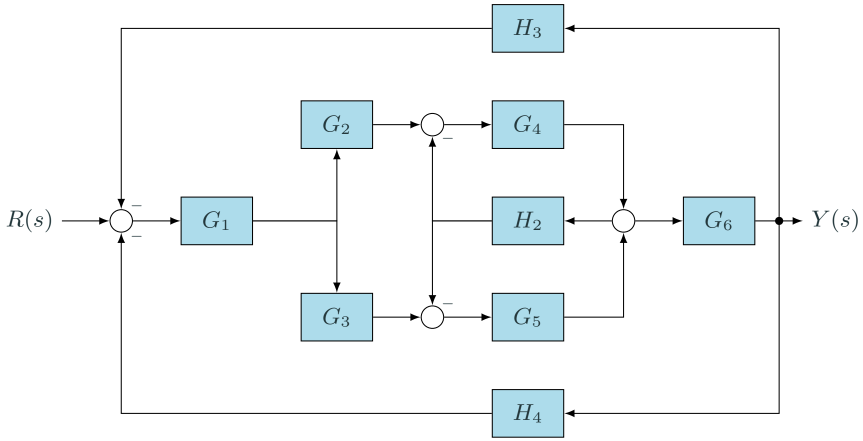
Start at the output and work backwards against the arrows

$$\Gamma = \frac{1}{Js^2}(W - (K_\gamma + sD_\gamma)\Gamma)$$
$$(Js^2 + sD_\gamma + K_\gamma)\Gamma = W$$

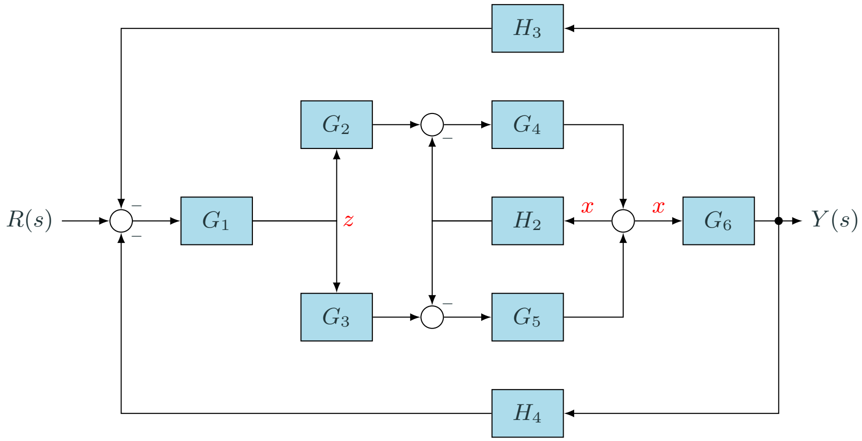


Where we recall that D sets the damping and K the response rate.

Example: Complex System



Example: Complex System



Add auxiliary variables for internal loops, and wherever convenient to simplify.

Example: Complex System

Start at the output and work back **against** the arrows.
A block is a multiplication, a summation is addition.

$$
\begin{aligned}
Y &= G_6 x \\
x &= G_4(G_2 z - H_2 x) + G_5(G_3 z - H_2 x) \\
z &= G_1(R - H_3 Y - H_4 Y)
\end{aligned}
$$

Solve for Y as a function of R

$$
\begin{aligned}
x &= (G_4G_2 + G_5G_3)z - (G_4H_2 + G_5H_2)x \\
(1 + G_4H_2 + G_5H_2)x &= (G_4G_2 + G_5G_3)z \\
x &= \frac{G_4G_2 + G_5G_3}{1 + G_4H_2 + G_5H_2} z \\
Y &= G_6 \frac{G_4G_2 + G_5G_3}{1 + G_4H_2 + G_5H_2} z
\end{aligned}
$$

Example: Complex System

$$
Y = \underbrace{G_6 \frac{G_4G_2 + G_5G_3}{1 + G_4H_2 + G_5H_2}}_{Q} z \qquad z = G_1 R - (H_3 + H_4) Y
$$

Solve to get the transfer function

$$
\frac{Y}{R} = \frac{QG_1}{1 + Q(H_3 + H_4)}
$$

If we want to do more algebra, we can eliminate Q

$$
\frac{Y}{R} = \frac{G_1G_2G_4G_6 + G_1G_3G_5G_6}{(G_4 + G_5)H_2 + G_2G_4G_6H_3 + G_2G_4G_6H_4 + G_3G_5G_6H_3 + G_3G_5G_6H_4 + 1}
$$

Why are these types of systems important?

1. We can predict their behaviour from data easily
2. We can store and manipulate complex systems
3. We can shape system behaviour

3. We can shape system behaviours

Time domain

- PID
- Model predictive control
- ...

3. We can shape system behaviours

Time domain

- PID
- Model predictive control
- ...

Frequency domain

- Loopshaping controllers
- $\mathcal{H}_\infty$ - robust optimal control
- ...

3. We can shape system behaviours

Time domain

- PID
- Model predictive control
- ...

Frequency domain

- Loopshaping controllers
- $\mathcal{H}_\infty$ - robust optimal control
- ...

Pole/zero domain

- Pole placement
- Linear quadratic regulation
- ...

3. We can shape system behaviours

Time domain

- PID
- Model predictive control
- ...

Frequency domain

- Loopshaping controllers
- $\mathcal{H}_\infty$ - robust optimal control
- ...

Pole/zero domain

- Pole placement
- Linear quadratic regulation
- ...

Many very well-established techniques that are proven and work well at large scales.

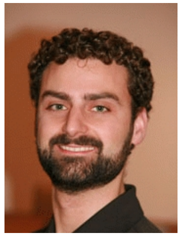
Key points to review

Please review:

- Computation of Laplace transforms
- Manipulation of block diagrams
- Inverse Laplace transforms
- System response to impulse, step, ramp, etc

Administration

Teachers

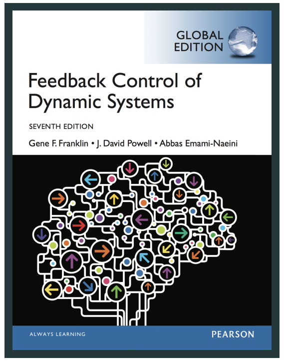


Professor
Colin Jones
Laboratoire d'Automatique
ME C2 405
colin.jones@epfl.ch

Travaux Pratique
Christophe Salzmann
Laboratoire d'Automatique
ME C2 426
christophe.salzmann@epfl.ch

Reference Material

We will mostly follow the textbook:



- The sections of the text that we are covering will appear on Moodle
- Lecture notes and pre-recorded videos are on Moodle

You are responsible for the material in the text **and** in the lecture notes

Activities

1. Lectures
   - Two hours per week
   - Lectures are not recorded, but high-quality pre-recordings are on Moodle
2. TPs
   - Seven TPs done via a MOOC interface driving a physical device
   - Can do the TPs in-person **or** remotely
3. Exercises
   - Written / computer exercises
   - 13 exercise sets

Detailed schedule on Moodle

How to Get Help

In person During lectures, or during afternoon exercise / TP sessions

Ed Discussion Please post your questions publicly - others will benefit!

Recorded videos Lectures have been pre-recorded and are available on Moodle.

Grading and Exams

100% Final exam

- One question from the TPs (MOOC) worth 20%
- Questions based on the lectures / exercises worth 80%

Examples: Other Varieties of Control

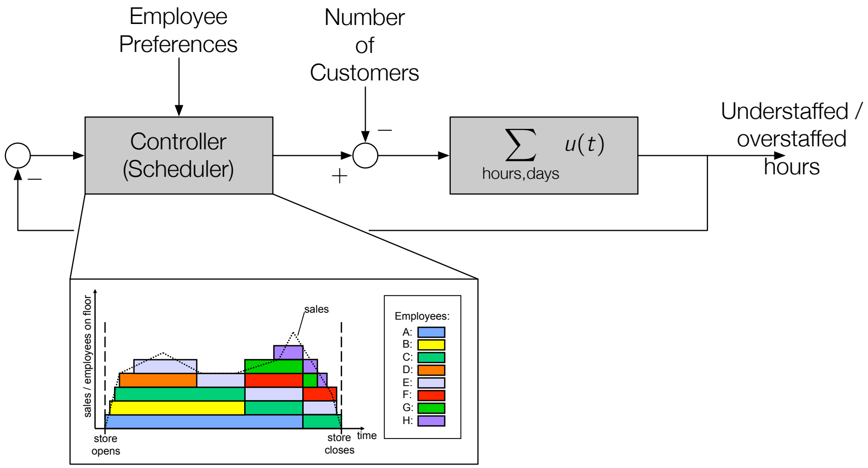
Employee Scheduling : The Challenge

Too few salespeople
=
Unhappy customers / less sales

Too many salespeople
=
Excessive wages

What can control do?

The Control Problem

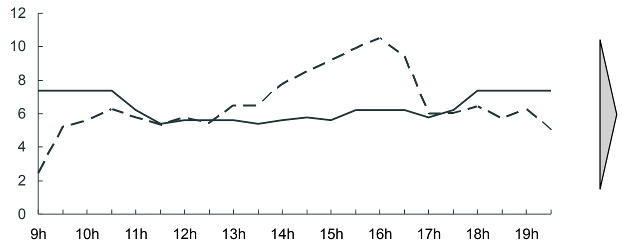


SW-

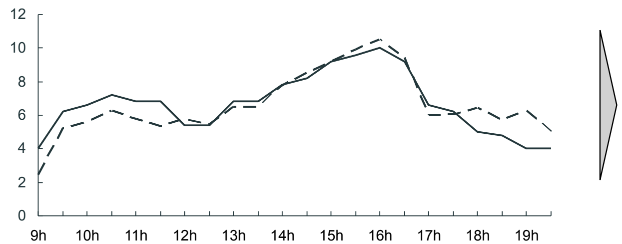
… ACHIEVES SAVINGS BY MATCHING RESOURCES TO DEMAND EXAMPLE

Average number of weekday staff*

— Scheduled
– – Demanded



- Original scheme (manually scheduled, unoptimized)
- FTE hours per week
  - –Excess scheduled: 65
  - –Unmet demanded: 85
  - –Mismatched: 150



- Optimized schedule** (automatically produced)
- FTE hours per week
  - –Excess scheduled: 15
  - –Unmet demanded: 39
  - –Mismatched: 54
- Estimated savings of up to 46 overtime hours per week (~10% of total) needed to meet expected demand

* For retail store with 14 staff (11.5 FTEs)
** Sample optimized schedule provided by Apex Optimization GmbH

1

Example: ‘Fulfilment Centers’

Kiva systems

Sold to Amazon in March, 2012 for $775m USD

Inerter

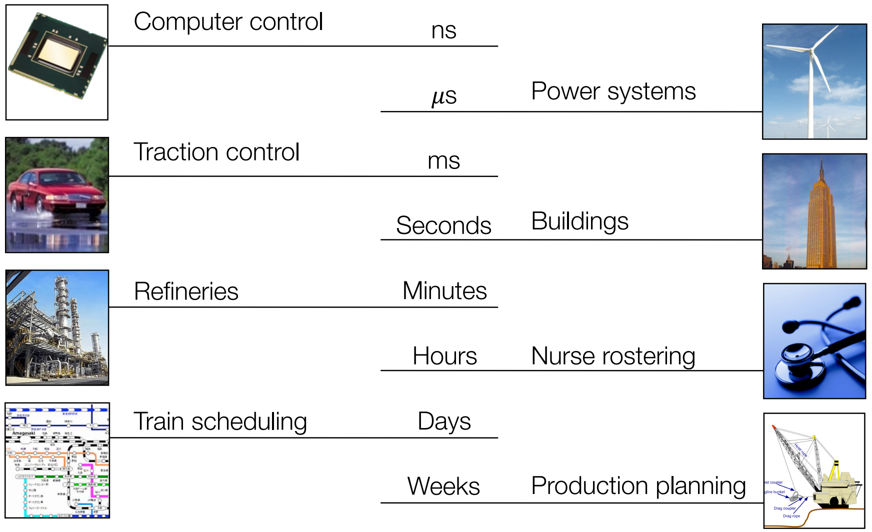
The Inerter in F1 Racing

Slides from Prof. Malcom Smith

Demand Response

Demand Response Slides

Control Applications at all Space and Time Scales



Summary

- Feedback control is everywhere
- It is used to:
  - Stabilize unstable systems
  - Make behaviors repeatable / predictable
  - Maximize performance
  - Understand what complex systems are doing