Questions and Answers Vol. 1

or

"Coffee makes the world go round."

P Pounds

16 April 2016

University of Queensland

But first…

Some house keeping

Calendar at a glance

Week	Dates	Lecture	Reviews	Demos	Assessment submissions
1	19/2 – 24/2	Introduction			
2	26/2 – 2/3	Principles of Mechatronic Systems design			Problem analysis
3	5/3 – 9/3	Professional Engineering Topics			
4	20/3 – 24/3	Introduction to Practical PCB Design	Progress review 1		
5	19/3 – 23/3	Your soldering is (probably) terrible			
6	26/3 – 29/3	Introduction to firmware design			
Break	30/4 – 13/4				
7	16/4 – 20/4		Progress seminar	25% demo	
8	23/4 – 27/4				
9	30/4 – 4/5			50% demo	
10	8/5 – 11/5	No lecture	Progress review		
11	14/5 – 18/5			75% demo	Preliminary report
12	21/5 – 25/5				
13	28/5 – 1/6	Closing lecture		Final testing	Final report and reflection

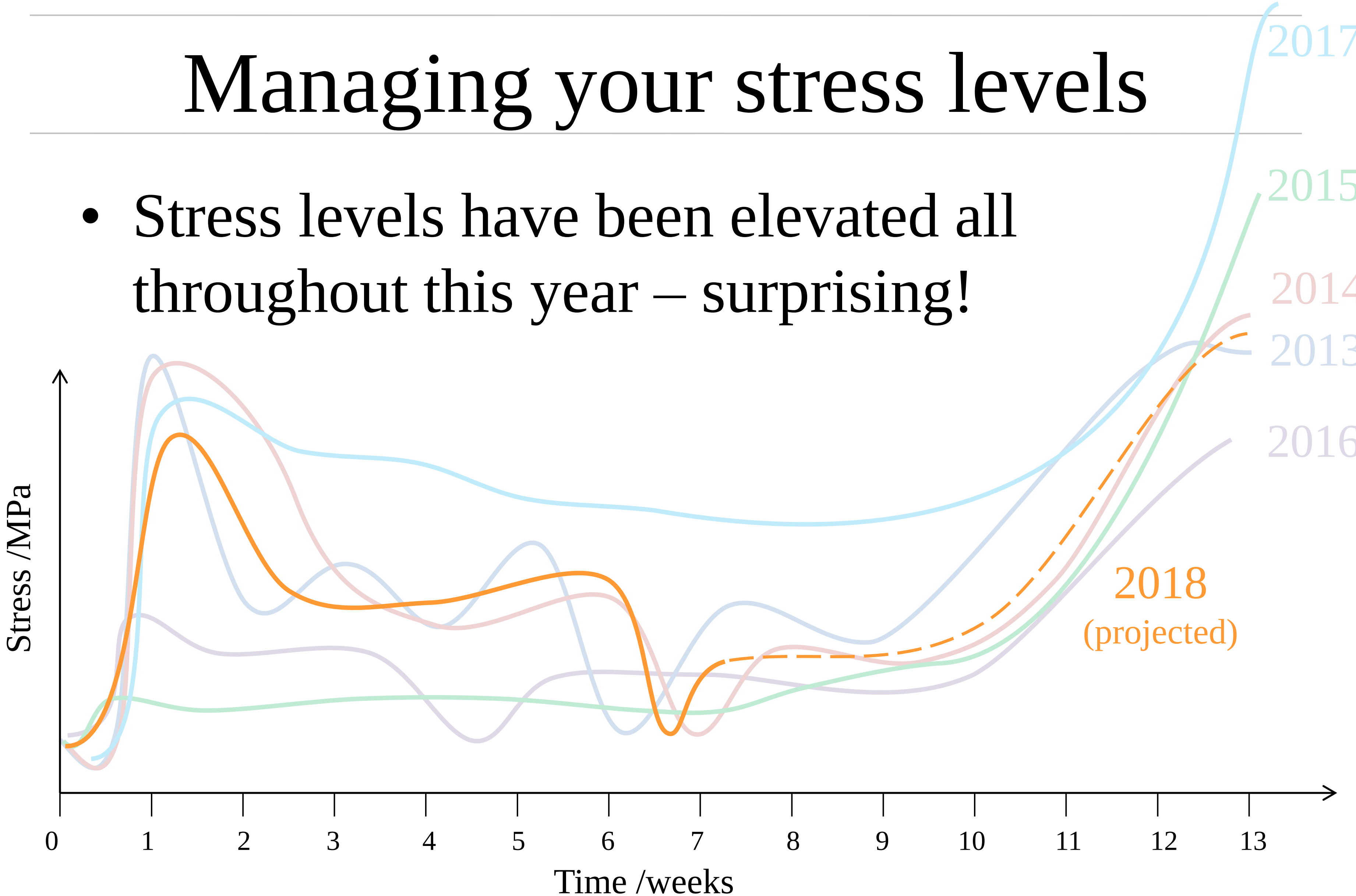
You are here →

Progress seminars

- Ongoing!
- Please make sure you fill out your PAFs and return them before you leave
- If you haven't yet returned your PAF sheet from Progress Review 1, then do so immediately – I AM COMING FOR YOU

Managing your stress levels

- Stress levels have been elevated all throughout this year – surprising!



Lecture nominations

- This week's nominated lecture topic was control theory
  - Hooray!
- What about next week?
  - Let's vote on that now!

FAQ Roundup

- **Can you look at my part design and give me feedback?**
  - Hell yeah! That's what I'm here for! Bring it!

- **How much will this part cost?**
  - Uhh... ask Jason Herriot?

- **What's going on with the incremental demos?**
  - Hey, let's talk about that...

Incremental Demo

- Runs next Wednesday in the usual timeslot
- A terrific opportunity to get closed-box testing experience (and possibly marks)
  – Even if your team isn't testing, you can really learn a lot from the experience of others!
- If you wanted to test, you should have sent me an email requesting a demo already…
  – But if I have one or two free slots, I might as a one-off, accept late requests…

Incremental Demo

- This demo is the 25% demo:
  - Total possible marks capped at 25%
  - Reduced testing set (no bonus marks)
  - Simplified scenario – no DSN block!
- This is as easy and simple as it gets
  - It's only harder from here!

A short primer on control theory

Nominated topic, ahoy!

A short primer on control theory

- When people ask me for a nominated topic, they're really asking "Hey, Doc Pounds, what do I need to know about *X* to solve this design problem"

- Well, alrighty then*

* No – I'm not going to give you the answers. I'm going to give you methods.

A short primer on control theory

How to solve a control problem in 5 easy steps

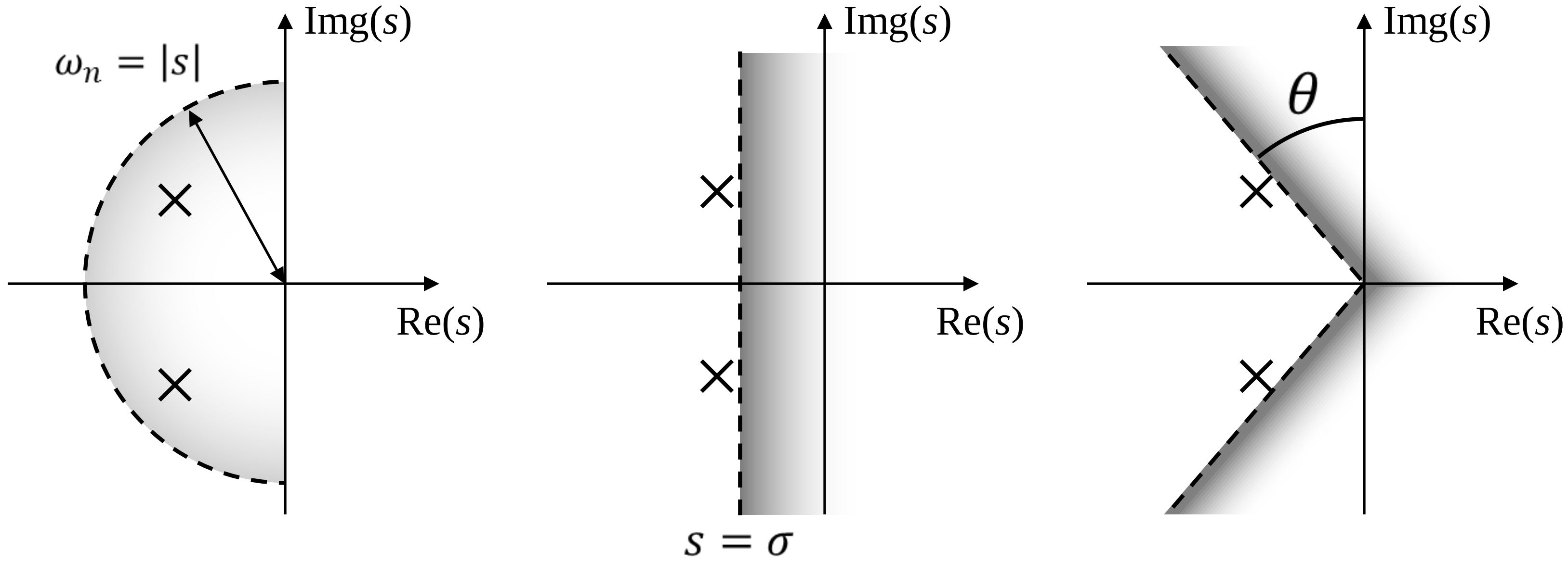
- Step 1. Determine the design specification
- Step 2. Identify the system
- Step 3. Assess stability/performance
- Step 4. Design a controller to meet the spec
- Step 5. Verify the design

Determine the specification

- How good do we have to be?
  – What does the client demand?
  – What is required to perform the task?
  – Slew rate? Rise time? Settling time?
- What are the likely disturbances like?
  – Magnitude and frequency?
  – Spectral characteristics?

Determine the specification

- Convert your time-domain requirements into Laplace or z-domain bounds*



$$|s| = \frac{1.8}{t_r}$$

$$s = \frac{4.6}{t_s}$$

$$\theta = \sin^{-1}\zeta$$

* These are the boundaries for the *s* plane; similar ones exist for discrete systems.

$\dot{x} = v$

Identify the system

- Start with a Newton-Euler (or Lagrangian) description of the system
  - What are the rigid body dynamics?
  - Are there actuator dynamics?
  - Capture all necessary system characteristics*

$$\dot{x} = v$$
$$\dot{v} = a$$
$$\dot{a} = f(x, v, u)$$

… etc

* How can you tell if a characteristics is necessary or not? It changes the output

Identify the system

- What are the input and output states?
  - What value do you want to regulate (angle?), and what parameters (voltage? current?) do you actually have direct control over?
- Compute Laplace/discrete transfer function (if SISO) or the transition matrix (if MIMO)

$$\frac{x(s)}{u(s)} = \frac{H_n(s)}{H_d(s)}$$

Assess system stability

- Look at the denominator's poles
  - If all poles are in the left-half of the *s* plane, then the system is stable – if not, then unstable
  - Pole positions also indicate system response
  - Stability is the very first thing you need to guarantee in your system!

$$\frac{x(s)}{u(s)} = \frac{H_n(s)}{H_d(s)}$$

Design a controller

- The goal is to move the system poles into the regions of the *s* plane that provide the desired performance
  - … but how??

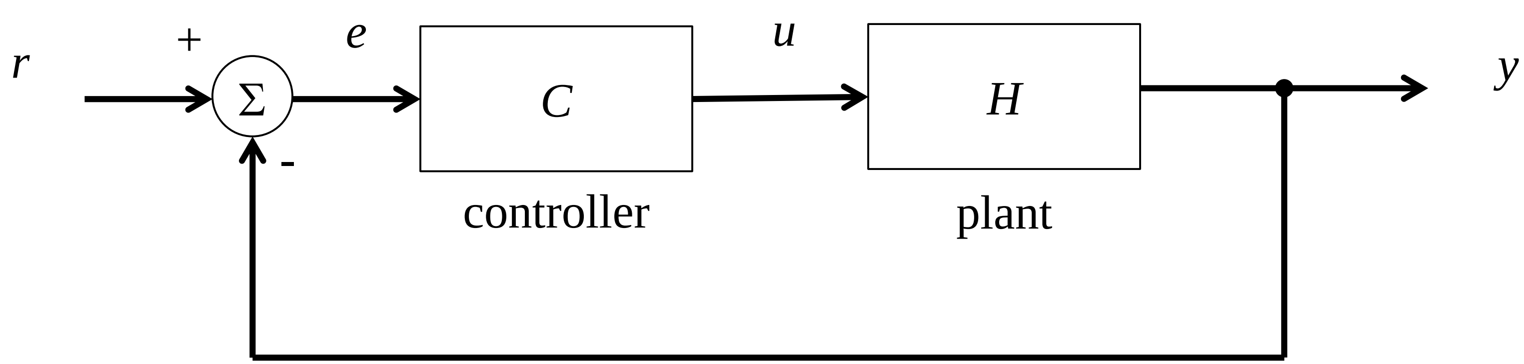
- This is where control design comes in…

Quick refresher: the root locus

- The transfer function for a closed-loop system can be easily calculated:

$$y = CH(r - y)$$
$$y + CHy = CHr$$
$$\therefore \frac{y}{r} = \frac{CH}{1 + CH}$$

Unlike previously, the next few slides use *y* instead of *x*, *r* instead of *u*, and *u* instead of something else entirely, because I already had them and was too darn ~~lazy~~ tired to change them. Just remember:
y = x; r = u; u = ??
It's ok, don't be scared!



Quick refresher: the root locus

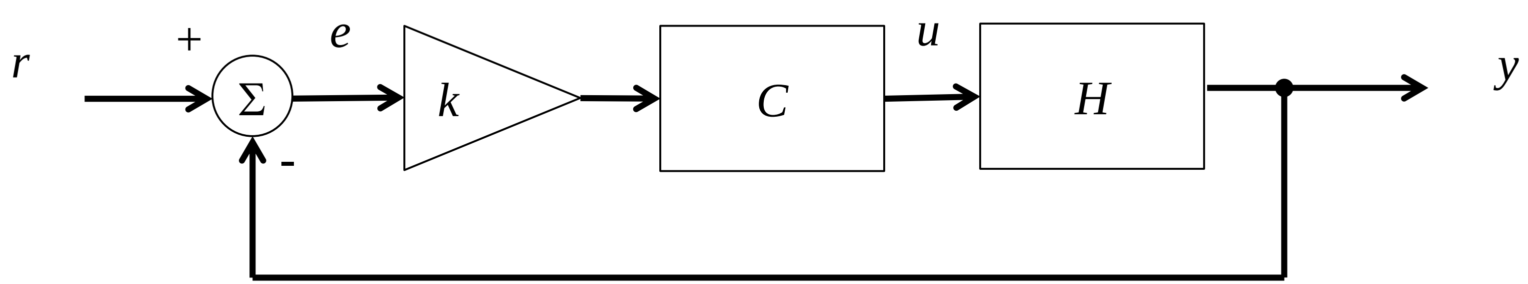
- We often care about the effect of increasing gain of a control compensator design:

$$\frac{y}{r} = \frac{kCH}{1 + kCH}$$

Multiplying by denominator:

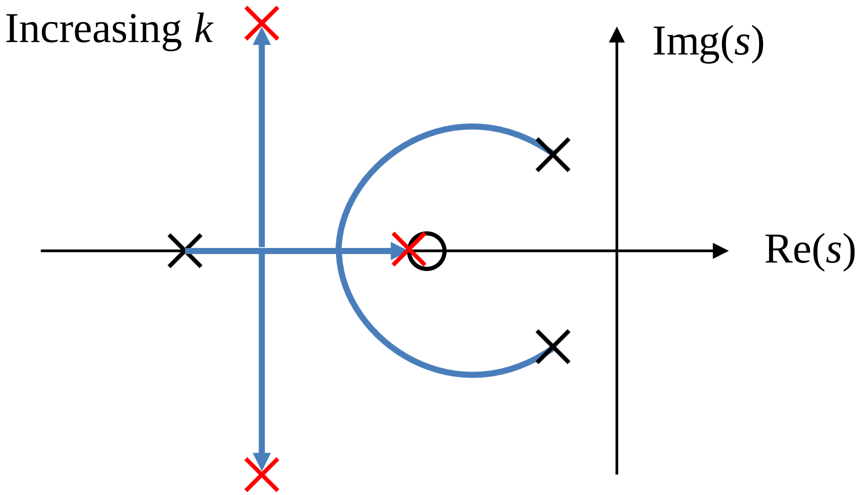
$$\frac{y}{r} = \frac{kC_nH_n}{C_dH_d + kCnHn}$$

characteristic polynomial



Quick refresher: the root locus

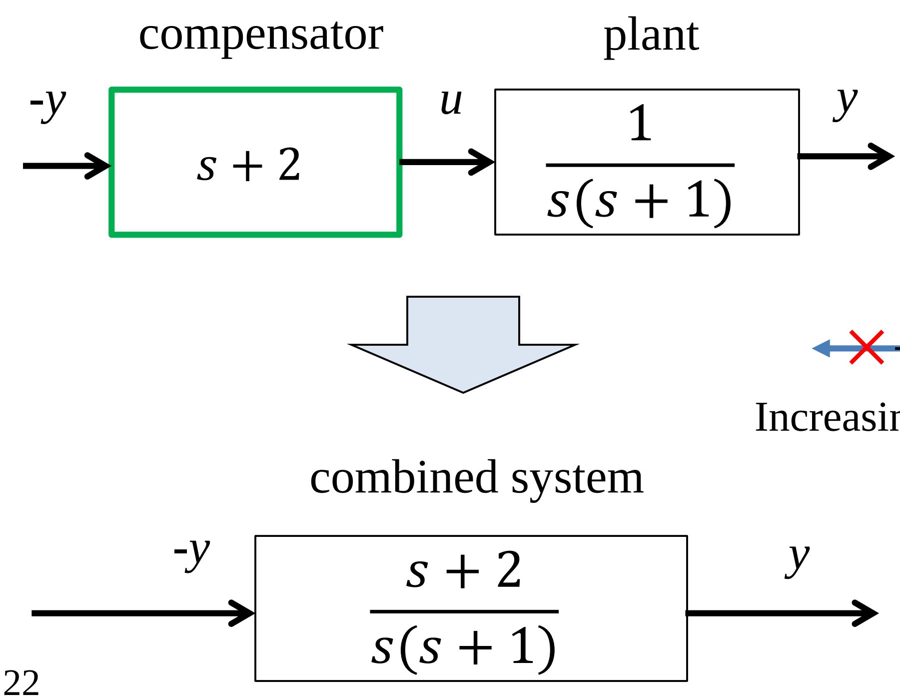
- Pole positions change with increasing gain
  - The trajectory of poles on the pole-zero plot with changing *k* is called the "root locus"
  - This is sometimes quite complex

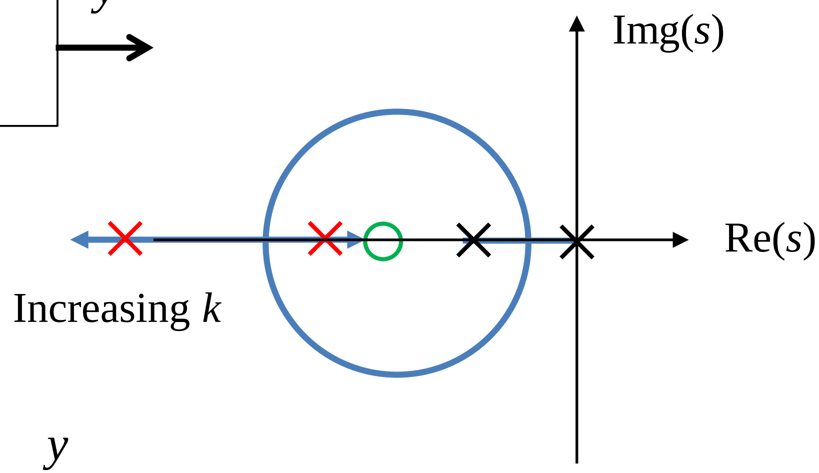


(In practice you'd plot these with computers)

Dynamic compensation

- We can do more than just apply gain!
  - We can add dynamics into the controller that alter the open-loop response

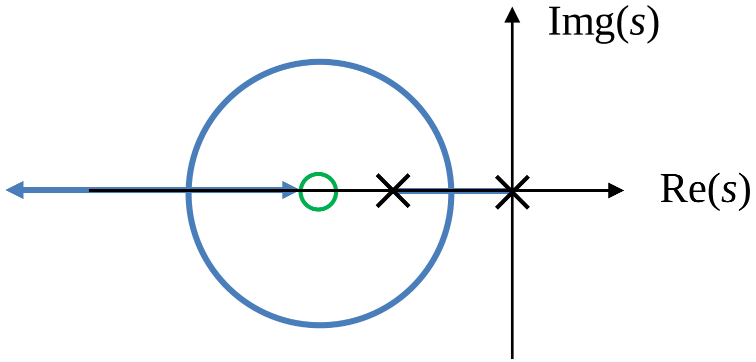




But what dynamics to add?

- Recognise the following:
  - A root locus starts at poles, terminates at zeros

    "Holes eat poles"
  - Closely matched pole and zero dynamics cancel
  - The locus is on the real axis to the left of an odd number of poles (treat zeros as 'negative' poles)



Some standard approaches

- Control engineers have developed time-tested strategies for building compensators
- Three classical control structures:
  - Lead
  - Lag
  - Proportional-Integral-Derivative (PID)

    (and its variations: P, I, PI, PD)

How do they work?

Lead/lag compensation

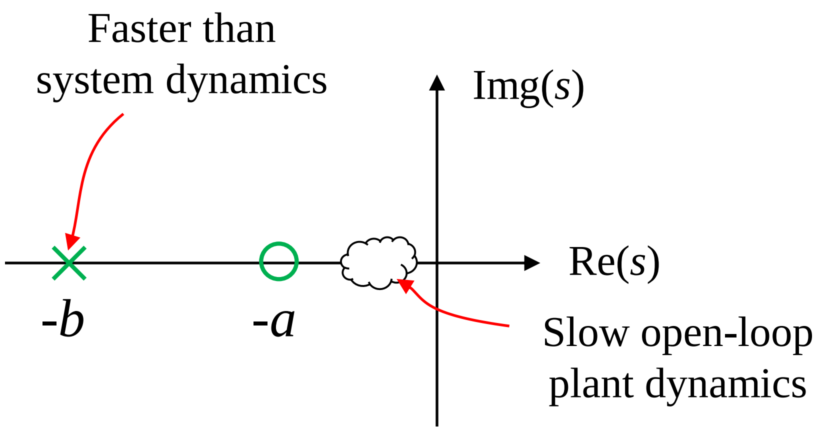
- Serve different purposes, but have a similar dynamic structure:

$$D(s) = \frac{s+a}{s+b}$$

Note:

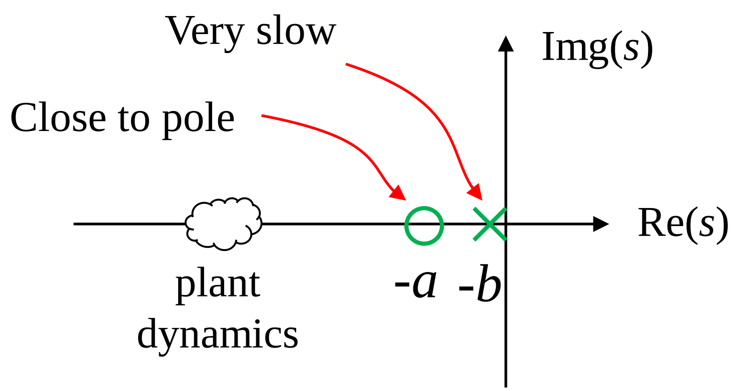
Lead-lag compensators come from the days when control engineers cared about constructing controllers from networks of op amps using frequency-phase methods. These days pretty much everybody uses PID, but you should at least know what the heck they are in case someone asks.

Lead compensation: $a < b$



- Acts to decrease rise-time and overshoot
  - Zero draws poles to the left; adds phase-lead
  - Pole decreases noise
- Set a near desired ω_n; set b at ~3 to 20x a

Lag compensation: $a > b$



- Improves steady-state tracking
  - Near pole-zero cancellation; adds phase-lag
  - Doesn’t break dynamic response (too much)
- Set b near origin; set a at ~3 to 10x b

PID – the Good Stuff

- Proportional-Integral-Derivative control is the control engineer's hammer*
  - For P,PI,PD, etc. just remove one or more terms

$$C(s) = k\left(1 + \frac{1}{\tau_i s} + \tau_d s\right)$$

Proportional
Integral
Derivative

*Everything is a nail

PID – the Good Stuff

- PID control performance is driven by three parameters:
  - k: system gain
  - τ_i: integral time-constant
  - τ_d: derivative time-constant

You're already familiar with the effect of gain.

What about the other two?

Integral

- Integral applies control action based on accumulated output error
  - Almost always found with P control
- Increase dynamic order of signal tracking
  - Step disturbance steady-state error goes to zero
  - Ramp disturbance steady-state error goes to a constant offset

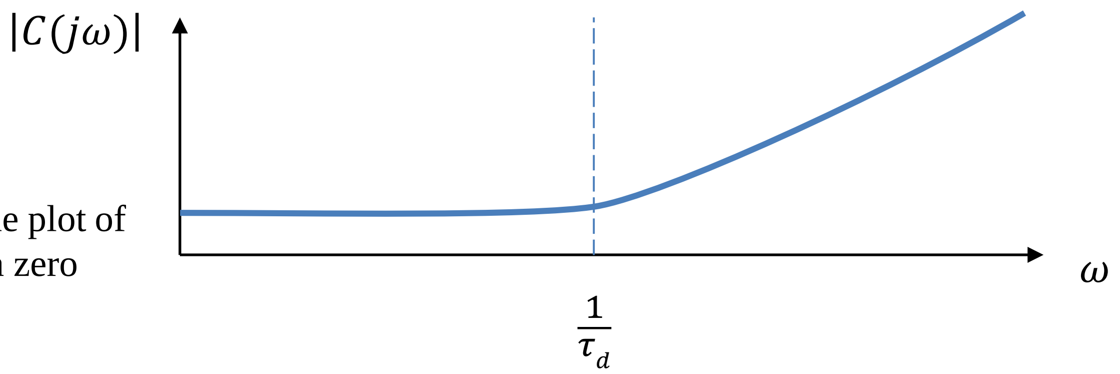
Derivative

- Derivative uses the rate of change of the error signal to anticipate control action
  - Increases system damping (when done right)
  - Can be thought of as ‘leading’ the output error, applying correction predictively
  - Almost always found with P control*

*What kind of system do you have if you use D, but don’t care about position? Is it the same as P control in velocity space?

Derivative

- It is easy to see that PD control simply adds a zero at $s = -\frac{1}{\tau_d}$ with expected results
  - Decreases dynamic order of the system by 1
  - Absorbs a pole as $k \rightarrow \infty$
- Not all roses, though: derivative operators are sensitive to high-frequency noise



Bode plot of a zero

PID

- Collectively, PID provides two zeros plus a pole at the origin
  - Zeros provide phase lead
  - Pole provides steady-state tracking
  - Easy to implement in microprocessors
- Many tools exist for optimally tuning PID
  - Zeigler-Nichols
  - Cohen-Coon
  - Automatic software processes

Be alert

- If gains and time-constants are chosen poorly, all of these compensators can induce oscillation or instability.

- However, when used properly, PID can stabilise even very complex unstable third-order systems

Verify the design

- Easy way to start is to throw it into Matlab
  - Show that it meets the spec in realistic scenarios
  - Try `sisotool` and `rltool` to get started
  - Excellent primers online (and in the ELEC3004 course notes) on how to drive Matlab controls
- Eventually, you need to test it on the bench
  - Showing it works "for real" on a simulated-mass test system is an excellent way of getting kudos

Need more?

- Still hankering for extra controls stuff, or more detail on these topics? Let me know!
  - I’ve got whole lectures on digital controls that I’m happy to send you

- Still confused? I love controls – send me an email and I’ll do my best!
  - I’m also always happy to help out with non-controls stuff, too! ☺

Questions

Tune-in next time for…

Questions and Answers Vol. 2

or

“By the bean alone do you set your mind in motion”

Fun fact: The world’s most expensive coffee – Kopi Luwak – is made from coffee beans that have passed through the digestive tract of a civet. Almost 90% of kopi luwak advertised for sale is fake.

Dr. Pounds' amazing scan-u-tron