http://elec3004.com

Servoregulation/PID

ELEC 3004: **Systems**: Signals & Controls

Dr. Surya Singh

Lecture 17

(with material from FPW + Åström and Hägglund)

elec3004@itee.uq.edu.au

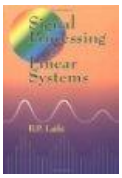
http://robotics.itee.uq.edu.au/~elec3004/

May 9, 2016

Lecture Schedule:

Week	Date	Lecture Title
1	29-Feb	Introduction
	3-Mar	Systems Overview
2	7-Mar	Systems as Maps & Signals as Vectors
	10-Mar	Data Acquisition & Sampling
3	14-Mar	Sampling Theory
	17-Mar	Antialiasing Filters
4	21-Mar	Discrete System Analysis
	24-Mar	Convolution Review
	28-Mar	Holiday
	31-Mar	
5	4-Apr	Frequency Response & Filter Analysis
	7-Apr	Filters
6	11-Apr	Digital Filters
	14-Apr	Digital Filters
7	18-Apr	Digital Windows
	21-Apr	FFT
8	25-Apr	Holiday
	28-Apr	Introduction to Feedback Control
9	3-May	Holiday
	5-May	Feedback Control & Regulation
10	9-May	Servoregulation/PID
	12-May	Introduction to (Digital) Control
11	16-May	Digital Control Design
	19-May	Stability
12	23-May	Digital Control Systems: Shaping the Dynamic Response & Estimation
	26-May	Applications in Industry
13	30-May	System Identification & Information Theory
	2-Jun	Summary and Course Review

Follow Along Reading:



B. P. Lathi
Signal processing and linear systems
1998
TK5102.9.L38 1998

G. Franklin,
J. Powell,
M. Workman
Digital Control of Dynamic Systems
1990

TJ216.F72 1990
[Available as UQ Ebook]

Today

➔P - I - D

- FPW
  - Chapter 4: Discrete Equivalents to Continuous
  - Transfer Functions: The Digital Filter

- FPW
  - Chapter 5: Design of Digital Control Systems Using Transform Techniques

Next Time

Lead/Lag

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

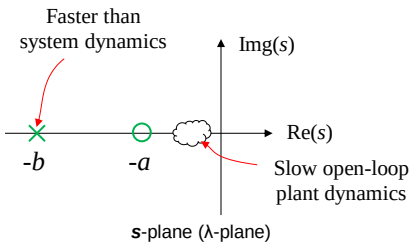
- Serve different purposes, but have a similar dynamic structure:

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

Note:

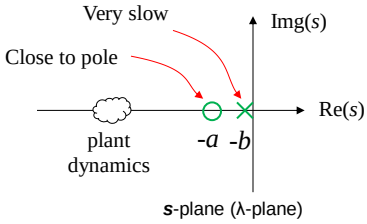
- Lead-lag compensators come from the days when control engineers cared about constructing controllers from networks of op amps using frequency-phase methods.

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
(Intro)

Proportional Control

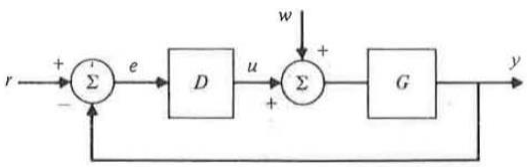
A discrete implementation of proportional control is identical to continuous; that is, where the continuous is

$$u(t) = K_p e(t) \quad \Rightarrow \quad D(s) = K_p,$$

the discrete is

$$u(k) = K_p e(k) \quad \Rightarrow \quad \boxed{D(z) = K_p}$$

where $e(t)$ is the error signal as shown in Fig 5.2.



Derivative Control

For continuous systems, derivative or rate control has the form

$$u(t) = K_p T_D \dot{e}(t) \quad \Rightarrow \quad D(s) = K_p T_D s$$

where T_D is called the *derivative time*. Differentiation can be approximated in the discrete domain as the first difference, that is,

$$u(k) = K_p T_D \frac{(e(k) - e(k-1))}{T} \quad \Rightarrow \quad \boxed{D(z) = K_p T_D \frac{1 - z^{-1}}{T} = K_p T_D \frac{z-1}{Tz.}}$$

In many designs, the compensation is a sum of proportional and derivative control (or PD control). In this case, we have

$$D(z) = K_p \left(1 + \frac{T_D(z-1)}{Tz}\right).$$

or, equivalently,

$$\boxed{D(z) = K \frac{z - \alpha}{z}}$$

Derivative Control [2]

- Similar to the lead compensators
  - The difference is that the pole is at $z = 0$

    [Whereas the pole has been placed at various locations along the z-plane real axis for the previous designs.]

- In the continuous case:
  - pure derivative control represents the ideal situation in that there is no destabilizing phase lag from the differentiation
  - the pole is at $s = -\infty$

- In the discrete case:
  - z=0
  - However this has phase lag because of the necessity to wait for one cycle in order to compute the first difference

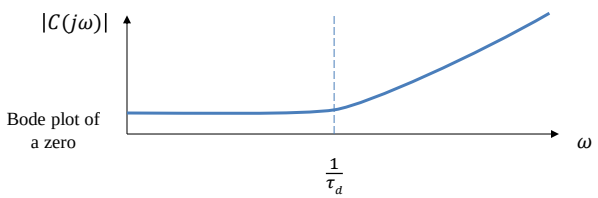
Derivative

- 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*

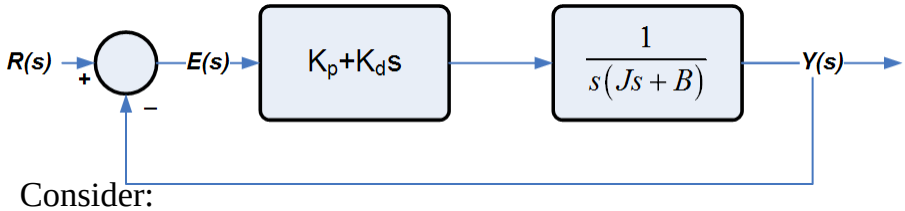
**What kind of system do you have if you use D, but don't care about position?* Is it the same as P control in velocity space?

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 $k \to \infty$
- Not all roses, though: derivative operators are sensitive to high-frequency noise



PD for 2nd Order Systems



- Consider:

$$\frac{Y(s)}{R(s)} = \frac{(K_P + K_D s)}{Js^2 + (B + K_D)s + K_P}$$

- Steady-state error: $e_{ss} = \frac{B}{K_P}$
- Characteristic equation: $Js^2 + (B + K_D)s + K_P = 0$
- Damping Ratio: $\zeta = \frac{B+K_D}{2\sqrt{K_P J}}$

➔It is possible to make e_{ss} and overshoot small (↓) by making B small (↓), K_P large ↑, K_D such that ζ:between [0.4 – 0.7]

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

Let's try it!

Integral Control

For continuous systems, we integrate the error to arrive at the control,

$$u(t) = \frac{K_p}{T_I}\int_{t_o}^{t} e(t)dt \;\Rightarrow\; D(s) = \frac{K_p}{T_I s},$$

where T_I is called the *integral*, or *reset time*. The discrete equivalent is to sum all previous errors, yielding

$$u(k) = u(k-1) + \frac{K_p T}{T_I}e(k) \;\Rightarrow\; \boxed{D(z) = \frac{K_p T}{T_I(1-z^{-1})} = \frac{K_p T z}{T_I(z-1)}.} \quad (5.60)$$

Just as for continuous systems, the primary reason for integral control is to reduce or eliminate steady-state errors, but this typically occurs at the cost of reduced stability.

Integral: P Control only

- Consider a first order system with a constant load disturbance, w; (recall as $t \to \infty$, $s \to 0$)

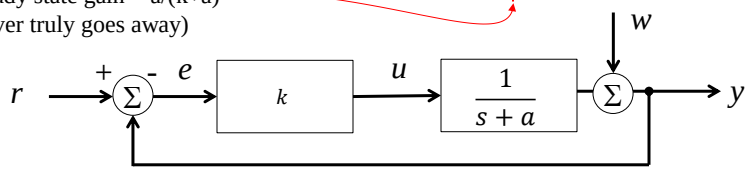
$$y = k\frac{1}{s+a}(r-y) + w$$

$$(s+a)y = k\,(r-y) + (s+a)w$$

$$(s+k+a)y = kr + (s+a)w$$

$$y = \frac{k}{s+k+a}r + \frac{(s+a)}{s+k+a}w$$

Steady state gain = a/(k+a)
(never truly goes away)



Now with added integral action

$$y = k\left(1 + \frac{1}{\tau_i s}\right)\frac{1}{s+a}(r - y) + w$$

$$y = k\frac{s + \tau_i^{-1}}{s}\frac{1}{s+a}(r - y) + w$$

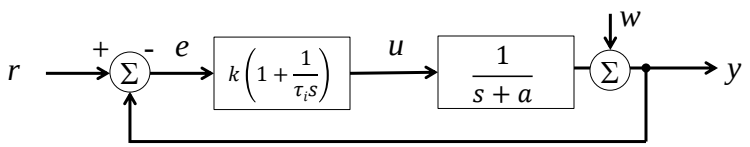
Same dynamics

$$s(s+a)y = k(s + \tau_i^{-1})(r - y) + s(s+a)w$$

$$(s^2 + (k+a)s + \tau_i^{-1})y = k(s + \tau_i^{-1})r + s(s+a)w$$

$$y = \frac{k(s + \tau_i^{-1})}{(s^2 + (k+a)s + \tau_i^{-1})}r + \frac{s(s+a)}{k(s + \tau_i^{-1})}w$$

Must go to zero for constant w!



PID – Control for the PID-dly minded

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

$$C(s) = k\left(1 + \frac{1}{\tau i s} + \tau d s\right)$$

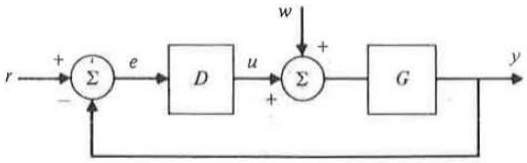
Proportional
Integral
Derivative

*Everything is a nail. That's why it's called "Bang-Bang" Control ☺

PID

- Three basic types of control:
  - Proportional
  - Integral, and
  - Derivative

- The next step up from lead compensation
  - Essentially a combination of proportional and derivative control



PID Control

$$D(z) = K_p\left(1 + \frac{Tz}{T_I(z-1)} + \frac{T_D(z-1)}{Tz}\right).$$

The user simply has to determine the best values of

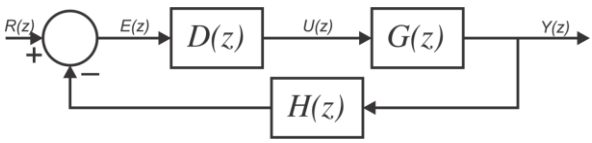
- K_p
- T_D and
- T_I

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

PID as Difference Equation

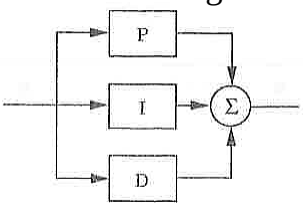


$$\frac{U(z)}{E(z)} = D(z) = K_p + K_i\left(\frac{Tz}{z-1}\right) + K_d\left(\frac{z-1}{Tz}\right)$$

$$\boxed{u(k) = \left[K_p + K_iT + \left(\frac{K_d}{T}\right)\right]\cdot e(k) - [K_dT]\cdot e(k-1) + [K_i]\cdot u(k-1)}$$

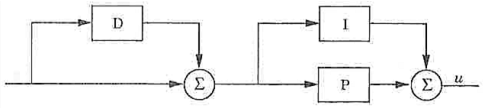
PID Implementation

- Non-Interacting



$$C(s) = K\left(1 + \frac{1}{sT_i} + sT_d\right)$$

- Interacting Form



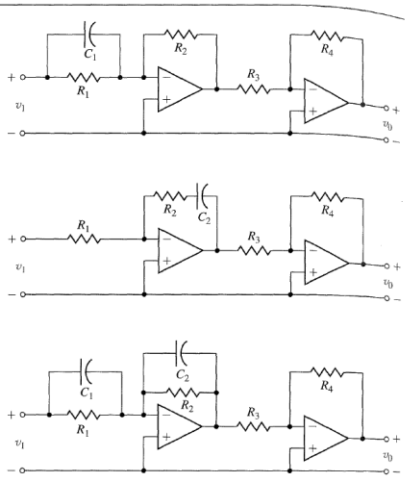
$$C'(s) = K\left(1 + \frac{1}{sT_i}\right)(1 + sT_d)$$

- Note: Different K,T_i and T_d

Operational Amplifier Circuits for Compensators

Type of Controller	$G_c(s) = \frac{V_o(s)}{V_i(s)}$
PD	$G_c = \frac{R_4R_2}{R_3R_1}(R_1C_1s + 1)$
PI	$G_c = \frac{R_4R_2(R_2C_2s + 1)}{R_3R_1(R_2C_2s)}$
Lead or lag	$G_c = \frac{R_4R_2(R_1C_1s + 1)}{R_3R_1(R_2C_2s + 1)}$

Lead if $R_1C_1 > R_2C_2$
Lag if $R_1C_1 < R_2C_2$



- (Yet Another Way to See PID)

Source: Dorf & Bishop, *Modern Control Systems*, p. 828

PID Algorithm (in various domains):

FPW § 5.8.4 [p.224]

- PID Algorithm (in Z-Domain):

$$D(z) = K_p\left(1 + \frac{Tz}{T_I(z-1)} + \frac{T_D(z-1)}{Tz}\right)$$

- As Difference equation:

$$u(t_k) = u(t_{k-1}) + K_p\left[\left(1 + \frac{\Delta t}{T_i} + \frac{T_d}{\Delta t}\right)e(t_k) + \left(-1 - \frac{2T_d}{\Delta t}\right)e(t_{k-1}) + \frac{T_d}{\Delta t}e(t_{k-2})\right]$$

- Pseudocode [Source: Wikipedia]:

```
previous_error = 0, integral = 0
start:
  error = setpoint - measured_value
  integral = integral + error*dt
  derivative = (error - previous_error)/dt
  output = Kp*error + Ki*integral + Kd*derivative
  previous_error = error
  wait(dt)
  goto start
```

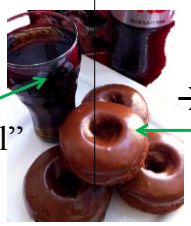

Another way to see P I|D

- Derivative

  D provides:
  - High sensitivity
  - Responds to change
  - Adds "damping" & ∴ permits larger K_P
  - Noise sensitive
  - Not used alone (∵ its on rate change of error – by itself it wouldn't get there)

→ "**D**iet Coke of control"

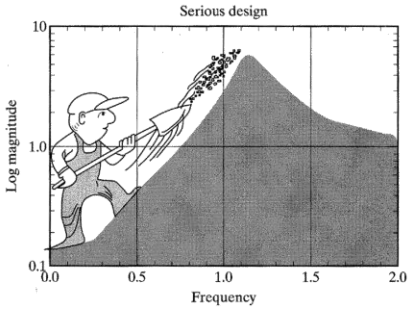
- Integral
  - Eliminates offsets (makes regulation ☺)
  - Leads to Oscillatory behaviour
  - Adds an "order" but instability (Makes a 2nd order system 3rd order)



→ "**I**nteresting cake of control"

Seeing PID – No Free Lunch

- The energy (and sensitivity) moves around (in this case in "frequency")



- Sensitivity reduction at low frequency unavoidably leads to sensitivity increase at higher frequencies.

Source: Gunter Stein's interpretation of the water bed effect – G. Stein, *IEEE Control Systems Magazine*, 2003.

PID Intuition & Tuning

- Tuning – How to get the "magic" values:
  - Dominant Pole Design
  - Ziegler Nichols Methods
  - Pole Placement
  - Auto Tuning

- Although PID is common it is often poorly tuned
  - The derivative action is frequently switched off!
  (Why ∵ it's sensitive to noise)
  - Also lots of "I" will make the system more transitory & leads to integrator wind-up.

PID Intuition

$$u(t) = K\left[e(t) + \frac{1}{T_i}\int e(s)\,ds + T_d\frac{de(t)}{dt}\right]$$

- P:
  - Control action is proportional to control error
  - It is necessary to have an error to have a non-zero control signal
- I:
  - The main function of the integral action is to make sure that the process output agrees with the set point in steady state

PID Intuition

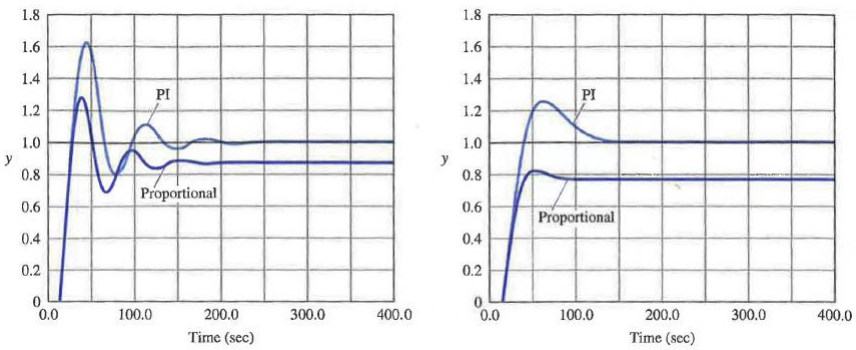
$$u(t) = K\left[e(t) + \frac{1}{T_i}\int e(s)\,ds + T_d\frac{de(t)}{dt}\right]$$

- P:
- I:
- D:
  - The purpose of the derivative action is to improve the closed loop stability.
  - The instability "mechanism" "controlled" here is that because of the process dynamics it will take some time before a change in the control variable is noticeable in the process output.
  - The action of a controller with proportional and derivative action may e interpreted as if the control is made proportional to the *predicted* process output, where the prediction is made by extrapolating the error by the tangent to the error curve.

PID Intuition

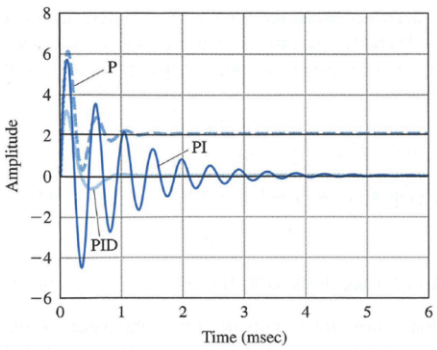
Effects of increasing a parameter independently					
Parameter	Rise time	Overshoot	Settling time	Steady-state error	Stability
K_p	↓	⇑	Minimal change	↓	↓
K_I	↓	⇑	⇑	Eliminate	↓
K_D	Minor change	↓	↓	No effect / minimal change	Improve (if K_D small)

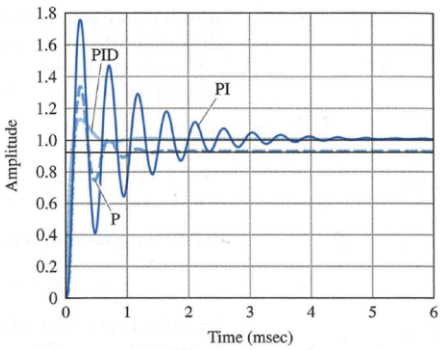
PID Intuition: P and PI



PID Intuition: P and PI and PID

- Responses of P, PI, and PID control to





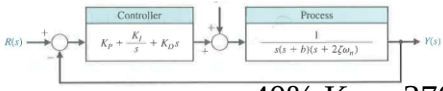
(a) step disturbance input (b) step reference input

PID Example

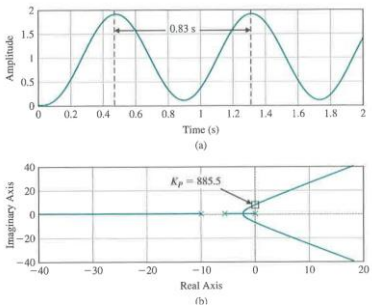
- A 3rd order plant: b=10, ζ=0.707, ω_n=4

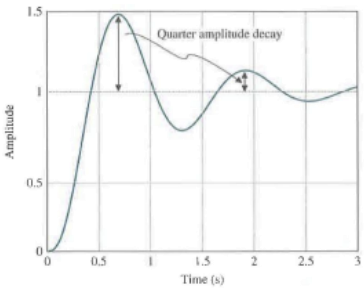
$$G(s) = \frac{1}{s(s+b)(s+2\zeta\omega_n)}$$

- PID:



- Kp=855: · 40% Kp = 370





Ziegler-Nichols Tuning – Reaction Rate

FPW § 5.8.5 [p.224]

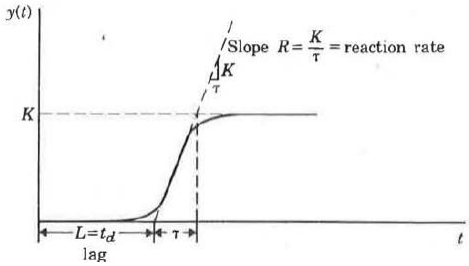
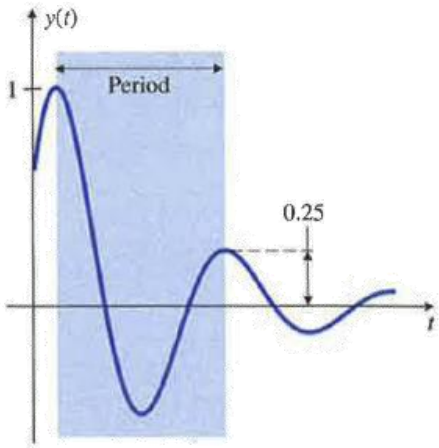


Table 5.2 Ziegler-Nichols tuning parameters using transient response.

	K_p	T_I	T_D
P	$1/RL$		
PI	$0.9/RL$	$3L$	
PID	$1.2/RL$	$2L$	$0.5L$

Quarter decay ratio



Ziegler-Nichols Tuning – Stability Limit Method

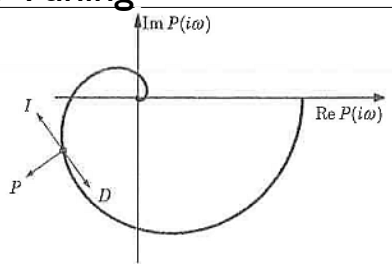
FPW § 5.8.5 [p.226]

- Increase K_p until the system has continuous oscillations
  - $\equiv K_U$: Oscillation Gain for "Ultimate stability"
  - $\equiv P_U$: Oscillation Period for "Ultimate stability"

Table 5.3 Ziegler-Nichols tuning parameters using stability limit.

	K_p	T_I	T_D
P	$0.5K_u$		
PI	$0.45K_u$	$P_u/1.2$	
PID	$0.6K_u$	$P_u/2$	$P_u/8$

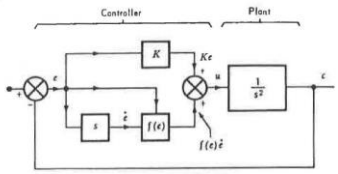
Ziegler-Nichols Tuning



$$C(i\omega_u) = K\left(1 + i\left(\omega_u T_d - \frac{1}{\omega_u T_i}\right)\right) \approx 0.6K_u(1 + 0.467i)$$

Poles are Eigenvalues: Some Implications

Stability of a 2nd order regulator



$$u = Ke + f(e)\dot{e}$$

state equations let $e = x_1$ and $\dot{e} = x_2$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -Kx_1 - f(x_1)\, x_2$$

assume for simplicity that K = 1.

$$0 = x_2^0$$

$$0 = -x_1^0 - f(x_1^0)x_2^0$$

The Jacobian matrix is

$$A = \begin{bmatrix} 0 & 1 \\ -1 & -f(0) \end{bmatrix}$$

- The linear behavior of the system in the close neighborhood of the origin is described by

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_1 - f(0)x_2$$

- AND, the characteristic equation is:

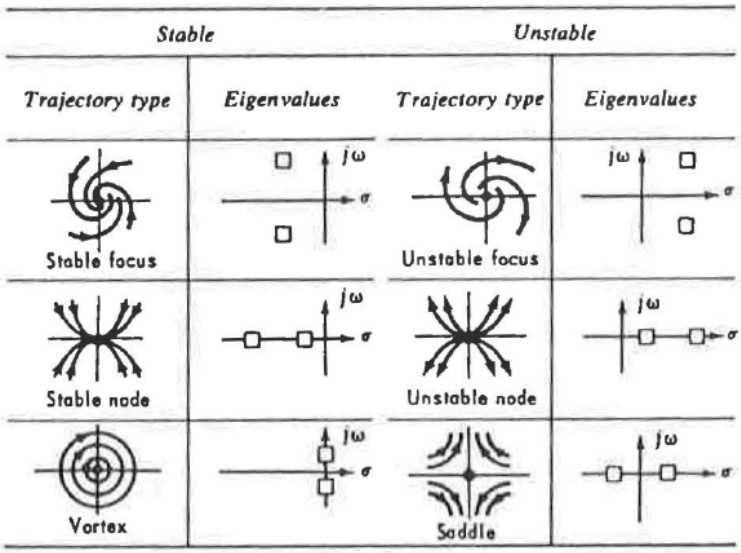
$$s[s + f(0)] + 1 = 0$$

with the eigenvalues

$$\lambda_1 = -\frac{1}{2} f(0) + \sqrt{\frac{1}{4} f^2(0) - 1}$$

$$\lambda_2 = -\frac{1}{2} f(0) - \sqrt{\frac{1}{4} f^2(0) - 1}$$

Various Types of Singularities (2nd order systems)



Next Time…

- **Digital Control via Emulation!**
- Review:
  - PID notes online
  - Chapter 5 of FPW
- Deep Pondering??