MAE 143C

<u>Computer architecture, software</u>

- Multithreading: one process (one core), multiple threads
  - POSIX: group making standards for multithreaded programming: pthreads
    - set: thread ID, priority, and other parameters
    - scheduling policy SCHED_FIFO (first in first out)
      - separate queues for each priority level
  - concurrency vs parallelism, multi-process computing
  - prioritization: set to max for anything that is essential for system stability
  - commands:
    - pthread_create : create thread, with parameters
    - pthread_join : wait until thread is terminated
    - pthread_setschedparam : modify priority of an existing thread
    - pthread_getschedparam(tid, policy, $param) : get priority of existing thread
      - we are using policy = SCHED_FIFO
- von Neumann architecture: Memory (RAM) stores both data and instructions so an instruction fetch and a data operation cannot occur at the same time because they share a common bus (von Neumann bottleneck)
  - Harvard architecture has separate pathways for instructions and data
  - “memory” = RAM, “storage” = hard drive
- Memory map: contains information about layout of memory. Operating system refers to this at boot
- Register size = number of bits (width) of one register, between CPU and RAM
  - 32 bit computers can only access 4GB of RAM
- Memory bus: CPU to RAM.
  - data bus: carries memory data
    - Wider data bus (more bits) means higher performance
  - address bus: selects memory address to write to/read from
- Clock speed: frequency generated by the oscillator that sets the tempo for the processor.
  - vs FLOPS (floating point operations per second), another measure of performance. Most microprocessors today carry out 4 FLOPs per core per clock cycle
- Beaglebone Black
  - 1GHz ARM Cortex-A8 Processor (32-bit)
    - vs 16MHz, 8-bit Arduino Uno
  - 4GB 8-bit eMMC flash storage
  - 2x PRU 32-bit microcontrollers
    - small computers for embedded systems applications
- Linux is a kernel, Ubuntu and Debian are operating systems. Same with Unix and OS X
  - GNU/Linux (GNU is not Unix)-848
  - Linus Torvalds, Richard Stahlman

<u>Serial communication</u>

- Serial communication concepts
  - vs parallel communication: (ex. sending 8 bits at a time, instead of 8 bits sequentially)

        - sacrifice speed for less pins/lines
    - synchronous vs asynchronous
        - synchronous has an extra line to always pair both sides with a clock signal (SPI, I2C)
        - asynchronous: without external clock signal
    - Baud rate: how fast data is sent over serial line, in bits per second
    - Framing data: start, data, parity, stop sections
        - Parity: error checking. Add up all bits of data byte, if even then 0, if odd then 1
    - RX: receiving, TX: transmitting
        -
- Serial communication types
    - SPI (serial peripheral interface)
        - 4 lines (unidirectional)
            - SCLK (serial clock):): samples taken at clock up
            - MOSI (Master Out Slave In)
            - MISO (Master In Slave Out)
            - SS (Slave Select): one for each slave
        - slave select line is pulled low to select from multiple slaves
        - Faster than I2C
        - receiver can be shift register
    - I2C (inter-integrated circuit)
        - 2 lines (bidirectional)
            - SCL (clock)
            - SDA (data line)
        - Messages: address frame, then data frames
            -
        - More complex than SPI but only requires 2 lines
        - supports multiple masters and slaves
        - Drivers are “open drain”: can only pull high signal to low
    - UART (universal asynchronous receiver/transmitter: intermediary between parallel and serial interfaces
        - buffer: stores data until microcontroller retrieves it
        - bit-banging: when microcontroller doesn’t have UART in IC form or onboard, so it directly controls serial interface

<u>Electronics</u>

- H-bridge [see ch20]
    - enables voltage to be applied across a load in either direction by switching 4 relays/transistors
- GPIO pin: pin whose behavior is controllable by user in real time. Used to read from various sensors, writing outputs to motors, LEDs, LCDs, etc.
    - How to access
        - “int fd = open(……)” , ex. LED problem in homework
        - set up interrupts, monitoring thru terminal, Linux kernel
        - /sys/class/GPI
- Quadrature encoder
    - incremental
    - Two optical sensors, 90 degrees out of phase

        - can infer position and direction of rotation
        - ex. If A leads B, the disk is rotating in a clockwise direction. If B leads A, then the disk is rotating in a counter-clockwise direction.
        -
- PWM
    - varying duty cycle (% of time that signal is high)
    - Human detection limit frequency for sight is lower than that for sound
    - servos: controlled by sending a PWM signal
- ADC, sensors
    -
    - resolution = 2^bits
    - full scale range = minimum reading ~ maximum reading. 2^(bits-1)
    - As sensitivity increases, range decreases
- Pull-up resistor: Connects input pin to voltage source by default (active high), to make sure reading isn't ambiguous. Switch (input) connects it to ground, which changes the reading to low.
    - pull down connects input pin to ground by default (active low), switch connects pin to voltage source
- IMU (inertial measurement unit)
    - The Robotics Cape includes a 9-Axis IMU which means it has 3 Orthogonally arranged accelerometers, gyroscopes, and magnetometers. Specifically, it uses the popular MPU-9150 IMU from Invensense. This sensor is particularly useful because of its integrated Digital Motion Processor, or DMP. The DMP can filter the sensor data and provide a quaternion orientation estimation directly available from the sensor in addition to the raw sensor data. This reduces the amount of floating-point arithmetic and trigonometry that your BeagleBone has to do for basic orientation estimation.
    - The IMU is attached to I2C_2 on pins P9_19 and P9_20 alongside the Barometer. To allow full sensor sampling rate, the Robotics Cape installer increases the default speed of this bus from 100khz to 400khz. Critically, the IMU's interrupt output pin is also connected to the BeagleBone on pin P9_25 which is configured for edge detection.
    - accelerometer (acceleration due to gravity and other forces)
        - internal capacitive plates, measures change in capacitance from moving plate
    - gyroscope (angular velocity)
        -
        - Probably measures capacitance as well. Others use piezoelectric
    - magnetometer (strength and direction of magnetic field)

<u>Links</u>

http://electronics.stackexchange.com/questions/29037/tradeoffs-when-considering-spi-or-i2c

https://www.youtube.com/watch?v=rH44ttR3G4Q&list=PLUMWjy5jgHK34zHva0RNs08PWDp1A-sKE&index=8

https%3A%2F%2Flearn.sparkfun.com%2Ftutorials%2Ftags%2Fconcepts%3Fpage%3D1&h=SAQFSJnQc

https://docs.oracle.com/cd/E19455-01/806-5257/6je9h0329/index.html

https://docs.oracle.com/cd/E19455-01/806-5257/6je9h0342/index.html

- if the closed-loop system is continuously excited (sinusoidally) at this resonant frequency ω_r, the magnitude of the response will be unbounded, and
- any tiny unmodeled error in either the plant or the controller could lead to closed-loop instability.

We might thus label the imaginary axis in the s-plane as an **axis of evil**: it is imperative to check any stable closed-loop system (that is, with its poles in the LHP) to ensure that its poles are in some sense "far" from being on this axis. Two valuable measures that may be read directly off the Bode plot (see, e.g., Figure 19.6) accomplish exactly this: the **phase margin (PM)** quantifies the amount that the phase of the open-loop system $G(i\omega)D(i\omega)$ is away from $-180°$ at the frequency ω_c for which the open-loop system gain equals 1, whereas the **gain margin (GM)** quantifies the factor by which the gain of the open-loop system $G(i\omega)D(i\omega)$ is away from 1 at the frequency ω_p for which the open-loop system phase equals $-180°$. If the PM is large, then there may be correspondingly large errors in the modeling of the phase of the system (due, for example, to unmodeled delays in the system) before risking closed-loop instability, whereas if the GM is large, then there may be correspondingly large errors in the modeling of the gain of the system (due, for example, to uncertainty in the actuator authority or sensor sensitivity) before risking closed-loop instability.

The Bode plot illustrated in Figure 19.6 also depicts the typical constraints considered during the design of the controller $D(s)$. A *large open-loop gain $|G(i\omega)D(i\omega)|$ is generally sought at low frequencies* to ensure adequate tracking of the reference input, and a *small open-loop gain is generally sought at high frequencies* to ensure adequate attenuation of high-frequency disturbances[6]. Thus, at some intermediate frequency [dubbed the **crossover frequency** ω_c], $|G(i\omega_c)D(i\omega_c)| = 1$. As in (18.17), the following convenient **approximate design guides** may be identified by examining a range of step responses[7]:

$$\omega_c \approx \omega_n \approx \omega_{BW}/1.4, \quad \zeta \approx \mathrm{PM}/100. \tag{19.15}$$

Thus, a target value for ω_c may be determined from the rise time or tracking constraints on the system, and a target value for the PM may be determined from the overshoot constraint on the system. Noting Figure 19.6, when performing controller design using a Bode plot, one typically tunes first the phase of the controller $D(s)$ to achieve the desired PM at the target value of the crossover frequency ω_c, then tunes the overall gain of the controller to actually achieve crossover at this target frequency. Next, if necessary, the gain of the controller at the low and high frequencies are adjusted to meet the tracking and robustness constraints, in addition to ensuring an adequate GM, and the overall gain readjusted to maintain crossover at the target frequency. Finally, the step response of the closed-loop system is checked, and any required fine tuning applied to meet the design constraints (e.g., increasing the crossover frequency to reduce the rise time, and increasing the PM to reduce the overshoot). As shown in §19.3, this process can often be achieved by a straightforward and methodical combination of lead compensation, lag compensation, and low-pass filtering.

As noted in Fact 18.15, a remarkable and useful feature of the Bode plot is that it is *additive*; that is, $\log|G(i\omega)D(i\omega)| = \log|G(i\omega)| + \log|D(i\omega)|$, and $\angle G(i\omega)D(i\omega) = \angle G(i\omega) + \angle D(i\omega)$. Thus, when examining the Bode plot of the plant $G(s)$, it is usually clear what is needed in terms of the gain and phase of the controller $D(s)$ such that the cascade of the controller and plant together have the appropriate overall behavior to meet the design guides discussed above, and thus, e.g., the rise time and overshoot constraints on the closed-loop system. Control design leveraging the Bode plot like this is referred to as **loop shaping**.

As noted at the end of §18.4.1, in systems with no RHP zeros or poles, the gain and phase curves are related in a simple fashion: a gain slope of -2 over a particular range of frequencies corresponds to $\sim 1/(i\omega)^2$ behavior of the transfer function, and thus a phase of about $-180°$; similarly, a gain slope of -1 corresponds to a phase of about $-90°$, and a gain slope of 0 corresponds to a phase of about $0°$. Thus, a rule of thumb for achieving a good PM (and, thus, good damping and low overshoot) in many systems is to *attempt to achieve crossover at a gain slope of approximately* -1; if crossover is attempted at a gain slope of closer to -2, the PM will often be unacceptably small (and, thus, the overshoot will be unacceptably high).

---

[6] That is, for small ω, $\frac{Y(i\omega)}{R(i\omega)} = \frac{G(i\omega)D(i\omega)}{1+G(i\omega)D(i\omega)} \approx 1$ if $|G(i\omega)D(i\omega)| \gg 1$; for large ω, $\frac{U(i\omega)}{W(i\omega)} = \frac{-G(i\omega)D(i\omega)}{1+G(i\omega)D(i\omega)} \approx 0$ if $|G(i\omega)D(i\omega)| \ll 1$.

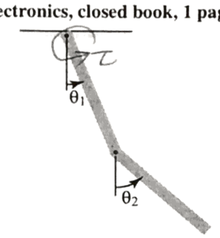
[7] For clarity of presentation, the definition of ω_{BW} is deferred to §19.2.4 and Figure 19.8b.

$L(s) = G(s)D(s)$
Root locus for stability
Bode plot ~~for~~ to ~~tune~~ performance ← adjustments
Step response: rise time, overshoot, settling time
check → Bode of T(s)

MAE143c: Embedded Control & Robotics, midterm (2015)

Modeling the double inverted pendulum

90 minutes, no electronics, closed book, 1 page of notes allowed.



Consider the driven two-pendulum system illustrated above, where each pendulum is a uniform rod of length ℓ and mass m, a motor is applied at the base to which applies a torque τ to the first pendulum, and the angles are measured, in radians, counterclockwise from the down orientation. This system is governed by the following four first-order equations:

$$\left[16 - 9\cos^2(\theta_1 - \theta_2)\right]\frac{d\theta_1}{dt} = \frac{6}{m\ell^2}\left[2p_1 - 3\cos(\theta_1 - \theta_2)p_2\right], \tag{1a}$$

$$\left[16 - 9\cos^2(\theta_1 - \theta_2)\right]\frac{d\theta_2}{dt} = \frac{6}{m\ell^2}\left[8p_2 - 3\cos(\theta_1 - \theta_2)p_1\right], \tag{1b}$$

$$\frac{dp_1}{dt} = -\frac{m\ell^2}{2}\left[\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}\sin(\theta_1 - \theta_2) + \frac{3g}{\ell}\sin\theta_1\right] + \tau, \tag{1c}$$

$$\frac{dp_2}{dt} = -\frac{m\ell^2}{2}\left[\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}\sin(\theta_2 - \theta_1) + \frac{g}{\ell}\sin\theta_2\right], \tag{1d}$$

where p_1 and p_2 are the generalized momenta of the system (see `https://en.wikipedia.org/wiki/Double_pendulum`).

1. Linearize this system about a (possibly time-varying) nominal trajectory $\{\bar\theta_1(t), \bar\theta_2(t), \bar p_1(t), \bar p_2(t)\}$, none of the components of which is (yet) assumed to be small. That is, in (1a) - (1d), take

$$\theta_1(t) = \bar\theta_1(t) + \theta_1'(t), \quad \theta_2(t) = \bar\theta_2(t) + \theta_2'(t), \quad p_1(t) = \bar p_1(t) + p_1'(t), \quad p_2(t) = \bar p_2(t) + p_2'(t), \quad \tau(t) = \bar\tau(t) + \tau'(t),$$

multiply out, and keep all terms which are linear in the perturbation (primed) quantities. The following identities will help:

$$\sin(x \pm y) = \sin x\,\cos y \pm \cos x\,\sin y, \quad \cos(x \mp y) = \cos x\,\cos y \pm \sin x\,\sin y.$$

simplify?

small angle approx

$\frac{d(\theta_2' + \pi)}{dt} \stackrel{?}{=} \frac{d\theta_2'}{dt}$

2a. Starting from the solution to problem 1 above, linearize this system about the (stable) equilibrium nominal trajectory given by the hanging condition $\bar\theta_1 = \bar\theta_2 = 0$.

2b. Starting from the solution to problem 1 above, linearize this system about the (unstable) equilibrium nominal trajectory given by the "jack-knifed" condition $\bar\theta_1 = 0$, $\bar\theta_2 = \pi$.

2c. Starting from the solution to problem 1 above, linearize this system about the (unstable) equilibrium nominal trajectory given by the upright condition $\bar\theta_1 = \bar\theta_2 = \pi$.

3a. Take the Laplace transform of the system of linearized equations governing the upright condition, as given in problem 2c above. Combine the resulting four algebraic equations to eliminate θ_1, p_1, and p_2, thereby determining a single transfer function $G_{\text{inner}}(s) = \theta_2(s)/\tau(s)$. [Hint: use a simplified notation for the intermediate algebra to minimize the work, introducing linear operators $\mathcal{L}_i$ for each of the coefficients as appropriate.]

3b. Proceed as in problem 3a above, but now combine the equations to eliminate τ, p_1, and p_2, thereby determining a single transfer function $G_{\text{outer}}(s) = \theta_1(s)/\theta_2(s)$.

4. Assuming $m = 1$, $\ell = 1$, and $g = 10$, and evaluating, rewrite the transfer functions $G_{\text{inner}}(s)$ and $G_{\text{outer}}(s)$ given in problems 3a and 3b with the coefficients given as numerical values. If possible, determine where the poles and zeros of each of these transfer functions are (during the midterm, only attempt this if it's possible for you to do by hand!).

$(x\theta_2' - x'\theta_1')(\bar p_2 + p_2')$ $\quad \sin(\theta_2 - \theta_1)$

$x\theta_2'\bar p_2 + \cancel{x\theta' p_2'} - x'\theta_1'\bar p_2$

MAE 143C Notes — Kazuya Otani

Bode plots

Slope and phase change gradually over two breakpoints

1. For small w, gain and phase approach $G(iw)$ $s \to 0$
2. $w =$ $|z_1|, |z_2|, \ldots$ $|p_1|, |p_2| \ldots$ are breakpoints

a. At first-order pole (zero) of mult k, slope of gain curve decreases (increases) by k

At LHP ~~first order~~ pole (zero) of mult k, phase decreases (increases) by $k \cdot 90°$

RHP first-order pole (zero) of mult k, phase increases (decreases) by $k \cdot 90°$

b. @ pair of complex-conjugate poles (zeros) of mult k, slope of gain decreases (increases) 2k

@ LHP complex-conjugate poles (zeros) of mult k, phase decreases (increases) by $k \cdot 180°$

RHP poles (zeros) phase increases (decreases) by $k \cdot 180°$

Root locus plots

At 0 Hz, most systems are gain of 1.

Larger w: faster response. Further left → more damping

1. Symmetric about real axis
2. Branches start at poles ($K = 0$), end at zeros ($K \to \infty$)

LPF: $G(s) = \frac{1}{s+p}$

3. Pairing poles and zeros: From right to left

Draw on real axis to the left of an odd number of real poles plus zeros counted from the right

4. Asymptotes

@ $\alpha = \frac{\sum p_i - \sum z_i}{n-m}$, $\theta_l = \frac{180° + (l-1)360°}{n-m}$ for $l = 1, \ldots, n-m$

5. $\phi_{dep} = \frac{\sum \phi_i - \sum_{i \neq k} \psi_i + 180° + (l-1)360°}{q_k}$

$\frac{dk}{ds} = 0$ @ breakaway points

$\phi_{arr} = \frac{\sum_{i \neq k} \phi_i - \sum \psi_i + 180° + (l-1)360°}{r_k}$

Bode plots are additive

If $G(s)D(s)$ has gain of 1 and phase of $-180°$ @ resonant frequency, $T(s)$ has pole on imaginary axis

gain margin — phase margin

Crossover frequency $|G(iw_c)D(iw_c)| = 1$

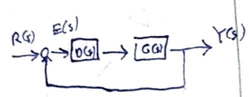
Large OL gain → small OL gain

Design guidelines

$w_c \approx w_n \approx w_{BW}/1.4$ $\zeta \approx PM/100$

$t_r \approx 1.8/w_n$, $t_s \approx 4.6/\sigma$, $M_p \approx e^{-\pi\zeta/\sqrt{1-\zeta^2}}$ $p_\pm = -\sigma \pm i w_d$

$w_n \gtrsim \frac{1.8}{t_r}$, $\sigma \gtrsim \frac{4.6}{t_s}$, $\begin{cases} \zeta \gtrsim 0.5 & \text{for } M_p \leq 15\% \\ \zeta \gtrsim 0.7 & \text{for } M_p \leq 5\% \end{cases}$



$T(s) = \frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)}$

zero ss error = too aggressive

Designing $D(z)$

1) Find $D(s)$ then translate w/ Tustin — will work if $w_c \sim 10\%$ Nyquist

$\frac{1}{2} \times$ sampling f

$G(z) = \frac{z-1}{z} \mathcal{Z}\left(\frac{G(s)}{s}\right)$

~~Loop shaping~~ RL

1. ~~Bode~~, get things in LHP

OR Lead, lag, LPF & Notch occasionally to fix

2. Bode look at PM & GM. $w_c \approx 1.8/t_{rise}$

3. Loop shaping: tracking (high gain @ low freq), robustness (low gain @ high freq)

$\sin(x \pm y) = \sin x \cos y \pm \cos x \sin y$

$\cos(x \mp y) = \cos x \cos y - \sin x \sin y$

$e^{i\omega t} = \cos(\omega t) + i \sin(\omega t)$

$\cos\theta = \frac{1}{2}(e^{i\theta} + e^{-i\theta})$

$\sin\theta = \frac{1}{2i}(e^{i\theta} - e^{-i\theta})$

Linearizing $\quad x = \bar{x} + x', \; \theta = \bar{\theta} + \theta', \; u = \bar{u} + u'$

Group coefficients

Use small perturbations to get linear equations

~~##~~ If perturbations squared or higher terms, then they $= 0$

Take x, θ, u varying w/ time

$x', \theta', u' \rightarrow$ quadratic or higher $= 0$ $\qquad \sin x' = x'$, $\; \cos x' = 1$

If you don't recognize it use Taylor

$$f(x) = f(a) + f'(a)\cdot(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots$$

Laplace

$f(t)$	$F(s)$
1	$1/s$
t	$1/s^2$
$\delta(t)$	1
$f'(t)$	$sF(s) - f(0)$

$\rightarrow f''(t) = s^2F(s) - sf(0) - f'(0)$

$$F(s) = \int_0^\infty f(t)e^{-st}\,dt$$

$$F(z) = \sum_{k=0}^{\infty} f_k z^{-k}$$

$z = e^{sh}$

Final value theorem

$\lim_{s\to 0} sF(s) = \lim_{t\to\infty} f(t)$

$\lim_{s\to\infty} sF(s) = \lim_{t\to 0} f(t)$

Z $\quad$ If $\lim_{k\to\infty} f_k$ is bounded, then

$\lim_{z\to 1} (z-1)F(z) = \lim_{k\to\infty} f_k$

$\lim_{z\to\infty} F(z) = f_0$

Partial fraction

$$F(x) = \frac{a}{s-x} + \frac{b}{s-y} + \frac{c}{s-z} \cdots$$

$$a = \lim_{s\to x} (s-x)F(x)$$

Derivative: s

Integral: $1/s$

Approximations

Euler's: $z = 1 + sh \rightarrow$ good around $s = 0$. scale and shift

Tustin's

$$z = \frac{1 + sh/2}{1 - sh/2} \quad \& \quad s = \frac{2}{h}\frac{z-1}{z+1}$$

$h =$ step size

Prewarping Tustin: scale s-plane before mapping

$$z = \frac{1 + fsh/2}{1 - fsh/2} \quad \& \quad s = \frac{2}{fh}\frac{z-1}{z+1} \qquad f = \frac{2[1-\cos(\omega h)]}{\omega h \sin(\omega h)}$$

$G(s) = D_{Lead}(s) \cdot D_{lag}(s) \cdot D_{low\text{-}pass}(s) \cdot K$

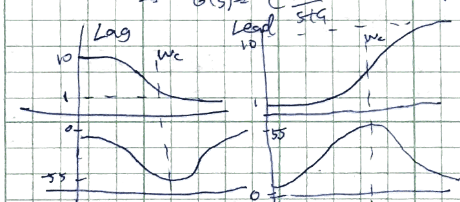
$D_{PID}(s) = K_p\left(1 + \frac{1}{T_I s} + T_D s\right)$

Lag $\quad K\frac{s+z}{s+p} \quad p < z$

Lead $\quad K\frac{s+z}{s+p} \quad z < p$

$D_{notch}(s) = K\frac{s^2 + z^2}{(s+p)^2}$

If $G(s) = C\frac{s+b}{s+a}$ $\quad \omega^2 = pz \rightarrow \omega = \sqrt{pz}$



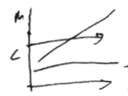
· For $p/z = 10$, phase max $\approx 55°$

As p/z increases, $\rightarrow 90°$

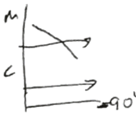
MAE 143C Final Notes

Lead-lag compensators

Zero (s):



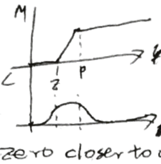
Pole $\left(\frac{1}{s}\right)$:



Gain = 1

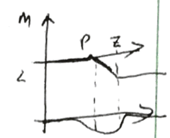
Lead $\frac{w_p}{w_z}\frac{s+w_z}{s+w_p}$

$w_z < w_p$ — zero closer to origin



Lag $\frac{w_p}{w_z}\frac{s+w_z}{s+w_p}$

$w_z > w_p$ — pole closer to origin



Design guidelines

$w_c = \frac{1.8}{t_r}$

$w_c = \sqrt{pz}$

Gain and phase margins

Protects from instability: less margin = less stable

For $\frac{G(s)}{1+G(s)}$ to go to ∞, $1+G(s) \to 0$

$1+G(s)=0 \longrightarrow G(s)=-1$ avoid this

Same as "-1" on Nyquist

0dB gain, $-180°$ phase

1 gain

Bode plot design

Requirements

- Phase margin @ Gain = 1
- Gain margin @ $-180°$
- Gain crossover: where gain crosses 1
- Bandwidth: where gain crosses -3dB
- Zero-frequency magnitude

See design guidelines

Step 1: System type: # of poles at origin

For step input: type 0 ~~finite~~ e_{ss}

$\boxed{\text{type 1}}$ ~~0~~ 0 e_{ss}

type 2 finite e_{ss}

∞ e_{ss}

Step 2: choose gain K

Step 3: choose phase, gain bumps

$e_{ss} = \lim_{s\to 0} s\,E(s)$

$= \lim_{s\to 0} \frac{1}{1+G(s)} R(s)$

Pole-zero cancellations: OK in LHP, not ok in RHP

Closed loop TF

$$T(s) = \frac{G(s)D(s)}{1+G(s)D(s)}$$

Changing gain K moves gain curve up, doesn't change phase.

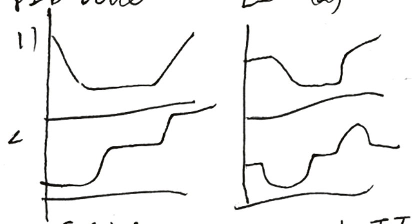
Nyquist freq order of mag. higher than crossover frequency.

~~$-5.5\,\frac{s+10}{s+100}\,\frac{s+12}{s}$~~

~~$0.07\,\frac{s+0.5}{s+5}\,\frac{s+9}{s+12}$~~

PID Bode

Lead lag



$$G_c(s) = K_p\left(1+\frac{1}{T_i s}+T_d s\right) = K_p\frac{T_i T_D s^2 + T_i s + 1}{T_i s}$$

Taylor Series

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \cdots$$

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \cdots$$

Linearization

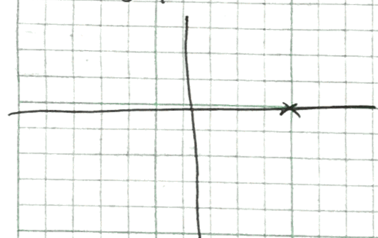
1. plug in $x = \bar{x} + x'$

Take out $\bar{x}$ nominal values: they satisfy equations

Taylor expansion

Eliminate terms non-linear in perturbation

6.1



Minimum Energy Stabilizing Controller

$G(s) = \frac{b(s)}{a(s)}$ $D(s) = \frac{y(s)}{x(s)}$ $T(s) = \frac{g(s)}{f(s)} = \frac{b(s)y(s)}{\boxed{a(s)x(s)+b(s)y(s)}}$

Diophantine equation

LHP poles of G(s) + reflections of RHP poles of G(s)

Choose order of y(s), x(s) so that a(s)x(s), b(s)y(s) have one higher order than f(s)

Tustin's @ Wc for accurate mapping around that frequency

DT, ADC leads to h/2 lag → phase lead around Wc must be built in

zero order hold (ZOH)

Direct digital design

Potential oscillations between timesteps

→ place poles far away from axis

DT final value theorem

$$\lim_{k\to\infty} y_k = \lim_{z\to 1} (z-1)Y(z)$$

CT

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s)$$

Deadbeat control: $D(z) = \frac{1}{G(z)} \frac{T(z)}{1-T(z)}$

$D(z) = \frac{b(z)}{a(z)}$

All poles on origin

$$\ell > \deg\{\text{denom}(T(z))\} + \deg\{a(z)\} - \deg\{b(z)\}$$

minimal-time deadbeat controller

↓

intersample ripple

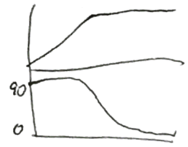
→ ripple-free deadbeat controller — settles in finite timesteps

Low pass

$\frac{W_c}{s+W_c}$

Highpass

$\frac{s}{s+W_c}$

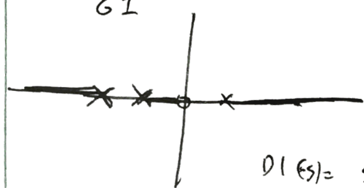


Non-minimum phase: system with 1 RHP zero

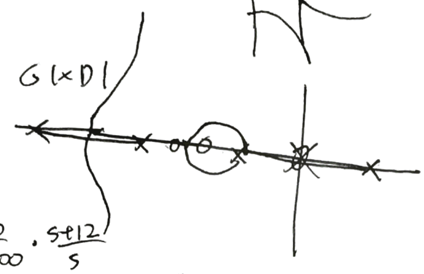
Notch $D(s)=\frac{K(s^2+z^2)}{(s+p)^2}$ $p>0, z>0$

My controller

G1

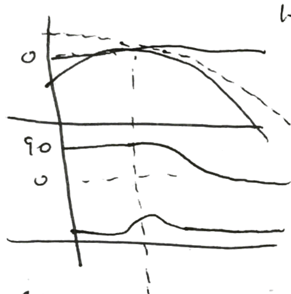


G1×D1

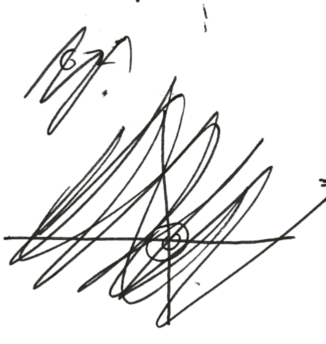


$D1(s)=\frac{s+10}{s+100}\cdot\frac{s+12}{s}$

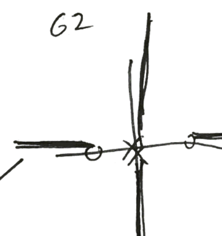
lead integrator/lag



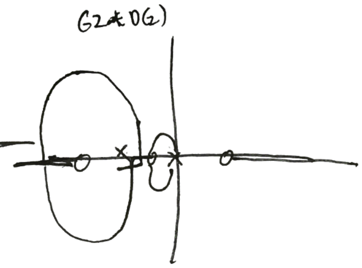
$D_2(s)=\frac{s+0.8}{s+8}$

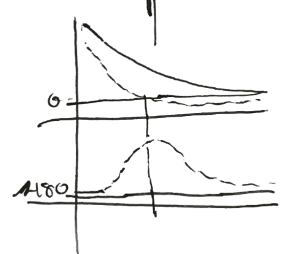


G2



G2*D(2)





143 CH4 Notes

- rise time 0.05 ~ 0.1 s for inner loop, ×10 for outer
  - lead and lag centered around same frequency

Inner loop rlocus

Stabilize unstable pole first, then move pole on LHP closer to origin to ~~the~~ the left

If step response s.s. ~~value~~ is 1.2~1.3, prefactor T.F. for controller

output of outer loop × $\frac{1}{\text{prefactor}}$

Debugging:

test_imu, test_encoders, test_motors

Try PD control

lead-lag for both loops

Add K (negative value) to inner loop code

K for outer loop

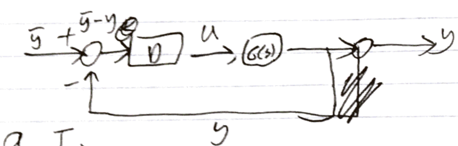
p = prefactor

probably copy lag from inner loop for outer loop

matlab: minreal

$$\frac{\overset{\text{output}}{O(z)}}{\underset{\text{input}}{I(z)}} = \frac{a_1 z + a_2}{b_1 z + b_2}$$

× z : step forward in time



$$O_i \times z = O_{i+1}$$

$$b_1 O_{i+1} + b_2 O_i = a_1 I_{i+1} + a_2 I_i$$

$$b_1 O_i + b_2 O_{i-1} = a_1 I_i + a_2 I_{i-1}$$

$$O_i = (a_1 I_i + a_2 I_{i-1} - b_2 O_{i-1}) / b_1$$

normalize transfer function to make $b_1 = 1$

Add lag to outer loop

MAE 143C HW 4 Kazuya Otani

19.11

2 3
4 1

$$G(s) = \frac{(s+2)(s-2)(s+4)(s-4)}{(s+1)(s-1)(s+3)(s-3)(s+5)(s-5)}$$

Make Minimum Energy stabilizing controller

$$D(s) = \frac{y(s)}{x(s)}, \quad G(s) = \frac{b(s)}{a(s)}, \quad T(s) = \frac{g(s)}{f(s)} = \frac{b(s)y(s)}{a(s)x(s)+b(s)y(s)}$$

stable open-loop poles of $G(s)$: $s = -1, -3, -5$

unstable openloop poles of $G(s)$: $s = 1, 3, 5 \longrightarrow$ reflection $= s = -1, -3, -5$

4 5
6 7

$$f(s) = (s+1)^2(s+3)^2(s+5)^2$$
$$= (s^2+2s+1)(s^2+6s+9)(s^2+10s+25)$$
$$= s^6+18s^5+127s^4+444s^3+799s^2+690s+225$$

$\cancel{y(x) = (s+4)(s-4)}$
$\cancel{= s^4-16s^2-4s^2+64}$
$\cancel{= s^4-20s^2+64}$

$$a(s) = (s+1)(s-1)(s+3)(s-3)(s+5)(s-5)$$
$$= (s^2-1)(s^2-9)(s^2-25)$$
$$= s^6-35s^4+259s^2-225$$
$$b(s) = (s+2)(s-2)(s+4)(s-4)$$
$$= (s^2-4)(s^2-16)$$
$$= s^4-20s^2+64$$

$$D(s) = \frac{y(s)}{x(s)} = \cancel{\frac{b_5 s^5+b_4 s^4+b_3 s^3+b_2 s^2+b_1 s+b_0}{a_3 s^3+a_2 s^2+a_1 s+a_0}}$$
$$= \frac{b_3 s^3+b_2 s^2+b_1 s+b_0}{a_1 s+a_0}$$

$$f(s) = a(s)x(s)+b(s)y(s)$$

$$s^6+18s^5+127s^4+444s^3+799s^2+690s+225$$
$$= (s^6-35s^4+259s^2-225)(a_1 s+a_0) + s^4-20s^2+64(b_3 s^3+b_2 s^2+b_1 s+b_0)$$

$$= \cancel{a_1 s^7 - 35a_1 s^5 + 259a_1 s^3 - 225a_1 s + a_0 s^6 - 35a_0 s^4 + 259a_0 s^2 - 225a_0}$$
$$\cancel{+ b_3 s^7 + b_2 s^6 + b_1 s^5 + b_0 s^4 - 20b_3 s^5 - 20b_2 s^4 - 20b_1 s^3 - 20b_0 s^2}$$
$$\cancel{+ 64b_3 s^3 + 64b_2 s^2 + 64b_1 s + 64b_0}$$

$$= s^7(a_1+b_3)+s^6(a_0+b_2)+s^5(-35a_1+b_1-20b_3)$$
$$+s^4(-35a_0+b_0-20b_2)+s^3(259a_1-20b_1+64b_3)$$
$$+s^2(259a_0-20b_0+64b_2)+s(-225a_1+64b_1)+(-225a_0+64b_0)$$

$$\underset{8\times 6}{\begin{array}{c} \begin{array}{cccccc} b_0 & b_1 & b_2 & b_3 & a_0 & a_1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -20 & 0 & -35 \\ 1 & 0 & -20 & 0 & -35 & 0 \\ 0 & -20 & 0 & 64 & 0 & 259 \\ -20 & 0 & 64 & 0 & 259 & 0 \\ 0 & 64 & 0 & 0 & 0 & -225 \\ 64 & 0 & 0 & 0 & -225 & 0 \end{bmatrix} \end{array}} \underset{6\times 1}{\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ a_0 \\ a_1 \end{pmatrix}} = \underset{8\times 1}{\begin{pmatrix} 0 \\ 1 \\ 18 \\ 127 \\ 444 \\ 799 \\ 690 \\ 225 \end{pmatrix}} \begin{array}{c} 7 \\ 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{array}$$

Using Diophantine algorithm

$$D(s) = \frac{1.8222\,s^5 + 5.8349\,s^4 - 43.3651\,s^3 - 127.3778\,s^2 + 67.1429\,s + 147.1429}{-1.8222\,s^3 - 5.8349\,s^2 + 16.0317\,s + 40.8540}$$

Strictly proper D(s)

$$D(s) = \frac{-364.4\,s^5 - 1167\,s^4 + 8673\,s^3 + (2.548\times10^4)s^2 - (1.34\times10^4)s - (2.943\times10^4)}{1.822\,s^6 + 60.55\,s^5 + 523.5\,s^4 + 645.2\,s^3 - 4432\,s^2 - 8171\,s}$$

Adding $\frac{1}{s}, \frac{1}{s+10}, \frac{1}{s+20}$