Dr. J. Tani, Prof. Dr. E. Frazzoli

151-0591-00 **Control Systems I** (Autumn 2018) **Exercise Set 12**

**Topic:** **Loop Shaping**

Discussion: 7. 12. 2018

grizzi@ethz.ch, 16th December 2018

**Learning objectives:** The student can

- Design a lead/lag controller given desired frequency domain specification.
- Derive the angle and magnitude of a lead/lag controller by hand.
- Classify when to use a lead controller and when to use a lag controller.
- State what the purpose of a lead/lag controller is.
- Handle lead/lag controller design for nonminimumphase systems.
- Comfortably use controller specifications to design a lead/lag controller.

# Background

As we have seen, disturbance rejection/reference tracking and noise rejection can be generally achieved if the loop shape $L(s)$ is "larg" enough at small frequencies and "small" at high frequencies. But how do we steer the open loop frequency response? We can use simple elements such as the *proportional compensator*, the *lead compensator* and finally the *lag compensator*. We will briefly review their characteristcs and how they can be used for *loop shaping*.

## Proportional (static) compensation

A proportional controller $k$ will effect the open loop plant shifting up the magnitude of the frequency response with no phase change. We can achieve good reference tracking but we will also make the plant more sensitive to noise since the entire transfer function, at low and high frequencies as well, will be shifted up.

## Lead compensation

The lead compensator has the form

$$C_{Lead} = \frac{s/a+1}{s/b+1} \qquad 0 < a < b$$

In the lead compensator the zero is at a smaller frequency than the pole's frequency. The effects are

- Increase the magnitude at high frequencies by a factor of $b/a$. Low frequencies ($\omega < a$) are not affected. The magnitude slope is increased of 20 dB/decade between $a$ and $b$.
- Increase the phase around $\sqrt{ab}$.

**Application:** the lead compensator is typically used to **increase the phase margin** choosing $\sqrt{ab}$ as the crossover frequency and tuning $a$, $b$ to get the desired phase increase.

**Side effect:** the magnitude at high frequency is increased of a factor of $b/a$ making the plant more sensitive to noise.

### Lag compensation

The lag compensator has the form

$$C_{Lead} = \frac{s/a + 1}{s/b + 1} \qquad 0 < b < a$$

In the lag compensator the pole is at a smaller frequency than the zero's frequency.Its effects are:

- Decrease the magnitude at high frequency by a factor of $b/a$. Low frequencies ($\omega < b$) are not affected. The magnitude slope is decreased of 20dB/decade between $b$ and $a$.
- Decrease of the phase around $\sqrt{ab}$.

**Application:** the lag compensator is typically used to imporove disturbance rejection/reference tracking at low frequency. The lag compensator is multiplied with a gain $b/a$ such that at low frequency the magnitude is shifted by $b/a$ while high frequencies ($\omega > a$) are not affected. The non normalized compensator has the form

$$C_{Lag} = k\frac{s/a + 1}{s/b + 1} = \frac{a}{b} \cdot \frac{s/a + 1}{s/b + 1} = \frac{s + a}{s + b}$$

**Side effect:** undesired reduction of phase margin if $\sqrt{ab}$ is chosen close to the crossover frequency.

### Non Minimum Phase Plants

With *non minimum phase plants* we refer to plants with either unstable poles, non minimum phase zeros or both of them. In such case, they impose constraints on the constrol design, especially in terms of the admissible crossover frequency.
The general approach when dealing with *non minimum phase plants* is to factor them in a minimum phase plant $P_{mp}(s)$ and an *all-pass filter* $D(s)$ such that it has negative phase. For example if $P(s) = (s-3)/(s-2)$ the factorization would be $P_{mp} = (s+3)/(s+2)$ and $D(s) = (s-3)/(s+3) \cdot (s+2)/(s-2)$.

The idea is to design the controller on $P_{mp}(s)$ such that the phase margin is large enough to account for the additional phase drop introduced by $D(s)$ at the crossover frequency (not changed by $D(s)$ being $|D(j\omega)| = 1$ for all $\omega$).

## Python for Control System Design

The package `control` provides many utility functions which can be efficiently deployed to design a control system and continuosly verify/test the results that you get as you solve the presented problems. You should already have installed the `control` library, for Problem Set 10. Examples are:

- `sys = control.tf(num, den)`: it returns a transfer function whit numerator coefficients num and denominator coefficients den in decreasing power order. This transfer functions can then be summed and multiplied together.

- `y, t, x = control.matlab.lsim(sys, U, T)`: it returns the output y, for the internally computed time t and the state evolution x for the specified system sys simulated for the time vector T under the input U.
- `mag, phase, omega = control.bode(sys, omega=omega, Plot=False)`: it returns the magnitude response of the system `sys` (for example generated using `control.tf()`) at the specified frequencies `omega`. It returns the frequency, the magnitude (absolute, not decibel), the phase (radians). Optionally, you can set the keyword argument `Plot` to `True` to automatically generate a logarithmic Bode plot.
  Remember to always call `matplotlib.pyploy.show()` in order to show the generated plot.
- `num, den = pade(T, n=order)`: it returns the Padé apprximation of the delay `T` of order `n`. The approximation is returned in terms of numerator and denominator coefficients of the corresponding transfer function. This function will be useful in exercise
- `gm, pm, wg, wp = control.margins(sys)`: it returns the stability margins for the LTI system `sys`. The returned variables are (in order):
  - `gm`: gain margin
  - `pm`: phase margin (in degrees)
  - `wg`: frequency for gain margin (where phase = -180°)
  - `wp`: frequency for phase margin (where gain = 1)
- `y, t = step(sys, T=time)`: it simulates the step response for the system `sys` for the specified time `T`. It returns in order: the output of the system, and the internally computed time vector `t`.

A detailed description of each function usage, input and output arguments as well as a list of additional material is available on the `control` library documentation page: `https://python-control.readthedocs.io/en/latest/intro.html`.

## Exercise 1 (Warm-Up)

**a)** Assume your open-loop system $L(s)$ is of type 0 and that you are using a proportional controller. You want to increase the closed-loop tracking performance of your open-loop system $L(s)$ at low frequencies. Currently your closed-loop tracking error is 10%. You would like this closed-loop error to be reduced to 2%. How does the low-frequency gain of your controller have to change to make this possible?

a) $k_{new} = 6.33 \cdot k_{old}$
b) $k_{new} = 4.66 \cdot k_{old}$
c) $k_{new} = 5.44 \cdot k_{old}$
d) $k_{new} = 5 \cdot k_{old}$

**b)** You want to increase the phase at the crossover frequency $c = 1rad/s$ by 30° by using a lead element $C(s) = \frac{s/a+1}{s/b+1}$. Which parameters $a$, $b$ should you choose?

a) $a = 1$, $b = 1$
b) $a = 1/\sqrt{3}$, $b = \sqrt{3}$
c) $a = \sqrt{3}$, $b = 1/\sqrt{3}$
d) $a = \sqrt{3}/2$, $b = 2/\sqrt{3}$

**c)** Let $P(s) = \frac{e^{-sT}}{(s+0.1)^2}$. You are given the frequency $\omega^* = 1$ rad/s at which $\angle P(j\omega^*) = -180$. What is the time delay $T$ of the system?

a) $T \approx 0.2\ s$

b) $T \approx 2.9\ s$

c) $T \approx 1.7\ s$

d) $T \approx 0.1\ s$

## Exercise 2 (A Non-minimum Phase System)

Design a lead-lag compensator $C(s)$ for a system modeled by Figure 1 with

$$P(s) = 5\frac{(1 - \frac{s}{200})}{(1 + \frac{s}{20})(1 + \frac{s}{4})}$$

so that

**a)** It has as fast a rise time as possible with a phase margin in the range from 40° to 45°.

**b)** Design your control system such that the steady state error to a step response is approximatelly the 0.5 %.

(*Hints*: In the first task you can use a *lead compensator* in order to add the missing phase margin. On the other hand, a properly placed *lag compensator* can increase the magnitude response at low frequencies.)

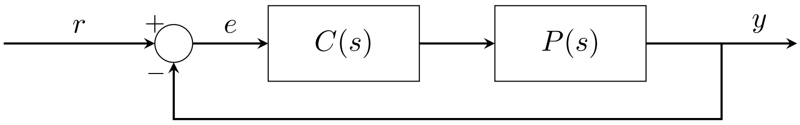


*Figure 1: The standard block diagram for a unit-feedback loop.*

## Exercise 3 (Time Delays, Homework: b-d)

It turns out that we did not perfectly model our tracking system in exercise 2 because we did not consider the time delay between our calculated control input and the plant.
Remember from the last exercise set that the transfer function of a time delay of $T_d$ seconds is:

$$G_d(s) = e^{-T_d s}$$

To more easily analyze systems with time delays, oftentimes practitioners approximate the time delay transfer function by a rational function called the *Padé approximation.* This is the best fitting approximation of a function by a rational function of a given order.
In this exercise you will look at a system with a delay and analyze it using these *Padé approximations.*

**a)** Draw by hand a Bode plot of $G_d$ for $T_d = 1$ sec.

**b)** Now the *Padé approximation* of the system is analyzed. For this part of the problem, assume $T_d = 1$ second.

1. Using PYTHON (`pade`), determine the first, second, and third order Padé approximations for $G_d$.
2. Using PYTHON's `bode`, plot on one figure the magnitudes and phases of (i) $G_d$ exactly and (ii-iv) the first through third order Padé approximations of $G_d$.

3. Similarly, using PYTHON's `step`, plot the step responses of all four transfer functions on one figure.

**c)** Now let us compare the performance of the compensator we designed in the last exercise for (i) $T_d = 0.01$ second AND (ii) $T_d = 0.1$ second.

1. For both choices of $T_d$, plot the Bode plots of the corresponding loop transfer function $L(s)$ when (i) there is no delay, (ii) the delay of $T_d$ seconds is modeled exactly, and (iii) the delay of $T_d$ seconds is modeled by a first order Padé approximation.
2. Similarly, for both choices of $T_d$ and each choice of model of $L(s)$, plot the closed-loop step responses of the system when (i) there is no delay and (ii) delay is modeled using the first order Padé approximation.

**d)** Use the first order Padé approximation to explain your results.

## Exercise 4 (Homework: Lateral Control of a Vectored Thrust Aircraft - *adapted from Åström & Murray*)

Consider a vectored thrust aircraft as the one showed in figure 2 with the parameters defined in table 1

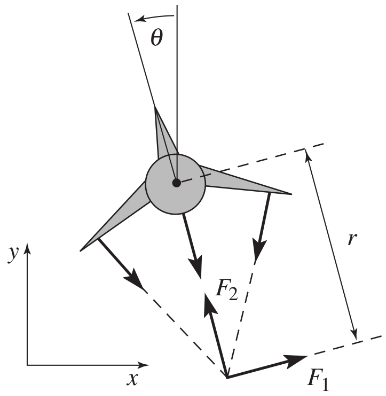


*Figure 2: Simplified model of a Vectored Thrust Aircraft.*

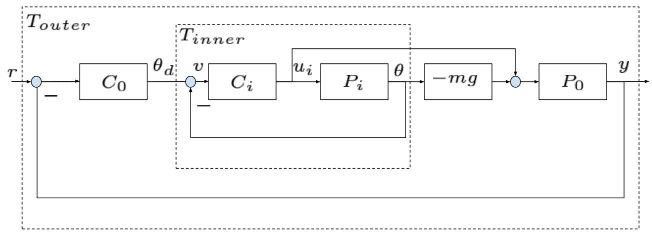


*Figure 3: Simplified model of a Vectored Thrust Aircraft.*

| **Symbol** | Description | Value |
|---|---|---|
| $m$ | Vehicle mass | 4.0 kg |
| $J$ | Vehicle inertia, $\varphi_3$ axis | 0.00475 kg m$^2$ |
| $r$ | Force moment arm | 25 cm |
| $c$ | Damping coefficient | 0.05 kg m/s |
| $g$ | Gravitational constant | 9.8 m/s$^2$ |

*Table 1: Summary of the aircraft model parameters.*

To control the lateral dynamics of the aircraft, we make use of a ïnner/outerloop design methodology as illustreted in figure 3. This diagram shows the process dynamics and controller divided into two components:

- an *inner loop* consisting of the roll dynamics
- an *outer loop* consisting of the lateral position dynamics and controller.

The approach that we take is to design a controller $C_i$ for the inner loop so that the resulting closed loop $T_{inner}$ is accurate enough (close to 1) up to the required bandwidth of the outer loop. This technique is generally referred to as *cascade control design*.
The lateral controller $C_0$ uses the approximation that the roll angle can be directly used as the input (after multiplication with the constant term $-mg$) for the lateral dynamics defined by the plant $P(s)$, being $T_{inner} \approx 1$ in the frequency range of interest. For this assumption to be true, we require the roll control system to be 10 times faster than the outer loop which has a bandwidth of 1 rad/s. Furthermore, we want the inner loop phase margin to be greater than 40°.

**a)** Design a controller which achieves the desired requirements. The inner loop plant is

$$P_i = \frac{r}{Js^2 + cs} \tag{1}$$

### Exercise 5 (Homework: A Dynamic Compensator)

Let Figure 1 model a tracking system with

$$P(s) = \frac{1}{s(0.1s + 1)}.$$

Design a dynamic compensator, $C(s)$, which meets the following performance specifications:

**a)** The steady-state error following ramp inputs must not exceed 2%.

**b)** The error in response to sinusoidal inputs up to 5 rad/sec should not exceed about 5%.

**c)** The crossover frequency should be about 50 rad/sec to meet bandwidth requirements while limiting the response to high-frequency noise.

**d)** The phase margin should be at least 50°, and the ratio of the break frequencies of the lead / lag elements should not exceed 5 to limit noise effects.