Lecture 1: Feedback Control Loop

1 Loop Transfer function

The standard feedback control system structure[1] is depicted in Figure 1. This represen-

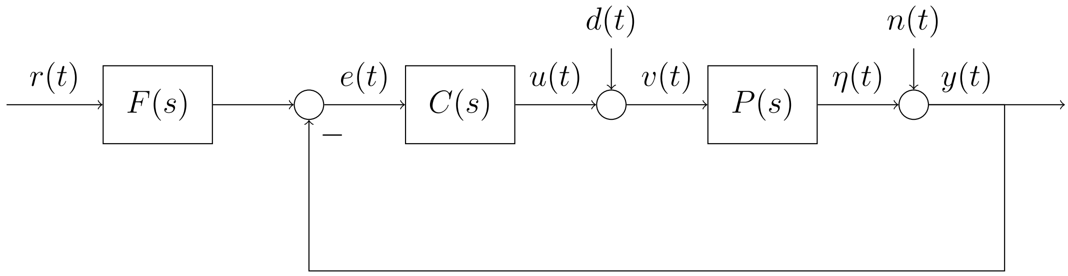

Figure 1: Standard feedback control system structure.

tation will be the key element for your further control studies.
The **plant** $P(s)$ represents the system you want to control: let's imagine a Duckiebot. The variable $u(t)$ represents the **real input** that is given to the system and $d(t)$ some **disturbance** that is applyied to it. These two elements toghether, can be resumed into an actuator for the Duckiebot example. The signal $v(t)$ represents the **disturbed input**. The signal $\eta(t)$ describes the **real output** of the system. The variable $y(t)$, instead, describes the **measured output** of the system, which can be eventually measured by a sensor with some **noise** $n(t)$. In the case of the Duckiebot, this can correspond to the position of the vehicle and its orientation (pose). The **feedback controller** $C(s)$ makes sure that the **tracking error** $e(t)$ between the measured output and the **reference** $r(t)$ approaches zero. $F(s)$ represents the **feedforward controller** of the system.

Remark. Note the notation: signals in time domain are written with small letters, such as $n(t)$. Transfer functions in frequency domain (Laplace/Fourier transformed) are written with capital letters, such as $P(s)$.

1.1 Loop Transfer Functions

Why transfer functions? In order to make the analysis of such a system easier, the loop transfer functions are defined. In fact, it is worth transforming a problem from the time domain into the frequency domain, solve it, and back transform it into time domain. The main reason behind this is that *convolutions* (computationally complex operations which relate signals) are multiplications (through Laplace/Fourier transformation) in the frequency domain.

1.1.1 The Gang of Six

The **loop gain** $L(s)$ is the open-loop transfer function defined by

$$L(s) = P(s)C(s). \tag{1.1}$$

[1]Note that this does not correspond to the most general form of a feedback control loop

The **sensitivity** $S(s)$ is the closed-loop transfer function defined by

$$\begin{aligned} S(s) &= \frac{1}{1+L(s)} \\ &= \frac{1}{1+P(s)C(s)}. \end{aligned} \tag{1.2}$$

Remark. Note that the sensitivity gives measure of the influence of disturbances d on the output y.

The **complmentary sensitivity** $T(s)$ is the closed-loop transfer function defined by

$$\begin{aligned} T(s) &= \frac{L(s)}{1+L(s)} \\ &= \frac{P(s)C(s)}{1+P(s)C(s)}. \end{aligned} \tag{1.3}$$

It can be shown that

$$S(s) + T(s) = 1. \tag{1.4}$$

Recalling that a performant controller minimizes the difference between the reference $R(s)$ and the output $Y(s)$, one can write this difference as an error $E(s)$, computed as

$$\begin{aligned} E(s) &= F(s)R(s) - Y(s) \\ &= F(s)R(s) - (\eta(s) + N(s)) \\ &= F(s)R(s) - (P(s)V(s) + N(s)) \\ &= F(s)R(s) - (P(s)(D(s) + U(s)) + N(s)) \\ &= F(s)R(s) - P(s)D(s) - N(s) - P(s)U(s) \\ &= F(s)R(s) - P(s)D(s) - N(s) - P(s)C(s)E(s). \end{aligned} \tag{1.5}$$

Furthermore, recalling that we started from $E(s)$, one gets the new equation

$$\begin{aligned} E(s) &= F(s)R(s) - P(s)D(s) - N(s) - P(s)C(s)E(s) \\ (1+P(s)C(s))E(s) &= F(s)R(s) - P(s)D(s) - N(s) \\ E(s) &= \frac{F(s)}{1+P(s)C(s)}R(s) - \frac{P(s)}{1+P(s)C(s)}D(s) - \frac{1}{1+P(s)C(s)}N(s). \end{aligned} \tag{1.6}$$

This procedure can be applied to every pair of signals of the feedback loop depicted in Figure 1. The following equations can be derived:

$$\begin{pmatrix} Y(s) \\ \eta(s) \\ V(s) \\ U(s) \\ E(s) \end{pmatrix} = \frac{1}{1+P(s)C(s)} \begin{pmatrix} P(s)C(s)F(s) & P(s) & 1 \\ P(s)C(s)F(s) & P(s) & -P(s)C(s) \\ C(s)F(s) & 1 & -C(s) \\ C(s)F(s) & -P(s)C(s) & -C(s) \\ F(s) & -P(s) & -1 \end{pmatrix} \cdot \begin{pmatrix} R(s) \\ D(s) \\ N(s) \end{pmatrix}. \tag{1.7}$$

Exercise 1. Derive all the other relations reported in Equation (1.7) on your own.

As you can notice, many terms in the relations introduced in Equation (1.7), are repeated. Using the defined sensitivity function $S(s)$ (Equation (1.2) and the complementary sensitivity $T(s)$ (Equation 1.3), one can define four new important transfer functions. The **load sensitivity** is defined as

$$P(s)S(s) = \frac{P(s)}{1 + P(s)C(s)}, \tag{1.8}$$

and gives us an intuition on how does the disturbance affect the output. The **noise sensitivity** is defined as

$$C(s)S(s) = \frac{C(s)}{1 + P(s)C(s)}, \tag{1.9}$$

and gives us an intuition on how does the noise affect the input. Moreover, one can define two more useful transfer functions:

$$C(s)F(s)S(s) = \frac{C(s)F(s)}{1 + P(s)C(s)}, \qquad T(s)F(s) = \frac{P(s)C(s)F(s)}{1 + P(s)C(s)}. \tag{1.10}$$

The new introduced four transfer functions toghether with the sensitivity and the complementary sensitivity functions, describe the so called *gang of six*.

1.1.2 The Gang of Four

The special case where $F(s) = 1$ (i.e., no feedforward), leads to the equivalence of some of the defined transfer functions. In particular, we are left with four transfer functions:

$$\begin{aligned}
S(s) &= \frac{1}{1 + P(s)C(s)} && \text{sensitivity function,} \\
T(s) &= \frac{P(s)C(s)}{1 + P(s)C(s)} && \text{complementary sensitivity function,} \\
P(s)S(s) &= \frac{P(s)}{1 + P(s)C(s)} && \text{load sensitivity function,} \\
C(s)S(s) &= \frac{C(s)}{1 + P(s)C(s)} && \text{noise sensitivity function.}
\end{aligned} \tag{1.11}$$

At this point one may say: I can define these new transfer functions, but why are they necessary? Let's illustrate this through an easy example.

Example 1. Imagine to deal with a plant $P(s) = \frac{1}{s-1}$ and that you control it through a PID controller of the form $C(s) = k \cdot \frac{(s-1)}{s}$. You can observe that the plant has a pole $s = 1$, which makes it unstable. If one computes the classic transfer functions learned in Control Systems I (Equations (1.1), (1.2), (1.3)), gets

$$\begin{aligned}
L(s) &= C(s)P(s) = \frac{1}{s-1} \cdot k \cdot \frac{(s-1)}{s} = \frac{k}{s}, \\
S(s) &= \frac{1}{1 + L(s)} = \frac{s}{s+k}, \\
T(s) &= \frac{L(s)}{1 + L(s)} = \frac{k}{s+k}.
\end{aligned} \tag{1.12}$$

You may notice, that none of these transfer functions contains the important information about the unstability of the plant. However, this information is crucial: if one computes the rest of the gang of four, one gets

$$\begin{aligned} P(s)S(s) &= \frac{\frac{1}{s-1}}{1+\frac{k}{s}} = \frac{1}{(s-1)(s+k)}, \\ C(s)S(s) &= \frac{k \cdot \frac{(s-1)}{s}}{1+\frac{k}{s}} = \frac{k(s-1)}{s+k}. \end{aligned} \tag{1.13}$$

These two transfer functions still contain the problematic term and are extremely useful to determine the influence of the unstable pole on the system, because they explicitly show it.

Exercise 2. Which consequence does the application of a small disturbance d on the system have?

1.1.3 Relations to Performance

By looking at the feedback loop in Figure 1, one can introduce a new variable

$$\varepsilon(t) = r(t) - \eta(t), \tag{1.14}$$

which represents the error between the reference signal and the real plant output. One can show, that this error can be written as

$$\varepsilon(s) = S(s)R(s) - P(s)S(s)D(s) + T(s)N(s). \tag{1.15}$$

From this equation one can essentially read:

- For a good reference tracking and disturbance attenuation, one needs a small $S(s)$ (or high $L(s)$).
- For a good noise rejection, one needs a small $T(s)$ (or small $L(s)$).

Exercise 3. Derive with what you learned in this chapter Equation (1.15).

1.1.4 Feed Forward

The feedforward technique complements the feedback one. If on one hand feedback is error based and tries to compensate unexpected or unmodeled phenomena, such as disturbances, noise, model uncertainty, the feedforward technique works well if we have some *knowledge* of the system (i.e. disturbances, plant, reference). Let's illustrate the main idea behind this concept through an easy example.

Example 2. The easiest example for this concept, is the one of perfect control. Imagine to have a system as the one depicted in Figure 2.
This is also known as perfect control/plant inversion, where we want to find an input $u(t)$, such that $y(t) = r(t)$. One can write

$$Y(s) = P(s)U(s) \tag{1.16}$$

and hence

$$R(s) = P(s)U(s) \Rightarrow U(s) = P(s)^{-1}R(s). \tag{1.17}$$

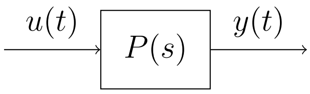

Figure 2: Standard perfect control system structure.

This is not possible when:

- The plant $P(s)$ has right-hand side poles, (unstable inverse).
- There are time delays, (non causal inverse): *how much of the future output trajectory information we need in order to perform the desired output tracking?*
- More poles than zeros, (unrealizable inverse).
- Model uncertainty, (unknown inverse).

But what does it mean for a system to be realizable or causal? Let's illustrate this with an example.

Example 3. If one has a transfer function with a number of zeros bigger than the number of poles, this represents pure differentiators, which are not causal. Imagine to deal with the transfer function

$$P(s) = \frac{(s+2)(s+3)}{s+1}. \tag{1.18}$$

This transfer function has two zeros and one pole. This can be rewritten as

$$\begin{aligned} P(s) &= \frac{(s+2)(s+3)}{s+1} \\ &= \frac{s^2+5s+6}{s+1} \\ &= \frac{s(s+1)+4s+6}{s+1} \\ &= s + \frac{4s+6}{s+1}, \end{aligned} \tag{1.19}$$

where s is a pure differentiator. A pure differentiator's transfer function can be written as the ratio of an output and an input:

$$G(s) = s = \frac{Y(s)}{U(s)}, \tag{1.20}$$

which describes the time domain equation

$$y(t) = \dot{u}(t) = \lim_{\delta_t \to 0} \frac{u(t+\delta_t) - u(t)}{\delta_t}, \tag{1.21}$$

which confirms us that this transfer function must have knowledge of future values of the input $u(t)$ (from $u(t+\delta_t)$) in order to react with the current output $y(t)$. This is per definition not physical and hence not realizable, not causal.

2 Control Objectives

In this section we are going to present the standard control objectives and relate them to what you learned in the course Control Systems I.
But what are the real objectives of a controller? We can subdivide them into four specific needs:

1. **Nominal Stability**: Is the closed-loop interconnection of a nominal plant and a controller stable?

2. **Nominal Performance**: Does the closed-loop interconnection of a nominal plant and a controller achieve specific performance objectives?

3. **Robust Stability**: Is the closed-loop interconnection of *any* disturbed nominal plant and a controller stable?

4. **Robust Performance**: Does the closed-loop interconnection of *any* plant and a controller achieve specific perfomance objectives?

One can essentially subdivide the job of a control engineer into two big tasks:

(I) **Analysis**: Given a controller, how can we check that the objectives above are satisfied?

(II) **Synthesis**: Given a plant, how can we design a controller that achieves the objectives above?

Let's analyse the objectives of a controller with respect to their relation to these two tasks.

2.1 Nominal Stability

During the course Control Systems I, you learned about different stability concepts. Moreover, you have learned the differences between internal and external stability: let's recall them here. Consider a generic nonlinear system defined by the dynamics

$$\dot{x}(t) = f(x(t)), \quad t \in \mathbb{R}, \quad x(t) \in \mathbb{R}^n, \quad f : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^n. \tag{2.1}$$

Definition 1. A state $\hat{x} \in \mathbb{R}^n$ is called an *equilibrium* of system (2.1) if and only if $f(\hat{x}) = 0 \quad \forall t \in \mathbb{R}$.

2.1.1 Internal/Lyapunov Stability

Internal stability, also called Lyapunov stability, characterises the stability of the trajectories of a dynamic system subject to a perturbation near the to equilibrium. Let now $\hat{x} \in \mathbb{R}^n$ be an equilibrium of system (2.1).

Definition 2. An equilibrium $\hat{x} \in \mathbb{R}^n$ is said to be *Lyapunov* stable if

$$\forall \varepsilon > 0, \quad \exists \delta > 0 \quad \text{s.t. } \|x(0) - \hat{x}\| < \delta \Rightarrow \|x(t) - \hat{x}\| < \varepsilon. \tag{2.2}$$

In words, an equilibrium is said to be *Lyapunov* stable if for any bounded initial condition and zero input, the state remains bounded.

Definition 3. An equilibrium $\hat{x} \in \mathbb{R}^n$ is said to be *asymptotically stable* in $\Omega \subseteq \mathbb{R}^n$ if it is Lyapunov stable and attractive, i.e. if

$$\lim_{t\to\infty} (x(t) - \hat{x}) = 0, \quad \forall x(0) \in \Omega. \tag{2.3}$$

In words, an equilibrium is said to be asymptotically stable if, for any bounded initial condition and zero input, the state converges to 0.

Definition 4. An equilibrium $\hat{x} \in \mathbb{R}^n$ is said to be *unstable* if it is not stable.

Remark. Note that stability is a property of the equilibrium and not of the system in general.

2.1.2 External/BIBO Stability

External stability, also called BIBO stability (Bounded Input-Bounded Output), characterises the stability of a dynamic system which for bounded inputs gives back bounded outputs.

Definition 5. A signal $s(t)$ is said to be bounded, if there exists a finite value $B > 0$ such that the signal magnitude never exceeds B, that is

$$|s(t)| \leq B \quad \forall t \in \mathbb{R}. \tag{2.4}$$

Definition 6. A system is said to be *BIBO-stable* if

$$\|u(t)\| \leq \varepsilon \quad \forall t \geq 0, \text{ and } x(0) = 0 \Rightarrow \|y(t)\| < \delta \quad \forall t \geq 0, \quad \varepsilon, \delta \in \mathbb{R}. \tag{2.5}$$

In words, for any bounded input, the output remains bounded.

2.1.3 Stability for LTI Systems

Above, we focused on general nonlinear system. However, in Control Systems I you learned that the output $y(t)$ for a LTI system of the form

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t), \end{aligned} \tag{2.6}$$

can be written as

$$y(t) = Ce^{At}x(0) + C\int_0^t e^{A(t-\tau)}Bu(\tau)\mathrm{d}(\tau) + Du(t). \tag{2.7}$$

The transfer function relating input to output is a rational function

$$P(s) = C(s\mathbb{I} - A)^{-1}B + D = \frac{b_{n-1}s^{n-1} + b_{n-2}s^{n-2} + \ldots + b_0}{s^n + a_{n-1}s^{n-1} + \ldots + a_0} + d. \tag{2.8}$$

Furthermore, it holds:

- The zeros of the numerator of Equation (2.8) are the *zeros* of the system, i.e. the values s_i which fulfill
$$P(s_i) = 0. \tag{2.9}$$

- The zeros of the denominator of Equation (2.8) are the *poles* of the system, i.e. the values s_i which fulfill $\det(s_i\mathbb{I} - A) = 0$, or, in other words, the eigenvalues of A.

One can show, that the following Theorem holds:

Theorem 1. The equilibrium $\hat{x} = 0$ of a linear time invariant system is stable if and only if the following two conditions are met:

1. For all $\lambda \in \sigma(A), \quad \text{Re}(\lambda) \leq 0$.
2. The algebraic and geometric multiplicity of all $\lambda \in \sigma(A)$ such that $\text{Re}(\lambda) = 0$ are equal.

Remark. We won't go into the proof of this theorem, because beyond the scope of the course. As an intuition, however, one can look at Equation (2.7). As you learned in Linear Algebra, the matrix exponential computation can be simplified with help of the diagonalization of a matrix. Moreover, if matrix A is diagonalizable, you can derive a form where you are left with exponential terms of the eigenvalues of A on the diagonal. If these eigenvalues are bigger than 0, the exponentials, which depends on time, diverge. If these eigenvalues are smaller than zero, the exponentials converge to 0 (asymptotically stable behaviour). In the case of zero eigenvalues, the exponentials converge, but not to 0 (stable behaviour). If the matrix A is not diagonalizable, i.e. the algebraic and the geometric multiplicity of an eigenvalue do not coincide, one should recall the Jordan form. In this case, some polynomial terms may be multiplied with the exponential ones in the diagonal: this could lead to unstable behaviour (stable vs. unstable because of 0 eigenvalue). For the rigorous proof of the Theorem, go to `https://en.wikibooks.org/wiki/Control_Systems/State-Space_Stability`.

2.1.4 Analysis

Which tools do we already know in order to analyze nominal stability? In the course Control Systems I you learned about

- **Root locus.** In order to recall the root locus method, have a look at Example 4.15, page 123 in [1].
- **Bode diagram:** the Bode diagram is a frequency explicit representation of the magnitude $|L(j\omega)|$ and the phase $\angle(L(j\omega))$ of a complex number $L(j\omega)$. Because of graphic reasons, one uses decibel (dB) as unit for the amplitude and degrees as unit for the phase. As a reminder, the conversion reads

$$\mathbb{X}_{\text{dB}} = 20 \cdot \log_{10}(\mathbb{X}), \quad \mathbb{X} = 10^{\frac{\mathbb{X}_{\text{dB}}}{20}}. \tag{2.10}$$

  Moreover, stable and unstable poles and zeros have specific consequences on the Bode diagram:

  – Poles cause a gradient of $-20\frac{\text{dB}}{\text{decade}}$ in the amplitude:

Pole	stable	unstable
Magnitude	$-20\frac{\text{dB}}{\text{decade}}$	$-20\frac{\text{dB}}{\text{decade}}$
Phase	$-90°$	$90°$

– Zeros cause a gradient of $20\frac{\text{dB}}{\text{decade}}$ in the amplitude:

Zero	stable	unstable
Magnitude	$20\frac{\text{dB}}{\text{decade}}$	$20\frac{\text{dB}}{\text{decade}}$
Phase	$90°$	$-90°$

An example of a Bode diagram is depicted in Figure 3.

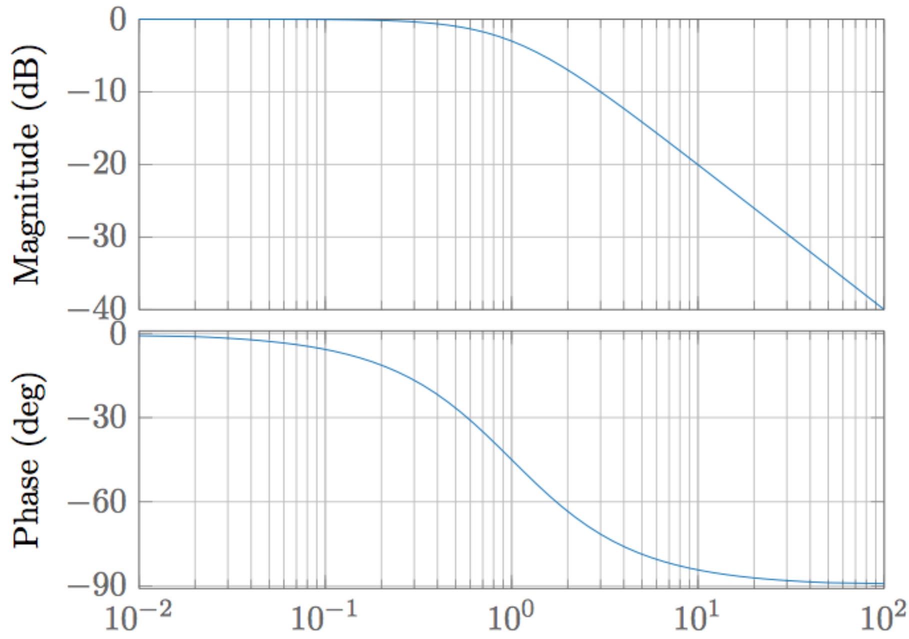

Figure 3: Example of a Bode diagram.

- **Nyquist diagram:** the Nyquist diagram is a frequency implicit representation of the complex number $L(j\omega)$ in the complex plane. An example of a Nyquist diagram is shown in Figure 4.

  Remark. In order to draw the Nyquist diagram, some useful limits can be computed:

$$\lim_{\omega\to 0} L(j\omega), \quad \lim_{\omega\to\infty} L(j\omega), \quad \lim_{\omega\to\infty} \angle L(j\omega). \tag{2.11}$$

- **Nyquist theorem:** a closed-loop system $T(s)$ is asymptotically stable if

$$n_c = n_+ + \frac{n_0}{2} \tag{2.12}$$

holds, where

  – n_c: Number of mathematical positive encirclements of $L(s)$ about critical point -1 (counterclockwise).

  – n_+: Number of unstable poles of $L(s)$ ($Re(\pi) > 0$).

  – n_0: Number marginal stable poles of $L(s)$ ($Re(\pi) = 0$).

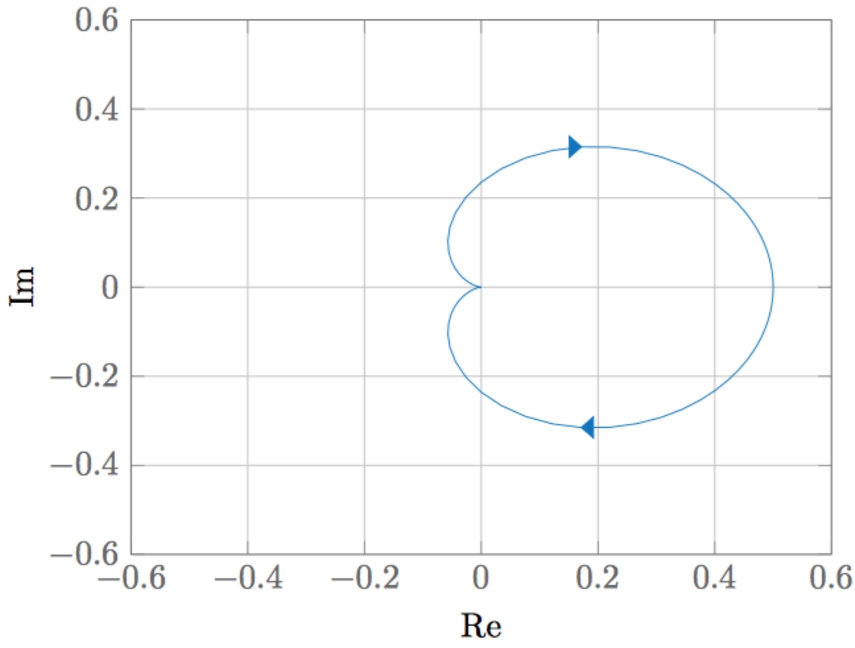

Figure 4: Example of a Nyquist diagram.

2.1.5 Synthesis: loop shaping

Plant inversion

As seen in the feedforward example, this method isn't indicated for non-minimum phase plants and for unstable plants: in those cases this would lead to non-minimum phase or unstable controllers. This method is indicated for *simple* systems for which it holds

- Plant is asymptotically stable.
- Plant is minimum phase.

The method is then based on a simple step:

$$L(s) = C(s) \cdot P(s) \Rightarrow C(s) = L(s) \cdot P(s)^{-1}. \tag{2.13}$$

The choice of the loop gain is free: it can be chosen such that it meets the desired specifications.

Loop shaping for Non-minimum Phase systems

A non-minimum phase system shows a wrong response: a change in the input results in a change in sign, that is, the system initially lies. Our controller should therefore be *patient* and for this reason one should use a *slow* control system. This is obtained by a crossover frequency that is smaller than the non-minimum phase zero. One begins to design the controller with a **PI-Controller**, which has the form

$$C(s) = k_p \cdot \frac{T_i \cdot s + 1}{T_i \cdot s}. \tag{2.14}$$

The parameters k_p and T_i can be chosen such that the loop gain $L(s)$ meets the known specifications. One can reach better robustness with **Lead/Lag** elements of the form

$$C(s) = \frac{T \cdot s + 1}{\alpha \cdot T \cdot s + 1}. \tag{2.15}$$

where $\alpha, T \in \mathbb{R}^+$. One can understand the Lead and the Lag elements as

- $\alpha < 1$: **Lead-Element**:
  - Phase margin increases.
  - Loop gain increases.
- $\alpha > 1$: **Lag-Element**:
  - Phase margin decreases.
  - Loop gain decreases.

As one can see in Figure 5 and Figure 6, the maximal benefits are reached at frequencies ($\hat{\omega}$), where the drawbacks are not yet fully developed.

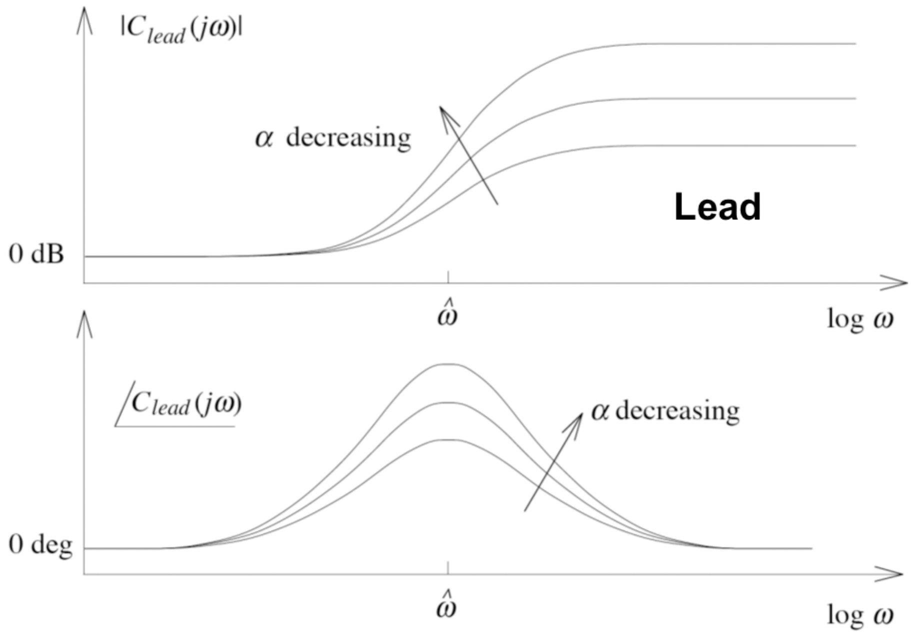

Figure 5: Bodeplot of the Lead Element

The element's parameters can be calculated as

$$\alpha = \left(\sqrt{\tan^2(\hat{\varphi}) + 1} - \tan(\varphi)\right)^2 = \frac{1 - \sin(\hat{\varphi})}{1 + \sin(\hat{\varphi})} \tag{2.16}$$

and

$$T = \frac{1}{\hat{\omega} \cdot \sqrt{\alpha}}. \tag{2.17}$$

where $\hat{\omega}$ is the desired **center frequency** and $\hat{\varphi} = \varphi_{new} - \varphi$ is the desired **maximum phase shift** (in rad).

The classic loop-shaping method reads:

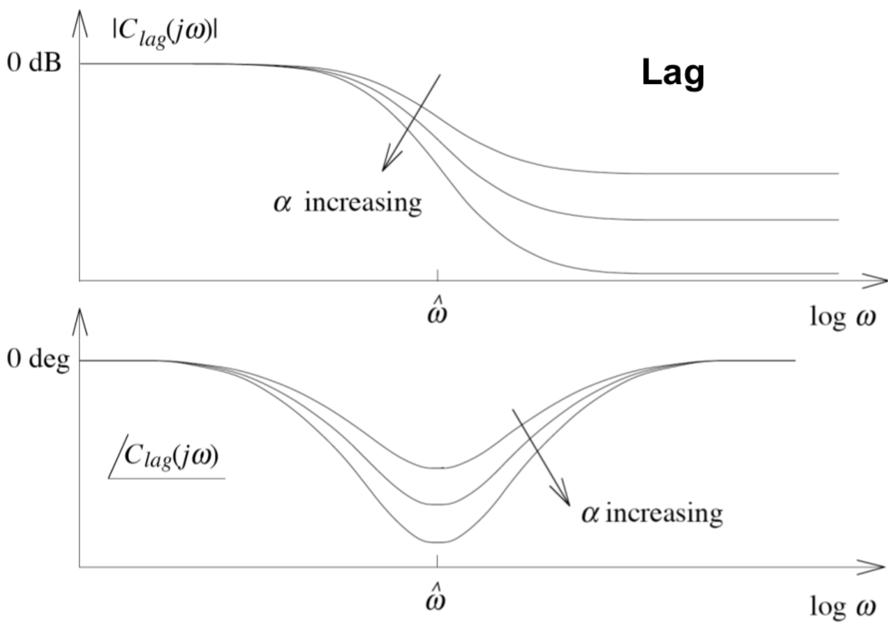

Figure 6: Bodeplot of the Lag Element

1. Design of a PI(D) controller.
2. Add Lead/Lag elements where needed[2]
3. Set the gain of the controller k_p such that we reach the desired crossover frequency.

Loop shaping for unstable systems

Since the Nyquist theorem should always hold, if it isn't the case, one has to design the controller such that $n_c = n_+ + \frac{n_0}{2}$ is valid. To remember is: stable poles *decrease* the phase by 90° and minimum phase zeros *increase* the phase by 90°

Realizability

Once the controller is designed, one has to look if this is really feasible and possible to implement. That is, if the number of poles is equal or bigger than the number of zeros of the system. If that is not the case, one has to add poles at high frequencies, such that they don't affect the system near the crossover frequency. One could e.g. add to a PID controller a *Roll-Off Term* as

$$C(s) = \underbrace{k_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right)}_{\text{PID Controller}} \cdot \underbrace{\frac{1}{(\tau \cdot s + 1)^2}}_{\text{Roll-Off Term}}. \tag{2.18}$$

[2] $L(j\omega)$ often suits not the learned requirements

2.2 Performance

Under performance, one can understand two specific tasks:

- **Regulation/disturbance rejection:** Keep a setpoint despite disturbances, i.e. keep $y(t)$ at $r(t)$. As an example, you can imagine you try to keep driving your Duckiebot at a constant speed towards a cooling fan.
- **Reference Tracking**: Reference following, i.e. let $y(t)$ track $r(t)$. As an example, imagine a luxury Duckiebot which carries Duckiecustomers: a temperature controller tracks the different temperatures which the different Duckiecustomers may want to have in the Duckiebot.

2.3 Robustness

All models are wrong, but some of them are useful. (2.19)

A control system is said to be **robust** when it is insensive to model uncertainties. But why should a model have uncertainties? Essentially, for the following reasons:

- **Aging**: the model that was good a year ago, maybe is not good now. As an example, think of the wheel deterioration which could cause slip in a Duckiebot.
- **Poor system identification:** there are entire courses dedicated to the art of system modeling (even at ETH). It is not possible not to come to assumptions, which simplify your real system to something that does not *perfectly* describe that.

References

[1] Karl Johan Amstroem, Richard M. Murray *Feedback Systems for Scientists and Engineers.* Princeton University Press, Princeton and Oxford, 2009.

[2] Sigurd Skogestad, *Multivariate Feedback Control.* John Wiley and Sons, New York, 2001.