Control Systems I

Lecture 14: RHP limits and Implementation

Readings: notes

Jacopo Tani

Institute for Dynamic Systems and Control
D-MAVT
ETH Zürich

December 21, 2018
<table>
<thead>
<tr>
<th>#</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sept. 21</td>
<td>Introduction, Signals and Systems</td>
</tr>
<tr>
<td>2</td>
<td>Sept. 28</td>
<td>Modeling, Linearization</td>
</tr>
<tr>
<td>3</td>
<td>Oct. 5</td>
<td>Analysis 1: Time response, Stability</td>
</tr>
<tr>
<td>4</td>
<td>Oct. 12</td>
<td>Analysis 2: Diagonalization, Modal coordinates</td>
</tr>
<tr>
<td>5</td>
<td>Oct. 19</td>
<td>Transfer functions 1: Definition and properties</td>
</tr>
<tr>
<td>6</td>
<td>Oct. 26</td>
<td>Transfer functions 2: Poles and Zeros</td>
</tr>
<tr>
<td>7</td>
<td>Nov. 2</td>
<td>Analysis of feedback systems: internal stability, root locus</td>
</tr>
<tr>
<td>8</td>
<td>Nov. 9</td>
<td>Frequency response</td>
</tr>
<tr>
<td>9</td>
<td>Nov. 16</td>
<td>Analysis of feedback systems 2: the Nyquist condition</td>
</tr>
<tr>
<td>10</td>
<td>Nov. 23</td>
<td>Frequency response II</td>
</tr>
<tr>
<td>11</td>
<td>Nov. 30</td>
<td>Specifications for feedback systems</td>
</tr>
<tr>
<td>12</td>
<td>Dec. 7</td>
<td>PID control</td>
</tr>
<tr>
<td>13</td>
<td>Dec. 14</td>
<td>Loop Shaping</td>
</tr>
<tr>
<td>14</td>
<td>Dec. 21</td>
<td><strong>RHP Limits and Implementation</strong></td>
</tr>
</tbody>
</table>
Point of the Situation

- **Analysis**: given a controller, determine properties of the closed loop system from knowledge of open loop transfer function (Root Locus, Bode, Nyquist).
- **Synthesis**: given a plant model (or not?) find a controller that provides:
  - **Stability**: closed loop poles have negative real part;
  - **Performance**: given time domain and/or frequency domain specifications are met (e.g.: steady-state error, rise time, overshoot, bandwidth, etc.);
  - **Robustness**: margin of gain and margin of phase are satisfactory.

- Proportional-Integrative-Derivative control (PID): very popular in practice.

- Loop shaping: powerful approach. “Steer” the magnitude of the open loop frequency response to avoid frequency domain obstacles (e.g., lead/lag compensators)
Command tracking/disturbance rejection vs. noise rejection

- **Sensitivity function** $S(s) = \frac{e(s)}{r(s)} = \frac{e(s)}{d(s)} = \frac{1}{1 + L(s)} = \frac{1}{1 + P(s)C(s)}$

- **Tracking and disturbance rejection**: $S(s)$ "small", i.e., $L(s)$ "large".

- **Complementary sensitivity** $T(s) = \frac{y(s)}{n(s)} = \frac{L(s)}{1 + L(s)}$.

- **Noise attenuation**: $T(s)$ "small", i.e., $L(s)$ "small".

- But $T(s) + S(s) = 1$ so they cannot be "small" at the same time!

- Solved by meeting disturbances and tracking specs at "low" frequencies, and noise attenuation specs at "high" frequency.
Recap: Bode-plot “obstacle course”

- Load disturbance attenuation
- High frequency measurement noise
- Robustness

\[ \omega_{gc} \]

\[ W_1 \]

\[ W_2^{-1} \]
A general procedure for open-loop stable systems

Proceed from “the left”, i.e., from low frequencies to high frequencies.

1. Figure out how many integrators are needed in \( C(s) \). This depends on the order of the ramp that must be tracked with zero steady-state error.

2. Fix the gain in such a way that the low-frequency asymptote clears the command-tracking/disturbance specification (the low-frequency Bode obstacle).

3. Add terms of the form \((1 + \tau s)\) at the numerator or denominator in such a way that the Bode magnitude plot intersects the 0dB line with a slope of about 20 dB/s (this would give you a 90-degree phase margin. Poles steer “down”, zeros steer “up.” Note that normalizing the zero-order term to 1 makes it so that these do not affect the Bode plot on the left of the pole/zero.

4. Past the crossover, add poles as needed to clear the high-frequency Bode obstacle (noise reduction/uncertainty).
A general iterative loop shaping design example

Consider the system with transfer function

\[ P(s) = 0.01 \frac{(s + 10)^2}{s(s^2 + s + 1)}; \]

Design a feedback control system such that

- The closed-loop is stable;
- The steady-state error to a unit step is zero.
- Commands up to 0.05 rad/s are followed with at most 1% error.
- The phase margin is no less than 40 degrees;
- Noise at frequencies higher than 100 rad/s is attenuated by at least 40 dB.
A general design example

\[ r(t) = \frac{t}{0!} \]

\[ \left| \frac{1 - G(s)}{G(s)} \right| \leq \frac{1}{100} \]

\[ \left| \frac{e^{-sT}}{s} \right| \leq \frac{1}{100} \]

\[ \left| \frac{1}{1 + L(s)} \right| \leq \frac{1}{100} \]

\[ \left| L(s) \right| \leq 100 \]

\[ \omega_m = 100 \]

\[ \omega_c = 2 \]

\[ \text{Magnitude (dB)} \]

\[ \text{Magnitude (dB)} \]

\[ \text{Phase (deg)} \]

\[ \text{Phase (deg)} \]

\[ \text{Frequency (rad/s)} \]

\[ \text{Frequency (rad/s)} \]

\[ L(s) = 100 R_1(s) \]

\[ L(s) = 100 R_1(s) \]
Can we extend the techniques learned so far to systems with poles/zeros in the right half plane? We can — but please always double check with Nyquist and/or the root locus.

That said... factor the plant transfer function as follows:

$$ P(s) = P_{mp}(s)D(s), $$

where

- $P_{mp}(s)$ is obtained by replacing all poles/zeros of $P(s)$ in the right half plane with their mirror image w.r.t. the imaginary axis.
- $D(s)$ contains all the poles/zeros of $P(s)$ in the right half plane, times the inverse all the mirror images introduced in $P_{mp}(s)$.

Example: if $P(s) = \frac{s-z}{s-p}$, $z, p > 0$, then $P_{mp}(s) = \frac{s+z}{s+p}$, and $D(s) = \frac{s-z}{s+z} \cdot \frac{s+p}{s-p}$. 
Pretend that you have to stabilize a “nice” plant $P_{mp}(s)$, but a little “Bode demon” placed $D(s)$ inside your control system.

Note that $|D(j\omega)| = 1$ for all $\omega$, i.e., $D(s)$ is an all-pass filter. Choose the sign of $D(s)$ so that the phase of $D(j\omega)$ is negative.

$D(s)$ does not alter the magnitude plot of $L(s)$, but messes up its phase plot—by introducing a possibly massive phase lag, and hence reducing the phase margin.
Loop shaping for non-minimum-phase system

\[ P(s) = \frac{1 - s}{s^2 + 2s + 2} \]

We have that \( D(s) = -\frac{s-1}{s+1} \). Note that \( \angle D(0) = 0 \), \( \angle D(j) = -90^\circ \), and \( \lim_{\omega \to \infty} \angle D(j\omega) = -180^\circ \).

- NMP zeros limit the crossover frequency and the feedback gain.
- In other words, NMP zeros limit how quickly the closed loop system can react.
Loop shaping for open-loop unstable system

\[ P(s) = \frac{1}{(s + 2)(s - 1)} \]

- We have that \( D(s) = \frac{s + 1}{s - 1} \). Note that \( \angle D(0) = -180^\circ \), \( \angle D(j) = -90^\circ \), and \( \lim_{\omega \to \infty} \angle D(j\omega) = 0^\circ \).

- Unstable open-loop poles force the crossover frequency to be large.
- In other words, unstable poles limit how slowly the control system can react.
Loop shaping summary

- Frequency domain specifications on the closed loop system can be represented on the Bode plot (of the open loop frequency response);

- Loop-shaping concept for control design

- Use standard elements such as
  - Gain: increase crossover frequency, reduce steady-state error, etc.
  - Lead: increase the phase margin at crossover frequency.
  - Lag: improve tracking and disturbance rejection.

  ... or use poles/zeros to “steer” the Bode plot through the obstacle course.

- Non-minimum-phase zeros limit the crossover frequency (closed-loop bandwidth)

- Open loop unstable poles require the crossover frequency to be higher.
Point of the situation

- At this point we have enough instruments to analyse a LTI system and design a controller to meet desired specifications in time or frequency domain.

- We have not exhausted what there is to say about control systems!

  - Implementation: How to implement a controller once we have designed it?

  - Nuisances: What have we not talked about, that is important to take into account in “real life”?

  - Next steps: where to go from here?
How to implement a compensator

- After a semester of hard work, you have learned how to design a dynamic compensator for controlling a SISO LTI system.

- This compensator usually takes the form of a transfer function, which we can write as

\[ C(s) = k + \frac{c_{n-1}s^{n-1} + c_{n-2}s^{n-2} + \ldots + c_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_0} \]

- How can we implement this mathematical object into a physical device?

  - In the past, and today for some specialized applications, this is implemented using custom analog electronics;
  
  - In typical applications today, this is implemented on a digital microcontroller, running a program implementing the desired compensator.

- How?
State space realization

- Recall that a realization of a transfer function

\[ C(s) = k + \frac{c_{n-1}s^{n-1} + c_{n-2}s^{n-2} + \ldots + c_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \ldots + a_0} \]

is the following set of \((A, B, C, D)\) matrices:

\[
A = \begin{bmatrix}
0 & 1 & 0 & 0 & \ldots & 0 \\
0 & 0 & 1 & 0 & \ldots & 0 \\
& \ddots & \ddots & \ddots & \ddots & \\
0 & 0 & 0 & 0 & \ldots & 1 \\
-a_0 & -a_1 & -a_2 & -a_3 & \ldots & -a_{n-1}
\end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix},
\]

\[
C = \begin{bmatrix} c_0 & c_1 & \ldots & c_{n-1} \end{bmatrix}, \quad D = [k]
\]

- Note: the “input” to this system is the error signal \(e = r - y\), the output is the control input \(u\).
Once we have a state-space realization, the implementation of a dynamic compensator is relatively straightforward, in functions that look like the following:

- Initialize the state to, e.g., \( x \leftarrow 0 \).
- Let \( dt \) be some small time interval.
- loop
  - Read the reference value \( r \).
  - Read the measured output value \( y \).
  - Compute the error \( e \).
  - Update the state as \( x \leftarrow x + (Ax + Be) \times dt \).
  - Compute the output as \( u = Cx + De \).
  - Send the command \( u \) to the actuators.
Examples

- Proportional control: \( C(s) = K \)

- PI control: \( C(s) = K_P + \frac{k_I}{s} \)

- Lead-lag: \( C(s) = k \frac{s + z}{s + p} \)
Non-proper transfer functions?

- What if we have a non-proper transfer function, e.g., including a derivative term $K_D s$.

- We need to change the code to approximate derivatives, e.g.:
  
  Initialize the state to, e.g., $x \leftarrow 0$.
  Initialize the error to, e.g., $e_{old} \leftarrow 0$.
  Let $dt$ be some small time interval.
  
  loop
  
  Read the reference value $r$.
  Read the measured output value $y$.
  Compute the error $e$.
  Update the state as $x \leftarrow x + (Ax + Be) \ast dt$.
  Compute the output as $u = Cx + De + K_D \ast (e - e_{old})/dt$.
  Send the command $u$ to the actuators.
  Store the error: $e_{old} \leftarrow e$. 
Choosing the sampling time $dt$

- In common implementations the commanded value for the input is maintained over the sampling period $dt$ ("Zero-Order Hold");

- The effect is similar to that of a time delay by $dt/2$.

- On the other hand, if our computer is not fast enough w.r.t. the system, we may expect large decreases in phase margin, and possibly instability.

- In those cases, if buying a faster computer is not an option, one can benefit from specialized discrete-time control design techniques, which are outside of the scope of this class.
Example

C(s) = 1 + 1/s, P(s) = 1/(10s + 1)
Time delay introduction

Time delays are ubiquitous in control systems:

- Delays are incurred when the controller is implemented on a computer, which needs some time to compute the appropriate control input, given a certain error. More in general, the evaluation of sensory information aimed at deciding the best course of action, will require a finite computation time.

- In some systems, delays may also be part of the physical plant.
  - When taking a shower, the water temperature is felt on the body after it has traveled through the valve, pipe, and shower head.
  - On an airplane, the effect of lift variation at the main wing are felt on the tail when the vortices shed by the wing reach the tail plane.
  - A rather extreme example is remote tele-operation: communication with a deep-space spacecraft or planetary rover may require several minutes.

How to take into account the effects of time delays in control system design and analysis?
The transfer function of a time delay

- A time delay is an operator that transforms an input signal \( t \mapsto u(t) \) into a delayed output signal \( t \mapsto y(t) \), with \( y(t) = u(t - T) \), where \( T \geq 0 \) is the amount of the delay.

- Clearly, this is a **linear operator**: the delayed version of a linear combination of signals is equal to the linear combination of the delayed signals.

- In order to compute the transfer function of this linear operator, consider an input of the form \( u(t) = e^{st} \). The output will be

\[
y(t) = e^{s(t - T)} = e^{-sT} u(t),
\]

and hence the transfer function of a delay of \( T \) seconds is \( e^{-sT} \).

- Notice that this is **NOT** a rational transfer function!
The frequency response of a time delay

- In terms of frequency response:

\[ |e^{j\omega T}| = 1, \quad \angle (e^{-j\omega T}) = -\omega T. \]

- This is not unexpected, since the time-delayed version of a sinusoid of unit amplitude and zero phase \( u(t) = \sin(\omega t) \) is another sinusoid, \( y = \sin(\omega(t - T)) \), with unit magnitude and a phase delayed by \( \omega T \).
Effects of time delays on the loop transfer function

- Next, we are going to consider the effects that time delays have on closed-loop stability, and discuss methods to take such effects into account when designing feedback control systems.

- Let us consider a system with loop transfer function \( L(s) = C(s)P(s) \), and include a time delay of \( T \) seconds (this can be either in the controller, or in the plant). We would get a new loop transfer function

\[
L'(s) = e^{-sT}L(s).
\]

- The frequency response of the system with the time delay is obtained from the “ideal” frequency response with no time delay, by shifting the phase back by \( \omega T \). In other words,

\[
|L'(j\omega)| = |L(j\omega)|, \quad \angle L'(j\omega) = \angle L(j\omega) - \omega T, \quad \forall \omega > 0.
\]
Example

- Consider a simple plant,
  \[ P(s) = \frac{1}{s + 1}, \]
  with a proportional controller \( C(s) = k. \)

- Hence, the ideal loop transfer function is
  \[ L(s) = \frac{k}{s + 1}. \]

A quick check with the root locus or the Nyquist plot will show that the feedback system would be stable for all \( k > 0 \) (in fact, for all \( k > -1 \)).

- What is the effect of a time delay on the closed-loop stability of such a system?
Example

- The loop transfer function in the presence of a time delay $T$ is
  \[ L'(s) = e^{-sT} \frac{k}{s + 1}. \]

- The polar plot of $L'$ is obtained from the polar plot of $L$ by “smearing” it: in practice all points $L(j\omega)$ must be rotated clockwise about the origin by an angle $\omega T$. 

---

J Tani, E. Frazzoli (ETH)  
Lecture 14: Control Systems I  
21/12/2018 27 / 35
Example

- It is easy to recognize that while the polar plot of $L$ never crosses the negative real axis, the polar plot of $L'$ crosses it an infinite number of times.

- In particular, the “first” crossing occurs when

$$\angle \left( \frac{k}{j\omega + 1} \right) - \omega T = -180^\circ,$$

i.e., when

$$\tan^{-1} \omega + \omega T = 180^\circ.$$

- The location of this point determines the gain margin, which will be finite for all $T > 0$ (it was infinite in the ideal case $T = 0$).
A similar analysis can be carried out on the Bode plot.
- The magnitude plot of $L'$ is exactly the same as the magnitude plot of $L$.
- The phase plot is obtained by adding the phase plot of $L$ to the phase plot of the time delay.

This lets you determine very easily the effect of a time delay on the phase margin. In fact, the following relationship holds:

$$\phi_{m,T} = \phi_{m,0} - \omega_c T,$$

where $\phi_{m,T}$ and $\phi_{m,0}$ are the phase margins with and without the time delay, respectively, and $\omega_c$ is the crossover frequency (which does not depend on the time delay).

The above equation summarizes the main effect of a time delay, that is a reduction of the phase margin.

Moreover, the phase margin reduction increases as the crossover frequency increases.
A design procedure:

The previous considerations suggest a procedure to design a feedback control system in the presence of time delay, as follows:

1. Design a feedback control system *ignoring* the time delay.

2. Check the effective phase margin, using (1). If the effective phase margin is too small (or negative, indicating closed-loop instability), redesign the controller according to one or both of these criteria:
   - Increase the phase at crossover, e.g., using a lead compensator.
   - Decrease the crossover frequency, e.g., reducing the gain, or possibly a phase lag controller (to maintain command-following performance).

3. Iterate until a satisfactory controller is found.
Alternative approaches to dealing with time delays

- We can approximate the time-delay transfer function to ratio of polynomials, so even root locus method can be used.

- A first choice would be a Taylor series expansion, which will take the form

\[ e^{-sT} = 1 - sT + \frac{1}{2}(sT)^2 - \frac{1}{6}(sT)^3 \ldots \]

- Let's truncate this series and maintain only the terms up to the second order in \((sT)\), i.e., let us write

\[ e^{-sT} \approx 1 - sT + \frac{1}{2}(sT)^2. \]

- Notice that this is a non-proper transfer function with two non-minimum phase zeros. This would be a good approximation for \(|sT| \ll 1\).

- However, the magnitude of the frequency response diverges for \(\omega \to \infty\), while we know that the magnitude of \(e^{-j\omega T}\) is always equal to one.
A better approximation can be obtained by using what is called a *Padé approximant*.

What we will do is approximate the exponential representing the time delay with a ratio of two polynomials.

For simplicity, let us limit ourselves to the ratio of first-order polynomials in $s$, i.e., let us write

$$e^{-sT} \approx k \frac{s + p}{s + q}.$$

By matching the rational approximation with appropriate number of terms in the Taylor expansion:

$$e^{-sT} \approx \frac{2/T - s}{2/T + s}.$$
Using the Padé approximation, we can represent a time delay on the root locus as a pole and zero, respectively at $\pm 2/T$. Notice the presence of a non-minimum-phase zero.

This approximation method is useful since it allows us to use the root locus method for control design, and gain the insight provided by it.

However, the results may not be accurate, and it is recommended that the root-locus method be used only as a back-of-the-envelope tool providing some additional insight in the control design process.

Remember: the only tool that would always provide you with correct answers in all cases when a time delay is present (including, e.g., unstable open-loop poles and non-minimum phase zeros) is the Nyquist plot.
Summary of the class

- State-space models of physical systems; equilibria and linearization.
- Response in the time domain (continuous and discrete time).
- Stability of a system (open loop)
- Closed-loop stability analysis, root locus and Routh-Criterion
- Transfer functions and frequency response
- The Nyquist stability condition
- Specifications for feedback systems
- Loop shaping using the Bode plot; poles/zeros, lead/lag, PID control
- Implementation, time delays
Next steps

- Laboratory Practice - blackboard ≠ hardware!
- Control Systems II - nuisances, state feedback, MIMO systems, "modern" techniques
- Optimal Control - focus on "cost" functions to minimize.
- Stochastic Systems - what if the best we can do is describe likelihoods of system dynamics? (very relevant in practice!)
- System Modeling - models are important.
- Recursive Estimation - a game changer for implementing powerful control methods in practice.
- Model Predictive Control - how to leverage system behavior predictions to control it better.