Control Systems I

Lecture 11: PID Control

Readings: A&M, Ch. 10, Guzzella, Chapter 11.2,

Emilio Frazzoli

Institute for Dynamic Systems and Control
D-MAVT
ETH Zürich

December 1st, 2017

Tentative schedule

#	Date	Topic
1	Sept. 22	Introduction, Signals and Systems
2	Sept. 29	Modeling, Linearization
3	Oct. 6	Analysis 1: Time response, Stability
4	Oct. 13	Analysis 2: Diagonalization, Modal coordinates.
5	Oct. 20	Transfer functions 1: Definition and properties
6	Oct. 27	Transfer functions 2: Poles and Zeros
7	Nov. 3	Analysis of feedback systems: internal stability, root locus
8	Nov. 10	Frequency response
9	Nov. 17	Analysis of feedback systems 2: the Nyquist condition
10	Nov. 24	Specifications for feedback systems
11	Dec. 1	PID Control
12	Dec. 8	Loop Shaping
13	Dec. 15	Implementation issues
14	Dec. 22	Robustness

Today's learning objectives

- Learn what a PID control is and how to design one:
    - Proportional control: what it is, pro's and con's
    - Derivative control: what it is, pro's and con's
    - Integral control: what it is, pro's and con't
- Tuning strategies for PID controllers.

A nice intro to PID control

- https://www.youtube.com/watch?v=4Y7zG48uHRo

Recall: Control Specifications

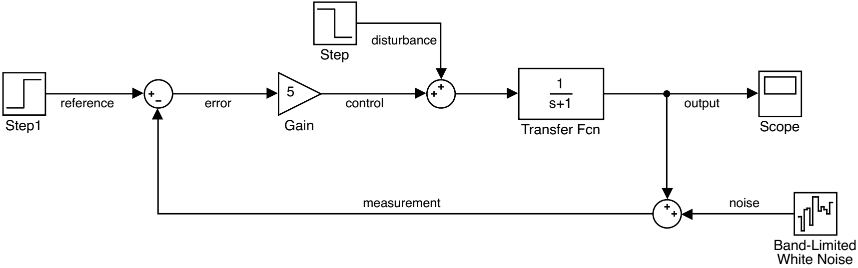
- **Type of the system** (order of ramp to track with zero steady-state error):
  - Number of integrators in $L(s)$
- **Time domain** specifications (max overshoot, settling time, rise time, ...):
  - Location of dominant closed-loop poles (damping ratio and real part)
- **Frequency domain** specifications (command tracking, disturbance/noise rejection, closed-loop bandwidth):
  - Bode obstacle course (low/high frequency)
  - Crossover frequency
- **Control synthesis:** how do we choose a feedback control system that achieves these objectives?

Controller design methods

What other methods do exist to design controllers $C(s)$ that meet design specifications? Many approaches, among them:

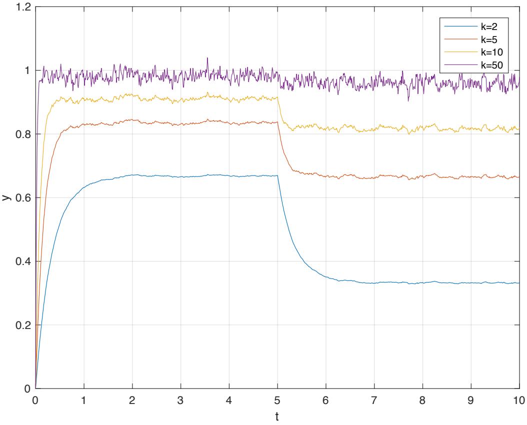
- PID, Loop Shaping, LQR, LQG-LTR, $H\infty$, Discrete-time optimal control, continuous-time optimal control, model predictive control,...
- Today we look at the most widely used approach for SISO systems: **PID control**.
- PID - control (proportional-integral-derivative control) is the most widely applied controller design because it is able to cope well with the majority of cases encountered in practice.

Proportional Control



- The control input tries to move the system in a direction that is opposite to the error, and is proportional to the error in magnitude.

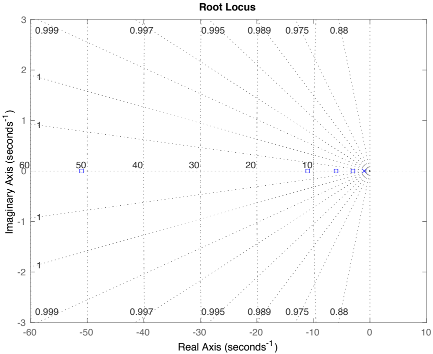
Proportional gain selection



As the proportional gain increases,

- The closed-loop system remains stable;
- The steady-state error decreases;
- The response becomes faster;
- The sensitivity to noise increases.

Proportional gain selection



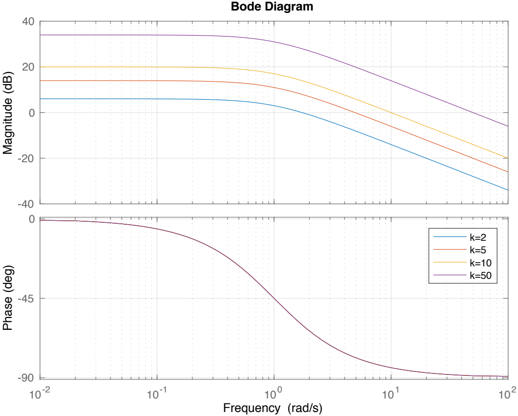
- Closed-loop transfer function

$$T(s) = \frac{L(s)}{1 + kL(s)} = \frac{1}{s + 1 + k},$$

i.e., the closed-loop pole is at $s = -1 - k$ (see root locus above).

- Steady-state error to a unit step: $e_{ss} = \lim_{s \to 0} \frac{1}{1+kL(s)} = \frac{1}{1+k}$

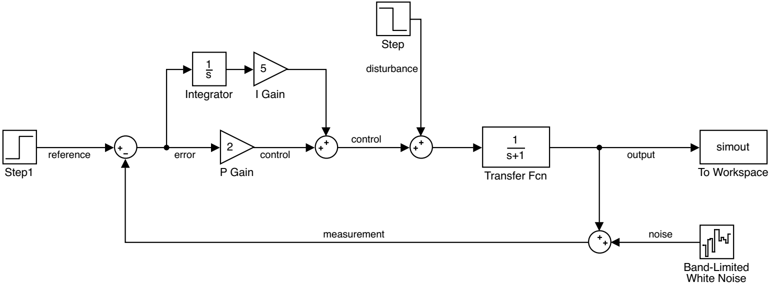
Proportional gain selection



As the proportional gain increases,

- Phase margin remains $> 90^\circ$;
- The crossover frequency increases;
- The low-frequency gain increases;
- The high-frequency gain increases;

Introducing an integrator

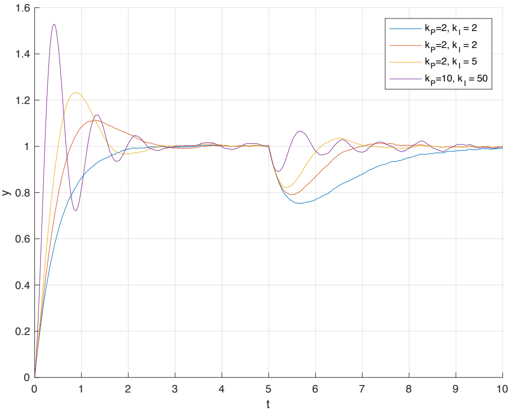


- Integrating the error allows one to detect potential "biases" in the system behavior.
- An integral control action tries to move the response in order to reduce the detected biases.
- PI control:

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau,$$

$$C(s) = k_P + \frac{k_I}{s} = \frac{k_P s + k_I}{s} = k_I \frac{k_P/k_I \cdot s + 1}{s}.$$

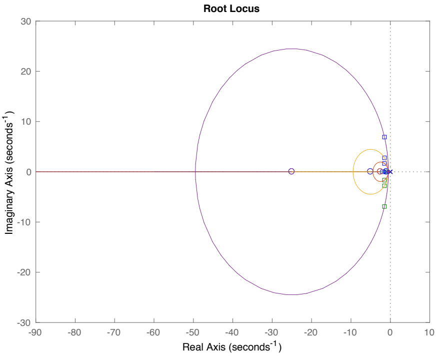
Integral gain selection



As the integral gain increases,

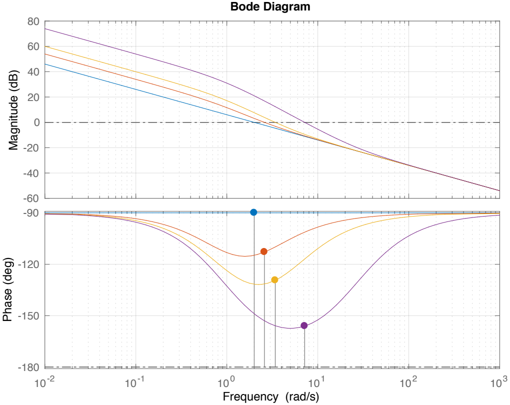
- The steady-state error is zero (as long as k_I is not zero)
- The response becomes more oscillatory (warning!)
- The sensitivity to noise does not change!

Integral gain selection



- Steady-state error to a unit step: $e_{ss} = \lim_{s\to 0} \frac{1}{1+C(s)L(s)} = 0$
- The root locus shows us that as the integral gain increases, the closed-loop poles go from being "slow" and overdamped to being "fast" but with low damping!

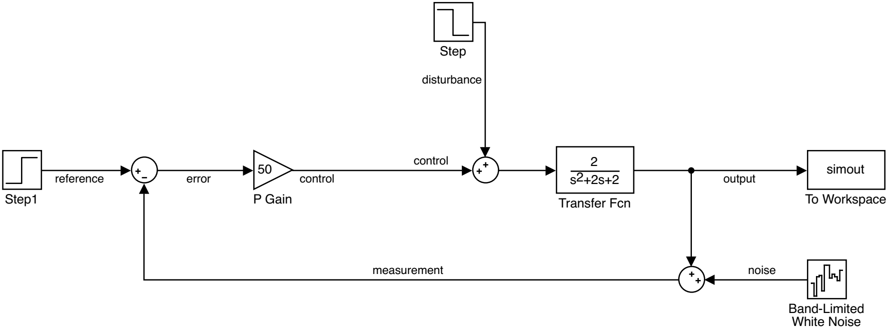
Integral gain selection



As the integral gain increases,

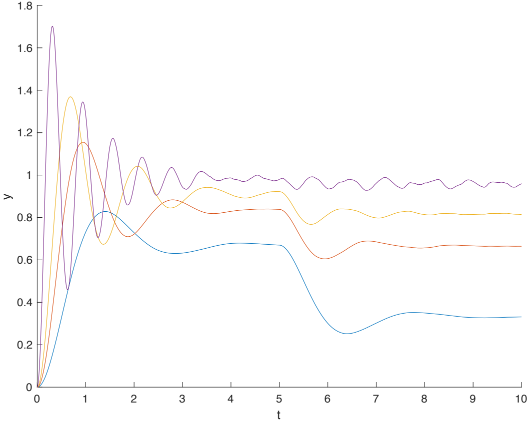
- Phase margin decreases;
- The crossover frequency increases;
- The low-frequency gain increases — but goes to infinity near 0 in all cases ;
- The high-frequency gain does not change.

Proportional Control — Higher order systems



- How do the previous consideration extend to higher-order systems, e.g., 2nd order?

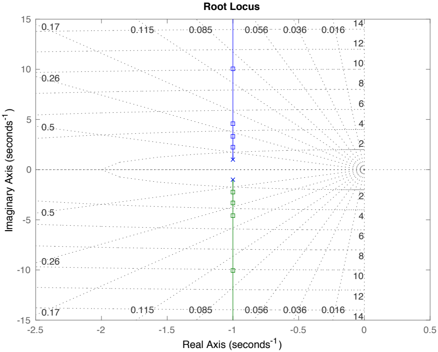
Proportional gain selection



As the proportional gain increases,

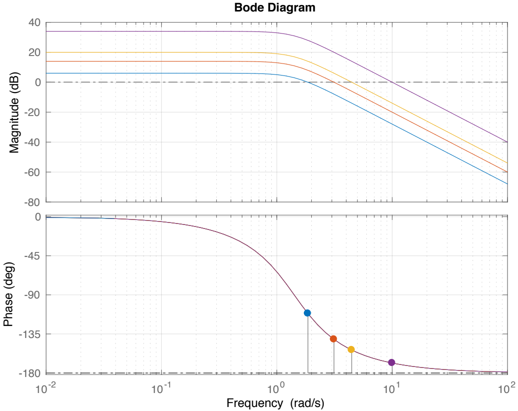
- The closed-loop system become more oscillatory (warning!);
- The steady-state error decreases;
- The response becomes faster;
- The sensitivity to noise increases.

Proportional gain selection



- The root locus shows that as the proportional gain increases, the closed-loop poles have decreasing damping ratio.
- Steady-state error to a unit step: $e_{ss} = \lim_{s \to 0} \frac{1}{1+kL(s)} = \frac{1}{1+k}$

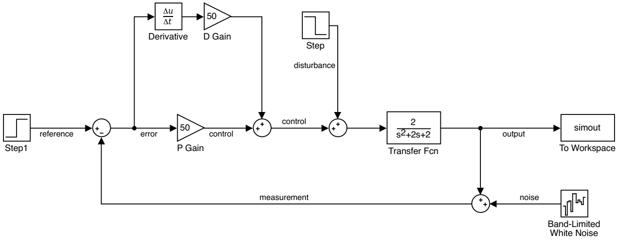
Proportional gain selection



As the proportional gain increases,

- Phase margin gets smaller and smaller!
- The crossover frequency increases;
- The low-frequency gain increases;
- The high-frequency gain increases;

Introducing a differentiator

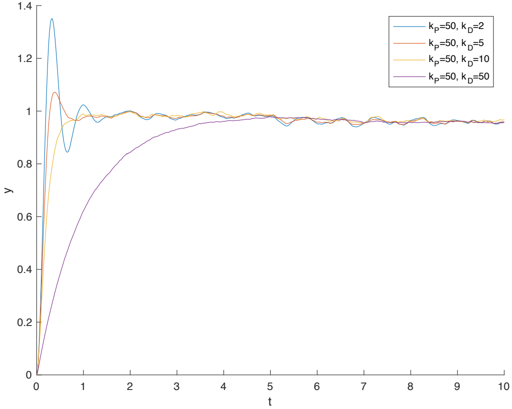


- Differentiating the error allows one to "predict" what the error will do in the near future.
- An derivative control action tries to avoid overshooting, hence damping the system.
- PD control:

$$u(t) = k_P e(t) + k_D \dot{e}(t)$$
$$C(s) = k_P + k_D s.$$

- Note that this is not a causal transfer function (not physically realizable in general). This is typically fixed by approximating the derivative as $s \approx \frac{s}{cs+1}$ for some large c.)

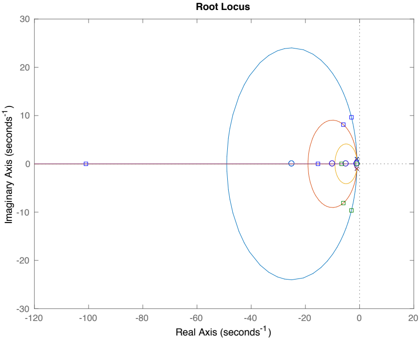
Derivative gain selection



As the derivative gain increases,

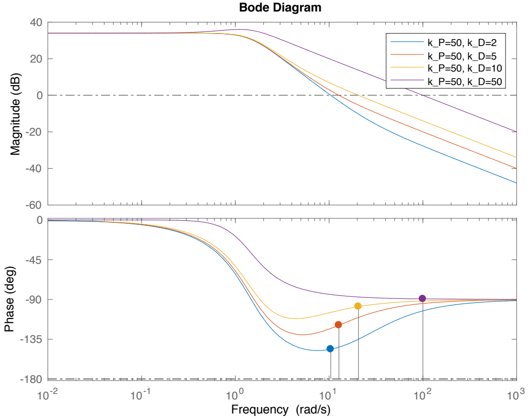
- The steady-state error not affected;
- The response becomes less oscillatory, but potentially slower
- The sensitivity to noise increases!

Derivative gain selection



- Steady-state error to a unit step: $e_{ss} = \lim_{s \to 0} \frac{1}{1+C(s)L(s)} = \frac{1}{1+k_P L(0)}$
- The root locus shows us that as the derivative gain increases, the closed-loop poles are "pulled" into the left half plane!

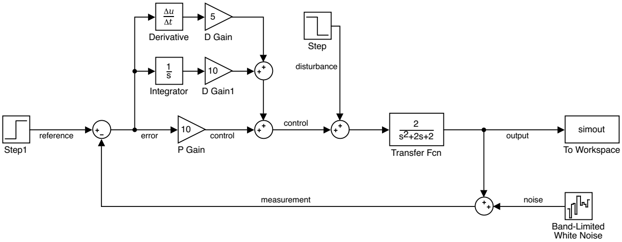
Derivative gain selection



As the derivative gain increases,

- Phase margin increases;
- The crossover frequency increases;
- The low-frequency gain does not change;
- The high-frequency gain increases.

Proportional-Integral-Derivative Control



- One can also combine the effects of an integrator and of a differentiator with the basic proportional controller.
- PID control:

$$u(t) = k_P e(t) + k_I \int_0^t e(\tau) d\tau + k_D \dot{e}(t),$$

$$C(s) = k_P + \frac{k_I}{s} + k_D s = \frac{k_D s^2 + k_P s + k_I}{s}.$$

PID Tuning

- PID tuning corresponds to choosing the parameters k_p, k_i and k_d to reach the feedback control design specifications.
- PID tuning can be done with tuning rules by hand or numerically using MATLAB or other tools (the latter requires a system model).
- There exist heuristic methods to tune a PID controller without a model of the plant $P(s)$, e.g. the tuning rules proposed by Ziegler and Nichols.
- My recommendation: think of a PID as

$$C(s) = k_{RL} \frac{(s - z_1)(s - z_2)}{s}$$

  i.e., as two zeros and one pole at the origin. Decide where you want these zeros (in the complex plane, or in terms of natural frequency and damping ratio on the Bode plot), and what you want the (root-locus) gain to be. Finally, compute the corresponding k_P, k_I, k_D.

Summary

- Proportional control
  - Decrease the steady-state error;
  - Increase the closed-loop bandwidth;
  - Increase sensitivity to noise;
  - Can reduce stability margins for higher-order systems (2nd order or more).
- Integral control
  - Eliminates the steady-state error to a step (if the closed-loop is stable);
  - Reduces stability margins, can make a higher-order system unstable.
- Derivative control
  - Reduce overshooting, increase damping;
  - Improves stability margins;
  - Increase sensitivity to noise.

Today's learning objectives

- Learn what a PID control is and how to design one:
  - Proportional control: what it is, what it does, pro's and con's
  - Derivative control: what it is, what it does, pro's and con's
  - Integral control: what it is, what it does, pro's and con's
- Tuning strategies for PID controllers.

Ziegler Nichols Tuning Rules

- Assumption: Plant can be approximated by the transfer function

$$P(s) = \frac{k}{\tau s + 1} e^{-Ts}$$

with $T/(T+\tau)$ small.

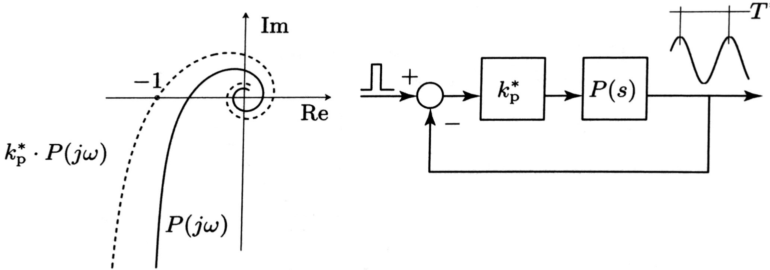
- Apply the controller $C(s) = k_p$ to the system starting at $k_p = 0$ and increase k_p until the system is in a steady-state oscillation, then note the "critical k_p" called k_p^* and the corresponding critical oscillation period T^*.

- Use k_p^* and T^* to calculate the control gains:

type	k_p	T_i	T_d
P	$0.5 \cdot k_p^*$	$\infty \cdot T^*$	$0 \cdot T^*$
PI	$0.45 \cdot k_p^*$	$0.85 \cdot T^*$	$0 \cdot T^*$
PD	$0.55 \cdot k_p^*$	$\infty \cdot T^*$	$0.15 \cdot T^*$
PID	$0.6 \cdot k_p^*$	$0.5 \cdot T^*$	$0.125 \cdot T^*$

Ziegler Nichols Tuning Rules

- Graphically:

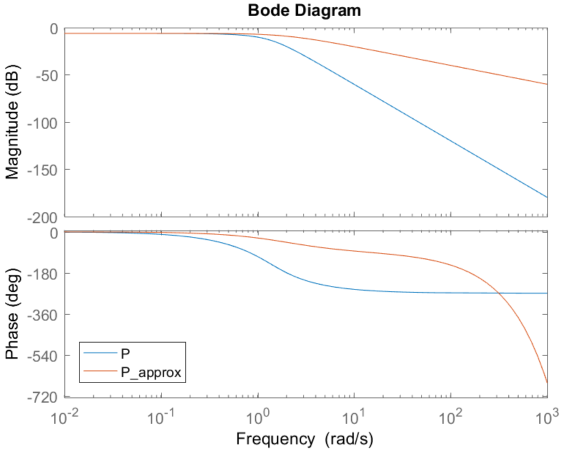


1

[1](Lino Guzzella "Analysis and Synthesis of Single-Input Single-Output Control Systems)

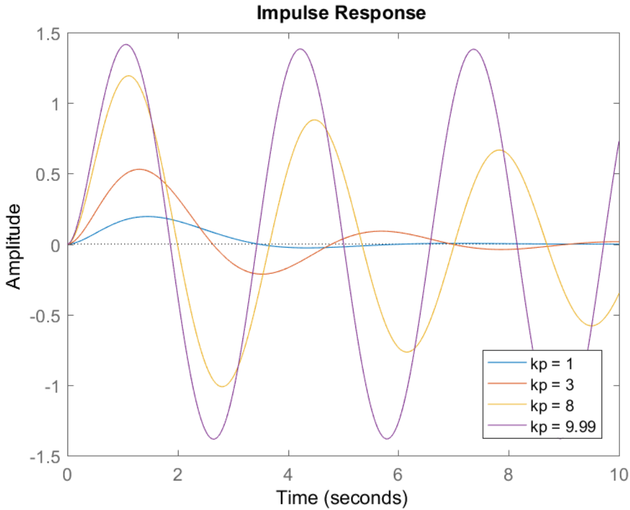
Ziegler Nichols Tuning Example

- Plant: $P(s) = \dfrac{1}{(s+1)\cdot(s^2+2s+2)}$
- Approximation: $P_{approx} = \dfrac{0.5}{0.5\cdot s+1}\cdot e^{-0.01s}$



Ziegler Nichols Tuning Example

- Set $T_i = \infty$, $T_d = 0, \tau = 0$ and increase gain k_p.
- Critical gain $k_p^* = 10$ with critical oscillation period $T^* = \frac{2\pi}{\omega^*} = \frac{2\pi}{2} = \pi$



Ziegler Nichols Tuning Example

- P, PI, PD, PID controller according to Ziegler and Nichols tuning rules.
- PID controller derived with MATLAB sisotool.
- Ziegler Nichols tuning rules can be useful when no model of the plant is available but generally other tuning rules provide better results.