Continuous-Time PID Control

A really quick and dirty overview for <u>model-based</u> PID design

Plant analysis

And the MATLAB Control Toolbox

Plant analysis

$$H(s) = \frac{A}{s^i} \frac{\prod_1^c \left(1 + s\tau_j\right) \prod_1^c \left(1 + 2\zeta_j \tau_{oj} s + s^2 \tau_{oj}^2\right)}{\prod_1^e \left(1 + s\tau_j\right) \prod_1^f \left(1 + 2\xi_j T_{oj} s + s^2 T_{oj}^2\right)}$$

- Can be represented using Bode elements.
- Type of system: Number of integrators i (type 0, type 1, …)
- Representations of block: Zero-Pole-Gain (zpk), Time constants...
    - E.g. (s-a) or (1+sa), leads to different system insights (time vs. frequency)

- First thing to do? Bode plot + Stability analysis!

MATLAB Control Toolbox

```
s = tf('s');
G = 1/(1+10*s)/(1+s)/(1+0.5*s)
>>
G =

                1
  ----------------------------
  5 s^3 + 15.5 s^2 + 11.5 s + 1

Continuous-time transfer function.
```

https://nl.mathworks.com/help/control/ref/tf.html

MATLAB Control Toolbox – `zpk`

```
>> G = zpk(G)

G =

            0.2
  ------------------
  (s+2) (s+1) (s+0.1)

Continuous-time zero/pole/gain model.
```

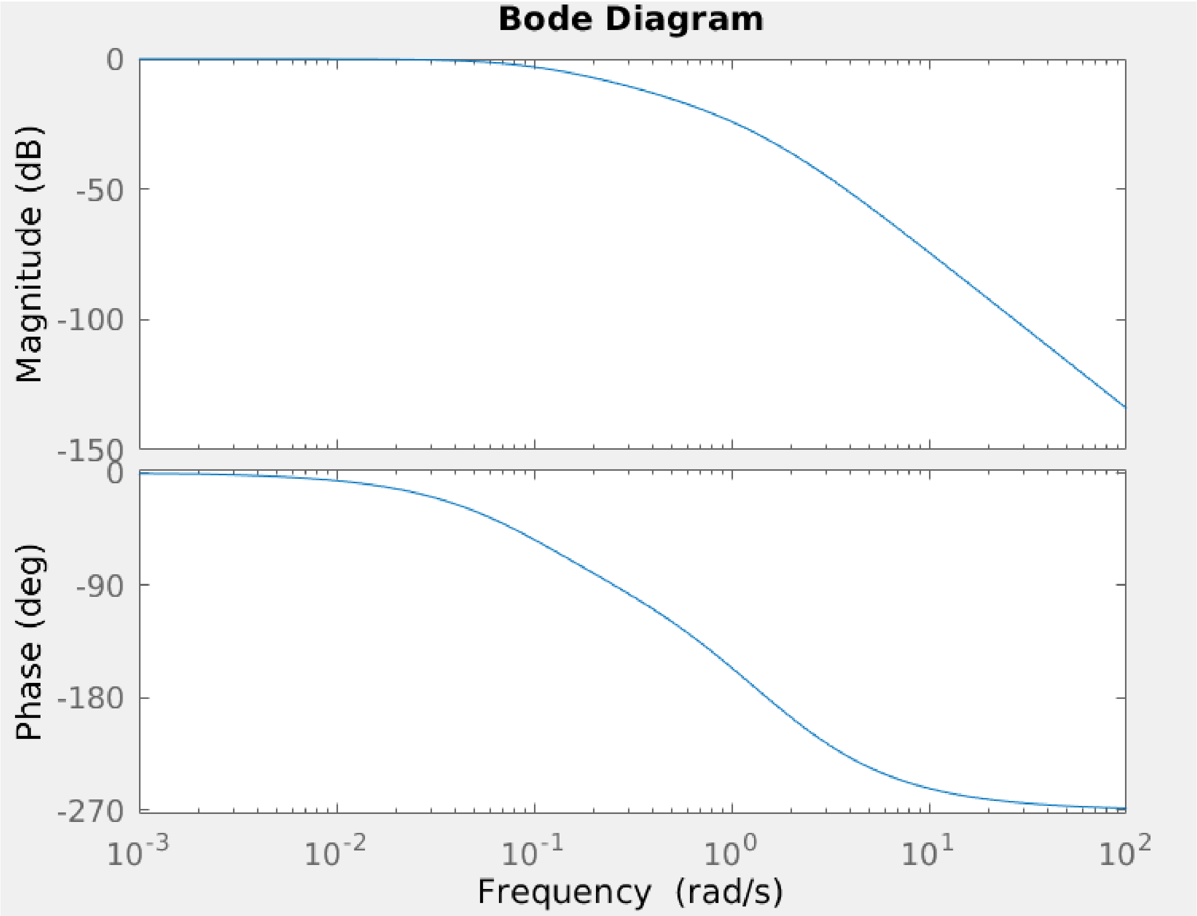
https://nl.mathworks.com/help/control/ref/zpk.html

MATLAB Control Toolbox – `zpk`

Options: `roots`, `frequency`, `time constant`

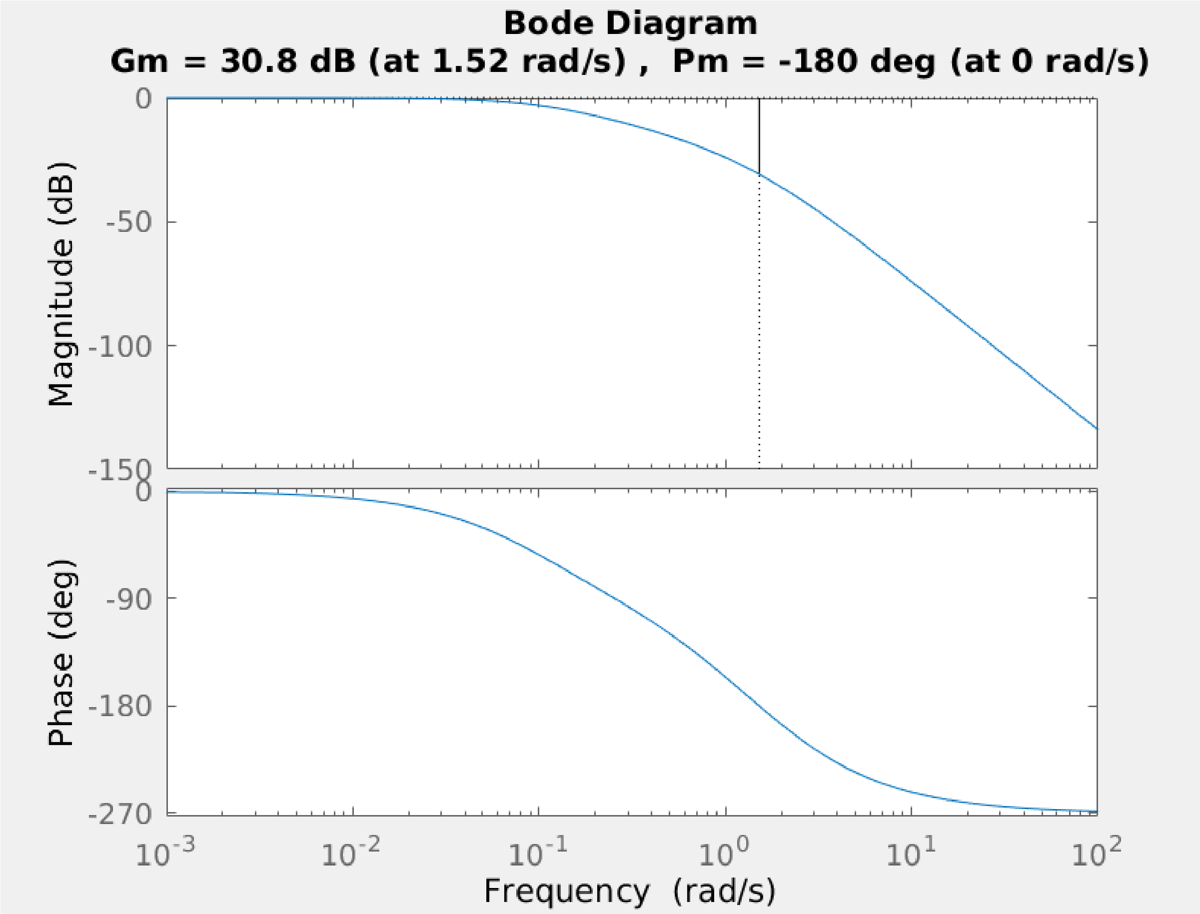
```
>> G.DisplayFormat = 'time constant'

G =

             1
  ---------------------
  (1+0.5s) (1+s) (1+10s)

Continuous-time zero/pole/gain model.
```

MATLAB Control Toolbox – `bode(G)`



https://nl.mathworks.com/help/control/ref/bode.html

MATLAB Control Toolbox – `margin(G)`



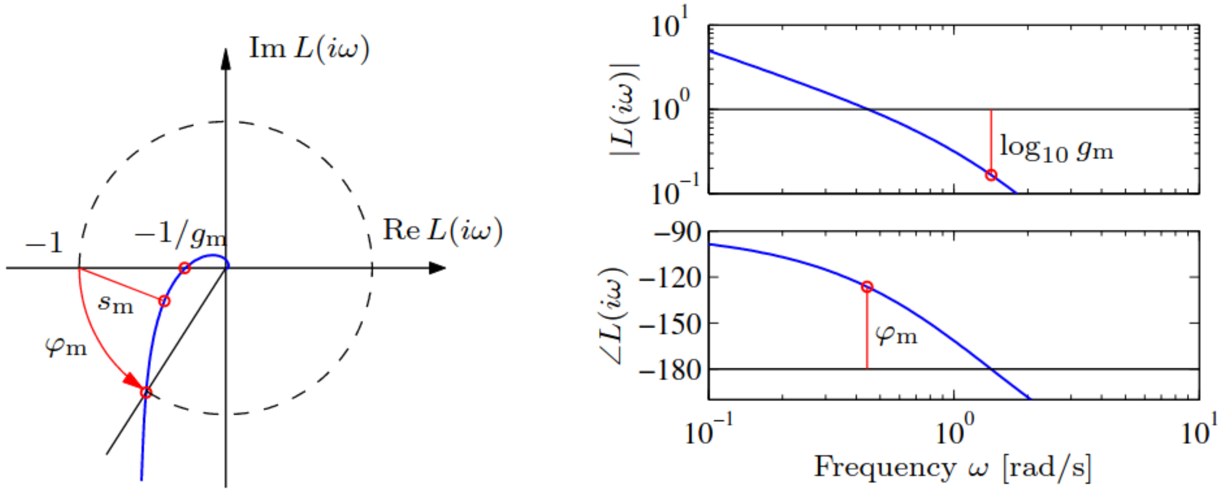
https://nl.mathworks.com/help/control/ref/margin.html

Intermezzo stability margins

- Phase margin: How much phase (delay) can I add to the system before it becomes unstable. For a cut-off frequency wc:
  - PM = arg L(j*wc) + 180deg
  - PM > 0, stable, PM = 0, marginally stable, PM < 0, unstable.

- Gain margin: How much gain can I add to the system before it becomes unstable. For a frequency wg, where L(j*wg) = -180 deg
  - GM = 1/|L(j*wg)|

Intermezzo stability margins



(a) Nyquist plot (b) Bode plot

Figure 10.11: Stability margins for a third-order loop transfer function $L(s)$. The Nyquist plot (a) shows the stability margin, s_m, the gain margin g_m, and the phase margin φ_m. The stability margin s_m is the shortest distance to the critical point -1. The gain margin corresponds to the smallest increase in gain that creates an encirclement, and the phase margin is the smallest change in phase that creates an encirclement. The Bode plot (b) shows the gain and phase margins.

http://www.cds.caltech.edu/~murray/books/AM08/pdf/fbs-loopanal_24Jul2020.pdf

MATLAB Control Toolbox – other commands

`help, step, dcgain, impulse, nyquist, pole, zero, pzmap, feedback, stepinfo, ...`

https://nl.mathworks.com/help/control/response-plots-and-data.html

Literature

Feedback Systems, *Astrom & Murray*

Chapter 8, 9

Control Engineering, *Keviczky, Bars, Hetthessy & Banyasz*

Chapter 2.3-5, 5.1-3, 5.6

Modern Control Engineering, *Ogata*

Chapter 5

Controller Design

Using resources from Matlab Exercises for Introductory Control Theory, Jeno Hetthessy, Ruth Bars, Andras Barta

Control Objectives

- Typical control objectives:
  1. Stability
     - Unstable systems are generally useless
  2. Steady-state accuracy
     - Also referred to as reference-tracking
  3. Transient response
     - Properties such as overshoot, settle time.
     - Overshoot depends on phase margin
     - Settling time depends on cut-off frequency

Closed-Loop Feedback with PID

PID Forms:

- P = K
- PI = K*(1+Ti*s)/(Ti*s)
- PD = K*(1+Td*s)/(1+Tf*s) *(Non-ideal implementation)*
- PID ≈ K*(1+Ti*s)/(Ti*s)*(1+Td*s)/(1+Tf*s) *(If Ti > Td-Tf > Tf)*

Design parameters: K, Ti, Td, Tf.

- Ti = Largest time constant of plant
- Td = Second largest time constant of plant
- Tf = Generally Td/5 - Td/10
- K = Used to specify Phase Margin

Intermezzo PID Forms

Ideal PID:

Cpid = (K/Ti)*(1+Ti*s+Ti*Td*s^2)/s

Realizable PID:

Cpid = (K/Ti)*(1+(Ti+Tf)*s+Ti*(Td+Tf)*s^2)/s/(1+Tf*s)

In some practical cases, a useful approximation:

Cpid ≈ (K/Ti)*((1+Ti*s)(1+Td*s))/(s*(1+Tf*s))

Intermezzo PID Forms

Why all these forms?

Pole placement!

For suitable Ti, Td, Tf:

$p_1 = 0$, $p_2 = -1/Tf$

$z_1 \approx -1/Ti$, $z_2 \approx -1/Td$

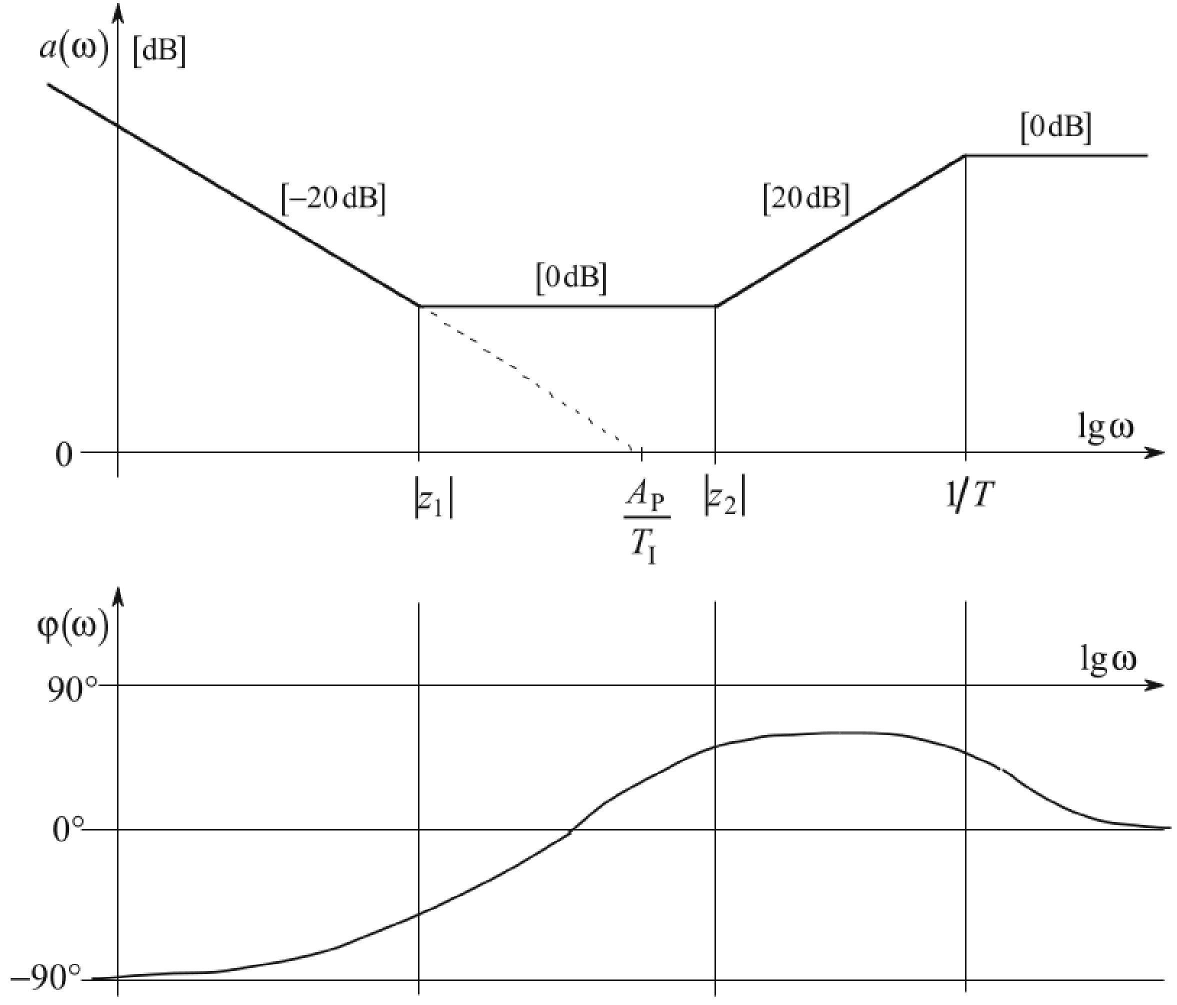


Fig. 8.7 Asymptotic Bode diagram of the approximate *PID* regulator

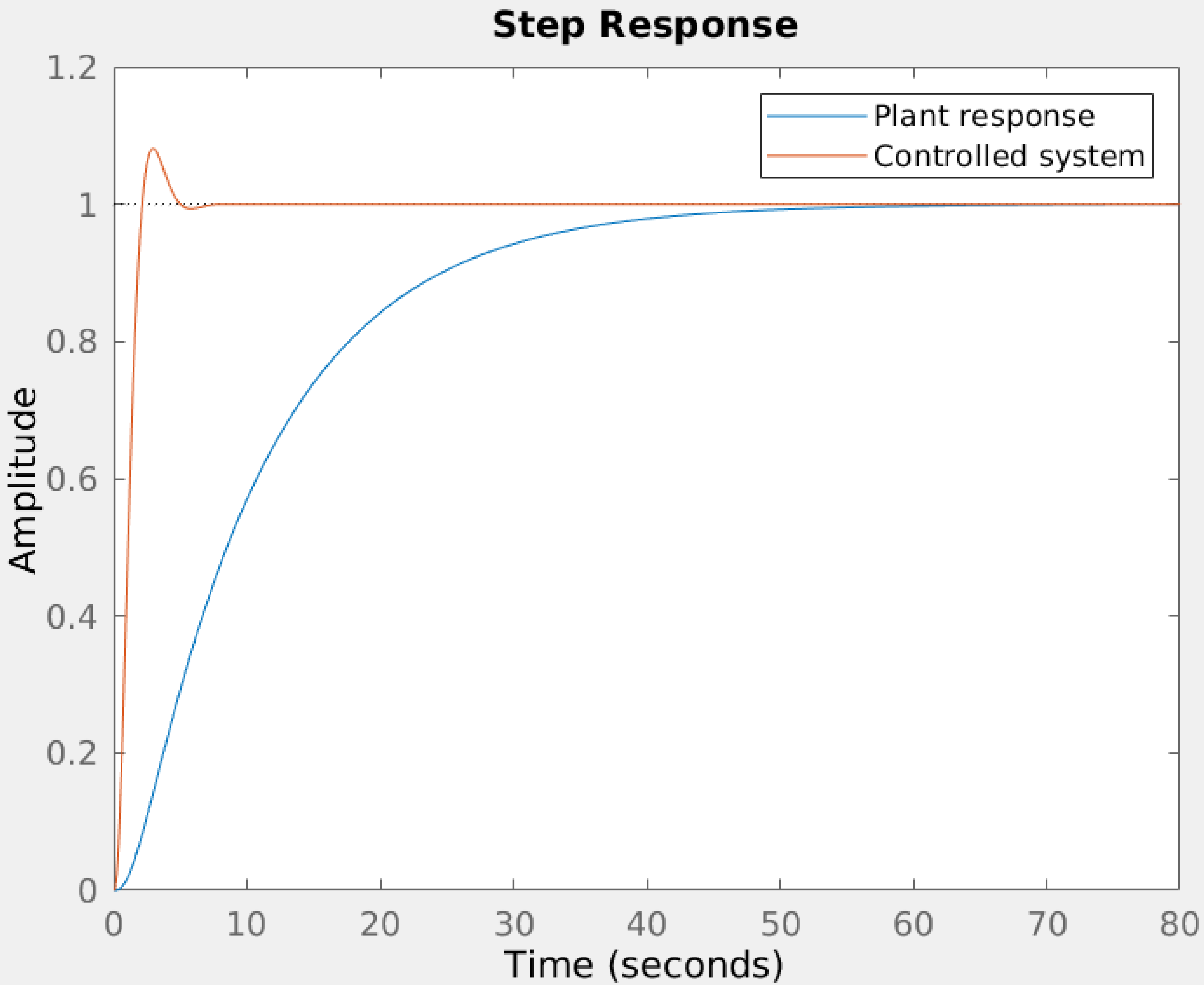
Example – PID design for 60 degree PM

```
% Plant G as defined earlier
C = (1+10*s)/(10*s)*(1+1*s)/(1+1/10*s);
[mag,phase,w] = bode(G*C);
K = margin(mag,phase-60,w); % Phase cross-over
C = K*C; L = G*C;
T = feedback(L,1); S = 1-T;
U = feedback(C,G);
step(T)
```

https://nl.mathworks.com/help/control/ref/feedback.html

Results

Can you do better..?



Response evaluation

- Steady-state value of transfer function using final value theorem
  - Steady-state → s = 0 → lim(s→0^+) T(s), where T(s) is your desired TF

    Note: For step input only, otherwise little bit more complex
  - `dcgain` function in MATLAB does this
- For step response, the `stepinfo` command is excellent!

- You'll need to figure out the correct transfer functions for question 1&2

https://nl.mathworks.com/help/control/ref/stepinfo.html

What else?

- When do you need the full PID controller?
- What if you have complex conjugate poles?
- What if your system is unstable?
- What if I can make my system infinitely fast?

Literature

Feedback Systems, *Astrom & Murray*

Chapter 10, 11

Control System Design, *Friedland*

Chapter 4.1-4, 4.7-9

Control Engineering, *Keviczky, Bars, Hetthessy & Banyasz*

Chapter 4.1-6, 6, 8.1, 8.5

Modern Control Engineering, *Ogata*

Chapter 7, 8.1-3, 8.7