01211433 VISION AND CONTROL OF INDUSTRIAL ROBOTS

ดร.วโรดม ตู้จินดา

ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์

ม.เกษตรศาสตร์

Lecture 4 : Classical Control Design

การออกแบบระบบควบคุมโดยวิธี **loopshaping**

Classical Control Design

- วิธีการจัดสัณฐานวงรอบ (loopshaping)
- พิจารณาเงื่อนไขด้านเสถียรภาพและสมรรถนะทั้งหมด เพื่อกำหนดเงื่อนไขให้กับผลตอบสนองความถี่ของ L
- แก้ปัญหาข้อขัดแย้งระหว่างเงื่อนไขด้านสมรรถนะ โดยแบ่งย่านความถี่
- จัดรูปผลตอบสนองความถี่ของ L ให้สอดคล้องกับข้อกำหนด

ย่านความถี่ของสัญญาณอินพุต

- คำสั่ง **r:** คำสั่งโดยทั่วไปจะไม่เปลี่ยนแปลงตามเวลาอย่างรวดเร็ว ดังนั้นจะอยู่ในย่านความถี่ต่ำ
- การรบกวน **d:** สัญญาณการรบกวนที่เข้ามา ณ ตำแหน่งอินพุตและเอาต์พุตของพลานต์มักจะอยู่ในย่านความถี่ต่ำ เช่น การสั่นสะเทือนทางกล เรโซแนนส์ หรือแรงที่ส่งจากโครงสร้างข้างเคียง
- สัญญาณรบกวนจากการวัด **n:** ตัวรับรู้ส่วนใหญ่จะมีความผิดพลาดมากขึ้นเมื่อความถี่สูงขึ้น ดังนั้นสัญญาณรบกวนจากการวัดจะอยู่ในย่านความถี่สูง

เงื่อนไขบังคับบนฟังก์ชันถ่ายโอนวงปิด

- **LOW : (สมรรถนะ)** ต้องการให้ $|S(j\omega)| << 1$ เพื่อสมรรถนะการตามรอยและการขจัดการรบกวนที่ดี
- **MID : (เสถียรภาพ)** เนื่องจาก $|S(j\omega)| >> 1$ แสดงถึงเสถียรภาพที่ไม่ดี จึงต้องการขอบเขตจำกัดด้านบนสำหรับ $|S(j\omega)|$ สังเกตว่าจากเงื่อนไขเชิงพีชคณิต $|S(j\omega)| >> 1$ จะทำให้

  $|T(j\omega)| >> 1$ ด้วย
- **HIGH : (สมรรถนะ)** ต้องการ $|T(j\omega)| << 1$ เพื่อสมรรถนะการขจัดสัญญาณรบกวนจากการวัดที่ดี

การแปลงเป็นเงื่อนไขบังคับบน L(jω)

- LOW: สำหรับ $S(j\omega)$ ค่าน้อย ๆ $$S(j\omega)=\frac{1}{1+L(j\omega)}\approx\frac{1}{L(j\omega)}$$
  - $|S(j\omega)| << 1$ แสดงนัยว่า $|L(j\omega)| >> 1$ กล่าวคือ ขอบเขตบน L สร้างได้โดยผกผันขอบเขตบน S
- MID: $|S(j\omega)| << M$ แปลงเป็น L(s) มีค่าเผื่อของเฟสเพียงพอ
- HIGH: สำหรับ $T(j\omega)$ ค่าน้อย ๆ
  - ดังนั้น $|T(j\omega)| << 1$ $$T(j\omega)=\frac{L(j\omega)}{1+L(j\omega)}\approx L(j\omega)$$
  - แสดงนัยว่า $|L(j\omega)| << 1$ กล่าวคือ ขอบเขตบน L สร้างได้โดยผกผันขอบเขตบน S

ความสัมพันธ์ระหว่างอัตราขยายและเฟสของโบเด (Bode gain-phase relationship)

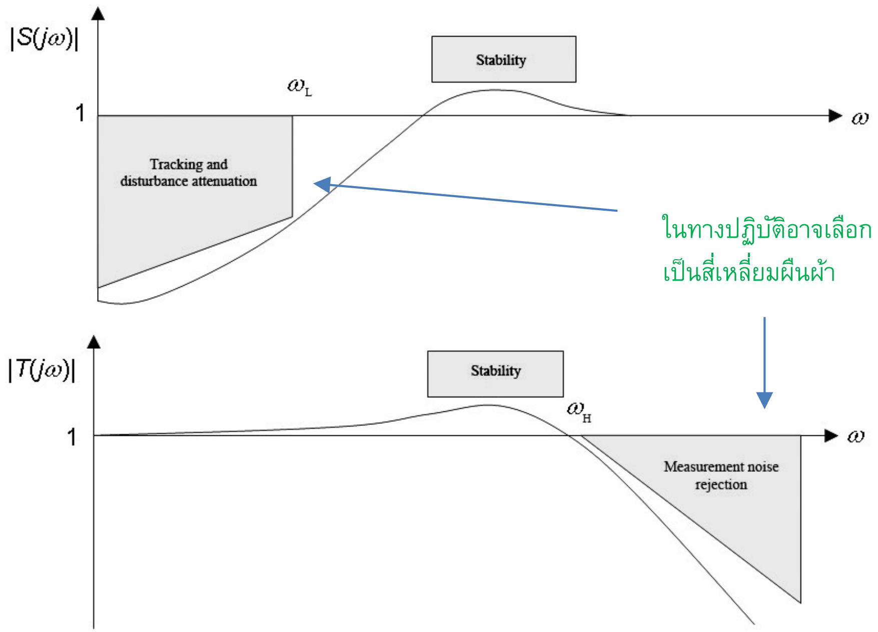
สำหรับระบบเฟสต่ำสุดที่เสถียร เฟสของฟังก์ชันถ่ายโอน $L(j\omega)$ ใด ๆ จะมีสัมพันธ์ได้อย่างเดียวกับอัตราขยายหรือขนาดของ $L(j\omega)$ โดยถ้าหากความชันของ $|L(j\omega)|$ เทียบกับ ω บนสเกลแบบ log-log มีค่าคงที่ประมาณช่วงความถี่เพิ่มขึ้นสิบเท่าหรือ 1 decade ความสัมพันธ์จะเขียนได้เป็น

$$\angle L(j\omega) \cong n \times 90^o$$

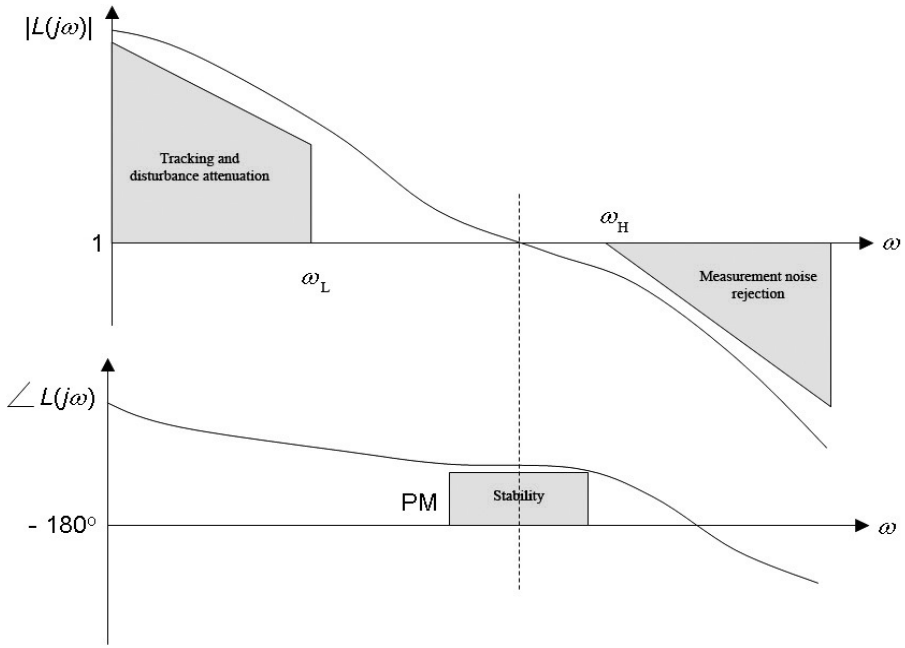
โดย n คือ ค่าความชันของ $|L(j\omega)|$ มีหน่วยเป็น decade ของขนาดต่อ decade ของความถี่

เมื่อพิจารณาแผนภาพโบเดที่พล็อตขนาดเป็น dB จะสรุปได้ว่าความชันของกราฟที่บริเวณ crossover frequency จะต้องมีค่าประมาณ -20 dB/decade และคงที่ในช่วงความถี่บริเวณนั้น

ขอบเขตบน S และ T
จากเงื่อนไขบังคับ



ขอบเขตบน L
จากเงื่อนไขบังคับ

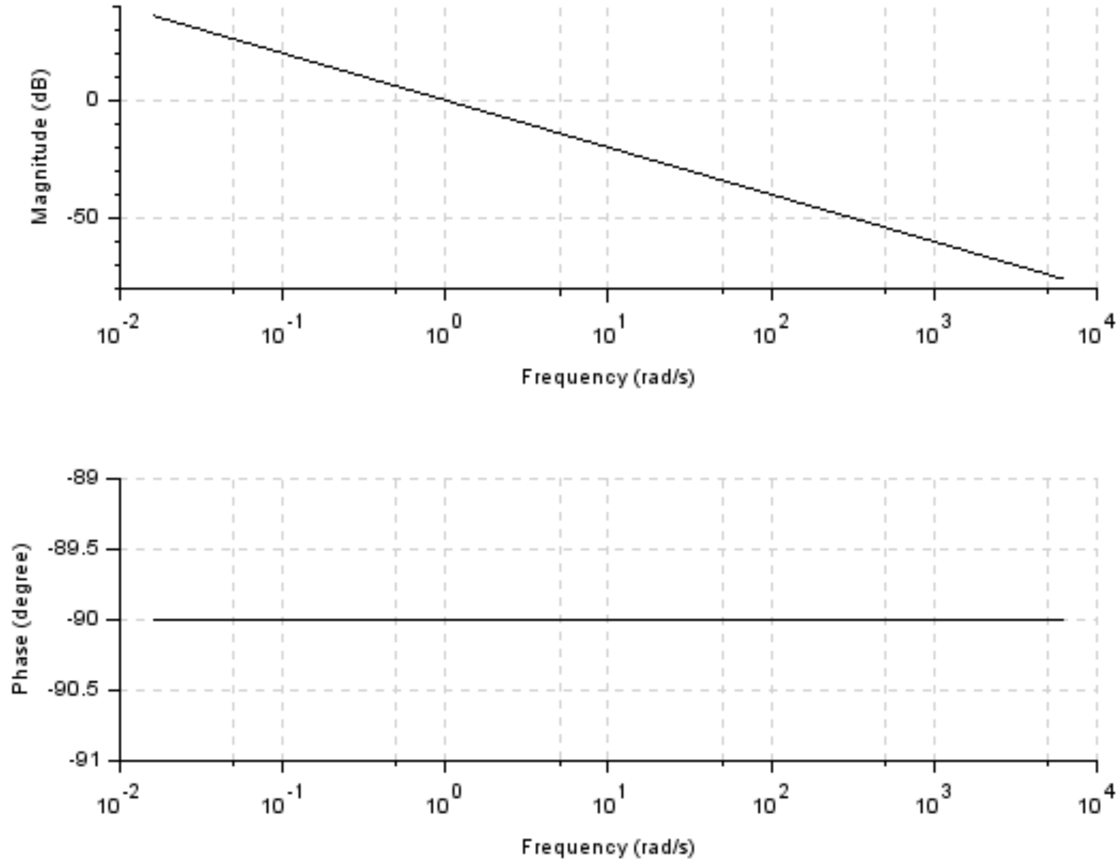


ผลของโพลและซีโรต่อผลตอบสนองความถี่

- Gain: K
- Integrator: $1/s$
- Derivative: s
- First order pole: $a/(s+a)$
- First order zero: $(s+a)/a$

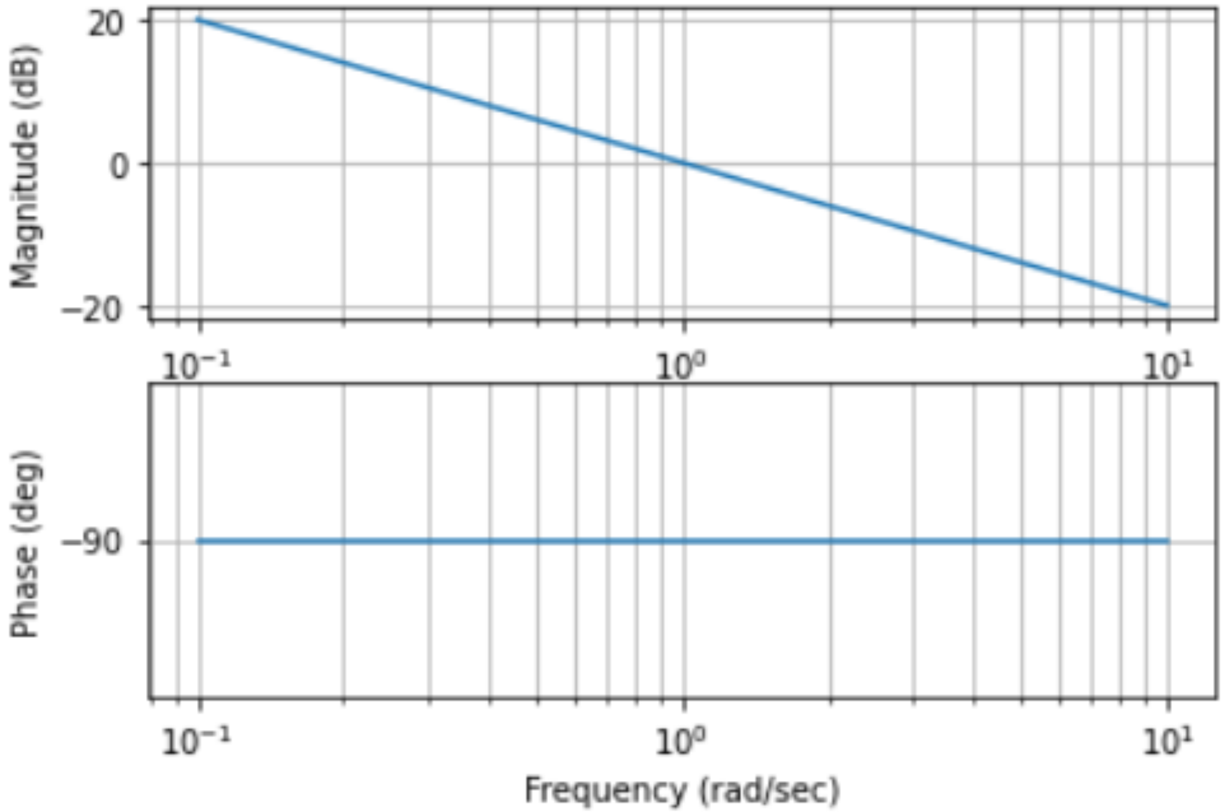
Integrator

```
-->Pint=syslin('c',1/s)
 Pint  =

   1
   -
   s

-->bode(Pint,'rad')
```

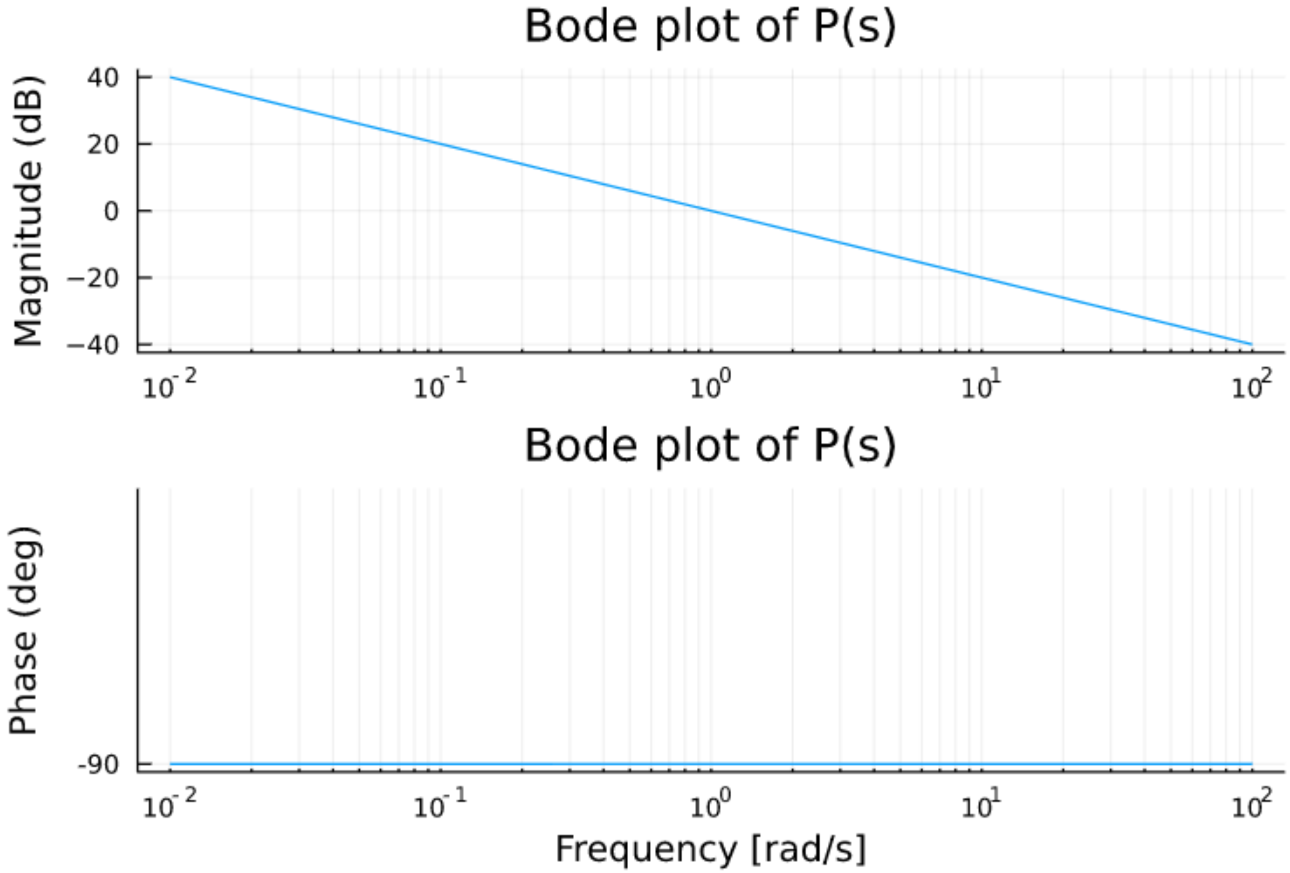


Integrator (Python)

In [2]:
```
s = ctl.tf('s')
H = 1/s
_,_,_ = ctl.bode_plot(H,dB=True)
```



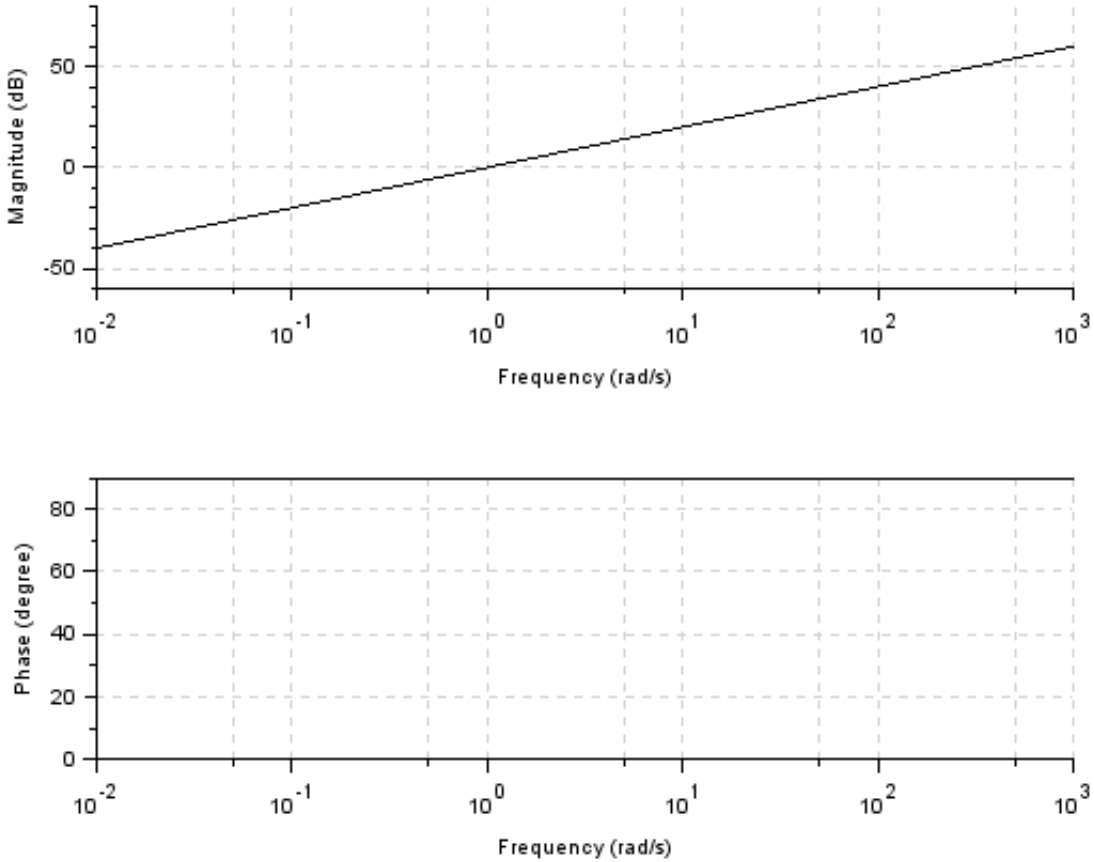
Integrator (Julia)




```julia
begin
    P = 1/s
    setPlotScale("dB")
    w = exp10.(LinRange(-2,2,2000))
    bodeplot(P,w; title="Bode plot of P(s)", legend=false)

end
```

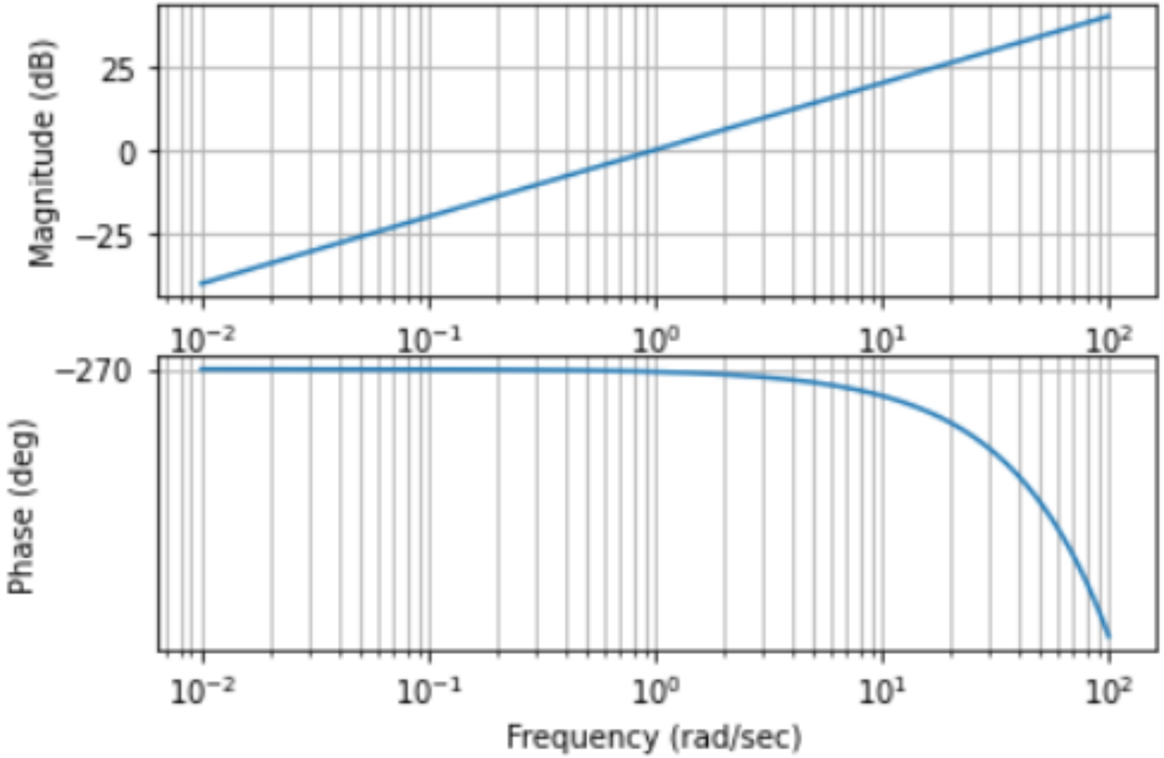
Derivative (approx.)

```
-->Pd = syslin('c',s/(1e-6*s+1));

-->bode(Pd,'rad')
```

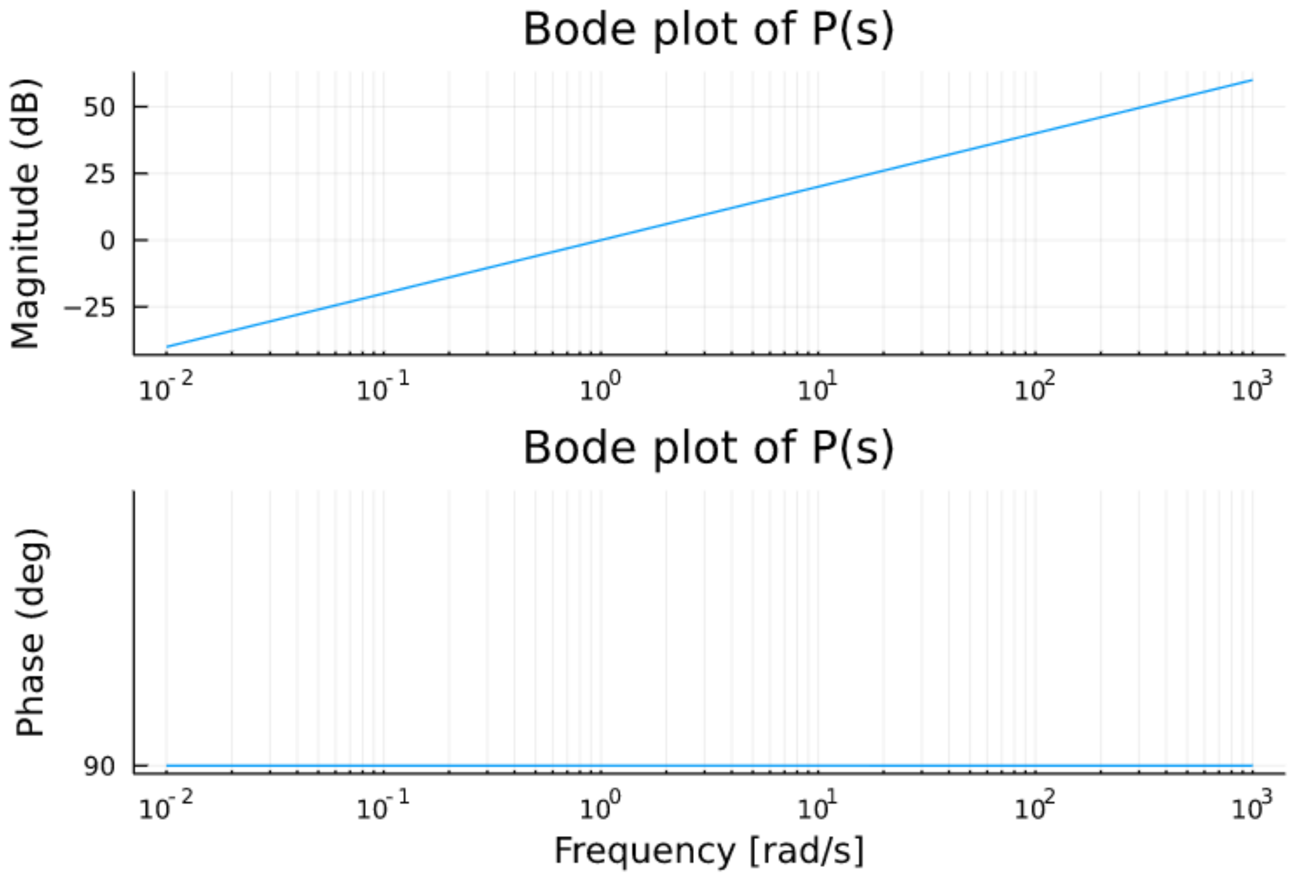


Derivative (approx.) Python

```
In [4]: H = s/(1e-6*s+1)
        _,_,_ = ctl.bode_plot(H,dB=True,omega_limits=(0.01,100))
```



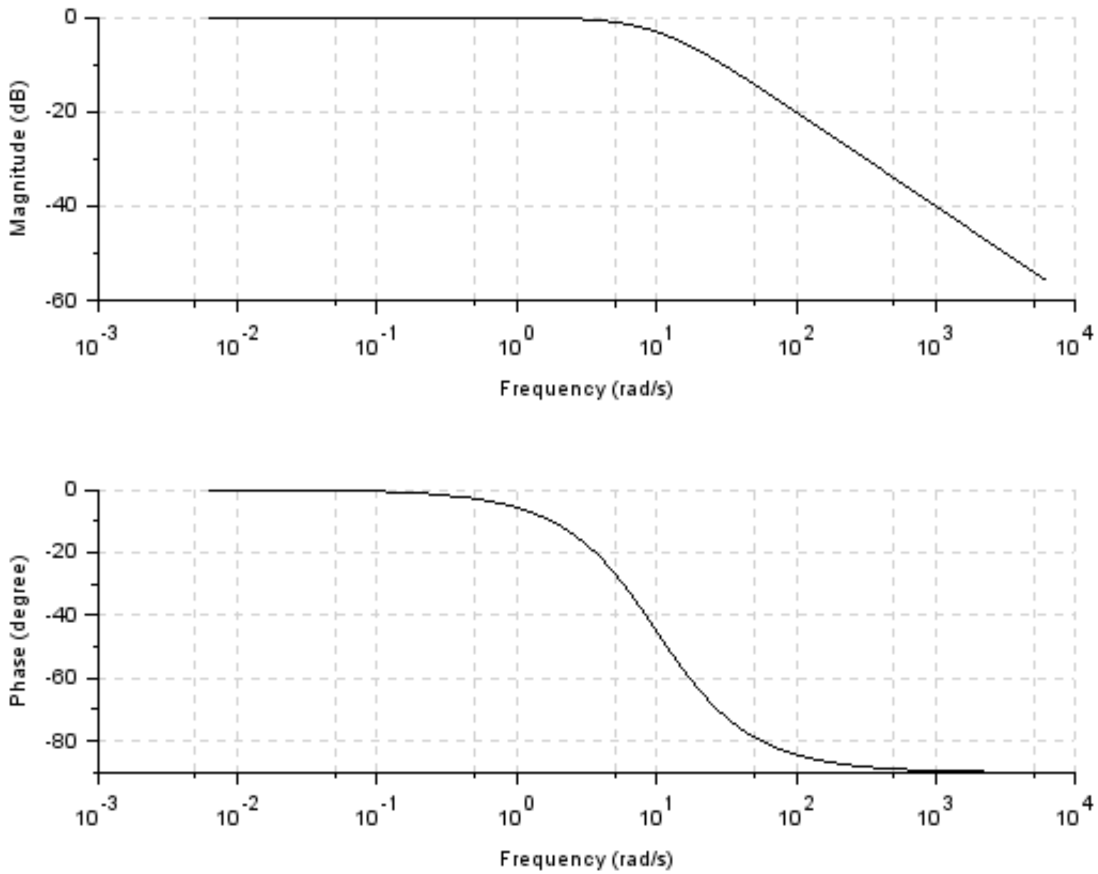
Derivative Julia




```
begin
    P = s
    setPlotScale("dB")
    w = exp10.(LinRange(-2,3,2000))
    bodeplot(P,w; title="Bode plot of P(s)",legend=false)

end
```

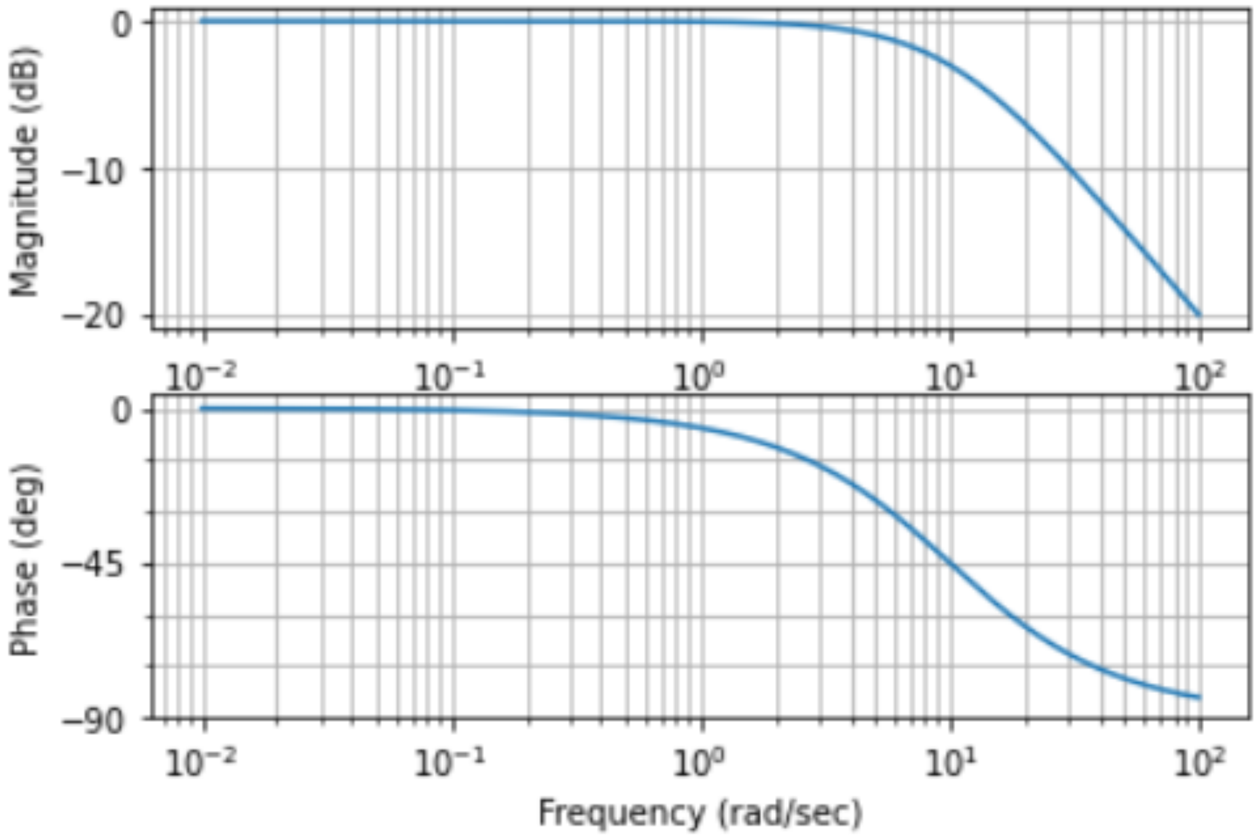
First order pole

```
-->Pp = syslin('c',10/(s+10));

-->bode(Pp,'rad')
```

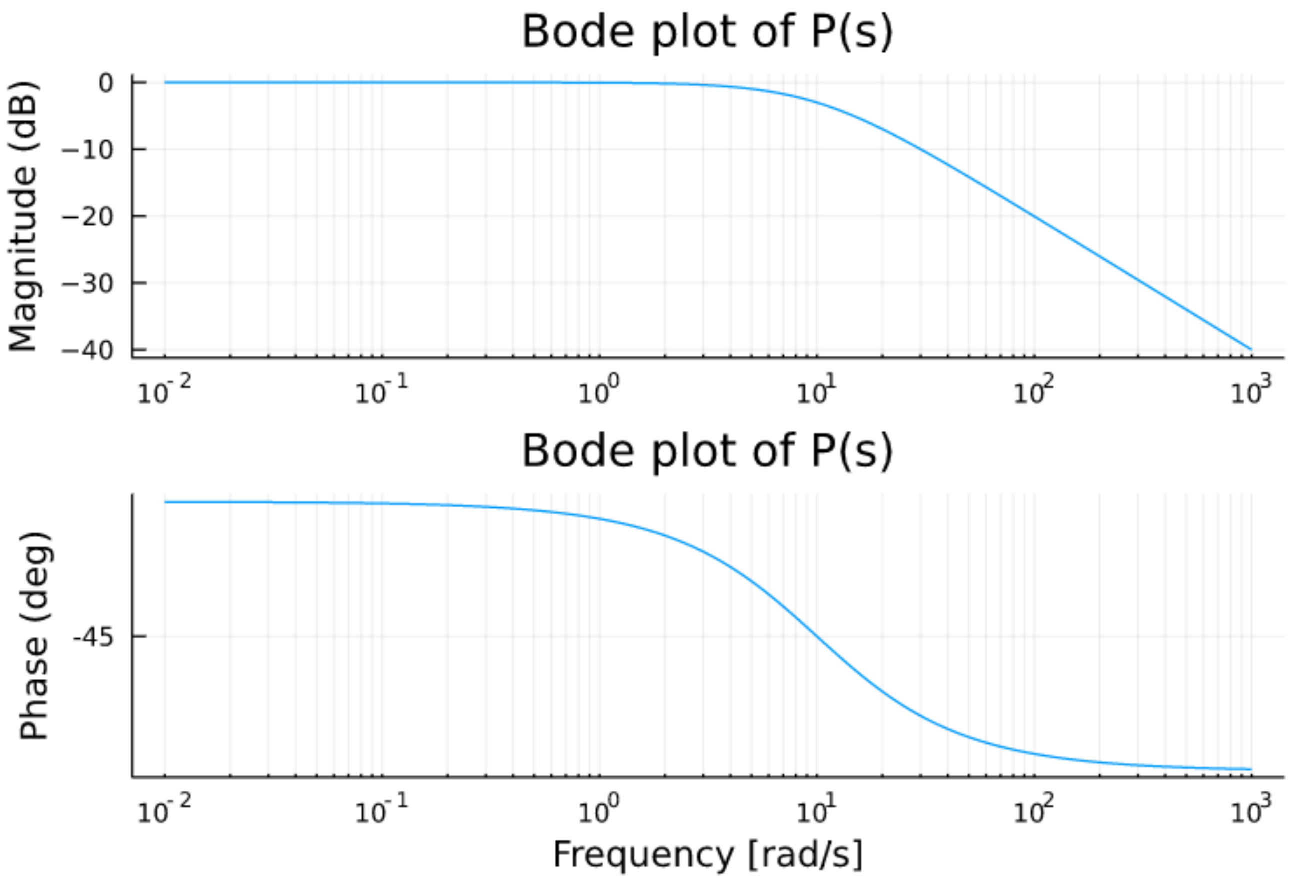


First order pole (python)

```
In [5]: H = 10/(s+10)
        _,_,_ = ctl.bode_plot(H,dB=True,omega_limits=(0.01,100))
```



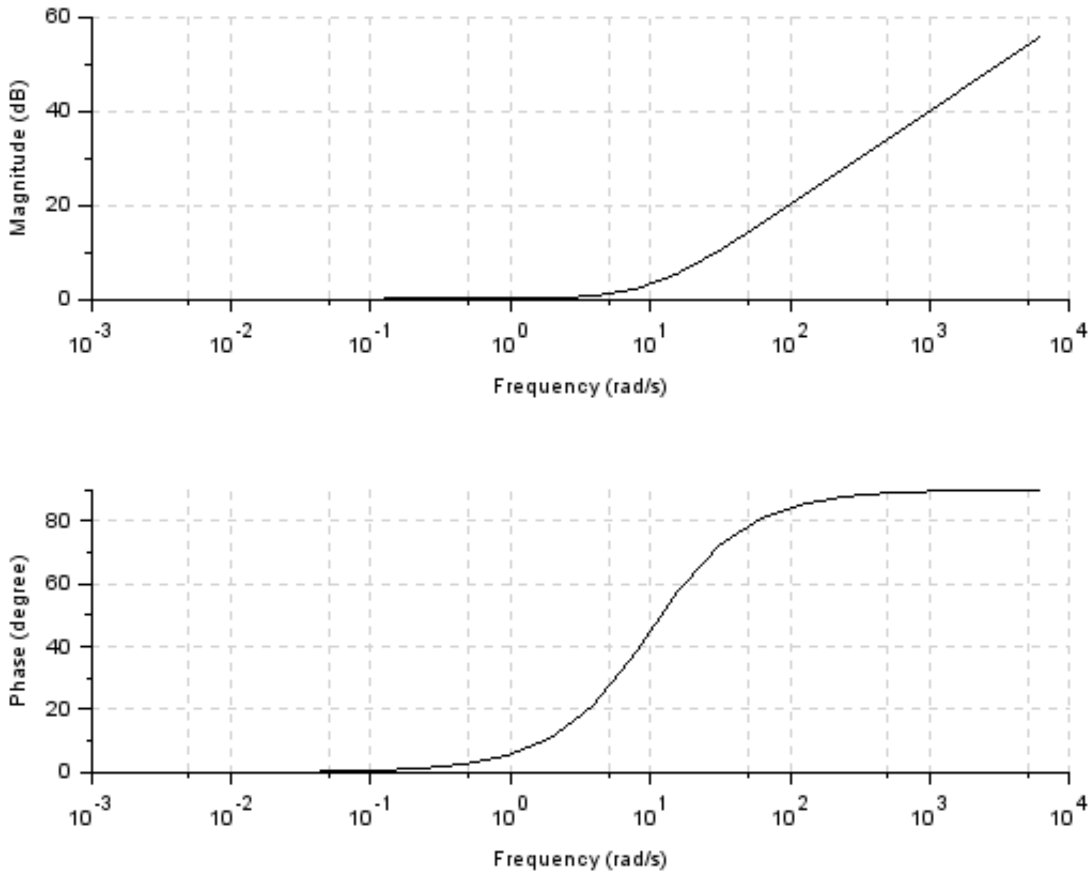
First order pole (Julia)




```
begin
    P = 10/(s+10)
    setPlotScale("dB")
    w = exp10.(LinRange(-2,3,2000))
    bodeplot(P,w; title="Bode plot of P(s)",legend=false)

end
```

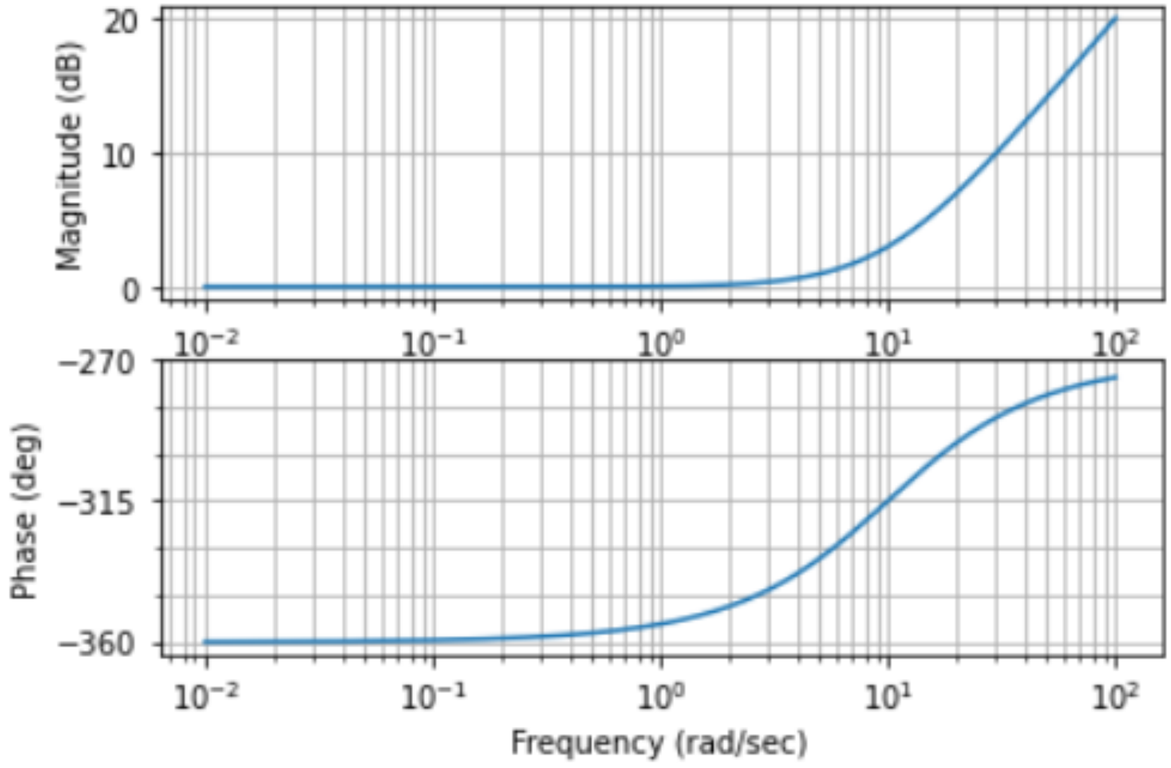
First Order Zero (approx.)

```
-->Pz = syslin('c',(s+10)/(10*(1e-6*s+1)));

-->bode(Pz,'rad')
```

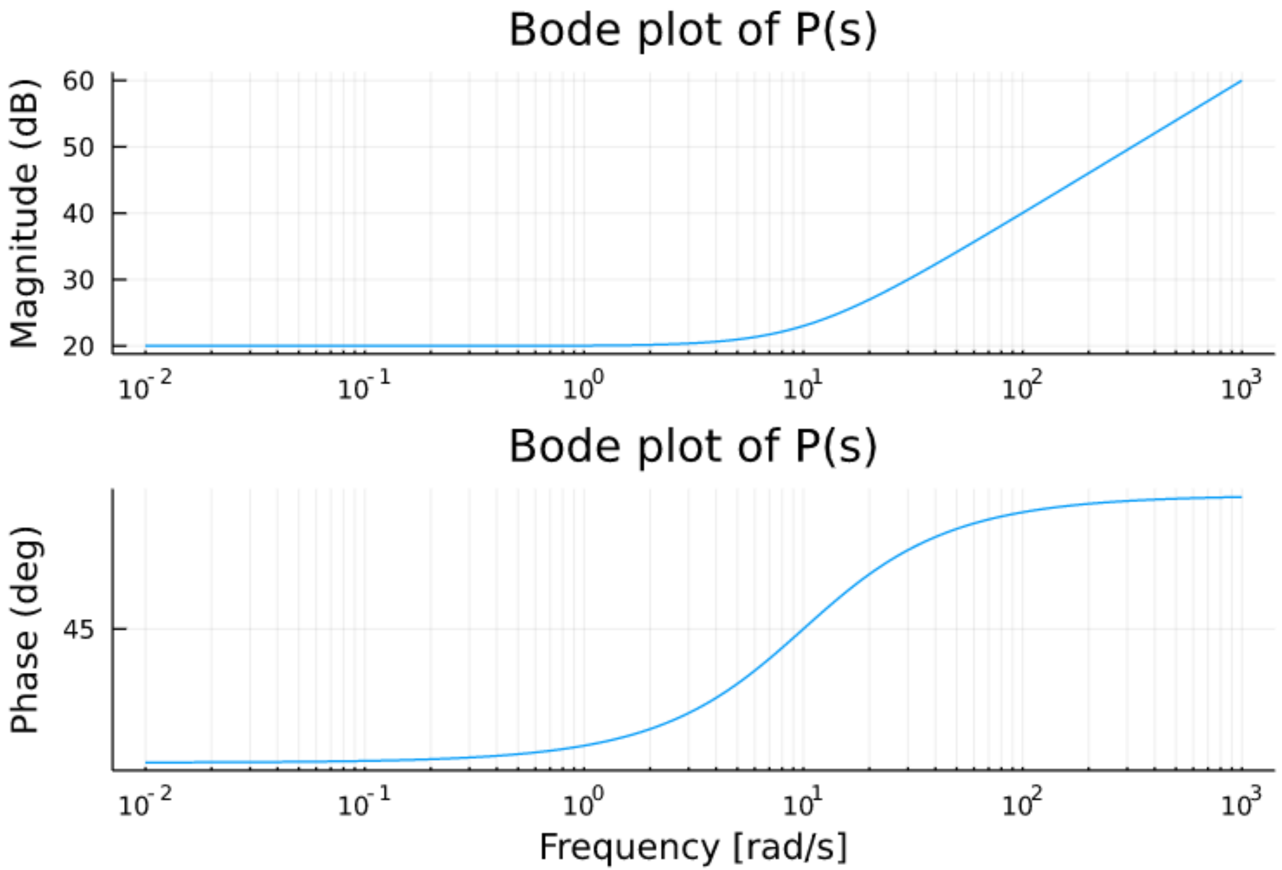


First Order Zero (approx.) python

```
In [6]: H = (s+10)/(10*(1e-6*s+1)) # approx. H = (s+10)
        _,_,_ = ctl.bode_plot(H,dB=True,omega_limits=(0.01,100))
```



First Order Zero Julia

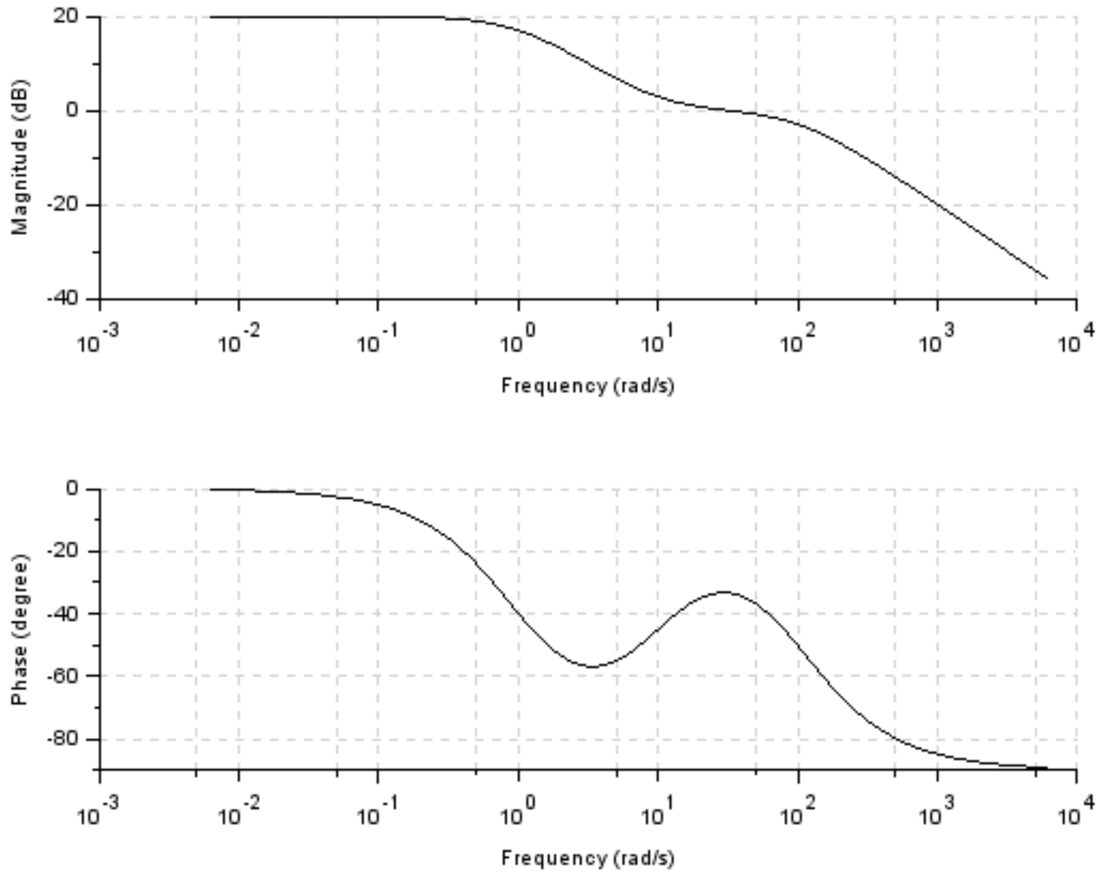



```
begin
    P = (s+10)
    setPlotScale("dB")
    w = exp10.(LinRange(-2,3,2000))
    bodeplot(P,w; title="Bode plot of P(s)",legend=false)

end
```

Combination

Ex. $$\frac{100(s+10)}{(s+1)(s+100)}$$

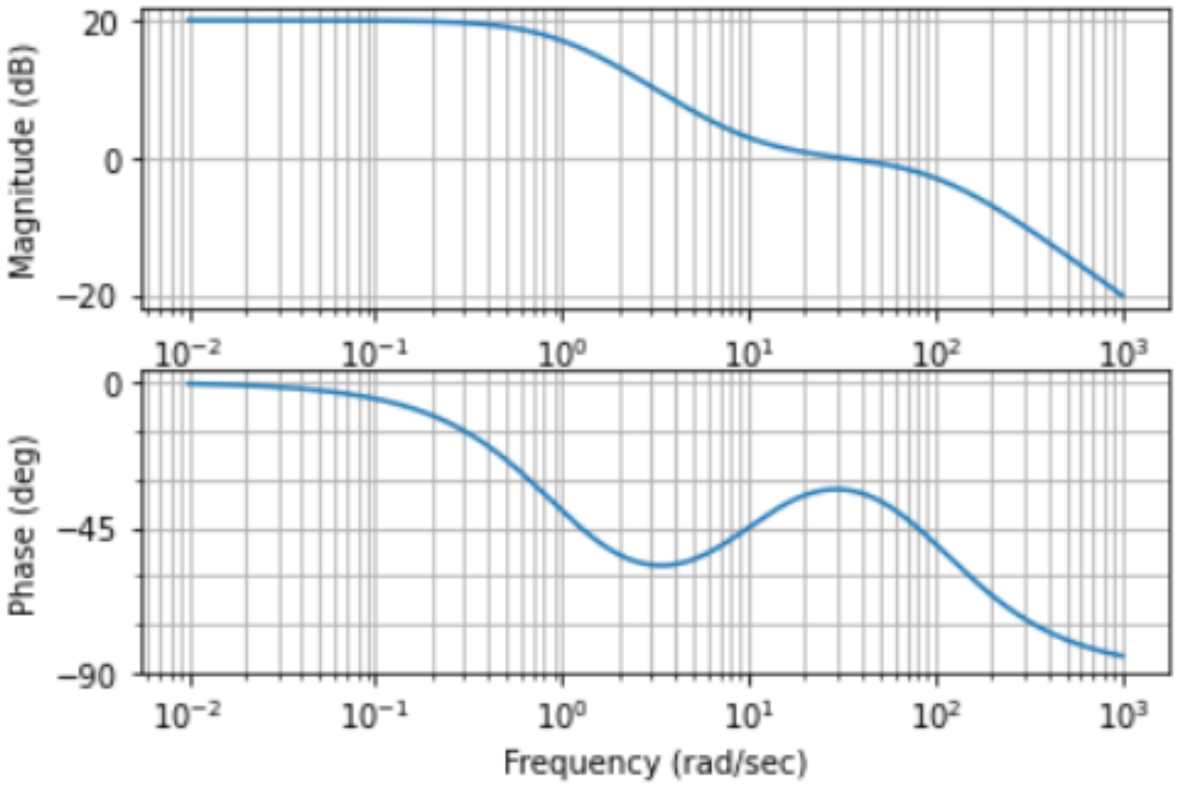
```
-->P = syslin('c',100*(s+10)/((s+1)*(s+100)));

-->bode(P,'rad')
```



Combination (python)

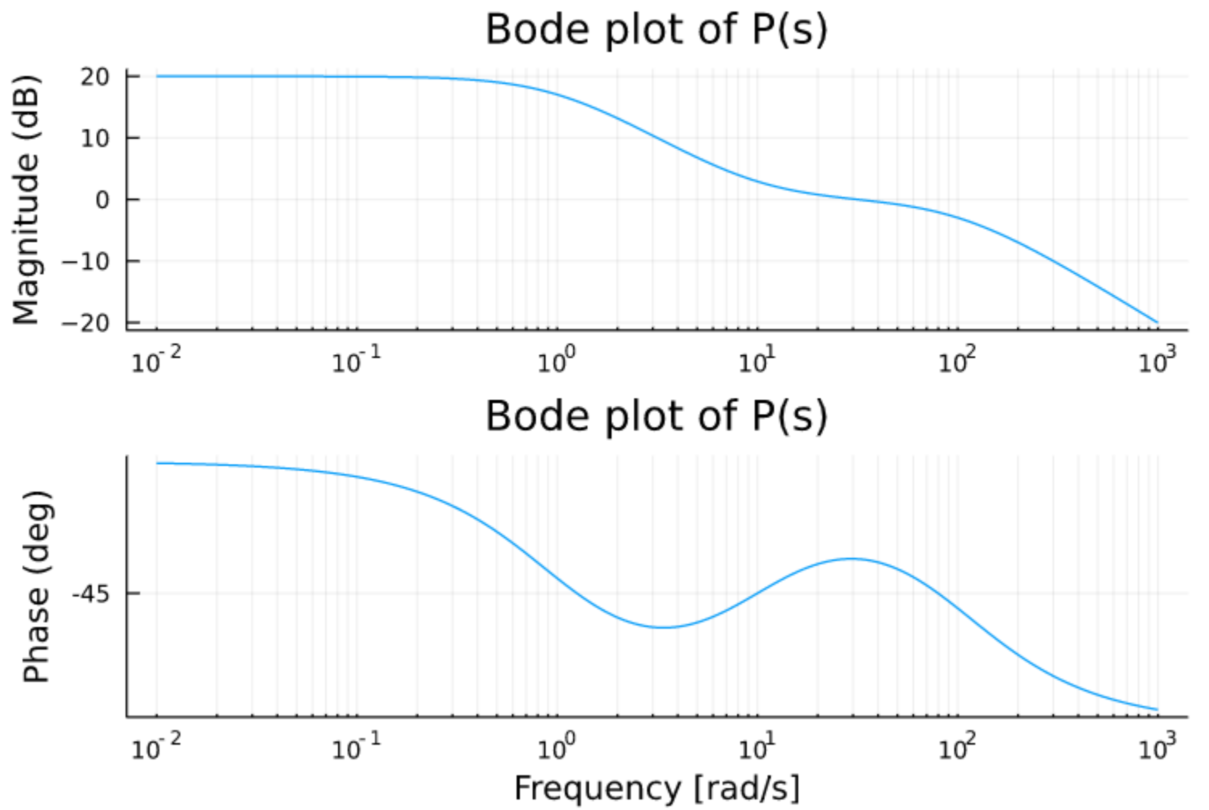
Ex. $$\frac{100(s+10)}{(s+1)(s+100)}$$

In [7]:
```
H = 100*(s+10)/((s+1)*(s+100))
_,_,_ = ctl.bode_plot(H,dB=True,omega_limits=(0.01,1000))
```



Combination (Julia)

Ex. $$\frac{100(s+10)}{(s+1)(s+100)}$$




```
begin
    P = 100(s+10)/((s+1)*(s+100))
    setPlotScale("dB")
    w = exp10.(LinRange(-2,3,2000))
    bodeplot(P,w; title="Bode plot of P(s)",legend=false)

end
```

ตัวอย่าง :

พลานต์ข้อต่อหุ่นยนต์ขับเคลื่อนโดยดีซีมอเตอร์

$$P(s) = \frac{1}{10s^2 + 0.1s}$$

ข้อกำหนดการออกแบบ

- ค่าแตกต่างการตามรอยในช่วงสถานะนิ่งเป็นศูนย์
- ขนาดการรบกวนในย่านความถี่ต่ำกว่า 1 rad/s ถูกลดทอนเหลือน้อยกว่า 0.01
- ขนาดสัญญาณรบกวนจากการวัดในย่านความถี่สูงกว่า 100 rad/s ถูกลดทอนเหลือน้อยกว่า 0.1
- ระบบวงปิดเสถียร โดยมีค่าเผื่อเฟสอย่างน้อย 40 องศา

เงื่อนไขสำหรับ Loopshaping

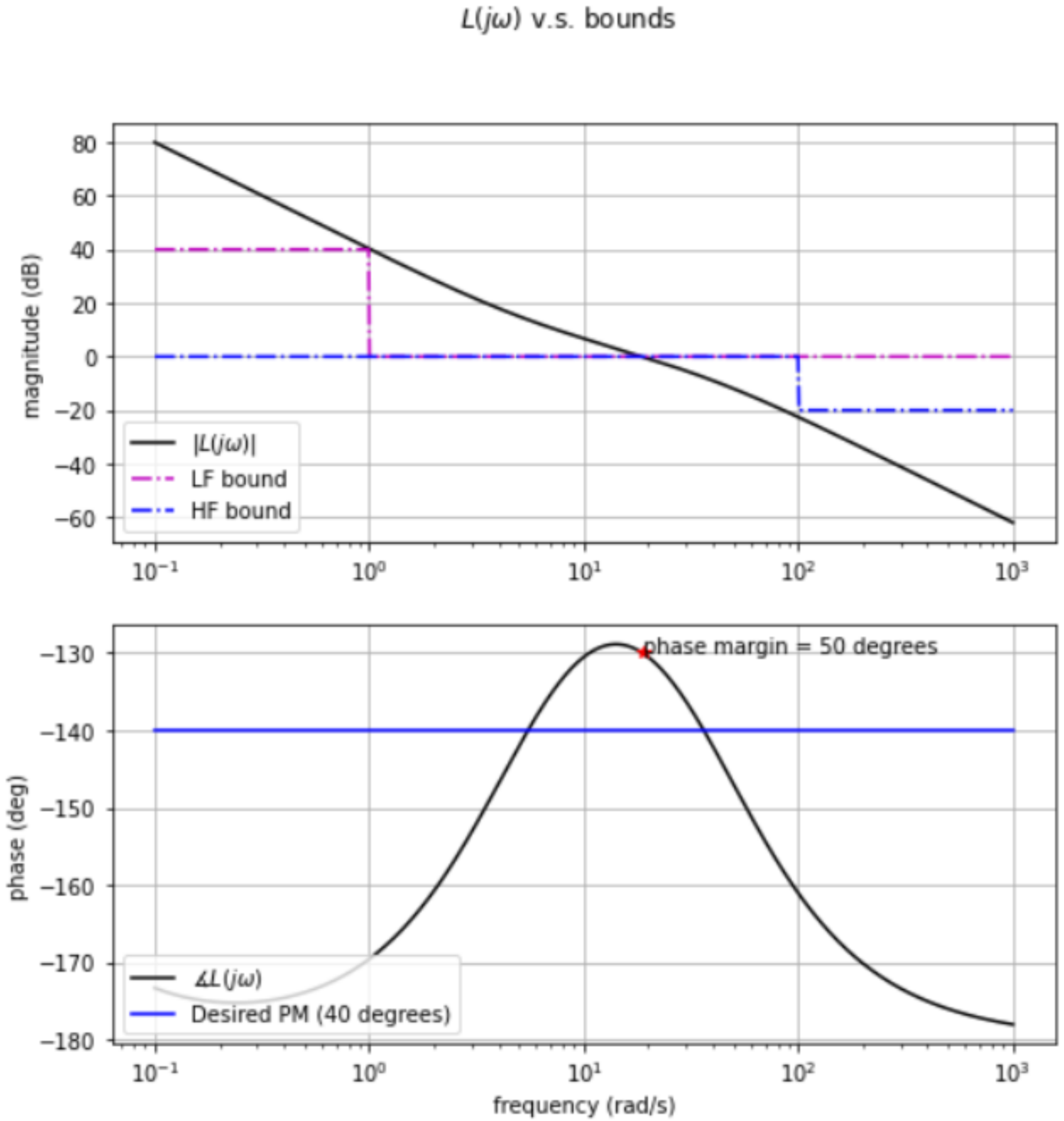
- $L(s)$ ต้องมี integrator สังเกตว่าสำหรับตัวอย่างนี้มี integrator อยู่ใน $P(s)$
- $|S(j\omega)| \leq -40$ dB $\rightarrow |L(j\omega)| \geq 40$ dB ในช่วงความถี่ต่ำกว่า 1 rad/s
- $|T(j\omega)| \leq -20$ dB $\rightarrow |L(j\omega)| \leq -20$ dB ในช่วงความถี่สูงกว่า 100 rad/s
- $L(j\omega)$ มีค่าเผื่อเฟสอย่างน้อย 40 องศา หรือ $\max|S(j\omega)| \leq 3.3$ dB*

Python : ใช้ฟังก์ชัน lshape()ใน ceb_m3.ipynb

Julia : ceb_m3.jl

ตัวอย่างตัวควบคุมที่ผ่านเงื่อนไข (Python)

```
In [12]: C = 8000*(s+5)/(s+40)
         lshape(C,P, lf, lfb, hf, hfb,pm )
```



ตัวอย่างตัวควบคุมที่ผ่านเงื่อนไข (Julia)

Low freq = 1.0 rad/s; Low freq bounds = 40 dB

High freq = 100 rad/s; High freq bounds = -20 dB; Phase margin = 40 degrees

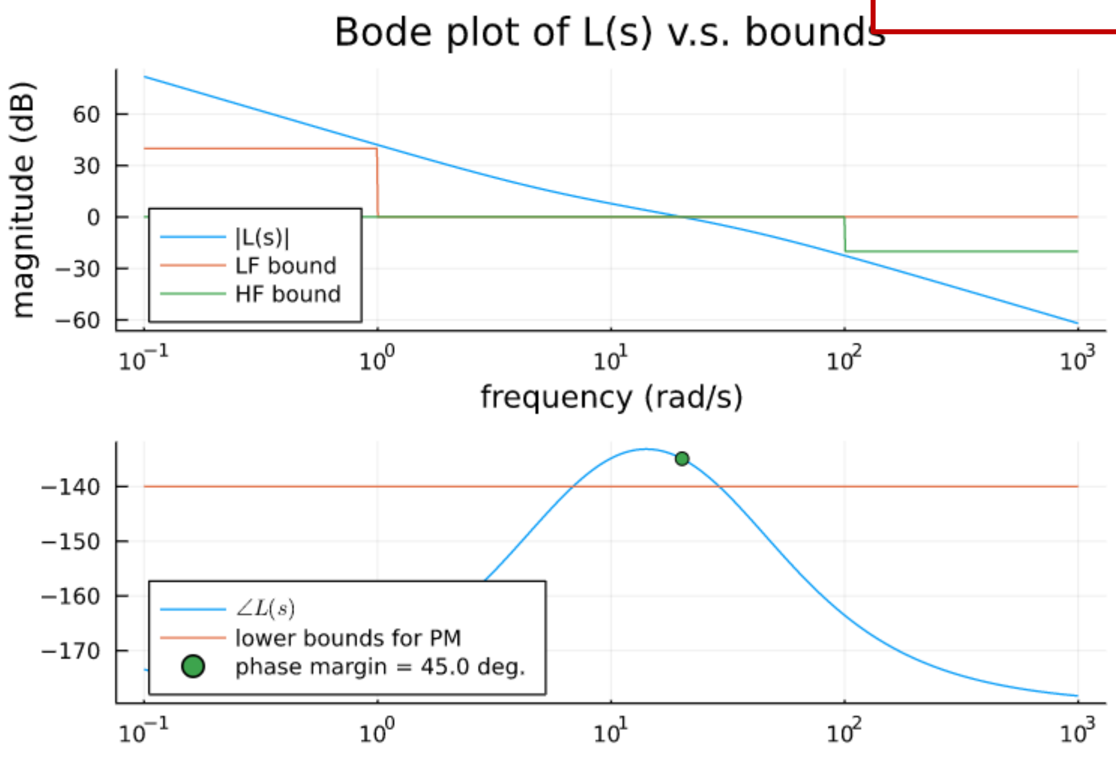
$$C(s) = k\frac{p(s+z)}{z(s+p)}$$

k = 1246 ;p = 35.72 ;z =

C(s) = k only

The resulting controller C(s) is

```
TransferFunction{Continuous, ControlSystems.S
44507.119999999995s + 249284.37911999997
--------------------------------------
      5.601s + 200.06771999999998

Continuous-time transfer function model
```

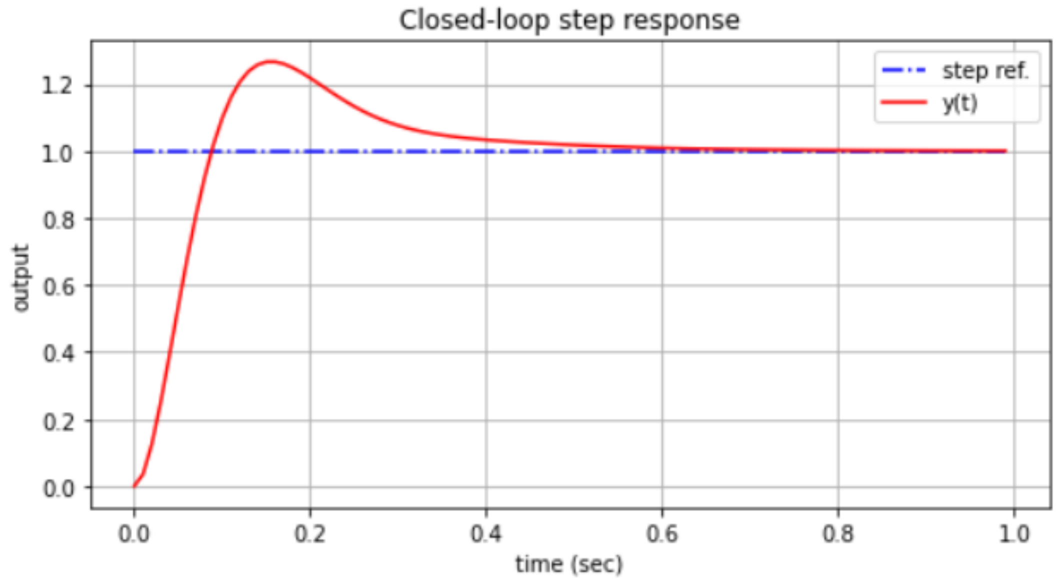



```
lshape(C, P, lf, lfb, hf, hfb, pm)
```

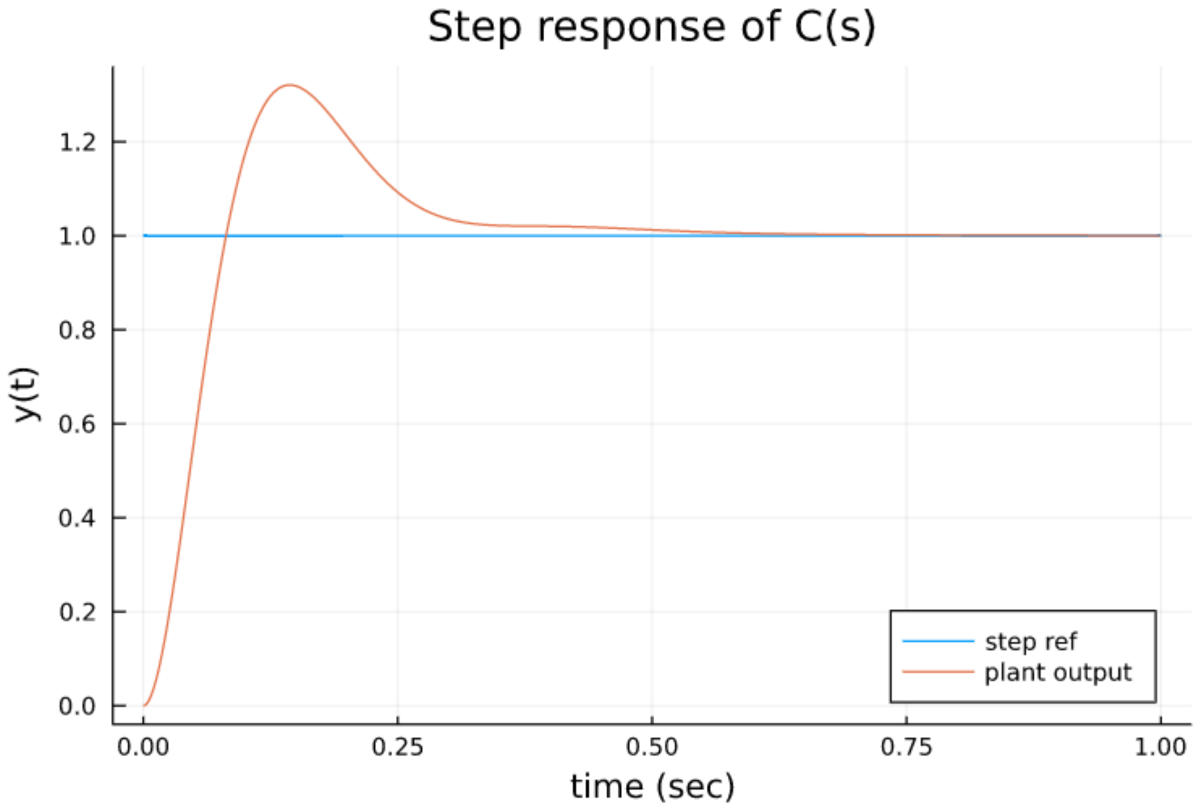
ผลตอบสนองขั้นบันได (python)

The closed-loop step response is simulated as follows.

In [8]:
```
L = C*P
T = L/(1+L)
tvec = np.arange(0,1.0,0.01)
r = np.ones(tvec.shape)
tout, y = ctl.step_response(T,tvec)
plt.figure(figsize=(8,4))
plt.plot(tout,r,'b-.',tout,y,'r-')
plt.grid('True')
plt.xlabel('time (sec)')
plt.ylabel('output')
plt.legend(['step ref.','y(t)'])
plt.title('Closed-loop step response')
plt.show()
```

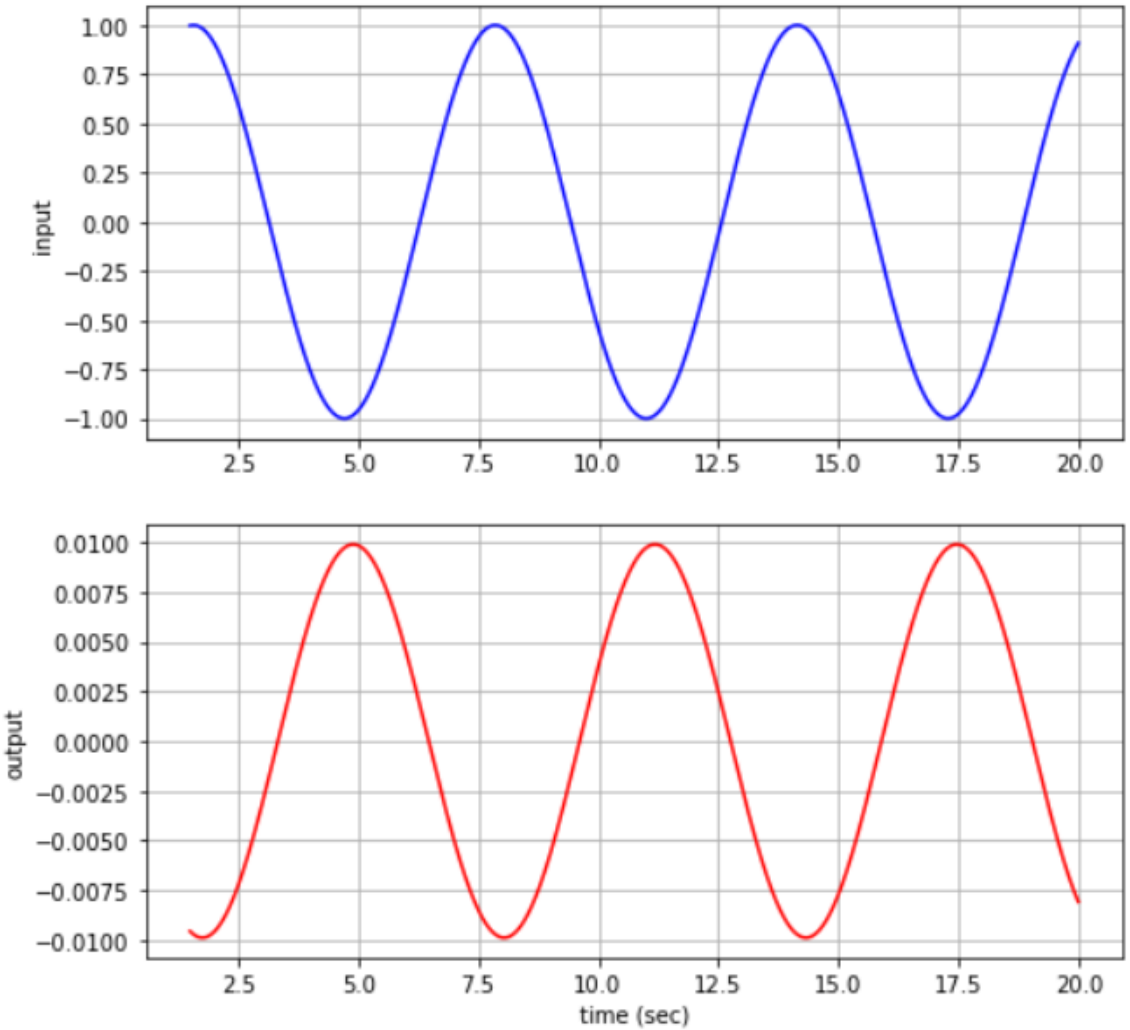


ผลตอบสนองขั้นบันได (Julia)




```
begin
    L = C*P
    T = minreal(L/(1+L))
    tvec1 = collect(Float64,0:0.001:1)
    y1,t1,x1 = step(T,tvec1)
    r1 = ones(size(t1))
    plot(t1,r1, label="step ref")
    plot!(t1,y1, label="plant output",xlabel="time (sec)",ylabel="y(t)",title="Step
response of C(s)",legend=:bottomright)
end
```

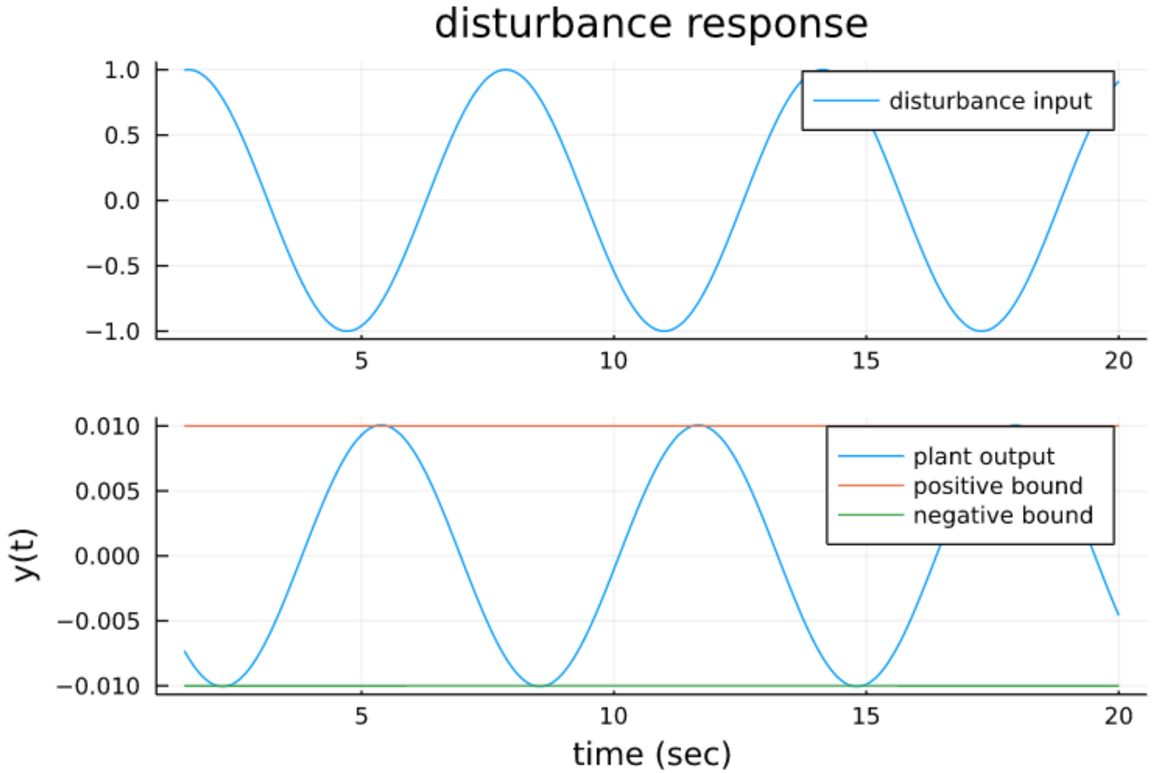
สมรรถนะการลดทอนการรบกวน d (python)



สมรรถนะการลดทอนการรบกวน d (Julia)

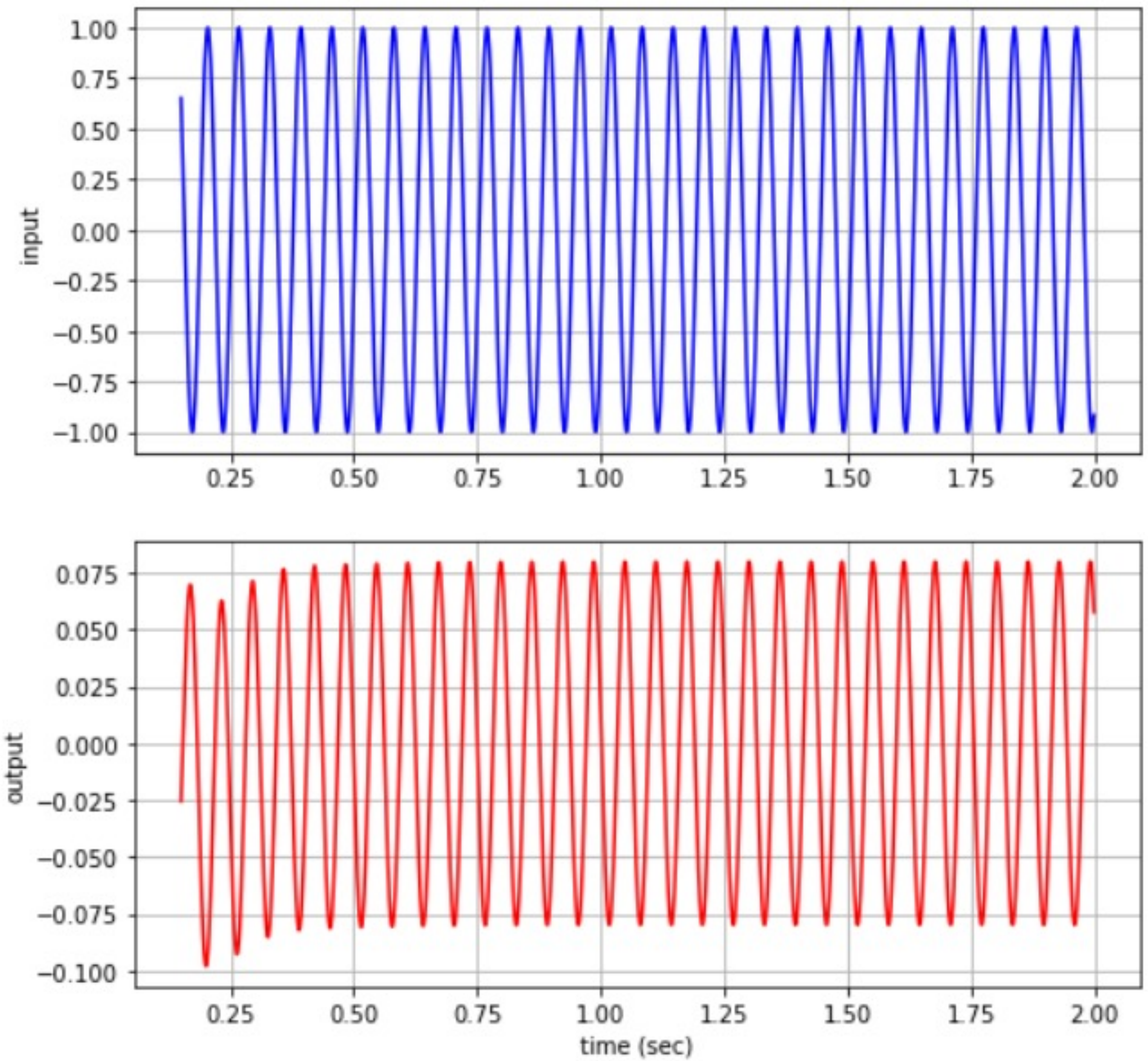
disturbance attenuation specs = 0.01

d(t) amplitude = 1.0 ; d(t) frequency = 1 rad/s; final time = 20 sec




```
begin
    S = minreal(1/(1+L))
    tvec2 = collect(Float64,0:0.01:t_final)
    u2 = da*sin.(df*tvec2)
    abnd = 0.01*da
    plot_response(S,u2,tvec2,abnd,"disturbance")
end
```

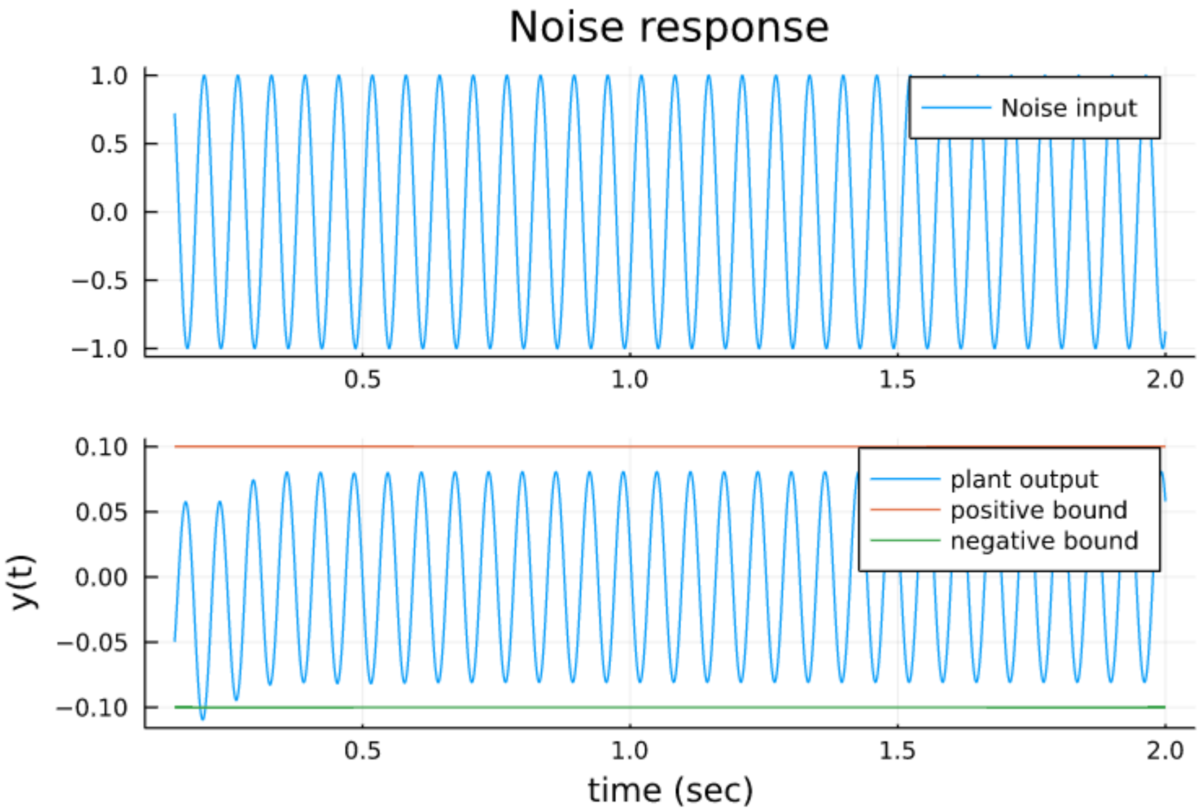
สมรรถนะการลดทอน noise (Python)



สมรรถนะการลดทอน noise (Julia)

noise attenuation specs = 0.1

n(t) amplitude = 1.0 ; n(t) frequency = 100 rad/s; final time = 2 sec




```
begin
    tvec3 = collect(Float64,0:0.001:tn_final)
    u3 = na*sin.(nf*tvec3)
    nbnd = 0.1*na
    plot_response(T,u3,tvec3,nbnd,"Noise")
end
```

HW#3 : (2 สัปดาห์)

พลานต์ข้อต่อหุ่นยนต์ขับเคลื่อนโดยดีซีมอเตอร์

$$P(s) = \frac{1}{7s^2 + 0.05s}$$

ข้อกำหนดการออกแบบ

- ค่าแตกต่างการตามรอยในช่วงสถานะนิ่งเป็นศูนย์
- ขนาดการรบกวนในย่านความถี่ต่ำกว่า 0.5 rad/s ถูกลดทอนเหลือน้อยกว่า 0.05
- ขนาดสัญญาณรบกวนจากการวัดในย่านความถี่สูงกว่า 80 rad/s ถูกลดทอนเหลือน้อยกว่า 0.15
- ระบบวงปิดเสถียร โดยมีค่าเผื่อเฟสอย่างน้อย 50 องศา

เงื่อนไขสำหรับ Loopshaping

- $L(s)$ ต้องมี integrator สังเกตว่าสำหรับตัวอย่างนี้มี integrator อยู่ใน $P(s)$
- $|S(j\omega)| \leq$? dB → $|L(j\omega)| \geq$? dB ในช่วงความถี่ต่ำกว่า 3 rad/s
- $|T(j\omega)| \leq$? dB → $|L(j\omega)| \leq$? dB ในช่วงความถี่สูงกว่า 200 rad/s
- $L(j\omega)$ มีค่าเผื่อเฟสอย่างน้อย 50 องศา

ใช้ HW3 notebook (Jupyter หรือ Pluto) โดยแสดง 1. ตัวควบคุมที่ออกแบบได้ 2. ผลตอบสนองความถี่ L(jw) เทียบกับ bounds 3. ผลตอบสนองขั้นบันได (tracking) 4. การขจัดการรบกวน (disturbance attenuation) 5. การขจัดสัญญาณรบกวนจากการวัด (noise attenuation) ทั้งหมดต้องแสดงให้เห็นว่าได้ตามข้อกำหนดทุกข้อ