

Control Systems

G V V Sharma*

CONTENTS

1	Routh Hurwitz Criterion	1
2	Bode Plot	1
3	Compensators	2
4	Nyquist Plot	2

Abstract—This manual is an introduction to control systems based on GATE problems. Links to sample Python codes are available in the text.

Download python codes using

```
svn co https://github.com/gadepall/school/trunk/control/codes
```

1 ROUTH HURWITZ CRITERION

2 BODE PLOT

2.1. For an LTI system, the Bode plot for its gain is as illustrated in the Fig. 2.1 The number of system poles N_p and number of system zeros N_z in the frequency range $1 \text{ Hz} \leq f \leq 10^7 \text{ Hz}$ is

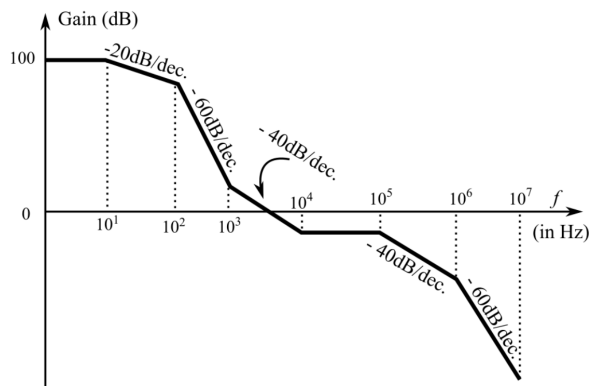


Fig. 2.1

Solution:- Let us consider a generalized transfer gain

$$H(s) = k \frac{(s - z_1)(s - z_2) \dots (s - z_{m-1})(s - z_m)}{(s - p_1)(s - p_2) \dots (s - p_{n-1})(s - p_n)}$$

$$\begin{aligned} \text{Gain} &= 20 \log |H(s)| = 20 \log |k| + 20 \log |s - z_1| \\ &+ 20 \log |s - z_2| + \dots + 20 \log |s - z_m| - 20 \log |s - p_1| \\ &- 20 \log |s - p_2| - \dots - 20 \log |s - p_n| \quad (2.1.1) \end{aligned}$$

Let us consider a $20 \log |s - z_1|$

Let $s = j\omega$

$$20 \log |s - z_1| = 20 \log \left| \sqrt{\omega^2 + z_1^2} \right|$$

Based on log scale plot approximations, to the left of z_1 $\omega \ll z_1$ and towards right $\omega \gg z_1$

For $\omega < z_1$

$$\begin{aligned} 20 \log |s - z_1| &= 20 \log \left| \sqrt{\omega^2 + z_1^2} \right| = 20 \log |z_1| \\ &= \text{constant i.e. Slope} = 0 \end{aligned}$$

For $\omega > z_1$

$$\begin{aligned} 20 \log |s - z_1| &= 20 \log \left| \sqrt{\omega^2 + z_1^2} \right| = 20 \log |\omega| \\ \text{i.e Slope} &= 20 \end{aligned}$$

When a zero is encountered the slope always increases by 20 dB/decade

Doing similar analysis for $-20 \log |s - p_1|$
We conclude

When a pole is encountered the slope always decreases by 20 dB/decade

$$\text{Slope} = \frac{d(20 \log H(f))}{df}$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

$$Slope = \begin{cases} 0 & 0 < f < 10^1 \\ -20 & 10 < f < 10^2 \\ -60 & 10^2 < f < 10^3 \\ -40 & 10^3 < f < 10^4 \\ 0 & 10^4 < f < 10^5 \\ -40 & 10^5 < f < 10^6 \\ -60 & 10^6 < f < 10^7 \end{cases}$$

$\Delta Slope$ = Change in slope at f

$$\Delta Slope = \begin{cases} -20 & f = 10^1 \\ -40 & f = 10^2 \\ +20 & f = 10^3 \\ +40 & f = 10^4 \\ -40 & f = 10^5 \\ -20 & f = 10^6 \end{cases}$$

Final Transfer function is

$$H(f) = \frac{K(f + 10^3)(f + 10^4)^2}{(f + 10^1)(f + 10^2)^2(f + 10^5)^2(f + 10^6)}$$

$$N_p = 6 \quad N_z = 3$$

Python plot of the obtained transfer function is shown in fig 2.1

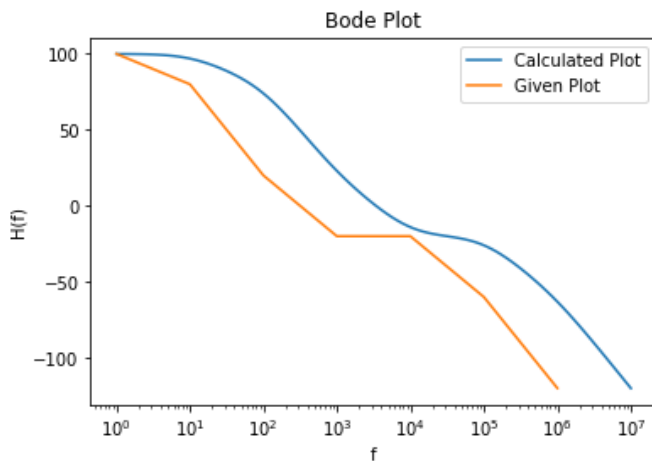


Fig. 2.1

3 COMPENSATORS

4 NYQUIST PLOT