***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 Signal Flow Graph

1.1 Mason's Gain Formula

1.2 Matrix Formula

2 Bode Plot

2.1 Introduction

2.2 Example

3 Second order System

3.1 Damping

3.2 Example

4 Routh Hurwitz Criterion

4.1 Routh Array

4.2 Marginal Stability

4.3 Stability

4.4 Example

5 State-Space Model

5.1 Controllability and Observability

5.2 Second Order System

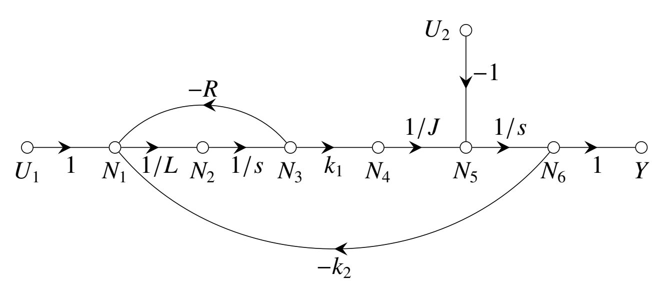
5.3 Example

5.4 Example

6 Nyquist Plot

6.1 Polar plots

6.1. In a system whose signal flow graph is shown in the figure, $U_1(s)$ and $U_2(s)$ are inputs. The transfer function $\frac{Y(s)}{U_1(s)}$ is



Solution:

$$\left.\frac{Y(s)}{U_1(s)}\right|_{U_2(s)=0} \tag{6.1.1}$$

Using Matrix Formula:
The transition equations are

$$N_1 = U_1 - RN_3 - k_2N_6 \tag{6.1.2}$$

$$N_2 = \frac{N_1}{L} \tag{6.1.3}$$

$$N_3 = \frac{N_2}{s} \tag{6.1.4}$$

$$N_4 = k_1N_3 \tag{6.1.5}$$

$$N_5 = \frac{N_4}{J} \tag{6.1.6}$$

$$N_6 = \frac{N_5}{s} \tag{6.1.7}$$

State Transition Matrix :

$$\mathbf{T} = \begin{pmatrix} 0 & 0 & -R & 0 & 0 & -k_2 \\ \frac{1}{L} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{s} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{J} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{s} & 0 \end{pmatrix} \tag{6.1.8}$$

$$\mathbf{U} = (\mathbf{I} - \mathbf{T})^{-1} \tag{6.1.9}$$

$$(\mathbf{I} - \mathbf{T}) = \begin{pmatrix} 1 & 0 & R & 0 & 0 & k_2 \\ \frac{-1}{L} & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{s} & 1 & 0 & 0 & 0 \\ 0 & 0 & -k_2 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{J} & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{s} & 1 \end{pmatrix} \tag{6.1.10}$$

U_{50} will be the gain of the system

$$\mathbf{U_{50}} = \frac{\begin{vmatrix} \frac{-1}{L} & 1 & 0 & 0 & 0 \\ 0 & \frac{-1}{s} & 1 & 0 & 0 \\ 0 & 0 & -k_2 & 1 & 0 \\ 0 & 0 & 0 & \frac{-1}{J} & 1 \\ 0 & 0 & 0 & 0 & \frac{-1}{s} \end{vmatrix}}{\begin{pmatrix} 1 & 0 & R & 0 & 0 & k_2 \\ \frac{-1}{L} & 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{-1}{s} & 1 & 0 & 0 & 0 \\ 0 & 0 & -k_2 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{-1}{J} & 1 & 0 \\ 0 & 0 & 0 & 0 & \frac{-1}{s} & 1 \end{pmatrix}} \tag{6.1.11}$$

Gain can be found by cofactor expansion or else by running the code in (1.2.5). Gain obtained is

$$\frac{Y(s)}{U_1(s)} = \frac{k_1}{s^2LJ + sRJ + k_1k_2} \tag{6.1.12}$$

The following code generates the transfer function:

```
codes/ee18btech11007/MasonsGain.py
```

7 Compensators

7.1 Phase Lead

7.2 Example

8 Gain Margin

8.1 Introduction

8.2 Example

9 Phase Margin

10 Oscillator

10.1 Introduction

10.2 Example