Bangladesh University of Business & Technology
Control System Lab
EEE 402

Experiment No: 03
Experiment Name: Stability Analysis Using ROUTH- HURWITZ Criteria.

Stability Analysis

- Total response of a system is the sum of the forced and natural responses.

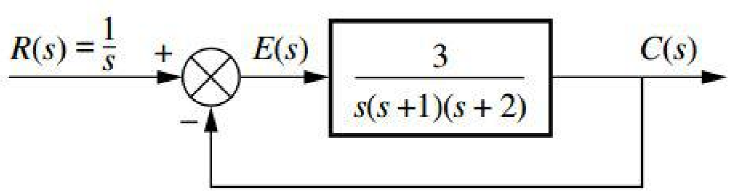
$$C(t) = C\text{-}forced(t) + C\text{-}natural(t)$$

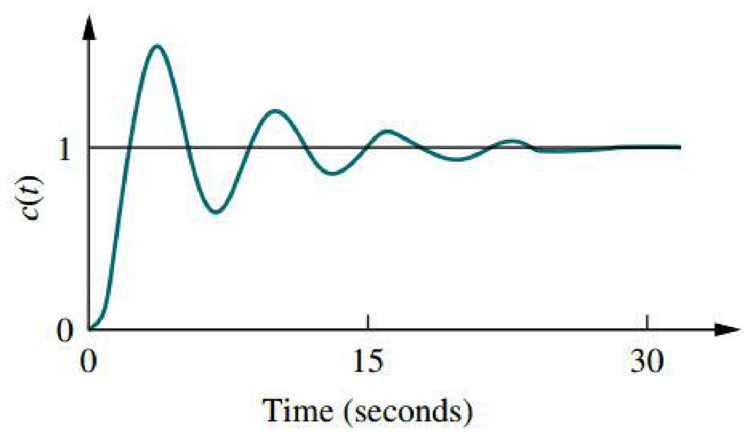
- A linear, time-invariant system is stable if the natural response approaches zero as time approaches infinity. A linear, time-invariant system is unstable if the natural response grows without bound as time approaches infinity. A linear, time-invariant system is marginally stable if the natural response neither decays nor grows but remains constant or oscillates as time approaches infinity. Thus, the definition of stability implies that only the forced response remains as the natural response approaches zero.
- Another definition of stability using the total response (BIBO):
  A system is stable if every bounded input yields a bounded output.
  A system is unstable if any bounded input yields an unbounded output.
- Physically, an unstable system whose natural response grows without bound can cause damage to the system, to adjacent property, or to human life.

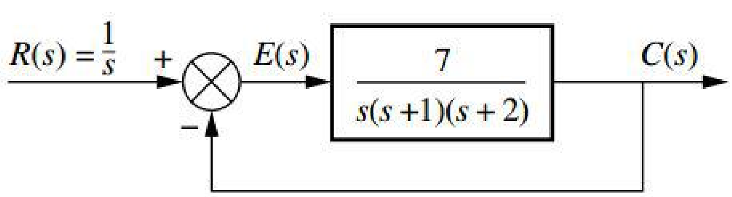
Now the question is - how do we determine if a system is stable?

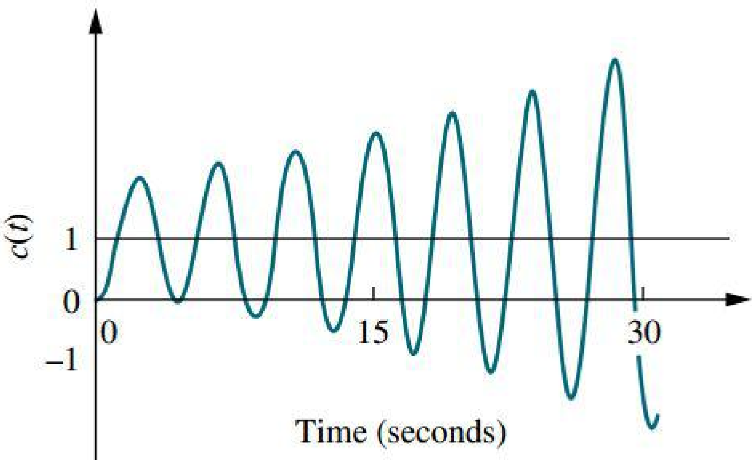
- Poles in the left half-plane (lhp) yield pure exponential decay. Hence, stable systems have closed-loop transfer functions with poles only in the left half-plane. Poles in the right half-plane (rhp) yield pure exponentially increasing responses. Thus, if the closed-loop system poles are in the right half of the s-plane and hence have a positive real part, the system is unstable. Marginally stable systems have closed-loop transfer functions with imaginary axis poles.

Look into the response of the systems below –









(b)

- Careful!!! You need to calculate the closed-loop transfer function to figure out the stability, not the open-loop transfer function.

Routh-Hurwitz Criterion

- Using this method, we can tell how many closed-loop system poles are in the left half-plane, in the right half-plane, and on the jw-axis. (Notice that we say how many, not where.) We can find the number of poles in each section of the s-plane, but we cannot find their coordinates.

- The method requires two steps: (1) Generate a data table called a Routh table and (2) interpret the Routh table to tell how many closed-loop system poles are in the left half-plane, the right half-plane, and on the jv-axis.

TABLE 6.1 Initial layout for Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2			
s^1			
s^0			

TABLE 6.2 Completed Routh table

s^4	a_4	a_2	a_0
s^3	a_3	a_1	0
s^2	$\frac{-\begin{vmatrix} a_4 & a_2 \\ a_3 & a_1 \end{vmatrix}}{a_3} = b_1$	$\frac{-\begin{vmatrix} a_4 & a_0 \\ a_3 & 0 \end{vmatrix}}{a_3} = b_2$	$\frac{-\begin{vmatrix} a_4 & 0 \\ a_3 & 0 \end{vmatrix}}{a_3} = 0$
s^1	$\frac{-\begin{vmatrix} a_3 & a_1 \\ b_1 & b_2 \end{vmatrix}}{b_1} = c_1$	$\frac{-\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$	$\frac{-\begin{vmatrix} a_3 & 0 \\ b_1 & 0 \end{vmatrix}}{b_1} = 0$
s^0	$\frac{-\begin{vmatrix} b_1 & b_2 \\ c_1 & 0 \end{vmatrix}}{c_1} = d_1$	$\frac{-\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$	$\frac{-\begin{vmatrix} b_1 & 0 \\ c_1 & 0 \end{vmatrix}}{c_1} = 0$

- Here, we work with the characteristic equation of the system.
- Start with the coefficient of the highest power.
- The number of roots of the polynomial that are in the right half-plane is equal to the number of sign changes in the first column.
- If the closed-loop transfer function has all poles in the left half of the s-plane, the system is stable. Thus, a system is stable if there are no sign changes in the first column of the Routh table.

Matlab Program

```
%% first it is required to get first two row of routh matrix

e=input('enter the coefficients of characteristic equation: ');

disp('-------------------------------------------------------------------------- ')

l=length(e);

m=mod(l,2);

if m==0

 a=zeros(1,(l/2));

 b=zeros(1,(l/2));
```

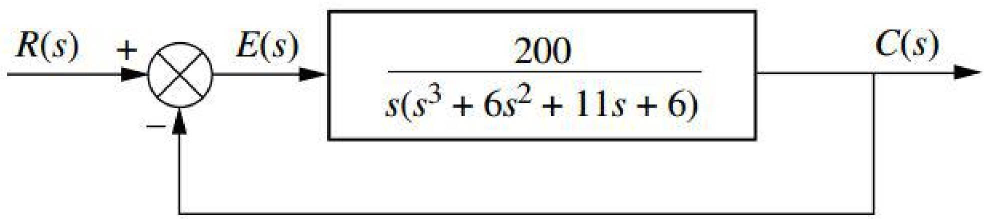
```
for i=1:(l/2)
 a(i)=e((2*i)-1);
 b(i)=e(2*i);
 end
 else
 e1=[e 0];
 a=rand(1,((l+1)/2));
 b=rand(1,((l+1)/2));
for i=1:((l+1)/2)
 a(i)=e1((2*i)-1);
 b(i)=e1(2*i);
 end
end

 %% now we generate the remaining rows of routh matrix
 l1=length(a);
 c=zeros(l,l1);
 c(1,:)=a;
c(2,:)=b;
for m=3:l
for n=1:l1 -1
 c(m,n)=-(1/c(m-1,1))*det([c((m-2),1) c((m-2),(n+1));c((m-1),1) c((m-1),(n+1))]);
 end
 end
 disp('the routh matrix:')
 disp(c)
```


Lecturer, Dept. of EEE, BUBT

```
%% now we check the stablity of system
if c(:,1)>0
 disp('System is Stable')
else
 disp('System is Unstable');
end
```

Exercise:



PROBLEM: Make a Routh table and tell how many roots of the following polynomial are in the right half-plane and in the left half-plane.

$$P(s) = 3s^7 + 9s^6 + 6s^5 + 4s^4 + 7s^3 + 8s^2 + 2s + 6$$

- If you have an entire row of zeroes or zero in the first column, then additional modification in the procedure as well as in the Matlab program is required.
- If the first element of a row is zero, division by zero would be required to form the next row. To avoid this phenomenon, an epsilon, e, is assigned to replace the zero in

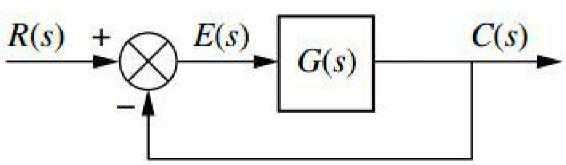
Lecturer, Dept. of EEE, BUBT

the first column. The value of e is assumed to be very close to zero. Other procedures remain the same.

Can you write the Matlab program for this kind of scenerio?

$$T(s) = \frac{10}{s^5 + 2s^4 + 3s^3 + 6s^2 + 5s + 3}$$

<u>Exercise:</u>



$$G(s) = \frac{1}{4s^2(s^2+1)}$$

$$G(s) = \frac{240}{(s+1)(s+2)(s+3)(s+4)}$$

Lecturer, Dept. of EEE, BUBT