Scilab Tutorial

Phani Raj[1]

[1]Department of Electrical Engineering
Indian Institute of Technology Bombay

February 5, 2019

Outline

Creating transfer functions

To create a transfer function, one must first create polynomials and express the transfer function as ratio of polynomials. The following command creates a symbol $'s'$ in which one can create a polynomial.

- s=poly(0,'s')

Now we can create polynomials in s.

- Den=s*(s+2)*(s+5)
- Num=s+1

Now we can create transfer function using keyword 'syslin'.

- Gol=syslin('c',Num,Den)

The preceding command creates a continuous time transfer function $Gol = \frac{s+1}{s(s+2)(s+5)}$. To create discrete time transfer function use $'d'$ in place of $'c'$ in 'syslin' command.

We can perform basic algebraic operations on transfer functions. Assuming a unity negative feedback system with Gol in the forward path, the closed loop transfer function can be computed using basic operations.

- Gcl=Gol/(1+Gol)

Step response

To compute step response, first create an array of time instants for which you would like to compute the step response. Then use the keyword 'csim' to compute the step response. Computed response can be plotted using scilab plotting tools. The following sequence of commands demonstrates how to compute step response.

- t=0:0.01:20;
- Resp=csim('step',t,Gcl);
- plot(t,Resp)
- xlabel('Time');
- ylabel('Response');
- title('Step response'); %Title of the figure
- xgrid(); %Creates grid

The keyword 'csim' is not limited to calculating step response. We can use it to calculate response due to some arbitrary input string $u(t)$. To do this replace $'step'$ with the input string u (without the invited commas).

State space representation

In order to create state space representation, first create the quartet A, B, C and D. This is done in the following commands.

- A=[0 1 0;0 0 1;-6 -11 -6];
- B=[0;0;1];
- C=[1 0 0];
- D=0;

Now to create a continuous time state space system, we again use the keyword 'syslin'.

- Sys=syslin('c',A,B,C,D)

In order to create a discrete time state space system, replace $'c'$ with $'d'$ in the 'syslin' command.

Step response of a state space representation

First create an array of time instants for which the response is desired. Then invoke 'csim' command. 'csim' command returns both the outputs and the states.

- t=0:0.01:20;
- [Output,States]=csim('step',t,Sys);
- scf(); % Creates a new window for plotting
- plot(t,Output)
- xlabel('Time');ylabel('Output');xgrid();
- scf(); % Creates a new window for plotting
- plot(t,States);
- xlabel('Time');ylabel('States');xgrid();
- legend('x1','x2',x3);

The command 'csim' allows us to calculate response due to an arbitrary input even with non zero initial conditions.

Transfer function to State space and vie-versa

To obtain state space representation of a given transfer matrix, we can use the command 'tf2ss'.

- Sysnew=tf2ss(Gcl)

To obtain transfer matrix of a given state space representation, we can use the keyword 'ss2tf'.

- Gclnew=ss2tf(Sys)

Root Locus

To obtain root locus of a given open loop transfer function, we can use the keyword 'evans'. Let the open loop transfer function be $G_{ol} = \frac{s+10}{s(s+3)(s+8)}$

- evans(Gol)

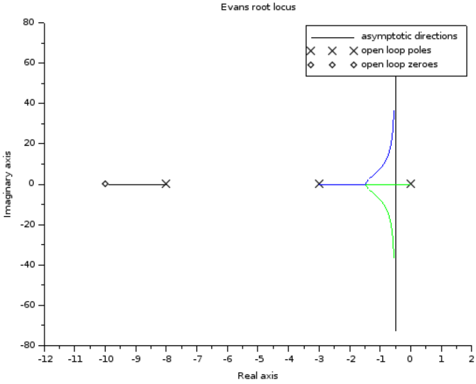


Figure: Root locus

Root Locus

In order to obtain a grid compatible with root locus. Use the keyword 'sgrid'.

- sgrid(0.8,2.5); % sgrid(damping ratio,natural frequency);

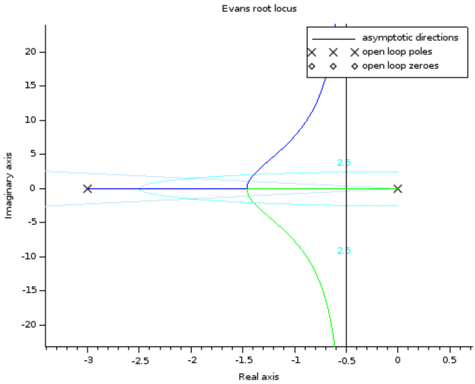
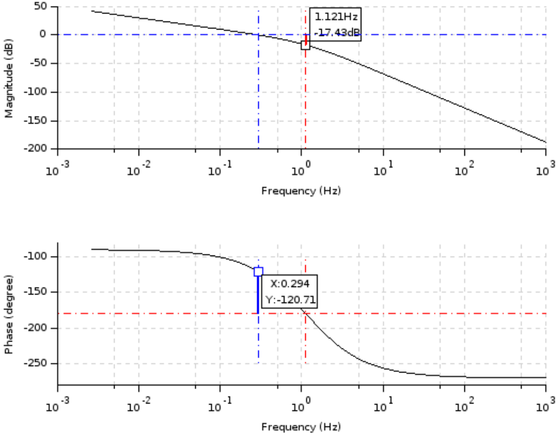


Figure: Root Locus with grid

Bode plot

In order to obtain Bode plot of the given open loop transfer function use the keyword 'bode'. One can also use the keyword 'show_margins'. Let $G_{ol} = \frac{100}{s(s+5)(s+10)}$

- show_margins(Gol,'bode');



Bode plot

In order to compute gain margin and phase margin explicitly, use the keywords 'g_margin' and 'p_margin' respectively.

- [Gain_Margin,Phase_Crossover]=g_margin(Gol)
- [Phase_Margin,Gain_Crossover]=p_margin(Gol)