Linear Control Systems

Lab Experiment 6: Block Diagram Reduction

Objective: The objective of this exercise will be to learn commands in MATLAB that would be used to reduce linear systems block diagram using series, parallel and feedback configuration.

List of Equipment/Software

Following equipment/software is required:

- MATLAB

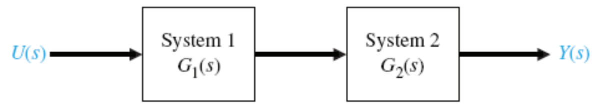
Category Soft-Experiment

Deliverables

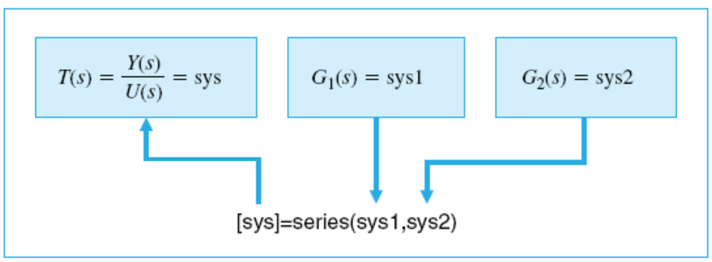
A complete lab report including the following:

- Summarized learning outcomes.
- MATLAB scripts and their results for examples, exercises and Dorf (text book) related material of this lab should be reported properly.

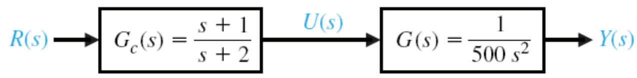
Series configuration: If the two blocks are connected as shown below then the blocks are said to be in series. It would like multiplying two transfer functions. The MATLAB command for the such configuration is “series”.



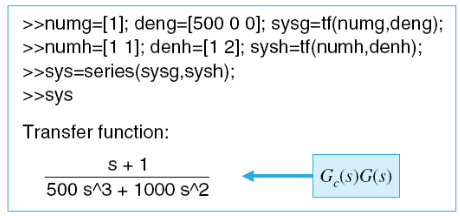
The series command is implemented as shown below:



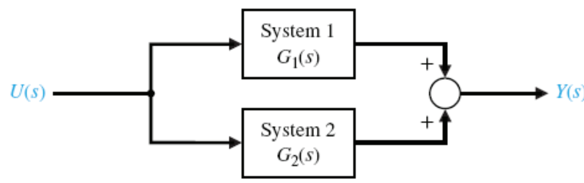
Example 1: Given the transfer functions of individual blocks generate the system transfer function of the block combinations.



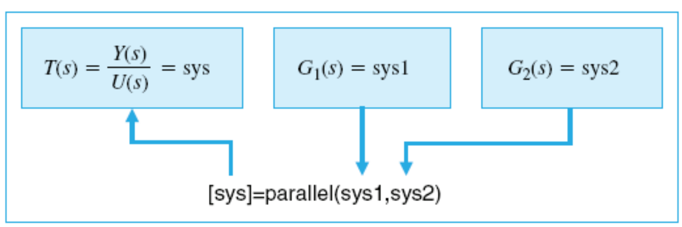
The result is as shown below:



Parallel configuration: If the two blocks are connected as shown below then the blocks are said to be in parallel. It would like adding two transfer functions.

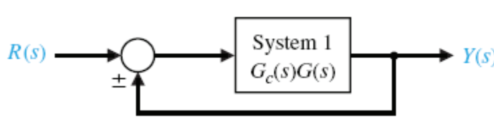


The MATLAB command for implementing a parallel configuration is "parallel" as shown below:

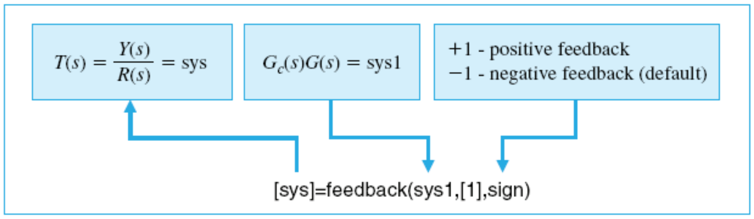


<u>Example 2</u>: For the previous systems defined, modify the MATLAB commands to obtain the overall transfer function when the two blocks are in parallel.

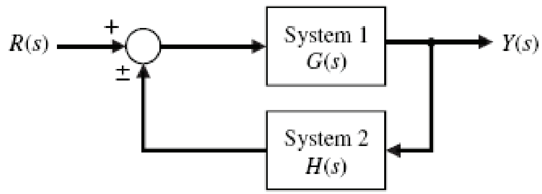
Feedback configuration: If the blocks are connected as shown below then the blocks are said to be in feedback. Notice that in the feedback there is no transfer function H(s) defined. When not specified, H(s) is unity. Such a system is said to be a unity feedback system.



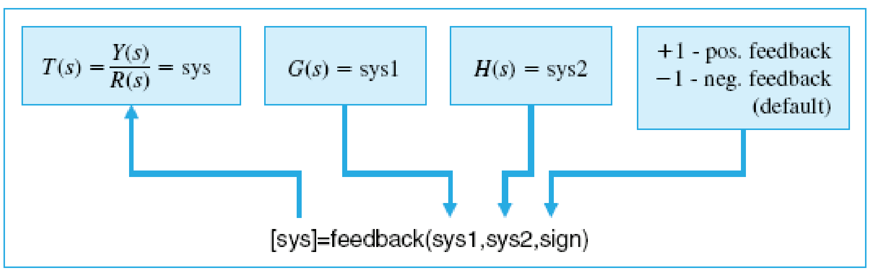
The MATLAB command for implementing a feedback system is "feedback" as shown below:



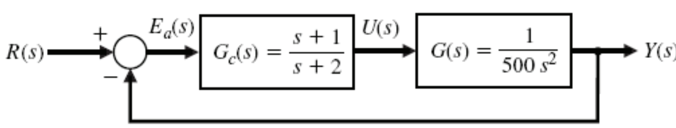
When H(s) is non-unity or specified, such a system is said to be a non-unity feedback system as shown below:



A non-unity feedback system is implemented in MATLAB using the same "feedback" command as shown:



<u>Example 3</u>: Given a unity feedback system as shown in the figure, obtain the overall transfer function using MATLAB:

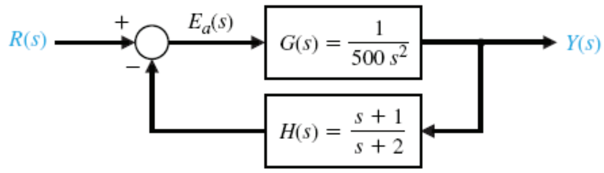


The result is as shown below:

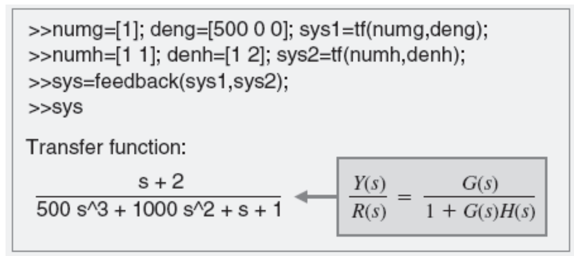
```
>>numg=[1]; deng=[500 0 0]; sys1=tf(numg,deng);
>>numc=[1 1]; denc=[1 2]; sys2=tf(numc,denc);
>>sys3=series(sys1,sys2);
>>sys=feedback(sys3,[1])

Transfer function:
              s + 1
  ------------------------------
  500 s^3 + 1000 s^2 + s + 1
```

$$\frac{Y(s)}{R(s)} = \frac{G_c(s)G(s)}{1 + G_c(s)G(s)}$$

Example 4: Given a non-unity feedback system as shown in the figure, obtain the overall transfer function using MATLAB:



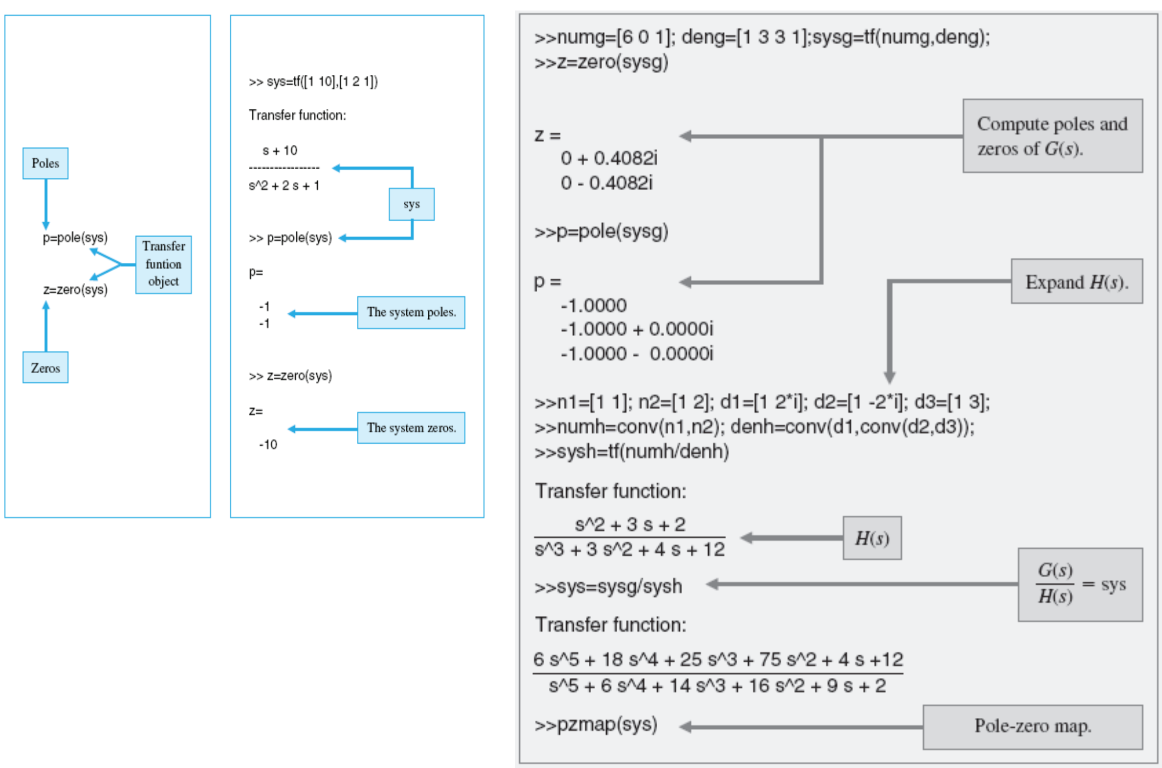
The result is as shown below:



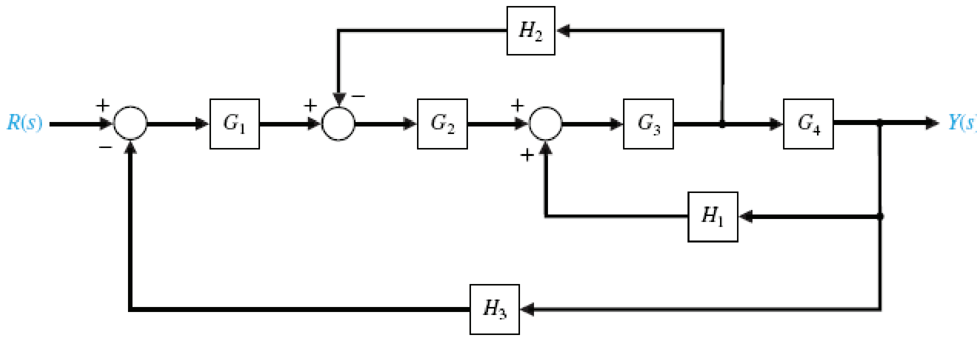
Poles and Zeros of System: To obtain the poles and zeros of the system use the MATLAB command "pole" and "zero" respectively as shown in example 5. You can also use MATLAB command "pzmap" to obtain the same.

Example 5: Given a system transfer function plot the location of the system zeros and poles using the MATLAB pole-zero map command.

For example:



Exercise 1: For the following multi-loop feedback system, get closed loop transfer function and the corresponding pole-zero map of the system.



Given $G_1 = \frac{1}{(s+10)}$; $G_2 = \frac{1}{(s+1)}$; $G_3 = \frac{s^2+1}{(s^2+4s+4)}$; $G_4 = \frac{s+1}{(s+6)}$; $H_1 = \frac{s+1}{(s+2)}$; $H_2 = 2$; $H_3 = 1$ (Reference: Page 113, Chapter 2, Text: Dorf.)

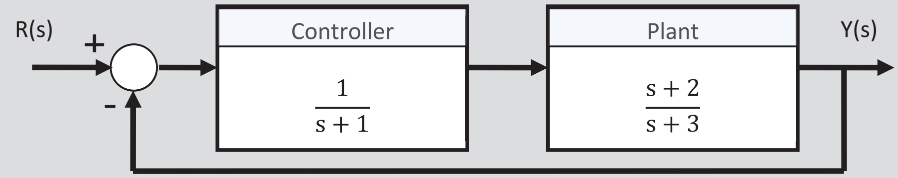
MATLAB solution:

```
>>ng1=[1]; dg1=[1 10]; sysg1=tf(ng1,dg1);
>>ng2=[1]; dg2=[1 1]; sysg2=tf(ng2,dg2);
>>ng3=[1 0 1]; dg3=[1 4 4]; sysg3=tf(ng3,dg3);
>>ng4=[1 1]; dg4=[1 6]; sysg4=tf(ng4,dg4);          Step 1
>>nh1=[1 1]; dh1=[1 2]; sysh1=tf(nh1,dh1);
>>nh2=[2]; dh2=[1]; sysh2=tf(nh2,dh2);
>>nh3=[1]; dh3=[1]; sysh3=tf(nh3,dh3);
>>sys1=sys2/sys4;                                    Step 2
>>sys2=series(sysg3,sysg4);
>>sys3=feedback(sys2,sysh1,+1);                      Step 3
>>sys4=series(sysg2,sys3);
>>sys5=feedback(sys4,sys1);                          Step 4
>>sys6=series(sysg1,sys5);
>>sys=feedback(sys6,[1]);                            Step 5

Transfer function:

              s^5 + 4 s^4 + 6 s^3 + 6 s^2 + 5 s + 2
---------------------------------------------------------------------
12 s^6 + 205 s^5 + 1066 s^4 + 2517 s^3 + 3128 s^2 + 2196 s + 712
```

Instruction: Please refer to Section 2.6 and Section 2.2 in Text by Dorf.

Exercise 2: Consider the feedback system depicted in the figure below

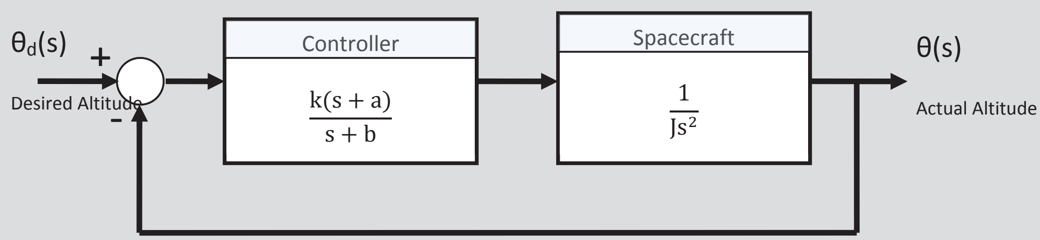
a. Compute the closed-loop transfer function using the ‘series’ and ‘feedback’ functions
b. Obtain the closed-loop system unit step response with the ‘step’ function and verify that final value of the output is 2/5.



Reference: Please see Section 2.5 of Text by Dorf for Exercise 3.

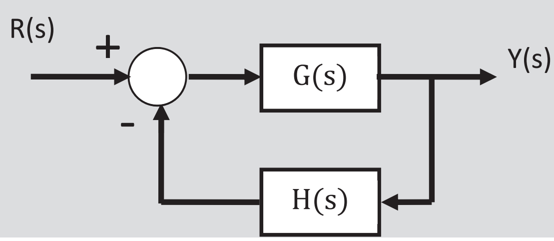
Exercise 3: A satellite single-axis altitude control system can be represented by the block diagram in the figure given. The variables ‘k’, ‘a’ and ‘b’ are controller parameters, and ‘J’ is the spacecraft moment of inertia. Suppose the nominal moment of inertia is ‘J’ = 10.8E8, and the controller parameters are k=10.8E8, a=1, and b=8.

a. Develop an m-file script to compute the closed-loop transfer function
$$T(s) = \theta(s)/\theta_d(s).$$
b. Compute and plot the step response to a 10^{o} step input.
c. The exact moment of inertia is generally unknown and may change slowly with time. Compare the step response performance of the spacecraft when J is reduced by 20% and 50%. Discuss your results.



Reference: Please see Section 2.9 of Text by Dorf for Exercise 4.

Exercise 4: Consider the feedback control system given in figure, where
$$G(s) = \frac{s+1}{s+2} \text{ and } H(s) = \frac{1}{s+1}.$$



a. Using an m-file script, determine the close-loop transfer function.
b. Obtain the pole-zero map using the ‘pzmap’ function. Where are the closed-loop system poles and zeros?
c. Are there any pole-zero cancellations? If so, use the ‘minreal’ function to cancel common poles and zeros in the closed-loop transfer function.
d. Why is it important to cancel common poles and zeros in the transfer function?

~~**Exercise 5**: Do problem CP2.6 from your text~~