NUST School of Electrical Engineering and Computer Science

Faculty Member:_______________ Date: _______________

Semester:______________ Section: _______________

Department of Electrical Engineering

EE-379: Control Systems

LAB 5: State space, response of systems to various inputs, and interconnections of systems

Student name	Reg. No.	Lab Report Marks /10	Viva /05	Total/15

LAB 5: State space, response of systems to various inputs, and interconnections of systems

1. Objectives

- Learn how to create state space models in MATLAB, Simulink and LabVIEW.
- Learn how to find time response of a system for various inputs in MATLAB, Simulink and LabVIEW.
- Interconnection of systems in MATLAB

2. State space representation in MATLAB

We have already seen two ways to represent models i.e. by a set of differential equations and by transfer functions. In this handout we will learn a third way of representing models called the state space representation. Learning this representation is vital because it is the basis of all the modern control techniques.

It is important to learn all these different types of representations. All of these representations are used in control and modeling literature. Moreover, each representation has its own benefits. For example as you will learn later, it is easy to plot a Bode diagram if you have a transfer function representation. Whereas, for pole placement (a controller design technique), state space representation is more suitable. Nonetheless at this stage you should be aware with these basic system representations.

A state space representation is a mathematical model of a physical system as a set first-order differential equations that relate the input variables, output variables and state variables. The state variables are the smallest possible subset of variables that can represent the entire state of the system at any given time. One advantage of state space is that unlike transfer functions it can also be used for non-linear and/or time variant systems. Don't worry if you don't understand too much about state variables or state space over here. It should be covered in more detail in the theory part of the course.

The state space model of an LTI system is of the following type

$$\frac{dx}{dt} = Ax + Bu$$
$$y = Cx + D$$

where

x is the state vector
y is the output vector
u is the input vector
A is the system matrix

B is the input matrix
C is the output matrix
D is the feed forward matrix

The vectors x, y and u are variables. So all we need to specify a state space representation of an LTI system are four matrices A, B, C and D. In most systems D is a zero matrix, and therefore is not specified explicitly

2.1. State space example

In this section we will give an example of state space representation of a simple RLC circuit.

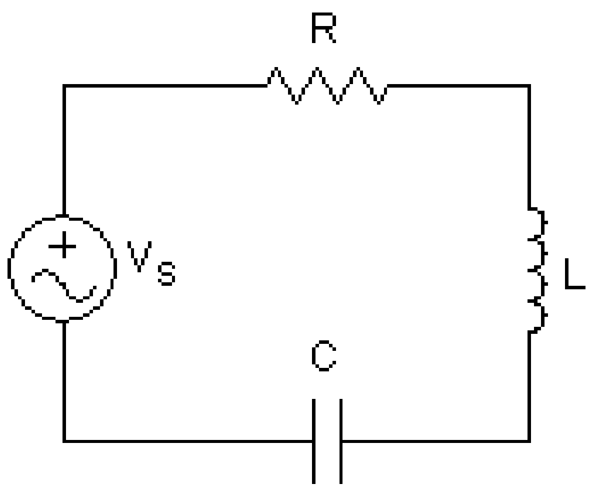


Figure 1: RLC Network

The RLC network in Figure 1 is a second order system. The number of first order equations in state space representation are equal to the system order. Therefore, we will require two simultaneous first order differential equations.

We have the following two equations for our circuit

$$L\frac{di(t)}{dt} + Ri(t) + v_c(t) = v_s(t) \tag{1}$$

$$i(t) = C\frac{dv_c(t)}{dt} \tag{2}$$

In equations (1) and (2), $v_s(t)$ is the input and for the state variables we can take the inductor current $i(t)$ and the capacitor voltage $v_c(t)$. We can rewrite equations (1) and (2) as follows

$$\frac{d}{dt}\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} \frac{-R}{L} & \frac{-1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}.v_s(t) \tag{3}$$

The selected two state variables can be used to solve for any other variable in the circuit. For example the voltage across the inductor can be written in terms of solved state variables and input as

$$v_L(t) = v_s(t) - Ri(t) - v_{c(t)} \tag{4}$$

Let us consider $v_L(t)$ as the output of the circuit. The complete state space representation is given below

$$\frac{d}{dt}\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} = \begin{bmatrix} \frac{-R}{L} & \frac{-1}{L} \\ \frac{1}{C} & 0 \end{bmatrix}\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}.v_s(t) \tag{5}$$

$$v_L(t) = \begin{bmatrix} -R & -1 \end{bmatrix}\begin{bmatrix} i(t) \\ v_c(t) \end{bmatrix} + [1]v_s(t) \tag{6}$$

As an exercise, identify the vector/scalar variable x, y, and u and the matrices A, B, C and D in the above state space representation.

2.2. Constructing a state space model in MATLAB

In MATLAB, the function `ss()` is used for creating state space models. We will use the example of RLC circuit to illustrate how a state space model is created in MATLAB. Follow the steps given below:

i. Define the model parameters of the RLC network

```
R= 1;
L= 1e-3;
C = 1e-6;
```

ii. Define the matrixes

```
A = [-R/L -1/L; 1/C 0];
B = [1/L; 0];
C = [-R -1];
D = [1];
```

iii. Construct the state-space representation

```
rlc_ss = ss(A,B,C,D)
```

Once we have the state space model we can convert it to the transfer function model by using the function tf()

```
rlc_tf = tf(rlc_ss)
```

We can also convert a transfer function to state space model using the function ss()

```
rlc_ss2 = ss(rlc_tf)
```

Exercise

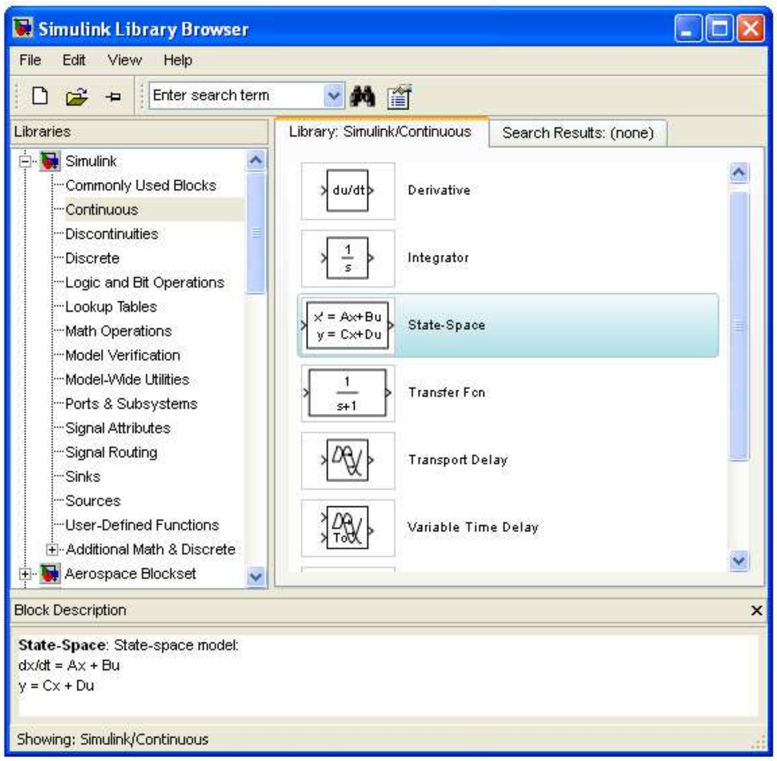
Use the transfer functions of motor speed, motor position and pendulum arm angle to find their state space representation.

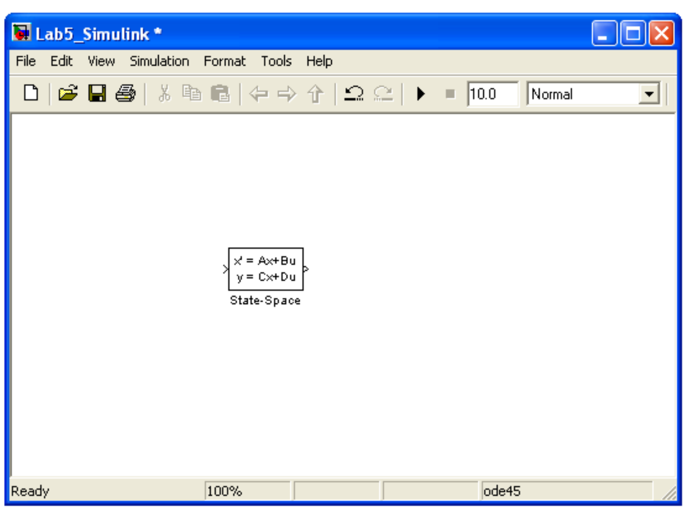
2.3. Constructing a state space model in Simulink

i. Open Simulink

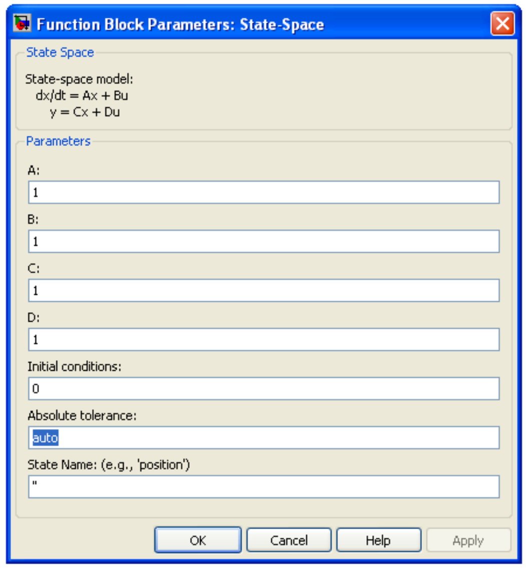
ii. Create a new Simulink model or file

iii. From the Simulink library browser go to Simulink>> Continuous and drag "State space model" into your Simulink file.





iv. Double click the state space model to open the block properties



v. Enter the values for matrices A, B, C and D. You can enter the values of matrices in terms of parameters R,L and C and define these parameters in the model properties, as we did in earlier lab handouts.

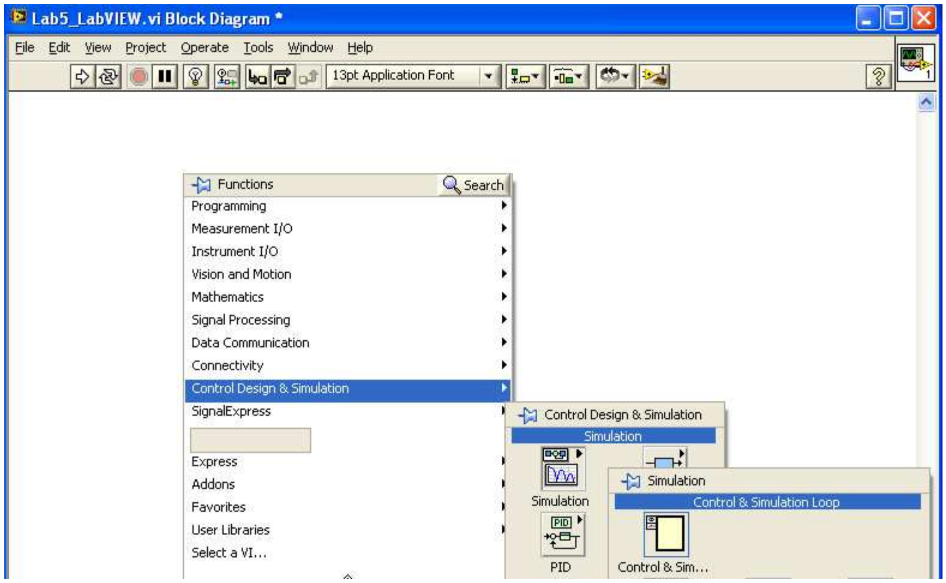
vi. Simulink state space modeling of the RLC network is complete.

2.4. Constructing a state space model in LabVIEW

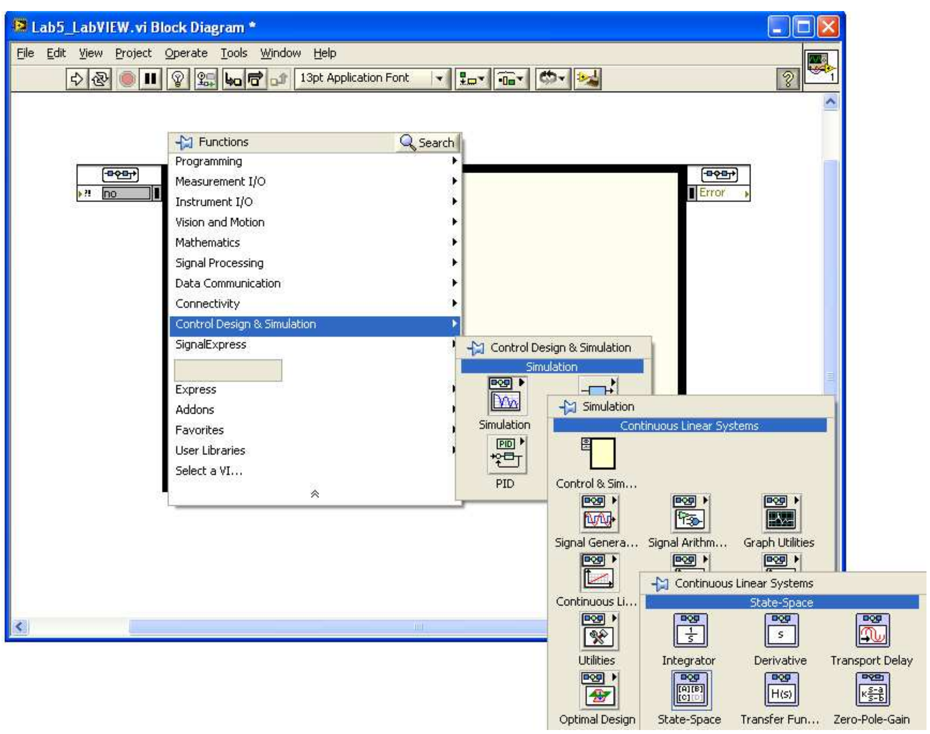
i. Open New VI

ii. Switch to Block diagram using shortcut key Ctrl+E

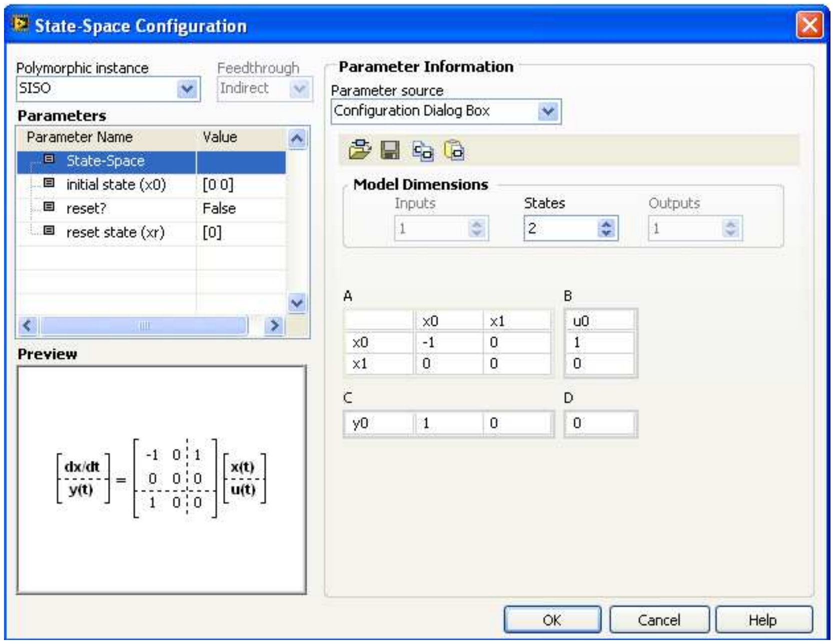
iii. In Block diagram Right click >> function palette >> Control Design and Simulation >> Simulation >> drag “Control and Simulation loop” into your block diagram.



iv. In the block diagram Right click >> function palette >> Control Design and Simulation >> Simulation >> Continuous Linear System >> drag “state space” into your block diagram.



v. Double click the state, a window will appear just like in figure



vi. Enter the values for matrices A, B, C and D to complete the state space modeling.

In MATLAB and SImulink we can enter values of these matrices in terms of parameters (i.e. R, L and C). In LabVIEW we will enter numeric values.

3. Time Response

We have learned different representations of systems: differential equations, transfer functions, state space, and Simulink models. Now we will simulate these models to get the response of the systems for various inputs. We can use any of the system representations to get the response. In this handout we will show some of the ways to get the response.

3.1. Step Response

A step input represents a sudden and constant change in the input signal. For example, if the speed of a motor is 0 rpm and we want it to change to 100 rpm, then our input signal will change from zero to a constant value. This type of input signal is very common in control systems, because we often want systems output to change from one point to another.

The response to a step input is called the step response. The step response indicates that how quickly a system output will reach the desired value i.e. the settling time. The step response can also be used to find the steady state error.

3.1.1. Step Response in MATALB

In MATLAB the step response can be calculated by using the function `step()`. For example the step response of the RLC circuit, we can use either of the following commands

```
step(rlc_ss);
step(rlc_tf);
```

you can also you the zpk representation

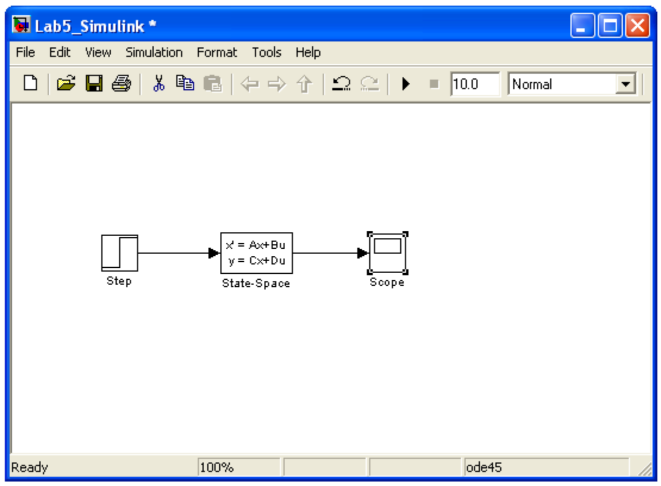
```
rlc_zpk = zpk(rlc_tf);
step(rlc_zpk);
```

Exercise

Using the models of motor and pendulum find the unit step response for motor speed, motor position and pendulum arm angle. You can use any model representation you like. Comment on the response that you get for all the systems.

3.1.2. Step response in Simulink

You can find the step response in Simulink using any representation (transfer function block, state space block or model block). Open one of the model files you have created in the earlier labs for the motor or pendulum. In the Simulink Library browser go to Simulink library >> Source >> drag the Step block into your model. Connect the output of the Step block to the input of the model. In the Simulink Library browser go to Simulink >> Sink >> drag the Scope block into your model.



In the menu, go to Simulation>>Start to start the simulation. Double click the Scope block to see the step response.

If instead of the Step block you place another source, you can see the response for that input.

Exercise

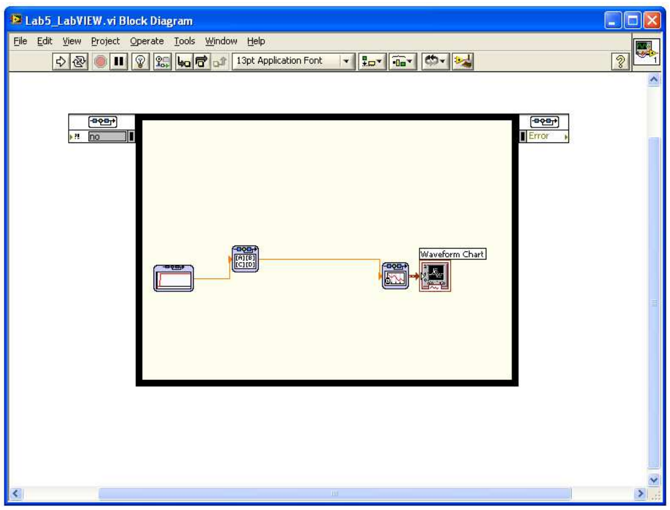
Using the models of motor speed, find the response to the following inputs in Simulink

- Unit step
- Square wave (Pulse Generator block)
- Sine wave (Sine Wave block)

You can adjust the source signal parameter by double clicking on the source signal block e.g. the pulse generator block. Try to change some of the parameters for the pulse generator block and see the effect on the response.

3.1.3. Step response in LabVIEW

Open and VI for a model that you have created in the previous labs. Go to Control Design and Simulation >> Simulation >> Signal generation >>drag “Step Signal” into your block diagram. Go to Control Design and Simulation >> Simulation >> Graph Utilities >> drag “Sim time Waveform” into your block diagram and connect it with the output of the your model block. Go to the front panel and run the VI to see the response.



You can use different sources to see the response to different inputs.

Exercise

Compare the unit step response for motor speed obtained from MATLAB, LabVIEW and Simulink. Are they similar? Are they expected to be similar?

3.2. Impulse Response in MATALB

The impulse signal has a Laplace transform 1. Which means that it has all the frequencies in equal magnitude. The impulse response shows the behavior of a system in response to a combination of all the frequencies. In matlab we can find the impulse response with the function `impulse()`. The usage is similar to the `step()` function.

Exercise

Find the impulse response of motor speed, motor position and pendulum arm angle.

3.3. Response of arbitrary inputs in MATLAB

MATLAB can also be used to find the time response of continuous linear systems for any arbitrary input using the function `lsim().` The syntax of the command is

```
lsim(sys,u,t)
```

where

sys is the system model
u is the input vector
t is the time vector

for example if you want to find the response to a ramp signal and you have created a model for dc motor speed by the name dcmotor_speed_tf, then use the following code

```
t = 0:0.1:5;  % vector of time staring at 0 with steps of 0.1 seconds
               %to a final value of 5 seconds
u = t; % input is a ramp
lsim(dcmotor_speed_tf,u,t)
```

You can also generate vectors u and t by using the function `gensig()`. In the MATLAB command window type: `doc gensig`. Read the help of this function to figure out how you can generate u and t for a square wave and sine wave.

Exercise

Find the response of motor speed to a ramp, square wave and sine wave inputs in MATLAB.

3.4. Time response using inverse Laplace transforms

Till now we have seen ways of finding plots of the time response to various inputs. However, it is also useful to learn how to find the mathematical expressions of these responses. For this we will use the familiar tool of inverse Laplace transforms.

The function for inverse Laplace transform in MATLAB is ilaplace(). Unfortunately, this function doesn't take system models as inputs. It only takes symbolic functions of frequency variable s.

The use of ilaplace() will be shown with the help of an example. Let's say we want to find the step response of the transfer function

$$\frac{s+1}{s^2+5s+6}$$

Then we can use the following code

```
syms s t;
G = (s+1)/(s^2+5*s+6);
step_input = 1/s; % Laplace of step function
ilaplace(G*step_input) %inverse Laplace of output
```

Exercise

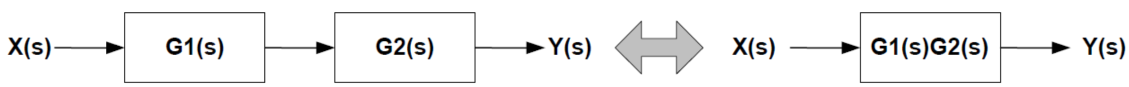
Find the mathematical expression for step, ramp and impulse for the motor speed transfer function. Compare the expression of step response with its plot obtained earlier.

4. Interconnection of systems in MATLAB

In Simulink and LabVIEW, systems are represented by a block. It is very easy to connect two systems in series or parallel in these software tools. However, in MATLAB it is not that straightforward.

In this section we will see how we can use MATLAB to interconnect systems in different configurations. This will be useful when we want feedback connections or want to connect a controller in series with the original system.

4.1. Series connection



Suppose we have two transfer functions in series as:

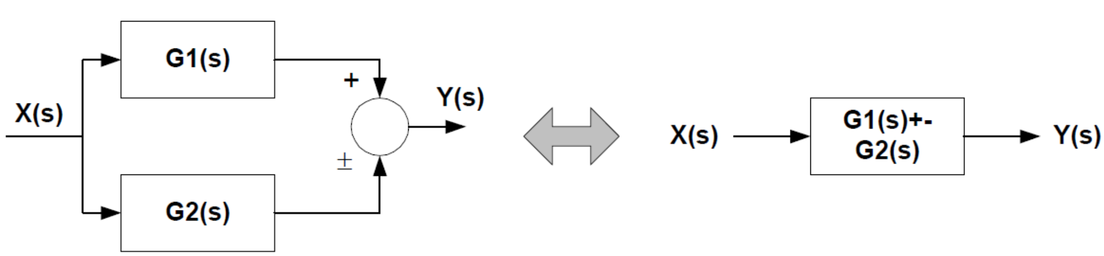
$$G1(s) = \frac{s+2}{s^2+5s+6}$$

$$G2(s) = \frac{s+4}{5s^2+3s+4}$$

The resultant transfer function can be evaluated using function series()

```
num1=[1 2];
den1=[1 5 6];
num2=[1 4];
den2=[5 3 4];
Gs1 = tf(num1,den1);
Gs2 = tf(num2,den2);
sys = series(Gs1,Gs2)
```

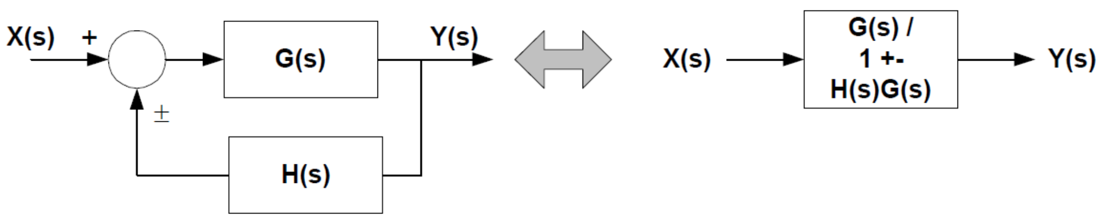
4.2. Parallel



If two transfer functions in are in parallel, the resultant transfer function can be evaluated using the function parallel()

```
num1=[1 2];
den1=[1 5 6];
num2=[1 4];
den2=[5 3 4];
Gs1 = tf(num1,den1);
Gs2 = tf(num2,den2);
sys = parallel(Gs1,Gs2)
sys = parallel(Gs1,-Gs2)
```

4.3. Feedback



If two transfer functions in are in connected in feedback, the resultant transfer function can be evaluated using the function feedback()

```
num1=[1 2];
den1=[1 5 6];
num2=[1 4];
den2=[5 3 4];
Gs = tf(num1,den1);
Hs = tf(num2,den2);
sys = feedback(Gs,Hs)  %negative feedback
sys = feedback(Gs,Hs,+1) %positive feedback
```

Exercise

Define a system which has a transfer function of just 5 i.e. a simple gain. Connect this system in negative feedback with the model of dc motor speed.