Overview:
This Experiment provides an introduction to Matlab and the LabVIEW Control Design and Simulation Module. A second order system is used to introduce the use of the software for analysis and simulation of a simple system.

\[ \text{Time response} = \text{Transient response} + \text{Steady state response} \]
Transient time response (Natural response) describes the behavior of the system in its first short time until it arrives the steady state value and this response will be our study focus. If the input is step function the output or the response is called step time response and if the input is ramp, the response is called ramp time response … etc.

\[ \text{Step time response specification :} \]
- \textit{Percent overshoot} \%OS: is the maximum fraction by which the response overshoots the steady state value expressed as a percentage. This characteristic is not found in a first order system and found in higher one for underdamped step response.
• **Settling time** $T_s$ : is the time required to fall within a certain percentage of the steady state value for a step input. For example the amount of time required for the step response to reach and stay within 2% of the steady state value OR in other words we can define it as the smallest amount of time required to reach the steady state value.

• **Peak time** $T_p$ : is the time required for the underdamped step response to reach the first maximum peak.

• **Rise time** $T_r$ : is the time required for the step response to go from 10% to 90% of the final value.

• **Steady state error** : is the difference between the input and the output of a system after the natural response has finished.

• **DC Gain** : The DC gain is the ratio of the steady state step response to the magnitude of a step input. For example if your input is step function with amplitude = 1 and found the step response output = 5 then the DC gain = 5/1 = 5. In other words it is the value of the transfer function when $s=0$.

**Second Order system :**

The general form of second order system is:

$$G(s) = \frac{a}{(s^2 + bs + c)} = \frac{K_{dc} \omega_n^2}{(s^2 + 2\zeta\omega_n s + \omega_n^2)}$$

Natural frequency $\omega_n$ is the frequency of oscillation of the system without damping.

Damping Ratio $\xi = \frac{b}{2\omega_n}$

Note that the system has a pair of complex conjugate poles at:

$$S = -\zeta \omega_n \pm j\omega_n \sqrt{1-\zeta^2} = -\sigma \pm j\omega$$

$\omega$ : damped frequency of oscillation

DC Gain is $\frac{a}{c} = K_{dc}$

Percent overshoot

$$OS\% = e^{-\zeta\pi} \times 100$$

Settling time

$$T_s = \frac{4}{\zeta\omega_n}$$

Peak time

$$T_p = \frac{\pi}{\omega_n \sqrt{1-\zeta^2}}$$

**Step time response:**

We know that the system can be represented by a transfer function which has poles (values make the denominator equal to zero), depending on these poles the step response divided into four cases:

1. **Underdamped response** ($0 < \zeta < 1$)

In this case the response has an overshooting with a small oscillation which results from complex poles in the transfer function of the system.
In this case, the system output for a unit-step input, can be written as:

\[ Y(s) = \frac{\omega_n^2}{S(S + \zeta \omega_n \pm j \omega_n \sqrt{1 - \zeta^2})} \]

\[ S = -\zeta \omega_n \pm j \omega_n \sqrt{1 - \zeta^2} = -\sigma \pm j \omega \]

2. **Critically response**: \((\zeta = 1)\)

In this case the response has no overshooting and reach the steady state value (final value) in the fastest time. In other words it is the fastest response without overshooting and is resulted from the existence of **real & repeated poles** in the transfer function of the system.

In this case, the system output for a unit-step input, can be written as:

\[ Y(s) = \frac{\omega_n^2}{S(S^2 + 2\zeta \omega_n S + \omega_n^2)} = \frac{\omega_n^2}{S(S + \omega_n)^2} \]

Note that the system has 2 repeated real poles at \(S = -\omega_n\)

3. **Overdamped response**: \((\zeta > 1)\)

In this case no overshooting will appear and reach the final value in a time larger than critically case. This response is resulted from the existence of **real & distinct poles** in the transfer function of the system.

In this case, the system output for a unit-step input, can be written as:

\[ Y(s) = \frac{\omega_n^2}{S(S^2 + 2\zeta \omega_n S + \omega_n^2)} = \frac{\omega_n^2}{S(S + \zeta \omega_n + \omega_n \sqrt{\zeta^2 - 1})} \]

Note that the system has 2 distinct real poles at:

\[ S = \zeta \omega_n \pm \omega_n \sqrt{\zeta^2 - 1} \]

4. **Undamped response**:

In this case a large oscillation will appear at the output and will not reach a final value and this because of the existence of **imaginary poles** in the transfer function of the system and the system in this case is called "Marginally stable".

\((\zeta = 0)\)

In this case, the system output for a unit-step input, can be written as:

\[ Y(s) = \frac{\omega_n^2}{S(S^2 + \omega_n^2)} = \frac{\omega_n^2}{S(S \pm j \omega_n)} \]

Note that the system has a pair of imaginary poles \(S = \pm j \omega\)
**Control System Analysis With Matlab commands:**

- **step**: This command is used to plot the step response of a system. For example we would to plot the step response of the following systems:

1. **Transfer function**:

   \[ H(s) = \frac{9}{s^2 + 2s + 9} \]

   ```matlab
   num=[9];
   den=[1 2 9];
   step(num,den)
   %Another way
   Sys=tf(num,den);
   step(Sys)
   ```

![Step Response](image)

2. **State space model**:

   \[
   \begin{bmatrix}
   x_1 \\
   x_2
   \end{bmatrix} = \begin{bmatrix}
   -0.5 & -0.7 \\
   0.5 & 0
   \end{bmatrix} \begin{bmatrix}
   x_1 \\
   x_2
   \end{bmatrix} + \begin{bmatrix}
   1 \\
   -1
   \end{bmatrix} u(t)
   \]

   \[
   y(t) = \begin{bmatrix}
   2 & 6.5
   \end{bmatrix} \begin{bmatrix}
   x_1 \\
   x_2
   \end{bmatrix} + [0] u(t)
   \]

   ```matlab
   a=[-0.5 -0.7 ; 0.5 0];
   b=[1 ; -1];
   c=[2 6.5];
   d=0;
   step(a,b,c,d)
   %Another way
   Sys=ss(a,b,c,d);
   step(Sys)
   ```
**Model Conversion in LabVIEW:**

The VIs found in the Model Conversion sub-palette are used to convert system models from one representation to another (i.e. from State Space representation to Transfer Function or Pole-Zero-Gain and vice versa).

To convert the State space model discussed previously into a Transfer Function Model, use a similar Block Diagram to the one shown below:
Alternatively, if you wish to convert the Transfer Function Model into a State Space Model, the following connection should be made:

![Diagram showing connection between Transfer Function Model and State Space Model](image)

**Control System Analysis With Graphical LabVIEW:**

1. Create a new VI.
2. Right-click on the block diagram and navigate to Control Design & Simulation » Control Design » Model Construction and drag the CD Construct Special TF Model VI onto the block diagram.

![Diagram showing block diagram with CD Construct Special TF Model VI](image)

1. See the figure below. Left-click on the polymorphic VI selector of the CD Construct Special TF Model VI in the block diagram and select 2nd Order Model. The second order model can be used to describe a mass-spring-damper system or an RLC circuit.

![Diagram showing polymorphic VI selector](image)

2. On the front panel, right-click to display the Controls palette. Navigate to Modern » Numeric » Horizontal Pointer Slide and drag a Horizontal Pointer Slide onto the front panel. While the name of the slide is highlighted, change it to Damping Ratio. Follow the same procedure to create another slide and make its name Natural Frequency.
3. On the front panel, click on the maximum number for the Damping Ratio slide and change it to 1. Change the maximum number for the Natural Frequency slide to 5.
4. On the block diagram, wire the Damping Ratio and the Natural Frequency slides to the Damping Ratio and Natural Frequency inputs respectively of the CD Construct Special TF VI.
5. Navigate to Control Design & Simulation »Control Design »Time Response and drag the CD Step Response VI onto the block diagram. The polymorphic instance of the CD Step Response VI will show SS for state space but this will change after the next step.

6. Wire the Transfer Function Model output of the CD Construct Special TF VI to the State-Space Model input of the CD Step Response VI. The polymorphic instance of the CD Step Response VI will automatically change to TF.

7. Right-click on the Step Response Graph output of the CD Step Response VI and select Create »Indicator.

8. Go to Programming »Structures »While Loop and place a While Loop on the diagram around the existing code.

9. Hover over the Loop Condition terminal at the bottom right of the loop, right-click, and Select Create Control to put a Stop button on the front panel for the VI.

10. The block diagram of your VI should look like this:

![Block Diagram](image1)

11. Run the VI after putting in nonzero values for Damping Ratio and Natural Frequency on the front panel. Adjust the parameters on the front panel and observe the system response. Your front panel should be similar to this:

![Front Panel](image2)

Additionally, statistics on the performance of the step response can be calculated. The implementation and results of the CD Parametric Time Response VI can be seen in the following two Figures.
**Dynamic System Simulation in LabVIEW:**

1. Go back to previous VI.
2. Right-click on the block diagram and navigate to Control Design & Simulation » Simulation »Control and drag a Control & Simulation Loop onto the block diagram beneath the While Loop. The Control & Simulation Loop can be set for precise timing like the Timed Loop in LabVIEW. However, the Control & Simulation Loop has built in ODE solver capability.
3. Navigate to Control Design & Simulation » Simulation » Continuous Linear Systems and drag a Transfer Function inside the Control & Simulation Loop.
4. Double-click on the Transfer Function to open the Transfer Function Configuration dialog box as shown below. Select the Terminal option for the Parameter source.

5. Wire the Transfer Function Model output of the Construct Special TF Model VI in the While Loop to the Transfer Function input of the Transfer Function block in the Control & Simulation Loop.
6. Select a Step Signal from the Control Design & Simulation » Simulation » Signal Generation subpalette and place it on the diagram. Double-click on the Step Signal block to open the configuration page and change the step time to be 0.
7. Wire the output of the Step Signal to the Input of the Transfer Function.
8. Select a Simulation Time Waveform from the Control Design & Simulation » Simulation » Graph Utilities subpalette and place it on the diagram.
9. Wire the output of the Transfer Function to the input of the Simulation Time Waveform.
11. Click OK to close the Configure Simulation Parameters dialog box.
12. The block diagram should now look like the one below. The resulting VI starts with a model creation and analysis/design loop and, when the stop button is pressed, moves into a simulation loop.

Control System Analysis With Simulink:

We can do the same work by simulink in Matlab:

- **Components**:
  For most of the systems we will encounter, we only need to be concerned with a small fraction of Simulink’s Component library. In particular, the components you should be familiar with are:

- "**Continuous**” library
  - Integrator—integrates a signal
  - State-Space—used to add a system block in state-space form
  - Transfer Fcn—used to add a system block in transfer function form
• "Math Operations" library
  Gain—a constant gain
  Sum—used to add two or more signals

• "Sinks" library
  Scope—used for viewing system output
  To workspace—used to transfer a signal to MATLAB

• "Sources" library
  Ramp—generates a ramp signal
  Sine Wave—generates a sinusoid
  Step—generates a unit step signal

1. Transfer Function (the above one)

   ![Diagram of transfer function with Step, Transfer Fcn, and Scope]

Notes:
- Double click on "Step" and change the step time parameter to zero.
- Double click on Transfer Fcn and enter the numerator & denominator of it.
- You can determine the period of simulation by changing the simulation stop time in the toolbar. Change it to 10 sec.
- After simulate the model, the result will appear in the "scope" and by double click on it and pressing "Auto scale" the response can be seen better.