Lab #7 Introduction to LabView

Introduction:
LabView is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabView uses dataflow programming, where the flow of data determines execution. In LabView, you build a user interface with a set of tools and objects. The user interface is known as the front panel. You then add code using graphical representations of functions to control the front panel objects. The block diagram contains this code. In some ways, the block diagram resembles a flowchart. You can purchase several add-on software toolsets for developing specialized applications. All the toolsets integrate seamlessly in LabView.
LabView empowers you to build your own solutions for scientific and engineering systems. LabView gives you the flexibility and performance of a powerful programming language without the associated difficulty and complexity. LabView gives thousands of successful users a faster way to program instrumentation, data acquisition, and control systems.

How Does LabVIEW Work?
LabView programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and multimeters. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.
A VI contains the following three components:

- **Front panel**: Serves as the user interface.
- **Block diagram**: Contains the graphical source code that defines the functionality of the VI.

Front Panel
The front panel is the user interface of the VI. You build the front panel with controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, pushbuttons, dials, and other input devices. Indicators are graphs, LEDs, and other displays. Controls simulate instrument input devices and supply data to the block diagram of the VI. Indicators simulate instrument output devices and display data the block diagram acquires or generates.
Block Diagram
After you build the front panel, you add code using graphical representations of functions to control the front panel objects. The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabView VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures.

Controls Palette
The Controls palette is available only on the front panel. The Controls palette contains the controls and indicators you use to create the front panel. Select View » Controls Palette or right-click the front panel workspace to display the Controls palette. You can place the Controls palette anywhere on the screen.
Functions Palette
The Functions palette is available only on the block diagram. The Functions palette contains the VIs and functions you use to build the block diagram. Select View» Functions Palette or right-click the block diagram workspace to display the Functions palette. You can place the Functions palette anywhere on the screen.

Examples:

1) Use LabView to represent the following transfer function; and then draw its step and impulse responses:

\[ G(s) = \frac{2s + 3}{s^2 + 3s - 2} \]
Output:

Control Panel:

Front Panel:
2) Use LabView to implement the following system:

\[
\begin{align*}
R(s) & \xrightarrow{\frac{s+2}{s^2+2s+3}} Y(s) \\
& \xrightarrow{\frac{s+3}{s^2-4s+1}}
\end{align*}
\]

Output:

Control Panel:

Front Panel:
3) Use LabView to implement the following system:

\[
\begin{align*}
R(s) & = \frac{s+2}{s^2+2s+3} \\
\frac{s+3}{s^2-4s+1} & \quad \text{Y(s)}
\end{align*}
\]

**Output:**

Control Panel:

Front Panel:
4) Use LabView to implement the following system:

\[
\begin{align*}
\text{Output:} \\
\text{Control Panel:} \\
\text{Front Panel:}
\end{align*}
\]
5) Use LabView to convert the following transfer function to state space representation; and then draw its bode plot (Magnitude and Phase).

\[ G(s) = \frac{s + 2}{s^2 + 2s + 3} \]

**Output:**

Control Panel:

![LabView Control Panel](image)

Front Panel:

![LabView Front Panel](image)
6) For the transfer function shown:

$$G(s) = e^{-A} \cdot \frac{s + 2}{s^2 + 2s + 3}$$

If $A$ is a delay with some value from 0 to 5, use LabView to show the effect of this delay on both step response and impulse response.

**Output:**

Control Panel:

![LabView Control Panel](image1)

Front Panel:

![LabView Front Panel](image2)
7) Use LabView to compare the value of two numbers (num1 and num2) and then show if num1 is greater, smaller or equal to num2?

**Output:**

Control Panel:

![LabView Block Diagram](image)

Front Panel:

![LabView Front Panel](image)
Exercises:

1) Create LabView program that takes a number representing Fahrenheit degrees and converts it to a number representing Celsius degrees. Use the following formula:

\[ C = (F - 32) \cdot \frac{5}{9} \]

Where C is the Celsius degree and F is the Fahrenheit degree.

2) For the following block diagram:

![Block Diagram]

a) Use MATLAB commands only (Don't use simulink) to find the transfer function of the block diagram.
   Hint: Use the commands feedback, series and parallel.

b) Draw both the step and impulse responses using MATLAB.

c) Now use LabView to find the transfer function of the block diagram.
   Hint: Use some blocks which are equivalent to the above MATLAB commands.

d) Draw both the step and impulse responses using LabView.

e) Draw bode plot (both magnitude and phase) of the system using LabView.