Labview Simulation

(Tom Co 1/21/2011)

I. Introduction

1. There are two methods for Labview simulation: the VI-approach and the Model Text-String -approach. The VI-approach is slightly more complicated to program but allows the user the modify parameters using controls such as sliders, while the Model-Text-String approach is simpler but the user can change the parameters by text input.

2. Consider the following second order system:

\[ \frac{d^2y}{dt^2} + 2\zeta \tau_n \frac{dy}{dt} + y = K_p \quad y(0) = y_0; \quad \frac{dy}{dt}(0) = v_0 \]

or in state space form as

\[ \begin{align*}
\frac{dx_1}{dt} &= x_2 \\
\frac{dx_2}{dt} &= \frac{K_p - 2\zeta \tau_n x_2 - x_1}{\tau_n^2} \\
x_1(0) &= y_0; \quad x_2(0) = v_0
\end{align*} \]

(1)

We will use the following nominal parameter values: \( y_0 = 1, v_0 = 0, K_p = 2, \zeta = 0.5 \) and \( \tau_n = 1.5 \).

3. For the VI-approach, two major steps are needed:
   a) Build the ODE sub-VI that contains the ODE equations.
   b) Build the Simulator VI that will use the ODE sub-VI.

For convenience, you can use the files: ODE_subvi.vi and simulator.vi that is available through the link:

www.chem.mtu.edu/~tbco/cm416/LabviewSimulator.zip

After downloading you can modify these file for your needs as shown in Figure 1 and 2.
Figure 1. The ODE subvi. To modify the number of states and/or parameters, drag and resize the block encircled in this figure.

dx1=x2;
dx2=(-2*zeta*tau*x2-x1+K)/(tau*tau);
Figure 2. The Simulator VI. To use a different model subvi, right-click on the circled block above then select browse...
Appendix A. How to Build the ODE_subvi.vi

There are 5 elements (3 input ports and 2 output ports) that need to be formed:

a) Error out (output)
b) Parameter array (input)
c) State array (input)
d) Time (input)
e) Derivatives array (output)

1. **Inserting Error-Out Control:**

   From the **Controls** palette, drag **Modern→Array, Matrix & Cluster→Error Out 3D.ctl** into the front panel.

   ![Error-Out Control](image)

   **Figure 1. Error-Out Control.**
2. **Building the blocks for Parameter Array Input**

We will assume three parameters: **a, b and c** strictly in strict sequence.

a) Drag the **Programming → Cluster & Variant → Variant-to-Data** icon and drop in the block diagram window.

![Figure 2. Variant-to-Data control.](image)

b) Hover over the **variant** in-port in the left side, right-click to show menu and then select **Create → Control**.

![Figure 3. Add a variant input.](image)
c) From Functions palette, insert **Programming→Cluster & Variant→Bundle** block then resize to allow 3 elements.

![Figure 4. Including the Bundle block.](image)

**Figure 4. Including the Bundle block.**

d) From the Functions palette, drag the **Programming→Numeric→Numeric Constant** icon close to the **bundle** block.

![Figure 5. Including a Numeric Constant block.](image)

**Figure 5. Including a Numeric Constant block.**
e) Change the type from integer to double: right click then select properties. Then go to the **Data Range** tab. Click on the “Long” icon to show menu, then select **Double**. This should change the color from blue to orange. After this connect it to the other two elements in the bundle block. Next, connect the **bundle** block output to the type input of the **Variant-to-Data** block, see Figure 6.

![Figure 6. Changing the data type of the Variant-to-data block.](image)

f) From the Function palette, drag **Programming→Cluster & Variant→Unbundle** block and connect with **Variant-to-data** block as shown in Figure 7.

![Figure 7. Unbundle block.](image)
3. **Prepare the State vector input.**
From the Functions palette, include Programming→Array→Index Array block. Then hover on the left side, right click then select Create→Control. Drag the block to make it have two indexed output: for \( x_1 \) and \( x_2 \).

Figure 9. Index array input.
4. Combine the states and parameters based on the differential equations. We will use the Formula block to code the derivative functions. Include a numeric control for the time variable (e.g. add input in the formula block then right-click and select create\rightarrow control.). Rearrange the input blocks in the vertical arrangement shown in Figure 10 (this is critical so that the order is correct when creating a subVI in the step 6.

Figure 10. Formula block to code the differential equation.

5. Form the Derivative array output block.
   a) From the Functions palette, include Programming\rightarrow Array\rightarrow Build Array block. Resize to make it accept two input points.
   b) Hover at the right side of the Build Array block, then right click and select create\rightarrow indicator. See Figure 10.
6. Select all the components then from menu in the Block Diagram window, select Edit ➔ Create SubVI.

![Diagram showing subVI creation process]

Figure 11. Creating the subVI.

7. Click on the subVI icon. then save this subvi. Edit the icon.
Appendix B. Building the Simulator

Figure 12. The Simulator VI.

A. Functions palette: Programming \rightarrow Cluster & Variant \rightarrow to Variant
B. Hover left side of [to Variant] block, right-click, then select create\rightarrow control
C. Functions palette: Programming \rightarrow Applications Control\rightarrow Static VI Reference
   (Right-click, then select: Strictly typed VI Reference ; a star should appear)
D. Functions palette: Programming \rightarrow Structures \rightarrow For Loop
E. (Right-click then select Disable Indexing)
F. Functions palette: Programming \rightarrow Cluster & Variant \rightarrow Bundle
G. Functions palette: Mathematics \rightarrow Differential Eq\rightarrow ODE Solve vi
   (Make sure the ODE:VI mode is selected)
H. Functions palette: Programming \rightarrow Array \rightarrow Transpose 2D Array
I. Functions palette: Programming \rightarrow Array \rightarrow Array subset