Tutorial for Simulation of a Batch Process

tbco 3/27/2016, 3/27/2017, 4/11/2021

Scenario:

Once two tanks have been filled to desired levels, they are drained simultaneously, and the process repeats for the next batch.

Part 1. Dynamic Simulation of the Level of Two Tanks

I. Obtain Dynamic Models

$$\frac{dh_1}{dt} = \frac{1}{A_1}\left(k_{in1}F_{in1,max} - k_{out1}C_{v1}h_1\right)$$

$$\frac{dh_2}{dt} = \frac{1}{A_2}\left(k_{in2}F_{in2,max} - k_{out2}C_{v2}h_2\right)$$

$$h_1(0) = h_{10} \quad ; \quad h_2(0) = h_{20}$$

II. Open a Simulink page

1. Open MATLAB, and type in command line: simulink . Alternatively, use the menu button and select "Simulink" (see Figure 1).

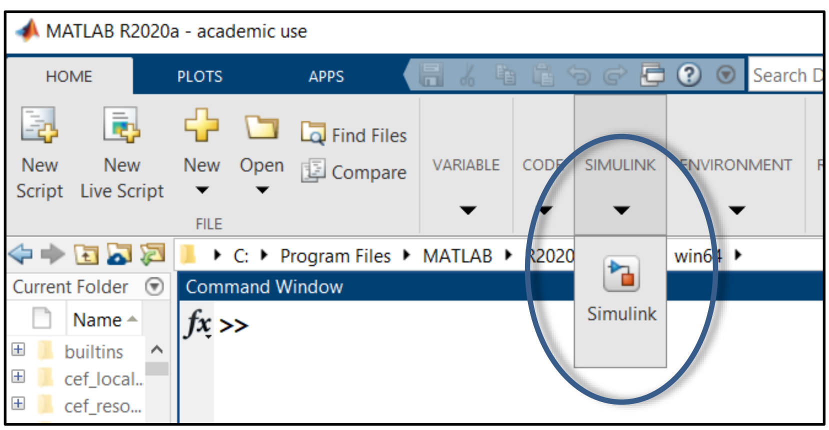

Figure 1. Starting Simulink.

2. Select "Blank Model" and a model window should pop out (see Figure 2)

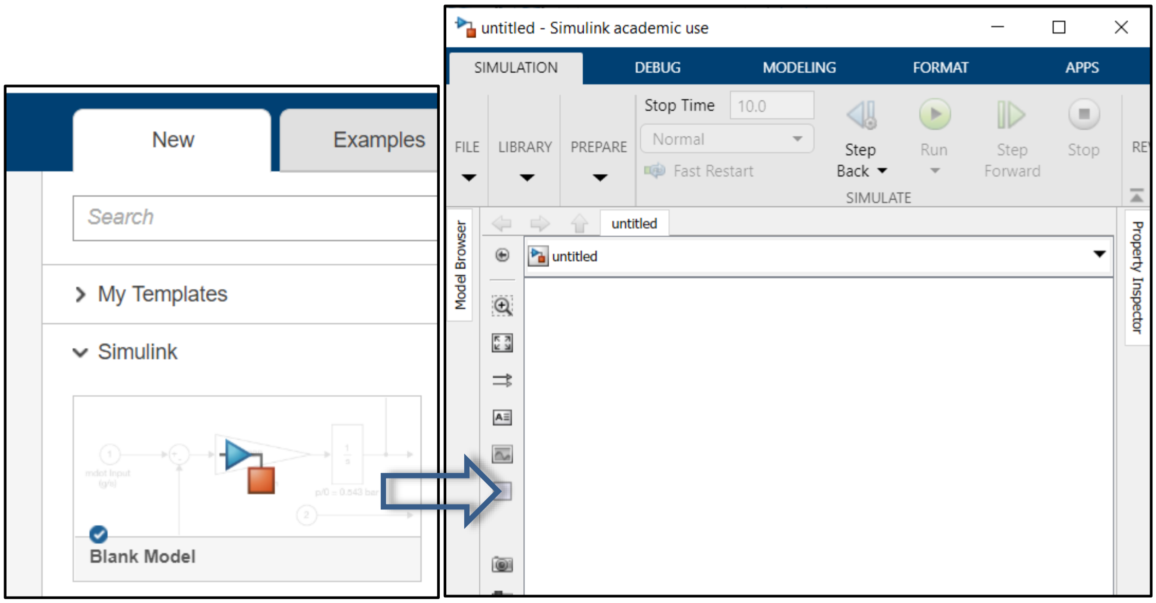

Figure 2. Creating a blank model.

III. Generate Model for Tank 1.

1. Include the blocks into the model window: (see Figure 3)

Method 1. Select menu item "Library Browser"→"Commonly Used Blocks", then drag-drop the block into the model window.

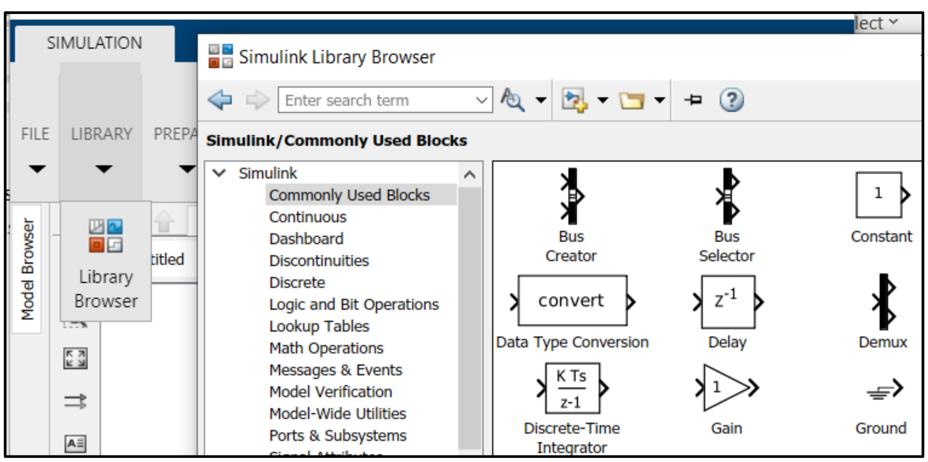

Figure 3. Using the library browser to select blocks.

Method 2. (left)-click on the mouse at any blank spot of the model window. Start typing the block names then click on the selection window that pops out.

Include the following blocks into the model window (see Figure 4).

a) Four “Constant” blocks
b) Two “Product” blocks
c) One “Sum” block
d) One “Gain” block
e) One “Integrator” block
f) One “Scope” block

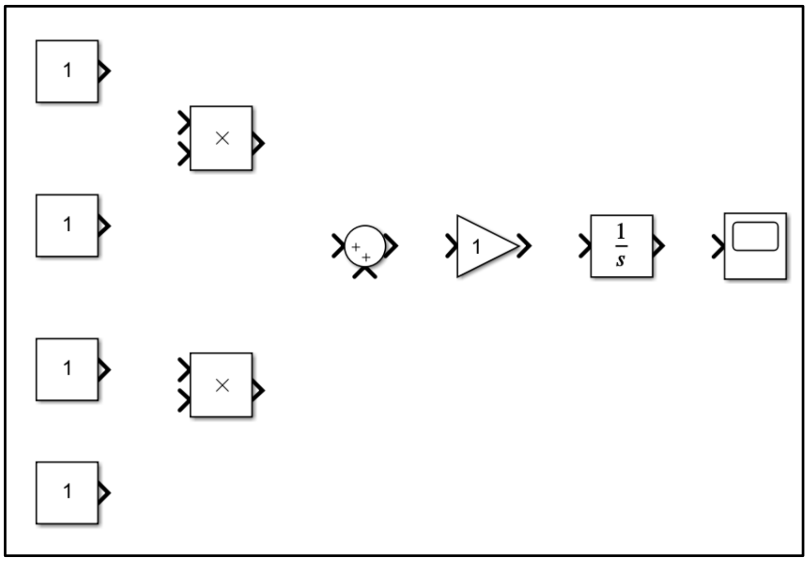

Figure 4. Including blocks into the Simulink model window.

2. Change the properties of the sum block to make it a difference block.
    a. Double-click on the sum block icon to get the properties window.
    b. Change the entry in the <List of Signs> to be “+-“.
    c. (Optional) You can change the shape to be “rectangular”.
    d. Remember to click “OK”. Otherwise the changes will not be applied.
    e. If necessary, you can resize the blocks.
3. Change the properties of the second product block to increase the number of inputs to 3.

4. Rename blocks (double-click on label below the blocks and then edit) then connect lines (click-drag from one port to the other port) among the blocks as shown in Figure 4. (Hint: for the line from “hout” to “Product”, click on the input of the “Product” then drag to the “hout” line.)

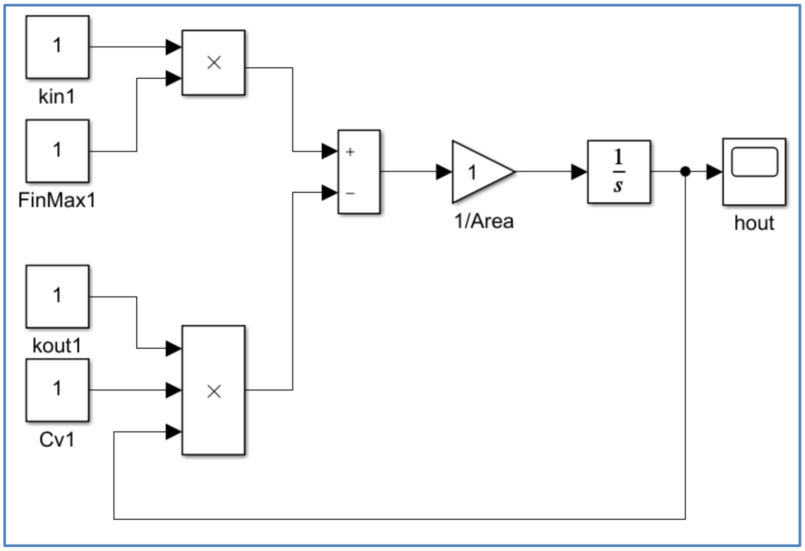

Figure 4. After renaming blocks and connecting the lines.

5. Set constant values: kin1=0.9, FinMax1=0.1, kout1=0.5 and Cv1=0.1. This is done by double-clicking on the the respective icons, then change the values in the "Constant value" entry in the properties window that pop-up. (Remember to click "OK" or hit [Enter] key to apply the changes.). Further, let the area for tank1 be $A_1 = 1.5$, then double-click on the gain and enter "1/1.5" (You can expand the "gain" icon if you wish to see the value/formula). (See Figure 5).

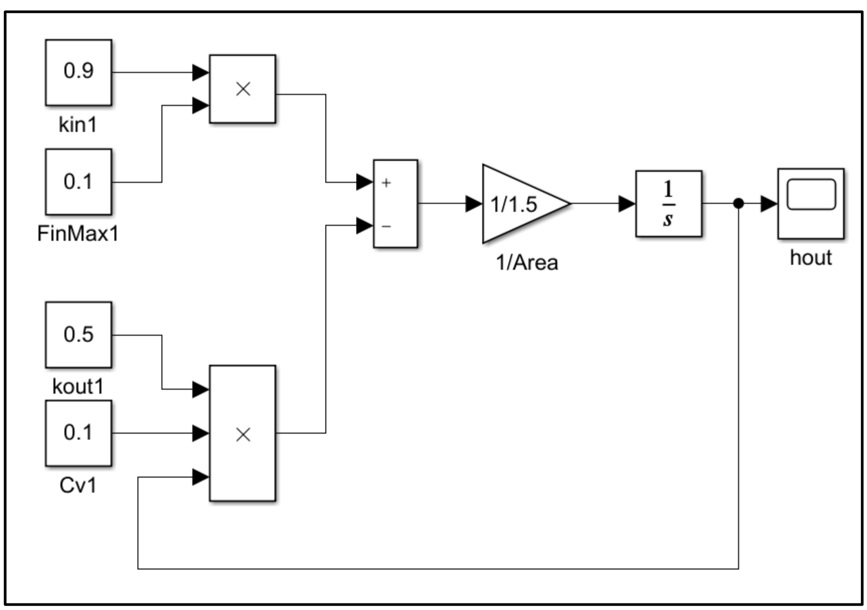

Figure 5. Renaming, revaluing and connecting the blocks.

6. Simulate the process for tank 1.
   a. Modify the simulation parameters
      (i) Key-in [Ctrl-E] to access the simulation configuration window
      (ii) Change the “Stop time” entry to 150.
      (iii) (Optional but recommended) Change the “relative tolerance” entry to 1e-6. (see Figure 6.)

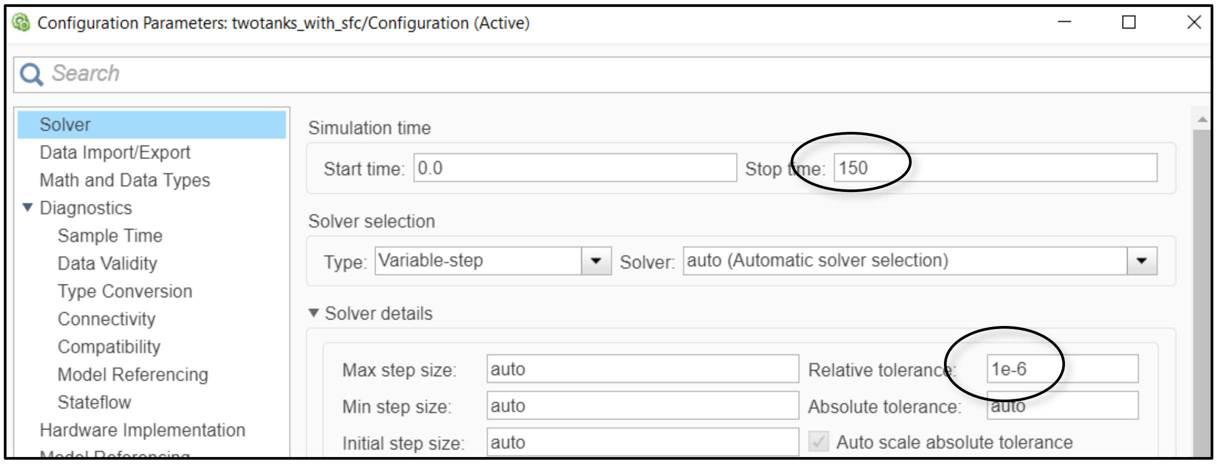

Figure 6. Changing simulation configuration.

   b. Run the simulation.

   Click on the “Play (Run)” button. See Figure 7. (Note the first time you click this button, the program will be compiling the model which will take some time, so just wait unit you get back the “Ready” prompt at the bottom of the Simulink model window before proceeding to next step.

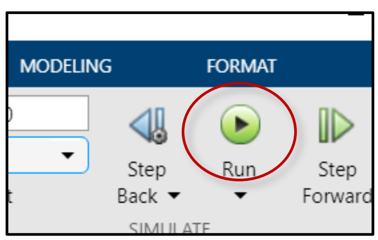

Figure 7. Click on the “Run” button to start simulation.

   c. Display the results.

   Double click on the “Scope” icon (or the one renamed as “hout”). The graph window should pop-out. (See Figure 8).

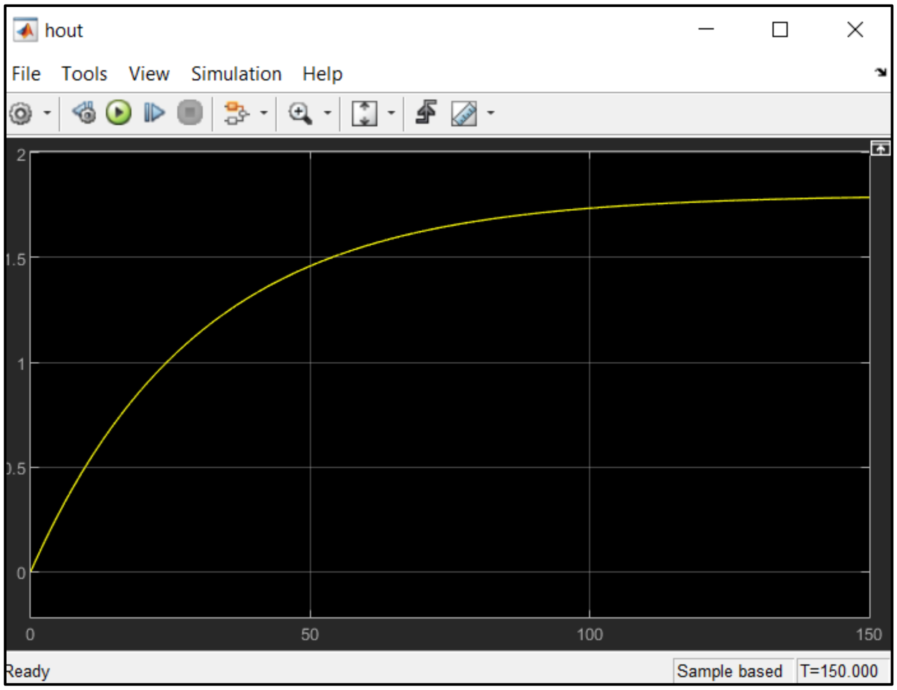

Figure 8. The response as shown in the scope.

IV. Creating subsystems

1. Group tank1 block to be a subsystem.

a) Select all the components in the Simulink model window, except the four “constant” blocks and the “hout” block.
b) Right-click the selection then key-in [Ctrl-G] to shrink the group into a subsystem. (You can resize the box).
c) Rename the subsystem as “Tank 1”. Then double click on the Tank 1 subsystem block to zoom-in (open) the details of the subsystem.
d) Relabel the input ports In1, In2, In3 and In4 to be kin1, FinMax1, kvout1 and Cv1, respectively. Also, rename the outport Out1 to be h1. See Figure 9.
e) Click [Up-arrow] to return to the parent level of the model. See Figure 9.

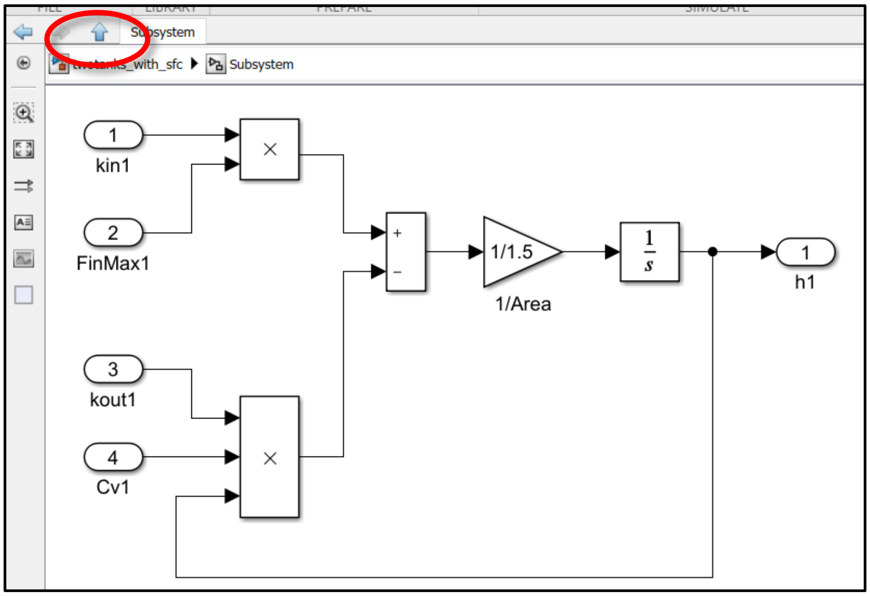

Figure 9. After renaming inports and outports.
The [Up-arrow] will return to parent model.

2. Generate a copy of the subsystem for tank 2.

a) On the main level of the model, select all blocks except the “scope” block. Then right-click-select <copy> and right-click-select <paste> (or the usual CTRL-c followed by CTRl-v sequence.)

b) Click on the new subsystem and relabel the blocks, e.g. kin1 to kin2, etc., as well with area of tank 2 with $A_2 = 1.2$. After returning to the main level, modify the values of the constant blocks so that kin2=0.8, FinMax2=0.2, kvout2=0.4, and Cv2=0.15 (see Figure 10).

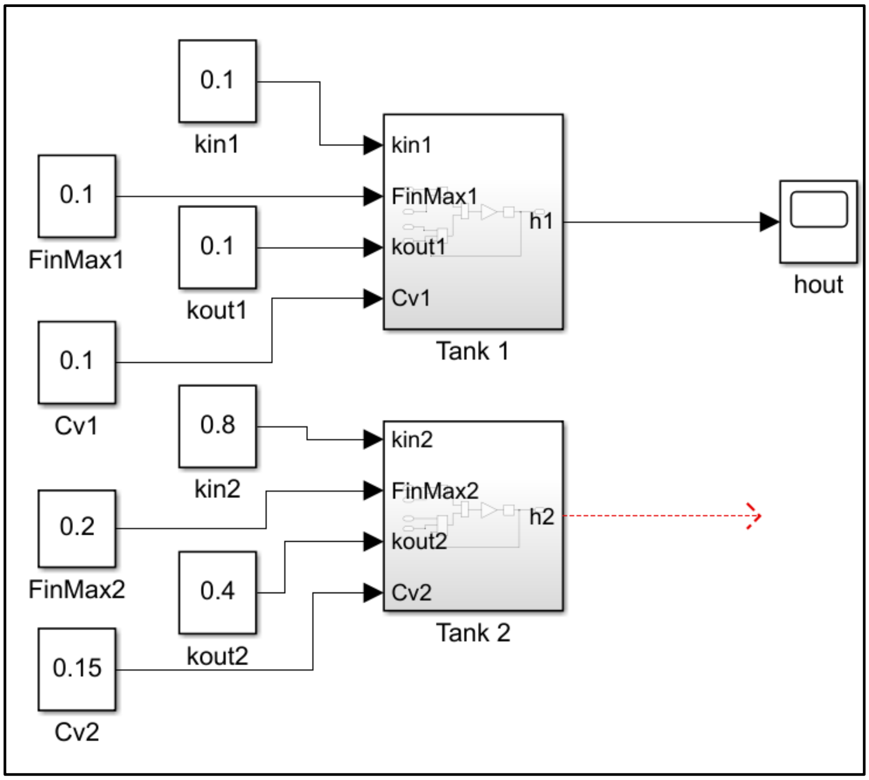

Figure 10. Modifying tank 2 labels and values.

V. **<u>Merging the outputs h1 and h2 from the subsystems for graphing.</u>**

1. First, remove the signal line from h1 to the “scope” block. Next, include an “mux” (multiplexer) block into the model window.
2. Connect both h1 and h2 lines to the input ports of the mux block.
3. Direct the output to the scope block. Figure 11.

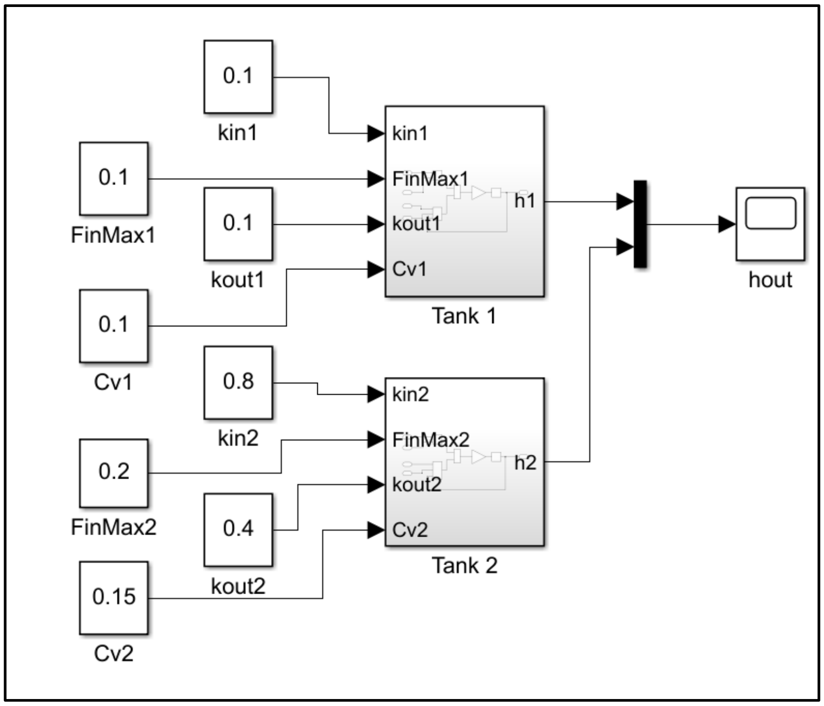

Figure 11. Merging h1 and h2 to a common plot.

VI. Run the simulation.

1. Save the model. For our purposes we will assume the name to be "twotanks_with_sfc_tutorial" (saved as *.slx file)
2. Run the simulation. You should get the response shown in Figure 12. Note: in the scope window, you can select the [View]→[Legend] window to identify which plot refers to which tank.

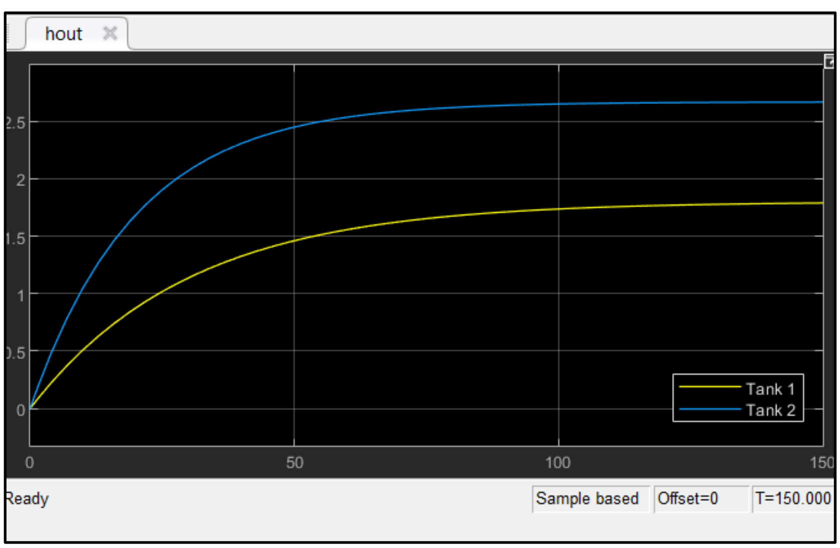

Figure 12. Simulation output for the two tank system.

Part 2. Converting SFC into Stateflow block

Assume that we have the sequential function chart shown in Figure 13 which is based on the following recipe:

1. Initialize by transferring recipe parameters and empty both tanks.
2. After both tanks are empty, start filling both tanks simultaneously.
3. When either tank reaches its maximum level, put it on “standby” mode.
4. After both tanks are in the “standby” modes, drain both tanks simultaneously.
5. When either tank reaches its minimum level, put it on “stop drain” mode.
6. After both tanks are in the “stop drain” mode, repeat from step 2.

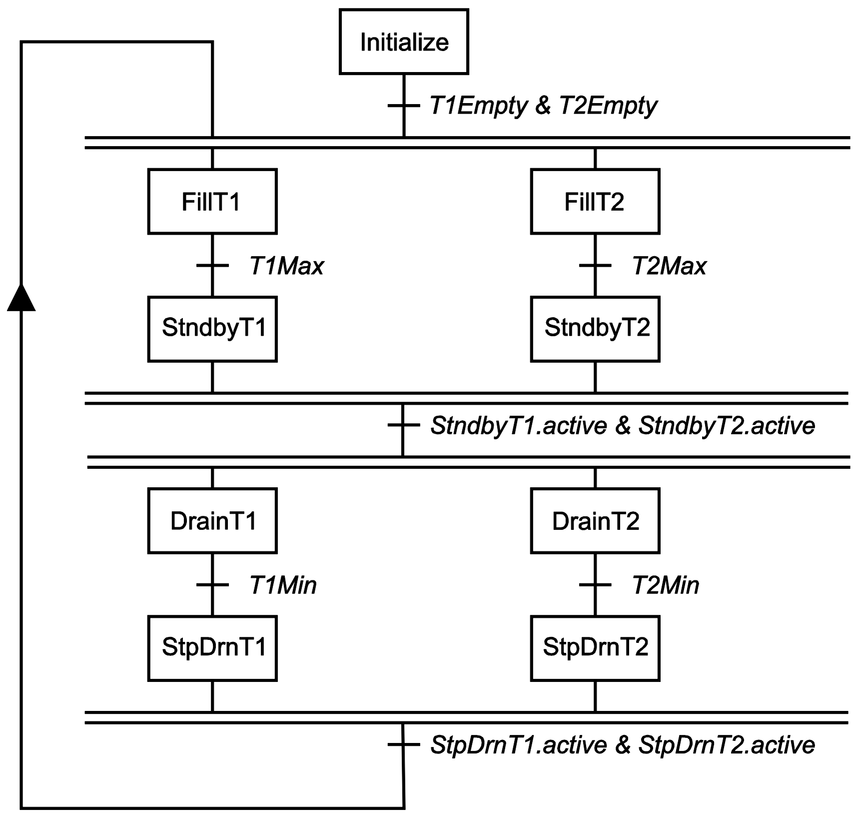

Figure 13. Sequential function chart for two-tank filling unit.

I. Prepare a Stateflow chart.

1. Include a Stateflow chart.

Method 1: From [Library Browser]→[Stateflow] menu item, drag-drop a "Chart" block
Method 2: Left-click anywhere in a blank spot in the model window, then type "chart".

2. Enlarge the block as shown in Figure 14.

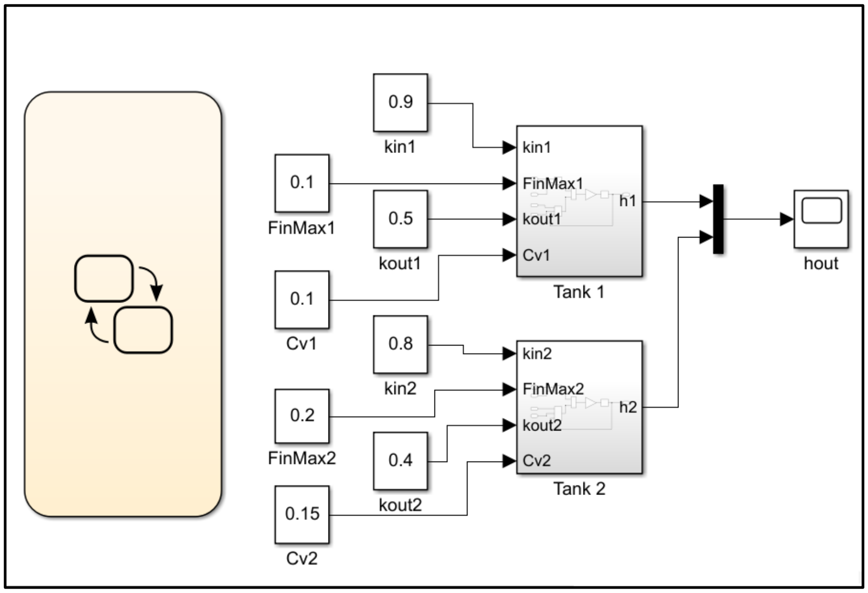

Figure 14. Including a Stateflow chart into the model window.

3. Open the chart editor.

Double-click on the "chart" block or subsystem to pop-up the Stateflow editor as shown in Figure 15.

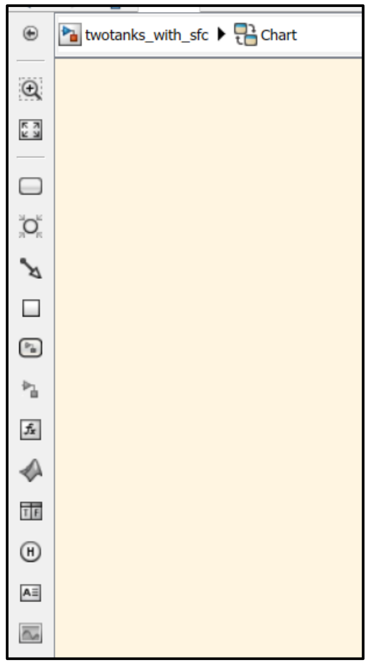

Figure 15. The Stateflow chart editor.

4. Include a set of input and output ports.

Method 1:

a) [Right-click] to generate the menu set shown in Figure 16, then select **[Add Input & Output]→[Data Input…]** to generate another window as shown in Figure 17.

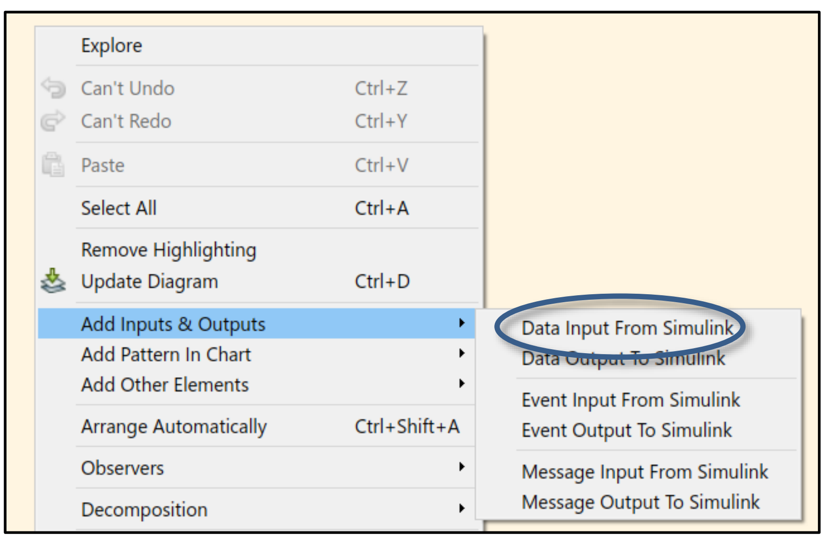

Figure 16. Right-click to generate the menu.

b) Modify the following entries of “Name”, “Scope”, “Port”, “Complexity” and “Type” to those shown in Figure 17.

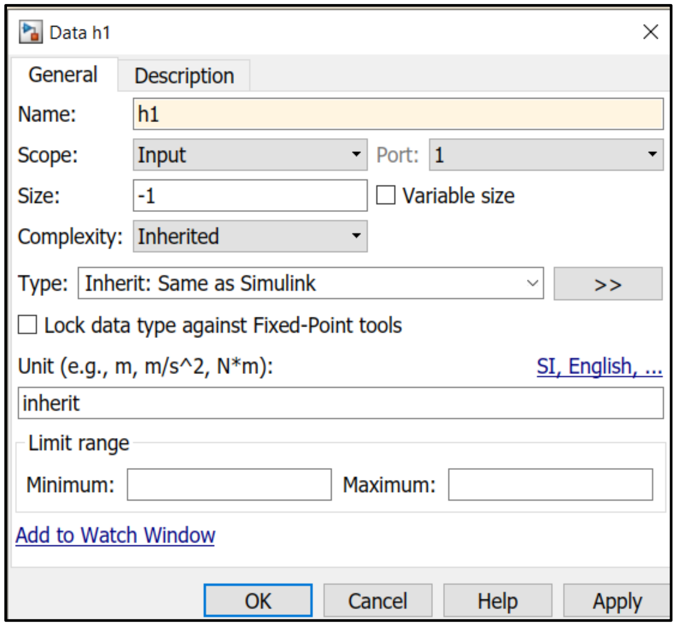

Figure 17. Modify entries for Stateflow data.

Method 2: **[Right-click]→[Explore]→[Add]→[Data].** Then modify the table entries appearing in the right portion of the “explore” window to entries shown in Figure 17.

Repeat the procedure for one more data input and four data output and change to the following:

Name: h2	Scope: Input	Port: 2	Complexity: Inherited	Type: Inherited
Name: kin1	Scope: Output	Port: 1	Complexity: Inherited	Type: Double
Name: kout1	Scope: Output	Port: 2	Complexity: Inherited	Type: Double
Name: kin2	Scope: Output	Port: 3	Complexity: Inherited	Type: Double
Name: kout2	Scope: Output	Port: 4	Complexity: Inherited	Type: Double

Alternatively: [Right-click] then select [explore]. Then one can adjust, add or delete data.

5. Include an event port.

We want the ability to trigger (or wake-up) the chart when events occur from outside the chart. In our example, we will use a square wave to be our clock which will wake-up our Stateflow chart when the wave jumps up or down.

Method 1:

a) **[Right-click]→ [Add Inputs & Outputs]→[Event Input from Simulink]** to pop-up the window to define an event port.

b) Change the entry for the "Name" to "clock", plus change the selection of "Trigger" to be "Either" then click [Ok] as shown in Figure 18.

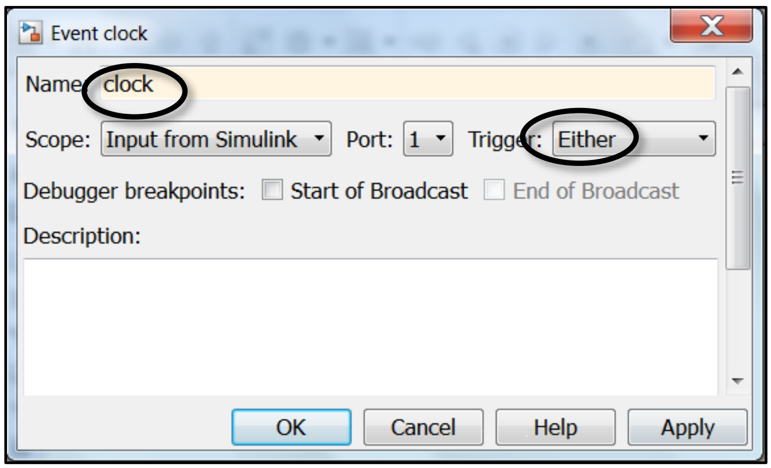

Figure 18. Modifying an event property.

Method 2: **[Right-Click]→[Explore]→[Add]→[Event]**, then modify entries to match those in Figure 18.

6. Link Stateflow with Simulink.

a) Go to the parent level.

b) Delete the constants blocks named kin1, kout1, kin2 and kout2

c) Connect the input lines as well as the output lines (including those of h1 and h2) as shown in Figure 19.

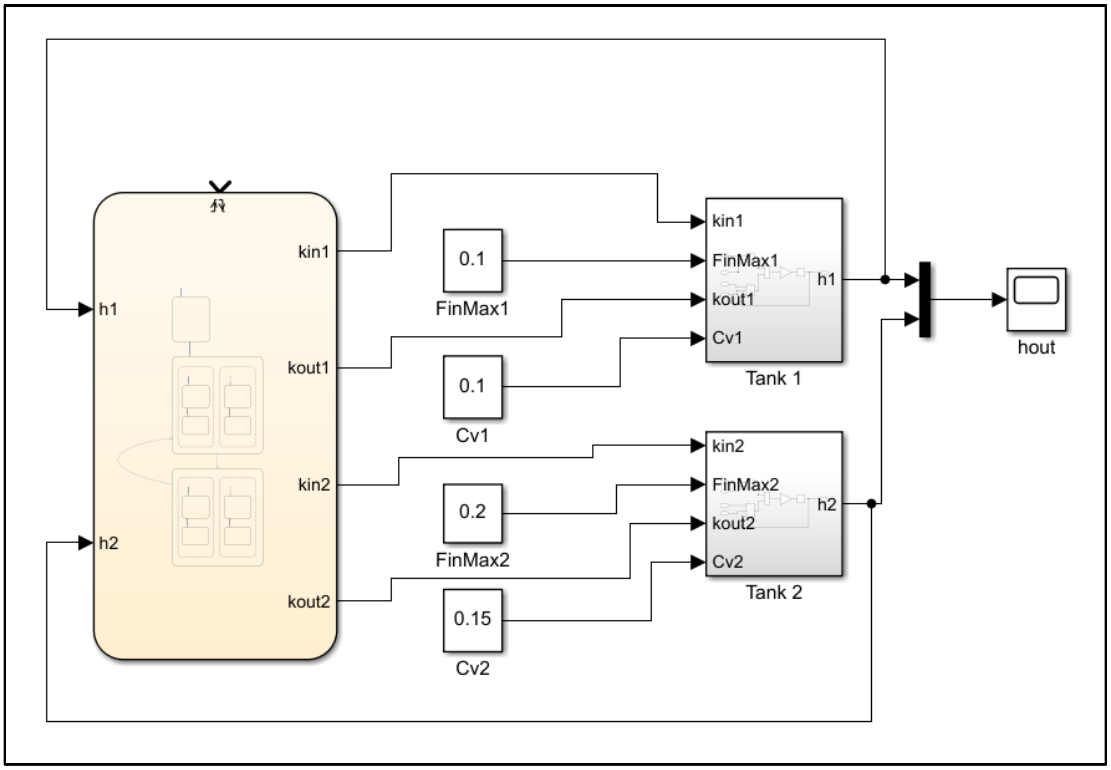

Figure 19. Linking chart input and output ports to other Simulink ports.

7. **<u>Add a square signal block to act as clock mechanism.</u>**

a) Include a "Signal Generator" block (either from [library browser]→ [Source]→[Signal Generator] or type "Signal Generator".) Optional: rename the label as "Clock". Then link the block to the event input as shown in Figure 20.

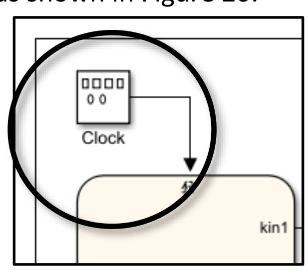

Figure 20. Linking Signal Generator (clock) to event port.

b) Double-click on the "Clock" icon. A properties window should pop-up. Choose [square] under "Wave-form" and also input "500" for frequency (see Figure 21.) then click [OK].

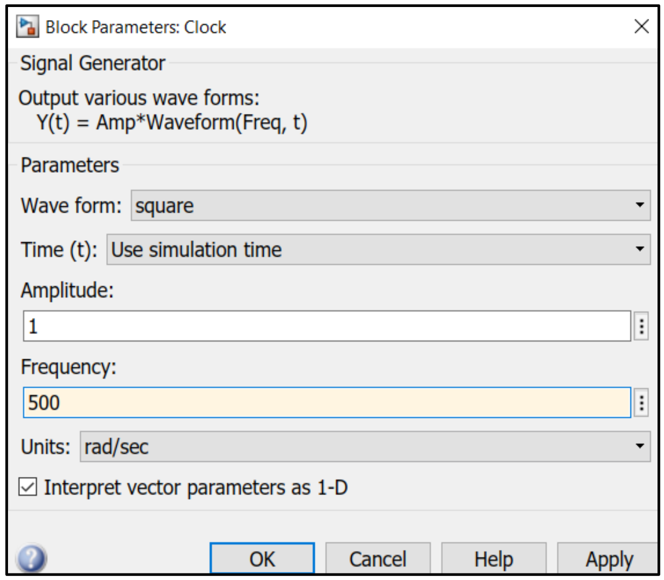

Figure 21. Change the parameters of signal generator (renamed as "Clock" here).

8. <u>Setup local variables.</u>

We will include four local variables, namely hmin1, hmax1, hmin2 and hmax2, which are the minimum and maximum set levels for each tank.

Method 1: **[Right-click]→[Add other elements]→[Local data]**
Method 2: **[Right-click]→[Explore]→[Data]**

Modify entries to match shown below:

Name: hmin1	Scope: Local	Complexity: Inherited	Type: Double
Name: hmax1	Scope: Local	Complexity: Inherited	Type: Double
Name: hmin2	Scope: Local	Complexity: Inherited	Type: Double
Name: hmax2	Scope: Local	Complexity: Inherited	Type: Double

II. Program the SFC into Stateflow Chart. (See Figure 13 for reference)

1. Create the initialization block.

a) Open the Chart editor (double-click on the chart icon) and drag-drop a **[State]** block (rounded box icon) from the selection list on the left edge of the editor) into the chart to contain the initialization actions (see Figure 22).

b) Rename the state as "Initialize"

c) Resize the block and enter the action lines shown in Figure 22.

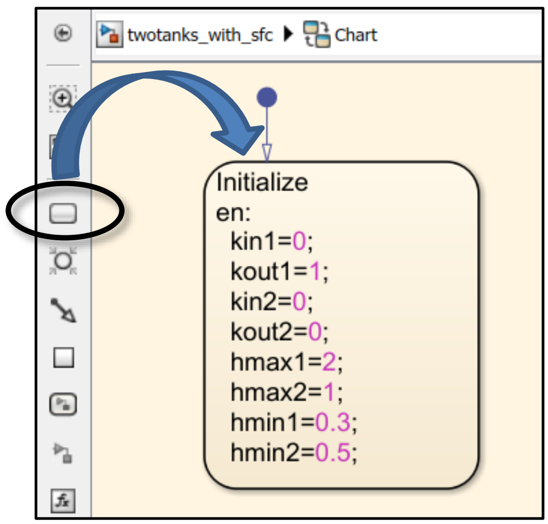

Figure 22. Include a state block into the chart.

Notes:

a) The line "en:" is a prefix denoting the lines and commands that follow to the right or below it are the actions "upon entry", equivalently "upon activation", of the state.
b) Each command must be ended by a semicolon.
c) The arrow-with-a-dot object in the top side is the "default transition". It indicate that this will be the first state that will be processed.
d) The values chosen for kin1, kout1, kin2 and kout2 will close the inlet valves while opening the outlet valves of both tanks to empty both tanks.
e) The formulas for hmax1, hmax2, hmin1 and hmin2 sets up the values for set maximum and set minimum levels of the tanks.

2. Create a block to contain the synchronous filling of the tanks.

a) Drag-drop another [State] block and rename it “FillBoth”.

b) Connect a transition line from “Initialize” block to “FillBoth” block

- Hover the cursor near the bottom edge of the “Initialize” block until the cursor changes to a “cross” symbol.
- Click-drag a transition line to the top edge of the “FillBoth” block.
- When the transition is selected start typing a condition:

[(h1<=0) && (h2<=0)]

(see Figure 23.)

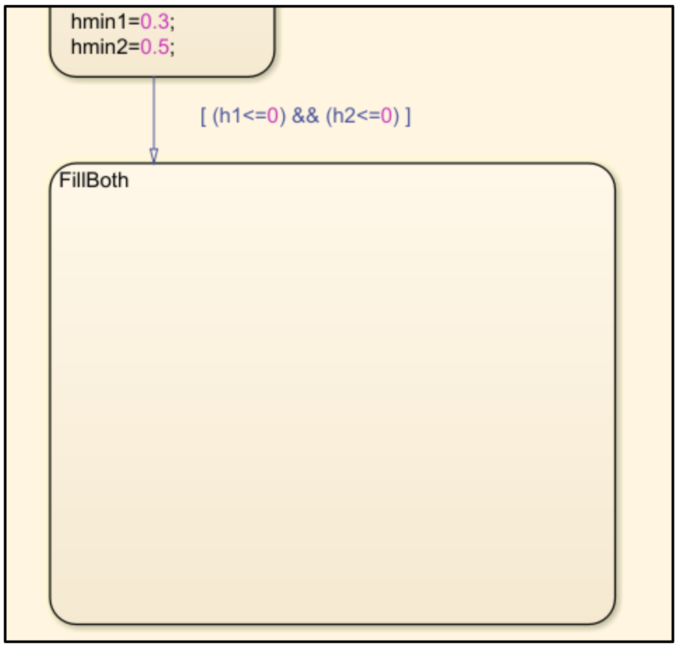

Figure 23. Including a transition line with conditions from “Initialize” to “FillBoth”.

(Note: If a “?” appears instead of the blinking cursor, just click on the “?” to go to input mode of a blinking cursor)

Remark: We suggest running the simulation after inclusion of a transition in order to test for errors.

c) Inside the “FillBoth” block, select **[Right-click]→[Decomposition]→[Parallel]** item

d) Next, drag-drop another state block inside “FillBoth” and name it “Tank1”. (Notice that the box has a dashed-line border to signify that it is one of a set of parallel synchronous event.)

e) Drag-drop another state block inside “Tank1” and name it “Fill”. (Notice that it will again have an arrow-with-circle attached to it signifying it is the first block that will be activated inside “Tank1”).

f) Enter action lines inside the “Fill” block to match what is shown in Figure 24.

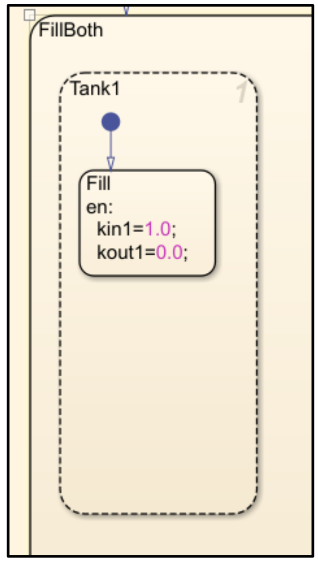

Figure 24. Filling of Tank1.

g) Include another state block below the “Fill” block and name it as “StopFill”.

h) Include some command lines as shown in Figure 24. (This closes the inlet valve to tank 1.)

i) Connect a transition line from “Fill” to “StopFill” with the following condtion: **[(h1>=hmax1)]** (See Figure 25).

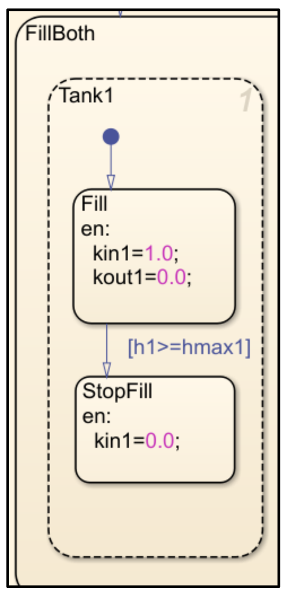

Figure 25. Connecting a “StopFill” block to “Fill” block.

Suggestion: Run the simulation to test the Stateflow chart but extend the run to time=100.

j) Select “Tank 1” block including its contents, then copy-and-paste inside “FillBoth”. Change the various names and variables to process tank 2 instead of tank 1. (see Figure 26).

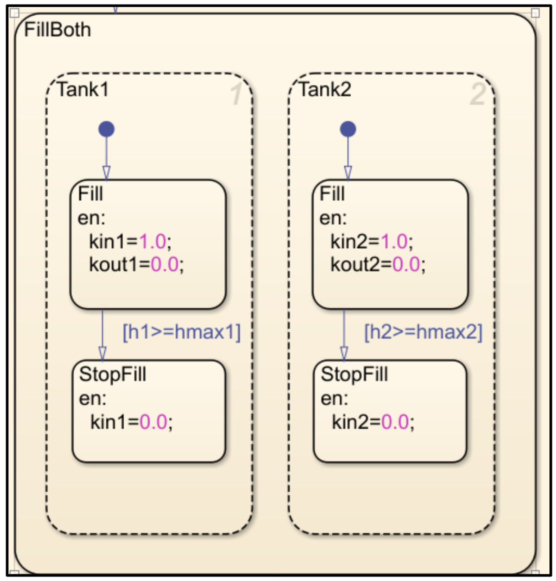

Figure 26. Copy “Tank1” to make it “Tank2”.

3. Create a block to contain the synchronous draining of the tanks.

a) Drag-drop another [State] block and rename it “DrainBoth”.
b) Connect a transition line from “FillBoth” block to “DrainBoth” block with the condtion:

[in(FillBoth.Tank1.StopFill) && in(FillBoth.Tank2.StopFill)]

(see Figure 27.)

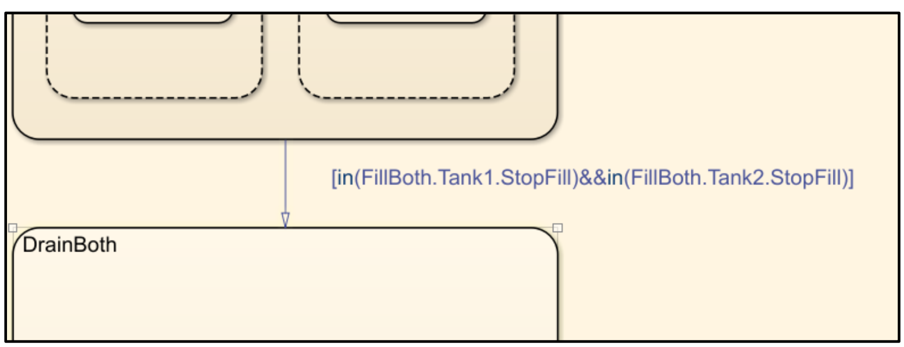

Figure 27. Transition from “FillBoth” to “DrainBoth”.

c) Inside “DrainBoth” block, select **[Right-click]→[Decomposition]→[Parallel (And)]**.
d) Copy the contents inside “FillBoth” and paste them into “DrainBoth”
e) Modify the names, command lines and conditions to process the draining of both tanks. (see Figure 28).

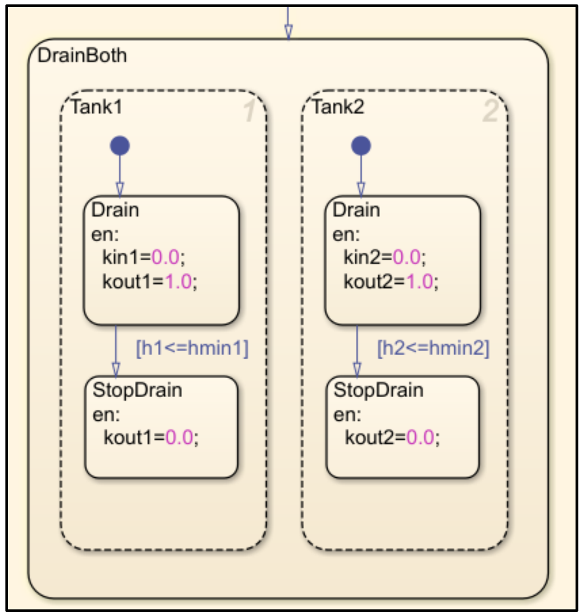

Figure 28. Draining of both tanks.

4. **<u>Creating a repeating loop.</u>**

a) Connect a transition line from "DrainBoth" to "FillBoth" with the following condition:

[in(DrainBoth.Tank1.StopDrain)&&in(DrainBoth.Tank2.StopDrain)]

Note: If the conditions are too long it can be split up to several lines using ellipsis ("…")

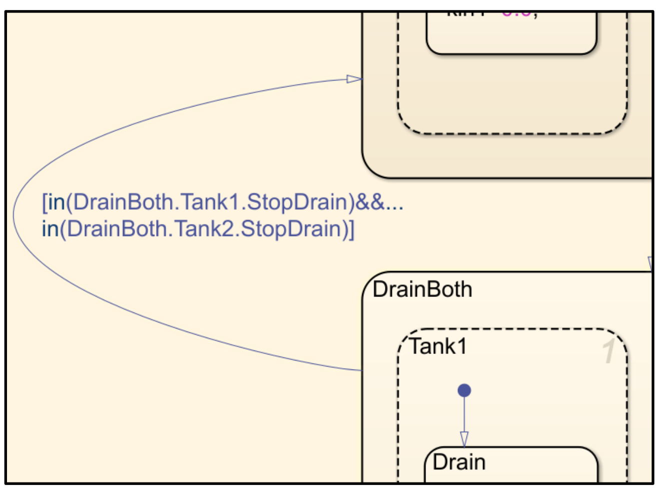

Figure 29. Including a loop transition.

b) Run the simulation and check the outcome via the scope in the main level.

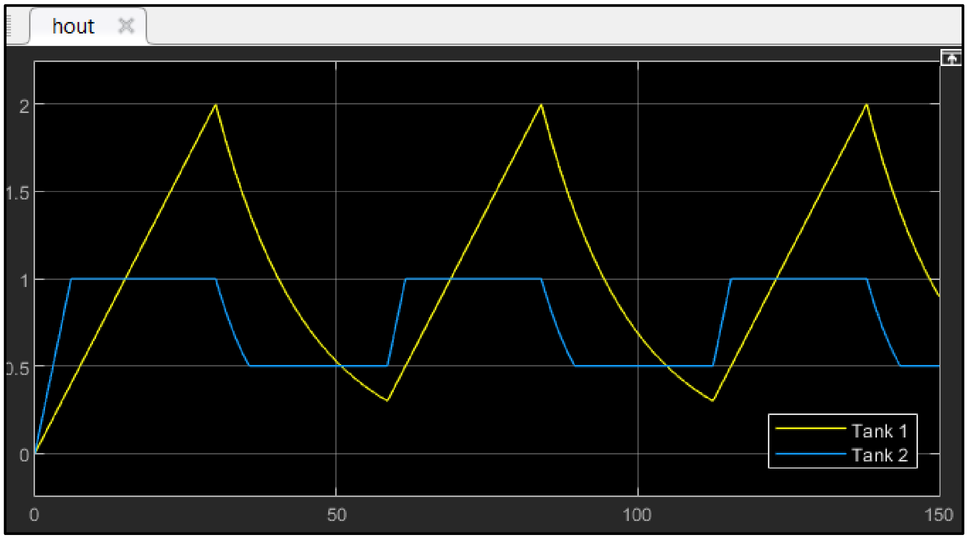

Figure 30. Plot showing the tank heights based on SFC recipe.

Brief Summary of the Stateflow program steps used:

1. Define/add input ports, output ports, an event port and local data variables (and other variable definitions).
2. Link the ports to other Simulink elements including blocks that can act as event generators.
3. Using a recipe (such as those constructed as sequential function charts (SFC)) include the different states starting with the exclusive-OR blocks (blocks with solid-line boundaries). Consider the block containing parallel states as one of these blocks. (When building the parallel states make you can change the [Decompostion] type to [AND] mode.)
4. Include the transition arrows and attach them with the appropriate condition labels.
5. Add a default transition arrow inside each parallel/AND block (blocks with dashed-line boundaries) to point at the block that Stateflow will activate upon its entry into that parallel/AND block.

III. Exporting Results

In some cases, where the data need to be collected for further analysis or reporting, one could add another block in the Simulink page to send the data to Matlab as an array. To do so, drag-drop a **[to Workspace]** block. Then tap the signal line that goes to the scope (see Figure 31).

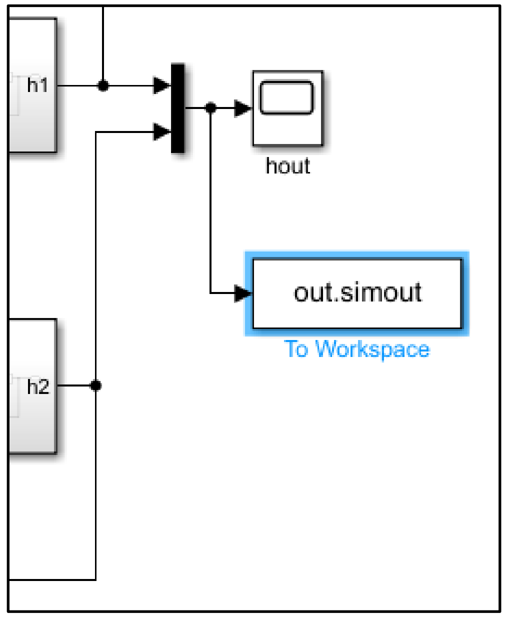

Figure 31. Adding a [simout] block.

Double-click on [To Workspace] block to pop-up a properties window (see Figure 32) and change the entries accordingly.

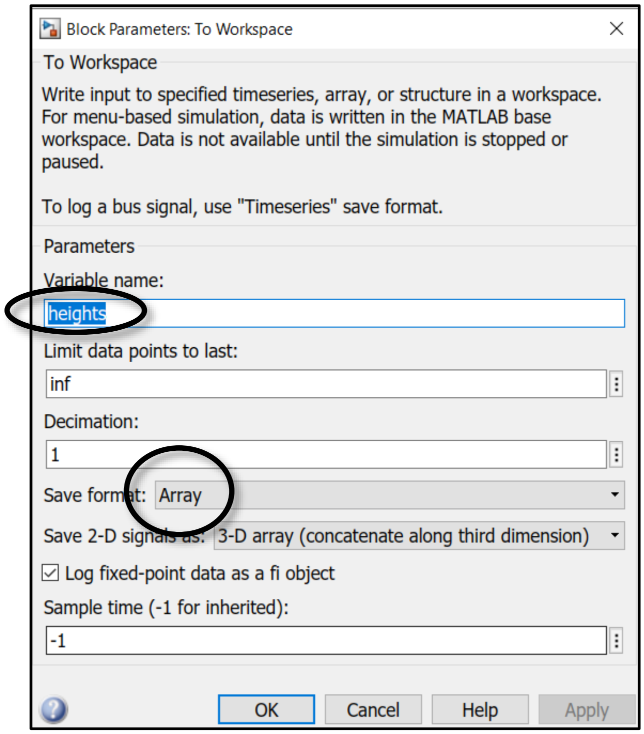

Figure 32. The property window of a [To Workspace] block.

After a simulation, data will be available in the main Matlab workspace. We could extract the data into one large array using the following command:

>> twotankheights = [out.tout, out.heights];

This array will contain three columns in which the first column is the time data, the second and third columns are the heights of tank1 and tank2, respectively. One could then export this array as a text file using the following command:

>> save('two_tanks_output.txt','-ascii','-double','twotankheights')

The text file **'twotanks_output.txt'** can then be imported by other programs such as Microsoft Excel.