To design a controller that switches the inflow between the two tanks to maintain the water volumes above the desired levels, we can use a feedback control approach. We can measure the water levels in both tanks and use this information to determine which tank to fill and for how long.

Let e! denote the error in the water level in Tank i, defined as e!=x!−r!, where r! is the desired water level in Tank i. Then, the control algorithm can be designed as follows:

Measure e1 and e2, the errors in the water levels in Tank 1 and Tank 2, respectively.

If both e1 and e2 are positive (i.e., both tanks are below the desired water levels), then fill the tank with the smaller error. Specifically, set the inflow to Tank 1 if e1<e2, and to Tank 2 otherwise.

If both e1 and e2 are negative (i.e., both tanks are above the desired water levels), then drain the tank with the larger error. Specifically, set the inflow to the tank with the larger error, with the flow rate equal to the difference between the outflow rate and the inflow rate required to bring the water level to the desired level.

If only one of e1 and e2 is positive, then fill the tank with the positive error.

Note that this control algorithm assumes that the inflow rate w is greater than the sum of the outflow rates v1 and v2. If this is not the case, the control algorithm needs to be modified to prevent overflowing.

This control algorithm can be implemented using a microcontroller or a programmable logic controller (PLC). The water levels in the tanks can be measured using level sensors or other suitable sensors. The inflow to each tank can be controlled using solenoid valves or other suitable flow control devices.

Overall, the control algorithm can help maintain the water volumes above the desired levels in the two tanks, even in the presence of leaks and varying inflow rates.

Open Ptolemy II and create a new workspace.

In the palette, navigate to the "DE" (Discrete Event) domain and drag an instance of the "HybridAutomaton" class onto the workspace.

Double-click on the "HybridAutomaton" block to open its properties.

In the "Variables" tab, add two variables: "x1" and "x2", with initial values of 0 and 1, respectively.

In the "Constants" tab, add four constants: "r1", "r2", "v1", and "v2", with values of 0, 0, 0.5, and 0.5, respectively.

In the "Inputs" tab, add one input port: "w", with a default value of 0.75.

In the "Modes" tab, add two modes: "Tank 1" and "Tank 2". In "Tank 1" mode, the dynamics are given by "der(x1) = w - v1" and "der(x2) = -v2". In "Tank 2" mode, the dynamics are given by "der(x1) = -v1" and "der(x2) = w - v2".

In the "Transitions" tab, add two transitions: "T1" and "T2". In "T1", the source mode is "Tank 1", the destination mode is "Tank 2", and the guard is "x1 >= r1". In "T2", the source mode is "Tank 2", the destination mode is "Tank 1", and the guard is "x2 >= r2".

Connect the "HybridAutomaton" block to a "DE Plotter" block using two "DE Multiport Switch" blocks.

Connect the "w" input port to a "DE Ramp" block with a slope of 0.

Set the initial state of the "HybridAutomaton" block to (0, 1), as specified in the prompt.

Simulate the model for 100 seconds and plot the values of "x1" and "x2" as a function of time.