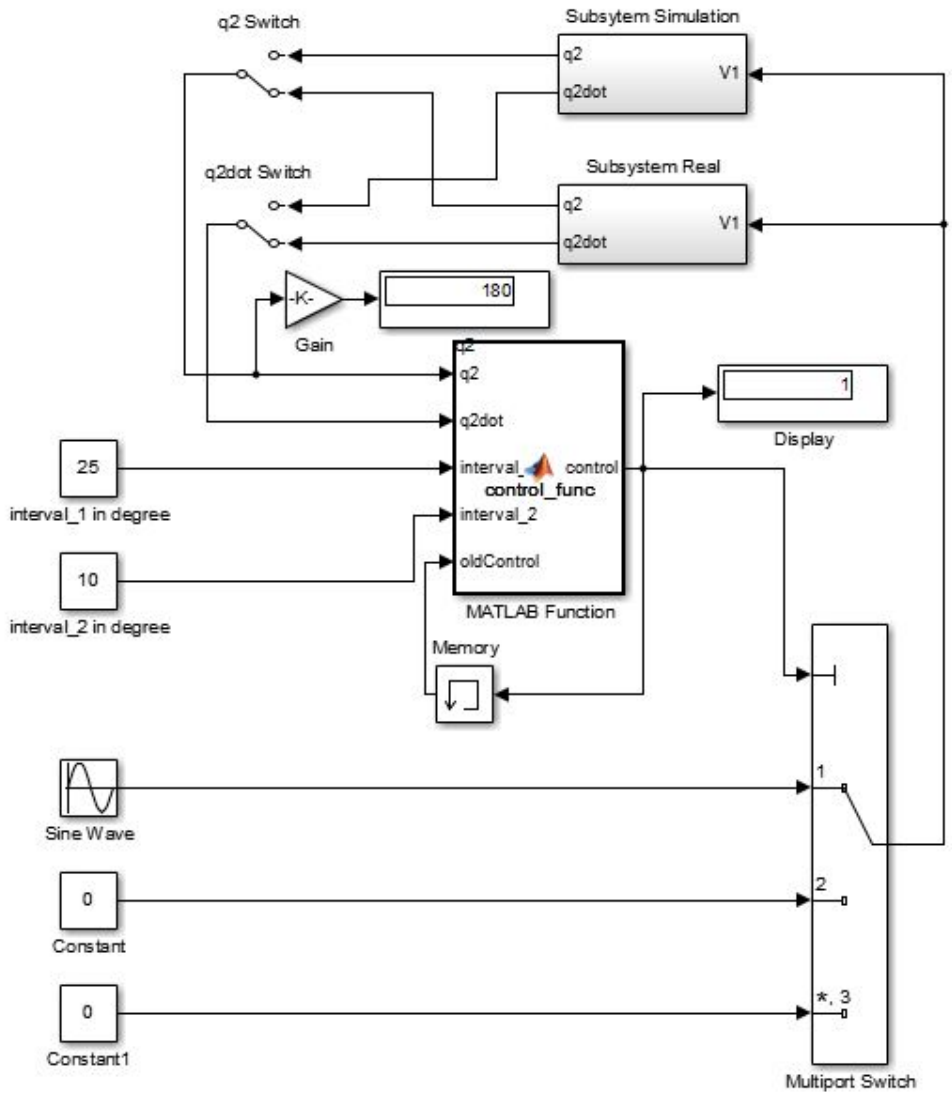
1. Items done this session:

We continue to implement the previous lab.
In the previous lab, we misunderstand the objective of the assignment. What we thought was implementing the control function based on q2's region, yet it should be implemented with the information of the previous state for keeping the state order . After change the control function, the stick successfully swing to the top and going back to the state 1.

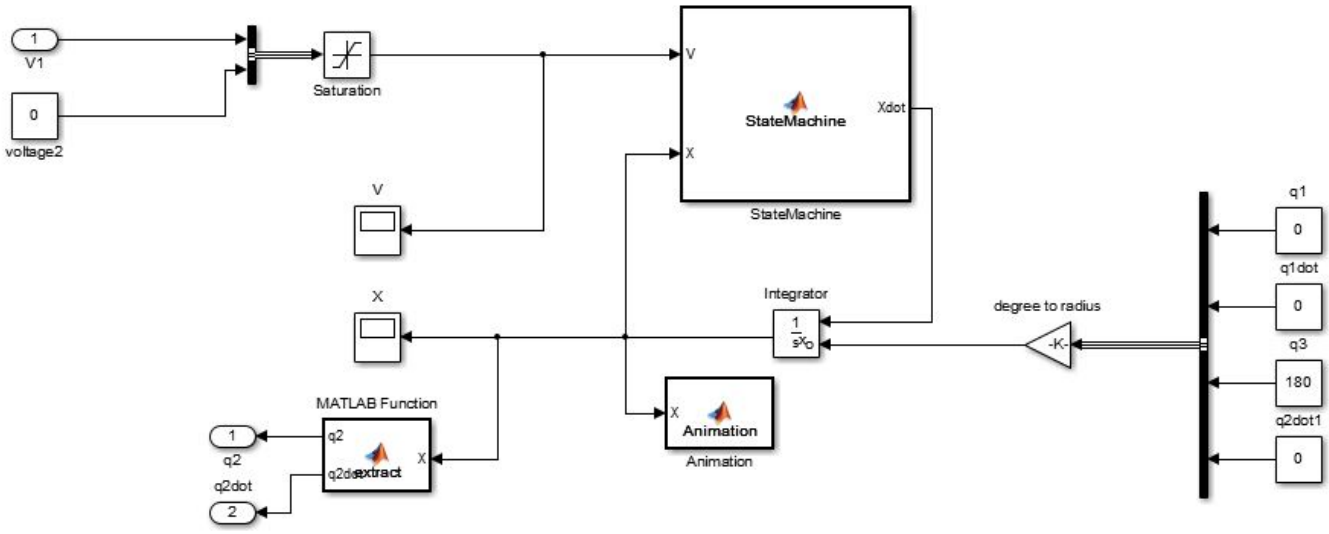
Our final model design. The "Subsystem Simulation" and "Subsystem Real" include the simulation collect and real collect from the previous assignment respectively. Two switch to choose either using simulation or real.



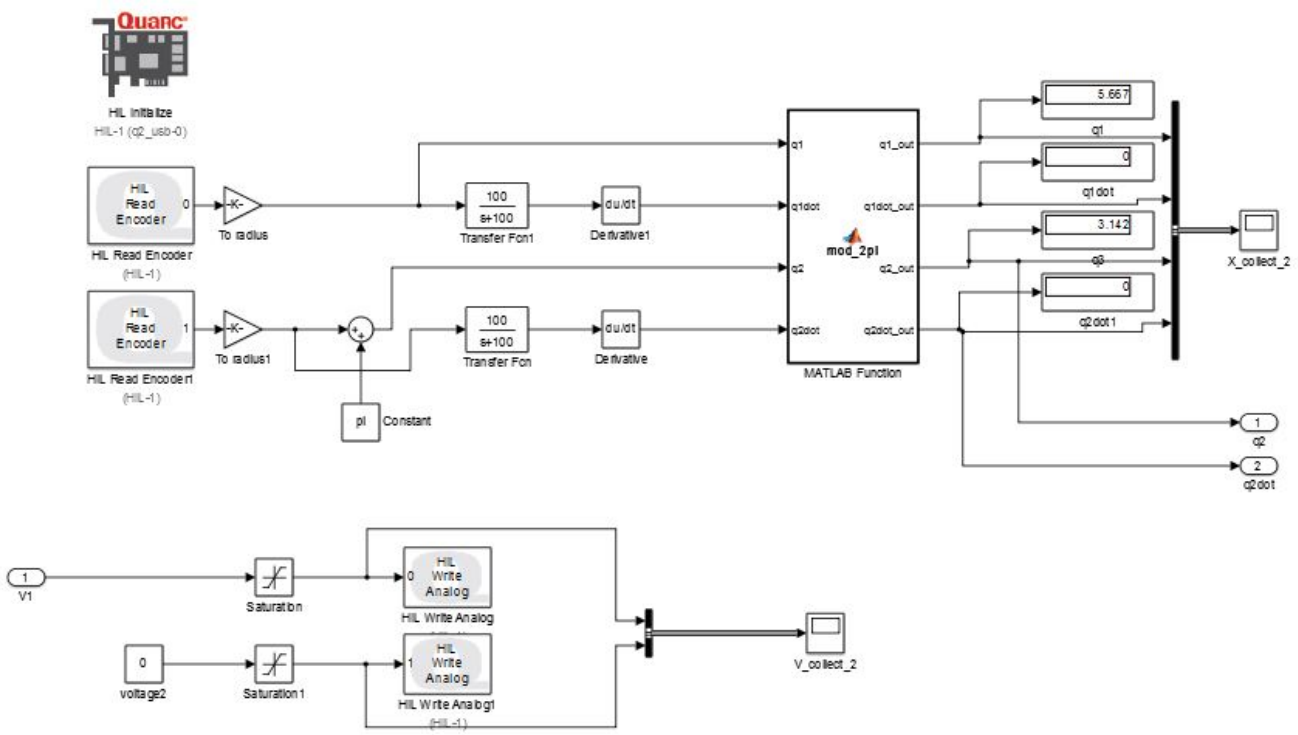
The "control_func" is a MATLAB function for decision of control state. In the function, the first condition statement assign current state and the second condition statement considering both the previous state and the current state to choose corresponding control state, so that the state will only change in the desired order.

```
function control  = control_func(q2, q2dot, interval_1, interval_2, oldControl)
% interval_1 = 25;
% interval_2 = 10;
point_a = (interval_1)*pi/180;
point_b = (360-interval_1)*pi/180;
point_c = (180-interval_2)*pi/180;
point_d = (180+interval_2)*pi/180;


if(q2 < point_a || q2 > point_b)
    currentState = 3;
elseif(q2 > point_c && q2 < point_d)
    currentState = 1;
else
    currentState = 2;
end

if(oldControl == 1 && currentState == 3 )
    control = 3;
elseif oldControl == 2 && currentState == 1
    control = 1;
elseif oldControl == 3 && currentState == 2
    control = 2;
else
    control = oldControl;
end
end
```

The "Subsystem Simulation", except taken V1 as input from the upper model, is identical to the the model collecting data in simulation from the previous assignment.



And the "Subsystem Real", except taken V1 as input from the upper model, is identical to the model collecting data for real world from the previous assignment.



"mod_2pi" is a MATLAB function that converts q1 and q2 into range of [0. 2pi].

```
function [q1_out, q1dot_out, q2_out, q2dot_out] = mod_2pi(q1, q1dot, q2, q2dot)
q1_out = mod(q1, 2*pi);
q1dot_out = q1dot;
q2_out = mod(q2, 2*pi);
q2dot_out = q2dot;
end
```

2. Items for next session:

Continue on the next assignment.

3. Problems / Concerns:
The implementation this time convert q1 and q2 into range of [0,2pi], however for the future control system work might need the range of [-pi,pi], which should be deliberately convert.