5.b.)

W\_V2 = 40;

omega\_n\_V2 = omega\_n\_theta/W\_V2;

zeta\_V2 = 1;

P.airspeed\_pitch\_kp = (models.a\_V1 - 2\*zeta\_V2\*omega\_n\_V2)/P.K\_theta\_DC/P.gravity;

P.airspeed\_pitch\_ki = -omega\_n\_V2\*omega\_n\_V2/P.K\_theta\_DC/P.gravity;

P.airspeed\_pitch\_kd = 0.0;

kp =

0.1469

ki =

0.0361

kd =

0

5.c.)

if(firstTime)

PIR\_pitch\_hold(0,0,0,firstTime, P);

PIR\_alt\_hold\_using\_pitch(0,0,0,firstTime, P);

PIR\_airspeed\_hold\_using\_throttle(0,0,0,firstTime, P);

PIR\_airspeed\_hold\_using\_pitch(0,0,0,firstTime, P);

end

h\_hold = 5;

if h\_hat < h\_hold

% Climb Logic

delta\_t = 1;

theta\_c = PIR\_airspeed\_hold\_using\_pitch(Va\_c, Va\_hat, 0.0, firstTime, P);

elseif h\_hat > h\_c + h\_hold

% Descend Logic

delta\_t = 0;

theta\_c = PIR\_airspeed\_hold\_using\_pitch(Va\_c, Va\_hat, 0.0, firstTime, P);

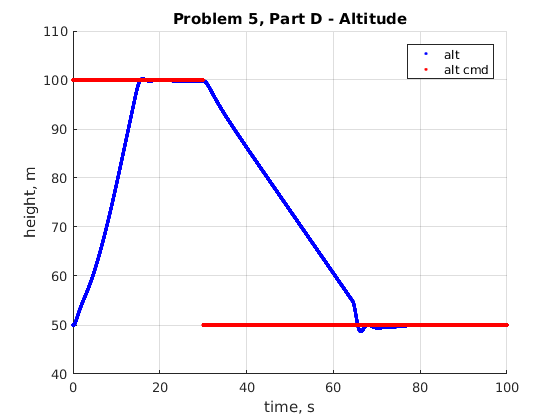
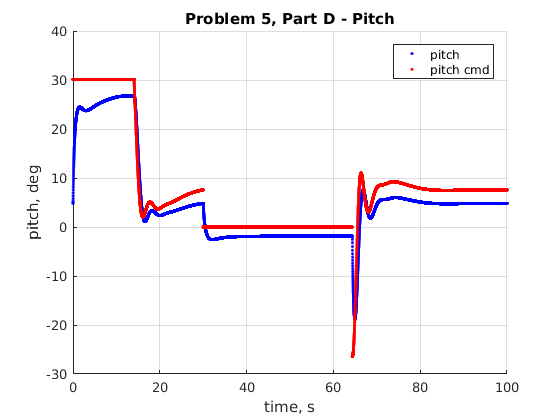
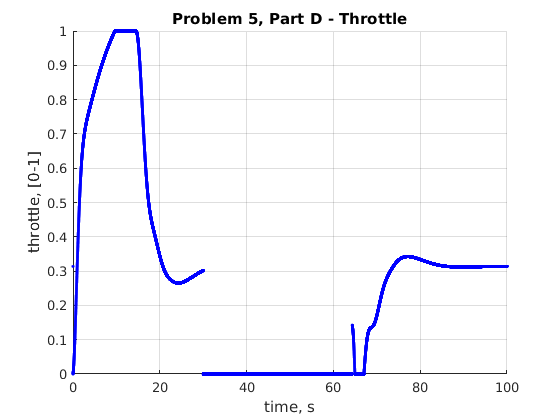
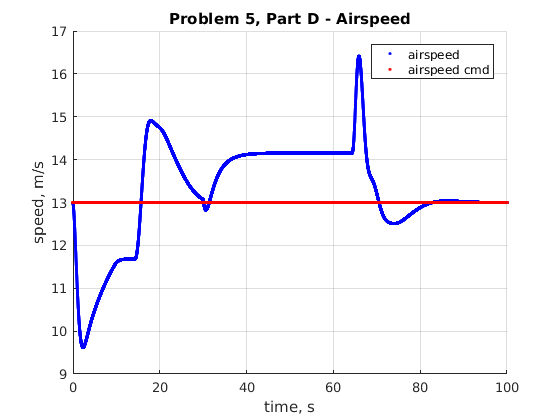
else

% Altitude Hold Logic

delta\_t = PIR\_airspeed\_hold\_using\_throttle(Va\_c, Va\_hat, 0.0, firstTime, P);

theta\_c = PIR\_alt\_hold\_using\_pitch(h\_c, h\_hat, 0, firstTime, P);

end

5.d.)

5.e.)

I would make sure there is no undershoot when going vehicle is commanded to go down in altitude.