Parameters we need to know

P.ct = 1.4865e-07; % Motor Thrust Coefficient

P.cq = 2.9250e-09; % Motor Torque Coefficient

function x\_next = mavStateTransitionFcn(x,uu)

dt = 0.005; % Time step size

pn = x(1);

pe = x(2);

pd = x(3);

u = x(4);

v = x(5);

w = x(6);

phi = x(7);

theta = x(8);

psi = x(9);

p = x(10);

q = x(11);

r = x(12);

fx = 0;

fy = 0;

fz = 0;

ell = 0;

m = 0;

n = 0;

w1 = uu(1);

w2 = uu(2);

w3 = uu(3);

w4 = uu(4);

P.gravity = 9.8331;

P.d = 0.223;

% Physical parameters of airframe

P.mass = 4.69 + payload + batteries; % kg 1.0230;

P.ct = 1.4865e-07; % Motor Thrust Coefficient

P.cq = 2.9250e-09; % Motor Torque Coefficient

% Moments about the x y and z axis

P.Jx = 0.0095;

P.Jy = 0.0095;

P.Jz = 0.0186;

P.Jm = 3.7882e-06;

P.Jb = [P.Jx,0,0; 0,P.Jy,0; 0,0,P.Jz];

P.Jbinv = inv(P.Jb);

P.Jxz = 0.0;

P.dctcq = [...

0,3.3037e-08,0,-3.3037e-08;...

-3.3037e-08,0,3.3037e-08,0;...

-2.9250e-09,2.9250e-09,-2.9250e-09,2.9250e-09...

];

% Motor Parameters

P.motor\_m = 0.23; % kg 0.0730;

P.motor\_dm = 0.2223;

P.motor\_h = 0.0318;

P.motor\_r = 0.0140;

% ESC Parameters

P.ESC\_m = 0.009; %kg 0.0300;

P.ESC\_a = 0.0254;

P.ESC\_b = 0.0572;

P.ESC\_ds = 0.0826;

% HUB (?) Parameters

P.HUB\_m = 0.4310;

P.HUB\_r = 0.0564;

P.HUB\_H = 0.0429;

% Arm Parameters

P.arms\_m = 0.0450; % Mass

P.arms\_r = 0.0325; % Radius(?)

P.arms\_L = 0.1857; % Length

P.arms\_da = 0.0508;

P.T = 0.0760;

P.minThr = 5;

P.cr = 80.5840;

P.b = 976.20;

P.plusConfig = 1;

W = [w1; w2; w3; w4];

Dist\_tau = [ell; m; n];

Dist\_F = [fx; fy; fz];

tau\_motorGyro = [q\*P.Jm\*2\*pi/60\*(-w1-w3+w2+w4); p\*P.Jm\*2\*pi/60\*(w1+w3-w2-w4); 0]; % Note: 2\*pi/60 required to convert from RPM to radians/s

Mb = (P.dctcq\*(W.^2))+ tau\_motorGyro + (Dist\_tau); % Mb = [tau1 tau2 tau3]'

% Thrust due to motor speed

% Force should be in units of Newtons for simplicity in calculating

% the acceleration in the angular velocity state equation

Fb = [0; 0; sum(P.ct\*(W.^2))]; %[0, 0, sum(ct\*w.^2)]'

% Obtain dP dQ dR

omb\_bi = [p; q; r];

OMb\_bi = [ 0,-r, q;

r, 0,-p;

-q, p, 0];

b\_omdotb\_bi = P.Jbinv\*(Mb-OMb\_bi\*P.Jb\*omb\_bi);

Rib = [...

cos(theta)\*cos(psi)...

(sin(phi)\*sin(theta)\*cos(psi) - cos(phi)\*sin(psi))...

(cos(phi)\*sin(theta)\*cos(psi) + sin(phi)\*sin(psi));

cos(theta)\*sin(psi)...

(sin(phi)\*sin(theta)\*sin(psi) + cos(phi)\*cos(psi))...

(cos(phi)\*sin(theta)\*sin(psi) - sin(phi)\*sin(psi));

-sin(theta) sin(phi)\*cos(theta) cos(phi)\*cos(theta)...

];

Rbi = Rib';

ge = [0; 0; -P.gravity];

gb = Rbi\*ge;

Dist\_Fb = Rbi\*Dist\_F;

% Compute Velocity and Position derivatives of body frame

vb = [u;v;w];

b\_dv = (1/P.mass)\*Fb+gb+Dist\_Fb-OMb\_bi\*vb; % Acceleration in body frame (FOR VELOCITY)

i\_dp = Rib\*vb;

pndot = i\_dp(1);

pedot = i\_dp(2);

pddot = i\_dp(3);

udot = b\_dv(1);

vdot = b\_dv(2);

wdot = b\_dv(3);

phidot = p + sin(phi)\*tan(theta)\*q + cos(phi)\*tan(theta)\*r;

thetadot = cos(phi)\*q - sin(phi)\*r;

psidot = (sin(phi)/cos(theta))\*q + (cos(phi)/cos(theta))\*r;

pdot = b\_omdotb\_bi(1);

qdot = b\_omdotb\_bi(2);

rdot = b\_omdotb\_bi(3);

x\_next = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0];

x\_next(1) = pn + pndot\*dt;

x\_next(2) = pe + pedot\*dt;

x\_next(3) = pd + pddot\*dt;

x\_next(4) = u + udot\*dt;

x\_next(5) = v + vdot\*dt;

x\_next(6) = w + wdot\*dt;

x\_next(7) = phi + phidot\*dt;

x\_next(8) = theta + thetadot\*dt;

x\_next(9) = psi + psidot\*dt;

x\_next(10) = p + pdot\*dt;

x\_next(11) = q + qdot\*dt;

x\_next(12) = r + rdot\*dt;

end