APÉNDICE E. CÓDIGO DEL SIMULADOR DINÁMICO

El código en Matlab para el simulador dinámico se presenta en este apartado, siendo cada entidad de código (archivo *M* o *MAT*) colocada en su respectiva sección.

E.1. DinamicaInversa.m

```
Clear

% --- Condiciones iniciales

% -- Fuerza y momento externos

fex = 0;
fey = 0;
fez = 0;
nex = 0;
ney = 0;
nez = 0;

% -- Para velocidad

vx = 0;
vy = 0;
vz = 0;
vel_alfa = 0;
vel_beta = 0;
vel_gamma = 0;

dx0 = [vx; vy; vz; vel_alfa; vel_beta; vel_gamma];

% -- Para posición

px = 0;
py = 0;
pz = 0.46167699973624;
alfa = 0;
beta = 0;
gamma = 0;

x0 = [px; py; pz; alfa; beta; gamma];

% -- Para torques

Torques = zeros(6,1);

% --- Configuración del tiempo

cont = 1;
t(cont) = 0;

% tmax = input('Dame el tiempo de simulacion: ');
tmax = 10;

% tstep = input('Dame el incremento de tiempo: ');
tstep = tmax / 100;

% --- Modelo

while t(cont) < tmax

    % -- Para aceleración

    acel = movfun (t(cont), x0, dx0, 'acel');

    ax = acel(1);
    ay = acel(2);
    az = acel(3);
    acel_alfa = acel(4);
    acel_beta = acel(5);
    acel_gamma = acel(6);

    % -- Actualización de datos

    X(cont,:) = [px py pz alfa beta gamma];
    dX(cont,:) = [vx vy vz vel_alfa vel_beta vel_gamma];
    ddX(cont,:) = [ax ay az acel_alfa acel_beta acel_gamma];

    % -- Geometría

    b = 110 / 100;
    d = 34 / 100;
    a = 76.8 / 100;
    c = 14.5 / 100;

    B1 = [(sqrt(3)/6)*(2*b + d); d/2; 0];
    B2 = [(-sqrt(3)/6)*(b - d); (b + d)/2; 0];
    B3 = [(-sqrt(3)/6)*(b + 2*d); b/2; 0];
    B4 = [(-sqrt(3)/6)*(b + 2*d); -b/2; 0];
    B5 = [(-sqrt(3)/6)*(b - d); -(b + d)/2; 0];
    B6 = [(sqrt(3)/6)*(2*b + d); -d/2; 0];

    Base = [B1 B2 B3 B4 B5 B6];

    cA = cos(alfa);
    cB = cos(beta);
    cG = cos(gamma);
    cG60 = cos(gamma + pi/3);
    cG30 = cos(gamma + pi/6);
    sA = sin(alfa);
    sB = sin(beta);
    sG = sin(gamma);
    sG60 = sin(gamma + pi/3);
    sG30 = sin(gamma + pi/6);
    cBcG = cB * cG;
    cBcG60 = cB * cG60;
    cBsG30 = cB * sG30;
    cAsG = cA * sG;
    cAsG60 = cA * sG60;
    cAcG30 = cA * cG30;
    sBcG = sB * cG;
    sBcG60 = sB * cG60;
    sAsBcG30 = sA * sB * cG30;
    sAsBsG = sA * sB * sG;
    sAsBsG60 = sA * sB * sG60;
    sBsG30 = sB * sG30;
    sAcBsG = sA * cB * sG;
    sAcBsG60 = sA * cB * sG60;
    sAcBcG30 = sA * cB * cG30;

    a_sqrt3 = a / sqrt(3);
    c_sqrt3 = c / sqrt(3);
```

```
    T1 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (cBcG + sAsBsG) + px;
           a_sqrt3 * cAsG60 + c_sqrt3 * cAsG + py;
           a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (-sBcG + sAcBsG) + pz];
    T2 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
           a_sqrt3 * cAsG60 + c_sqrt3 * cAcG30 + py;
           a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];
    T3 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
           -a_sqrt3 * cAsG + c_sqrt3 * cAcG30 + py;
           -a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];
    T4 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
           -a_sqrt3 * cAsG + c_sqrt3 * -cAsG60 + py;
           -a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];
    T5 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
           a_sqrt3 * -cAcG30 + c_sqrt3 * -cAsG60 + py;
           a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];
    T6 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (cBcG + sAsBsG) + px;
           a_sqrt3 * -cAcG30 + c_sqrt3 * cAsG + py;
           a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (-sBcG + sAcBsG) + pz];

    Top =  [T1 T2 T3 T4 T5 T6];
    XTop(:,:,cont) = Top;

    % -- Cinemática de posición inversa

    angles(1) = 2;
    angles(2) = 1;
    angles(3) = 2;
    angles(4) = 1;
    angles(5) = 2;
    angles(6) = 1;

    p = 16.2 / 100;
    l = 78.6 / 100;

    Sigma = [-2 * pi / 3; -2 * pi / 3; 0; 0; 2 * pi / 3; 2 * pi / 3];

    for i = 1:6

        A(i) = 2 * p * (sin(Sigma(i)) * (Top(1,i) - Base(1,i)) - cos(Sigma(i)) * (Top(2,i) - Base(2,i)));
        B(i) = 2 * p * Top(3,i);
        C(i) = l^2 - p^2 - (Top(1,i) - Base(1,i))^2 - (Top(2,i) - Base(2,i))^2 - Top(3,i)^2;

        angle_cos(i) = acos(C(i) / sqrt(A(i)^2 + B(i)^2));
        angle_tan(i) = atan2 (A(i), B(i));

        if angles(i) == 1
            theta(i) = real(angle_cos(i) + angle_tan(i));
        elseif angles(i) == 2
            theta(i) = real(-angle_cos(i) + angle_tan(i));
        end

        Q(:,i) = [-p * sin(Sigma(i)) * sin(theta(i)) + Base(1,i); p * cos(Sigma(i)) * sin(theta(i)) + Base(2,i); -p * cos(theta(i))];
        XQ(:,i,cont) = Q(:,i);

    end

    % -- Cinemática Diferencial

    % - Definiciones

    E = [Top(1,:) - px; Top(2,:) - py; Top(3,:) - pz];
    P = Q - Base;
    L = Top(:,:) - Q;
    S = [cos(Sigma'); sin(Sigma'); zeros(1, 6)];

    pg = 0.538;
    lg = 0.5;

    Pg = pg * P;
    Lg = (1 - lg) * L;

    % - Para velocidad

    % Velocidad lineal de la Plataforma
    vtop = [vx; vy; vz];
    % Velocidad angular de la Plataforma
    wtop = [vel_alfa + vel_gamma * cos(alfa) * sin(beta); vel_beta - vel_gamma * sin(alfa); vel_gamma * cos(alfa) * cos(beta)];

    for i = 1:6
        w(i) = (dot (L(:,i), vtop) + dot (cross (E(:,i), L(:,i)), wtop)) / dot (S(:,i), cross(P(:,i), L(:,i)));
        wP(:,i) = w(i) .* S(:,i);
        wL(:,i) = (1/l^2) * (cross (L(:,i), vtop) + dot (L(:,i), E(:,i)) * wtop - dot (L(:,i), wtop) * E(:,i) - w(i) * (dot (L(:,i), P(:,i)) * S(:,i) - dot (L(:,i), S(:,i)) *
P(:,i)));
    end
```

```
% - Para aceleración

% Aceleración lineal de la Plataforma
atop = [ax; ay; az];
% Aceleración angular de la Plataforma
etop = [acel_alfa + vel_beta * vel_gamma * cos(alfa) * cos(beta) - vel_alfa * vel_gamma * sin(alfa) * sin(beta) + acel_gamma * cos(alfa) *
sin(beta); acel_beta - vel_alfa * vel_gamma * cos(alfa) - acel_gamma * sin(alfa); - vel_beta * vel_gamma * cos(alfa) * sin(beta) - vel_alfa * vel_gamma
* sin(alfa) * cos(beta) + acel_gamma * cos(alfa) * cos(beta)];

for i = 1:6

    Ei_cross = [0, -E(3,i), E(2,i); E(3,i), 0, -E(1,i); -E(2,i), E(1,i), 0];
    Li_cross = [0, -L(3,i), L(2,i); L(3,i), 0, -L(1,i); -L(2,i), L(1,i), 0];
    Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
    Pgi_cross = [0, -Pg(3,i), Pg(2,i); Pg(3,i), 0, -Pg(1,i); -Pg(2,i), Pg(1,i), 0];

    eP_a(:,:,i) = [(S(:,i) * L(:,i)' / dot (S(:,i), cross (P(:,i), L(:,i)))), (S(:,i) * cross (E(:,i), L(:,i))' / dot (S(:,i), cross (P(:,i), L(:,i)))];
    eP_v(:,i) = (dot (E(:,i), wtop) * dot (L(:,i), wtop) - norm(wtop)^2 * dot (L(:,i), E(:,i)) + norm(wP(:,i))^2 * dot (L(:,i), P(:,i)) + norm(wL(:,i))^2 * l^2) *
S(:,i) / dot (S(:,i), cross (P(:,i), L(:,i)));
    eP(:,i) = eP_a(:,:,i) * [atop; etop] + eP_v(:,i);

    aP_a(:,:,i) = -Pgi_cross * eP_a(:,:,i);
    aP_v(:,i) = -Pgi_cross * eP_v(:,i) - norm(wP(:,i))^2 * Pg(:,i);
    aP(:,i) = aP_a(:,:,i) * [atop; etop] + aP_v(:,i);

    eL_a(:,:,i) = (1/l^2) * ([Li_cross, (dot (L(:,i), E(:,i)) * eye(3) - E(:,i) * L(:,i)')] + (P(:,i) * L(:,i)' - dot (L(:,i), P(:,i)) * eye(3)) * eP_a(:,:,i));
    eL_v(:,i) = (1/l^2) * (dot (wtop, E(:,i)) * cross (L(:,i), wtop) - norm(wtop)^2 * cross (L(:,i), E(:,i)) + norm(wP(:,i))^2 * cross (L(:,i), P(:,i)) + (P(:,i) *
L(:,i)' - dot (L(:,i), P(:,i)) * eye(3)) * eP_v(:,i));
    eL(:,i) = eL_a(:,:,i) * [atop; etop] + eL_v(:,i);

    aL_a(:,:,i) = [eye(3), -Ei_cross] + Lgi_cross * eL_a(:,:,i);
    aL_v(:,i) = dot (wtop, E(:,i)) * wtop - norm(wtop)^2 * E(:,i) - norm(wL(:,i))^2 * Lg(:,i) + Lgi_cross * eL_v(:,i);
    aL(:,i) = aL_a(:,:,i) * [atop; etop] + aL_v(:,i);

end

% - Derivación de las Jacobianas

for i = 1:6

    Jf(i,:) = [L(:,i)', (cross (E(:,i), L(:,i))')] / dot (S(:,i), cross (P(:,i), L(:,i)));

    Si_x_Pgi = cross (S(:,i), Pg(:,i));
    JP(:,:,i) = [Si_x_Pgi(1) * Jf(i,:); Si_x_Pgi(2) * Jf(i,:); Si_x_Pgi(3) * Jf(i,:); S(1,i) * Jf(i,:); S(2,i) * Jf(i,:); zeros(1,6)];

    Li_cross = [0, -L(3,i), L(2,i); L(3,i), 0, -L(1,i); -L(2,i), L(1,i), 0];
    LPS_minus_LSP = dot(L(:,i), P(:,i)) * S(:,i) - dot(L(:,i), S(:,i)) * P(:,i);
    JLw(:,:,i) = (1/l^2) * ([Li_cross, (dot (L(:,i), E(:,i)) * eye(3) - E(:,i) * L(:,i)')] - [LPS_minus_LSP(1) * Jf(i,:); LPS_minus_LSP(2) * Jf(i,:);
LPS_minus_LSP(3) * Jf(i,:)]);

    Ei_cross = [0, -E(3,i), E(2,i); E(3,i), 0, -E(1,i); -E(2,i), E(1,i), 0];
    Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
    JLv(:,:,i) = ([eye(3), -Ei_cross] + Lgi_cross * JLw(:,:,i));

    JL(:,:,i) = [JLv(:,:,i); JLw(:,:,i)];

end

% -- Propiedades de masa e inercia

mtop = 84.5;
mP = 1.2;
mL = 4.3;

Itop = [8.11, -5.24E-4, -6.27E-5; -5.24E-4, 8.12, -4.58E-5; -6.27E-5, -4.58E-5, 16.2];
IP = [5.24E-3, 0, 6.62E-4; 0, 5.54E-3, 0; 6.62E-4, 0, 8.62E-4];
IL = [2.83E-1, 8.53E-9, 0; 8.53E-9, 2.83E-1, 0; 0, 0, 7.80E-4];

uRtop = [cB*cG + sA*sB*sG, -cB*sG + sA*sB*sG, cA*sB; cA*sB, cA*cG, -sA; -sB*cG + sA*cB*sG, sB*sG + sA*cB*cG, cA*cB];
uItop = uRtop * Itop * uRtop';

for i = 1:6
    cS = cos(Sigma(i));
    sS = sin(Sigma(i));
    cT = cos(theta(i));
    sT = sin(theta(i));
    uRP(:,:,i) = [cS, -sS*cT, sS*sT; sS, cS*cT, -cS*sT; 0, sT, cT];
    uIP(:,:,i) = uRP(:,:,i) * IP * uRP(:,:,i)';
end

L_unit = L / l;

for i = 1:6
```

```
        theta_L(i) = L_unit(3,i);
        phi_L(i) = atan2(L_unit(2,i), L_unit(1,i));
    end

    for i = 1:6
        cTL = cos(theta_L(i));
        sTL = sin(theta_L(i));
        cPL = cos(phi_L(i));
        sPL = sin(phi_L(i));
        uRL(:,:,i) = [cPL*cTL, -sPL, cPL*sTL; sPL*cTL, cPL, sPL*sTL; -sTL, 0, cTL];
        uIL(:,:,i) = uRL(:,:,i) * IL * uRL(:,:,i)';
    end

    % -- Dinámica

    fe = [fex; fey; fez];
    ne = [nex; ney; nez];
    g = [0; 0; -9.81];

    M = [mtop * eye(3), zeros(3); zeros(3), uItop];
    for i = 1:6
        M = M + JP(:,:,i)' * [mP * aP_a(:,:,i); uIP(:,:,i) * eP_a(:,:,i)] + JL(:,:,i)' * [mL * aL_a(:,:,i); uIL(:,:,i) * eL_a(:,:,i)];
    end

    V = [zeros(3,1); (cross (wtop, uItop * wtop))];
    for i = 1:6
        V = V + JP(:,:,i)' * [mP * aP_v(:,i); (uIP(:,:,i) * eP_v(:,i) + cross (wP(:,i), uIP(:,:,i) * wP(:,i)))] + JL(:,:,i)' * [mL * aL_v(:,i); (uIL(:,:,i) * eL_v(:,i) +
cross (wL(:,i), uIL(:,:,i) * wL(:,i)))];
    end

    G = [mtop * g; zeros(3,1)];
    for i = 1:6
        G = G + JP(:,:,i)' * [mP * g; zeros(3,1)] + JL(:,:,i)' * [mL * g; zeros(3,1)];
    end
    G = -1 * G;

    E = -1 * [fe; ne];

    Torques = inv(Jf') * (M * [atop; etop] + V + G + E);

    % -- Actualizacion de condiciones

    Tau(cont,:) = Torques';
    tprev = t(cont);
    cont = cont + 1;
    t(cont) = tprev + tstep;

    vel = movfun (t(cont), x0, dx0, 'vel');

    vx = vel(1);
    vy = vel(2);
    vz = vel(3);
    vel_alfa = vel(4);
    vel_beta = vel(5);
    vel_gamma = vel(6);

    pos = movfun (t(cont), x0, dx0, 'pos');

    px = pos(1);
    py = pos(2);
    pz = pos(3);
    alfa = pos(4);
    beta = pos(5);
    gamma = pos(6);

end

t(cont) = [];

% Graficación

figure (1);

subplot (2,3,1);
plot (t, X(:,1), 'd', t, X(:,2), '+', t, X(:,3), 'o');
axis square;
h = legend('px', 'py', 'pz', 'Location', 'NorthOutside');

subplot (2,3,4);
plot (t, X(:,4) * 180/pi, '*', t, X(:,5) * 180/pi, '.', t, X(:,6) * 180/pi, 'x');
axis square;
h = legend('alfa', 'beta', 'gamma', 'Location', 'NorthOutside');

subplot (2,3,2);
```

```
plot (t, dX(:,1), 'd', t, dX(:,2), '+', t, dX(:,3), 'o');
axis square;
h = legend('vx', 'vy', 'vz', 'Location', 'NorthOutside');

subplot (2,3,5);
plot (t, dX(:,4) * 180/pi, '*', t, dX(:,5) * 180/pi, '.', t, dX(:,6) * 180/pi, 'x');
axis square;
h = legend('vel alfa', 'vel beta', 'vel gamma', 'Location', 'NorthOutside');

subplot (2,3,3);
plot (t, ddX(:,1), 'd', t, ddX(:,2), '+', t, ddX(:,3), 'o');
axis square;
h = legend('ax', 'ay', 'az', 'Location', 'NorthOutside');

subplot (2,3,6);
plot (t, ddX(:,4) * 180/pi, '*', t, ddX(:,5) * 180/pi, '.', t, ddX(:,6) * 180/pi, 'x');
axis square;
h = legend('acel alfa', 'acel beta', 'acel gamma', 'Location', 'NorthOutside');

figure (2);

plot (t, Tau(:,1), 'd', t, Tau(:,2), '+', t, Tau(:,3), 'o', t, Tau(:,4), '*', t, Tau(:,5), '.', t, Tau(:,6), 'x');
axis square;
h = legend('Torque 1', 'Torque 2', 'Torque 3', 'Torque 4', 'Torque 5', 'Torque 6', 'Location', 'EastOutside');

set(gcf,'DoubleBuffer','on');

figure (3);

axis([-1 1 -1 1 0 1]);
grid on;
xlabel('x(t)');
ylabel('y(t)');
zlabel('z(t)');
title('Trayectoria para la dinámica inversa') ;

tinit = 0;
patch ('Vertices', Base', 'Faces', [1 2 3 4 5 6], 'FaceVertexCData', 1, 'FaceColor', 'none');

for k = 1 : length(t)

      Xleg = [Base(1,:); XQ(1,:,k); XTop(1,:,k)];
      Yleg = [Base(2,:); XQ(2,:,k); XTop(2,:,k)];
      Zleg = [Base(3,:); XQ(3,:,k); XTop(3,:,k)];

      h1 = line(Xleg, Yleg, Zleg, 'LineWidth', 1, 'Color', 'k', 'Marker', 'o', 'MarkerSize', 2);
      h2 = patch ('Vertices', XTop(:,:,k)', 'Faces', [1 2 3 4 5 6], 'FaceVertexCData', 1, 'FaceColor', 'k');

      F(k) = getframe(gcf);

      if k ~= length(t)
            delete (h1);
            delete (h2);
      end

end

movie2avi (F, 'robotmovie.avi','compression', 'None','fps',10);
```

E.2. movfun.m

```
function y = movfun (tf, x0, dx0, type)

syms tx;

% Demo para z
posfun = [0; 0; 0.35 * (0.5 - 0.5 * cos(2*pi*tx/5)); 0; 0; 0];

% Demo para gamma
%posfun = [0; 0; 0.35/2 * (0.5 - 0.5 * cos(pi*tx/5)); 0; 0; 15*pi/180 * (0.5 - 0.5*cos(pi*tx/5)) * cos(2*pi*tx)];

% Demo para x-y
%posfun = [0.2 * (0.5 - 0.5*cos(pi*tx/5)) * cos(pi*tx); 0.2 * (0.5 - 0.5*cos(pi*tx/5)) * sin(pi*tx); 0.35/2 * (0.5 - 0.5 * cos(pi*tx/5)); 0; 0; 0];

% Demo para alfa-beta
%posfun = [0; 0; 0.35/2 * (0.5 - 0.5 * cos(pi*tx/5)); 10*pi/180 *(0.5 - 0.5*cos(pi*tx/5)) * cos(pi*tx); 10*pi/180 *(0.5 - 0.5*cos(pi*tx/5)) * sin(pi*tx); 0];
```

```
% Demo para directa (gamma)
%posfun = [0; 0; 0.15/2 * (0.5 - 0.5 * cos(pi*tx/5)); 0; 0; 5*pi/180 * (0.5 - 0.5*cos(pi*tx/5)) * cos(pi*tx/2)];

dposfun = diff(posfun, 'tx');
ddposfun = diff(dposfun, 'tx');

switch type
	case 'acel'
		y = subs (ddposfun, tf);
	case 'vel'
		y = subs (dposfun, tf) + dx0;
	case 'pos'
		y = subs (posfun, tf) + dx0 * tf + x0;
end

%---

End
```

E.3. DinamicaDirecta.m

```
% Parámetros

px_0 = 0;
py_0 = 0;
pz_0 = 0.46167699973624;
alfa_0 = 0;
beta_0 = 0;
gamma_0 = 0;

vx_0 = 0;
vy_0 = 0;
vz_0 = 0;
vel_alfa_0 = 0;
vel_beta_0 = 0;
vel_gamma_0 = 0;

tmax = input('Dame el tiempo de simulacion: ');
t = 0;

% Calculo

% y(1)->  px, y(2)->  py, y(3)-> pz, y(4)->  alfa,    y(5)->  beta,    y(6)->
gamma
% y(7)->  vx, y(8)->  vy, y(9)-> vz, y(10)-> vel_alfa, y(11)-> vel_beta,
y(12)-> vel_gamma

options = odeset('Mass', @mass, 'MStateDependence', 'strong',
'MassSingular', 'maybe');

[t, Y] = ode15s(@fdyn, [0 tmax], [px_0, py_0, pz_0, alfa_0, beta_0,
gamma_0, vx_0, vy_0, vz_0, vel_alfa_0, vel_beta_0, vel_gamma_0],
options);

px = Y(:,1);
py = Y(:,2);
pz = Y(:,3);
alfa = Y(:,4);
beta = Y(:,5);
gamma = Y(:,6);

vx = Y(:,7);
vy = Y(:,8);
vz = Y(:,9);
vel_alfa = Y(:,10);
vel_beta = Y(:,11);
vel_gamma = Y(:,12);

ax = diff(vx)./diff(t);
ax(length(t)) = ax(length(t)-1);

ay = diff(vy)./diff(t);
ay(length(t)) = ay(length(t)-1);

az = diff(vz)./diff(t);
az(length(t)) = az(length(t)-1);
```

```
acel_alfa = diff(vel_alfa)./diff(t);
acel_alfa(length(t)) = acel_alfa(length(t)-1);

acel_beta = diff(vel_beta)./diff(t);
acel_beta(length(t)) = acel_beta(length(t)-1);

acel_gamma = diff(vel_gamma)./diff(t);
acel_gamma(length(t)) = acel_gamma(length(t)-1);

figure (4);

subplot (2,3,1);
plot (t, px, 'd', t, py, '+', t, pz, 'o');
axis square;
h = legend('px', 'py', 'pz', 'Location', 'NorthOutside');

subplot (2,3,4);
plot (t, alfa * 180/pi, '*', t, beta * 180/pi, '.', t, gamma * 180/pi, 'x');
axis square;
h = legend('alfa', 'beta', 'gamma', 'Location', 'NorthOutside');

subplot (2,3,2);
plot (t, vx, 'd', t, vy, '+', t, vz, 'o');
axis square;
h = legend('vx', 'vy', 'vz', 'Location', 'NorthOutside');

subplot (2,3,5);
plot (t, vel_alfa * 180/pi, '*', t, vel_beta * 180/pi, '.', t, vel_gamma * 180/pi,
'x');
axis square;
h = legend('vel alfa', 'vel beta', 'vel gamma', 'Location', 'NorthOutside');

subplot (2,3,3);
plot (t, ax, 'd', t, ay, '+', t, az, 'o');
axis square;
h = legend('ax', 'ay', 'az', 'Location', 'NorthOutside');

subplot (2,3,6);
plot (t, acel_alfa * 180/pi, '*', t, acel_beta * 180/pi, '.', t, acel_gamma *
180/pi, 'x');
axis square;
h = legend('acel alfa', 'acel beta', 'acel gamma', 'Location', 'NorthOutside');
```

E.4. mass.m

```
function Mt = mass(t,y)
Mt = zeros(6,6);

% y(1)-> px, y(2)-> py, y(3)-> pz, y(4)-> alfa, y(5)-> beta, y(6)-> gamma

px = y(1);
py = y(2);
pz = y(3);
alfa = y(4);
beta = y(5);
gamma = y(6);

% Geometria

b = 110 / 100;
d = 34 / 100;
a = 76.8 / 100;
c = 14.5 / 100;

B1 = [(sqrt(3)/6)*(2*b + d); d/2; 0];
B2 = [(-sqrt(3)/6)*(b - d); (b + d)/2; 0];
B3 = [(-sqrt(3)/6)*(b + 2*d); b/2; 0];
B4 = [(-sqrt(3)/6)*(b + 2*d); -b/2; 0];
B5 = [(-sqrt(3)/6)*(b - d); -(b + d)/2; 0];
B6 = [(sqrt(3)/6)*(2*b + d); -d/2; 0];

Base = [B1 B2 B3 B4 B5 B6];
```

```
cA = cos(alfa);
cB = cos(beta);
cG = cos(gamma);
cG60 = cos(gamma + pi/3);
cG30 = cos(gamma + pi/6);
sA = sin(alfa);
sB = sin(beta);
sG = sin(gamma);
sG60 = sin(gamma + pi/3);
sG30 = sin(gamma + pi/6);
cBcG = cB * cG;
cBcG60 = cB * cG60;
cBsG30 = cB * sG30;
cAsG = cA * sG;
cAsG60 = cA * sG60;
cAcG30 = cA * cG30;
sBcG = sB * cG;
sBcG60 = sB * cG60;
sAsBcG30 = sA * sB * cG30;
sAsBsG = sA * sB * sG;
sAsBsG60 = sA * sB * sG60;
sBsG30 = sB * sG30;
sAcBsG = sA * cB * sG;
sAcBsG60 = sA * cB * sG60;
sAcBcG30 = sA * cB * cG30;

a_sqrt3 = a / sqrt(3);
c_sqrt3 = c / sqrt(3);
```

```
T1 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (cBcG + sAsBsG) + px;
       a_sqrt3 * cAsG60 + c_sqrt3 * cAsG + py;
       a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (-sBcG + sAcBsG) + pz];

T2 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
       a_sqrt3 * cAsG60 + c_sqrt3 * cAcG30 + py;
       a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];

T3 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
       -a_sqrt3 * cAsG + c_sqrt3 * cAcG30 + py;
       -a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];

T4 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
       -a_sqrt3 * cAsG + c_sqrt3 * -cAsG60 + py;
       -a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];

T5 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
       a_sqrt3 * -cAcG30 + c_sqrt3 * -cAsG60 + py;
       a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];

T6 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (cBcG + sAsBsG) + px;
       a_sqrt3 * -cAcG30 + c_sqrt3 * cAsG + py;
       a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (-sBcG + sAcBsG) + pz];

Top =  [T1 T2 T3 T4 T5 T6];

% Cinemática Inversa

angles(1) = 2;
angles(2) = 1;
angles(3) = 2;
angles(4) = 1;
angles(5) = 2;
angles(6) = 1;

p = 16.2 / 100;
l = 78.6 / 100;

Sigma = [-2 * pi / 3; -2 * pi / 3; 0; 0; 2 * pi / 3; 2 * pi / 3];

for i = 1:6

      A(i) = 2 * p * (sin(Sigma(i)) * (Top(1,i) - Base(1,i)) - cos(Sigma(i)) * (Top(2,i) - Base(2,i)));
      B(i) = 2 * p * Top(3,i);
      C(i) = l^2 - p^2 - (Top(1,i) - Base(1,i))^2 - (Top(2,i) - Base(2,i))^2 - Top(3,i)^2;

      angle_cos(i) = acos(C(i) / sqrt(A(i)^2 + B(i)^2));
      angle_tan(i) = atan2 (A(i), B(i));

      if angles(i) == 1
```

```
        theta(i) = real(angle_cos(i) + angle_tan(i));
    elseif angles(i) == 2
        theta(i) = real(-angle_cos(i) + angle_tan(i));
    end

    Q(:,i) = [-p * sin(Sigma(i)) * sin(theta(i)) + Base(1,i); p * cos(Sigma(i)) * sin(theta(i)) + Base(2,i); -p * cos(theta(i))];

end

% Definiciones

E = [Top(1,:) - px; Top(2,:) - py; Top(3,:) - pz];
P = Q - Base;
L = Top - Q;
S = [cos(Sigma'); sin(Sigma'); zeros(1, 6)];

pg = 0.538;
lg = 0.5;

Pg = pg * P;
Lg = (1 - lg) * L;

% Relaciones de Aceleracion

for i = 1:6

    Ei_cross = [0, -E(3,i), E(2,i); E(3,i), 0, -E(1,i); -E(2,i), E(1,i), 0];
    Li_cross = [0, -L(3,i), L(2,i); L(3,i), 0, -L(1,i); -L(2,i), L(1,i), 0];
    Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
    Pgi_cross = [0, -Pg(3,i), Pg(2,i); Pg(3,i), 0, -Pg(1,i); -Pg(2,i), Pg(1,i), 0];

    eP_a(:,:,i) = [(S(:,i) * L(:,i)' / dot (S(:,i), cross (P(:,i), L(:,i)))), (S(:,i) * cross (E(:,i), L(:,i))' / dot (S(:,i), cross (P(:,i), L(:,i))))];
    aP_a(:,:,i) = -Pgi_cross * eP_a(:,:,i);

    eL_a(:,:,i) = (1/l^2) * ([Li_cross, (dot (L(:,i), E(:,i)) * eye(3) - E(:,i) * L(:,i)')] + (P(:,i) * L(:,i)' - dot (L(:,i), P(:,i)) * eye(3)) * eP_a(:,:,i));
    aL_a(:,:,i) = [eye(3), -Ei_cross] + Lgi_cross * eL_a(:,:,i);

end

% Derivación de las Jacobianas

for i = 1:6

    Jf(i,:) = [L(:,i)', (cross (E(:,i), L(:,i))')] / dot (S(:,i), cross (P(:,i), L(:,i)));

    Si_x_Pgi = cross (S(:,i), Pg(:,i));
    JP(:,:,i) = [Si_x_Pgi(1) * Jf(i,:); Si_x_Pgi(2) * Jf(i,:); Si_x_Pgi(3) * Jf(i,:); S(1,i) * Jf(i,:); S(2,i) * Jf(i,:); zeros(1,6)];

    Li_cross = [0, -L(3,i), L(2,i); L(3,i), 0, -L(1,i); -L(2,i), L(1,i), 0];
    LPS_minus_LSP = dot(L(:,i), P(:,i)) * S(:,i) - dot(L(:,i), S(:,i)) * P(:,i);
    JLw(:,:,i) = (1/l^2) * ([Li_cross, (dot (L(:,i), E(:,i)) * eye(3) - E(:,i) * L(:,i)')] - [LPS_minus_LSP(1) * Jf(i,:); LPS_minus_LSP(2) * Jf(i,:);
LPS_minus_LSP(3) * Jf(i,:)]);

    Ei_cross = [0, -E(3,i), E(2,i); E(3,i), 0, -E(1,i); -E(2,i), E(1,i), 0];
    Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
    JLv(:,:,i) = ([eye(3), -Ei_cross] + Lgi_cross * JLw(:,:,i));

    JL(:,:,i) = [JLv(:,:,i); JLw(:,:,i)];

end

% Propiedades de masa e inercia

mtop = 84.5;
mP = 1.2;
mL = 4.3;

Itop = [8.11, -5.24E-4, -6.27E-5; -5.24E-4, 8.12, -4.58E-5; -6.27E-5, -4.58E-5, 16.2];
IP = [5.24E-3, 0, 6.62E-4; 0, 5.54E-3, 0; 6.62E-4, 0, 8.62E-4];
IL = [2.83E-1, 8.53E-9, 0; 8.53E-9, 2.83E-1, 0; 0, 0, 7.80E-4];

uRtop = [cB*cG + sA*sB*sG, -cB*sG + sA*sB*sG, cA*sB; cA*sB, cA*cG, -sA; -sB*cG + sA*cB*sG, sB*sG + sA*cB*cG, cA*cB];
uItop = uRtop * Itop * uRtop';

for i = 1:6
    cS = cos(Sigma(i));
    sS = sin(Sigma(i));
    cT = cos(theta(i));
    sT = sin(theta(i));
    uRP(:,:,i) = [cS, -sS*cT, sS*sT; sS, cS*cT, -cS*sT; 0, sT, cT];
    uIP(:,:,i) = uRP(:,:,i) * IP * uRP(:,:,i)';
end

L_unit = L / l;
```

```
for i = 1:6
	theta_L(i) = L_unit(3,i);
	phi_L(i) = atan2(L_unit(2,i), L_unit(1,i));
end

for i = 1:6
	cTL = cos(theta_L(i));
	sTL = sin(theta_L(i));
	cPL = cos(phi_L(i));
	sPL = sin(phi_L(i));
	uRL(:,:,i) = [cPL*cTL, -sPL, cPL*sTL; sPL*cTL, cPL, sPL*sTL; -sTL, 0, cTL];
	uIL(:,:,i) = uRL(:,:,i) * IL * uRL(:,:,i)';
end

% Ecuación del Espacio de Estados

M = [mtop * eye(3), zeros(3); zeros(3), uItop];
for i = 1:6
	M = M + JP(:,:,i)' * [mP * aP_a(:,:,i); uIP(:,:,i) * eP_a(:,:,i)] + JL(:,:,i)' * [mL * aL_a(:,:,i); uIL(:,:,i) * eL_a(:,:,i)];
end

Mt = [eye(6) zeros(6); zeros(6) M];

%---

End
```

E.5. fdyn.m

```
function dy = fdyn(t,y)
dy = zeros(12,1);

% y(1)->  px, y(2)->  py, y(3)-> pz, y(4)->  alfa,    y(5)->  beta,    y(6)->
gamma
% y(7)->  vx, y(8)->  vy, y(9)-> vz, y(10)-> vel_alfa, y(11)-> vel_beta,
y(12)-> vel_gamma

px = y(1);
py = y(2);
pz = y(3);
alfa = y(4);
beta = y(5);
gamma = y(6);

vx = y(7);
vy = y(8);
vz = y(9);
vel_alfa = y(10);
vel_beta = y(11);
vel_gamma = y(12);

Torques = torquefun(t,3);

% Geometria

b = 110 / 100;
d = 34 / 100;
a = 76.8 / 100;
c = 14.5 / 100;

B1 = [(sqrt(3)/6)*(2*b + d); d/2; 0];
B2 = [(-sqrt(3)/6)*(b - d); (b + d)/2; 0];
B3 = [(-sqrt(3)/6)*(b + 2*d); b/2; 0];
B4 = [(-sqrt(3)/6)*(b + 2*d); -b/2; 0];
B5 = [(-sqrt(3)/6)*(b - d); -(b + d)/2; 0];
B6 = [(sqrt(3)/6)*(2*b + d); -d/2; 0];

Base = [B1 B2 B3 B4 B5 B6];

cA = cos(alfa);
cB = cos(beta);
cG = cos(gamma);
cG60 = cos(gamma + pi/3);
cG30 = cos(gamma + pi/6);
sA = sin(alfa);
sB = sin(beta);
sG = sin(gamma);
sG60 = sin(gamma + pi/3);
sG30 = sin(gamma + pi/6);
cBcG = cB * cG;
cBcG60 = cB * cG60;
cBsG30 = cB * sG30;
cAsG = cA * sG;
cAsG60 = cA * sG60;
cAcG30 = cA * cG30;
sBcG = sB * cG;
sBcG60 = sB * cG60;
sAsBcG30 = sA * sB * cG30;
sAsBsG = sA * sB * sG;
sAsBsG60 = sA * sB * sG60;
sBsG30 = sB * sG30;
sAcBsG = sA * cB * sG;
sAcBsG60 = sA * cB * sG60;
sAcBcG30 = sA * cB * cG30;

a_sqrt3 = a / sqrt(3);
c_sqrt3 = c / sqrt(3);

T1 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (cBcG + sAsBsG) + px;
	a_sqrt3 * cAsG60 + c_sqrt3 * cAsG + py;
	a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (-sBcG + sAcBsG) + pz];

T2 = [ a_sqrt3 * (cBcG60 + sAsBsG60) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
	a_sqrt3 * cAsG60 + c_sqrt3 * cAcG30 + py;
	a_sqrt3 * (-sBcG60 + sAcBsG60) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];

T3 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBsG30 + sAsBcG30) + px;
	-a_sqrt3 * cAsG + c_sqrt3 * cAcG30 + py;
	-a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBsG30 + sAcBcG30) + pz];
```

```
T4 = [ -a_sqrt3 * (cBcG + sAsBsG) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
       -a_sqrt3 * cAsG + c_sqrt3 * -cAsG60 + py;
       -a_sqrt3 * (-sBcG + sAcBsG) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];

T5 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (-cBcG60 - sAsBsG60) + px;
       a_sqrt3 * -cAcG30 + c_sqrt3 * -cAsG60 + py;
       a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (sBcG60 - sAcBsG60) + pz];

T6 = [ a_sqrt3 * (cBsG30 - sAsBcG30) + c_sqrt3 * (cBcG + sAsBsG) + px;
       a_sqrt3 * -cAcG30 + c_sqrt3 * cAsG + py;
       a_sqrt3 * (-sBsG30 - sAcBcG30) + c_sqrt3 * (-sBcG + sAcBsG) + pz];

Top =  [T1 T2 T3 T4 T5 T6];

% Cinemática Inversa

angles(1) = 2;
angles(2) = 1;
angles(3) = 2;
angles(4) = 1;
angles(5) = 2;
angles(6) = 1;

p = 16.2 / 100;
l = 78.6 / 100;

Sigma = [-2 * pi / 3; -2 * pi / 3; 0; 0; 2 * pi / 3; 2 * pi / 3];

for i = 1:6

       A(i) = 2 * p * (sin(Sigma(i)) * (Top(1,i) - Base(1,i)) - cos(Sigma(i)) * (Top(2,i) - Base(2,i)));
       B(i) = 2 * p * Top(3,i);
       C(i) = l^2 - p^2 - (Top(1,i) - Base(1,i))^2 - (Top(2,i) - Base(2,i))^2 - Top(3,i)^2;

       angle_cos(i) = acos(C(i) / sqrt(A(i)^2 + B(i)^2));
       angle_tan(i) = atan2 (A(i), B(i));

       if angles(i) == 1
              theta(i) = real(angle_cos(i) + angle_tan(i));
       elseif angles(i) == 2
              theta(i) = real(-angle_cos(i) + angle_tan(i));
       end

       Q(:,i) = [-p * sin(Sigma(i)) * sin(theta(i)) + Base(1,i); p * cos(Sigma(i)) * sin(theta(i)) + Base(2,i); -p * cos(theta(i))];

end

% Definiciones

E = [Top(1,:) - px; Top(2,:) - py; Top(3,:) - pz];
P = Q - Base;
L = Top - Q;
S = [cos(Sigma'); sin(Sigma'); zeros(1, 6)];

pg = 0.538;
lg = 0.5;

Pg = pg * P;
Lg = (1 - lg) * L;

% Relaciones de Velocidad

% Velocidad lineal de la Plataforma
vtop = [vx; vy; vz];
% Velocidad angular de la Plataforma
wtop = [vel_alfa + vel_gamma * cos(alfa) * sin(beta); vel_beta - vel_gamma * sin(alfa); vel_gamma * cos(alfa) * cos(beta)];

for i = 1:6
       w(i) = (dot (L(:,i), vtop) + dot (cross (E(:,i), L(:,i)), wtop)) / dot (S(:,i), cross(P(:,i), L(:,i)));
       wP(:,i) = w(i) .* S(:,i);
       wL(:,i) = (1/l^2) * (cross (L(:,i), vtop) + dot (L(:,i), E(:,i)) * wtop - dot (L(:,i), wtop) * E(:,i) - w(i) * (dot (L(:,i), P(:,i)) * S(:,i) - dot (L(:,i), S(:,i)) * P(:,i)));
end

% Relaciones de Aceleracion

for i = 1:6

       Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
       Pgi_cross = [0, -Pg(3,i), Pg(2,i); Pg(3,i), 0, -Pg(1,i); -Pg(2,i), Pg(1,i), 0];

       eP_v(:,i) = (dot (E(:,i), wtop) * dot (L(:,i), wtop) - norm(wtop)^2 * dot (L(:,i), E(:,i)) + norm(wP(:,i))^2 * dot (L(:,i), P(:,i)) + norm(wL(:,i))^2 * l^2) * S(:,i)
/ dot (S(:,i), cross (P(:,i), L(:,i)));
       aP_v(:,i) = -Pgi_cross * eP_v(:,i) - norm(wP(:,i))^2 * Pg(:,i);
```

```
    eL_v(:,i) = (1/l^2) * (dot (wtop, E(:,i)) * cross (L(:,i), wtop) - norm(wtop)^2 * cross (L(:,i), E(:,i)) + norm(wP(:,i))^2 * cross (L(:,i), P(:,i)) + (P(:,i) * L(:,i)'
- dot (L(:,i), P(:,i)) * eye(3)) * eP_v(:,i));
    aL_v(:,i) = dot (wtop, E(:,i)) * wtop - norm(wtop)^2 * E(:,i) - norm(wL(:,i))^2 * Lg(:,i) + Lgi_cross * eL_v(:,i);

end

% Derivación de las Jacobianas

for i = 1:6

    Jf(i,:) = [L(:,i)', (cross (E(:,i), L(:,i))')] / dot (S(:,i), cross (P(:,i), L(:,i)));

    Si_x_Pgi = cross (S(:,i), Pg(:,i));
    JP(:,:,i) = [Si_x_Pgi(1) * Jf(i,:); Si_x_Pgi(2) * Jf(i,:); Si_x_Pgi(3) * Jf(i,:); S(1,i) * Jf(i,:); S(2,i) * Jf(i,:); zeros(1,6)];

    Li_cross = [0, -L(3,i), L(2,i); L(3,i), 0, -L(1,i); -L(2,i), L(1,i), 0];
    LPS_minus_LSP = dot(L(:,i), P(:,i)) * S(:,i) - dot(L(:,i), S(:,i)) * P(:,i);
    JLw(:,:,i) = (1/l^2) * ([Li_cross, (dot (L(:,i), E(:,i)) * eye(3) - E(:,i) * L(:,i)')] - [LPS_minus_LSP(1) * Jf(i,:); LPS_minus_LSP(2) * Jf(i,:);
LPS_minus_LSP(3) * Jf(i,:)]);

    Ei_cross = [0, -E(3,i), E(2,i); E(3,i), 0, -E(1,i); -E(2,i), E(1,i), 0];
    Lgi_cross = [0, -Lg(3,i), Lg(2,i); Lg(3,i), 0, -Lg(1,i); -Lg(2,i), Lg(1,i), 0];
    JLv(:,:,i) = ([eye(3), -Ei_cross] + Lgi_cross * JLw(:,:,i));

    JL(:,:,i) = [JLv(:,:,i); JLw(:,:,i)];

end

% Propiedades de masa e inercia

mtop = 84.5;
mP = 1.2;
mL = 4.3;

Itop = [8.11, -5.24E-4, -6.27E-5; -5.24E-4, 8.12, -4.58E-5; -6.27E-5, -4.58E-5, 16.2];
IP = [5.24E-3, 0, 6.62E-4; 0, 5.54E-3, 0; 6.62E-4, 0, 8.62E-4];
IL = [2.83E-1, 8.53E-9, 0; 8.53E-9, 2.83E-1, 0; 0, 0, 7.80E-4];

uRtop = [cB*cG + sA*sB*sG, -cB*sG + sA*sB*sG, cA*sB; cA*sB, cA*cG, -sA; -sB*cG + sA*cB*sG, sB*sG + sA*cB*cG, cA*cB];
uItop = uRtop * Itop * uRtop';

for i = 1:6
    cS = cos(Sigma(i));
    sS = sin(Sigma(i));
    cT = cos(theta(i));
    sT = sin(theta(i));
    uRP(:,:,i) = [cS, -sS*cT, sS*sT; sS, cS*cT, -cS*sT; 0, sT, cT];
    uIP(:,:,i) = uRP(:,:,i) * IP * uRP(:,:,i)';
end

L_unit = L / l;

for i = 1:6
    theta_L(i) = L_unit(3,i);
    phi_L(i) = atan2(L_unit(2,i), L_unit(1,i));
end

for i = 1:6
    cTL = cos(theta_L(i));
    sTL = sin(theta_L(i));
    cPL = cos(phi_L(i));
    sPL = sin(phi_L(i));
    uRL(:,:,i) = [cPL*cTL, -sPL, cPL*sTL; sPL*cTL, cPL, sPL*sTL; -sTL, 0, cTL];
    uIL(:,:,i) = uRL(:,:,i) * IL * uRL(:,:,i)';
end

% Ecuación del Espacio de Estados

fex = 0;
fey = 0;
fez = 0;
nex = 0;
ney = 0;
nez = 0;

fe = [fex; fey; fez];
ne = [nex; ney; nez];

g = [0; 0; -9.81];

V = [zeros(3,1); (cross (wtop, uItop * wtop))];
for i = 1:6
    V = V + JP(:,:,i)' * [mP * aP_v(:,i); (uIP(:,:,i) * eP_v(:,i) + cross (wP(:,i), uIP(:,:,i) * wP(:,i)))] + JL(:,:,i)' * [mL * aL_v(:,i); (uIL(:,:,i) * eL_v(:,i) + cross
(wL(:,i), uIL(:,:,i) * wL(:,i)))];
```

```
end

G = [mtop * g; zeros(3,1)];
for i = 1:6
      G = G + JP(:,:,i)' * [mP * g; zeros(3,1)] + JL(:,:,i)' * [mL * g; zeros(3,1)];
end
G = -1 * G;

E = -1 * [fe; ne];

ddx = Jf' * Torques - V - G - E;

acel_ang = [1, 0, -tan(beta); 0, 1, -tan(alfa)/cos(beta); 0, 0, 1/(cos(alfa)*cos(beta))] * ddx(4:6) - [vel_beta * vel_gamma * cA * cB - vel_alfa * vel_gamma
* sA * sB; -vel_alfa * vel_gamma * cA; -vel_beta * vel_gamma * cA * sB - vel_alfa * vel_gamma * sA * cB];

% Para ode15s

% y(1)-> px, y(2)-> py, y(3)-> pz,  y(4)->  alfa,     y(5)->  beta,     y(6)-> gamma
% y(7)-> vx, y(8)-> vy, y(9)-> vz,  y(10)-> vel_alfa, y(11)-> vel_beta, y(12)-> vel_gamma

dy(1) = y(7);
dy(2) = y(8);
dy(3) = y(9);
dy(4) = y(10);
dy(5) = y(11);
dy(6) = y(12);
dy(7:9) = ddx(1:3);
dy(10:12) = acel_ang;

%---

end
```

E.6. torquefun.m

```
function y = torquefun (t, type)

switch type

      case 1
      y = [ -31.63691858289651; 31.63691858839332; -31.63691858289658; 31.63691858839335; -31.63691858289652; 31.63691858839331];
      break;

      case 2
      y = 20 * cos(t) + 10 * [-1; 1; -1; 1; -1; 1];
      break;

      case 3
      t0 = 0 : 0.1 : 10;
      Tdata = [
      -19.80920168353985  19.76466087589335 -19.80920167406651  19.76466084271792 -19.80920185138935  19.76466102968121;
      -19.82639043279160  19.77073796589080 -19.82639042400725  19.77073793519485 -19.82639058806476  19.77073810805629;
      -19.87673290457883  19.78993853369579 -19.87673289782435  19.78993851027561 -19.87673302289020  19.78993864180281;
      -19.95669087638690  19.82507265943393 -19.95669087288879  19.82507264760645 -19.95669093564061  19.82507271340727;
      -20.06079085876569  19.88050195621960 -20.06079085955123  19.88050195953148 -20.06079084063079  19.88050193971543;
      -20.18219393682571  19.96167279009211 -20.18219394263119  19.96167281106802 -20.18219382810859  19.96167269160162;
      -20.31337865676528  20.07449912174318 -20.31337866794255  20.07449916168305 -20.31337845059458  20.07449893538945;
      -20.44685358053673  20.22463961055669 -20.44685359697046  20.22463966938749 -20.44685327725806  20.22463933658750;
      -20.57582512363038  20.41673354152239 -20.57582514468150  20.41673361770641 -20.57582473130537  20.41673318653247;
      -20.69476504170228  20.65367774044880 -20.69476506619257  20.65367783095095 -20.69476457612533  20.65367731737359;
      -20.79984458987369  20.93603521989541 -20.79984461612284  20.93603532021210 -20.79984407405912  20.93603474780305;
      -20.88922395145960  21.26165973543926 -20.88922397737773  21.26165983969669 -20.88922341460227  21.26165923911582;
      -20.96320242231606  21.62559640250736 -20.96320244554473  21.62559650363799 -20.96320189828414  21.62559591179290;
      -21.02424467550067  22.02028045115446 -21.02424469358633  22.02028054116114 -21.02424420096723  22.02027999996144;
      -21.07689999358862  22.43601253762295 -21.07690000416555  22.43601260793496 -21.07689960594416  22.43601216256606;
      -21.12762493427701  22.86165003797868 -21.12762493523414  22.86165007989386 -21.12762466947975  22.86164977657429;
      -21.18450790550119  23.28542750813703 -21.18450789511059  23.28542751333204 -21.18450779601778  23.28542739658716;
      -21.25688136677980  23.69580942074916 -21.25688134377008  23.69580938182720 -21.25688143974130  23.69580949140225;
      -21.35480039659760  24.08228306529431 -21.35480036016720  24.08228297633103 -21.35480067225380  24.08228334395450;
      -21.48837176703484  24.43601490024134 -21.48837171681025  24.43601475728459 -21.48837225753620  24.43601540356416;
      -21.66693949044673  24.75031490023679 -21.66693942640108  24.75031470171788 -21.66694019893288  24.75031563348479;
      -21.89816976535908  25.02087696647550 -21.89816968771467  25.02087671349467 -21.89817068529892  25.02087792190348;
      -22.18712208472952  25.24578752786252 -22.18712199386958  25.24578722432030 -22.18712319945530  25.24578868395519;
      -22.53542981304332  25.42531871137089 -22.53542970945718  25.42531836392801 -22.53543109545618  25.42532003307201;
      -22.94072738573241  25.56154669169398 -22.94072727000962  25.56154630955551 -22.94072879822484  25.56154813166091;
      -23.39644271817655  25.65785849352989 -23.39644259105548  25.65785808806367 -23.39644421289715  25.65785999433749;
      -23.89202390248465  25.71842794161448 -23.89202376494594  25.71842752582899 -23.89202542212863  25.71842943873687;
```

```
    -24.41360169387809 25.74774849301157 -24.41360154725895 25.74774808093220 -24.41360317319461 25.74774991834213;
    -24.94502211825126 25.75030277333608 -24.94502196434560 25.75030237932681 -24.94502348639829 25.75030405896782;
    -25.46913253991013 25.73042414441276 -25.46913238101222 25.73042378249201 -25.46913372366317 25.73042522639085;
    -25.96917716254073 25.69236720554815 -25.96917700139770 25.69236688874042 -25.96917809020588 25.69236802731832;
    -26.43015311675040 25.64055834485479 -26.43015295640251 25.64055808460667 -26.43015372247787 25.64055886016048;
    -26.83999028767829 25.57995315264804 -26.83999013119460 25.57995295832320 -26.83999051576804 25.57995332767633;
    -27.19044096244828 25.51639382088339 -27.19044081259322 25.51639369934165 -27.19044077120787 25.51639363553147;
    -27.47759554901938 25.45684442386557 -27.47759540790931 25.45684437912717 -27.47759491378323 25.45684387209010;
    -27.70197636908739 25.40939066113619 -27.70197623790327 25.40939069413565 -27.70197528412368 25.40938975046036;
    -27.86820160476559 25.38292535259734 -27.86820148357728 25.38292546103405 -27.86820008395365 25.38292410295697;
    -27.98425357300872 25.38649929799890 -27.98425346073809 25.38649947629398 -27.98425164932587 25.38649774019440;
    -28.06042586221640 25.42839084126751 -28.06042575673206 25.42839108063368 -28.06042358620184 25.42838901531867;
    -28.10805804425761 25.51502211934442 -28.10805794256492 25.51502240797154 -28.10805548169761 25.51502007301085;
    -28.13819073990832 25.64990652182133 -28.13819063838356 25.64990684517949 -28.13818796900457 25.64990430944964;
    -28.16028516410191 25.83283194063758 -28.16028505872452 25.83283228190161 -28.16028227235242 25.83282962234922;
    -28.18114861748820 26.05945759487300 -28.18114850404448 26.05945793547586 -28.18114569843929 26.05945523595451;
    -28.20419022122695 26.32143233185770 -28.20419009548760 26.32143265217790 -28.20418737117648 26.32143000209406;
    -28.22909958226871 26.60704739588532 -28.22909944016403 26.60704767605799 -28.22909689692881 26.60704516852637;
    -28.25199623556477 26.90234202467102 -28.25199607339161 26.90234224548831 -28.25199380666049 26.90233997475560;
    -28.26604300571798 27.19250791609349 -28.26604282039942 27.19250805992916 -28.26604091750604 27.19250611797145;
    -28.26245797593096 27.46339985759357 -28.26245776532122 27.46339990926984 -28.26245630168075 27.46339838166063;
    -28.23180587474056 27.70295444609894 -28.23180563792909 27.70295439361521 -28.23180467330716 27.70295335495176;
    -28.16540929204078 27.90233999589272 -28.16540902958236 27.90233983096313 -28.16540860475853 27.90233934028079;
    -28.05670051379648 28.05670052337851 -28.05670022778808 28.05670024176576 -28.05670036199637 28.05670033842836;
    -27.90233969655436 28.16540900747960 -27.90233939050575 28.16540860915251 -27.90234008003233 28.16540930967403;
    -27.70295387312458 28.23180530378666 -27.70295355162070 28.23180479294637 -27.70295476983770 28.23180608908828;
    -27.46339903768839 28.26245713811224 -27.46339870589828 28.26245652312234 -27.46340040482745 28.26245838145471;
    -27.19250688477606 28.26604193129700 -27.19250654790849 28.26604122452146 -27.19250866118087 28.26604358747195;
    -26.90234082408948 28.25199496396424 -26.90234048690903 28.25199418147963 -26.90234293377445 28.25199696975768;
    -26.60704607247406 28.22909816014306 -26.60704573897146 28.22909732127235 -26.60704842887203 28.22910043746362;
    -26.32143093408135 28.20418870034478 -26.32143060734874 28.20418782696809 -26.32143344453969 28.20419116005049;
    -26.05945617124555 28.18114705254424 -26.05945585355427 28.18114616817883 -26.05945874133816 28.18114959870213;
    -25.83283053811367 28.16028361052446 -25.83283023114802 28.16028273921923 -25.83283307545896 28.16028614489667;
    -25.64990518455450 28.13818925168632 -25.64990488972658 28.13818841682172 -25.64990760200360 28.13819167828359;
    -25.51502088772633 28.10805667180053 -25.51502060648987 28.10805589491181 -25.51502310595211 28.10805890135776;
    -25.42838975112214 28.06042465027683 -25.42838948521152 28.06042395001620 -25.42839170074117 28.06042660447697;
    -25.38649837981575 27.98425255874258 -25.38649813135765 27.98425195006718 -25.38650000318899 27.98425417396083;
    -25.38292463074116 27.86820081604935 -25.38292440221476 27.86820030968063 -25.38292588353465 27.86820204634451;
    -25.40939015304274 27.70197582324635 -25.40938994709680 27.70197542540396 -25.40939100552874 27.70197664231865;
    -25.45684413930157 27.47759525212092 -25.45684395846522 27.47759496450600 -25.45684457729204 27.47759565400892;
    -25.51639376138912 27.19044090927903 -25.51639360773442 27.19044072927061 -25.51639378665355 27.19044090768347;
    -25.57995331128516 26.83999046245434 -25.57995318610719 26.83999038348601 -25.57995294132309 26.83999008873402;
    -25.64055870635459 26.43015349451927 -25.64055860992632 26.43015350661259 -25.64055797345601 26.43015279457269;
    -25.69236774690718 25.96917771114335 -25.69236767834701 25.96917780157780 -25.69236669651136 25.96917674152915;
    -25.73042483593000 25.46913322229797 -25.73042479319553 25.46913337639007 -25.73042352436557 25.46913204602881;
    -25.75030358004004 24.94502289491579 -25.75030356003930 24.94502309686515 -25.75030207177383 24.94502157736494;
    -25.74774937639349 24.41360252515289 -25.74774937519122 24.41360275889668 -25.74774774093849 24.41360113044651;
    -25.71842886156425 23.89202475052602 -25.71842887465483 23.89202500053489 -25.71842717020054 23.89202333867069;
    -25.65785941036024 23.39644354850058 -25.65785943297881 23.39644380045571 -25.65785773282114 23.39644217334723;
    -25.56154756813280 22.94072816827481 -25.56154759557786 22.94072840958543 -25.56154596940816 22.94072687627639;
    -25.42531951439946 22.53543052278578 -25.42531954233964 22.53543074293447 -25.42531805181171 22.53542935239523;
    -25.24578823027716 22.18712270199022 -25.24578825501636 22.18712289271913 -25.24578695099082 22.18712168348691;
    -25.02087754806246 21.89817027586382 -25.02087756673823 21.89817043123202 -25.02087648718332 21.89816943139692;
    -24.75031534847270 21.66693988527914 -24.75031535916826 21.66694000162662 -24.75031452787266 21.66693922896964;
    -24.43601521046955 21.48837204252466 -24.43601521223066 21.48837211837112 -24.43601463842929 21.48837158055320;
    -24.08228324037534 21.35480055415505 -24.08228323313728 21.35480059005007 -24.08228291207450 21.35480028485431;
    -23.69580947023660 21.25688141264477 -23.69580945467872 21.25688141097513 -23.69580936904166 21.25688132668715;
    -23.28542744707690 21.18450785035549 -23.28542742443354 21.18450781510740 -23.28542754649902 21.18450793116091;
    -22.86164988546392 21.12762479267164 -22.86164985732266 21.12762472914514 -22.86165015159260 21.12762501715130;
    -22.43601231524809 21.07689978316007 -22.43601228336400 21.07689969763717 -22.43601270942771 21.07690012286650;
    -22.02028018150877 21.02424441598509 -22.02028014765270 21.02424431535890 -22.02028066301952 21.02424483866944;
    -21.62559610781552 20.96320213440703 -21.62559607366585 20.96320202578653 -21.62559663635372 20.96320260590969;
    -21.26165943642057 20.88922365558533 -21.26165940349502 20.88922354589594 -21.26165997423050 20.88922414192053;
    -20.93603493486920 20.79984430498983 -20.93603490448643 20.79984420059346 -20.93603544845096 20.79984477444230;
    -20.65367748467724 20.69476478420186 -20.65367745793541 20.69476469055949 -20.65367794606210 20.69476520923941;
    -20.41673332676086 20.57582490646158 -20.41673330452020 20.57582482786093 -20.41673371438987 20.57582526528808;
    -20.22463944477707 20.44685341259680 -20.22463942763837 20.44685335196091 -20.22463974403599 20.44685369021450;
    -20.07449900900469 20.31337854260619 -20.07449899728225 20.31337850139359 -20.07449921245958 20.31337873132335;
    -19.96167273055278 20.18219387667086 -19.96167272424544 20.18219385486918 -19.96167283790530 20.18219397605946;
    -19.88050194629657 20.06079084880475 -19.88050194506649 20.06079084501307 -19.88050196405534 20.06079086517585;
    -19.82507269212940 19.95669090929310 -19.82507269529899 19.95669092088074 -19.82507263297958 19.95669085479837;
    -19.78993859910060 19.87673297018318 -19.78993860567746 19.87673299350889 -19.78993848096293 19.87673286166482;
    -19.77073805186187 19.82639051885230 -19.77073806059639 19.82639054952672 -19.77073789665440 19.82639037655277;
    -19.76466096888020 19.80920177655930 -19.76466097835363 19.80920180973477 -19.76466080103071 19.80920162277142;
     ];
    y  = interp1(t0, Tdata, t)';
    break;

end

%---

end
```

E.7. Parameters.mat

```
Base =
0.73323484187082    -0.21939310229206    -0.51384173957877    -0.51384173957877    -0.21939310229206     0.73323484187082
0.17000000000000     0.72000000000000     0.55000000000000    -0.55000000000000    -0.72000000000000    -0.17000000000000
0                    0                    0                    0                    0                    0

g =
0
0
-9.81000000000000

IL =
0.28300000000000     0.00000000853000     0
0.00000000853000     0.28300000000000     0
0                    0                    0.00078000000000

IP =
0.00524000000000     0                    0.00066200000000
0                    0.00554000000000     0
0.00066200000000     0                    0.00086200000000

Itop =
8.11000000000000    -0.00052400000000    -0.00006270000000
-0.00052400000000    8.12000000000000    -0.00004580000000
-0.00006270000000    -0.00004580000000    16.20000000000000

L =
-0.42781654946951    0.39923771114463    0.02857883832489    0.02857883832489    0.39923771114463    -0.42781654946951
0.21400000000000    -0.26350000000000    -0.47750000000000    0.47750000000000    0.26350000000000    -0.21400000000000
0.62367699973624    0.62367699973624    0.62367699973624    0.62367699973624    0.62367699973624    0.62367699973624

Lg =
-0.21390827473476    0.19961885557231    0.01428941916244    0.01428941916244    0.19961885557231    -0.21390827473476
0.10700000000000    -0.13175000000000    -0.23875000000000    0.23875000000000    0.13175000000000    -0.10700000000000
0.31183849986812    0.31183849986812    0.31183849986812    0.31183849986812    0.31183849986812    0.31183849986812

mL = 4.30000000000000
mP = 1.20000000000000
mtop = 84.50000000000000

P =
0.00000000000000    -0.00000000000000    0                    0                    -0.00000000000000    0.00000000000000
-0.00000000000000    0.00000000000000    0.00000000000000    -0.00000000000000    -0.00000000000000    0.00000000000000
-0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000

Pg =
0.00000000000000    -0.00000000000000    0                    0                    -0.00000000000000    0.00000000000000
-0.00000000000000    0.00000000000000    0.00000000000000    -0.00000000000000    -0.00000000000000    0.00000000000000
-0.08715600000000    -0.08715600000000    -0.08715600000000    -0.08715600000000    -0.08715600000000    -0.08715600000000

px = 0
py = 0
pz = 0.46167699973624

Q =
0.73323484187083    -0.21939310229206    -0.51384173957877    -0.51384173957877    -0.21939310229206     0.73323484187083
0.17000000000000     0.72000000000000     0.55000000000000    -0.55000000000000    -0.72000000000000    -0.17000000000000
-0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000    -0.16200000000000

S =
-0.50000000000000    -0.50000000000000    1.00000000000000    1.00000000000000    -0.50000000000000    -0.50000000000000
-0.86602540378444    -0.86602540378444    0                    0                    0.86602540378444     0.86602540378444
0                    0                    0                    0                    0                    0

Sigma =
-2.09439510239320
-2.09439510239320
0
0
2.09439510239320
2.09439510239320

Top =
0.30541829240131    0.17984460885257    -0.48526290125388    -0.48526290125388    0.17984460885257    0.30541829240131
0.38400000000000    0.45650000000000    0.07250000000000    -0.07250000000000    -0.45650000000000    -0.38400000000000
0.46167699973624    0.46167699973624    0.46167699973624    0.46167699973624    0.46167699973624    0.46167699973624
```