Final Project

```
% Diego Colón
% MAE 4733
% Final Design Poject Solution
```

1 - State Space Model

Taking the eqautions provided in the exam can be expressed in a state space form ($x = f(x,u), y = \phi(x,h)$, with state $x = \begin{bmatrix} h & \dot{h} & z & \dot{z} & \theta & \dot{\theta} \end{bmatrix}^T$ and input $u = \begin{bmatrix} F & \tau \end{bmatrix}^T$ as:

$$\dot{x} = f(x,u) = \begin{bmatrix} \dot{h} \\ \dfrac{\mathrm{F}\cos(\theta)}{M+2m} - g \\ \dot{z} \\ -\dfrac{\mu\dot{z}}{M+2m} - \dfrac{\mathrm{F}\sin(\theta)}{M+2m} \\ \dot{\theta} \\ \dfrac{\tau}{2\mathrm{md}^2 + J} \end{bmatrix}$$

$$y = \phi(x,u) = \begin{bmatrix} h \\ z \\ \theta \end{bmatrix}$$

2 - Equilibium Trajectory

```
%% Create all the system variables
syms s
syms h h_ast h_dot h_ddot z z_dot z_ddot theta theta_dot theta_ddot
syms g M m mu d J
syms F tau

%% Write the VTOL Dynamics
h_ddot = -g +(cos(theta)*F)/(M+2*m);
z_ddot = -(mu*z_dot)/(M+2*m) - (sin(theta)*F)/(M+2*m);
theta_ddot = tau/(2*m*d^2 + J);


%% Create State, Dynamics and Input
x = [h; h_dot; z; z_dot; theta; theta_dot];
process = [h_dot; h_ddot; z_dot; z_ddot; theta_dot; theta_ddot];
meas = [h; z; theta];
u = [F;tau];

%% Solve for trajectory to linearize about
S = solve(process,[x;u]);
x_star = [S.h+10;S.h_dot;S.z;S.z_dot;S.theta;S.theta_dot];
u_star = [S.F;S.tau];
y_star= [S.h+10;S.z;S.theta];
```

```
%% Linearize Dynamics
A = simplify(subs(jacobian(process,x),[x;u],[x_star;u_star]));
B = simplify(subs(jacobian(process,u),[x;u],[x_star;u_star]));
C = simplify(subs(jacobian(meas,x),[x;u],[x_star;u_star]));
D = simplify(subs(jacobian(meas,u),[x;u],[x_star;u_star]));

%% Create symbols and values array for substitution
k = [g; M; m;  mu; d; J];
k_val = [9.81; 1; 0.25;  0.1; 0.3; 0.0042];

%% MIMO TF
G = collect(simplify(C*inv(s*eye(size(A, 1)) - A)*B + D),s);

%% Extract relevant TF
G_HF = collect(simplify(subs(G(1,1),k,k_val)),s);
G_ZT = collect(simplify(subs(G(2,2),k,k_val)),s);
G_ThetaT = collect(simplify(subs(G(3,2),k,k_val)),s);
G_ZTheta = collect(simplify(G_ZT/G_ThetaT),s);
```

Altitude Controller Design (Pole Placement)

```
syms Kp Kd Ki
Cp1 = Kp + Kd*s;
CL1 = collect(simplify((G_HF*Cp1)/(1 + G_HF*Cp1)),s)
```

CL1 =

$$\frac{(2\,\mathrm{Kd})\,s + 2\,\mathrm{Kp}}{3\,s^2 + (2\,\mathrm{Kd})\,s + 2\,\mathrm{Kp}}$$

```
[Num,Den] = numden(G_HF);
G1 = tf(sym2poly(Num),sym2poly(Den));

t_r = 2;
zeta = 0.707;

omega_n = (2.16*zeta+0.6)/t_r;
Kp = (3/2)*omega_n^2;
Kd = 3*zeta*omega_n;
C1 = tf([Kd,Kp,0],[1,0])
```

```
C1 =

  2.256 s^2 + 1.697 s
  -------------------
           s

Continuous-time transfer function.
```

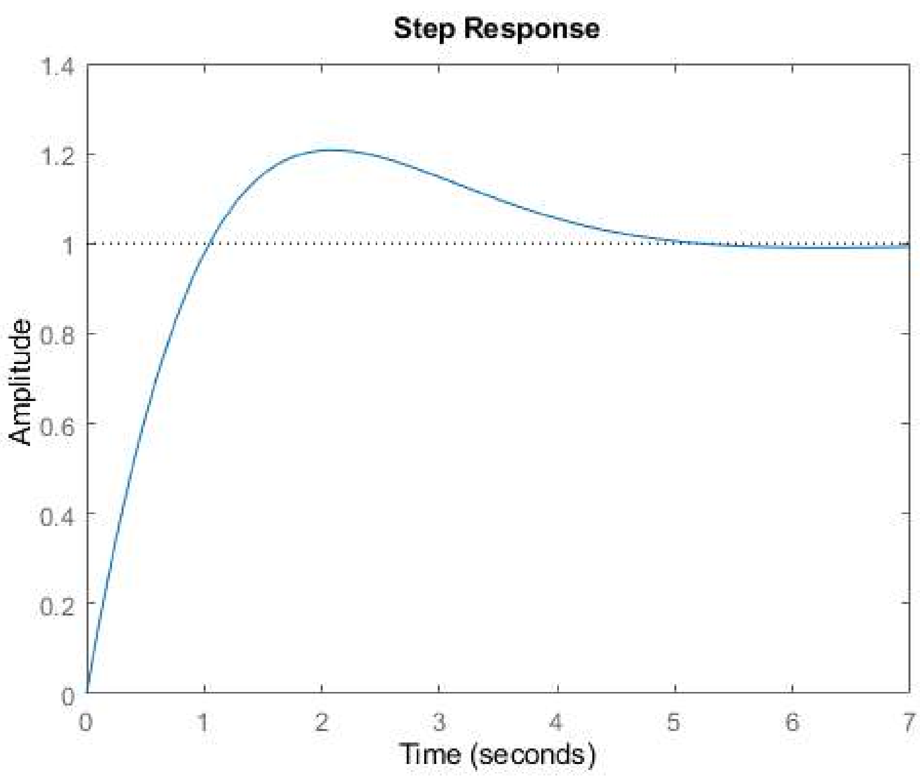
```
CL_alt = minreal(feedback(C1*G1,1))
```

```
CL_alt =

     1.504 s + 1.131
  ---------------------
  s^2 + 1.504 s + 1.131

Continuous-time transfer function.
```

```
stepinfo(CL_alt)
```

```
ans = struct with fields:
        RiseTime: 0.7956
    SettlingTime: 4.6010
     SettlingMin: 0.9331
     SettlingMax: 1.2079
       Overshoot: 20.7910
      Undershoot: 0
            Peak: 1.2079
        PeakTime: 2.0823
```

```
step(CL_alt)
```




```
isstable(CL_alt)
```

```
ans = logical
   1
```

```
% System did not meet specificaltions, redesign iterations began.
% Final Design:

zeta = 0.8;
omega_n = 0.4;
Kp = (3/2)*omega_n^2;
Kd = (3/2)*(2*zeta*omega_n);
C1 = tf([Kd,Kp,0],[1,0])
```

```
C1 =

  0.96 s^2 + 0.24 s
  -----------------
          s
```

```
Continuous-time transfer function.
```

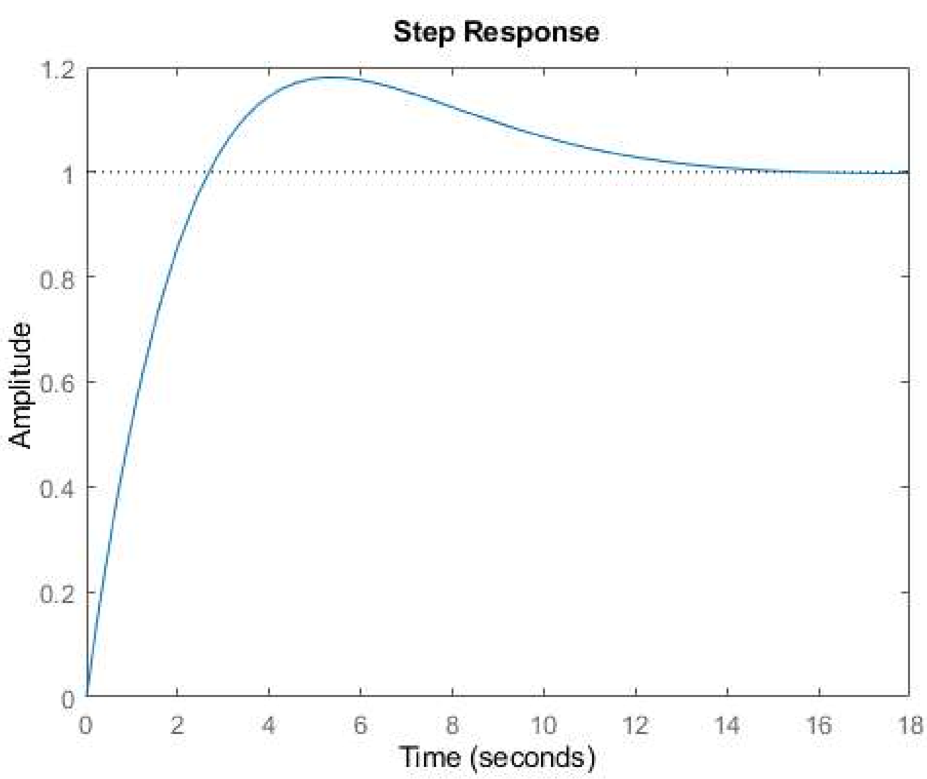
```
CL_alt = minreal(feedback(C1*G1,1))
```

```
CL_alt =

     0.64 s + 0.16
  -------------------
  s^2 + 0.64 s + 0.16

Continuous-time transfer function.
```

```
stepinfo(CL_alt)
```

```
ans = struct with fields:
        RiseTime: 2.0171
    SettlingTime: 12.6344
     SettlingMin: 0.9276
     SettlingMax: 1.1798
       Overshoot: 17.9763
      Undershoot: 0
            Peak: 1.1798
        PeakTime: 5.3247
```

```
step(CL_alt)
```




```
isstable(CL_alt)
```

```
ans = logical
   1
```

Inner Loop Controller Design (Pole Placement)

```
syms Kp Kd Ki
Cp2 = Kp + Kd*s;
CL2 = collect(simplify((G_ThetaT*Cp2)/(1 + G_ThetaT*Cp2)),s)
```

CL2 =

$$\frac{(2500\,\mathrm{Kd})\,s + 2500\,\mathrm{Kp}}{123\,s^2 + (2500\,\mathrm{Kd})\,s + 2500\,\mathrm{Kp}}$$

```
[Num,Den] = numden(G_ThetaT);
G2 = tf(sym2poly(Num),sym2poly(Den))
```

```
G2 =

   2500
  -------
  123 s^2

Continuous-time transfer function.
```

```
t_r = 0.3;
M = 5;

zeta = abs(log(M/100))/sqrt(pi^2 + abs((log(M/100))^2));
omega_n = 1.8/t_r;

Kp = (123/2500)*(omega_n^2);
Kd = (123/2500)*(2*zeta*omega_n);
C2 = tf([Kd,Kp,0],[1,0])
```

```
C2 =

  0.4074 s^2 + 1.771 s
  --------------------
           s

Continuous-time transfer function.
```

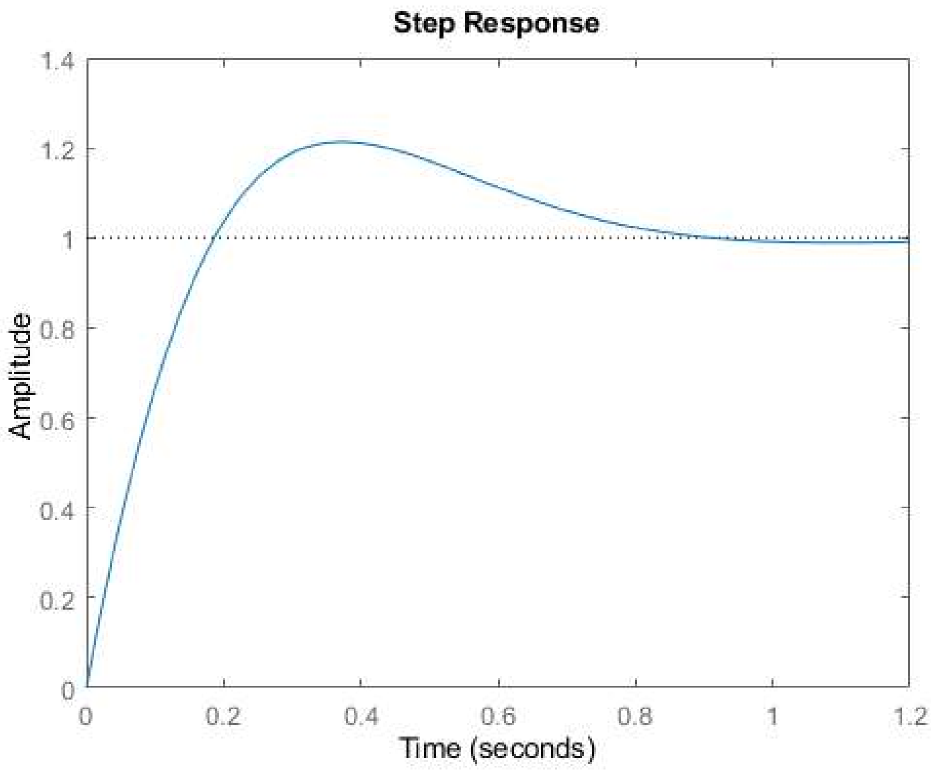
```
CL_roll = minreal(feedback(C2*G2,1))
```

```
CL_roll =

     8.281 s + 36
  ------------------
  s^2 + 8.281 s + 36

Continuous-time transfer function.
```

```
stepinfo(CL_roll)
```

```
ans = struct with fields:
        RiseTime: 0.1423
    SettlingTime: 0.8111
     SettlingMin: 0.9038
     SettlingMax: 1.2136
       Overshoot: 21.3587
      Undershoot: 0
            Peak: 1.2136
        PeakTime: 0.3781
```

```
step(CL_roll)
```




```
isstable(CL_roll)
```

```
ans = logical
   1
```

```
% System did not meet specificaltions, redesign iterations began.
% Final Design:

a = -0.25;
b = -15;

Kp = (a*b)/(2500/123);
Kd = (-a-b)/(2500/123);
C2 = tf([Kd,Kp,0],[1,0])
```

```
C2 =

  0.7503 s^2 + 0.1845 s
  ---------------------
            s

Continuous-time transfer function.
```

```
CL_roll = minreal(feedback(C2*G2,1))
```

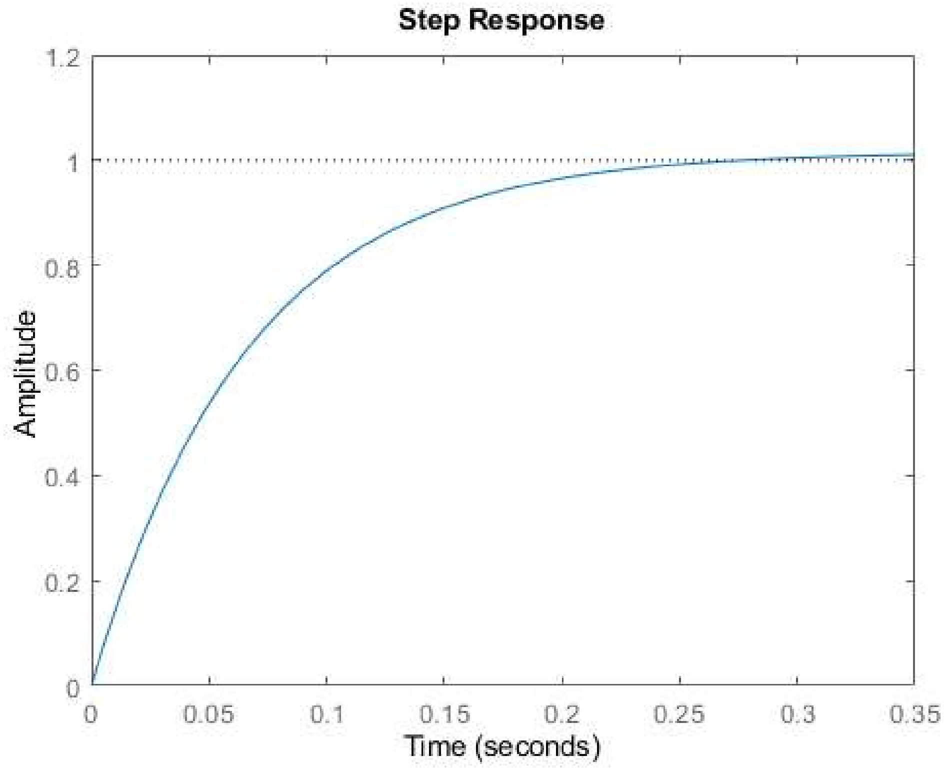
```
CL_roll =

     15.25 s + 3.75
  --------------------
  s^2 + 15.25 s + 3.75
```

```
Continuous-time transfer function.
```

```
stepinfo(CL_roll)
```

```
ans = struct with fields:
        RiseTime: 0.1376
    SettlingTime: 0.2227
     SettlingMin: 0.9048
     SettlingMax: 1.0145
       Overshoot: 1.4507
      Undershoot: 0
            Peak: 1.0145
        PeakTime: 0.5557
```

```
step(CL_roll)
```




```
isstable(CL_roll)
```

```
ans = logical
   1
```

Outer Loop Controller Design (Pole Placement)

```
syms Kp Kd Ki
Cp3 = Kp + Kd*s;
CL3 = collect(simplify((dcgain(CL_roll)*G_ZTheta*Cp1)/(1 + dcgain(CL_roll)*G_ZTheta*Cp1)),s)
```

CL3 =

$$\frac{(2943\ \mathrm{Kd})\ s + 2943\ \mathrm{Kp}}{-300\ s^2 + (2943\ \mathrm{Kd} - 20)\ s + 2943\ \mathrm{Kp}}$$

```
[Num,Den] = numden(G_ZTheta);
```

```
G3 = tf(dcgain(CL_roll)*sym2poly(Num),sym2poly(Den));

t_r = 3;
t_s = 8;

Kp = (-300/2943)*(1.8/t_r)^2;
Kd = (-300/2943)*(8/t_s) + 20/(-2943);
C3 = tf([Kd,Kp,0],[1,0])
```

```
C3 =

  -0.1087 s^2 - 0.0367 s
  ----------------------
            s

Continuous-time transfer function.
```

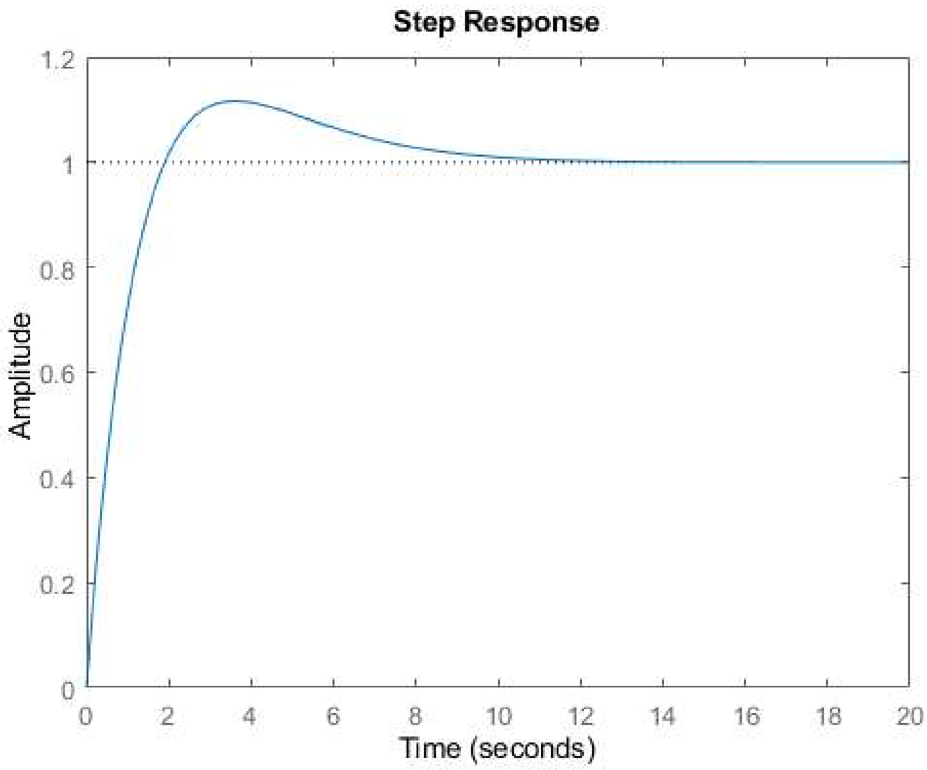
```
CL_lat = minreal(feedback(C3*G3,1))
```

```
CL_lat =

     1.067 s + 0.36
  --------------------
  s^2 + 1.133 s + 0.36

Continuous-time transfer function.
```

```
stepinfo(CL_lat)
```

```
ans = struct with fields:
        RiseTime: 1.3778
    SettlingTime: 8.6187
     SettlingMin: 0.9192
     SettlingMax: 1.1163
       Overshoot: 11.6250
      Undershoot: 0
            Peak: 1.1163
        PeakTime: 3.5758
```

```
step(CL_lat)
```




```
isstable(CL_lat)
```

```
ans = logical
   1
```

```
% System did not meet specificaltions, redesign iterations began.
% Final Design:

Kp = -1;
Kd = -1;
C3 = tf([Kd,Kp,0],[1,0])
```

```
C3 =

  -s^2 - s
  --------
     s

Continuous-time transfer function.
```

```
CL_lat = minreal(feedback(C3*G3,1))-1
```

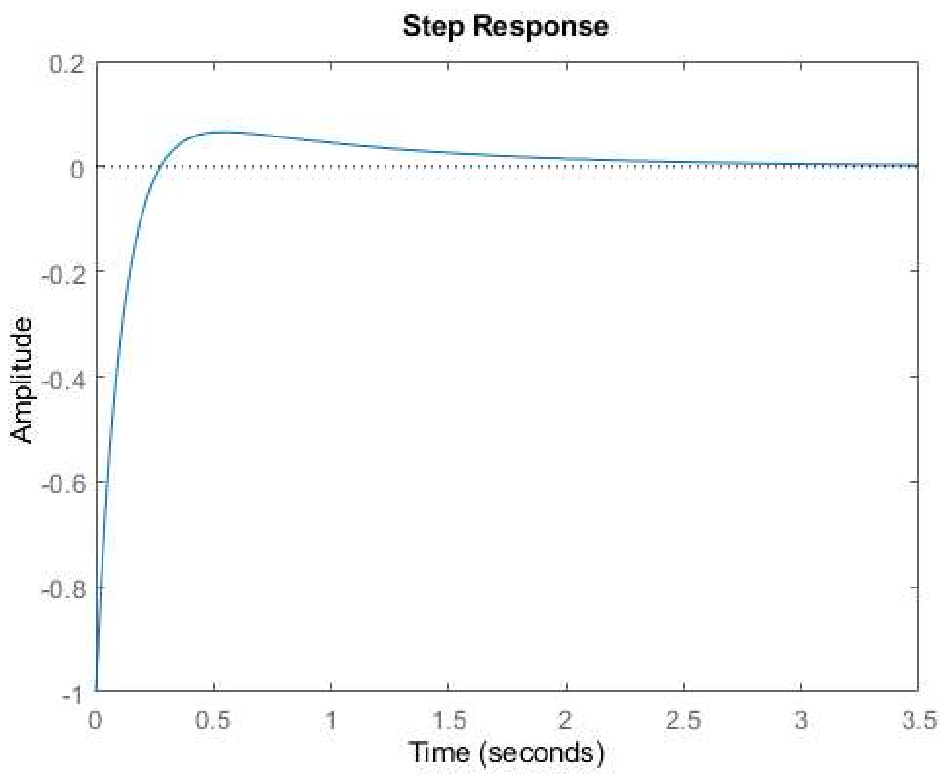
```
CL_lat =

    -s^2 - 0.06667 s
  --------------------
  s^2 + 9.877 s + 9.81

Continuous-time transfer function.
```

```
stepinfo(CL_lat)
```

```
ans = struct with fields:
```

```
        RiseTime: 0.1817
    SettlingTime: 1.7240
     SettlingMin: -0.0875
     SettlingMax: 0.0653
       Overshoot: Inf
      Undershoot: Inf
            Peak: 1
        PeakTime: 0
```

```
step(CL_lat)
```




```
isstable(CL_lat)
```

```
ans = logical
   1
```

Complete (Pole Placement)

```
CL_pos = minreal(feedback(C3*CL_roll*G3,1))
```

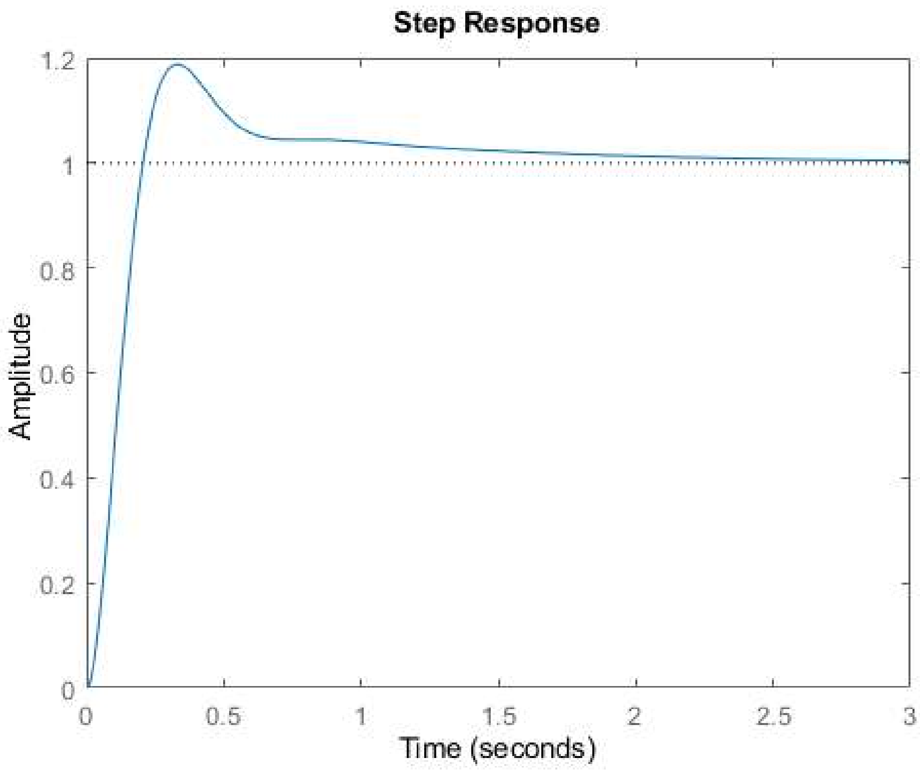
```
CL_pos =

           149.6 s^2 + 186.4 s + 36.79
  ---------------------------------------------
  s^4 + 15.32 s^3 + 154.4 s^2 + 186.6 s + 36.79

Continuous-time transfer function.
```

```
stepinfo(CL_pos)
```

```
ans = struct with fields:
        RiseTime: 0.1405
    SettlingTime: 1.6137
```

```
    SettlingMin: 0.9168
    SettlingMax: 1.1880
      Overshoot: 18.7961
     Undershoot: 0
           Peak: 1.1880
       PeakTime: 0.3298
```

```
step(CL_pos)
```




```
isstable(CL_pos)
```

```
ans = logical
   1
```

Altitude Control (Loop Shaping)

Inner Loop (Loop Shaping)
Outer Loop (Loop Shaping)