```
% SS derivation from: https://www.kth.se/
polopoly_fs/1.588039.1550155544!/Thesis%20KTH%20-%20Francesco
%20Sabatino.pdf
% Some constants from: http://lup.lub.lu.se/luur/download?
func=downloadFile&recordOId=8847641&fileOId=8859343

% x = [phi; = y_angle
%     theta;= z_angle
%     psi;  = x_angle
%     p;    = phi_dot
%     q;    = theta_dot
%     r;    = psi_dot
%     u;    = x_dot
%     v;    = y_dot
%     w;    = z_dot
%     x;
%     y;
%     z];

% Define constants
g = 9.8;            % m/s^2
m = 0.5;            % kg
Ix = 460 * 10^-6;   % N-m-s^2
Iy = 460 * 10^-6;   % N-m-s^2
Iz = 920 * 10^-6;   % N-m-s^2
motor_radius = 0.1; % m
```

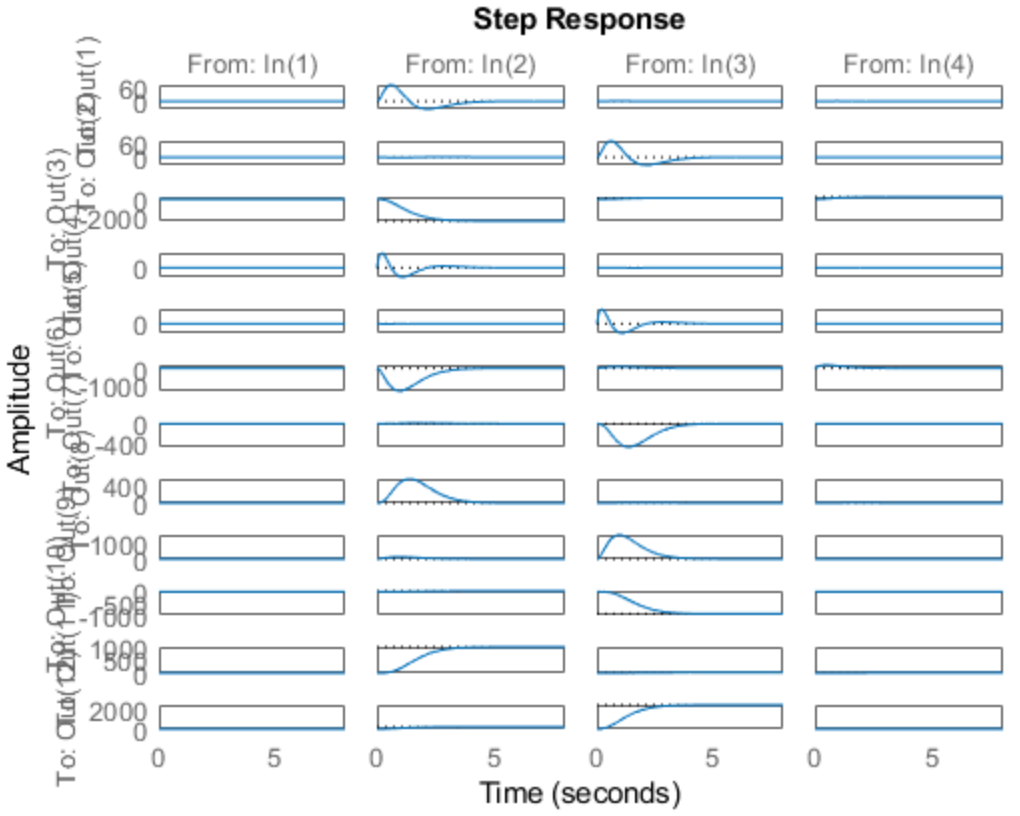
Define state matrices

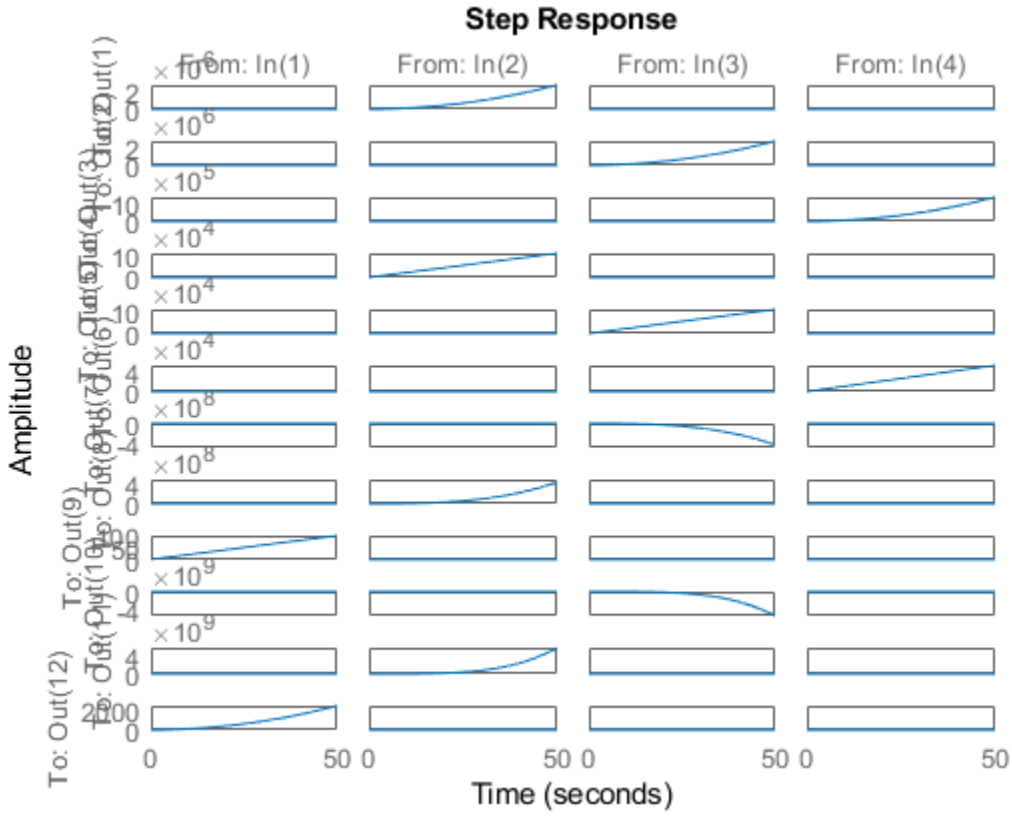
```
A_unstab = [0 0 0 1 0 0 0 0 0 0 0 0;...
    0 0 0 0 1 0 0 0 0 0 0 0;...
    0 0 0 0 0 1 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 -g 0 0 0 0 0 0 0 0 0 0;...
    g 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 1 0 0 0 0 0;...
    0 0 0 0 0 0 0 1 0 0 0 0;...
    0 0 0 0 0 0 0 0 1 0 0 0];

A_unctrl = [0 0 0 1 0 0 0 0 0 0 0 0;...
    0 0 0 0 1 0 0 0 0 0 0 0;...
    0 0 0 0 0 1 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 -g 0 0 0 0 0 0 0 0 0 0;...
    g 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0;...
    0 0 0 0 0 0 1 0 0 0 0 0;...
```

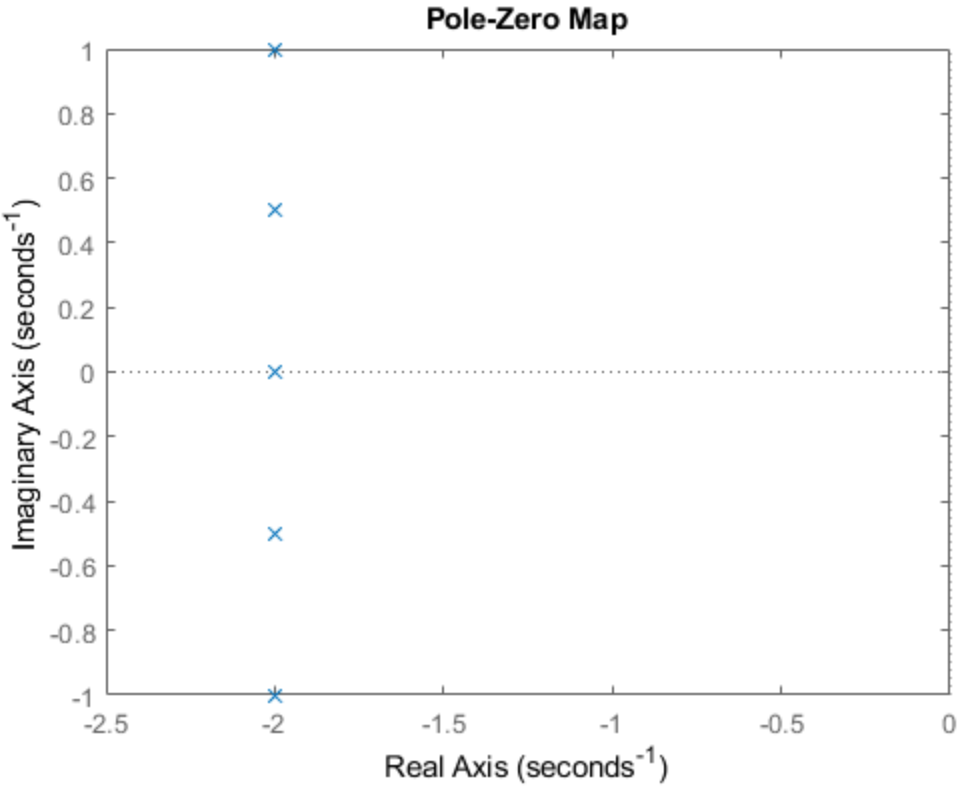
```
    0 0 0 0 0 0 0 1 0 0 0 0 0;...
    0 0 0 0 0 0 0 0 1 0 0 0 0;...
    0 0 0 0 0 0 0 0 0 0 0 0 1];

B_ctrl = [0 0 0 0;...
    0 0 0 0;...
    0 0 0 0;...
    0 1/Ix 0 0;...
    0 0 1/Iy 0;...
    0 0 0 1/Iz;...
    0 0 0 0;...
    0 0 0 0;...
    1/m 0 0 0;...
    0 0 0 0;...
    0 0 0 0;...
    0 0 0 0];

B_unctrl = [0 0 0 0;...
    0 0 0 0;...
    0 0 0 0;...
    0 1/Ix 0 0;...
    0 0 1/Iy 0;...
    0 0 0 1/Iz;...
    0 0 0 0;...
    0 0 0 0;...
    1/m 0 0 0;...
    0 0 0 0;...
    0 0 0 0;...
    0 0 0 0;...
    0 0 0 0];

C = eye(size(A_unstab));
C_unctrl = eye(size(A_unctrl));
% check observability and controllability
% Reference goal rank
fprintf('Controllable system\n');
fprintf('Number of states is: %d\n', size(A_unstab, 1));
fprintf('Rank of Controllability Matrix is: %d\n', rank(ctrb(A_unstab,
 B_ctrl)));
fprintf('Rank of Observability Matrix is: %d\n', rank(obsv(A_unstab,
 C)));

fprintf('\n\nUncontrollable system\n');
fprintf('Number of states is: %d\n', size(A_unctrl, 1));
fprintf('Rank of Controllability Matrix is: %d\n', rank(ctrb(A_unctrl,
 B_unctrl)));
fprintf('Rank of Observability Matrix is: %d\n', rank(obsv(A_unctrl,
 eye(size(A_unctrl)))));

Ctr = ctrb(A_unctrl, B_unctrl);
extra_row = [0 0 0 0 0 0 0 0 0 0 0 0 1]';
Q = horzcat(Ctr(:,1:8), Ctr(:,10:11), Ctr(:,14:15), extra_row);
P = Q^-1;
A_rectrl = P*A_unctrl*(P^-1);
```

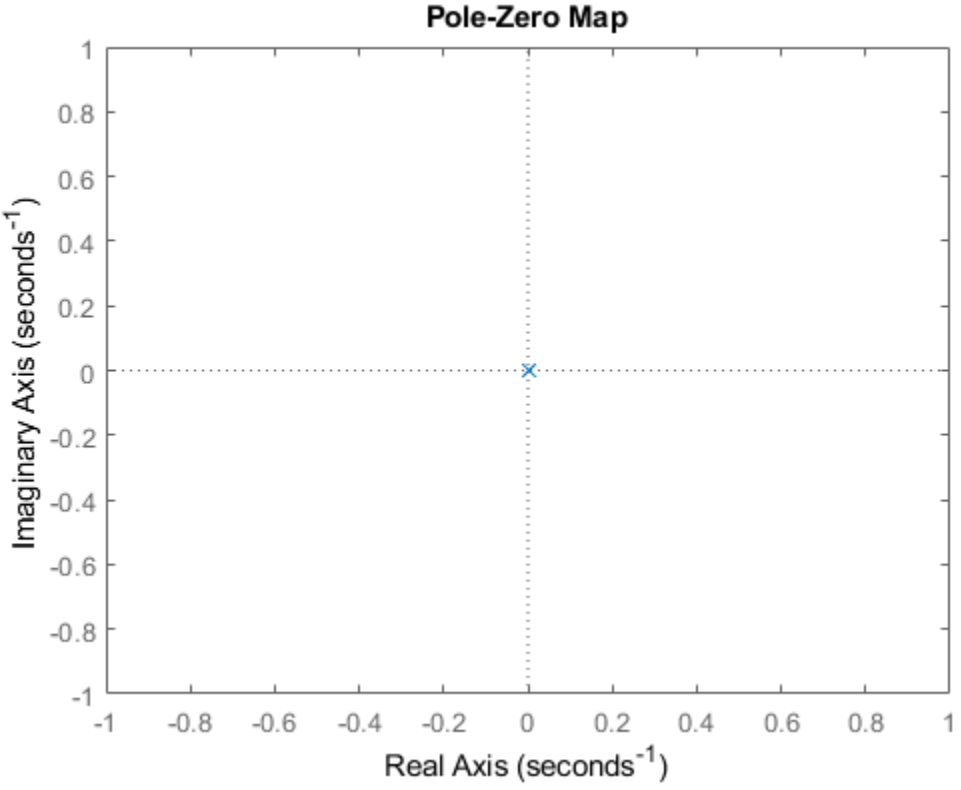
```
B_rectrl = P*B_unctrl;
C_rectrl = C_unctrl*(P^-1);
A_rectrl = A_rectrl(1:12,1:12);
B_rectrl = B_rectrl(1:12,:);
C_rectrl = C_rectrl(:,1:12);

fprintf('\nKalman-decomposed system\n');
fprintf('Number of states is: %d\n', size(A_rectrl, 1));
fprintf('Rank of Controllability Matrix is: %d\n', rank(ctrb(A_rectrl,
 B_rectrl)));
fprintf('Rank of Observability Matrix is: %d\n', rank(obsv(A_rectrl,
 C_rectrl)));

% Initialization
dt = 0.1;
k = 1;
time = 10;
steps = round(time / dt);
x = zeros(size(A_unstab, 1), steps);
x_dot = zeros(size(A_unstab, 1), steps);
x(:,k) = [1; 1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 10];

x_uns = zeros(size(A_unstab, 1), steps);
x_dot_uns = zeros(size(A_unstab, 1), steps);
x_uns(:,k) = [0.1; 0.1; 0; 0; 0; 0; 0; 0; 0; 0; 0; 10];

goal = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 10];

p_r = -2;
p_i1 = 0.5i;
p_i2 = 1i;
p = [p_r, p_r, p_r, p_r, p_r+p_i1, p_r-p_i1, p_r+p_i1, p_r-p_i1,...
    p_r+p_i2, p_r-p_i2, p_r+p_i2, p_r-p_i2];
k_ctrl = place(A_unstab, B_ctrl, p);
A_stab = (A_unstab - B_ctrl * k_ctrl);

fprintf('Eigenvalues of systems\n');
fprintf('Unstabilized system Eigenvalues: ');
eig(A_unstab)
fprintf('\nStabilized system Eigenvalues: ');
eig(A_stab)

sys_cs = ss(A_stab, B_ctrl, C, 0);
sys_cu = ss(A_unstab, B_ctrl, C, 0);

figure();
step(sys_cs)
figure();
step(sys_cu)

figure();
pzmap(sys_cs)
figure();
pzmap(sys_cu)
```

```
Controllable system
Number of states is: 12
Rank of Controllability Matrix is: 12
Rank of Observability Matrix is: 12


Uncontrollable system
Number of states is: 13
Rank of Controllability Matrix is: 12
Rank of Observability Matrix is: 13

Kalman-decomposed system
Number of states is: 12
Rank of Controllability Matrix is: 12
Rank of Observability Matrix is: 12
Eigenvalues of systems
Unstabilized system Eigenvalues:
ans =

     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0
     0


Stabilized system Eigenvalues:
ans =

  -2.0000 + 1.0000i
  -2.0000 - 1.0000i
  -2.0000 + 1.0000i
  -2.0000 - 1.0000i
  -2.0000 + 0.5000i
  -2.0000 - 0.5000i
  -2.0000 + 0.5000i
  -2.0000 - 0.5000i
  -2.0000 + 0.0000i
  -2.0000 + 0.0000i
  -2.0000 + 0.0000i
  -2.0000 - 0.0000i
```


Step Response
From: In(1)
From: In(2)
From: In(3)
From: In(4)
Amplitude
Time (seconds)


Step Response
From: In(1)
From: In(2)
From: In(3)
From: In(4)
Amplitude
Time (seconds)


Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-2.5
-2
-1.5
-1
-0.5
0


Pole-Zero Map
Imaginary Axis (seconds^{-1})
Real Axis (seconds^{-1})
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

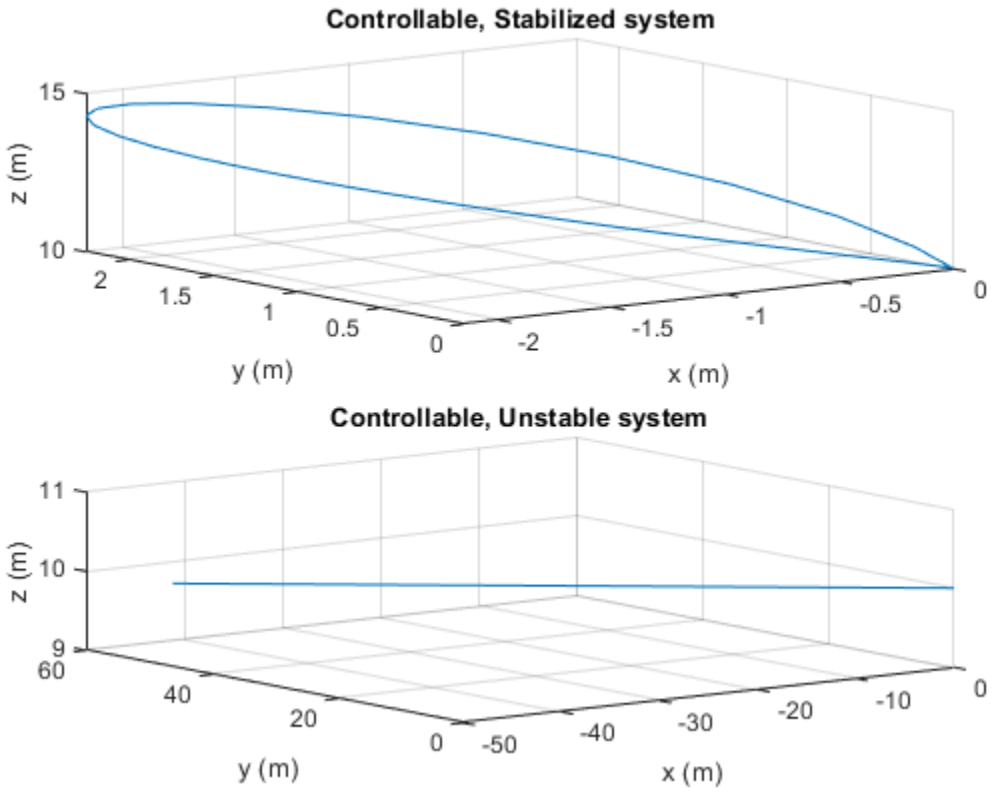
Thrusts from each motor, [FL, FR, BR, BL] in N Assuming motor directions are: [CW, CCW, CW, CCW] (From the top)

```
y_CW = [1 0 1 0];
y_CCW = [0 1 0 1];
r_right = [1 0 0 1];
r_left = [0 1 1 0];
p_fwd = [0 0 1 1];
p_back = [1 1 0 0];
u_rotors = [1, 1, 1, 1];

% Total motor force
f_m = sum(u_rotors);
% Torque about x (Pitch)
t_x = sum(u_rotors.*p_fwd - u_rotors.*p_back);
% Torque about y (Roll)
t_y = sum(u_rotors.*r_right - u_rotors.*r_left);
% Torque about z (Yaw)
t_z = sum(u_rotors.*y_CW - u_rotors.*y_CCW);

u_val = 0.00001;
% [f_sum, f_y, -f_x, yaw]
u = [0; 0; u_val; 0];

% A is not invertible, so this method fails
%A_dt = expm(A*dt);
%B_dt = A^-1*(A_dt-eye(size(A)))*B;
%C_dt = C;

for k = 1:steps

    u(:,k) = -k_ctrl * (x(:,k) - goal);
    x_dot(:,k) = A_unstab*x(:,k) + B_ctrl*u(:,k);
    x(:,k+1) = x_dot(:,k) * dt + x(:,k);

    x_dot_uns(:,k) = A_unstab*x_uns(:,k);
    x_uns(:,k+1) = x_dot_uns(:,k) * dt + x_uns(:,k);
end
```

Plotting

```
plt_x = x(10,:);
plt_y = x(11,:);
plt_z = x(12,:);

plt_x_uns = x_uns(10,:);
plt_y_uns = x_uns(11,:);
plt_z_uns = x_uns(12,:);

figure();

% Plot Stab system
subplot(211);
plot3(plt_x, plt_y, plt_z);
```

```
title('Controllable, Stabilized system');
xlabel('x (m)');
ylabel('y (m)');
zlabel('z (m)');
grid();

% Plot Unstab system
subplot(212);
plot3(plt_x_uns, plt_y_uns, plt_z_uns);
title('Controllable, Unstable system');
xlabel('x (m)');
ylabel('y (m)');
zlabel('z (m)');
grid();
```



Published with MATLAB® R2020a