2MAE404 MIMO control
Homework report 1

Maciej Michałów   Marco Wijaya

October 12, 2022

1 Introduction

The goal of the assignment is the implementation of the equations of motion of the Flying Chardonnay drone, in preparation for the work on developing control for it.

2 Equations of motion implementation

The equations of motion are fully derived in the assignment. The MATLAB implementation of the right hand side function is therefore very straightforward:

```
function x_dot = drone_dynamics(x,u,w,drone)

%% Input parameters
% x = [vn; vd; theta; thetad; gamma; gammad; F] - state vector
% u = [T1; T2] - input vector
% w = [wn; wd] - wind velocity vector
% drone - physical parameters of the drone

%% Drone parameters
md = drone.mass; % mass of drone
mc = drone.cupmass; % mass of cup
l = drone.l; % length of rod
ld = drone.ld; % distance to propeller
J = drone.J; % drone moment of inertia
CD = drone.CD; % drone drag coefficient
g = 10; % gravity

%% Dynamics model
A = [md, 0, 0, 0, 0, 0, sin(x(3)+x(5));
     0, md, 0, 0, 0, 0, cos(x(3)+x(5));
     mc, 0, 0, -mc*l*cos(x(3)+x(5)), 0, -mc*l*cos(x(3)+x(5)), -sin(x(3)+x(5));
     0, mc, 0, mc*l*sin(x(3)+x(5)), 0, mc*l*sin(x(3)+x(5)), -cos(x(3)+x(5));
     0, 0, 0, J, 0, 0, 0;
     0, 0, 1, 0, 0, 0, 0;
     0, 0, 0, 0, 1, 0, 0];
```

```
b = [-(u(1)+u(2))*sin(x(3))-CD*(x(1)-w(1));
     md*g-(u(1)+u(2))*cos(x(3))-CD*(x(2)-w(2));
     -mc*l*(x(4)+x(6))^2*sin(x(3)+x(5));
     mc*g-mc*l*(x(4)+x(6))^2*cos(x(3)+x(5));
     (u(2)-u(1))*ld;
     x(4);
     x(6)];

x_dot = A\b;
x_dot = x_dot(1:6); % the 7th component is F, which is not used
end
```

3 Use of the function

The values of the constants are also taken from the report:

```
drone.mass = 1; % mass of drone, md = 1 kg
drone.cupmass = 1; % mass of cup, mc = 1 kg
drone.l = 1; % length of rod, l = 1 m
drone.ld = 1; % distance to propeller, ld = 1 m
drone.J = 1; % drone moment of inertia, J = 1 kg*m^2
drone.CD = 1; % drone drag coefficient, CD = 0.01
```

As described in the assignment, the value of $\dot{x}$ can be computed for the specified values of x, u and w:

```
x = [1; 0.1; deg2rad(10); deg2rad(10); deg2rad(5); deg2rad(5)];
u = [4.8; 5.3];
w = [2; -3];
x_dot = drone_dynamics(x,u,w,drone)
```

The computed value is:

$$\dot{x} = [0.8934, 3.1009, 0.1745, 0.5000, 0.0873, 2.1485]^T$$

4 Conclusions

The equations of motion seem to be correctly implemented. The need of matrix inversion every step can look like an unnecessary burden on computational power, but, according to the MATLAB code profiler, it takes less than 1 [ms] to compute, so it is unlikely to be a serious bottleneck.