Preperation Report-AERO simulation

Tom Shahar

Omer Cohen

Part 1:

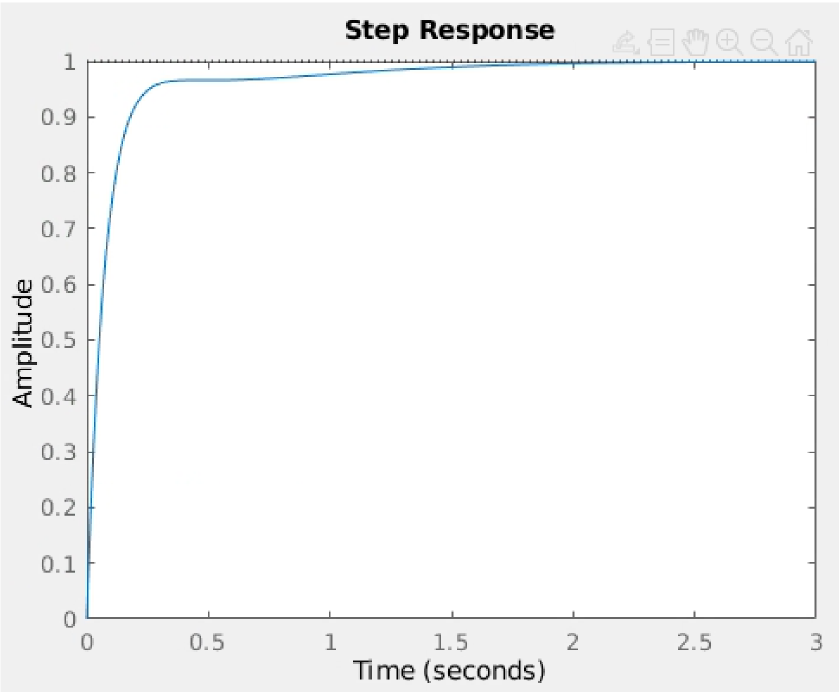
Theoretical backround:

1. Once we have the transfer function of the step response, we can use the following connection , assuming Y(s) is ut step response
$$\mathrm{U(s)} = \frac{1}{\mathrm{s}}\ , \mathrm{Y(s)} = \mathrm{U(s)} * \mathrm{G(s)} \Rightarrow \mathrm{G(s)} = \frac{\mathrm{Y(s)}}{\frac{1}{s}} = \mathrm{sY(s)}$$
2. We can see out transfer function in the following form :
$$\mathrm{G(s)} = \frac{\mathrm{K}}{s^2 + 2\zeta\omega_c s + \omega_c^2}$$
So once we have two of $[\zeta, \omega_c, K]$ we could find the transfer function, we will use the following connection between the rise time and peak time, rise time is defined as the time it takes for the response to rise from 10% to 90% of its final value, and the peak time is the time it takes for the response to reach its first peak.
We used the following connection we found in previous courses material :
$$\mathrm{M_P} = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}}$$
$$\ln\,(\mathrm{M_P}) = -\frac{\zeta\pi}{\sqrt{1-\zeta^2}}$$
$$\ln(\mathrm{M_P})^2\,(1-\zeta^2) = -\zeta^2\pi$$
$$\ln(\mathrm{M_P})^2 = \zeta^2(\ln(\mathrm{M_P})^2 - \pi)$$
$$\zeta = \frac{\ln(M_P)}{\sqrt{\ln(\mathrm{M_P})^2 - \pi}}$$
And for ω_c we can measure tp and use the following connection:
$$\omega_c =$$
And once the unit step is given foe us, we can measure what is the gain K and set in in the formula above.

3. In order to fing the PID parameters, I used the following matlab code (with chat GPT help) and used the PID tuner app, and modify the controller K's values so it meet the dementions, the final Ki,Kp,Kd are in the code as well :

```matlab
% Define the open-loop transfer function
close all;
Gs = tf([1],[1 5 10]);

% Define the desired step response characteristics
desired_overshoot = 0; % Zero overshoot
desired_ss_error = 0; % Zero steady-state error
desired_settling_time = 0.25; % Settling time less than 0.25 sec

% Design the PID controller using the PID tuner app in MATLAB
pidTuner(Gs, 'pid');

% Alternatively, define the PID controller gains manually
Kp = 68;
Ki = 89.8;
Kd = 13.52;
Cs = pid(Kp, Ki, Kd);

% Create the closed-loop transfer function with the PID controller
T = feedback(Cs*Gs, 1);

% Compute the step response characteristics of the closed-loop system
step_info = stepinfo(T);

% Check if the desired performance criteria are met
if (step_info.Overshoot <= desired_overshoot) && ...
   (step_info.SettlingTime <= desired_settling_time) && ...
   (step_info.SteadyStateError <= desired_ss_error)
    disp('Controller design successful!');
else
    disp('Controller design failed to meet desired performance criteria.');
end
figure(18);
step(T);
hold on;
```

We received the following step response :



Parameters: Kp =68 , Ki = 89.8 , Kd =13.52

Part 2:

4. Based on the Flight dynamics block output there are 3 state variables:

   The first variable is Pitch. The pitch represents the motors axis angle about the horizon. This variable will go into a PID controller that will monitor the angle's change and will try to minimize it.
   The second variable is the X position. This variable like the previous one will also go into a PID controller that will try to minimize the position's change.
   The third variable is the Z position which represents the altitude of the drone. A PID controller will receive input of the target altitude, in addition to the actual altitude (Z position). The controller will try to minimize the deviation of the actual altitude from the given target.

5. Sway PID controller:
   Inputs:
   - Current X position.
   - X position's target.

   Output:

   - Required pitch to minimize the position's deviation.

   Pitch PID controller:

   Inputs:

   - Current pitch.
   - Pitch limit – output of the sway controller.

   Output:

   - Required voltage for the motors to minimize the gap between the inputs.

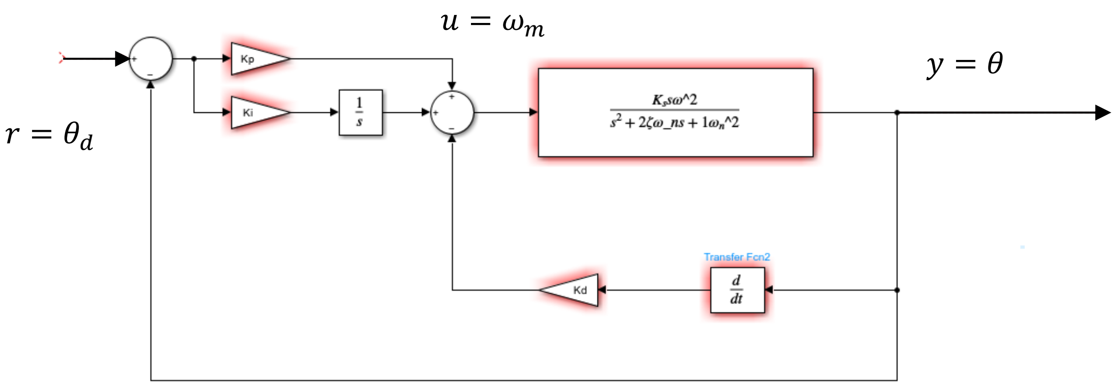
   Throttle PID controller:

   Inputs:
   - Current altitude (Z position).
   - Target altitude (Z target).

   Output:

   - Required voltage for the motors to minimize the gap between the inputs.

6. We will implement a pitch PID controller:



As we can see, in the inner loop the differential amplifier is parallel to the 2'nd order plant. And in theouter loop we can see that the PI controller is cascaded to the inner loop.

7. If the new system is combined from 4 motors the new state variables will be:
X position – same as before for the 2d case.
Z position – altitude.
Y position – a new variable that represents the position on the Y axis.
Pitch x & pitch y – now we have two options for changing the angle about the horizon. We will assign one angle for the perspective of the X axis and another angle for the Y axis.