3F1 Flight Control Report

Abhishek Shenoy (ams289)
Pembroke

31/10/2020

1 Introduction

In this report, we investigate the use of controllers to stabilise different flight plant models. More specifically we investigate the following controllers for different aircraft models:

- Manual control of the aircraft
- Pilot induced oscillation
- Sinusoidal disturbances
- Unstable aircraft model
- Proportional feedback controller
- PID controller

The general block diagram of the whole investigation is shown in Fig. 1.

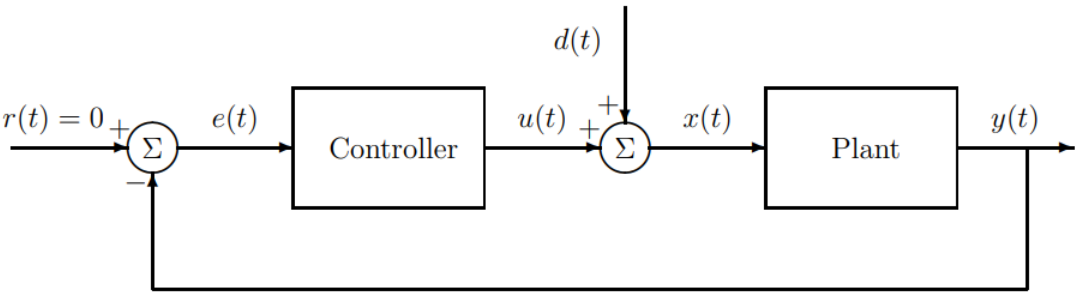


Figure 1: System Block Diagram

2 Manual Control

We consider a simple aircraft model differential equation for the basis of this section.

$$\ddot{y}(t) + M\dot{y}(t) = Nx(t) \tag{1}$$

Taking the Laplace transform of the model equation:

$$G(s) = \frac{N}{s^2 + Ms} \tag{2}$$

Now running multiple manual tests, we come up with a suitable manual controller response that is analogous to the model of proportional gain k in series with a pure time delay D whose Laplace transfer function is therefore given by Eq. 3.

$$K(s) = ke^{-Ds} \tag{3}$$

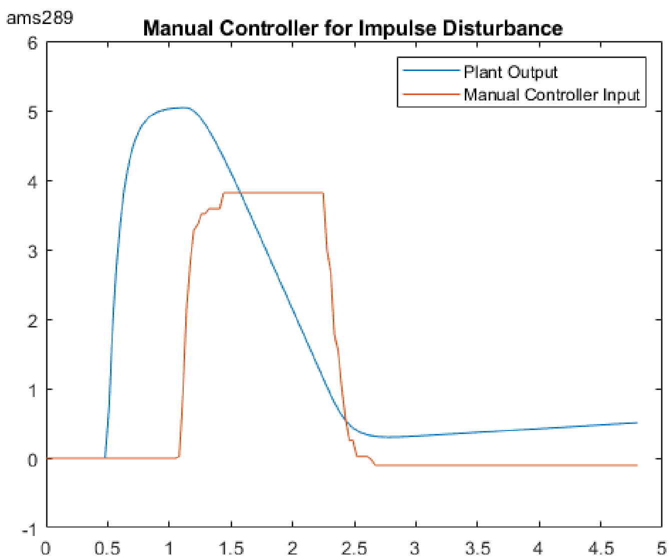


Figure 2: Tested response of manual controller for impulse disturbance

From Fig. 2, we can determine the delay to be 0.6s with a gain of 0.76.

We can now plot a Bode diagram of the model manual controller as shown in Fig. 3.

We can calculate the phase margin by finding the phase corresponding to 0dB gain and measure its distance to 180 degrees. From Fig. 3, we can see that the phase margin $\phi \approx 58\,\text{deg}$.

We can now also calculate the extra time delay the control loop can tolerate before becoming unstable.

$$t_{\text{tolerance}} = \frac{\phi}{\omega} = \frac{58\frac{\pi}{180}}{0.759} = \frac{1.02}{0.759} = 1.34s$$

The open-loop Nyquist plot is shown in Fig. 4.

Now considering a step disturbance of magnitude 5, a run time of 10 seconds, we try to keep the plane level and the response is shown in Fig. 5.

We can see that the behaviour is indeed that of an integral action as the gain rises steadily to compensating for the large increase in error due to the use of a step disturbance instead of an impulse disturbance.

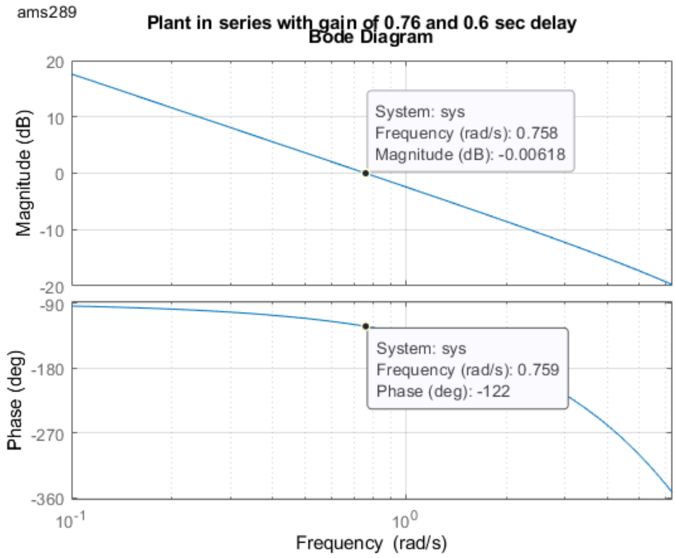


Figure 3: Bode diagram of model transfer function

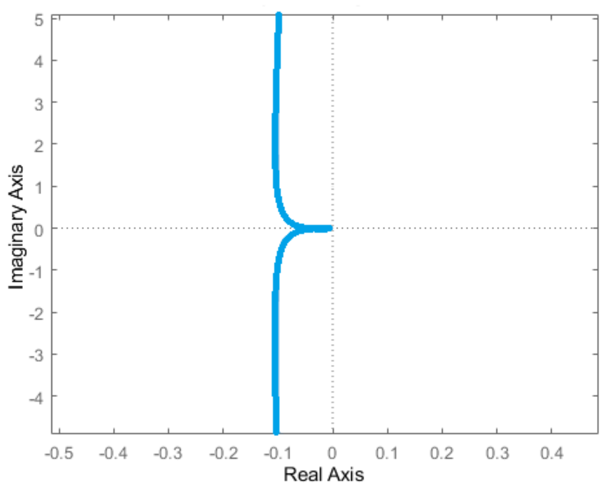


Figure 4: Sketched Nyquist diagram of model transfer function

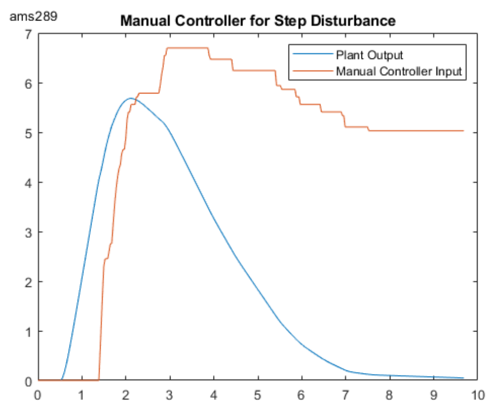


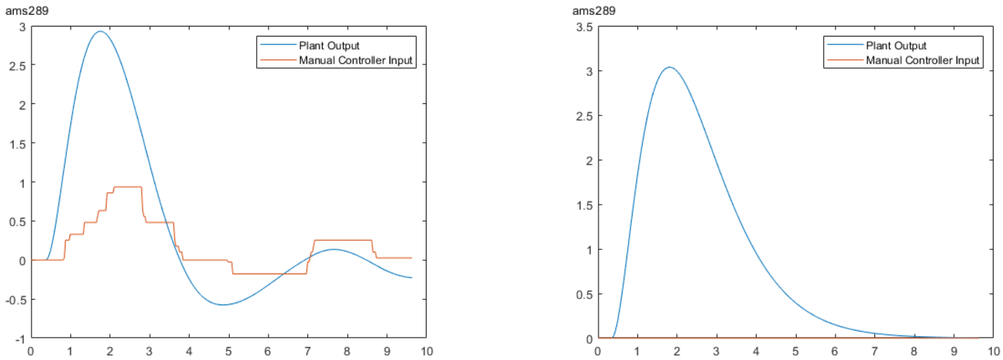
Figure 5: Tested response of manual controller for step disturbance

2.1 Pilot Induced Oscillation

A poorly designed controller has the transfer function given by Eq. 4.

$$G_1(s) = \frac{c}{(Ts+1)^3} \tag{4}$$

We choose the parameters as follows $c = \frac{\sqrt{8}}{k}$ and $T = \frac{4D}{\pi}$ and we take an impulse of weight $5kD$ and attempt to bring the plane back to level flight with $k = 1.2$ and $D = 0.6$:



(a) Typical plant PIO response

(b) Zero input plant response

Figure 6: Flight system responses

The bode plot of the open-loop is shown in Fig. 7.

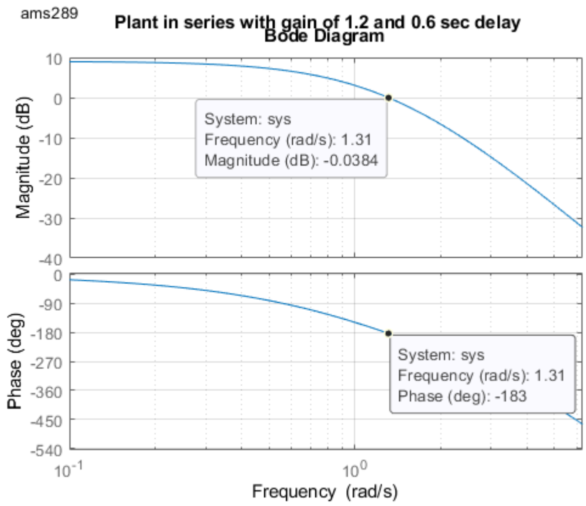


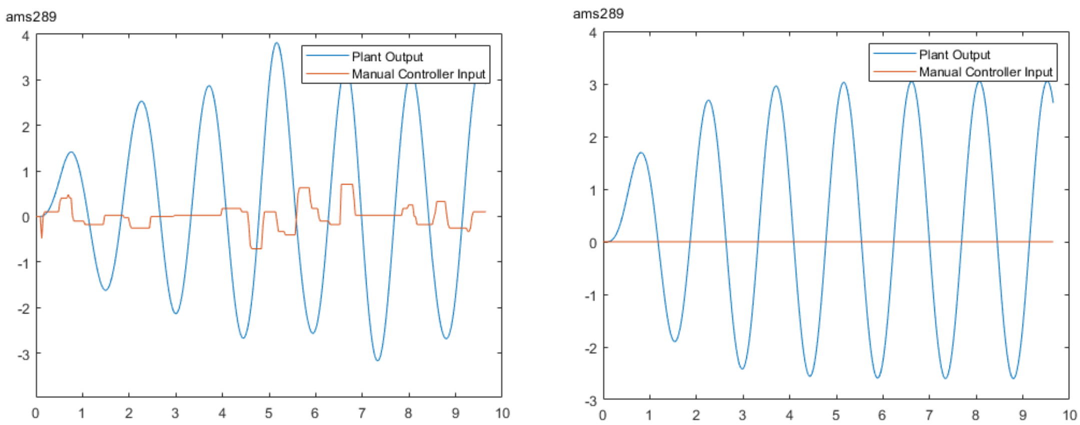
Figure 7: Bode plot of PIO system

We can see from Fig. 6a that the observed period of oscillation does not match the theoretical prediction of $T_{\text{pred}} = \frac{4D}{\pi} \approx 0.76$ whereas the actual period in the graph is much larger at $T_{\text{actual}} = 6.2s$. From Fig. 7, we can see that the phase margin is 3 degrees which is very close to 0. We can see that the aircraft is naturally stable, but the pilot's intention to stabilize the aircraft actually causes the oscillatory behaviour hence called "pilot induced oscillation". A more suitable controller is one that has a large phase margin to prevent this oscillatory behaviour.

2.2 Sinusoidal Disturbances

Considering a more realistic model of F4E fighter aircraft, we consider a specific linearised system about a specific operating point. This model is based on an altitude of 35000 ft and Mach 1.5.

Considering a sinusoidal disturbance of magnitude 1 and a frequency of 0.66 Hz, the responses for a typical input and zero input is shown in Fig. 8a and Fig. 8b respectively.



(a) Typical plant response for sinusoidal disturbance

(b) Zero input plant response for sinusoidal disturbance

Figure 8: Flight system responses

From Fig. 10, it is possible to see that the error cannot be reduced. The bode plot of the open-loop is shown in Fig. 9.

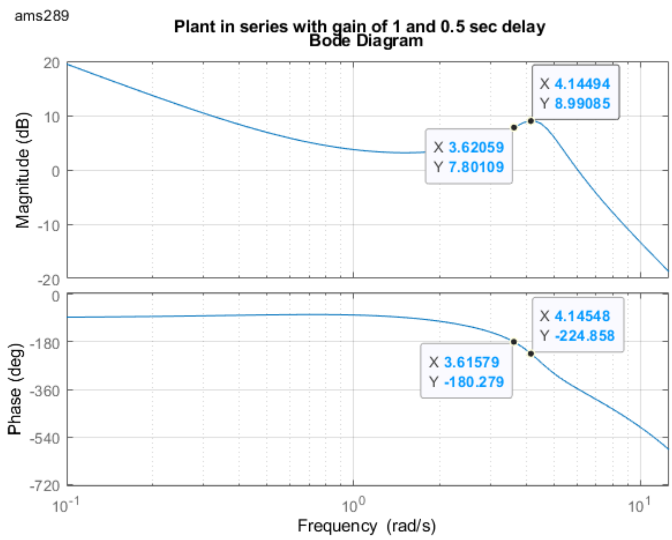


Figure 9: Bode plot of model system

From Fig. 9, we can see that the maximum proportional gain for which the closed loop system is stable (ie. the gain margin) is -7.8 dB. The gain and phase at $0.66Hz$ ($4.145\ rad\ s^{-1}$) is 8.99 dB and -225 degrees respectively.

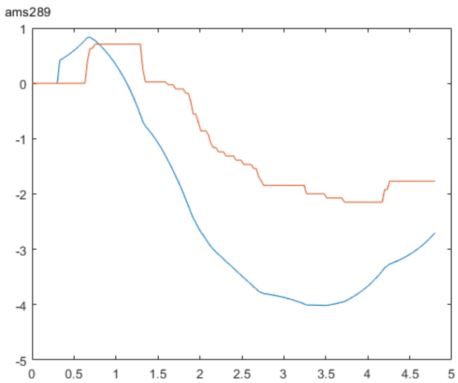
For a plant transfer function G and a controller transfer function K, the open loop transfer function from d to y is simply G as the reference signal r is 0 and the closed loop transfer function is $L = \dfrac{G}{1+GK}$.

2.3 An Unstable Aircraft

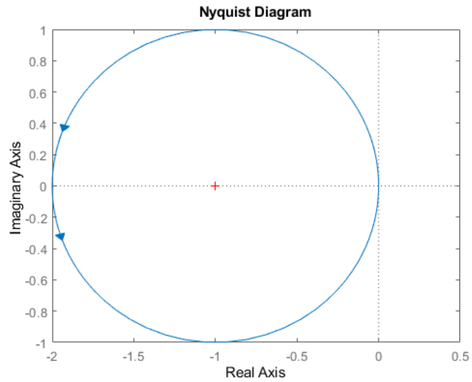
We now analyse the behaviour of an unstable aircraft by considering the following model:

$$G(s) = \frac{2}{sT - 1} \tag{5}$$

We experiment with a range of values for T and decide to use T=0.5 for manual control. The response is shown in Fig. 10a and the Nyquist diagram is shown in Fig. 10b.



(a) Response of unstable system to manual control



(b) Nyquist diagram of unstable system

Figure 10: Flight system responses

The Nyquist diagram of the plant shown in Fig. 10b has a circle with an anti-clockwise encirclement of -1 and therefore the plant itself is unstable. The Nyquist diagram touches the x-axis at -2 and hence a proportional gain greater than 0.5 will be able to stabilise this aircraft. Even if a time delay D is introduced into the system, the minimum proportional gain required does not change. There would however be more encirclements around the origin although this does not affect the stability.

3 Autopilot

Now we consider the following model for a transport aircraft on approach to landing:

$$G_3(s = \frac{6.3s^2 + 4.3s + 0.28}{s^5 + 11.2s^4 + 19.6s^3 + 16.2s^2 + 0.91s + 0.27} \tag{6}$$

Using a negative feedback proportional controller, we are able to stabilise the aircraft. We then change the value of K_c to induce a steady oscillation which we find to be $K_c = 17.6$. The period of this oscillation is given by $T_c = 1.86s$. The oscillatory response is shown in Fig. 11.

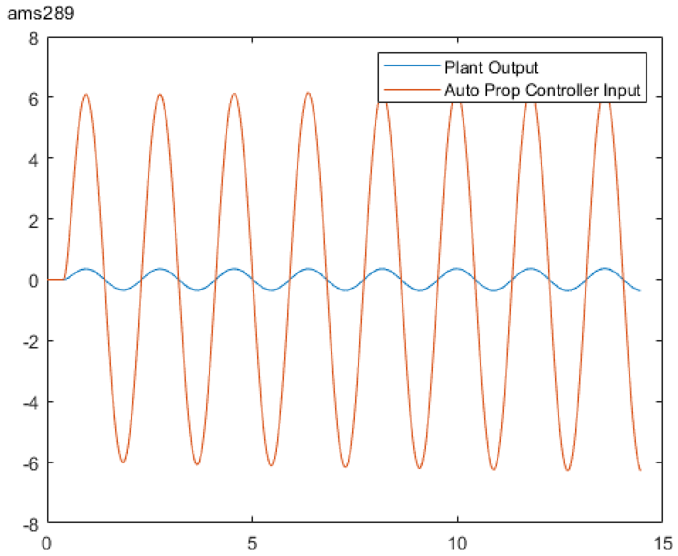


Figure 11: Autopilot oscillatory control

3.1 PID Controller

The autopilot is to be implemented with a proportional-integral-derivative (PID) controller of the form:

$$u(t) = K_p \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de}{dt} \right) \tag{7}$$

The transfer function of this controller is therefore:

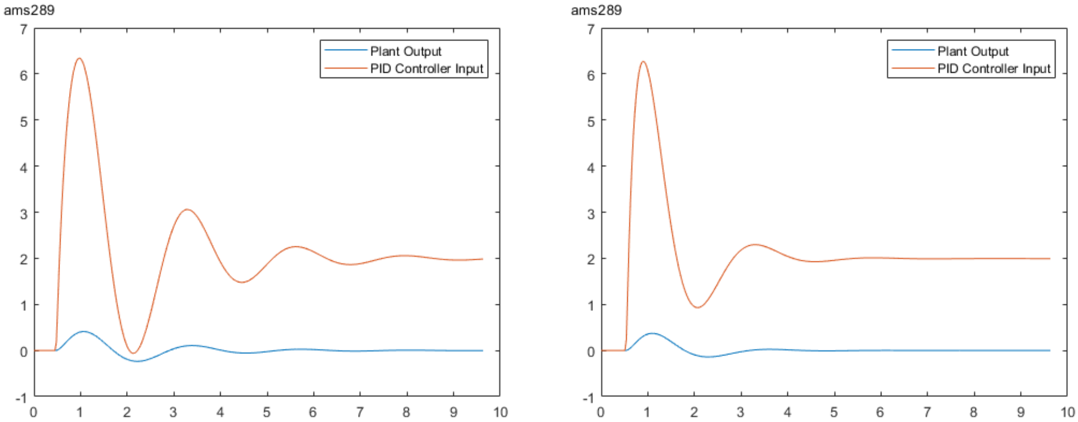
$$K(s) = \frac{K_p(1 + sT_i + s^2 T_i T_d)}{sT_i} \tag{8}$$

The constants are selected as follows based on the Ziegler-Nichols rules in process control.

$$K_p = 0.6K_c, \quad T_i = 0.5T_c, \quad T_d = 0.125T_c \tag{9}$$

3.2 Integrator Wind-up

A phenomena called integrator wind-up can cause unwanted overshoot/undershoot after a period of input saturation due to the integrator being 'wound up' and having to unload



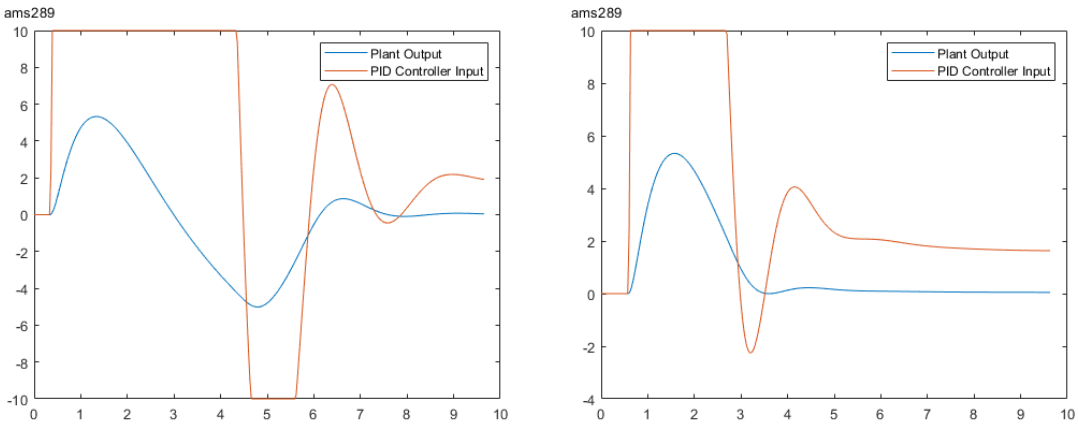
(a) Default PID Controller (b) PID Controller with 40% gain on derivative

Figure 12: PID Controller Responses

again. One simple corrective remedy is to prevent the integrator state from ever getting too large.

In Section 3.1, the impulse input is set to 2 whereas in this section the impulse input is set to 20 with the step disturbance at the same value of 2. Hence to compensate for the extra impulse to prevent integrator wind-up, we can estimate the value to be around 0.1.

More accurately we can derive this from the fact that at zero steady-state error $u(t) = d(t)$ where $u(t)$ is the controller equation given in Eq. 7. For zero steady-state error, only the integral term is non-zero and from this we can find that the number we need to set is equal to $\dfrac{T_i}{K_p} d(t)$ where $d(t)$ is the step disturbance of weight 2. Therefore for the code provided, the smallest number which guarantees a zero steady-state error in response to a step disturbance of weight 2 is $2\dfrac{T_i}{K_p} \approx 0.176$.



(a) Integrator Wind-Up (b) Integrator Wind-Up Resolved

Figure 13: PID Controller integrator wind-up testing

Worksheet

2. Simplified aircraft model. Transfer function $= G(s) = \dfrac{N}{s^2 + Ms}$

`num` = N `den` = [1 M 0]

where N = M = 10

Controller transfer function $= ke^{-Ds}$

$k = 0.76$ $D = 0.6$

Phase margin = 58 degrees

Amount of extra time delay which can be tolerated = 1.34 seconds

2.1. PIO. Period of oscillation (observed) = 6.2

Period of oscillation (theoretical) = 0.76

2.2. Sinusoidal disturbances.

Maximum stabilising gain $= -7.8db$

Gain at 0.66 Hz $= 8.99dB$ Phase at 0.66 Hz $= -225$ degrees

Open loop T.F. $(y \to d) = G$ Closed loop T.F. $(y \to d) = \dfrac{G}{1 + GK}$

2.3. Fastest pole. $T = 0.5$

3. Autopilot. Proportional gain $K_c = 17.6$

Period of oscillation $T_c = 1.86s$

3.1 Transfer function of PID controller $= \dfrac{K_p(1 + sT_i + s^2T_iT_d)}{sT_i}$

PID constants: $K_p = 0.6 * 17.6 = 10.56$ $T_i = 0.93$ $T_d = 0.2325$

Final value of $T_d = 0.3255$

4 MATLAB Code

```
num=[6.3 4.3 0.28]; den=[1 11.2 19.6 16.2 0.91 0.27];   % Plant TF
runtime=10;      % target simulation interval in seconds
wght=[20,2,0,0];         % impulse, step and sinusoidal gain and frequency

samper=30;       % target sampling period in milliseconds

srate=(samper+1.3)/1000;         % anticipated average sampling period
grphc1

Kc = 17.6; Tc = 1.86;
integ=0;deriv=0;yprev=0;
Kp=0.6*Kc; Ti=0.5*Tc; Td=0.125*Tc;

Td = Td * 1.4;
for i=1:count
        set(hh,'Xdata',hx,'Ydata',hy+y*hz);

    integ=integ + y*srate;
    integ=sign(integ)*min(abs(integ),0.176);
    deriv=(y-yprev)/srate;
    pp=-Kp*(y+integ/Ti+deriv*Td);
    yprev=y;

        pp=sign(pp)*min(max(0,abs(pp) - 0.0),10);
        set(jh,'Xdata',jx,'Ydata',jy+pp*jz);
        drawnow;

        ylist(i)=y;
        ulist(i)=pp;
        x=adis*x + bdis*(pp+disturb(i));
        y=cdis*x + ddis*(pp+disturb(i));

        while (time2-time1<samper)
                time2=clock;time2=1000*(60*time2(5)+time2(6));
        end
        thetimes(i)=time2;
        time1=time2;

        if (y<-10 | y>10 )
                flg=1;crashind=i+1;
                thetimes(i+1)=thetimes(i)+samper;
                ylist(i+1)=y;
                ulist(i+1)=sign(p(1,2))*min(abs(p(1,2)),10);
                break;
        end

end

grphc2
```