AM404 Laboratory Experiment
Design, Implementation and Testing of a PID Controller for a Model Aircraft

Report due date: 04/24/08
Report must have date and time of the experiment performed, as well as the participants of your group. One report per group is required.

Objective: To design an implement a pitch controller for a model aircraft by using the Ziegler-Nichols tuning rule for PID controllers. This is a general method that permits the design of feedback controllers for systems with unknown transfer functions.

The experiment will be conducted on a model aircraft with three degrees of freedom (pitch, roll and yaw). Lift is provided by two high-speed fans mounted to the front of the model. The instrumentation of the model is capable of measuring all three angular rotations, but we shall only be interested in the pitch angle, which is changed by identically rotating both airfoils. The transfer function relating airfoil rotation to pitch angle is not known. Your objective is to design an "auto-pilot" controller that restores and maintains the aircraft at zero pitch following an externally applied deflection. You will design and implement PID-type controllers using the Ziegler-Nichols tuning rules (described hereafter), as well as test and describe the performance of your controllers.

The experiment consists of three parts. Part I is preliminary work and reading that work and reading that should be completed before attempting the experiment. Part II provides a step-by-step description of the task to be carried out in the lab. Part III prompts you to plot, analyze and discuss your results.

Part I Preliminary work *– to be completed before attempting the lab*

The Zeigler-Nicholstuning rule is a general experimental method that permits the design of PID controllers for systems with unknown transfer functions. The general form of the controller transfer function is

$$G_c(s) = K_p\left(1 + \frac{1}{T_i s} + T_d s\right) \tag{1}$$

We initially set $T_i = \infty$ and $T_d = 0$, which gives us proportional control action only. We keep increasing K_p from 0 to some critical value K_{cr}, at which the output exhibits sustained oscillations (For $K_p = K_{cr}$, the closed-loop system is thus __________________ stable.) The period P_{cr} of these oscillations can be determined directly from the output. Ziegler Nichols suggested that the values of K_p, T_i, and T_d should be set according to the following formulas for P, PI and PID controllers:

Type of controller	K_p	T_i	T_d
P	$0.5K_{cr}$	∞	0
PI	$0.45K_{cr}$	$P_{cr}/1.2$	0
PID	$0.6K_{cr}$	$0.5P_{cr}$	$0.125P_{cr}$

Note that these rules will yield a set of values of K_p, T_i, and T_d that will give stable operation of the closed-loop system, but not necessarily the best performance. Additional fine tuning may be required to reach the design specification and improve performance.

The elevation (pitch) controller for the model aircraft is implemented according to the transfer function

$$G_{ec}(s) = -\left(K_{ep} + \frac{K_{ei}}{s} + K_{ed} s\right) \tag{2}$$

Using equation (1) and the Zeigler-Nichols tuning rules, obtain expressions for K_{ep}, K_{ei}, and K_{ed} as a function of K_{cr} and P_{cr} for each controller type:

Type of controller	K_{ep}	K_{ei}	K_{ed}
P			
PI			
PID			

Part II Experiment Steps to running projects for AM 404 aircraft control demonstration:

IMPORTANT SAFETY NOTICE

Try to keep experiments to about five minutes at a time as the motors will heat up and could burn out if run for too long. Let motors rest for about five minutes after five minutes of running.

1. Make sure that both Universal Power Modules (large black boxes) are turned on.
2. Open MATLAB. Type **cd c:\am404** in the command window to set the correct path.
3. Type **am404setup** to set the controller gains. The first, second and third column give the proportional (Kp), Integral (Ki) and Derivative (Kd) gains for elevation (subscript e), roll (subscript r) and yaw (subscript y) control. Type whos to see a full list of variables. Make sure that **K_ep=K_ei=K_ed=0** (i.e there is initially no pitch control whatsoever).
4. Type **am404experiment** to launch the experiment. In the WinCon menu of the Simulink window, click **build**. A window called Untitled - WinCon server will pop up. Identify the green START button, but *do not* click on in yet.
5. In the WinCon menu of the Simulink window, click **open plot** and make sure that **elevation** is selected. This will open a Scope window that provides you with a plot of pitch angle versus time. Click **update** and select **buffer** and set this to 100 seconds. Under the Axis Menu, choose **Fixed** and set the **Fixed Range** from -10 to 10. Also, enlarge this window to full screen.
6. Level the arm by holding on the aft end (i.e on the side of the fans). Click on the green 'Start' button and wait until the wings have finished calibrating (it does not take very long). Then power up the fans by turning on the motor power supply (red button to the right of the middle shelf).
   FROM THIS POINT FORWARD, A MEMBER OF YOUR GROUP SHOULD GENTLY HOLD THE TOP OF THE LUCITE FRAME AT THE CENTER OF THE LEVER ARM WHENEVER THE FANS ARE RUNNING, SO AS TO PREVENT YAW ROTATION OF THE MODEL AIRCRAFT.
7. Clearly the flying wing is in an unstable mode – it will fall if not externally supported. Now introduce proportional control by simply typing **K_ep=-50** in the MATLAB command window.
8. Level the model aircraft, click on the green 'Start' button and wait until the wings have finished calibrating. Make sure that a member of your group is gently holding the Lucite frame to prevent yaw rotation. Turn on the fans.
9. Release the plane from the level position and observe what happens. Click on the red STOP button and turn off the fans. Describe what you observe and comment on the stability of the system with Proportional (P) pitch control.
10. Once more, level the model aircraft, click on the green 'Start' button and wait until the wings have finished calibrating. Turn on the fans. Whilst looking at the

Scope window, slowly push the aft end of the plane downwards until you reach about -5 degrees of pitch angle (it doesn't have to be exactly five degrees, but make sure to use the same value in all subsequent experiments). Be careful not to yaw or roll the model. Release the aft end of the plane and observe what happens for 60 seconds or so. Click on the red STOP button and turn off the fans.

11. Save the waveform displayed in the Scope window by clicking on the File Menu/ Save/Save as M-file. Make sure to copy all the data into your USB drive at the very end of the lab. <u>All that you need to do to redisplay this waveform in MATLAB is to be in the right directory and to type the filename (without the .m) at the command prompt.</u>
12. In the experiment that you just performed, what was the reference input and what was the disturbance? Can you guess why we chose a relatively small pitch deflection (~5 degrees) rather than a larger one?
13. Repeat Step 12 above by increasing **K_ep** in steps of -50, until you observe sustained oscillations in pitch (again, observe the output for at least 60 seconds before deciding whether this is the case). Save only the waveform that exhibits sustained oscillations to the disk, as described in Step 13 above. Note down the value of **K_ep** that gave you sustained oscillations: K_{cr} = ____________________
14. Replot the waveform that you just saved in MATLAB . Create a unique folder for your group and save the waveforms into file names that sound representative of their contents. At the end of the lab, you need to copy this data into an external USB drive. Use the MATLAB command **ginput** to measure the period of oscillations P_{cr} on the plot: P_{cr} = _______________ s.
15. Use your value of K_{cr} and P_{cr} to fill out Table 2 to determine the three different possible controllers for your unstable plant. We are now going to evaluate the performance of the different controllers that have been designed. To do this, you need first to set the values of K_ep, K_ei, K_ed in MATLAB corresponding to a particular controller. Then repeat steps 10-11 to test your controllers.
16. Set the values for the variables **K_ep, K_ei** and **K_ed** for the P controller in MATLAB. Make a rough estimate of the steady state error.
17. Now implement the PI controller. Does this improve on the performance of the P controller in any way? Is it worse than the performance of the P controller in any way?
18. Now implement the PID controller. How does the PID controller compare with the P and PI controllers?
19. As a final test of your PID controller, repeat 10-11 (you don't have to save the data this time), but gently push down on the aft end of the plane until the fans touch the table, then release the wing. Describe what you observe. Can you provide a qualitative explanation of why PID control performs well, even

for relatively large deflections in pitch? (Hint: What does PID control introduce?)

Part III Analysis and discussion of results

The lab report should include at least, but not limited to the following:
1. **Title page** This should include only the title of the experiment, the date and time of the experiment, your name, and all other students who did the experiment with you using your apparatus.
2. **Theory** Briefly discuss the basic idea behind the PID controller and the Zeigler-Nichols rule.
3. **Data** Show the tables, values of gains and other relevant data.
4. **Plots** Make sure your plots have captions and labels.
- Pitch angle with time for the P controller with K_ep = - K_{cr} . Indicate the period of oscillation P_{cr} on the plot.
- Pitch angle with time for each of the P, PI and PID controllers with steady state error rise time, overshoot, settling time and any other relevant quantities shown.
5. **Discussion and Conclusions** Compare the three controllers and give your conclusions. Give a qualitative discussion of how the three types of controllers functioned.

Revision history:
Dr.Constantin-C.Coussios (04/24/03)
Dr.Pierre Dupont and Dr.Hua Wang (04/27/04)
Dhananjay Raghunathan (04/04/06)
Dhananjay Raghunathan (04/03/08)