Report for a Simplified Auto-Pilot Model and Simulation in CD++

Diane Hould B. Eng. P. Eng.
M. A. Sc. CU Student
1125 Colonel By Drive
Ottawa, Ontario
K1S 5B6
dhould@connect.carleton.ca

ABSTRACT

This report discusses the model and simulation for a simple auto-pilot. The simulation is coded in CD++ and consists of two (2) atomic models that are coupled. The model represents the auto-pilot control in the "y-axis" (pitch) and the simulation controls the aircraft elevators, the flight controls that control the pitch on the aircraft. The elevators have a range from 40 to -50 degree angle in 10° increments. The simulation is verified with three (3) chosen variables the two (2) extremities and a non-zero point in the middle.

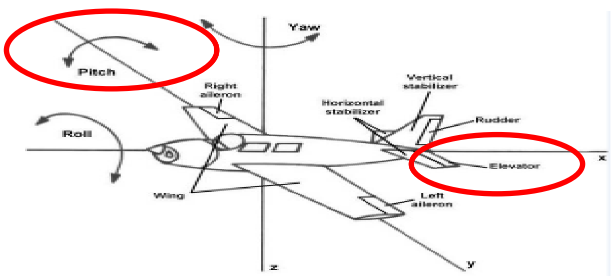

Figure 1: CD++ Autopilot Concept Model

Preface

Modeling and simulation are tools that are used in the aviation industry to determine and predict the real world responses to varying inputs. The cost of modeling and simulation within aviation is high however it is still more efficient in time and money then to perform testing on the final product and incorporate changes post final build.

1. OVERVIEW

An autopilot is a system controller designed to control aircraft flight controls without a manual input. It is a Proportional Integral Derivative (PID) controller. The input is a derivative of the output. The second derivative is the rate of the change within the flight controls. This is the acceleration from the desired pitch to the actual pitch. The derivative of the rate is the input into the autopilot controller that will determine the necessary pitch to control the aircraft based on it's current pitch.

2. CONCEPT

The autopilot model has two (2) components, the autopilot controller and the aircraft controller. The operator enters a desired pitch, for example straight and level flight. The autopilot controller sends a control surface deflection command (elevator angle) to the aircraft controls. The aircraft controls then act on the command and output an actual pitch. The actual pitch is then inputted back into the autopilot to loop through the process until steady state is achieved. Figure 1 is a diagram of the concept model for the autopilot.

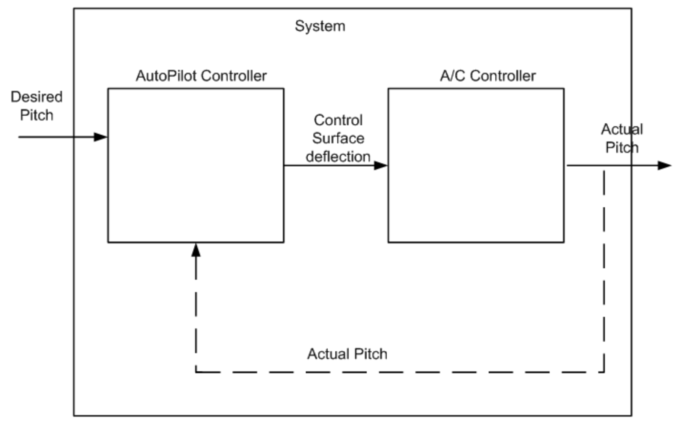

Figure 1: CD++ Autopilot Concept Model

Steady state for the simulation is reached when the output variable is steady. The output value is observed until the output value is equal to the output value of the previous iteration.

3. AUTOPILOT CONTROLLER ATOMIC MODEL

The autopilot is the controller for the system. It controls the aircraft flight controls based on the aircrafts actual position and the desired position. The autopilot controller operate as defined in the following subsections.

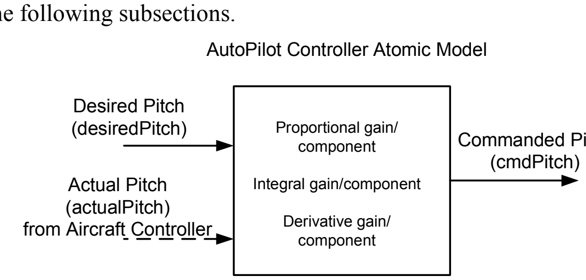

Figure 2: Autopilot Controller Atomic Model

Figure 2 is a diagram of the autopilot controller atomic model. As seen in the figure the inputs are the desired and actual pitch, within the model the integral gain and components are applied to the input to determine the pitch command to the aircraft controller.

3.1 Initializing the Parameters

There are several parameters identified as part of the autopilot controller. The ones that are initialized are the lastDelta, represents the difference between the actual pitch and the desired pitch during the previous iteration. LastIC is the integral component from the previous iteration.

lastDelta = 0.0
lastIC = 0.0
cmdPitch = 0.0

cmdPitch represents the pitch command from the autopilot controller to the aircraft controller).

3.2 Input

desiredPitch
actualPitch

3.3 Computations

deltaPitch = desiredPitch – actualPitch
dt = time since last iteration in seconds
ku = 0.5
pu = 20
ku and pu are the values that were tuned
pg = 0.60*ku where pg is the proportional gain
ig = 1.20*(ku/pu) where ig is the integral gain
dg = 0.075*(ku*pu) where dg is the derivative gain
pc = pg*deltaPitch where pc is the proportional component
dc = dg*((deltaPitch – lastdelta)/dt) where dc is the derivative component
ic = ((1-dt)*lastic) + (ig*deltaPitch*dt) where ic is the integral component (this is bounded by the derivative and proportional components)
lastdelta = deltaPitch at previous iteration
lastic = ic at previous iteration
cmdPitch = pc + ic + dc

3.4 Output

cmdPitch

4. AIRCRAFT CON CONTROLLER ATOMIC MODEL

The aircraft controller atomic model is shown in figure 3. The commanded pitch from the autopilot controller is used to determine the actual pitch based on the rate of change for the actual pitch. The actual pitch rate (actPitchRate) is the rate of change for the elevators of the aircraft. A tolerance is applied to the delta pitch. When the difference between the commanded pitch and the actual pitch is less than 0.5°, the simulation is to do nothing, passivate. This is to dampen the oscillation effect of the autopilot attempting to reach the precise commanded value.

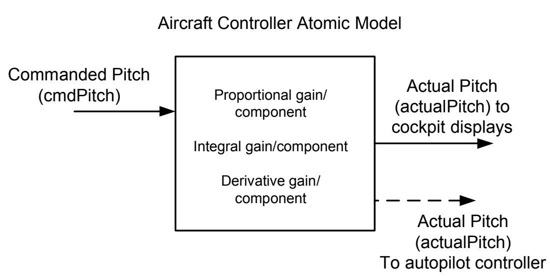

Figure 3: Aircraft Controller Atomic Model

The aircraft controller is defined in the following subsections.

4.1 Initializing the Parameters

actualPitch = 0.0
actPitchRate = 0.0

4.2 Input

cmdPitch

4.3 Computations

deltaPitch = cmdPitch – actualPitch
if (deltaPitch > 0.5) then actPitchRate = 0.01
if (deltaPitch > -0.5) then actPitchRate = 0.01
if ((deltaPitch > 0.5) and (deltaPitch > -0.5)) then actPitchRate = 0.0
actualPitch = actualPitch + actPitchRate

4.4 Output

actualPitch

5. OSCILLATION DAMPER/FINE TUNING

This model is subject to oscillating around the desired pitch. In order to minimize the oscillation around the desired pitch, the desired pitch will be represented by a range and not a defined value. This will prevent the aircraft from "jerky" responses as a reaction to the autopilot while it oscillates around the desired pitch. The desired band has a tolerance of +/- 0.5 degree in pitch.

Although the inputs into the autopilot are within 10 degree increments, the aircraft sensitivity will report actual elevator pitch that are in 1 degree increments.

6. VERIFICATION AND VALIDATION

6.1 Verification

Confirming that the solution converges provides a level of confidence that the model is behaving as required. This verifies the simulation to the model. To verify that the mathematical equations were implemented correctly three (3) desired pitch values were calculated to determine the actual pitch using the same equations used in the simulation. When the pitch values are entered in the event file (.ev) the expectation is for the output value to be the same as the calculated values for the actual pitch. The time to converge is calculated to confirm the actual pitch values are achieved when expected.

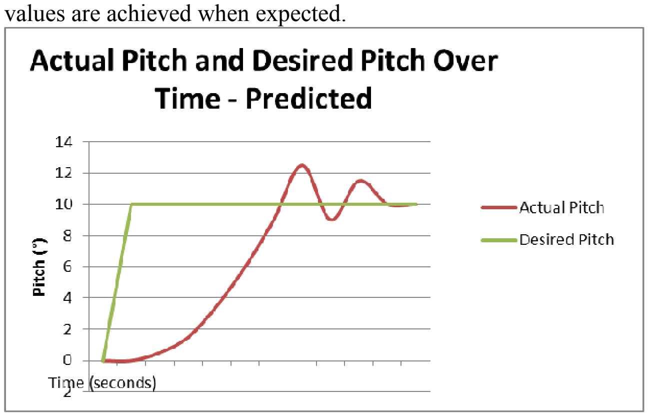

Figure 4: Actual and Desired Pitch Over Time Graph

Figure 4 is the actual pitch over time commencing at an elevator position of 0° and increasing to 10° following a desired pitch input of 10°. The ku, pu and dt values were modified until the shape of the curve was achieved,

6.1.1 Actual Pitch - Calculated

The desired pitch values used for the calculated method are the two (2) extremities for the flight control (elevators) 40° and -50° and 10° to represent a number in the middle. The initial aircraft position is 0° for all three (3) positions. Table below shows the calculated values obtained. Obtaining the calculated values was tedious and the actual pitch obtained of 9.5 degrees took close to 1000 iterations (manual operations) before being within tolerance of the desired pitch. Each iteration represents 0.001 of a second. This number is the same value (time advance value) used in the simulation.

Table 1. Actual Pitch – Calculated

Desired (°)	Actual (°)	Time
40	39.5	6.5 sec
10	9.5	1.4 sec
-50	49.5	8.5 sec

6.1.2 Actual Pitch - Simulation

The event file used for the simulation contained the same three (3) pitch values used for the calculated method to determine the actual pitch values. The events were defined 10 seconds apart and aircraft position returned to 0°. All elevator positions were tested within the simulation. The table below shows the pitch values used and the corresponding outputted actual pitch values. The initial position for the aircraft was 0 and the aircraft would return to 0 prior to receiving a command for the next desired pitch. To ensure the desired pitch command was not issued prior to the aircraft re-initializing itself to the 0 position, the worst case time of 8.5 seconds prior to issuing a desired pitch command was implemented in the event file, the value of 10 was used. The tolerance values of 0.5 and -0.5 were selected to observe the behaviour of the simulation at the simulation decision points.

Table 2. Valid Actual Pitch – Simulation Results

Desired (°)	Actual (°)[1]	Iterations
40	39	3950
30	29	2500
20	19	1500
10	9	500
0	0[2]	0[3]
-10	11	1500
-20	21	2500
-30	31	3500
-40	41	4500
-50	51	5500

[1] NOTE: These values (actual and iterations) are estimates based on the simulation. These values to be confirmed and provided under separate cover.

[2] 0 is 0 because the initial position of the aircraft is 0 and the deltaPitch is within the desired Pitch tolerance. No movement of elevators is required.

[3] The intent of using 0 is to ensure the mathematics is correct and the behaviour is as predicted. The event file uses 0 as the baseline and thus why it is expected a 0 number of iterations,

6.1.3 Verification – Outside Range Valid/Invalid Data

The valid range for the elevator flight control is from 40 to -50 °. A test case was created to ensure the simulation reacted the way it was intended. The simulation should not react to any invalid values for desired pitch requests that are outside the valid range for the elevators. These inputs, such as 41 and -50.5 are to be ignored. Valid elevator values but outside the valid range, such as 50 and -60 are to be ignored as well. These values are entered into the event file. When these invalid and/or out of range elevator positions are inputted the expected output is 0. The initial position of the elevators is 0°.

Table 3. Invalid Actual Pitch – Simulation Results

Desired (°)	Actual (°)[4]
40.1	0
-50.5	0
50	0
-60	0

6.1.4 Verification – Inside Range/Invalid Data

Test case defined to confirm the simulation will not respond to an invalid value for elevator position when an invalid value such as 45 and 3 are inputted as desired pitch and it is within the valid range. This test case was not created because the assumption was the restriction for the data input would occur at the autopilot controller interface. Therefore this condition would not exist within the simulated environment.

6.2 VALIDATION

Validating the model to the real world was not performed as part of the testing. A complex model was simplified for this report and validating the simulation quantitatively is not feasible. If this model was to be validated in the real world a possible scenario would be to validate the model at the system level. When and input is provided to the autopilot the expected output of the aircraft position can be confirmed on the aircraft instrumentation. Not all autopilot inputs need to be validated during flight test, flight tests are costly and validation can be done on a subset of inputs and when the model and simulation are validated then the remaining inputs can be confirmed using the model..

[4] NOTE: These values are estimates based on the simulation. These values to be confirmed and provided under separate cover.

7. DISCUSSION

The simulation values are competitive to the calculated values for actual pitch. Dampening was a trial and error effort. Manipulation of the constants was required to achieve the desired output. To minimize this effort in future, the optimizing of the output can be achieved by defining the criteria for the desired output and automating the defining of the constants within the simulation.

The simulation is only as good as it's input and validating the simulation using the same equations as the simulations only verifies the implementation of the equations and not the equations themselves. In real world applications the simulation would be validated with a few flight tested inputs. During the flight test the aircraft provide an input into the autopilot and confirm the aircraft's position using the aircraft instrumentation.

If the elevator position increments were defined in the software as in some applications the test case to confirm that invalid values for desired pitch within the valid range would need to be tested.

An additional test case to show the simulation response to one extreme of the elevator position when in the other extreme of the elevator position to confirm no issues with the cross-over from a negative to a positive pitch.

8. REFERENCES

[1] Wainer, A., Gabriel 2005. CD++ A Tool for DEVS and Cell-DEVS Modeling and Simulation Users Guide, Department of Systems and Computer Engineering Carleton University.

[2] Carleton University, Methodologies for Discrete-Event Modeling and Simulation, www.sce.carleton.ca/courses/sysc-5104