**Civil Engineering Case Study – Aviation Safety**

**Yousef Al-Jazzazi**

***ENGR-AD 1000, Fall 2022***

**STEP 1: *Problem Identification and Statement***

The problem is checking if the location detecting systems are up to aviation safety standards.

**STEP 2: *Gathering of Information and Input and Output Description***

Aviation safety is a subject of the highest priority, with aircraft positions of paramount importance. To ensure that the position is accurate, two different sensors keep a continuous record of the aircraft’s location every quarter second. Assume that location is recorded in (x, y, z) coordinates, measured relative to the air traffic control tower. The x-coordinate is the east-west location (east positive), the y-coordinate is the north-south location (north positive), and the z-coordinate is the altitude. All values are in kilometers. The first sensor takes its data from a GPS (global positioning system) and stores it in a data file saved to the aircraft’s “black box”. The second sensor takes its data from the air traffic control RADAR system and saves it to another file in the GANS database (In Abu Dhabi, air traffic control is provided by the Global Air Navigation Services (GANS), a private corporation). To determine the data consistency, the two positions of the airplane should be compared by finding the distance between them and checking whether the distance is within the aviation safety standards. (notice in this problem that the difference between the two reported positions is the inaccuracy. From this point forward when we refer to the error, it will just be called the distance for consistency reasons)

Calculating the distance between the two reported positions on a 3D system:

We’ll call the first position point 1 and the second position 2  
We use the Pythagorean identity to find the distance between the x and y position of point A and that of point B. We’ll call that distance L. By using the Pythagorean theorem again-this time L being a leg- we will find distance D. This is illustrated in the graph below

Diagram, schematic

Description automatically generated

Credit: the organic chemistry tutor YouTube channel

Notice here that point 1 is assumed to have coordinates of (0,0,0 ). We can generalize this formula by having each point be the difference between the point’s position at that coordinate giving us an equation to find the distance:

Text

Description automatically generated with medium confidence

Dealing with Files:

As mentioned above, the position of the airplane is calculated through two sensors. The first sensor uses GPS, and the file from this sensor will be named file1. The second sensor takes its data from the air traffic control RADAR system and saves it to another file in the GANS database, and the file from this sensor will be named files2.

Note: Those two files will be given by the user. Our algorithm is concerned with comparing the two files rather than producing them.

The third file is where the time of inaccuracies of the positions occurs alongside the difference in location. This file will be produced by the program and is going to be named ERR.LOG.

We will assume that the file starts with the data and doesn’t have the time. (We are assuming that it starts with time equals to 0, and with every point, we are going to increment the time by a quarter of a second).

This will be organized through a menu where it is going to ask the user what function of the program they want to work with, namely (1) calculating the error between the two sensors recordings and saving it in the error file, (2) displaying the error file information, or (3) quit the program. In case the user selects the first option (calculating the error and saving it into the error file), the program prompts the user for the respective inputs (tolerance value and the two data file names), calculates the errors, and saves the error values that exceed the tolerance in the ERR.LOG file. The program then returns to the main menu. If the user selects the second option, the program displays the content of the ERR.LOG file in a tabular format. An error message will be printed in case the ERR.LOG file does not exist. The program then returns to the main menu. If the user selects the exit option, the program prints a terminating message and quits the program.

**I/O description:**

The program will take three inputs and produce one output:

**Error bound**

**Name of file2**

**Name of file1**

ERR.LOG

There are two exception scenarios that my algorithm considers. The first is when there are more values in one of the input files; the program will store the distance as 0. The other scenario is when the value of z value is less than 0 which is impossible practically as the airplane cannot be under the station; the program will store the distance as -1

**STEP3: Test Cases and Algorithm Design:**

1. Test cases

Test case #1:

File 1: “GANS.txt”

12 12 43

0 0 12

File 2: “BlackBox.txt”

0 0 0

23 32 43

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Menu Input**  **[int]** | **File Name 1 Input [String]** | **File Name 2 Input [String]** | **Accepted Value Input [double]** | **What happens** | **Output** |
| 1 | “Yousef.txt” | “BlackBox.txt | 123 | There is no file named Yousef.txt | It will give an error telling the user that Youse.txt doesn’t exist. It will keep asking them for file name until it works. |

**Continue**

|  |  |  |  |
| --- | --- | --- | --- |
| **Go back to menu** | **Output** | **Go back to menu** | **Output** |
| 2 | Time[s] Distance[km] | 3 | Thanks for using our program |

Test case #2:

File 1: “GANS.txt”

-12 12 33

2 14 15

File 2: “BlackBox.txt”

12 12 31

2 14 16

56 28 32

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Menu Input**  **[int]** | **File Name 1 Input [String]** | **File Name 2 Input [String]** | **Accepted Value Input [double]** | **Go back to menu** |
| 1 | “GANS.txt” | “BlackBox.txt | 12 | 2 |

**Output:**

|  |  |
| --- | --- |
| Time[s] | Distance[km] |
| 0 | 24.08 |
| 0.5 | 0 |

Test case #3:

File 1: “GANS.txt”

12 33 22

35 24 14

File 2: “BlackBox.txt”

12 31 21

34 21 15

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Menu Input**  **[int]** | **Output** | **Go back to menu** | **File Name 1 Input [String]** | **File Name 2 Input [String]** | **Accepted Value Input [double]** | **What is going to happen** | **Go back to menu** |
| 2 | The file has not been created | 1 | “GANS.txt” | “BlackBox.txt | -1 | The file is going to take the absolute value | 2 |

**Output:**

|  |  |
| --- | --- |
| Time[s] | Distance[km] |
| 0 | 2.236 |
| 0.25 | 3.3166 |

Test case #4:

File 1: “GNS.txt”

56 45 -1

1234 1234 1234

File 2: “BlackBox.txt”

59 40 1

12340 1234 1234

1546 123 345

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Menu Input**  **[int]** | **Output** | **Menu Input [int]** | **File Name 1 Input [String]** | **File Name 2 Input [String]** | **Accepted Value Input [double]** | **Go back to menu** |
| gg | Please input one of the options | 1 | “GNS.txt” | “BlackBox.txt | 42 | 2 |

**Output:**

|  |  |
| --- | --- |
| Time[s] | Distance[km] |
| 0 | -1 |
| 0.25 | 105.6 |
| 0.5 | 0 |

**Continue:**

|  |  |
| --- | --- |
| **Go back menu** | **Output** |
| 3 | Thanks for using our program |

1. Algorithm (Pseudocode)

Include the iostream, cmath,fstream,iomanip and libraries

Initialize function menu type void with parameter bool pointer

Initialize functions calcError and displayError type void with no parameters

main function

Display to the user a message explaining what the program does

Declare bool variable repeat and store true in it

Start a while loop with condition repeat is true

Call function menu and pass variable repeat as a parameter

End Loop

End main function

Start void function menu with a bool pointer as a pramater

Display a message that explains the options of the menu and ask user for input choice

Declare string variable choice

input value from user and store it in variable choice

if choice is equal to 1

call calcError function

else if choice is equal to 2

call displayError function

else if choice is equal to 3

set repeat equal to false

else choice is equal to anything else

display message that asks the user to input one of the options

End menu function

Start calcError void function

Declare two ifstream variables file1, file2

Declare ofstream variable errLog

Declare string variable nameFile

Declare double variable tolerance

Display a message asking the user for first file name input (or 0 if they want to exit the function)

Input value from the user and store it in nameFile

Open file1 using the nameFile string variable

Start while loop that keeps iterating as long as opening file1 failed

If nameFile is equal to 0

Exit function

Display error message that no file with that name exist and asking the user to input another name

Input value from the user and store it in nameFile

Open file1 using the nameFile string variable

End Loop

Declare string variable nameFile

Display a message asking the user for second file name input (or 0 if they want to exit the function)

Input value from the user and store it in nameFile

Open file2 using the nameFile string variable

Start while loop that keeps iterating as long as opening file2 failed

If nameFile is equal to 0

Exit function

Display error message that no file with that name exist and asking the user to input another name

Input value from the user and store it in nameFile

Open file2 using the nameFile string variable

End Loop

Opening “ERR.LOG” file using ofstream variable errLog

Display a message asking the user for the tolerance value

Input from user and store in tolerance variable

Declare variables type double xfile1, yfile1, zfile1

Declare variables type double xfile2, yfile2, zfile2

Declare variable type double time

Start a while loop where the condition is inputing the values from file1 into xfile1, yfile1, zfile1 and inputing values from file2 into double xfile2, yfile2, zfile2 works

Declare variable type double distance

If statement with condition zfile1 or zfile2 is negative

Store -1 and the time into errLog file

Increment time by 0.25

Ignore the rest of the while loop in this iteration

Store value sqrt [( xfile1-xfile2)^2 +( yfile1-yfile2)^2 + ( zfile1-zfile2)^2] into distance variable

If distance is greater than absolute value tolerance

Store distance and time into errLog file

Increment time by 0.25

End Loop

Start while loop that iterates as long as one of the files still has some values

input the values from file1 into xfile1, yfile1, zfile1

input values from file2 into double xfile2, yfile2, zfile2

Store 0 and time in errLog file

Increment time by 0.25

End Loop

Close files file1, file2, and errLog

End CalcError function

Start void Function displayError

Declare ifstream variable errLog

Open file “ERR.Log” using errLog. If it does not exist it should not create it

If opening file doesn’t work

Display error message

Exist function

Display “Distance” and “Time” in a tabular format where each cell has width of 15 characters

Declare variables type double distance and time

Start while loop with condition inputting from errLog file into distance and time works

Display distance and time values in a tabular format where each cell has width of 15 characters

End Loop

Close errLog file

End function

**Step 4: Implementation**

/\*

Name: Yousef Al-Jazzazi

netID: ya2225

Asssignment 1 for CPE class

Project name: Civil Engineering Case Study – Aviation Safety

Description: The program checks whether the location detecting systems in airplanes are up to aviation safety standards.

\*/

#include <iostream>

#include <cmath>

#include <fstream>

#include <iomanip>

using namespace std;

//Initilizing functions

void menu(bool&);

void calcError();

void displayError();

int main() {

cout << "This function of this program is to calculate the inaccuracy of an ariplane positioning system"

<< "\nThe airplane store its positions using two different systems."

<< "The program checks the inaccuracy by calculating the differenct between the two systems"

<< endl;

bool repeat = true;

//we pass repeat variable as a pointer, so it can be modified from inside menu function

while(repeat)

menu(repeat);

return(0);

}

//A function to display the menu and ask the user what function in the program they want to use (or whether they want to exit the program)

void menu(bool& repeat) {

cout << "There are two functions of the program."

<< "\nPress 1 to calculate the error"

<< "\nPress 2 to display the error. Display the error only after calculating it"

<< "\nPress 3 to exit the run" << endl;

//I choose the combination of string and if statement rather than an switch statement because

//in the case a user input a string in an int, it will result in a failed comand with an infinite loop

string choice;

string choice;

cin >> choice;

if (choice == "1")

calcError();

else if(choice == "2")

displayError();

else if (choice == "3") {

cout << "Thanks for using our porgramm" << endl;

//The menu function will be called continuesly until repeat is false which is when the user choice is 3

repeat = false;

}

else

cout << "Please input one of the options" << endl;

}

//A function to calculating the error and storing them in a file

void calcError() {

ifstream file1, file2;

ofstream errLog;

string nameFile;

double tolerance(0);

cout << "To calculate the error. Please enter the names of the two files that store the position of the airplane" << endl;

cout << "Start by entering the name of the first file: (Press 0 to go back to the main menu)" << endl;

cin >> nameFile;

file1.open(nameFile, ios::in);

//if the name inputed by the user doesnt work, keep asking the user for a valid name.

//The user can opt out by inputting a 0

while (file1.fail()) {

if (nameFile == "0")

return;

cerr << "Error. The file doesn't exist. Please try again" << endl;

cin >> nameFile;

file1.open(nameFile, ios::in);

}

cout << "Please enter the name of the second file: (Press 0 to go back to the main menu)" << endl;

cin >> nameFile;

file2.open(nameFile, ios::in);

//if the name inputed by the user doesnt work, keep asking the user for a valid name.

//The user can opt out by inputting a 0

while (file2.fail()) {

if (nameFile == "0")

return;

cerr << "Error. The file doesn't exist. Please try again" << endl;

cin >> nameFile;

file2.open(nameFile, ios::in);

}

errLog.open("ERR.LOG", ios::out);

cout << "Please enter the magnitude of the accepted" << endl;

cin >> tolerance;

double xfile1(0), yfile1(0), zfile1(0);

double xfile2(0), yfile2(0), zfile2(0);

double time(0);

while (file1 >> xfile1 >> yfile1 >> zfile1 && file2 >> xfile2 >> yfile2 >> zfile2) {

double distance(0);

//Because the airplane cannot be under the radar. If one of the heights is negative, the value -1 will be stored as the distance

if (zfile1 < 0 || zfile2 < 0) {

errLog << -1 << " " << time << "\n";

//Time incremented by 0.25 because we know that the position is calculated ever quarter of a second

time += 0.25;

continue;

}

//Equation of two points in a 3d coordinates

//Derived from applying the pythagorean theorem twice

distance = sqrt((xfile1 - xfile2) \* (xfile1 - xfile2) + (yfile1 - yfile2) \* (yfile1 - yfile2) + (zfile1 - zfile2) \* (zfile1 - zfile2));

//Only store in the error file if the distance is greater than the inputted tolerance value

if (distance > abs(tolerance))

errLog << distance << " " << time << "\n";

time += 0.25;

}

//Another while loop in case there are more values in one file

//the distance will be stored as 0 indicating this scenario

while (file1.eof() != file2.eof()) {

file1 >> xfile1 >> yfile1 >> zfile1;

file2 >> xfile2 >> yfile2 >> zfile2;

errLog << 0 << " " << time << "\n";

time += 0.25;

}

//closing all the files

file1.close();

file2.close();

errLog.close();

}

//A function to display the errors stored in the error file

void displayError() {

ifstream errLog;

errLog.open("ERR.LOG", ios::\_Nocreate);

//In case the error file has not been created

if (errLog.fail()) {

cerr << "File doesnt exist. Please calculate the error before displaying the error\n\n\n" << endl;

return;

}

cout << left << setw(15) << "Time[s]" << setw(15) << "Distance[km]" << "\n";

double distance(0), time(0);

while (errLog >> distance >> time)

cout << left << setw(15) << time << setw(15) << distance << "\n";

errLog.close();

}

**Step 5: Test Cases and verification**

**Corrections: Names GNS and blackBox where substituted with file1 and file2 for simplicity**

**Corrections: menu input was changed into string instead of an int**

**Test #1:**

File 1: “file1.txt”

12 12 43

0 0 12

File 2: “file2.txt”

0 0 0

23 32 43

**Text

Description automatically generated**

**Note: instead of asking the name of the two files and then checking. I decided to check after asking for each file for clarity reason**

**Test #2:**

File 1: “file1.txt”

-12 12 33

2 14 15

File 2: “file2.txt”

12 12 31

2 14 16

56 28 32

**Text

Description automatically generated**

**Test # 3:**

File 1: “GANS.txt”

12 33 22

35 24 14

File 2: “BlackBox.txt”

12 31 21

34 21 15

**Text

Description automatically generatedWhen calling the function 2, the program didn’t produce the error because ERR.LOG file already exist from previous runs. If the file is deleted this what is produced Text

Description automatically generated**

**Test #4 :**

Test case #4:

File 1: “GNS.txt”

56 45 -1

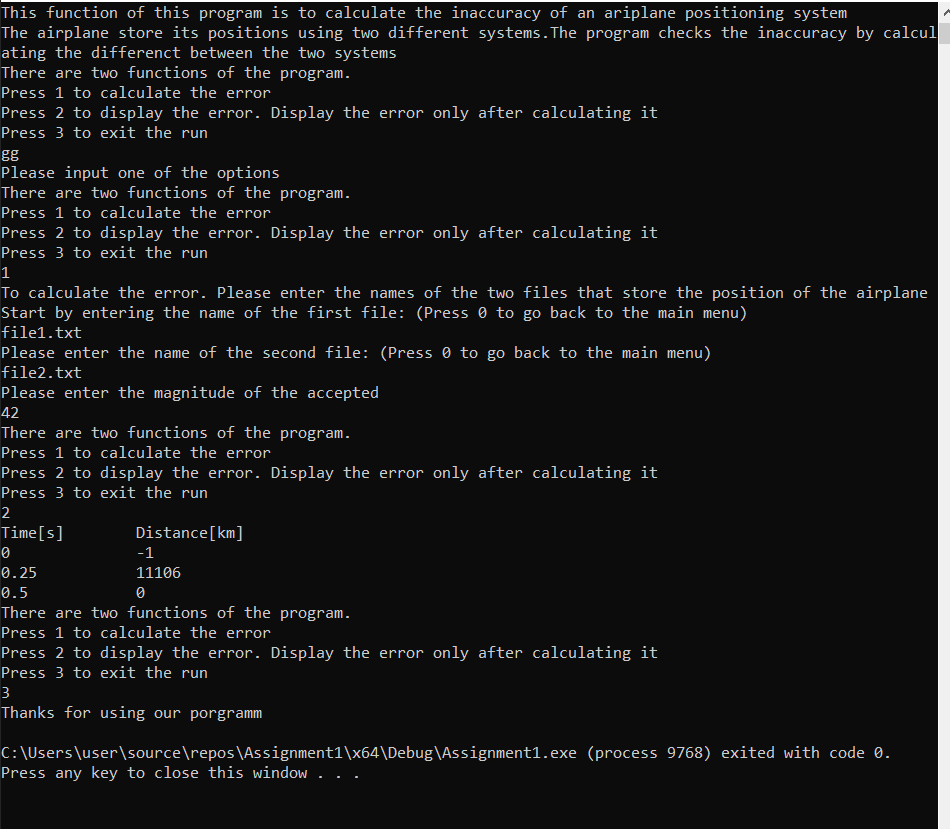
1234 1234 1234

File 2: “BlackBox.txt”

59 40 1

12340 1234 1234

1546 123 345

****

**Correction: at the beginning, inputing a string an int value resulted in an infinite loop, so I declared choice variable as string instead of an int**

**Correction: for the second value at int 0.25, the value calculated by hand is wrong after checking again.**