

**Omar Mohamed Atia Mohamed Shehab**

**Professor Mohamed Eid**

**Computer Programming for Engineers**

**9 October 2022**

---

**Step 1: Problem Identification and Statement:**

- The purpose of the problem we are addressing is to work on calculating the location of the aircraft every qatar second to check if it exceeds the tolerance value or not. This is done to obtain safe aviation for the planes.
- This is implemented using the data provided by the two sensors(GANS & GPS) and calculating the distance of the coordinates provided by the sensors.

**Step 2: Gathering Information:****→ The law of distance:**

As the main factor of the problem is the distance of the plane above the earth during aviation, therefore law of distance is the essential tool to be used in order to proceed in the process which is given in figure 1:

The coordinates of each point (A,B)

If  $A = (x_1, y_1, z_1)$  ;  $B = (x_2, y_2, z_2)$ , then  
Distance between A and B

represents the coordinates of the plane and

$$= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

the coordinates of a point on the ground that

Figure 1

will be stored from the two sensors **GANS** and **GPS**.

In addition, the origin point (0,0,0) is supposed

to be the starting point of the plane. The

coordinates of two points in 3d dimension is

shown in figure 2. After calculating the distance

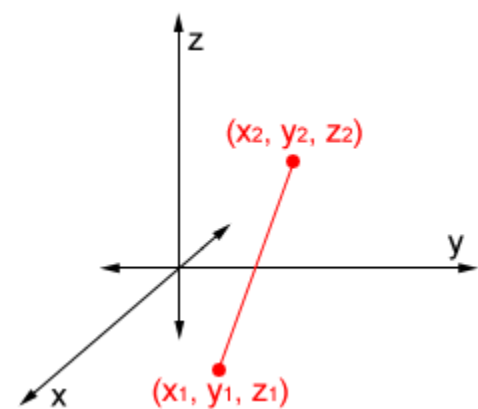


Figure 2

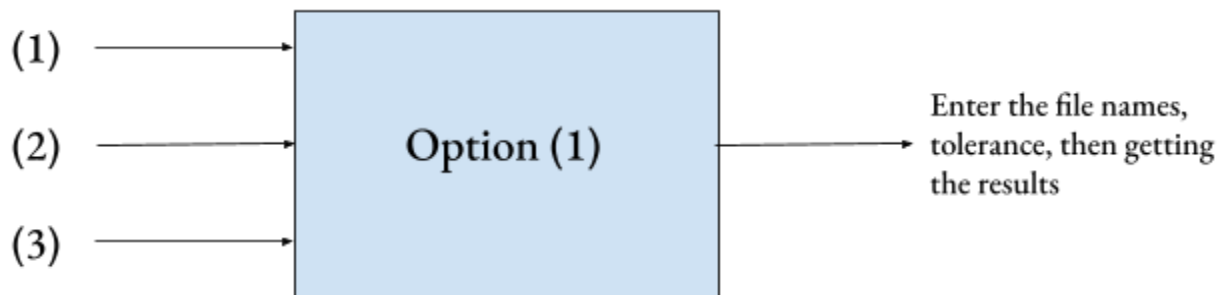
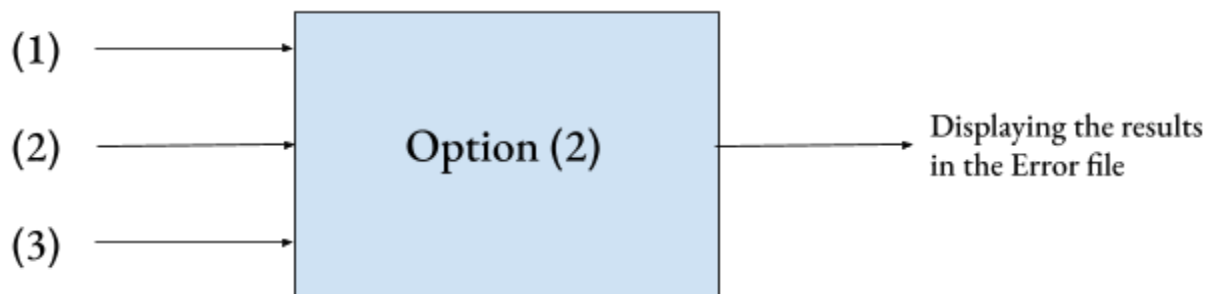
between the two points, we should compare it with the

tolerance to determine if this distance is valid or not.

**→ I/O Menu:**

There will be a message sent to the user as soon as the program runs which is:

1. Calculate the error between the two sensors' recordings and save them in the error file.
2. Display the error file information.
3. Quit the program.

**→ Input/output for option(1):****→ Input/output for option(2):****→ Option three will quit the program.**

### Step 3: Test cases & Algorithms

#### A. Test Cases:

##### 1. Entering invalid input(rather than the inputs in the menu):

```
Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
5
Invalid input, please Enter a correct number
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
```

The program will display an error message and ask the user to enter a valid number again with the menu popping up again.

##### 2. Choosing option (1):

```
Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
1
Please Enter the Two file names
file1.txt
file2.txt
Please Enter the tolerance:
4
```

Asking the user for the file names, then entering the tolerance and the distance will be printed in addition to the time taken for each point which is incremented by 0.25 seconds.

And the distance for the coordinates of the first two points will be calculated through the formula as follows:

$$distance = \sqrt{(1 - 4)^2 + (2 - 5)^2 + (3 - 6)^2} = 5.196152423$$

### 3. Choosing option 3:

When the user chooses this option, the program will show this message and will quit.

```
Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
3
Quitted
```

#### ● Results of the process:

P1 (x1, y1, z1)	P2 (x2, y2, z2)	Distance	Tolerance	Output in err file
(1, 2, 3)	(4, 5, 6)	5.19615	4	5.19615
(12, 13, 14)	(1, 2, 3)	19.0526	4	19.0526
(0, 0, 1)	(0, 1, 4)	3.16228	4	Will not be printed
(-6.3, -1.3, -2.5)	(-6, -8, -9)	9.3397	4	9.3397
(3.3, 5.5, 4.4)	(-3, -5, -4)	14.8492	4	14.8492
(3, 1, 6.645751311)	(0, 1, 4)	4	4	Will not be printed

**B. Algorithm:****MAIN:**`Declare tolerance, x1, y1, z1, x2, y2, z2, dis as float``Declare time as double and assign (0) to its value``Declare valid as boolean``While as valid is true Repeat``Printing "Choose your option" Newline``Printing "(1) Calculate the error between the two  
    sensors recordings and saving in the error file"``Newline``Printing "(2) Display the error file information"``Newline``Printing "(3) Quit the program" Newline``Declare num as integer``Read input value into num``If num equals 1``Declare name1 and name2 as string``Print "Please Enter The Two file names" endl``Read input values into name1 and name2``(Validate input)``Declare file1 as input file`

```
    Assigning name1 as the name of the input file
to file1
    Declare file2 as input file
    Assigning name2 as the name of the input file
to file2
    If file1 failed or file2 failed then
        Print "Error in displaying the files"
        Read input values into name1 and name2

    Declare err as an output file("Err.log")
    If err failed then
        Print "Error in displaying the file"
    Newline
        Exit the file
    Print "Please enter a non-negative tolerance: "
    Newline
    Read input value into tolerance
    While tolerance exists
        If tolerance greater than 0
            Break
        Print "Please Enter a non-negative
tolerance: " Newline
        Read input value into tolerance
    Print set width(10) before "Distance: " set
width(10) before "Time: "
    While as not end of file1 and not end of file2
repeat
```

```
    Read input value into x1, y1, z1
    Read input value into x2,y2, z2
    Assign square root of (x1-x2)^2 + (y1-y2)^2 +
(z1-z2)^2
    If dis is greater than the tolerance then
        Increment time by 0.25
        Print dis,set width(10)before time in the
error file, endlne
        Print dis, time, endlne

    Assign false to valid
    Else if num equals 2
        Declare display as input file with the value
"Err.log"
        If display failed then
            Print "Failed to open the files" Newline
        Declare number, sec as double
        Repeat while reading number and sec from
display
            Print set width(5) before number and sec
endlne
        Break
    Else if num equals 3
        Print "Quitted" endlne
        Assign false to valid
    Else
```



```
Print "Invalid input, please Enter a correct  
number" endl
```

```
Printing "(1) Calculate the error between the  
two sensors recordings and saving in the error file"  
Newline
```

```
Printing "(2) Display the error file  
information" Newline
```

```
Printing "(3) Quit the program" Newline
```

```
Read input value into num
```

```
Return 0
```

---

#### Step 4: Code

```
/*-----*/  
/* Omar Mohamed Atia Shehab */  
/* Date: October 9, 2022 */  
/* Assignment1 */  
/* Aviation Safety program */  
  
#include<iostream>  
#include<fstream>  
#include<iomanip>  
#include<cmath>  
  
using namespace std;  
int main(){  
    //declaring variables  
    float tolerance;  
    float x1, y1,z1; // coordinates of the first point(point on earth)  
    float x2, y2, z2;// coordinates of the second point(point of plane)  
    float dis; //the distance between the two points  
    double time = 0; // the time that will be calculated after each process  
    bool valid = true;//a boolean value to be used for looping  
    //looping to implement the process based on the value of "valid"
```

```
while(valid){
    //printing the options for the user
    cout<<"Choose your option:\n";
    cout<<"(1) Calculate the error between the two sensors recordings
and saving in the error file\n";
    cout<<"(2) Display the error file information\n";
    cout<<"(3) Quit the program\n";
    //reading the input value that the user will enter
    int num;
    cin>>num;
    //checking if the user entered one
    if(num == 1){
        //declaring two variables as strings to be the name files from
the user
        string name1 , name2;
        //asking the user to enter the names of the two files
        cout<<"Please Enter the Two file names"<<endl;
        //reading the input values of the two variables
        cin>>name1>>name2;
        //identifying files as input files for later use
        ifstream file1;
        // opening the files with names given by the user
        file1.open(name1);
        ifstream file2;
        file2.open(name2);
        //checking if they exist or not
        if(file1.fail() || file2.fail()){
            //printing error message if the files failed
            cout<<"Error in displaying the files\nplease try again\n";
            //reading the names entered by the user
            cin>>name1>>name2;
        }
        //declaring a err as output file
        ofstream err("Err.log");
        //checking if it failed or not
        if(err.fail()){
            //printing error message when the files failed to open
            cout<<"Error in displaying the file\n";
            exit(1);
        }
    }
}
```

```
}
//Asking the user to enter the tolerance
cout<<"Please Enter a non-negative tolerance: \n";
//Reading the value of the tolerance
cin>>tolerance;
while(tolerance){
    if(tolerance > 0)
        break;
    cout<<"please enter a non-negative tolerance: \n";
    cin>>tolerance;
}
/* if(tolerance < 0){
    cout<<"Please try again, tolerance cannot be negative\n";
    cin>>tolerance;
} */
//setting the table of the data to be printed
cout<<setw(10)<<"Distance:"<<setw(10)<<"Time: "<<endl;
//looping to the end of the file to get the data stored there
while(!file1.eof() && !file2.eof()){
    //readint the coordinates from the files
    file1>>x1>>y1>>z1;
    file2>>x2>>y2>>z2;
    //calculating the distance
    dis = sqrt(pow((x1-x2),2)+pow((y1-y2),2)+pow((z1-z2),2));
    //checking if the distance is greater than tolerance or not
    if(dis > tolerance){
        //incrementing the time by quarter
        time += 0.25;
        //printing the results in the error file
        err<<dis<<setw(10)<<time<<endl;
        //printing the results in the consol
        cout<<setw(10)<<dis<<setw(10)<<time<<endl;
    }
}
//assigning false to valid variable to stop the loop
valid = false;
} //when the user chooses number "2"
else if(num == 2){
    //changing the stream of error file to be an input file
```

```
    ifstream display("Err.log");
    //checking if the file failed to open
    if(display.fail()){
        //printing error message
        cout<<"Failed to open the files\n";
    }
    //declaring two variables one for distance and one for time
    double number,sec;
    //looping through the files to get the data from it
    while(display>>number>>sec){
        cout<<setw(5)<<number<<setw(5)<<sec<<endl;
    }
    break;
} //when user chooses option 3
else if(num == 3){
    cout<<"Quitted"<<endl;
    valid = false;
} else{
    cout<<"Invalid input, please Enter a correct number"<<endl;
    cout<<"(1) Calculate the error between the two sensors
recordings and saving in the error file\n";
    cout<<"(2) Display the error file information\n";
    cout<<"(3) Quit the program\n";
    cin>>num;
}
}
return 0;
}
```

**Step 5: Tests and Verification (and Debugging):****1. Test1: Entering invalid input(rather than the inputs in the menu):**

```

Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
5
Invalid input, please Enter a correct number
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program

```

- **Test 2: Option 1 Calculation**

```

Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
1
Please Enter the Two file names
file1.txt
file2.txt
Please Enter a non-negative tolerance:
4
Distance:      Time:
5.19615        0.25
19.0526        0.5
9.3397         0.75
14.8492        1

```

- **Test 3: Option 2 Calculation**

```

Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
2
5.19615 0.25
19.0526 0.5
9.3397 0.75
14.8492 1

```

- **Test 4: Quitting the program**

```
Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
3
Quitted
```

- **Test 5: Entering negative tolerance**

```
Choose your option:
(1) Calculate the error between the two sensors recordings and saving in the error file
(2) Display the error file information
(3) Quit the program
1
Please Enter the Two file names
file1.txt
file2.txt
Please Enter a non-negative tolerance:
-2
please enter a non-negative tolerance:
-2
please enter a non-negative tolerance:
-2
please enter a non-negative tolerance:
-2
3
Distance:      Time:
5.19615       0.25
```

- **Data in file1 and file2:**

file1.txt	file2.txt
media > omar > 3642BD0B42BCD13B > Co	media > omar > 3642BD0B42BCD13B >
1 1 2 3	1 4 5 6
2 12 13 14	2 1 2 3
3 0 0 1	3 0 1 4
4 -6.3 -1.3 -2.5	4 -6 -8 -9
5 3.3 5.5 4.4	5 -3 -5 -4
6 3 1 6.645751311	6 0 1 4
7	

- **Validity of the program:**

- The menu accepts only three options: **1, 2, 3**.

- When the distance is **less than** the tolerance, the program will display

- “The distance is less than the tolerance, please try again”**

- All variables have the same data type (**double**) and it accepts up to **1.7E +**

- 308** except the value of the options as it ranges from **1 to 3** so it's **integer**.

- The coordinates of the points can accept **negative numbers**.

- The names of the files should be **file1.txt and file2.txt**

- The tolerance should be positive.