

**Kingdom of Saudi Arabia**

**Ministry of Education**

**King Faisal University**

**College of Computer Sciences & Information Technology**

**Analysis and Design of Algorithms**

**Project: Simulator for Air Traffic Controller**

**Phase II**

**by**

Khaled Al Mana; 214033090

Salman Al Quaimi; 214013885

Abdulaziz Al Maglouth; 214016619

**Instructor**

Dr. Asrar Haque

**November 2016**

**ACKNOWLEDGEMENT**

I would like to thank Allah first then, our instructor Dr.Asrar Al Haque who guided us how to search and develop our mathematical skills. Thankfully, problem solved through the hard working, we have improved our mathematical skills, and designing better algorithms.

**UNDERTAKING**

This work has been done by us (students below.), for the subject Algorithms Analysis and Design course CS311 offered at Computer Science Department, College of Computer Sciences and Information Technology, King Faisal University. Equations and formulas that has been used is according to what we have studied.

All the design, analysis and development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Student 1: Khaled Al Mana; 214033090

Student 2: Salman Al Quaimi; 214013885

Student 3: Abdulaziz Al Maghlouth; 214016619

Note: sign across your name

**ABSTRACT**

*This is Algorithm design and analysis project phase 2 to show if the two airplanes have safe distance between each other. This project will solve a real life problem occurred in 1996 between the two airplanes Kazakhstan flight 1907 became very close to the Saudi Arabian flight 763[1].*

Table of Contents

[LIST OF TABLES vi](#_Toc471811894)

[1. Introduction 1](#_Toc471811895)

[2. Mathematical Model 1](#_Toc471811896)

[*2.1* *Assumptions* 1](#_Toc471811897)

[*2.2* *Abbreviations and Acronyms* 1](#_Toc471811898)

[*2.3* *Equations* 2](#_Toc471811899)

[3. Designed Algorithm 3](#_Toc471811900)

[*3.1* *Approach* 3](#_Toc471811901)

[*3.2* *Algorithm* 3](#_Toc471811902)

[*3.3* *An Example* 6](#_Toc471811903)

[4. Analysis of Algorithm 7](#_Toc471811904)

[Time Complexity 7](#_Toc471811905)

[Space Complexity 7](#_Toc471811906)

[5. Implementation Details 7](#_Toc471811907)

[6. Testing the Simulator 8](#_Toc471811908)

[7. Conclusion 8](#_Toc471811909)

[Major Contributions 9](#_Toc471811910)

[Phase I: 9](#_Toc471811911)

[Khaled Al Mana 9](#_Toc471811912)

[Salman Al Quaimi 9](#_Toc471811913)

[Abdulaziz Al Maghlouth 9](#_Toc471811914)

[Phase II: 10](#_Toc471811915)

[Khaled Al Mana 10](#_Toc471811916)

[Salman Al Quaimi 10](#_Toc471811917)

[Abdulaziz Al Maghlouth 10](#_Toc471811918)

[References 10](#_Toc471811919)

# LIST OF TABLES

Table ‎2.1 The 7 SI Base Units 1

# Introduction

There are over 100,000 flights every day [2], but we haven’t thought of those airplanes and how they are scheduled. One time in 1996, Kazakhstan flight 1907 became very close to the Saudi Arabian flight 763 till they hit. The problem is how to show if the distance between airplanes are safe, to avoid such horrible accidents.

# Mathematical Model

In this section equations have been developed in order to solve the problem will be shown.

## *Assumptions*

To solve the safe distance between airplanes, we have assumed that the airplane fly in a straight line. Which in reality, it does not. Airplanes fly in curve way like an arc. Also, Airplanes coordination, speed, destinations assumed to be given.

## *Abbreviations and Acronyms*

Units used in Algorithms and example are in the following table.

Table ‎2.1 The 7 SI Base Units

|  |  |  |
| --- | --- | --- |
| **Unit** | **Symbol** | **Quantity** |
| Kilometer | km | length |
| second | s | time |
| Hour | H | time |
| meter | m | length |

## *Equations*

These are some important variable you need to know:

d = the distance between any two plane whether it is height or width

X2 is x-coordination of an airplane

X1 is x-coordination of another airplane

Tix is x-coordination of a terminal

Aix is x-coordination of airplane

In this project, we have used the following equations to solve the problem.

2. Line Equation = ( , - )
4. We used Pythagoras’s formula [3] to find the distance between two planes.
5. we use speed, time and distance equation to find the time required to reach to intersection point (if exist).
6. To help finding the equations x, y, z of the line.
7. To find the equations x, y, z to find the line equation.
8. In the end, the two airplanes are in safe distance if the distance between these two plane are greater than 100 kilometers (in terms of x, y, z), also if the time needed to reach to the intersection point is greater than 10 minutes these two airplanes.

# Designed Algorithm

In this section, we will discuss the approach used to design the AlretDistance(A[N]) algorithm.

## *Approach*

We have designed the algorithm based on the idea of brute force approach, to show whether an airplane is too close to the another or not by checking every airplane with other airplanes.

## *Algorithm*

// Safe Distance Algorithm AlertDistance(A[N],T[N],S[N])

//Input: Let Array P airplanes with x, y and z coordination, Array T terminals with x, y and z coordination, and Array S speed of each airplane.

//Output: Alert if two airplanes are not in safe distance.

*AlertDistance(A[N],T[N],S[N])*

*dmin 🡨 100 Km*

*d = 0, d1=0, d2=0 // C= intersection point*

*for each 5 Minutes do*

*for i 🡨 0 to n-2 do*

*for j 🡨 i+1 to n-1 do*

*If d < dmin*

*Intersection (Ai, Aj, Ti, Tj)*

*else*

*Show “Safe”*

*Intersection (Ai, Aj, Ti, Tj) {*

*// find if there is intersection and show if it is dangerous.*

*R1  = ( , - ) // Line 1*

*x1 =*

*y1 =*

*z1 =*

*R2  = ( , - ) // Line 2*

*x1 =*

*y1 =*

*z1 =*

*x2 =x1*

*Solve the equation to find the value of t.*

*Find the value of u1 in y and u2 in z.*

*If u1 =u2*

*Find c by replace u1 in x2 y2, z2.*

*SpeedNTime (C, Ai, Aj, Si, Sj)*

*else*

*Show “Danger! Airplane A, and B are next to each other.”*

*}*

*SpeedNTime (C, Ai, Aj, Si, Sj){*

*// Find the time of airplanes intersection point.*

*if t1 = t2*

*Show “two airplanes will intersect in the same time.”*

*else if t2 -t1 < 10 min*

*Show “two airplanes will intersect in difference of time.”*

*else*

*Show “Safe distance between these two planes”*

*}*

## *An Example*

This will work in an example of two airplanes heading to different destinations where those two airplanes intersect in some point of their route. So, before the intersection point, a danger alert should be shown.

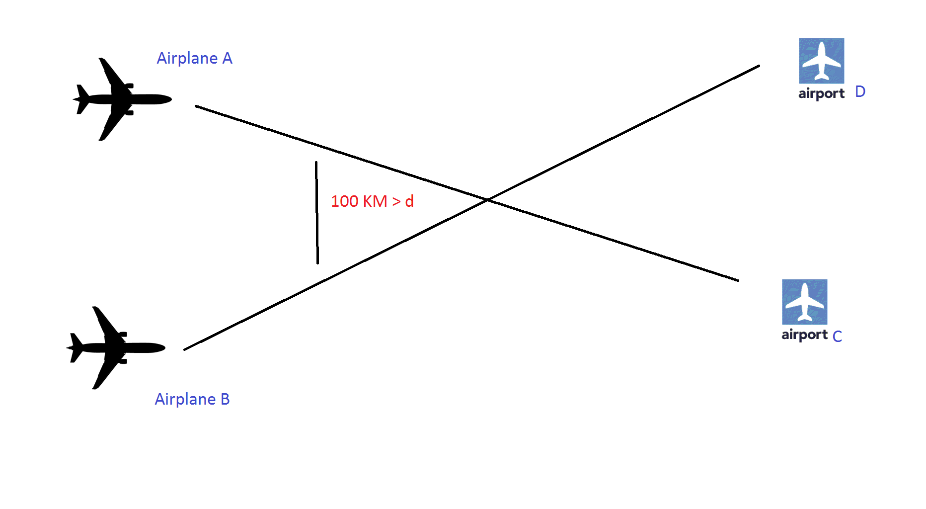


Figure 3.3

For example: suppose we have 2 dimensional space, airplane1 (2,2) flies to terminal1 (6,3), airplane2(2,4) flies to terminal2 (5,8), the algorithm will show that the two airplane will intersect in (0.157,1.539). We find that the distance between airplane1 and the intersection point is 1.899 and the distance between airplane2, and the intersection point is 3.074. To calculate the Z time, we divide the distance by speed of that airplane. So the time need for airplane 1 to reach to the intersection point is 1.266 minutes (assume the speed is 1.5km/h) and the time need for airplane2 to reach to the intersection point is 2.049 minutes. In this situation the two airplane are in danger because the difference of time is less than 10 minutes.

**Note: The algorithm is not restricted to these numbers.**

# Analysis of Algorithm

## Time Complexity

1. Input size: n
2. Basic operation: comparison
3. Best, worst and average are same
4. Let c(n) is total number of comparison

So, after simplifying this equation using useful summation formulas, rules, and since we have two loops we found that

## Space Complexity

Since we have 3 arrays(A[n] planes, T[n] terminals, S[n] plane’s Speed) and group of variables (c, d, d1, d2, i, j, dmin, R1, R2, x1, y1, z1, x2, y2, z2, u1, u2, t1, t2), we have

3n

# Implementation Details

We have used a “Java programming language” to implement the algorithm and the simulation of the airplanes. We believe it would be better to implement it with python, but we have used “Java” since it is an object oriented programming and we’re all familiar with.

# Testing the Simulator

We are assuming that the data is already available from air traffic. Input is read from text files and the program will display if there is a safe distance between any two given airplanes or not. We have assumed multiple inputs and tested it theoretically, as well as we have shown that the results are correct. Then, we have tested the same data in the simulation, and we have got the same result in the theoretical. Finally, by doing this we have proven that test of the simulation has passed well.

# Conclusion

Considering the real life situation to such problem, and how technical work could be so sensitive to the human beings lives through transportation. As problem solvers, we had to think about many phases and look from many perspectives. We believe that this problem has many solutions, but we cannot say that we have designed the optimal algorithm. Considering to what we have studied in this semester and learned by ourselves during this short time, we have given our best developing this algorithm and we believe it can be improved. Also, this algorithm may be used in many other solutions with some modifications like space work. Finally, this phase has mathematical formulas that can help to develop many aircraft applications. We are thankful to do such work.

# Major Contributions

## Phase I:

## Khaled Al Mana

Assigning the variables of the algorithm and developing better equations.

Analysis the real problem and how to avoid the crash to keep by keeping safe distance of each plane.

Implementing the algorithm.

## Salman Al Quaimi

Finding the best Equation of the algorithm to find the intersection points of two airplanes.

Analysis the Equation and understanding how the crash will appear.

Implementing the algorithm.

## Abdulaziz Al Maghlouth

Designing the Structure of the Algorithm and how to analyze the equation.

Find the time and the speed of the planes to reach to some point.

Implementing the algorithm.

## Phase II:

## Khaled Al Mana

* GUI
* Input from files
* Problem Solver

## Salman Al Quaimi

* Equations
* Animation
* Simulation Debugger

## Abdulaziz Al Maghlouth

* Logs and Messages
* General Debugger
* Code Tidying

# References

[1] - <https://aviation-safety.net/database/record.php?id=19961112-0>

[2]- <http://www.garfors.com/2014/06/100000-flights-day.html>

[3]- <https://en.wikipedia.org/wiki/Pythagorean_theorem>