



# Aircraft Collision Avoidance Controller

Presenters: **Adit Negi, Jay Barot**

CS6376: Hybrid and Embedded Systems

Instructor: Professor Abhishek Dubey

Vanderbilt University, Fall 2023



# Introduction



The **traffic collision avoidance system (TCAS)** has been a vital safety feature in large transport aircraft since the 1990s.



It has **proven** highly **effective** in preventing midair collisions.



However, as we look to the **future** of air traffic control and management, **challenges** arise.



**High-density airspace** and the integration of **unmanned aerial vehicles (UAVs)** pose new complexities.

# Related Work



The **Aircraft Collision Avoidance System X (ACAS X)** introduces a partially observed Markov decision process for future collision avoidance.



ACAS X focuses on reducing collision risk and false alarms, primarily issuing **vertical avoidance** actions due to computation and storage constraints.



Traditional radar-based systems face limitations in coverage, restricting implementation to close encounter scenarios and increasing pilot workload.



ACAS Xp, a future version, relies solely on **autonomous dependent surveillance-broadcast (ADS-B)** for broader application in general aviation aircraft.

# Autonomous Dependent Surveillance Broadcast (ADS-B)



ADS-B is a **vital component** of the U.S. **Federal Aviation Administration's** NextGen air transportation system.



It broadcasts state and trajectory intent information from navigation satellites to other aircraft and ground stations.



ADS-B's **broader surveillance coverage** compared to radar facilitates early conflict detection and resolution.

**Challenges** arise in adapting the current TCAS for ADS-B data, leading researchers to explore new approaches.



*Holdsworth et al.* suggest collision avoidance planning with ADS-B and dynamic programming.

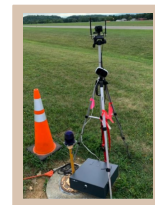
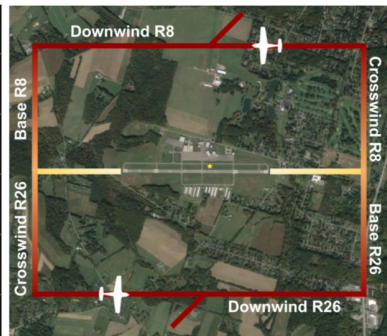
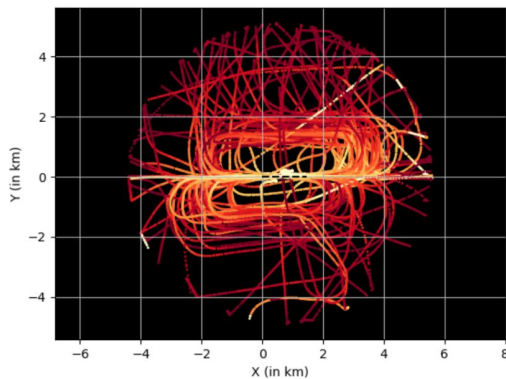


*Kochenderfer et al.* propose using a partially observable Markov decision process to validate ADS-B reports in the collision avoidance system.



*Lin et al.* present a sampling-based path planning method using ADS-B to avoid collisions with commercial aircraft.

# Dataset

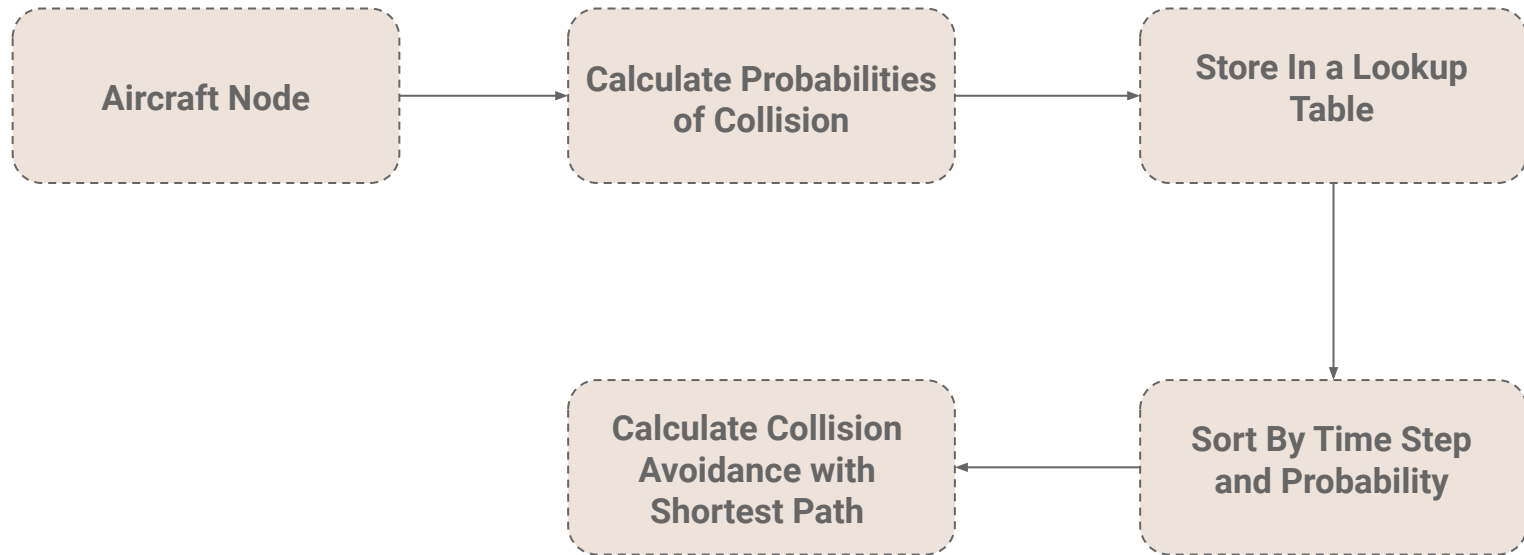


Example of data collected by ADS-B Receiver

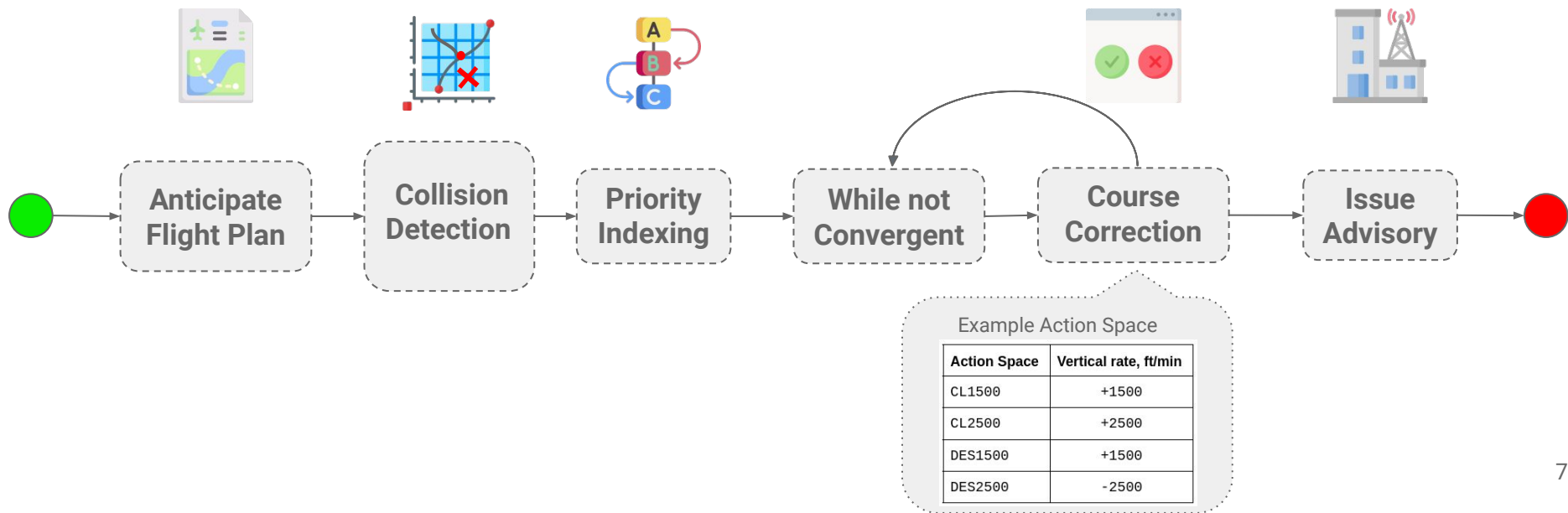
Frame #	Aircraft ID	x (km)	y (km)	z (km)	wind <sub>x</sub> (m/s)	wind <sub>y</sub> (m/s)
0	10620674	1.3407	0.0026	0.3353	0.0	0.0
1	10620674	1.3135	0.0021	0.3353	0.0	0.0
2	10620674	1.2863	0.0017	0.3353	0.0	0.0
⋮	⋮	⋮	⋮	⋮	⋮	⋮
405	10620674	-3.8946	1.5872	0.9751	0.0	0.0

**Carnegie Mellon University**  
The Robotics Institute

# Collision Detection



# Algorithm Flowchart





# Output Walkthrough

```
# Running controller
-----
# ADS-B Controller Receiver Data Info:
-----
Dataset rows: 360722
Unique aircrafts: 164
-----
# Controller Data Area Info:
-----
Plane data range: Along the runway: -4.131442787589634 to 5.863296403919421 km
Plane data range: Perpendicular to the runway: -4.881616372506984 to 5.0951506539017 km
System height data range: 0.0 to 1.82118 km
-----
```

1

```
# Controller Iteration: 1
-----
# Monitoring Collisions
-----
Collision # 1
      Frame AircraftID      x      y      z      windx      windy
86216  17138.0  11217673.0  1.762349 -0.00078  0.3048 -3.341066 -1.343522
103469 17138.0  11080625.0  1.757077  0.00211  0.3048  0.000000  0.000000
-----
Collision # 2
      Frame AircraftID      x      y      z      windx      windy
104676 24113.0  10628960.0  1.119112 -0.001655  0.33528  1.914331  1.718007
159338 24113.0  10520887.0  1.124637  0.004383  0.33528  0.798856  2.445003
-----
Collision # 3
-----
```

2

```
# Collision Monitor Summary
-----
Total number of collisions: 10
10 occurrences of 2 aircrafts collisions.
-----
# Setting priority order...
# Trajectory correction...
# Updating trajectory data...
```

3

```
# Controller Iteration: 2
-----
# Monitoring Collisions
-----
Collision # 1
      Frame AircraftID      x      y      z      windx      windy
225749 24113.0  10628960.0  1.119112 -0.001655  0.94488  1.914331  1.718007
351707 24113.0  10520887.0  1.124637  0.004383  0.94488  0.798856  2.445003
-----
```

4

```
# Collision Monitor Summary
-----
Total number of collisions: 1
1 occurrences of 2 aircrafts collisions.
-----
# Setting priority order...
# Trajectory correction...
# Updating trajectory data...
```

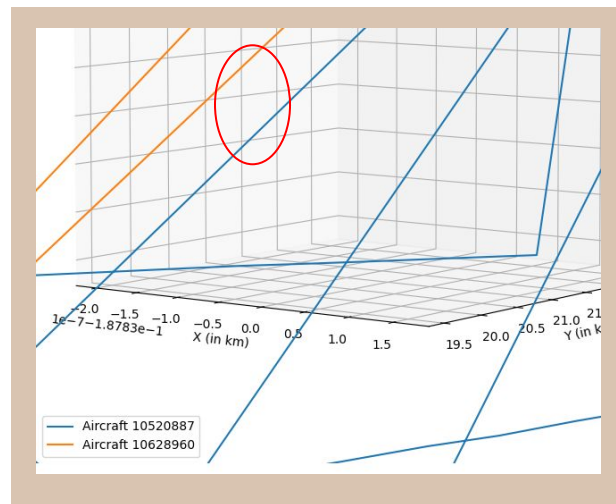
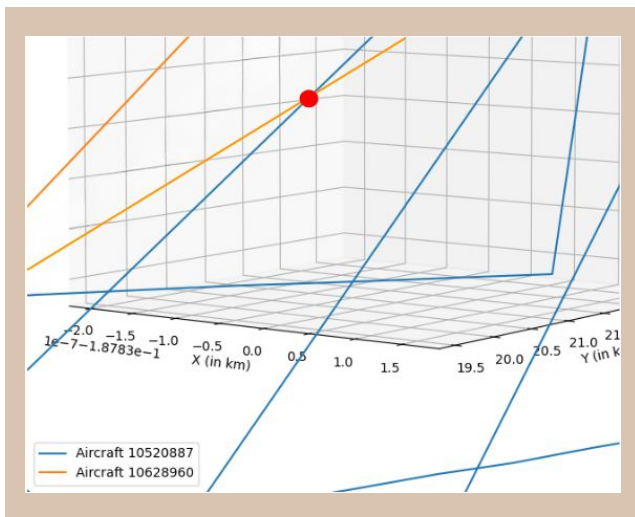
5

```
# Controller Iteration: 3
-----
# Monitoring Collisions
-----
# No more collisions found: Course correction completed
-----
# Controller finished running
```

6



# Visualization





```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Vectors
{
    class Program
    {
        static void Main(string[] args)
        {
            // Erstelle ein Array von 50 Personen
            Person[] personen = new Person[50];

            // Fülle das Array mit 50 Personen
            for (int i = 0; i < personen.Length; i++)
            {
                personen[i] = new Person(i * 10 + 1, "Person", i * 10 + 1, i * 10 + 1, i * 10 + 1, i * 10 + 1, i * 10 + 1, i * 10 + 1);
            }

            // Gehe über alle Personen und drucke sie aus
            foreach (var person in personen)
            {
                Console.WriteLine(person.ToString());
            }
        }
    }

    class Person
    {
        public int ID;
        public string Name;
        public int Firma;
        public int Geburtsdatum;
        public int Geburtsort;
        public int Höhe;
        public int Gewicht;
        public int Alter;
        public string Geschlecht;

        public Person(int ID, string Name, int Firma, int Geburtsdatum, int Geburtsort, int Höhe, int Gewicht, int Alter, string Geschlecht)
        {
            this.ID = ID;
            this.Name = Name;
            this.Firma = Firma;
            this.Geburtsdatum = Geburtsdatum;
            this.Geburtsort = Geburtsort;
            this.Höhe = Höhe;
            this.Gewicht = Gewicht;
            this.Alter = Alter;
            this.Geschlecht = Geschlecht;
        }

        public override string ToString()
        {
            return $"ID: {ID}, Name: {Name}, Firma: {Firma}, Geburtsdatum: {Geburtsdatum}, Geburtsort: {Geburtsort}, Höhe: {Höhe}, Gewicht: {Gewicht}, Alter: {Alter}, Geschlecht: {Geschlecht}";
        }
    }
}
```

Name	Firma	Geburtsdatum	Geburtsort	Höhe	Gewicht	Alter	Geschlecht
Personen 1 bis 10							
Person 1	1	1	1	1	1	1	1
Person 2	2	2	2	2	2	2	2
Person 3	3	3	3	3	3	3	3
Person 4	4	4	4	4	4	4	4
Person 5	5	5	5	5	5	5	5
Person 6	6	6	6	6	6	6	6
Person 7	7	7	7	7	7	7	7
Person 8	8	8	8	8	8	8	8
Person 9	9	9	9	9	9	9	9
Person 10	10	10	10	10	10	10	10
Personen 11 bis 20							
Person 11	11	11	11	11	11	11	11
Person 12	12	12	12	12	12	12	12
Person 13	13	13	13	13	13	13	13
Person 14	14	14	14	14	14	14	14
Person 15	15	15	15	15	15	15	15
Person 16	16	16	16	16	16	16	16
Person 17	17	17	17	17	17	17	17
Person 18	18	18	18	18	18	18	18
Person 19	19	19	19	19	19	19	19
Person 20	20	20	20	20	20	20	20
Personen 21 bis 30							
Person 21	21	21	21	21	21	21	21
Person 22	22	22	22	22	22	22	22
Person 23	23	23	23	23	23	23	23
Person 24	24	24	24	24	24	24	24
Person 25	25	25	25	25	25	25	25
Person 26	26	26	26	26	26	26	26
Person 27	27	27	27	27	27	27	27
Person 28	28	28	28	28	28	28	28
Person 29	29	29	29	29	29	29	29
Person 30	30	30	30	30	30	30	30
Personen 31 bis 40							
Person 31	31	31	31	31	31	31	31
Person 32	32	32	32	32	32	32	32
Person 33	33	33	33	33	33	33	33
Person 34	34	34	34	34	34	34	34
Person 35	35	35	35	35	35	35	35
Person 36	36	36	36	36	36	36	36
Person 37	37	37	37	37	37	37	37
Person 38	38	38	38	38	38	38	38
Person 39	39	39	39	39	39	39	39
Person 40	40	40	40	40	40	40	40
Personen 41 bis 50							
Person 41	41	41	41	41	41	41	41
Person 42	42	42	42	42	42	42	42
Person 43	43	43	43	43	43	43	43
Person 44	44	44	44	44	44	44	44
Person 45	45	45	45	45	45	45	45
Person 46	46	46	46	46	46	46	46
Person 47	47	47	47	47	47	47	47
Person 48	48	48	48	48	48	48	48
Person 49	49	49	49	49	49	49	49
Person 50	50	50	50	50	50	50	50



Demo Link: <https://youtu.be/RvGEqUlp-sI>



# Thank You