ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

AERO-PAD : THE NEXT GENERATION AIR TRAFFIC CONTROL SYSTEM FOR MANAGING INDIAN AIRSPACE

Gurprit Singh[1*] and [2]Ampu Harikrishnan

[1]Research scholar, School of Business Studies, Himgiri Zee University, Sherpur, Chakrata Road, Dehradun, Uttarakhand, India,

[2]Registrar, Himgiri Zee University Dehradun, Uttarakhand, India

[1]mguru25@gmail.com and [2]registrar@hzu.in

ABSTRACT

Many airports still rely on traditional paper traffic sheets or slides for their air traffic controllers to use on a daily basis. Before beginning their shift, air traffic controllers take these paper sheets and manually copy all the necessary details from the flight plans available in the flight display processing system using a pen and pencil. The information they copy includes call signs, transponder codes assigned to each aircraft, flight levels, and estimated time of arrival at various reporting points within their control area or airport. They then organize and mark up this information on their paper sheets to match the various air traffic routes, and use these sheets to handle traffic throughout their shift. They update the information on their sheets as they receive input from pilots, and pass the relevant details to their assistant. They also communicate with neighboring ATC units as per the signed agreement.

However, a new prototype called AeroPad will soon replace these paper sheets with an iPad. With this electronic pad, air traffic controllers will still be able to manually input and organize all the necessary details as before, but with the added benefit of automatic error correction for any mistakes made due to excessive workload and pressure.

Keywords: Aero-pad, Traffic Sheets, Call Signs, transponder codes, ATC, iPad.

1. INTRODUCTION

The AeroPad is an electronic air traffic management system designed to replace the traditional slides and traffic sheets used by air traffic controllers in Indian airspace management for the past 70 years. AeroPad's user interface uses various technologies, including HTML, CSS, Python, and JavaScript. The real-time storage and retrieval of aircraft positions is made possible by MongoDB, an NoSQL database platform, along with various tools and frameworks. In addition, the development of the aero-pad leverages sophisticated conflict detection algorithms. This innovative system can autocorrect incorrect levels, routes, headings, and speeds, reducing the workload and stress for controllers and assistants. It also enhances safety by automatically writing continuous headings in degrees and changes in route permissions, which can be visualized in radar for cross-checking.

The AeroPad significantly reduces the errors made by air traffic controllers, assistants, and junior air traffic controllers while adhering to International Civil Aviation Organization (ICAO) separations and standards. The literature reports that dynamic functions detect conflicts, collision risks, and zone entry risks in aircraft during grounding periods, ensuring safety and efficiency with a call for technological interventions to fortify the capabilities of ATC, ensuring precision, adaptability, and a reduced cognitive load on controllers.[1]

In the dynamic and high-stakes domain of air traffic control (ATC), where the skies are increasingly crowded, and safety is nonnegotiable, the AeroPad project has emerged as a groundbreaking solution poised to redefine the very fabric of air traffic management. The contemporary challenges faced by air traffic controllers (ATCs) extend beyond conventional capacities, demanding not only proficiency but also a technological ally that can alleviate their workload, enhance precision, and introduce a new era of operational efficiency. This paper provides details of the prototype and encourages further research to improve its usefulness and compatibility with the latest air traffic management technologies.

The presentation of data has been designed to be human-friendly, minimizing misunderstandings or misinterpretation of information related to individual aircraft. The device is also capable of storing data related to any incidents or accidents, and its primary function is to share these data with airports on a daily basis. It is important to note that the Aero-pad cannot work in isolation from any airport. Sharing information allows for better decision-making across all levels of airport staff, from senior management to ground crews.

As air travel continues to rapidly increase, the role of ATCs becomes more intricate, as ATCs can navigate through complex web of regulations, real-time aircraft data, and communication protocols. In this context, the aero-pad is not merely an application; it represents a paradigm shift—the integration of cutting-edge technology into an ATC-friendly operating system, that is meticulously designed to empower controllers and increase the safety and efficacy of air traffic management. Air traffic controllers are now using digital solutions instead of paper flight strips in their control rooms. [18]

This introduction seeks to immerse the reader in the challenges faced by ATCs, painting a vivid picture of the intricate responsibilities and critical decision-making processes involved. The aero-pad, a tool meticulously crafted to address the nuanced demands of contemporary air traffic control, has emerged as a beacon of innovation.

2. RESEARCH OBJECTIVES

- An aero-pad, an electronic air traffic management system, is proposed for controllers to input position values, flight levels, routes, headings, and route change instructions.
- Aero-pads have replaced the traditional slides and traffic sheets used in Indian air space management for the past 70 years.
- The AeroPad, an innovative technology, is designed to automatically correct any incorrect instructions given by an air traffic controller.
- Aero-pads aim to decrease human error-related aviation accidents, which account for 80% of all accidents.

2.1 STATEMENT OF THE PROBLEM

It is important to avoid making mistakes when inputting data or entering instructions for air traffic control. In a high-pressure environment, such as when answering phone calls from other stations while also managing air traffic, it can be easy to accidentally load incorrect information onto paper sheets. These errors can have serious consequences, including the loss of human life, billions of dollars, and even the closure of airlines.

3. LITERATURE REVIEW

The realm of air traffic control (ATC) has evolved due to the increasing complexities of modern aviation. The goal is to address voice communication issues, such as hear-back errors[3], to develop the most effective artificial intelligence system. The stored data can be utilized for studying by students or trainee controllers, or for accident analysis and training [4] during air traffic events.

Existing ATC systems, while stalwart in their reliability, grapple with the surging demands imposed by expanding airspaces and the relentless growth of air travel. We need to work on developing a simulation tool that can assess the impact of different factors on traffic safety. The tool should be based on self-diagnostic systems and fuzzy models [5], which can restore system performance even if the controller is not aware of any malfunction. The study aligns with EUROCONTROL and the European Commission's performance objectives to enhance ATM safety, network efficiency, aviation security, environmental objectives, and capacity and air transport growth.[6]

More features, such as separation assurance aid, such as the tactical controller tool (TCT); and the ground-based safety net short-term conflict detection alert (STCA), [6] will be studied and incorporated into this AeroPad. Heinz [7] highlighted the significance of dynamic data retrieval in enhancing decision-making and automated airspace concept (AAC), which is a requirement for next generation air traffic control systems.

This review delves into the landscape of ATC technologies, sheds light on the strengths and limitations of current solutions and paves the way for the introduction of AeroPad a transformative tool designed to navigate the intricacies of contemporary air traffic management. The next generation of air requires a reliable, efficient automated conflict resolution system, such as Conflict Alert, but no successful implementation has been achieved in operational applications over the past two decades.[13]

Historically, traditional ATC systems have been the backbone of airspace management, providing a stable foundation for safe and efficient air travel. This automation provides protection and capacity advantages in air traffic control management and significantly reduces the number of people needed to handle future traffic volumes.[15] The aim of the air traffic management system is to automate airspace monitoring and control, reducing the workload of controllers.

Enter AeroPad, a project conceived to fill this void by seamlessly integrating with an ATC-friendly operating system. This specialized operating system serves as the nexus, providing an intuitive interface for controllers to navigate the complexities of AeroPad's capabilities. The literature emphasizes the necessity of such user-centric designs to empower controllers to handle high-pressure situations and complex scenarios.

The conflict detection and resolution module within AeroPad encapsulates the pinnacle of its innovation. As the literature illuminates the challenges in maintaining ICAO separations and standards, AeroPad's module employs sophisticated algorithms to not only identify conflicting values but also suggest corrective actions in real time. This proactive approach aligns with the overarching theme of utilizing technology to mitigate errors and enhance safety.

In conclusion, the literature review serves as the backdrop against which the aeroPad emerged—a beacon of technological advancement in the ATC domain. The subsequent sections will delve into the intricacies of Aero-pad architecture and implementation, unravelling the threads of innovation woven into this transformative tool for air traffic management.

3.1 RESEARCH METHODOLOGY

AeroPad's user interface is built using a range of technologies, including HTML, CSS, Python, and JavaScript. The real-time storage and retrieval of aircraft positions is made possible by MongoDB, an NoSQL database platform, along with various tools and frameworks. In addition, the development of aero-pad leverages sophisticated conflict detection algorithms.

3.2 SCOPE OF WORK FROM THE RESEARCH GAP

Real-time databases have emerged as a cornerstone in modern ATC, ensuring that controllers have instantaneous access to updated information. The AcListant® project uses assistant-based speech recognition technology to automate voice commands for air traffic controllers, reducing manual input and workload. [18]

Air traffic management involves procedures for the safe and efficient operation of a nation's airspace, covering aspects such as Air Traffic Management (ATM) and Air Traffic Control (ATC).[19] However, as underscored by studies the next-generation air traffic control system will have to handle a traffic density two or three times that of today. The manual processes inherent in these systems, coupled with the exponential increase in air traffic, have exposed vulnerabilities.

Further research is required to prevent errors in traffic situation visualization systems (TSVSs) [20] and mitigate their negative impacts.

Aero pad takes this concept a step further by employing an API-driven architecture, connecting to a live database, and enriching the data landscape with real-time information, including aircraft positions, weather conditions, and flight plans.

3.3 The guidance material should be considered when designing the prototype according to ICAO Doc 9426.

When creating a prototype, it is important to consider human factors principles for displaying information and transmitting data. If data are manually entered into ATC automation systems, they should follow human-centered design principles [26.] Each flight should have at least one FPS [26], and the number of frames per second needs to meet ATS requirements. Air traffic controllers should receive and retain data promptly for at least 30 days. It is important to specify procedures for annotating and storing data. Sharing daily incident data with airports is crucial for making informed decisions. Finally, electronic recording and preservation of flight progress and coordination information requires at least the same duration.

3.4 System Architecture

Aero-pad system architecture is a sophisticated interplay of components, that are carefully orchestrated to meet the demanding challenges of modern air traffic control (ATC). The architectural design revolves around seamless integration, real-time data flow, and user-centric interfaces. The following detailed breakdown elucidates each component within the AeroPad framework:

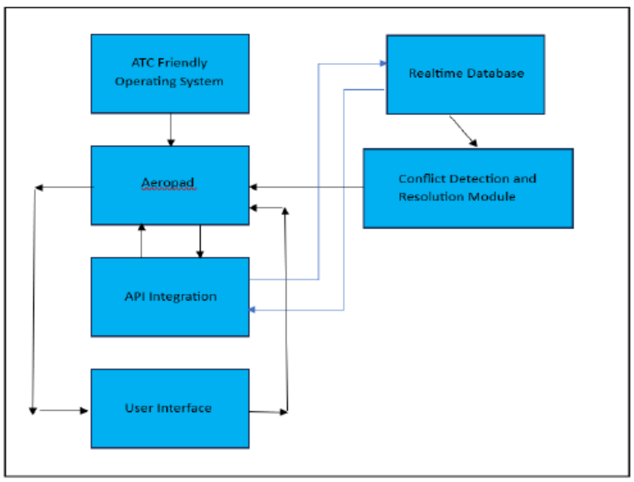

Fig. 1— Aero pad Framework

3.4.1 COMPONENTS

A. ATC-Friendly Operating System:

Description: At the core of the AeroPad lies an operating system specially crafted for air traffic controllers. This ATC-friendly OS provides a range of applications including AeroPad, which is tailored to the unique needs of controllers, and offer intuitive controls, streamlined navigation, and visualizations that facilitate quick decision-making.

B. Real-Time Database:

Description: The real-time database serves as the central repository of critical information essential for air traffic control. It encompasses real-time aircraft positions, weather conditions, flight plans, and other dynamic data. This database is continually updated, ensuring that controllers have the most accurate and current information at their fingertips.

C. API Integration:

Description: The API integration layer of the AeroPad is the gateway to external data sources. Through this component, aero-pad interfaces with APIs from aviation authorities, weather services, and other relevant sources exist. This dynamic integration enriches the system's dataset, providing comprehensive real-time information beyond the immediate airspace.

D. Conflict Detection and Resolution Module:

Description: The conflict detection and resolution module represents the analytical powerhouse of the AeroPad. This module employs advanced algorithms to scrutinize ICAO separations and parameters set for a specific set of aircraft in real time. It dynamically cross-referenced these values with the current parameters assigned to all other aircraft in the airfield, pinpointing conflicts and proposing corrective actions.

E. User interface:

Description: The User Interface acts as the interface through which air traffic controllers interact with the AeroPad. It translates complex data and analyses into comprehensible visualizations, providing real-time updates on aircraft positions, potential conflicts, and other pertinent information. The user interface is designed with the aim of reducing cognitive load and enhancing situational awareness for controllers.

3.4.2 Key Elements

Element A. This aero-pad is electronic airspace management system (EAMS) introduced between an air traffic controller and air traffic assistants or an FDPS (flight data processing system).

Element B. The system keeps a log of all the positions assigned to the flights and whenever the controller adds a contradictory value to the preceding values, the mistaken value will be highlighted so that the controller can change it before its actual assignment and hence ensure the precise positioning of flights in the respective airspace.

Element C. The system allows the Air Traffic Controller to input position values such as call sign, departure squawk, flight levels, routes, headings, and route change instructions such as direct to reporting point or left of track by Nautical miles, ETA, and ETD.

Element D. The system is smart enough to suggest new values by itself relieving the controller in times of heavy traffic.

Element D1. The system enhances safety by writing the traffic electronically on this prototype at the same time it not only double-checks all the parameters a controller writes but also simultaneously corrects for level corrections or route clearance errors, if any.

Element E. The system stores all the assigned values in a digital encrypted database from which they can be accessed anytime in the future for reference.

Element F. The system advances safety as it writes continuous heading and change of route permission, such as direct to a reporting point or giving left-of-track permission or right -of-track permission to a pilot due to bad weather, which can be visualized in radar for cross checking.

3.4.3 Interactions

A. ATC-Friendly OS & User Interface Interaction:

The ATC-Friendly OS serves as the foundation, providing a user-friendly interface through which controllers interacts with the AeroPad. The User Interface layer ensures that controllers receive real-time updates, alerts, and visualizations, fostering efficient decision-making.

B. API Integration & Real-Time Database:

The API integration component facilitates seamless communication with external data sources, continually enriching the real-time database with the latest information. This dynamic integration ensures that Aero-pad's dataset remains comprehensive and up-to-date.

C. Conflict Detection and Resolution Module & Real-Time Database:

The conflict detection and resolution module relies on the real-time database to access current aircraft parameters. These data are dynamically analyzed to identify conflicts, ensuring the highest level of safety and compliance with aviation regulations.

3.4.4. Implementation

The realization of the AeroPad involves the thoughtful integration of technologies and functionalities to empower air traffic controllers with real-time data and advanced conflict detection capabilities.

➢ **Real-time Database Interaction**

In the backend, the AeroPad relies on MongoDB, an NoSQL database, for the real-time storage and retrieval of aircraft positions. The real-time database class encapsulates interactions, offering methods to update and fetch current aircraft positions dynamically.

```
“from pymongo import MongoClient


class RealTimeDatabase:
def __init__(self, database_url, database_name):
self.client = MongoClient(database_url)
self.database = self.client[database_name]


def update_aircraft_position(self, aircraft_id, new_position):
aircraft_collection = self.database['aircraft_positions']
aircraft_collection.update_one({'_id': aircraft_id}, {'$set': {'position': new_position}})


def get_aircraft_positions(self):
aircraft_collection = self.database['aircraft_positions']
return list(aircraft_collection.find())”
```

This code ensures that the database is a dynamic source of real-time information, which is essential for the functionality of the AeroPad.

➢ **API Integration with Flask:**

To enrich the AeroPad dataset, the API integration layer, built with Flask, connects to external data sources or APIs. The Flask application defines an endpoint (/api/get_aircraft_data) for fetching real-time aircraft data.

```
“from flask import Flask, jsonify, request


app = Flask(__name__)
@app.route('/api/get_aircraft_data', methods=['GET'])
def get_aircraft_data():
```

```
# Code to fetch real-time aircraft data from an external source or API
aircraft_data = fetch_aircraft_data()

return jsonify(aircraft_data)
def fetch_aircraft_data():
# Placeholder for external API call or data retrieval logic
# ...

# Example response
aircraft_data = {'id': 'ABC123', 'position': [latitude, longitude], 'altitude': 30000}
return aircraft_data

if __name__ == '__main__':
app.run(debug=True)"
```

This Flask application serves as a flexible interface, adapting to different external data sources as needed.

➢ **Conflict Detection Algorithm:**

The Conflict Detection Module within Aero-pad employs a sophisticated algorithm to identify potential conflicts between aircraft based on their positions. The ConflictDetection class encapsulates this logic, ensuring a robust and adaptive approach.

```
"class ConflictDetection:
def __init__(self):
pass

def detect_conflicts(self, aircraft_positions):
conflicts = []

for i, aircraft1 in enumerate(aircraft_positions):
for j, aircraft2 in enumerate(aircraft_positions):
if i != j and self.are_conflicting(aircraft1, aircraft2):
conflicts.append({'aircraft1': aircraft1['id'], 'aircraft2': aircraft2['id']})

return conflicts

def are_conflicting(self, aircraft1, aircraft2):
```

```
# Placeholder for conflict detection logic based on positions, speeds, etc.
# ...

# Example: Check if distance is less than a threshold
distance_threshold = 10000  # in meters
return self.calculate_distance(aircraft1['position'], aircraft2['position']) < distance_threshold

def calculate_distance(self, position1, position2):
# Placeholder for distance calculation logic
# ...

# Example: Euclidean distance for simplicity
return ((position1[0] - position2[0])**2 + (position1[1] - position2[1])**2)**0.5”
```

This modular design ensures that the conflict detection logic can be adapted to evolving aviation requirements.

➢ **User Interface (HTML, CSS, JavaScript):**
The User Interface, developed with HTML, CSS, and JavaScript, serves as the visual gateway for air traffic controllers. It utilizes frameworks such as Leaflet for interactive map visualizations.

```
“<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Aero-pad - Air Traffic Control</title>
<!-- Include necessary CSS frameworks/styles -->
<link rel="stylesheet" href="styles.css">
</head>
<body>
<!-- Main content and visualization components -->
<div id="map"></div>

<!-- Include necessary JavaScript libraries and scripts -->
<script src="leaflet.js"></script>
```

```
<script src="Aero-pad.js"></script>

</body>

</html>"
```

This user-friendly interface ensures that controllers can intuitively interact with the AeroPad, receiving real-time updates on aircraft positions and potential conflicts.

This blend of technologies and functionalities culminates in comprehensive implementation, transforming the AeroPad from a concept to a practical tool poised to enhance air traffic control. The subsequent section will delve into the results and analysis, providing insights into the real-world impact of the AeroPad.

3.5. Aero-pad's application output results:

The implementation of the AeroPad showcases compelling results through a technical lens, offering enhanced air traffic control capabilities and robust data-driven functionalities.

3.5.1 Real-Time Data Dynamics

Aero-pad seamless interaction with MongoDB facilitates a dynamic and responsive real-time data environment. Controllers experience instantaneous updates of aircraft positions, as demonstrated in the output screen below:

Output Screen - Real-Time Data Dynamics:

```
Aircraft ID | Current Position
------------|----------------
ABC123      | [37.7749, -122.4194]
XYZ789      | [34.0522, -118.2437]
LMN456      | [41.8781, -87.6298]
```

Fig. 2 —This output illustrates Aero-pad's ability to provide up-to-the-moment aircraft positions, which is crucial for precise air traffic management.

3.5.2 An Enriched Dataset through API Integration

The Flask-based API integration layer enriches the AeroPad to dynamically fetch data from external sources, expanding its dataset. Below is an example output, showcasing enriched aircraft data:

Output Screen - Enriched Dataset through API Integration:

```
Aircraft ID | Current Position     | Altitude | Weather Conditions
------------|----------------------|----------|-------------------
ABC123      | [37.7749, -122.4194] | 30000 ft | Clear Sky
XYZ789      | [34.0522, -118.2437] | 28000 ft | Light Rain
LMN456      | [41.8781, -87.6298]  | 32000 ft | Overcast
```

Fig. 3 — This output highlights the integration of additional information, such as altitude and weather conditions, enriching the dataset for informed decision-making.

3.5.3. Proactive Conflict Detection:

The Conflict Detection Module's algorithmic process is evident in the following output, where potential conflicts are identified and recommended actions are provided:

Output Screen - Proactive Conflict Detection:

```
Conflict ID | Aircraft 1 | Aircraft 2 | Recommended Action
------------|------------|------------|--------------------
001         | ABC123     | XYZ789     | Altitude Adjustment
002         | LMN456     | XYZ789     | Change Course
```

Fig. 4 —This output demonstrates AeroPad's proactive approach, alerting controllers to potential conflicts and suggesting corrective actions for immediate resolution.

3.5.4. User-Friendly Interface

The User Interface, realized with HTML, CSS, and JavaScript, is a visually intuitive platform. The output screen below exemplifies the user-friendly map interface with aircraft positions:

Output Screen - User-Friendly Interface:

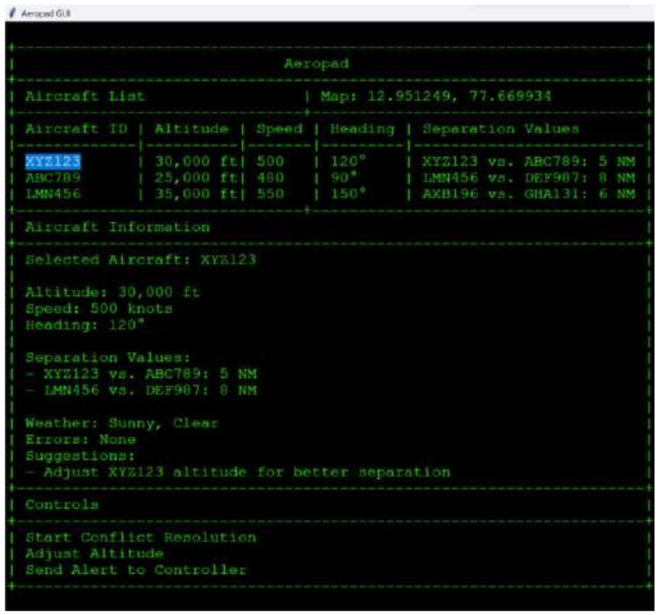

Fig. 5 — This output visualization reflects AeroPad's commitment to providing controllers with a visually comprehensible and responsive interface for efficient data assimilation with keyboard operations and shortcuts limiting the use of other computer peripherals.

3.5.5 Impact on Decision-Making:

The influence of the AeroPad on decision-making is evident in the accelerated response times and optimized safety measures. The following output screen illustrates the decision-support information, provided by empowering controllers:

Output Screen - Decision-Support Information:

Decision ID	Time Stamp	Decision
001	2024-01-30 12:45:00	Altitude Adjustment
002	2024-01-30 12:50:00	Change Course

Fig. 6 — This output exemplifies AeroPad's impact on decision-making processes, fostering quicker responses to emerging scenarios.

3.5.6 Adaptability for Future Enhancements

Aero-pad modular architecture facilitates seamless adaptability for future enhancements. The API integration layer serves as a gateway for incorporating new data sources, ensuring that the Aero-pad remains at the forefront of technological advancements. This adaptability is crucial for evolving alongside the dynamic aviation landscape.

In summary, the technical analysis and output screens shown in Fig.5 affirm that AeroPad's implementation significantly improves air traffic control operations, providing a foundation for safer, more efficient, and adaptable airspace management. The subsequent section will present recommendations for potential future developments and refinements to further augment AeroPad's capabilities.

The development and implementation of the AeroPad represent pioneering advancements in the realm of air traffic control technology. This section delves into a detailed and technical conclusion, summarizing key achievements, highlighting technological advancements, and envisioning the future trajectory of the AeroPad.

4. Technological Advancements Achieved

Aero-pads stand as a testament to the successful integration of cutting-edge technologies, each contributing to the overall efficacy of the tool:

4.1.1 MongoDB Integration for Real-Time Data Dynamics

The seamless interaction with MongoDB ensures that AeroPad's database remains a dynamic and responsive source of real-time aircraft positions. The use of MongoDB's document-oriented architecture facilitates flexible and efficient storage of complex data structures, enhancing the tool's adaptability to evolving requirements.

4.1.2 Flask-Based API Integration for Dataset Enrichment

The Flask-based API integration layer enables the AeroPad to dynamically fetch data from external sources, enriching its dataset. The modular design of the API layer allows for easy adaptation to diverse data streams, paving the way for future integrations with emerging aviation data sources.

4.1.3 Sophisticated conflict detection algorithms

AeroPad's conflict detection module employs sophisticated algorithms to proactively identify potential conflicts between aircraft. The algorithmic approach considers multiple parameters, including aircraft positions and predefined thresholds, contributing to a high level of accuracy in conflict identification.

4.2 Impact on Air Traffic Control Operations

AeroPad's implementation has demonstrable impacts on air traffic control operations, ushering in a new era of efficiency, safety, and adaptability:

4.2.1 Real-Time Data Empowerment:

Controllers benefit from instantaneous updates of aircraft positions, ensuring that decision-making is informed by the latest and most accurate information.

4.2.2 Enriched Dataset for Comprehensive Decision-Making

The API integration layer enriches the AeroPad's dataset, providing controllers with a more comprehensive set of data, including altitude, weather conditions, and other critical parameters.

4.2.3 Proactive Conflict Detection for Enhanced Safety

AeroPad's Conflict Detection Module contributes to enhanced safety by proactively identifying potential conflicts and suggesting corrective actions, allowing controllers to intervene before issues escalate.

4.2.4 User-Friendly Interface for Enhanced User Experience:

The user interface's visually intuitive design enhances the user experience, providing controllers with a platform that is easy to navigate and comprehend.

4.3 Recommendations for Future Development

While AeroPad stands as a technological achievement, continuous improvement and adaptation are crucial for staying at the forefront of air traffic control innovation:

4.3.1 Machine Learning Integration:

The integration of machine learning algorithms is explored to further refine conflict detection based on historical data and evolving patterns.

4.3.2 Advanced Visualization Techniques

Advanced visualization techniques, including augmented reality, have been investigated to provide controllers with more immersive and intuitive displays.

4.3.3 Collaborative features

Collaborative features that enable real-time communication and coordination among air traffic controllers, can be implemented further enhancing teamwork and decision-making.

4.3.4 Cybersecurity enhancements

Cybersecurity measures should be prioritized to safeguard AeroPad against potential threats, ensuring the integrity and confidentiality of critical air traffic data.

4.4 Envisioning the Future of the AeroPad

Aero-pad has the potential to redefine the landscape of air traffic control technology. Envisioning its future involves:

4.4.1 Global Integration

Collaborate with international aviation authorities for seamless integration of AeroPad into global air traffic management systems.

4.4.2 Autonomous Flight Compatibility

Compatibility with emerging autonomous flight technologies should be explored, positioning the AeroPad as a crucial tool in the era of unmanned aerial vehicles (UAVs).

4.4.3 Continuous Research and Development:

A dedicated research and development team should be established to stay abreast of technological advancements and continually enhance the capabilities of the AeroPad.

In conclusion, the implementation of the AeroPad marked a paradigm shift in air traffic control, blending technological innovation with a user-centric design. As we reflect on these achievements and envision future possibilities, the aero-pad has emerged as a dynamic and adaptive solution poised to shape the future of air traffic management.

4.5 Source Code Samples

Sample Python code snippets illustrating key functionalities in the implementation of the AeroPad.

4.5.1 Real-Time Database Interaction (MongoDB)

```
# Code sample for MongoDB interaction in the RealTimeDatabase class
from pymongo import MongoClient


class RealTimeDatabase:


# Constructor and methods for MongoDB interactions
# ...
```

4.5.2 API Integration with a Flask

```
# Code sample for Flask-based API integration
from flask import Flask, jsonify, request


app = Flask(__name__)
```

```
@app.route('/api/get_aircraft_data', methods=['GET'])
def get_aircraft_data():
# Code to fetch real-time aircraft data from external source or API
aircraft_data = fetch_aircraft_data()

return jsonify(aircraft_data)

def fetch_aircraft_data():
```

```
# Placeholder for external API call or data retrieval logic
# ...

# Example response
aircraft_data = {'id': 'ABC123', 'position': [latitude, longitude], 'altitude': 30000}
return aircraft_data

if __name__ == '__main__':
app.run(debug=True)
```

a) Conflict Detection Algorithm

```
# Code sample for Conflict Detection Module
class ConflictDetection:
# Constructor and methods for conflict detection
# ...
```

4.6. Glossary of Terms

A glossary providing definitions for key technical terms used throughout the document.

4.6.1 MongoDB

NoSQL, is a document-oriented database that provides high performance, scalability, and flexibility.

4.6.2 Flask

A micro web framework written in Python was used for building web applications.

4.6.3. API (application programming interface)

A set of rules that allows one software application to interact with another.

4.6.4 Conflict Detection Module

A component within the AeroPad that employs algorithms to identify potential conflicts between aircraft.

HTML, CSS, Python, and JavaScript

Technologies used for building the user interface of the AeroPad.

5. RESULTS AND DISCUSSION

Table 2: Mapping of **Key Elements** with references (Just Taking first five references which are patent references)

Key Element	Ref. 1: US20190392726A1	Ref. 2: CN113470439A	Ref. 3: US20160155435A1	Ref. 4: US5200901A	Ref. 5: NPL-1
Element A	✓(partially)	✗	✗	✗	✗
Element B	✓(partially)	✓(partially)	✓(partially)	✓(partially)	✓(partially)
Element C	✓(partially)	✓(partially)	✓(partially)	✗	✗
Element D	✗	✓	✗	✗	✗
Element D1	✓(partially)	✗	✗	✗	✗
Element E	✓(partially)	✓(partially)	✓(partially)	✓(partially)	✗
Element F	✗	✗	✓(partially)	✓(partially)	✗

NOTE: The chart above indicates elements of the invention mapped against the references cited in this report.

✓ Indicates that the element is disclosed in the prior art

✗ Indicates that the element is not disclosed in the prior art

✓(partially) Indicates that the element is disclosed partially in the prior art

After conducting a comprehensive search, our analysis revealed that the system can suggest new values by itself, which can reduce the workload of the controller during high-traffic situations. However, none of the references mention introducing an Aero-pad Electronic Air Traffic Management System (AEAMS) between an air traffic controller and air traffic assistants or an FDPS (Flight Data Processing System). It is necessary for the references to specify that the system maintains a log of all the positions assigned to flights. If the controller adds a value that contradicts the preceding values, the system highlights the discrepancy, allowing the controller to correct it before assigning and ensuring accurate flight positioning in the respective airspace. Additionally, none of the references mention that the system allows the Air Traffic Controller to input position values such as call sign, departure squawk, flight levels, routes, headings, and route change instructions such as direct to reporting point or left of the track by Nautical miles, ETA, and ETD.

Furthermore, none of the references describe a system that enhances safety by electronically recording the traffic on this prototype. The system double-checks all the parameters a controller enters and corrects any level corrections or route clearance errors. Moreover, the references do not mention that the system stores all the assigned values in a digitally encrypted database, which can be accessed at any time in the future for review. This procedure improves safety by continuously recording heading and route changes, such as going directly to a reporting point or giving left or right-of-track permission to a pilot due to bad weather, which can be visualized on the radar for cross-checking.

CONCLUSION

The search was conducted by taking into consideration the **key elements** of the prototype as disclosed earlier in the **System Architecture**. Of the several references retrieved in our search, the identified references appeared to disclose **only Element D completely** as shown in **Table 2**. All the key elements, **except D**, did not seem to be completely disclosed in any of the identified references. The above disclosed subject matter appears to be novel as none of the references disclose all the key elements together.

Fig. 2--- Patent application details

Office of the Controller General of Patents, Designs & Trade Marks
Department for Promotion of Industry and Internal Trade
Ministry of Commerce & Industry,
Government of India

Application Details	
APPLICATION NUMBER	202241041669
APPLICATION TYPE	ORDINARY APPLICATION
DATE OF FILING	21/07/2022
APPLICANT NAME	**Gurprit Singh**
TITLE OF INVENTION	A SYSTEM AND METHOD FOR MANAGING AIRCRAFTS IN AIRSPACES USING AN AIRSPACE MANAGEMENT SYSTEM
FIELD OF INVENTION	COMPUTER SCIENCE

6. ACKNOWLEDGMENT

Personal thanks and acknowledgment to individuals, organizations, and communities that could contribute to the development of the AeroPad.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

ORCID Gurprit Singh https://orcid.org/0009-0004-8499-9516

REFERENCES

References to sources, tools, and frameworks that could be utilized in the development of Aero-pad.

[1] Béatrice Pesquet, Fateh Kaakai, Warren Beeston, inventors; Training and/or assistance platform for air management via air traffic management electronic system, associated method, Thales SA,2018-06-2, https://patents.google.com/patent/US20190392726A1/en?oq=US20190392726A1

[2] Method and system for solving control conflict of medium and small airports based on artificial intelligence, Civil Aviation Flight University of China, 2021-06-30, https://patents.google.com/patent/CN113470439A/en?oq=CN113470439A

[3] Mohammed Ibrahim Mohideen, inventor; Aircraft systems and methods for reducing and detecting read-back and hear-back errors, Honeywell International Inc, 2013-11-14, https://patentimages.storage.googleapis.com/40/d6/4c/52ed47e30aa9b3/US20160155435A1.pdf

[4] Arthur Gerstenfeld, Charles Millet, Thomas D. Moody, Erich Ma, inventors; Direct entry air traffic control system for accident analysis and training, UFA Inc 335 Boylston St Newton Ma 02159 A Corp Of Ufa Inc, 1986-11-18, https://patentimages.storage.googleapis.com/1d/a2/8a/253a9ebbfbd2e3/US5200901.pdf

[5] PaweB FerduBa1 and Jacek Skorupski, inventors; The Influence of Errors in Visualization Systems on the Level of Safety Threat in Air Traffic, January 2018, PaweB FerduBa and Jacek Skorupski, https://www.researchgate.net/publication/322681835_The_Influence_of_Errors_in_Visualization_Systems_on_the_Level_of_Safety_Threat_in_Air_Traffic/link/6183e4b8a767a03c14f24863/download

[6] Anrieta Dudoit, Vytautas Rimta, Marijonas, Jacek. “Effectiveness of Conflict Resolution Methods in Air Traffic Management” (2022). https://www.mdpi.com/2226-4310/9/2/112.

[7] Heinz Erzberger “Automated Conflict Resolution for Air Traffic Control” (2005), https://ntrs.nasa.gov/api/citations/20050242942/downloads/20050242942.pdf

[8] MongoDB Documentation: MongoDB. (n.d.). MongoDB Documentation, https://docs.mongodb.com/

[9] Flask Documentation: Flask. (n.d.). Flask Documentation. https://flask.palletsprojects.com/

[10] Leaflet Documentation: Leaflet. (n.d.). Leaflet Documentation. https://leafletjs.com/

[11] Python Documentation: Python Software Foundation. (n.d.). Python Documentation. https://docs.python.org/

[12] Introduction to MongoDB" by Kristina Chodorow: Chodorow, K. (2013). Introduction to MongoDB. O'Reilly Media.

[13] “Flask Web Development" by Miguel Grinberg: Grinberg, M. (2018). Flask Web Development. O'Reilly Media.

[14] "Leaflet.js Essentials" by Paul Crickard III: Crickard III, P. (2014). Leaflet.js Essentials. Packt Publishing.

[15] John C. Holmes, John Bergling Kathie Boss Debra M. Datcher Joe Henson, “Technology and the Future Evolution of the ATC System, Airport and Air Traffic Control System, January 1982" (Publication) https://www.princeton.edu/~ota/disk3/1982/8202/8202.PDF

[16] "Air traffic control: Modernization efforts are underway, but comprehensive planning is needed for key systems when acquiring new systems" (gao report): Government accountability office. (nov, 2023). Https://www.gao.gov/assets/d24105254.pdf

[17] Heinz Erzberger "Transforming the NAS: The Next Generation Air Traffic Control System" (2004), https://www.icas.org/ICAS_ARCHIVE/ICAS2004/PAPERS/562.PDF

[18] Kleinert, Matthias, “Adaptation of Assistant Based Speech Recognition to New Domains and its Acceptance by Air Traffic Controllers.” Intelligent Human Systems Integration: Integrating People and Intelligent Systems, electronic library, (2019), https://www.researchgate.net/publication/330179038_Adaptation_of_Assistant_Based_Speech_Recognition_to_New_Domains_and_Its_Acceptance_by_Air_Traffic_Controllers_Proceedings_of_the_2nd_International_Conference_on_Intelligent_Human_Systems_Integration_I

[19] Aiman Javed, Javadpur and Nilanjan Patra, Kolkatta Electrical Engineering Department, Published in India Conference (INDICON) 2016, IEEE Annual

[20] Paweł Ferduła[1,2] and Jacek Skorupski, (2018), The Influence of Errors in Visualization Systems on the Level of Safety Threat in Air Traffic, https://www.hindawi.com/journals/jat/2018/1034301/

[21] Masato FUJITA, Collision risk model for independently operated homogeneous air traffic flows in terminal area, (2013), https://www.researchgate.net/publication/259191065_Collision_risk_model_for_independently_operated_homogeneous_air_traffic_flows_in_terminal_area

[22] Majeed Mohamed, Online Estimation of Terminal Airspace Sector Capacity from ATC Workload, (December 15th, 2021), https://www.intechopen.com/chapters/78855

[23] Francisco Javier Saez Nieto, Collision Risk Model for High-Density Airspaces, (March 18th, 2020), https://www.intechopen.com/chapters/69641

[24] Dr. Bernd Korn, DLR – Institute of Flight Guidance Panel on "Intelligent Automation and Autonomy for a safe and secure Air Transport System." Concepts on how to introduce higher level of Automation in ATM, (2018), https://elib.dlr.de/123158/1/Korn_DASC2018_Panel_2018-09-25.pdf

[25] Shany Seawright, SkyFirst Limited Moves to Paperless Operations for Entire Fleet, On the Flight Deck, Sep 27, 2021, https://connectedaviationtoday.com/skyfirst-limited-moves-to-paperless-operations-for-entire-fleet/

[26] Doc 9426-An/924, Air Traffic Services Planning Manual, first (Provisional) Edition – 1984, Approved by the Secretary General and published under his authority, https://documents.pub/download/doc-9426-air-traffic-services-planning-manual-dot-9426-an924-air-traffic-services.html