Project 085 Strategies for Improving En-Route Fuel Efficiency

Georgia Institute of Technology

Project Lead Investigator
Professor Dimitri N. Mavris
Director, Aerospace Systems Design Laboratory
School of Aerospace Engineering
Georgia Institute of Technology
Phone: (404) 894-1557
Fax: (404) 894-6596
Email: dimitri.mavris@ae.gatech.edu

University Participants

Georgia Institute of Technology (Georgia Tech)

- P.I.s: Dr. Dimitri N. Mavris, Dr. Michelle Kirby
- FAA Award Number: 13-C-AJFE-GIT-161
- Period of Performance: March 19, 2024, to September 30, 2025
- Tasks:
  1. Development of a path optimization algorithm and use case examples
  2. Speed Control and Vectoring Identification
  3. Fuel Burn Surrogate Modeling
  4. Fleet-wide Speed Control Strategy

Project Funding Level
Georgia Tech has received $400,000 in funding from the Federal Aviation Administration (FAA) for this project. In terms of cost-share details, Georgia Tech has agreed to a total of $400,000 in matching funds. This total includes salaries for the project director, research engineers, and graduate research assistants, as well as computing, financial, and administrative support, including meeting arrangements. Georgia Tech has also agreed to provide tuition remission for the students, paid for by state funds.

Investigation Team
Prof. Dimitri N. Mavris, (P.I.), Tasks 1–4
Dr. Michelle Kirby, (co-Investigator), Tasks 1–4
Dr. Ameya Behere, (research faculty), Tasks 1–4
Dr. Holger Pfaender (research faculty), Tasks 1–4
Dr. Cedric Justin (research faculty), Tasks 1–4
Mr. Jirat Bhanpato, (research faculty), Tasks 1-4
Jouwairia Ahabchane (graduate student), Tasks 2
Titien Berberian (graduate student), Tasks 1
Jacob Hawkins (graduate student), Tasks 1
Ghizlane Jari (graduate student), Tasks 3
Yasser Moumtaz (graduate student), Tasks 4
Vishva Rana (graduate student), Tasks 2

Project Overview

In the face of anticipated increased aviation demand for the future, multiple avenues exist to achieve system-wide fuel efficiency including aircraft and engine technologies, operational improvements, and policy measures. This research project focuses on the strategies of near and long-term operational improvements (OIs) and quantify their impact on reducing the aviation fuel efficiency, with specific focus on en-route efficiency. The anticipated outcomes for the FAA are a series of recommended strategies for improving the en-route fuel efficiency of aviation.

Task 1 - Development of a Path Optimization Algorithm and Use Case Example

Georgia Institute of Technology

Objectives

The objective of this task is to develop a flight trajectory path optimization algorithm that can measure the amount of fuel a flight would save by implementing speed control to avoid some of the delays spent in holding patterns or vectoring that occurs during system disruptions or high-volume activity. This is done by combining Flight Operations Quality Assurance (FOQA) data with fuel burn surrogate models to get an estimation of the fuel savings. Currently, the focus is on the holding pattern identification task as a primary use case for using speed control during cruise to avoid inefficient flight in holding patterns, with future plans to include vectoring.

Research Approach

Compared to the path optimization algorithm presented in last year's annual report, the new path optimization algorithm has many upgrades. Most of the improvements come from using FOQA data as the basis of the simulation. Last year's algorithm sought to find a path between two points that would minimize the amount of fuel used by the aircraft. While this algorithm was useful, it was difficult to draw conclusions about how to best modify an aircraft's speed to avoid burning fuel in a holding pattern. To fix this issue, a new type of path optimization algorithm was created. This new algorithm simulates how the total amount of fuel an aircraft uses would change if the aircraft slowed down to avoid at least a portion of a holding pattern. By using FOQA data from a real flight as its starting point, the algorithm can test how speed control would affect fuel usage in real life scenarios and provide recommendations for how aircraft in certain situations should modify their speed to save fuel.

Path Optimization Algorithm

To quantify the potential fuel savings available to a flight, a simulation environment was created that helps assess what would happen if a flight were to implement speed control tactics. Results are compared to the baseline FOQA data from the real flight. This simulation uses the fuel burn surrogate models discussed in Task 3 to estimate the amount of fuel that would be saved by implementing speed control at different speeds for different durations.

Before the simulation is run, the holding pattern detection algorithm, discussed in Task 2, is run on each FOQA flight and the information on any holding patterns is stored in a database for that given flight. Once a flight has been identified as having experienced a hold, its information can be passed on to the path optimization algorithm for further analysis. Now that the flights with holds have been identified, the cases run on each flight need to be defined. To define a set of cases to run for a flight, the Mach numbers flown at during speed control and the percentages of the hold to absorb must be specified. The user has the option to provide specific Mach numbers and absorption percentages to test, or they can leave these fields blank and the simulation will instead use default values. If no Mach numbers are provided, the simulation will select a default set of Mach numbers that depends on the airframe. The simulation then pulls information on the length of the hold and how much fuel was used during the hold from the database created by the holding pattern detection algorithm using a flight identification number, denoted as flight ID. The flight ID is also used to get additional metadata on the flight, such as which airframe and engine combination was used. Information on the airframe and engines is then used to select which fuel burn surrogate model to use for calculating the new fuel flow values when speed control is implemented. Once the general flight information has been collected, a Design of Experiments (DoE) is used to vary the time to absorb and Mach number ranges (either those provided by the user or the default values if none were provided) and running each possible combination in the trajectory simulator. The code is parallelized so that each case can be run simultaneously, which reduces the overall run time. The speed improvement provided by this parallelization of the code is

limited by how many processes a computer can execute at once. For example, if the simulation runs 32 cases on a computer that can run 8 processes at once, it will run the cases in four batches of eight cases.
For each case in the DoE, the trajectory simulator tries to determine when speed control should start to absorb the requested amount of time by flying at the requested Mach number for that specific DoE case. Using latitudes and longitudes listed in the FOQA data, the simulation can determine how far the aircraft moved during each timestep throughout all of cruise. Next, the ground speed of the aircraft, as it travels from point to point, is calculated. To ensure this calculation is accurate, the wind conditions need to be accounted for. How the wind conditions affect the ground speed, as well as definitions for all relevant variable abbreviations, is shown in Figure 1.

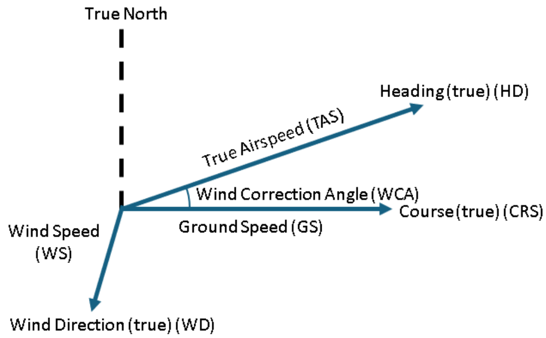


Figure 1. A graphical representation of the values used in the ground speed calculation.

Because the wind and aircraft are not guaranteed to be moving in the same direction, the heading of the aircraft and its course will not be aligned. This difference is accounted for by the wind correction angle, which can be found using Equation 1:

$$\sin WCA = \frac{WS}{TAS}\sin(WD - CRS) \qquad \text{(Eq. 1)}$$

Once the wind correction angle is known, the ground speed can be calculated using Equation 2:

$$GS = TAS\sqrt{1 - (\sin WCA)^2} - WS\cos(WD - CRS) \qquad \text{(Eq. 2)}$$

With the distances between each point and the new ground speeds known, the amount of time it would take to travel between each point while implementing speed control can be calculated by dividing the distances by the ground speeds. The amount of time absorbed at each timestep can then be calculated by subtracting the original amount of time taken to move between points from these new times.

Now that the amount of time absorbed at each location is known, the location where speed control should start can be identified. This can be done by taking the individual absorption times and summing them up starting from the end of the cruise phase and working backwards. The summation continues until the total absorbed time reaches the requested absorption time. At this point, the time at which speed control should start is recorded. If the summation reaches the beginning of the cruise phase before the total absorbed time reaches the requested absorption time, this means that the flight is not long enough to absorb the requested amount of time at the requested Mach number. If this occurs, the simulation will end the case by producing an error message explaining that that case is not feasible.

Once the starting point of speed control is identified, the amount of fuel used during speed control can be calculated. In order to calculate the fuel flow, the altitude, air temperature, gross weight, Mach number, rate of climb, and Mach change over a 20-sec period must be known. Almost all these values can be acquired from the FOQA data. Only the gross weight is unknown because the weight of the aircraft at any point in the flight is a function of all the instantaneous fuel flows before that point. This contradiction can be solved by creating an iteration scheme to converge on the correct gross weight. By using the gross weight listed in the FOQA data as an initial estimate, it is possible to calculate an initial estimate of the instantaneous fuel flows throughout the entire speed control phase. Using these estimated fuel flows, the gross weight estimate can be updated. This process can then be repeated until the gross weights converge, which means that the gross weights supplied to the surrogate models match the gross weight we would calculate using the fuel flows provided by the surrogates. The update to the weight estimate is given by Equation 3:

$$\overrightarrow{W_{n+1}} = W_{n,0} - \vec{f}(\overrightarrow{W_n}, \dots) \quad \text{(Eq. 3)}$$

where $\vec{W}$ is the weight of the aircraft at each timestep from the start of speed control to the end of cruise and $\vec{f}$ is the amount of fuel used from the start of speed control to each timestep. The simulation then stops iterating on the weight estimate once Equation 4 is satisfied:

$$\max(\|\overrightarrow{W_{n+1}} - \overrightarrow{W_n}\|) < \varepsilon \quad \text{(Eq. 4)}$$

where ε is a small number. While it would be simpler to iterate through each time step and calculate each instantaneous fuel flow sequentially, this method is much more efficient because it only requires a handful of iterations to converge as opposed to the hundreds or thousands of time steps that would need to be iterated through.

Once the fuel calculations are completed, the simulation is done. All that remains is to output a handful of measures of effectiveness for the case that determine how feasible and useful absorbing that much of the hold at that Mach number would be. Examples of some of the measures of effectiveness that are calculated and stored include speed control duration and net fuel savings. Net fuel savings is a crucial measure of effectiveness because it quantifies how much fuel would be saved by implementing speed control. This metric is calculated by comparing the amount of extra fuel used in cruise to the amount of fuel saved by avoiding part of the holding pattern. When making this calculation, it is assumed that the fuel flow during the holding pattern was constant, so the total amount of fuel saved is proportional to the percentage of the holding pattern that was avoided. To calculate the net fuel savings, Equation 5:

$$\Delta F = \frac{F_{Hold} T_{Abs}}{T_{Hold}} - (F_{Cruise,SC} - F_{Cruise,O}) \quad \text{(Eq. 5)}$$

where ΔF is the net fuel savings, F_{Hold} is the amount of fuel that was burned during the hold, T_{Abs} is the amount of time absorbed by speed control, T_{Hold} is the duration of the hold, $F_{Cruise,SC}$ is the amount of fuel burned during cruise when speed control is implemented, and $F_{Cruise,O}$ is the amount of fuel originally burned during cruise. The other important metric is speed control duration, which determines how much lookahead time for a hold is required to implement that combination of absorbed hold time at the specified Mach number. Even if a case can provide a large fuel benefit, it may require too long of a look ahead time to be practical. A process diagram that summarizes the overall logic of the path algorithm is included in Figure 2.

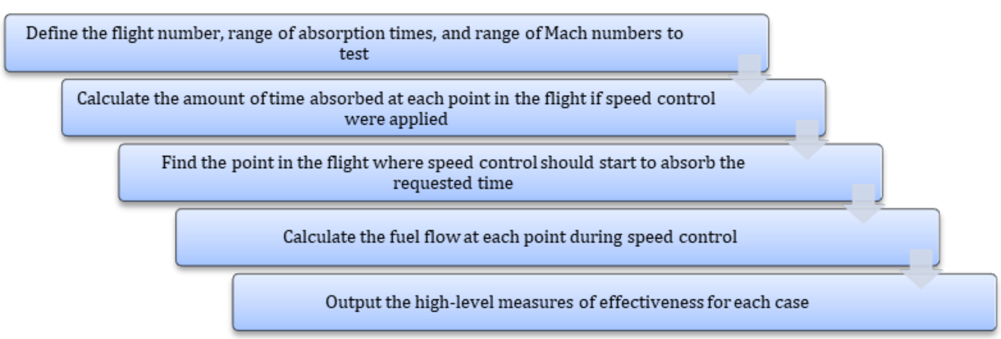


Figure 2. Process diagram for the path optimization algorithm

Reducing the Algorithm's Runtime

The 2024 version of the trajectory simulator was not optimized for execution time. This limitation poses challenges to analyzing a large number of flights and multiple strategies were implemented to significantly reduce the trajectory simulator's runtime. The first step taken to reduce the runtime was to only simulate the speed control portion of the flight as opposed to the entire cruise phase. The calculations were then optimized by vectorizing them. The 2024 version of the code iterated through each point in the flight sequentially and calculated the state of the aircraft at each of these points individually. To improve efficiency, these equations controlling the aircraft's state were modified to perform these calculations on vectors. This allows all state calculations for the entire speed control phase to be combined into one set of function calls, which significantly improves the performance of the simulation. Logic was also added to discard cases where it is not feasible to absorb the requested time at the requested speed during the cruise phase, which keeps the algorithm from wasting simulation time on impractical cases.

The runtime has improved for the new path optimization algorithm compared to the 2024 version is listed in Table 1. Overall, the various code optimization strategies have produced a 98% reduction in the runtime. This faster runtime will make it significantly easier to analyze all the flights available in the FOQA data and also extend to a system level assessment with a very large number of flights to simulate.

Table 1. Runtime comparisons for the 2024 and 2025 versions of the path algorithm.

GT Flight ID	Cases	2024 Version Runtime (minutes)	2025 Version Runtime (minutes)
GT2437319	20	24	0.5
GT2417663	25	35.5	0.6

Use Case Example
Once the simulation is completed, the environment produces contour plots that show how each case is performed. Using Once the simulation is completed, the simulation produces contour plots that show how each case performed. Using these graphs, it is possible to draw conclusions about the possible benefits of applying speed control to that flight. As an example, consider a Boeing® 767-400ER flight from Los Angeles International Airport (KLAX), Los Angeles, California to John F. Kennedy International Airport (KJFK), New York, New York that encountered a hold. The entire flight path of the flight is depicted in Figure 3, while a zoomed in view on the descent phase that shows the holding pattern the aircraft experienced is depicted in Figure 4. The aircraft spent 3.9 hr in cruise at an average speed of Mach 0.797 before entering a holding pattern during its descent. This holding pattern lasted approximately 24 min and burned 3,575 lb of fuel.

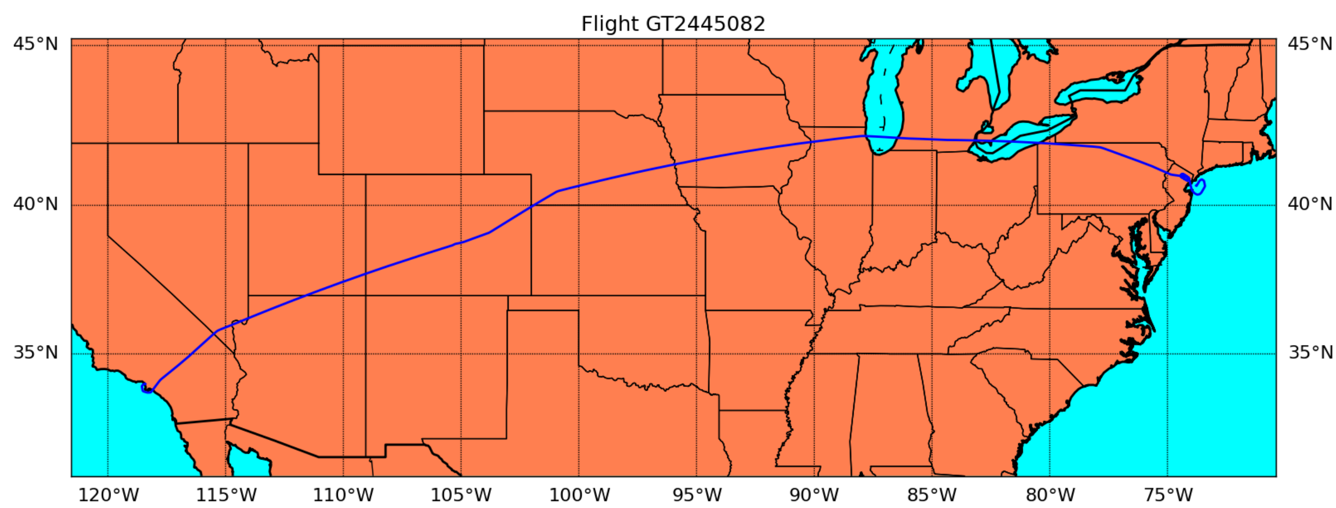


Figure 3. Flight path for the flight from Los Angeles to New York City

® Boeing is a registered trademark of The Boeing Company, Seattle, Washington.

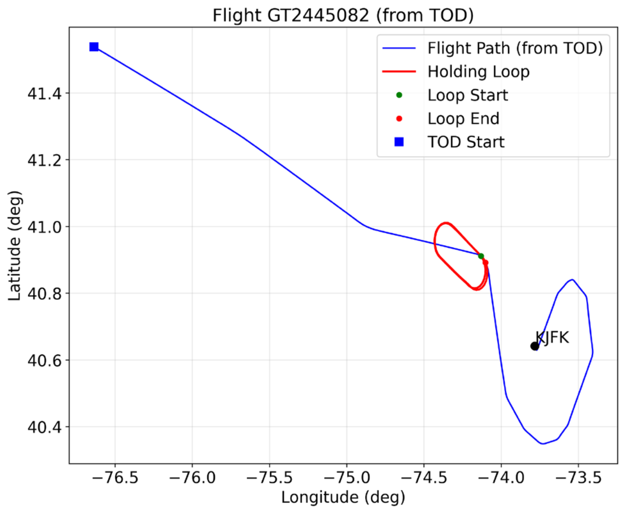


Figure 4. The descent phase of the flight with the holding pattern highlighted.

By applying the path optimization algorithm to this flight, the potential fuel burn benefits available if this flight had implemented speed control can be quantified. The net fuel savings calculated by the path optimization algorithm for a range of speed control Mach numbers and absorption times are shown in Figure 5 as a contour plot. The case that saves the most fuel occurs when the aircraft absorbs 6 min of the hold by slowing down to Mach 0.76 to save 860 lb of fuel in the upper left corner of the contour plot, but there are many other cases that would result in fuel savings. In general, cases with more fuel savings occur at high absorption times and low Mach numbers. This trend occurs because at low absorption times, the amount of fuel the aircraft avoids burning in the holding pattern is smaller compared to high absorption times. Meanwhile, at higher Mach numbers, the aircraft must implement speed control for longer, which negates the fuel savings that would be seen by avoiding the holding pattern because the aircraft has to fly at a Mach number with a higher fuel flow compared to the original cruise for a longer time. Some of these high Mach number cases are not even feasible because the required duration for speed control would exceed the cruise duration for that flight.

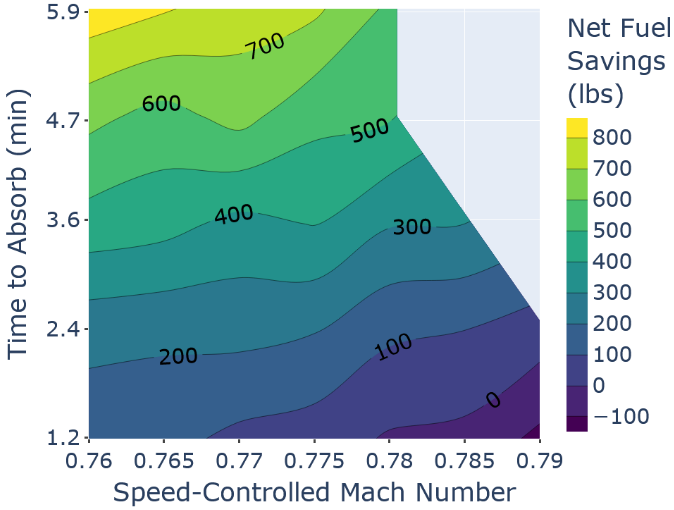


Figure 5. Net fuel savings for a 767-400ER flight from Los Angeles to New York City.

To ensure that these fuel savings are also practical to achieve, the required speed control duration of these cases must also be examined. A contour plot of the speed control durations for the tested cases is depicted in Figure 6. The case that saves the most fuel requires 2.5 hr of lookahead time to implement, which is usually too much time for there to be sufficient warning for a holding event. If that amount of time is not available, one of the other cases in the DoE that requires less lookahead time could be selected and still provide some fuel savings. For example, as part of Task 2, approximately one-third of flights in the FOQA flight data experienced a hold with a lookahead time of at least 40 min. With 40 min of lookahead time, cross-referencing the time to absorb in Figure 5 to the speed control duration in Figure 6 shows that absorbing approximately 2 min of a hold by flying at Mach 0.76 will take 40 min and save approximately 180 lb of fuel.

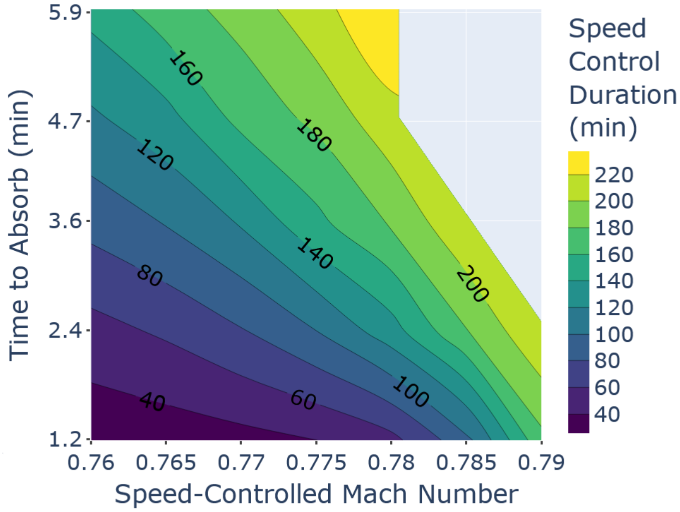


Figure 6. Speed control durations for a 767-400ER flight from Los Angeles to New York City.

These kinds of tradeoffs between lookahead time and net fuel savings highlight the importance of considering a flight's specific situation when making speed control recommendations. Depending on the situation, some flights will have more lookahead time and be able to achieve greater fuel savings than other flights. For example, a narrowbody on a short to medium haul flight distance will always have less potential to benefit from speed control than a widebody aircraft on an international flight because it will always have a limited lookahead time. The situation becomes even more complex when considering how multiple flights affected by the same holding event should act, as will be discussed in Task 4. Regardless, this example shows that saving hundreds of pounds of fuel on a single flight is possible and could lead to significant fleet-wide savings if more widely implemented.

Milestones

- Identified holding patterns in both FOQA and threaded track (TT) flights in 2023.
- Developed a path optimization algorithm for a trajectory simulator that uses FOQA data as a baseline to compare the effects of speed control to quantify fuel burn benefits and assessed a number of FOQA flights.

Major Accomplishments

- Successfully implemented a path optimization algorithm that can quantify how much fuel a flight would save if speed control were implemented. This trajectory simulator will assist in determining how often speed control can be used to save fuel and will be used to inform the fleet-wide speed control strategies discussed in Task 4.
- Optimized the path optimization algorithm to significantly reduce its runtime, which will make it much easier to acquire all of the fuel savings data needed to create fleet-wide speed control strategies.

Publications

None.

Outreach Efforts

Participated in ASCENT biannual meetings.

Awards
None.

Student Involvement
- Titien Berberian (graduate student) created the 2024 version of the path algorithm.
- Jacob Hawkins (graduate student) created the 2025 version of the path algorithm.

Plans for Next Period
- Use the path optimization algorithm to analyze all flights with available FOQA data and store the resulting fuel savings metrics in a database. This database could then be used as the training basis for a surrogate model that produces an expected fuel savings for a given set of speed control parameters.
- Modify the path optimization algorithm to accept threaded track data, which would allow many more flights to be analyzed.

References
None.

Task 2 – Speed Control, Holding Patterns and Vectoring
Georgia Institute of Technology

Objective
The objective of this task is to leverage FOQA data to understand how delays are currently or could be absorbed en-route and in terminal airspace through speed control, holding patterns and vectoring. For speed control, the goal is to use speed change analysis to uncover speed control events within the 50,000 FOQA flights and to characterize when an intentional cruise Mach adjustment occurs in cruise. For holding patterns, the work focuses on finalizing and applying a refined holding detection methodology across TT data to quantify when holding occurs, identify holding stacks, and determine the look-ahead time for potential early cruise speed control application. For vectoring, the initial work centers on developing a methodology for identifying air traffic control (ATC) vectoring events in terminal airspace and quantifying the additional time and fuel burn induced by vectoring.

Research Approach
This section presents the research approaches for speed control, holding patterns, and vectoring while outlining the methodology and methods used in this task.

Speed Control Detection
Changepoint (CP) analysis was applied to cruise-phase flight data to systematically identify speed-control events, which includes instances where the aircraft may have intentionally altered Mach number to meet flow-management objectives or manipulate estimated time of arrival (ETA). The motivation behind this approach is that manually reviewing thousands of flights for subtle speed shifts from a visual perspective in a plot of Mach number versus ground distance is inefficient, subjective, and inconsistent. By holding altitude constant and jointly examining related parameters, specifically turbulence, the algorithm distinguishes between intentional speed-control maneuvers and natural fluctuations in response to atmospheric effects. Instead of individually and visually identifying flights for potential speed control events, an algorithm incorporating changepoint analysis was developed and applied to all flights of interest. The full detection pipeline, from raw flight data preprocessing through changepoint filtering and classification, is depicted in Figure 7 and each step will be discussed further.

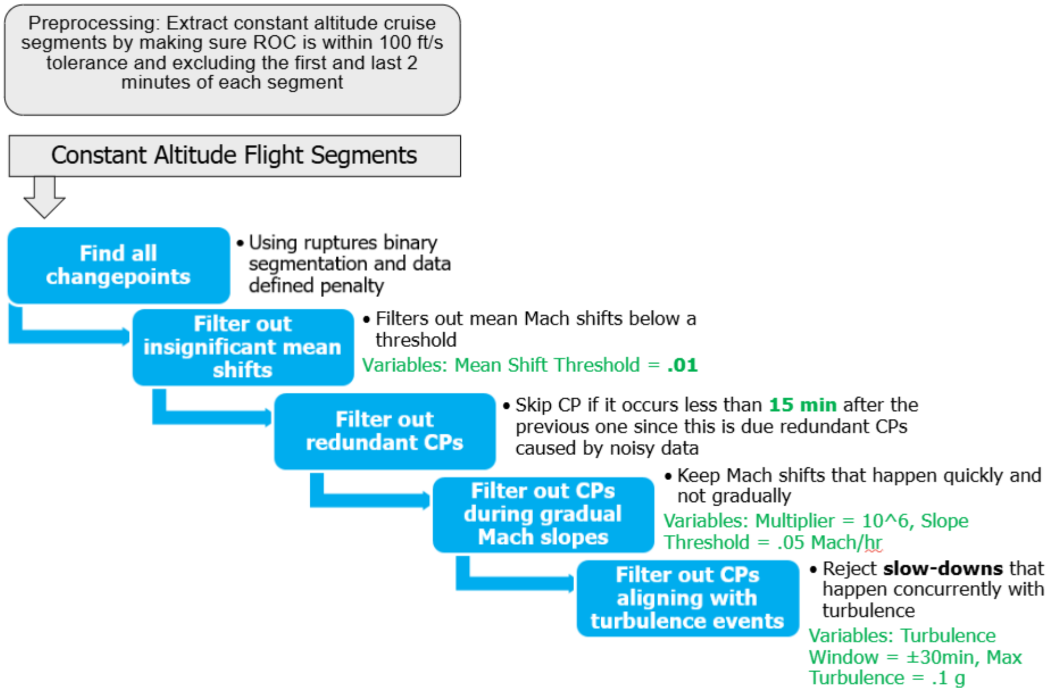


Figure 7. Changepoint (CP) analysis algorithm process.: Rate of Climb (ROC)

Data Preparation

Changepoint analysis is a family of statistical methods that identify locations in typically time series data where underlying properties such as mean level change (Aminikhanghahi & Cook, 2017). In this study, changepoint analysis is used to determine shifts in mean Mach level in FOQA flight time series data. The changepoint algorithm begins by segmenting each flight into constant-altitude cruise portions. To avoid large changes in Mach number or acceleration and deceleration behavior in transient flight phases from steady and level flight, the first and last 2 min of each segment are excluded. Within each valid cruise section, the Python library ruptures is used to perform CP detection, applying the binary segmentation algorithm with a data-defined penalty to detect changes in the mean Mach number (Truong et al., 2018). Binary segmentation iteratively finds the most significant single changepoint in a segment, splits the series at that location, and then repeats the procedure on the resulting sub-segments, yielding multiple changepoints in long time series (Fryzlewicz, 2014). This produces raw CPs representing any statistically significant shift in mean Mach number. However, many of these events are operationally insignificant, redundant, or gradual trends rather than discrete speed-control actions. For example, many speed shifts are gradual over time, indicating that this is not a pilot-induced or deliberate attempt to change speeds, but rather due to autothrottles or flight management systems. Therefore, a multi-stage filtering sequence was implemented to refine the CP set.

Algorithm Filtering

The first filter removes insignificant mean shifts by applying a threshold—mean Mach changes below 0.01 are discarded. This mean shift threshold (MST) variable acts as a sensitivity knob to how many CPs are kept from the large amount that are initially detected using ruptures' binary segmentation. The CP detection with various threshold values is depicted in Figure 8. The value 0.01 was picked to grasp as many significant shifts as possible while excluding CPs detected due to noisy data. The second filter removes redundant CPs caused by noisy data, skipping any CP that occurs within 15 min of a previous one. An example of a case where one CP would be filtered out is shown in Figure 9 below, where multiple CPs are detected next to each other. The third filter targets CPs that result from gradual Mach drifts rather than abrupt operational decisions. Using a slope-based criterion (with a slope threshold of 0.05 Mach/hr and a large multiplier to emphasize rapid changes), only sharp, sudden Mach shifts are retained. This ensures that the algorithm isolates discrete actions rather than slow trends. An example of these kinds of CPs is shown in Figure 10, which are filtered out.

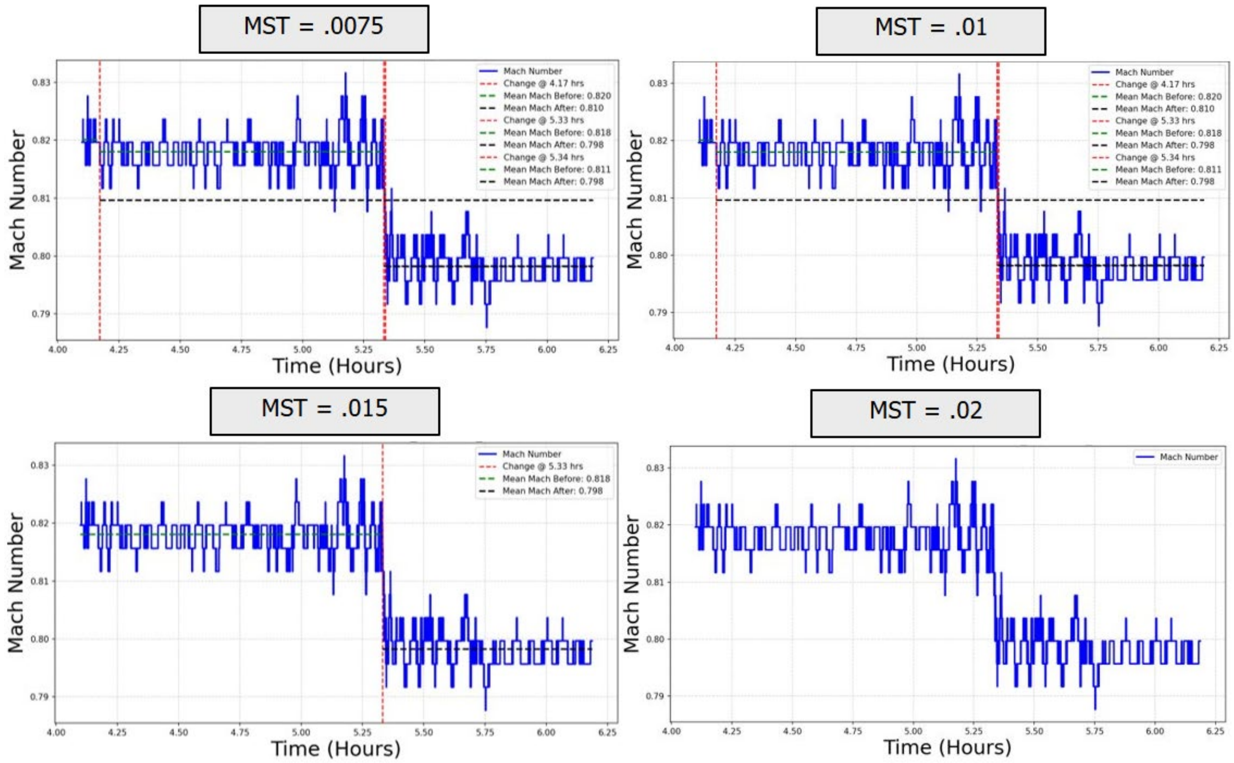


Figure 8. Sensitivity of changepoint (CP) analysis to various mean shift thresholds.

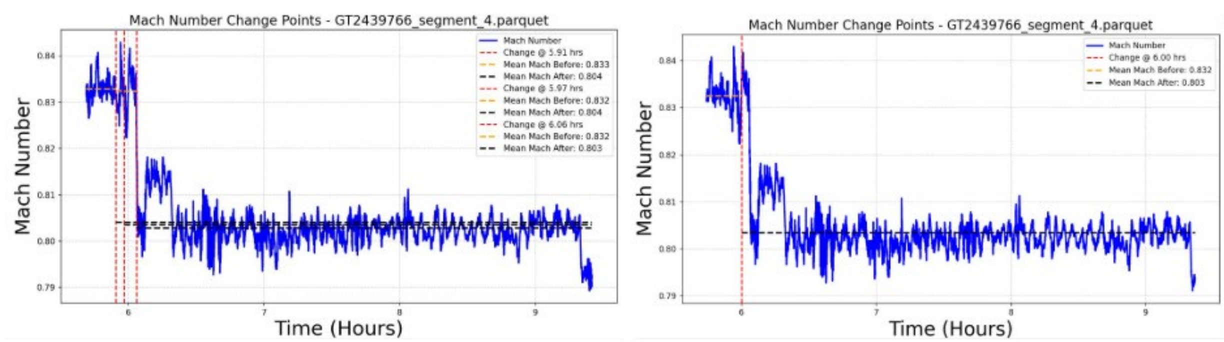


Figure 9. Segment with redundant changepoints (CPs).

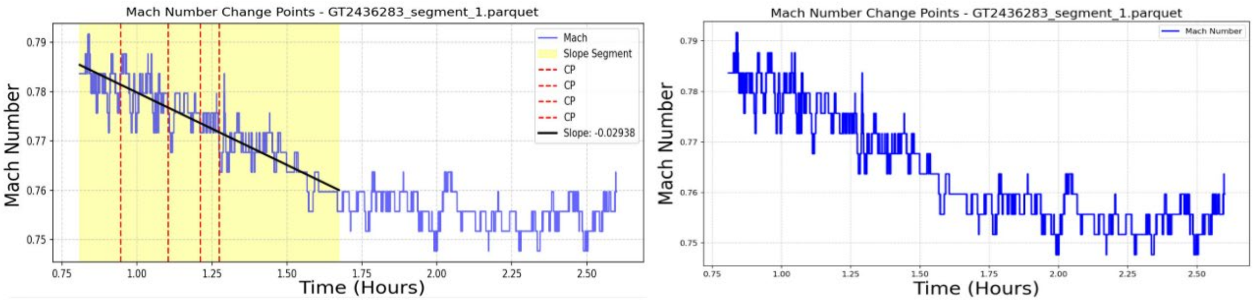


Figure 10. Segment with gradual changepoints (CP) Turbulence Filtering.

A final filter incorporates turbulence. Turbulence can cause passenger discomfort and structural loads that can be mitigated via slowdowns during cruise, rather than speed-ups, and altitude changes. To capture the speed change mitigation approach, the algorithm first classifies each CP as either a Mach increase or decrease. The turbulence filter uses a ±30-min time window, and a maximum turbulence threshold of 0.1 g is applied only to slow-down CPs. Having a window around the CP is important since turbulence and the speed change may not align exactly. This 0.1 g value is derived from subject matter experts from major commercial airlines. A turbulence value of 0.1g is considered a mild turbulence event but may still cause enough discomfort to prompt pilots to slow down. Any slow-down coinciding with a turbulence event is rejected as non-intentional but related to an atmospheric cause. Speed-ups are left unfiltered, since turbulence should not lead the flight crew to speed up. Through this series of filters, the algorithm isolates high-confidence CPs that represent meaningful, likely intentional speed-control maneuvers. The results of these modifications and their impact on CP identification are presented in subsequent analyses.

Speed Control Results and Discussion

As an example for the Airbus® A350-900 airframe, almost 20% of flights potentially use speed control. Out of 510 flights analyzed, there are 100 flights with at least one or more changepoints as shown in Figure 11. The total number of speed control events, which only included slowdowns, for each widebody airframe, is shown in Figure 12. The total number of medium and long-haul widebody flights that had speed control event is 2,922 of the 50,000 FOQA flights. The results show that speed control seems to be already practiced by some airlines. Thus, absorbing terminal delays via lower cruise speeds is an operationally promising path to improve en-route fuel efficiency. System-wide implementation may yield significant benefits and can be determined by considering the fuel burn impact of flights utilizing speed control and flights utilizing holding or vectoring. The results of this investigation provide clarity on the frequency of speed control used in the fleet currently and insights for parallel efforts such as the simulation of speed control events in place of holding.

® Airbus is a registered trademark of Airbus Operations GmbH, Hamburg, Germany.

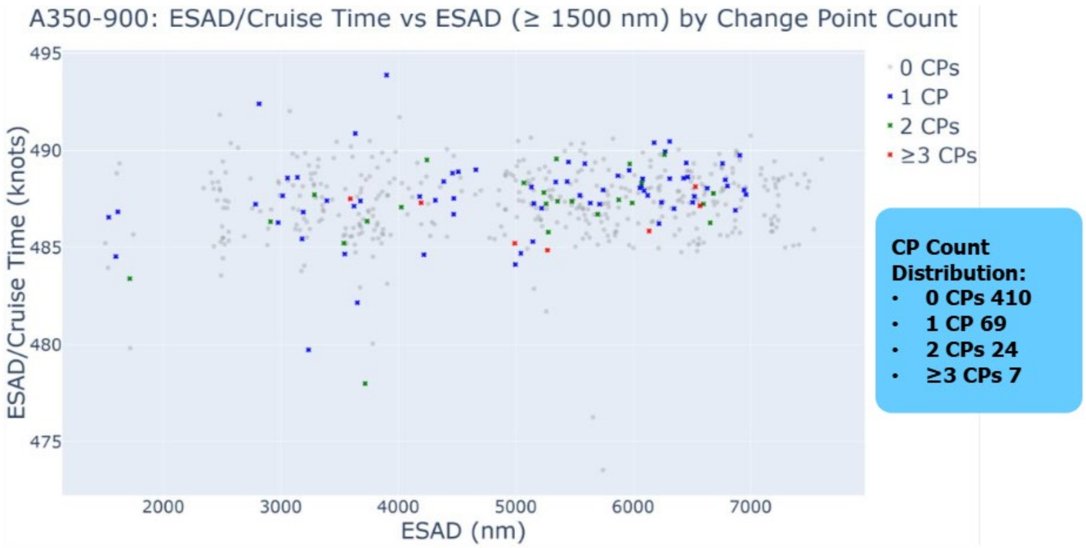


Figure 11. Changepoint (CP) breakdown for the A350-900. ESAD: equivalent still air distance.

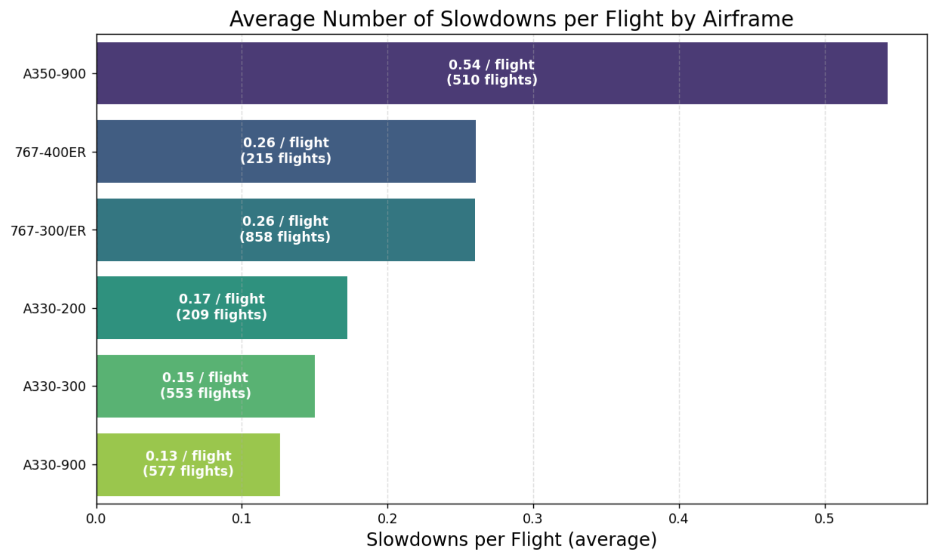


Figure 12. Slow-down breakdown for each widebody airframe identified with algorithm.

Summary and Observations of the Speed Control Detection Algorithm

The changepoint-based speed control detection algorithm provides a systematic and data-driven method for identifying intentional cruise-phase Mach changes across the entire FOQA dataset. By leveraging statistical CP detection, multi-stage filtering, and turbulence screening, the algorithm detects only those Mach shifts that are operationally meaningful, while removing noise, redundant detections, and unintentional shifts. The resulting dataset represents a set of potential speed-control events that can likely be attributed to flight crew decision-making.

Several observations can be made from these results. First, speed control seems to already be in practice by a sizeable portion of the fleet. Focusing on long-haul widebody flights, approximately 20% of A350-900 flights in the sample exhibited at least one potential speed-control event and nearly 3,000 events were detected across all long-haul widebody flights examined. Also, the average number of slowdowns per 100 flights for the A350-900 is 54, the most significant out of all widebody airframes. This analysis plays a crucial role in the broader investigation of how terminal area delays can be absorbed upstream in the trajectory to reduce the reliance on holding patterns and vectoring. The results quantify the prevalence of speed-control already used across the fleet, establishing a baseline against which holding and vectoring can be evaluated. The detected events can be integrated into fuel burn models to estimate the potential system-wide benefit of shifting some delay absorption from holding to speed control. The CP dataset informs parallel simulation efforts by providing realistic speed-control magnitudes, durations, and start times. These parameters help improve overall simulation fidelity and insights for modeling and comparison of the different alternative delay management strategies being explored.

Holding Pattern Detection in Threaded Track Data

The detection of holding patterns in TT data builds upon the methodology established in last year's report that focused on the FOQA flights, with several refinements implemented to improve accuracy and exclude go-arounds from the analysis. TT data required additional processing steps compared to FOQA since it does not contain flight phase labels. A fuzzy logic (Junzi et al., 2017) phase identification method was implemented to classify each point in a TT flight as climb, cruise, descent or level using membership functions based on altitude, vertical speed and true air speed. This approach was chosen because fuzzy logic provides a continuous membership-based representation of flight behavior which is well suited to TT data because it avoids hard thresholds and reduces misclassification in regions where climb, level and descent characteristics overlap. Compared to rule-based classification methods, fuzzy logic is more robust to noise and avoids misclassifying brief fluctuations in vertical speed which ensures a more reliable phased identification compared to threshold-based methods. This step ensures that every segment of the TT trajectory is classified into the correct flight phase, allowing the holding detection to be selectively applied to the cruise and descent phases of flight where holding patterns occur. Once the top of descent (ToD) is identified, a go-around detection logic flags flights having a transition from descent to a sustained climb or level-off at low altitude, and those flights are removed before the holding pattern detection algorithm is applied. An example of the fuzzy logic method and the go-around detection applied to a TT flight is shown in Figure 13 where the different phases of flight are categorized along with a go-around. After this filtering, the same geometric self-intersection logic developed previously for FOQA data is applied to the TT flights. An example of this approach is shown in Figure 14, where a FOQA flight with holding pattern in red along with the surrounding TT holds in blue arriving at Hartsfield-Jackson Atlanta International Airport (KATL) are displayed within a 100-nautical-mile radius.

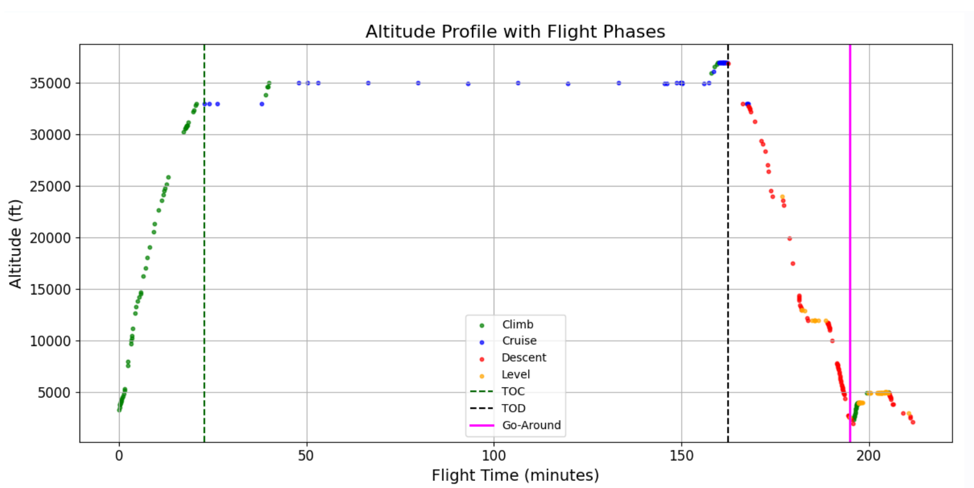


Figure 13. Example of top of descent (ToD) and go-around detection for a threaded track (TT) flight. ToC: top of climb.

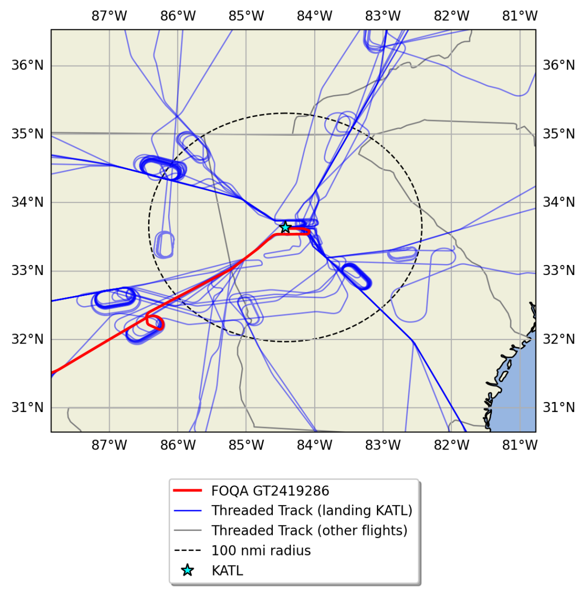


Figure 14. Example of Flight Operations Quality Assurance (FOQA) and threaded track (TT) holding patterns at Hartsfield-Jackson Atlanta International Airport (KATL).

Identifying Holding Stacks and Look-ahead Time

The identification of the holding stack shows how early the holding event becomes visible in the surrounding traffic environment and whether there is sufficient time for cruise speed control implementation to absorb a part or all of the holding pattern. Once the FOQA holdings are isolated, TT data are used to examine the surrounding traffic around the landing airport. The analysis begins with a ±30-min query window to identify TT flights holding within a 100-nmi radius centered around the landing airport. The window is then expanded in a 30-min increment on each side until no additional TT holding is identified. This iterative expansion reveals the holding stack which is defined by the earliest and latest TT holds around the FOQA holding event. This analysis considers FOQA and TT flights with holding patterns detected both in cruise and descent. The relative timing between the onset of the holding stack and the moment the FOQA aircraft enters holding provides a direct measure of the look-ahead time, which indicates whether speed control could be applied early enough to absorb part or all of the hold to potentially save fuel. An example of a holding stack is shown in Figure 15, where each TT hold is plotted relative to the FOQA hold start time. The pink shaded region represents the holding stack window, and the look-ahead time is measured from the earliest TT hold to the FOQA hold start.

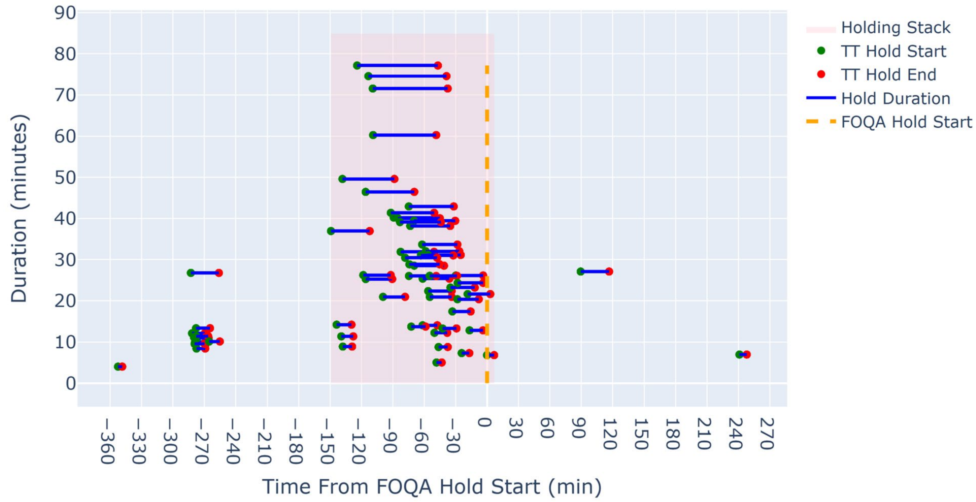


Figure 15. Holding stack of a Flight Operations Quality Assurance (FOQA) flight. TT: threaded track.

The distribution of FOQA flights across holding stack categories is shown in Figure 16. A total of 540 FOQA flights joined an ongoing holding stack, meaning the look ahead time was greater than zero minutes. Among these, 213 flights joined the stack at least 40 min after its formation, providing substantial opportunities for cruise phase speed control application. An additional 116 FOQA flights were the first aircraft to initiate the stack producing zero look-ahead time. Another subset of flights, totaling 202, exhibited isolated FOQA holds, and an additional set of 129 flights could not be evaluated due to the absence of TT data for international destinations. If this analysis is broadened by lowering the look-ahead time from 40 to 20 min, the set of eligible flights for speed control simulation expands. Under this threshold, the number of flights joining the stack at least 20 min after its formation is 350.

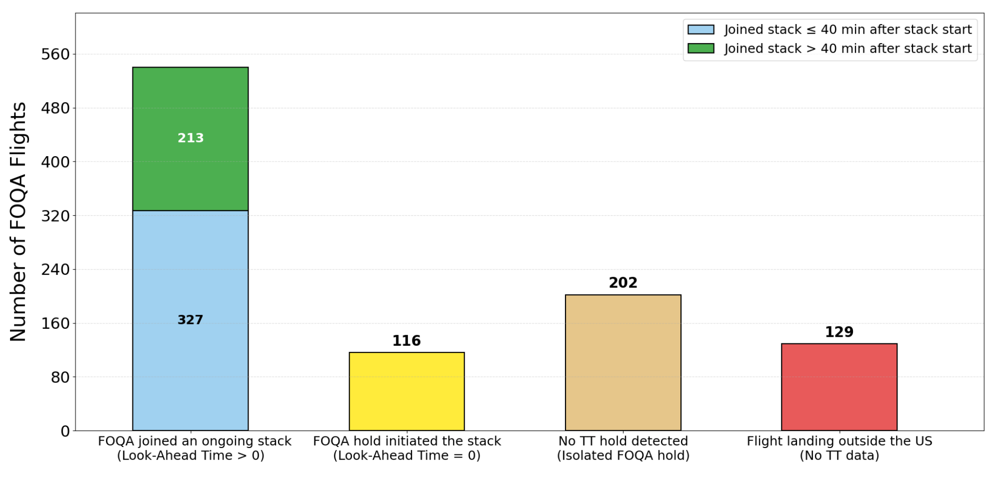


Figure 16. Flight Operations Quality Assurance (FOQA) flights holding stack analysis for 40 min look-ahead time. TT: threaded track.

Flight of Interest for Speed Control Simulation

To determine which FOQA flights represent strong candidates for speed control simulation, the analysis integrates three sources of data: (1) meteorological aerodrome report (METAR) weather observations (ISU, n.d.), (2) on time performance (OTP) (DOT, n.d.), and (3) FAA Air Traffic Control System Command Center Advisories (ATCSCC) (FAA, n.d.). The objective is to identify cases where moderate or severe weather conditions and congestion lead to holding stack formation with sufficient look-ahead time to implement speed control. METAR data provide a time-resolved view of weather evolution leading to each holding event. Seven weather scenarios are considered in this analysis: heavy precipitation, low visibility below 1 statute mile and low ceiling below 200 ft above ground level, thunderstorms accompanied by lightning and precipitation, freezing rain and ice, moderate to heavy snow, gusty winds exceeding 30 kt, and dense fog. OTP data complement this by classifying record delay minutes into five categories of which Extreme Weather and National Aviation System (NAS) are the most relevant for holding events. Extreme Weather captures severe conditions that restrict safe operations while NAS delays include delays from moderate weather conditions, reduced arrival capacity and congestion. FAA ATCSCCs provide a record of flow management actions issued during the holding events such as ground delay programs, ground stops and arrival delay programs.

A flight is considered a strong candidate for speed control simulation when these three data sources, collectively, indicate a holding stack that was initiated due to predictable or progressively deteriorating weather conditions and when the TT data reveal sufficient look-ahead time before FOQA aircraft enters holding. The overall process for identifying flights of interest for speed control simulation is summarized in Figure 17. The holding detecting algorithm first isolates FOQA flights with holding patterns then matches each FOQA flight with its associated domestic TT flights and identifies holds within this set of TT flights. The earliest and latest TT holds identify the holding stack start and end and the FOQA entry time relative to stack start provides the look-ahead time. Flights with look-ahead time greater than 20 min are then retained for further analysis. The second part of the analysis is to identify the potential holding causes. FOQA flights are matched with domestic TT flights in the same stack to retrieve their OTP delay categories. NAS and extreme weather delays are then extracted and METAR data are analyzed to distinguish between congestion-driven and weather-driven events. Relevant ATCSCC advisories associated with the holding stack window are also collected to determine whether the holding stack contributed to airport congestion and the associated impacting conditions. This analysis identified 169 FOQA flights of interest for speed control simulation, with the potential to leverage the associated TT holds to further expand the cases for fuel savings analysis.

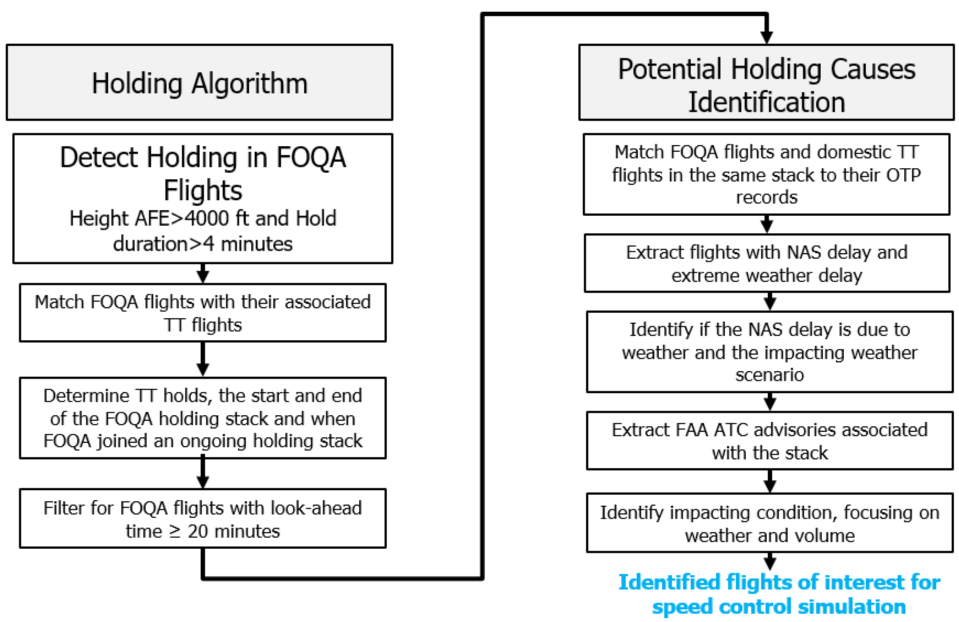


Figure 17. Identifying flights for speed control simulation process. AFE: above field elevation, FOQA: Flight Operations Quality Assurance, NAS: National Aviation System, OTP: on time performance, TT: threaded track.

Example of a Flight of Interest for Speed Control Simulation

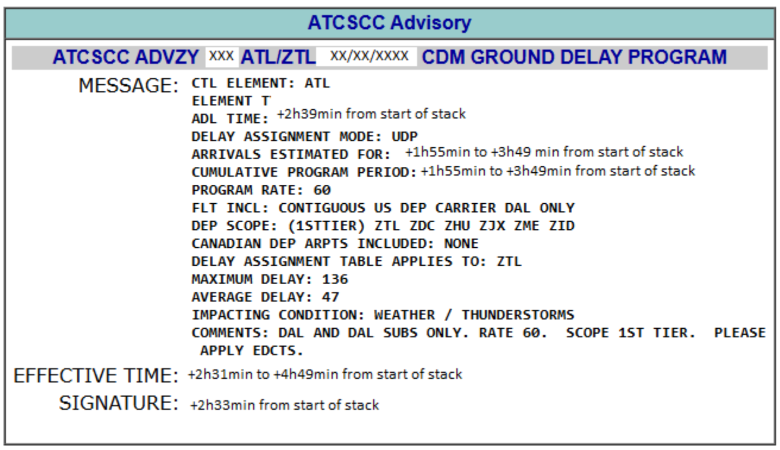
After defining the criteria used to identify FOQA flights suitable for speed control simulation through the combined analysis of METAR weather, OTP data, and FAA ATCSCC advisories, an example helps demonstrate how these inputs come together in an operational context. The following analysis shows an A321 flight originating from KLAX and landing at KATL that entered a holding stack driven by progressively deteriorating weather conditions. METAR observations, presented in Table 2, show early signs of convective development including distant lightning appearing roughly 30 min after the first TT hold, followed by increases in wind gusts, heavy rain and formation of cumulonimbus clouds. Cloud ceilings gradually lowered, and visibility reduced as conditions deteriorated. A summary of METAR abbreviations referenced in this analysis is provided in Table 3. The operational impact of these conditions for the same period is reflected in the OTP records, which indicate 126 min of total arrival delay, of which 101 min were attributed to NAS delay which is consistent with moderate weather or congestion at the landing airport. This delay is further reinforced by showing that the ATCSCC issued a ground delay program at KATL during the same window citing weather and specifically thunderstorms as the impacting weather condition as depicted in Figure 18. Taken together, the METAR, OTP and ATCSSC advisory data collectively provide consistent evidence that the holding stack was driven by approaching convective weather and was formed early enough to support an application of early speed control during cruise.

Table 2. METAR data before and during the holding stack window (ISU, n.d.).

Time relative to start of Holding Stack (HH:MM)	Wind Direction	Wind Speed	Gust	Visibility	Precipitation	Cloud Layer 1	Height 1	Lightning/ Thunderstorm
- 01:00	190°	18 kt	33 kt	8 SM	None	FEW	2900 ft	None
- 00:45	Variable	20 kt	-	9 SM	None	SCT	3100 ft	None
- 00:30	Variable	18 kt	26 kt	9 SM	None	SCT	3100 ft	None
- 00:15	210°	22 kt	28 kt	9 SM	None	BKN	3700 ft	None
Start of Stack	200°	20 kt	30 kt	9 SM	None	SCT	3500 ft	None
+ 00:15	-	19kt	30kt	9 SM	None	BKN	3500 ft	None
+ 00:30	-	18kt	24kt	10 SM	None	SCT	3500 ft	LTG DSNT NW
+ 00:45	210°	19kt	34kt	10 SM	None	SCT	3900 ft	LTG DSNT N
+ 01:00	200°	21kt	28kt	10 SM	None	SCT	3700 ft	LTG DSNT N AND SW
+01:15	270°	-	31kt	5 SM	-RA	SCT	2000 ft	LTG DSNT NE AND SW
+01:30	270°	15kt	22kt	2.5 SM	+RA BR	SCT	1300 ft	LTG DSNT N
+01:45	260°	9 kt	-	7 SM	-RA	FEW	2500 ft	LTG DSNT, TS NE-SE
+02:00	260°	9 kt	14 kt	10 SM	None	FEW	2500 ft	LTG DSNT, CB DSNT
+02:15	260°	10 kt	-	10 SM	None	FEW	2500 ft	LTG DSNT, CB DSNT
+02:30	260°	8 kt	-	10 SM	None	FEW	2500 ft	LTG DSNT, CB DSNT
+02:45	260°	8 kt	-	10 SM	None	FEW	2500 ft	LTG DSNT, CB DSNT
End of Stack +02:50	260°	7 kt	-	10 SM	None	BKN	4500 ft	LTG DSNT, CB DSNT

Table 3. METAR abbreviation guide.

Category	Code	Meaning
Precipitation	-RA	light rain
	+RA	heavy rain
	BR	mist
Lightning or Thunderstorm	DSNT	distant
	LTG	lightning
	TS	thunderstorm
Cloud Layer	FEW	few clouds
	SCT	scattered clouds
	BKN	broken clouds
	CB	cumulonimbus (cloud associated with thunderstorms)


ATCSCC Advisory

ATCSCC ADVZY XXX ATL/ZTL XX/XX/XXXX CDM GROUND DELAY PROGRAM

MESSAGE: CTL ELEMENT: ATL
ELEMENT T
ADL TIME: +2h39min from start of stack
DELAY ASSIGNMENT MODE: UDP
ARRIVALS ESTIMATED FOR: +1h55min to +3h49 min from start of stack
CUMULATIVE PROGRAM PERIOD: +1h55min to +3h49min from start of stack
PROGRAM RATE: 60
FLT INCL: CONTIGUOUS US DEP CARRIER DAL ONLY
DEP SCOPE: (1STTIER) ZTL ZDC ZHU ZJX ZME ZID
CANADIAN DEP ARPTS INCLUDED: NONE
DELAY ASSIGNMENT TABLE APPLIES TO: ZTL
MAXIMUM DELAY: 136
AVERAGE DELAY: 47
IMPACTING CONDITION: WEATHER / THUNDERSTORMS
COMMENTS: DAL AND DAL SUBS ONLY. RATE 60. SCOPE 1ST TIER. PLEASE APPLY EDCTS.

EFFECTIVE TIME: +2h31min to +4h49min from start of stack

SIGNATURE: +2h33min from start of stack

Figure 18. Air Traffic Control System Command Center Advisories (ATCSCC) ground delay program advisory at Hartsfield-Jackson Atlanta International Airport (KATL) (FAA, n.d.).

Summary and Observations of the Holding Detection Algorithm

The holding detection methodology developed in this effort builds upon and extends the approach established in last year's report. The geometric self-intersection logic for FOQA data has been adapted for TT data through the addition of fuzzy logic phase identification, which enables accurate classification of flight phases. This step removes flights with go-arounds and ensures that the holding detection is only applied during cruise and descent phases where holding patterns occur. Once holdings are identified, TT data are used to examine surrounding traffic and identify holding stacks through an iterative query expansion that captures the earliest and latest TT holds around each FOQA event. The relative timing between the onset of the holding stack and the instance the FOQA aircraft enters holding provides the look-ahead time, which indicates whether speed control could be applied early enough during cruise to absorb part or all the hold. The integration of METAR weather observations, OTP delay records and ATCSCC advisories provide insight into what caused each holding event and allows the selection of flights where the holding stack formed due to foreseeable conditions. This methodology has been validated through application to the dataset and has demonstrated reliable results in identifying holding events, holding stacks and the cause behind the stack formation.

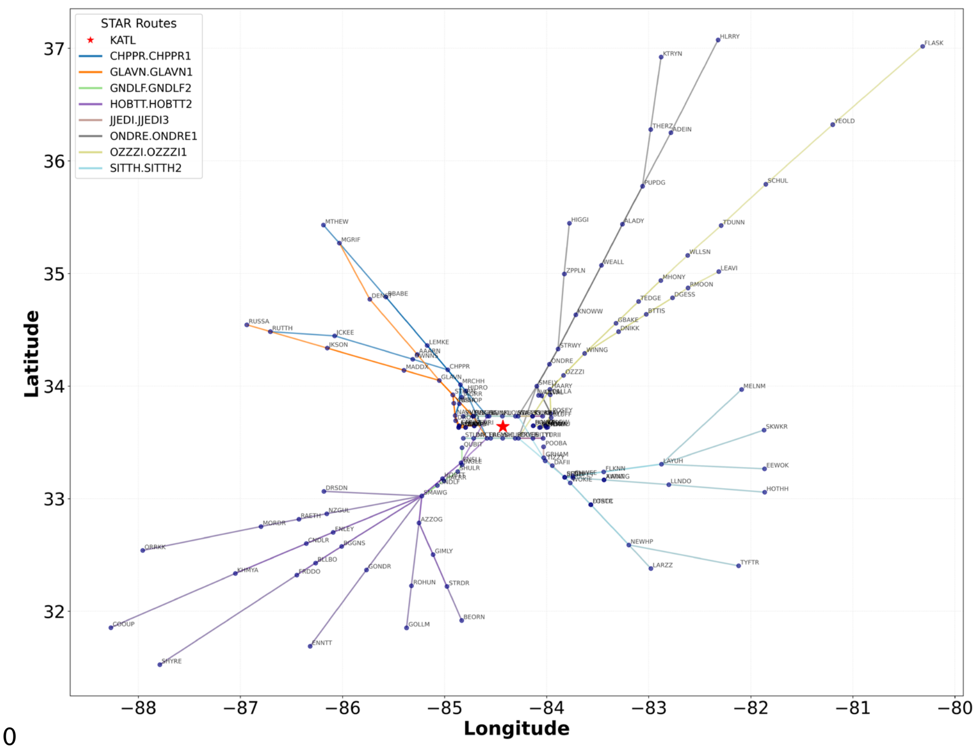
The outputs of this analysis directly support two subsequent tasks in the project. For Task 1, the identified flights of interest provide realistic cases for trajectory evaluation of speed control strategies, with the look-ahead time serving as the key input for determining how early speed control can be initiated and how much of the hold can be absorbed during cruise. Based on the distribution of FOQA flights across holding stack categories, an average look-ahead time of approximately 40 min represents a practical threshold for identifying cases where meaningful fuel benefits from cruise speed control become achievable, with longer cruise segments providing more opportunity for speed reductions and therefore greater fuel savings. This suggests that the primary candidates for speed control are wide-body on long-haul routes and narrow-body aircraft operating cross-continental flights. For Task 4, this methodology enables fleet-wide analysis by identifying all flights entering a holding stack at a given airport, along with their associated look-ahead times, to quantify fuel benefits achievable through coordinated speed control across all affected flights.

Vectoring Identification

Vectoring often occurs when arriving traffic flight plans must be changed due to congestion and weather disruptions to slow down the arrival rate at the destination airport. Vectoring is the navigational guidance that ATC provides to aircraft through instructions on heading, altitude, or speed, causing aircraft to deviate from their nominal arrival paths (SKYbrary Aviation Safety, n.d.). Instead of placing aircraft into holding patterns, controllers use vectoring, such as prolonged downwind legs or zigzagging, to absorb delay and maintain required spacing (Erzberger et al., 2016; FAA, 2026b). However, vectoring can be difficult to detect in trajectory data (Zhang et al., 2022). Challenges include separating vectoring from deviations driven by the choice of original flight plans filed, by weather, or by turbulence and uncovering vectored flights from variations in FOQA metrics. Additionally, differentiating between flights vectored for sequencing or weather and flights vectored due to arrival airport congestion poses a significant challenge. To address these challenges, the initial work is currently assessing various analytical techniques that might be best suited for detecting vectoring in FOQA data, leading to exploration of two potential approaches: a machine learning approach and a clustering-based trajectory analysis method. Given the infancy of the machine-learning approach research, this report focuses on the preliminary results derived from the clustering approach.

Data Preparation for FOQA Calendar Year 2023 and Arrival Routes

Data preparation used FOQA arrival trajectories from calendar year (CY) 2023 along with the corresponding FAA standard terminal arrival (STAR) procedures obtained from the FAA National Airspace System Resources (NASR) database (FAA, 2026a). The STAR data were collected to serve as reference paths for identifying potential ATC vectoring events. When an aircraft follows a particular STAR and then deviates significantly from the published route, this deviation may indicate vectoring by ATC. By comparing FOQA trajectories against published STAR routes, deviations from expected paths can be detected and flagged for further analysis. Flights with holding patterns were removed using the holding pattern detection algorithm, and flights were filtered from top of descent onward to focus on terminal areas as a first step in this analysis before expanding to en-route detection. For the STAR routes, the most recent waypoint information and route segments were selected based on effective dates, as defined by the FAA. The STAR entries for the airport were filtered, their codes identified, and the associated segments collected. The STAR waypoints were then combined with their location data so the routes could be traced and used as reference paths when comparing against FOQA trajectories. An example of 2023 STAR routes for KATL airport is shown in Figure 19.



0

Figure 19. Example of 2023 standard terminal arrival (STAR) routes configuration for Hartsfield-Jackson Atlanta International Airport (KATL).

Machine Learning Approach for Vectoring Identification

The objective is to use machine learning (ML) techniques to characterize ATC vectoring as the first step towards detecting and classifying vectoring events in FOQA data. From the identification of vectoring events, fuel burn impacts due to vectoring can be quantified. The One-Class Support Vector Machine (OC-SVM) is an anomaly detection method that learns the boundary to separating "normal" data from outliers. The OC-SVM approach is trained only with "normal" data, meaning it learns what the typical flight behavior looks like, and anything that deviates significantly is considered an anomaly. For vectoring detection, OC-SVM is fed flight trajectories, determines the regions of data that are "normal," and identifies any deviation from this normal behavior as a vectoring event.

One-Class Support Vector Machine for Vectoring Detection

The OC-SVM, a machine learning technique used in anomaly detection, will be used to detect vectoring events within the flight data (Basora et al., 2019; Guo et al., 2022). In this analysis, the en-route and terminal phases are considered separately due to differing trajectory flows. The en-route phase is characterized by high variability in flows due to flexibility in flight plans, while the terminal phase has lower variability because of sequencing and approach requirements. The initial focus is on detecting vectoring events within the terminal/approach area. The process starts with building a planned route for one airport's arrivals from FOQA flights and STAR data. Then the deviations along each flight, in terms of cross-track distance compared to the "reference route," will be computed. Next, the OC-SVM will be trained using these computed deviations. Finally, this OC-SVM will be used to detect vectoring events as consecutive anomaly seconds exceeding a predefined threshold (e.g., 20 sec). The analysis can be expanded to all flights by looking at the arrivals of all airports in the FOQA and TT data to obtain system-wide insights.

Clustering Approach for Vectoring Identification

A clustering approach is used to group arrival trajectories based on how similar their lateral shapes are, where lateral shapes refer to the horizontal path of the aircraft as projected on the ground. The method relies on the Ordering Points to Identify the Clustering Structure (OPTICS) algorithm (Scikit Learn, 2025), which identifies clusters at different density levels and does not require choosing the number of clusters in advance. The key parameters used are xi, which controls how clusters are separated, and min_samples, which sets the minimum number of flights needed to form a cluster. To compare the shapes of different trajectories, dynamic time warping (DTW) (DTAIDistance, n.d.) is used to measure the similarity between trajectories. DTW is a time series alignment technique that computes the optimal match between two sequences of different lengths or speeds, which makes it well suited for comparing FOQA flights that vary in duration and speed. These parameters provide a way to group flights with similar patterns and identify where deviations occur. To evaluate how the model is performing and the quality of the clusters, silhouette scores are calculated. This metric measures how similar a trajectory is to others in its own cluster compared to trajectories in neighboring clusters, with values ranging from -1 to 1, where higher values indicate well separated clusters.

Clustering Use Case and Preliminary Results

The clustering approach was applied to a two-week set of arrival trajectories into KATL to assess how well the method separates standard arrival flows from trajectories that diverge from the published STAR procedures. Each trajectory was trimmed from top of descent onward before being compared using DTW. An OPTICS parameter sweep over a range of xi and min_samples values was performed, and the silhouette score was used to identify the best parameter values. The configuration that yielded the highest silhouette score was min_samples = 35 and xi = 0.03, with a score of 0.704. This configuration revealed four primary arrival clusters that aligned closely with the STAR routes, as shown in Figure 20, indicating that the method can reasonably capture the dominant flow structure. A substantial fraction of flights was classified as noise, which suggests that additional feature refinement is needed to more reliably isolate vectoring. Overall, the two-week KATL use case demonstrates that clustering provides a useful first step toward identifying trajectories that diverge from expected arrival paths.

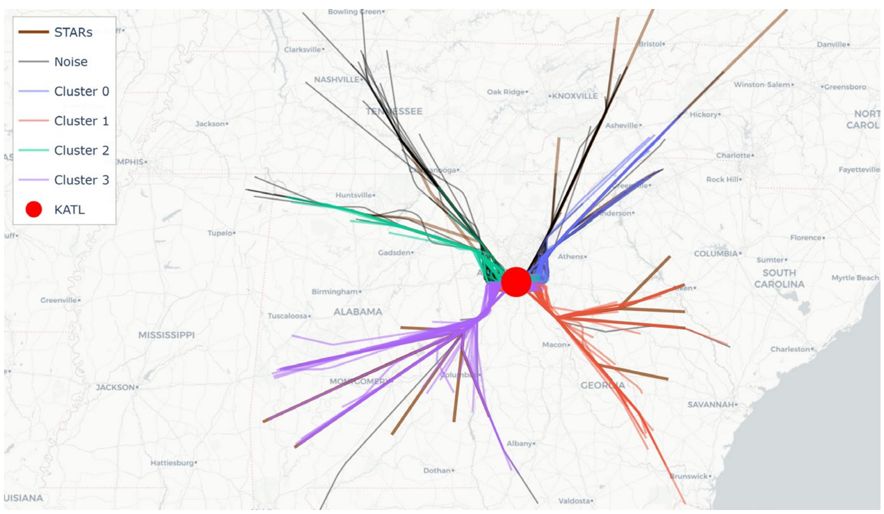


Figure 20. Clustering analysis on Hartsfield-Jackson Atlanta International Airport (KATL) arrivals (two-week period).

Milestones

- Completed speed control detection algorithm and analysis on all widebody 2023 FOQA flights.
- Extended holding pattern detection methodology from FOQA data to TT data.
- Developed an iterative query expansion method to identify holding stacks and calculate look-ahead time calculation for each FOQA flight relative to stack formation.
- Identified flights of interest for speed control simulation.

Major Accomplishments

- Implemented fuzzy logic phase identification to classify flight phases in TT data and remove go-arounds from the analysis.
- Integrated METAR weather observations, OTP delay records and ATCSCC advisories to determine the cause behind holding stack formation.

- Established a look-ahead time of approximately 40 min as a practical threshold for achieving meaningful fuel benefits from cruise speed control.
- Completed speed control algorithm to identify speed control events in the FOQA data.
- Obtained speed control results and performed analysis for 2023 widebody FOQA data.
- Created algorithm to support OC-SVM by assigning each flight its own terminal reference route using STAR data.
- Implemented the first version of the clustering methodology, including DTW similarity computation, and OPTICS clustering.
- Produced the first clustering results for a two-week KATL arrival dataset, identifying four dominant arrival patterns aligned with STAR routes.

Publications
None.

Outreach Efforts
Participated in ASCENT biannual meetings.

Awards
None.

Student Involvement
- Jouwairia Ahabchane (graduate student) created the holding pattern detection algorithm, analyzed the holding events and implemented the vectoring detection using clustering approach
- Vishva Rana (graduate student) worked on the changepoint detection algorithm to identify speed control events and vectoring detection utilizing one-class support vector machine

Plans for Next Period
- Assign each 2023 flight with its own STAR to create a list of “reference routes” for every flight.
- Calculate the deviations between these flights and their respective STARs.
- Train and use the OC-SVM to detect vectoring events.
- Assign STAR procedures to each cluster to better understand how arrival flows relate to published routes and to refine cluster interpretation.
- Incorporate STAR deviation metrics into the clustering analysis, expanding the feature set beyond lateral trajectory shape to help reduce noise.
- Identify vectored flights by combining clustering results with STAR deviation measures and quantify the additional time and fuel burn associated with ATC vectoring.
- Compare the two analytical approaches, the machine-learning method and the clustering-based trajectory analysis, to determine which more reliably identifies vectoring behavior.
- Expand the use case to TT data at a single airport, specifically KATL.

References
Aminikhanghahi, S. & Cook, D. J. (2017). A survey of methods for time series change point detection. Knowledge and Information Systems, 51, 339–367. https://doi.org/10.1007/s10115-016-0987-z

Basora, L., Olive, X., & Dubot, T. (2019). Recent Advances in Anomaly Detection Methods Applied to Aviation. *Aerospace, 6*(117). https://doi.org/10.3390/aerospace6110117

DOT. (n.d.). *On-Time Performance (OTP) data* [Data set]. U.S. Department of Transportation, Bureau of Transportation Statistics.

DTAIDistance, (n.d.). *Dynamic Time Warping (DTW) — usage documentation.* https://dtaidistance.readthedocs.io/en/latest/usage/dtw.html

Erzberger, H., Nikoleris, T., Paielli, R. A., & Chu, YC. (2016). Algorithms for Control of Arrival and Departure Traffic in Terminal Airspace. Proceedings of the Institution of Mechanical Engineers Part G Journal of Aerospace Engineering, 230(9), 1762-1779. https://doi.org//10.1177/0954410016629499

FAA. (n.d.). *Advisory Database Form.* Federal Aviation Administration. https://www.fly.faa.gov/adv/advAdvisoryForm

FAA. (2026a). 28 Day NASR Subscription. Federal Aviation Administration. https://www.faa.gov/air_traffic/flight_info/aeronav/aero_data/NASR_Subscription/
FAA. (2026b). Section 4. ATC Clearances and Aircraft Separation (Ch. 4). In *Aeronautical Information Manual* (Chg. 2). Federal Aviation Administration. https://www.faa.gov/air_traffic/publications/atpubs/aim_html/index.html
Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. Annals of Statistics, 42(6), 2243–2281. https://10.1214/14/AOS1245
Guo, Z., Yin, C., Zeng, W., Tan, X., & Bao, J. (2022). Data-Driven Method for Detecting Flight Trajectory Deviation Anomaly, *Journal of Aerospace Information Systems*, 19(12), 799–810. https://doi.org/10.2514/1.I011124
ISU. (n.d.). Iowa Environmental Mesonet. Iowa State University. https://mesonet.agron.iastate.edu/request/download.phtml
Junzi, S., Ellerbroek, J., & Hoekstra, J. (2017). Flight extraction and phase identification for large automatic dependent surveillance–broadcast datasets. *Journal of aerospace information systems,* 14(10), 566-572. https://doi.org/10.2514/1.I010520
Scikit Learn. (2025). *OPTICS Clustering Algorithm Documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.cluster.OPTICS.html
SKYbrary Aviation Safety. (2025). Basic Controller Techniques: Vectoring | SKYbrary Aviation Safety. https://skybrary.aero/articles/basic-controller-techniques-vectoring
Truong, Oudre, L., & Vayatis, N. (2018). *ruptures: change point detection in Python*. arXiv:1801.00826 (eprint). https://doi.org/10.48550/arXiv.1801.00826
Zhang, S., Zhang, Y., Tay, T., & Shankar, J. (2022). Learning-based Aircraft Trajectory Analysis Tool for Holding and Vectoring Identification with ADS-B Data. *2022 IEEE 25th International Conference on Intelligent Transportation Systems* (ITSC), Macau, China (pp. 1100-1105). https://10.1109/ITSC55140.2022.9921823

Task 3 - Fuel Burn Surrogate Modeling

Georgia Institute of Technology

Objective

The primary objective of this task is to develop a capability to predict fuel flow any time within a flight profile. This granular prediction capability is essential for accurately quantifying the potential fuel benefits derived from optimized flight trajectories. By enabling the evaluation of fuel flow at any given point, this research supports the assessment of en-route speed control to maximize fuel efficiency.

Research Approach

Data Preparation

The study utilized a massive dataset comprising 50,000 FOQA flights from 2023, spanning both widebody (e.g., A350, B767) and narrow-body (e.g., A321, B737) aircraft categories. A flight segmentation algorithm was developed to partition flight data into distinct main flight phases: climb, cruise, cruise step-climbs, and descent. The analysis placed a specific focus on the cruise segment which is defined as the period between top of climb (ToC) and ToD capturing both steady altitude phases and step climbs (i.e., high rate of climb and altitude jumps). To ensure data quality and remove high-frequency noise, a 1-min moving average smoothing technique was applied to critical variables, including mean sea level (MSL) altitude, temperature, corrected gross weight, Mach number, and total fuel flow. Following this, the processed segments were concatenated by aircraft type to form a unified training dataset.

Modeling Methodology

The modeling framework evolved from the previous tree-based approach to a regularized polynomial regression architecture, balancing predictive performance with physical interpretability. While previously, the tree-based models that used XGBOOST (XGBoost Developers, 2025) demonstrated higher accuracy, it functioned as a black box model and produced counterintuitive, non-smooth, relationships between the key model features. To address this, the current methodology shifted to a regularized least squares regression using a third-degree polynomial model. This approach provides the general polynomial equation used in making the fuel flow predictions, and it ensures smooth physical behavior between the model features. The polynomial model framework was used in building fuel flow models for all aircraft in the FOQA data. Within this framework, a comparative assessment was conducted between Elastic Net (Truong et al., 2018) (optimizing both alpha and the L1/L2 ratio) and L1 regularization (Least Absolute Shrinkage and Selection Operator) (LASSO) (Scikit-learn, n.d.). While Elastic Net regularization constrained coefficients magnitudes, it yielded lower predictive accuracy compared to the L1 approach. Consequently, the LASSO technique was selected for model architecture (Scikit Learn, n.d.).

The modeling process was grounded in feature selection and pre-processing protocol (see Figure 21). To ensure model stability and exclude sensor anomalies or extreme outliers, the training data were trimmed to strictly include values between the 5th and 96th percentiles. The core feature set defined for the cruise phase included best available MSL altitude (ft), outside air temperature (°C), corrected gross weight (lb), calculated Mach, and a temporal feature, Mach change over 20 s as an acceleration proxy, to capture transient speed effects. For step climb segments, this feature set was augmented with rate of climb, with fuel flow total (lb/hr) serving as the target response variable for all models. These features served as inputs to the LASSO framework. By applying L1 regularization, the model automatically performed feature selection, isolating the most significant polynomial terms and shrinking redundant terms to zero. To validate the model's robustness, a comprehensive cross-validation strategy was employed. This process involved iteratively splitting the data into training and validation subsets to ensure that the selected hyperparameters and polynomial terms generalize well to unseen data, rather than overfitting specific trajectory characteristics found in the training set. This training methodology was batch-applied consistently across all airframes in the dataset, ensuring a standardized modeling approach while producing a unique, optimized coefficient set for each specific aircraft type depending on their fuel flow behavior.

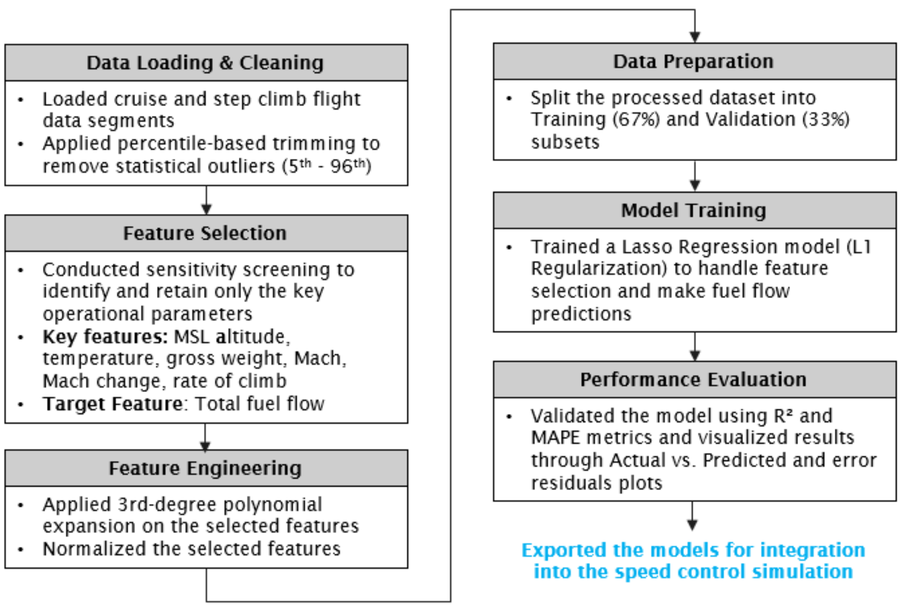


Figure 21. Surrogate model building process. MAPE: mean absolute percent error, MSL: mean sea level.

Models Validation

The validation assessment relied on a segmented analysis to evaluate model fidelity across distinct flight regimes. To assess model accuracy, performance metrics such as mean absolute percent error (MAPE) and R^2 were calculated separately for steady-state cruise and step-climb phases. The comprehensive breakdown of the analyzed fleet, including specific engine configurations, valid Mach ranges, and the volume of flight data for each airframe is listed in Table 4. Notably, the number of available flights varies across the fleet, a factor that directly influences the data density available for training and validation. Despite this variance, the surrogate models were validated across diverse operational profiles. The table lists the aggregate validation metrics for both flight phases, providing the numerical basis for the comparative analysis presented in the subsequent figures.

A side-by-side comparison of model performance across the fleet for both steady cruise and step climb phases is depicted in Figures 22 and 23. The steady cruise models consistently outperform the step climb models, maintaining error rate below 2.5% as shown in Figure 22. However, (R^2) shows that several airframes, particularly narrowbodies such as the 757-300 exhibit significantly higher R^2 values for step climb despite having high error rates, this potentially stems from the variance in the target variable in the step climb model, conversely, the low variance in target value for the steady-state cruise model makes achieving a high R^2 challenging as seen in Figure 23. This suggests that the achieved batch processing of all airframes using one modeling framework may not be ideal; segmentation strategies by tail number or different modeling techniques might be required to harmonize the accuracy metrics across the different flight phases. Current investigations are exploring advanced mitigation strategies, such as refined smoothing techniques and the previously mentioned segmentation techniques, to better isolate these transient behaviors and improve prediction accuracy for both phase models.

The validation residuals and actual-by-predicted results for a wide-body aircraft cruise segment are depicted in Figure 25. The data cover a typical operational cruise envelope, spanning fuel flow rates from approximately 10,500 lb/hr to 13,500 lb/hr. Across this range, the percentage error is primarily bound between -5% and +5%, with a MAPE of 1.84% and R^2 of 0.84.

Table 4. Comprehensive summary of the analyzed fleet parameters and validation metrics for cruise and Step Climb (SC) phases. FOQA: Flight Operations Quality Assurance, MAPE: mean absolute percent error.

Airframe with Engine Type	FOQA Flight Count	Cruise Mach Range	Cruise R²	Cruise MAPE	SC R²	SC MAPE
A350-900 Trent 1000	675	0.79 - 0.87	0.80	2.42%	0.85	3.37%
A330-900 Trent 7000	598	0.76 - 0.83	0.71	2.44%	0.79	3.28%
A330-300-CF6-80	311	0.78 - 0.83	0.75	2.11%	0.72	3.68%
A330-300 PW4000-100	450	0.78 - 0.83	0.76	2.02%	0.84	3.46%
A330-200 PW4000-100	232	0.78 - 0.83	0.77	2.09%	0.84	3.52%
A321-200 PW1100G	2173	0.74 - 0.79	0.66	1.89%	0.82	5.32%
A321-200 CFM56-5B-C	10271	0.74 - 0.80	0.58	2.06%	0.85	5.19%
A320-200 CFM56-5A	4665	0.74 - 0.80	0.54	2.13%	0.71	5.36%
A319-100 CFM56-5A	4100	0.75 - 0.80	0.59	2.07%	0.64	5.61%
A220-300 PW1500G	915	0.75 - 0.79	0.59	2.28%	0.71	4.85%
A220-100 PW1500G	1768	0.75 - 0.80	0.59	2.32%	0.75	5.19%
767-400/ER CF6-80	244	0.76 - 0.83	0.84	1.83%	0.87	3.45%
767-300/ER CF6-80	297	0.78 - 0.83	0.78	1.85%	0.85	3.58%
767-300/ER PW4000-94	773	0.76 - 0.83	0.78	2.01%	0.84	3.59%
757-300 PW2000	701	0.75 - 0.82	0.53	2.19%	0.88	4.51%
757-200 PW2000	5438	0.76 - 0.81	0.61	2.63%	0.76	6.02%
737-900 CFM56-7	9371	0.75 - 0.80	0.69	1.75%	0.89	4.62%
737-800 CFM56-7	3236	0.73 - 0.80	0.63	2.03%	0.88	4.92%
717-200 BR715	3706	0.70 - 0.78	0.61	1.90%	0.63	4.32%

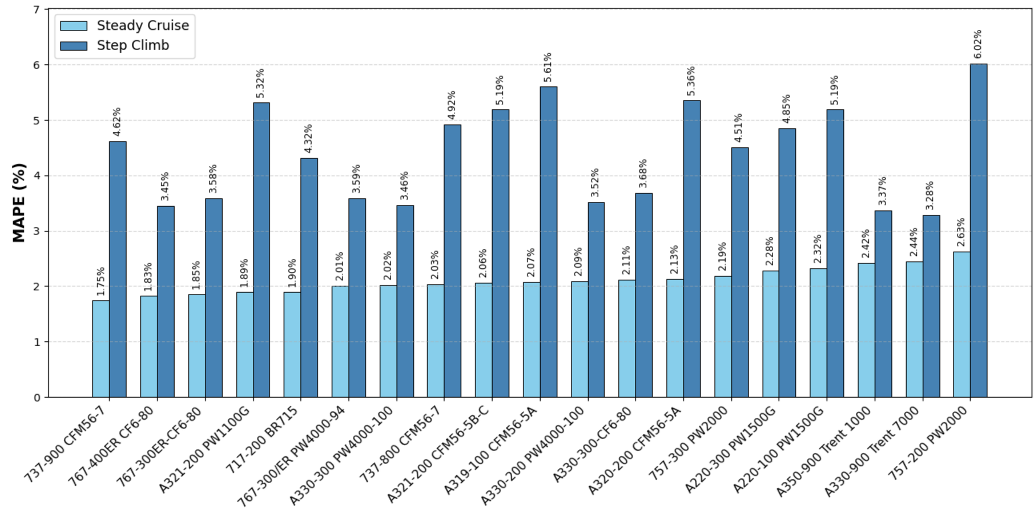


Figure 22. Comparative distribution of mean absolute percent error (MAPE) across the analyzed fleet for step cruise and step climb (SC) phases.

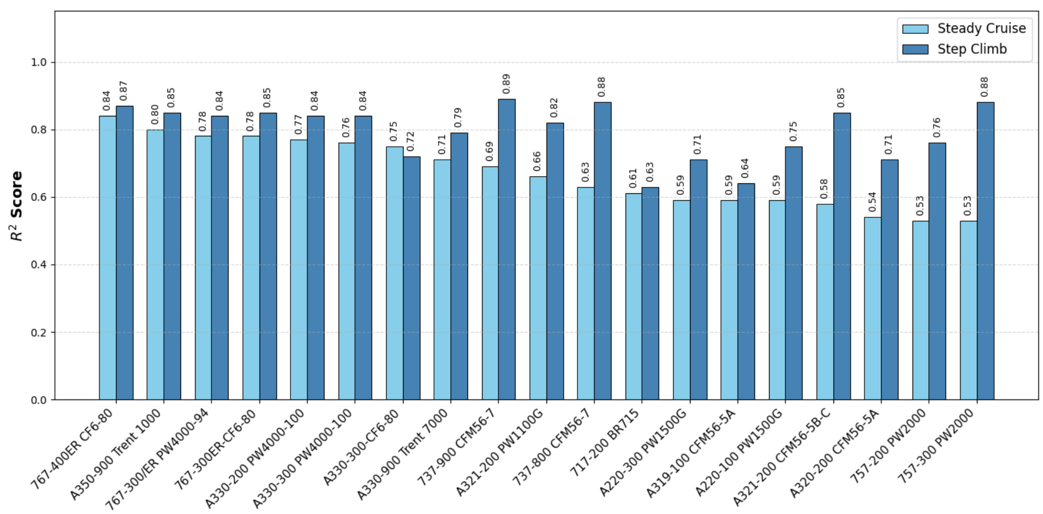


Figure 23. Comparative distribution of R^2 across the analyzed fleet for steady cruise and step climb (SC) phases.

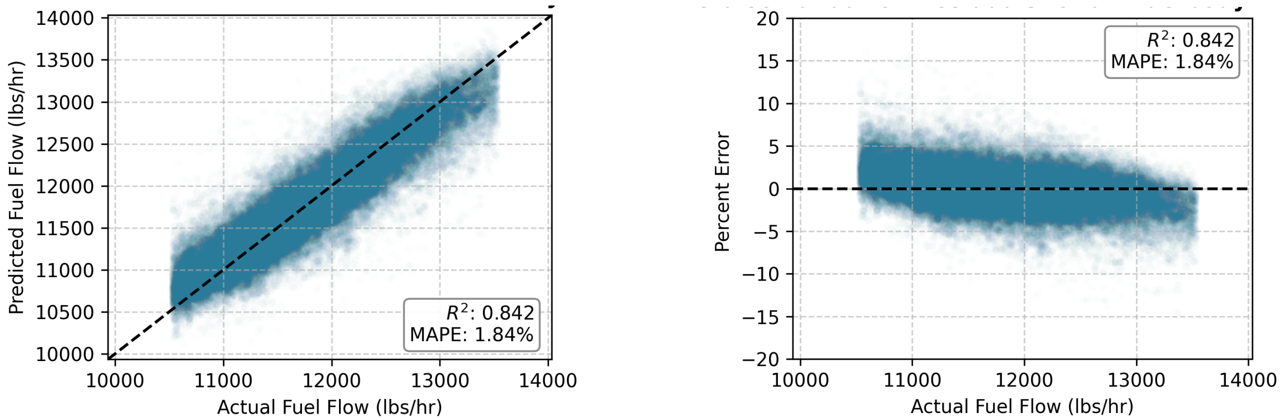


Figure 24. Wide-body aircraft actual by predicted (left) and percentage error residuals (right) of fuel flow prediction. MAPE: mean absolute percent error.

Milestones

- Developed surrogate models for all aircraft in the FOQA fleet for cruise and step-climb phases.

Major Accomplishments

- Successfully developed LASSO-regularized polynomial models that achieved a fleet-wide MAPE of approximately 2.1% during steady-state cruise (ranging from 1.75% to 2.63% by airframe). This validates the surrogate model's ability to provide predictions of fuel consumption under standard operating conditions.
- Demonstrated that these current surrogates operate orders of magnitude faster in simulation compared to previously attempted modeling techniques, this speed is critical for enabling large-scale fleet analysis and optimization.

Publications

None.

Outreach Efforts

Participated in ASCENT biannual meetings.

Awards

None.

Student Involvement

Ghizlane Jari (graduate student) was responsible for all elements of this task of creating fuel flow surrogate models

Plans for Next Period

The research roadmap for the upcoming period centers on the refinement of the existing surrogate framework. A primary objective is to enhance the model's handling of dynamic segments, specifically step climbs, to reduce prediction error. Concurrently, the research scope will expand to potentially include the development and validation of surrogate models for remaining flight phases.

References

Scikit Learn. (n.d.). *Lasso.* https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
Scikit-learn. (n.d.). *ElasticNet.* https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html
Truong, Oudre, L., & Vayatis, N. (2018). *ruptures: change point detection in Python.* arXiv:1801.00826 (eprint). https://doi.org/10.48550/arXiv.1801.00826

XGBoost Developers. (2025). *XGBoost Documentation — xgboost 3.1.1 documentation*. (2025). https://xgboost.readthedocs.io/en/stable/

Task 4 – Fleet-wide speed control strategy

Objective
The objective of this task is to develop an initial fleet-wide speed control strategy to maximize total fuel savings while respecting various operational constraints. This effort builds upon the single-flight analyses by extending the methodology developed to a fleet-wide scope. To achieve this, TT data are needed to augment the available FOQA data to capture all the flights by all airlines that might be impacted by predictable delays and that might benefit from some speed control tactics. Therefore, this task aims to generalize the quantification of fuel savings from single flights to all flights impacted by delay-inducing events.

Research Approach
In tasks highlighted previously, a simulation is developed to estimate fuel burn on a trajectory: a trajectory described in FOQA data is used as a baseline trajectory and speed control tactics are then investigated to see how much fuel could be saved by selecting a lower cruise speed and thus absorbing some of the time that would otherwise be spent in holding patterns. The trajectory simulation runs cases at different Mach numbers and for different amounts of absorbed time. It then outputs the fuel savings for each case alongside the duration of the speed control slow-down and the amount of fuel saved. This task aims to expand the single flight analysis to a system-wide analysis. While the holding pattern detection algorithm helped identify FOQA flights with holding patterns for which the simulation can be leveraged, this represents neither an event-wide assessment nor a system-wide assessment of fuel burn benefits as many other flights are also impacted and could benefit from slowing down early. A more comprehensive database of flights needs to be used, and the research team decided to augment the FOQA data with TT data. For example, for a severe weather event at an arrival airport, aircraft will enter holding patterns such as the ones depicted in Figure 25, where the aircraft in green represent flights for which the research team has FOQA data, while the other aircraft depicted in black have TT data. This task is an initial attempt to quantify a system level disruption event at a single airport and is in its infancy stage of formulation for the current effort.

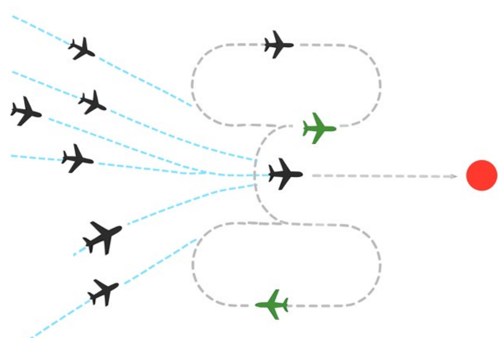


Figure 25. Multiple flights arriving at airport with a holding event due to severe weather.

Data Preparation
To carry out an event-wide or system-wide assessment, TT data are used to account for most of the air traffic coming into an airport where some disruption is first detected in the FOQA data. However, two challenges arise when using TT data. First, unlike FOQA data, which were limited to the fleet of aircraft operated by one major air carrier that provided the data, this is not inclusive of all aircraft types that operate on any given day. Hence, the surrogate models of aircraft fuel flow created in Task 3 do not cover the entire fleet of aircraft that may be impacted by terminal delays. Assumptions must be made for the initial system-wide assessment, and aircraft that are deemed similar will use the same fuel burn surrogate models. The fuel flow surrogates models are currently limited to a subset of narrow and wide-body aircraft and do not include turboprops, regional jets, business jets, and other general aviation aircraft. For instance, the A350-900 surrogate model could be used as a proxy to represent the fuel burn of a B787-10. Second, TT data are not as *parameter-rich* as FOQA data and a few parameters essential to estimate fuel flow are not available, as listed in Table 5. Indeed, TT data are

essentially a reconstructed continuous flight path that is processed, sequenced, and cleaned. However, it is missing environmental information such as wind direction, wind speed, outside air temperature, as well as operational parameters such as aircraft gross weight and Mach number. As a result, data pre-processing is needed to synthesize some of these parameters.

Table 5. Comparison of Flight Operations Quality Assurance (FOQA) data and threaded track (TT) data.

Category	FOQA	Threaded Track
Trajectory data	Sampled at 1Hz: Time, Latitude, Longitude, Altitude, True air speed, Ground speed and Heading	Non-uniformly sampled with up to 500s between entries, Latitude, Longitude, Altitude, True air speed and Ground speed
Aircraft/Engine data	Gross weight, fuel flow total, Total fuel used (all engines)	N/A
Environmental data	Wind speed, Wind direction, Drift angle, Air temperature and Headwind	N/A

To augment the TT data with the missing parameters needed for the fuel flow surrogates, several assumptions are initially made. First, all flights are modeled under zero wind conditions and standard atmospheric conditions In addition, the gross weight of the aircraft must be estimated. This is a challenging task as the aircraft gross weight in the TT data is unknown for each aircraft type flight. As such, approximation is needed based on using the weight of aircraft with similar capabilities and with similar remaining distances to fly based on the FOQA data and fuel flow surrogates developed in Task 3. This approach assumes that aircraft of similar sizes with similar distances to cover until their destination will statistically have similar gross weights. Although this is a simplifying assumption, it is likely reasonable first step to understand the system's behavior. FOQA data are leveraged to generate this surrogate model by regressing aircraft's gross weight against the remaining distance to destination for each subfleet of aircraft. These assumptions serve the first iteration, with future refinements that will include more realistic wind conditions.

Methodology

This initial framing of the methodology aims to investigate the fuel-saving potential of fleet wide speed control strategies while respecting operational requirements such as arrival sequencing and arrival times to the airport with a holding event. Several research questions arise as the team is formulating an approach for the assessment at a single airport for a single day. To understand the complexity of the problem, an example of weather disruption at KATL was investigated based on the holding detection analysis of the FOQA and TT data in Task 2. Flights arriving at KATL for a day in 2023 and entering a holding pattern to build a holding stack are listed in Table 6, where the FOQA flight is highlighted in green. As evident, there is a diversity of aircraft types, when a flight entered the holding stack, the starting and ending altitude of the first holding loop (although multiple may exist for a given flight), the duration of that loop, and the look ahead time.

Table 6. A stack of flights arriving at Hartsfield-Jackson Atlanta International Airport (KATL) and entering holding patterns due to severe weather.

c	Departure Airport	Arrival Airport	Loop 1 Start Altitude (ft)	Loop 1 End Altitude (ft)	Loop 1 Duration (min)	Look Ahead Time (min)
B752	KORF	KATL	8,048	4,037	31.1	0
B739	KSTL	KATL	6,530	5,027	16.1	3.42
B752	TIST	KATL	25,000	24,974	27.1	12.52
A320	MMMY	KATL	20,986	18,000	45.2	15.82
A20N	KTPA	KATL	33,002	33,000	42.7	20.68
A319	KDAL	KATL	22,000	22,003	39.9	25.48
A321	KDFW	KATL	24,839	21,002	34.9	27.12
A321	KDEN	KATL	27,934	24,000	27.6	36.45
B739	KSEA	KATL	26,268	24,999	23.4	42.30
A321	KOMA	KATL	34,000	29,000	48.4	43.52
B744	KORD	KATL	32,720	31,000	34.1	49.60
B763	SBGR	KATL	37,000	36,000	29.2	51.35
B738	TISX	KATL	35,000	35,000	21.9	53.90
A21N	KBOS	KATL	28,000	28,000	22	72.25
B739	MYNN	KATL	17,996	11,919	10.3	77.18
B739	KSFO	KATL	20,995	14,011	27.2	113.30
A21N	KDEN	KATL	24,006	20,511	23.6	119.82
B752	MGGT	KATL	23,714	21,000	17.6	120.20
B39M	KSEA	KATL	21,000	16,956	18.6	121.60
B739	MSLP	KATL	22,000	21,000	6.5	123.82
B738	KCOS	KATL	26,000	16,228	20.4	125.90
B752	MMUN	KATL	28,000	21,000	18.8	132.80
B739	TNCB	KATL	30,470	12,861	15.3	135.32
A359	RKSI	KATL	32,004	26,110	10.6	135.45
B752	KBZN	KATL	33,000	24,003	13.8	140.90
B739	MRLB	KATL	25,980	24,006	7.4	145.00
B752	KSNA	KATL	27835	23136	9.6	152.12
B752	TJSJ	KATL	29771	14920	10.6	156.50

This example raises several research questions to approach the methodology formulation. First, given the large variation in look ahead and duration of the first loop time, which flights have sufficient time during cruise to absorb the delay. Not all flights have the same look ahead time, nor total cruise time nor duration of the hold. Additionally, the sequence at which a flight enters and exits the stack is important to maintain the landing order, since the current research effort scope does not model airport landing capacity. Hence, how can that flight exit time of a holding stack be maintained. Depending on the specific aircraft type, average cruise speed, and length of cruise, how much time can be absorbed? And lastly, how can the absorption be distributed across the stack of flight to gain the most fuel savings from a system perspective? There exist numerous strategies that could be evaluated. At the extreme lower bound of fuel savings, all flights would go into the hold as it was, and no fuel savings would exist. At the other end, each individual flight would independently apply speed control, if feasible during cruise.

Bounding the problem in this way allows the identification of minimum and maximum levels of delay absorption that are operationally feasible, recognizing that many possible strategies exist between these two extremes. For instance, one strategy could be targeted at aircraft with highest fuel burn, prioritizing speed control for aircraft in the stack that fits that criteria. However, before exploring a range of operationally feasible tactics, examining the lower and upper bounds defines the outer limits of what is achievable. At the upper bound, each aircraft absorbs its individual optimal amount of delay, maximizing fuel savings but potentially altering the arrival times and disregarding arrival sequence in Atlanta. Therefore, flights with a longer look-ahead would absorb more of their terminal delays than others. This could, however, change the sequencing of flights exiting the holding pattern and lead to fairness issues. At the lower bound, all aircraft absorb the same amount of delay based on the flight with the shortest available look-ahead time. This preserves sequencing order and arrival times but restricts the achievable fuel benefits. In this case, fairness is not violated as the order aircraft leave the holding pattern remains unchanged at the cost of absorbing less delay. Quantifying these two limits provides the basis for developing operationally feasible tactics. Ultimately, the goal is to design practical, scalable speed-control strategies that deliver fleet-wide fuel benefits while respecting real-world constraints.

Milestones
None.

Major Accomplishments
Initial framing of the approach to assess an airport level speed control strategy to absorb delay.

Publications
None.

Outreach Efforts
Participated in ASCENT biannual meetings.

Awards
None.

Student Involvement
Yasser Moumtaz (graduate student) was responsible for all elements of enhancing Threaded Track data to resemble FOQA and enable fleet-wide analysis

Plans for Next Period
- Develop surrogate models for gross weight estimation from FOQA data.
- Map the TT aircraft types to the FOQA fuel flow surrogates.
- Execute a test case example for the upper and lower bounds.