# Modeling Continuous Descent Approaches for Decarbonizing Aviation through Monte Carlo Analysis of Thrust Inputs

Kaden Chang

## Abstract

Aviation is one of the leading causes of harmful emissions in the transportation industry. Airplanes burn fossil fuels and release contrails during flight, contributing to air pollution and magnifying the greenhouse effect [1]. In this study, a model was developed to analyze one potential solution to reducing aviation's carbon footprint: continuous descent arrivals (CDA). This model utilized Python to set atmospheric variables and parameters for the airplane as it landed through a CDA, without the steps typically seen in a step-down approach. CDAs are beneficial because they eliminate the need for thrust and the associated emissions when they come in for a step-down approach, because there is no longer the need for the thrust that maintains the altitude at each stage in the descent. This model varied thrust input and input time throughout a simulated CDA in a Monte Carlo simulation. The results show that while CDAs have the potential to greatly reduce aviation's emissions, they still require some thrust input to ensure the aircraft makes the runway. The timing and magnitude of this thrust input must be adjusted correctly, or there is the risk of missing the airport or landing too fast or too slow. This study found that when thrust of magnitudes anywhere from 70 to 100 kN was input about halfway through the descent, around 1000 seconds into the CDA, it resulted in the most successful landings. Eventually, models like this will allow for the development of CDAs that do not require any thrust input, putting the aviation industry a step in the right direction.

## Introduction

Aviation is responsible for 2.4% of global CO2 emissions. When combined with the other gases and contrails that aircraft release, it is responsible for 5% of global warming [1]. In addition to its already substantial impact on the environment, aviation grew faster than any other major form of transportation between 2000 and 2019 [2]. One potential way to reduce the emissions that are released by airplanes is to remove thrust in descents. This can be achieved by opting for a continuous descent approach (CDA) rather than a step-down approach – the traditional approach strategy for most commercial airlines. A study of the implementation of CDAs at seven major airports in China found that they reduce fuel consumption by 139kg per flight. This results in a direct reduction in emissions during the descent phase of flight and has the potential to reduce CO2 emissions across China by 67.6 metric tons (Mt) from 2025 to 2050 [3]. A metric ton equals 1000 kilograms (kg) or 2204 pounds (lbs). The FAA has begun implementing CDAs at major airports throughout the US, estimating that they have saved 2 million gallons of fuel and reduced 40 million pounds of emissions annually at each airport. This is equivalent to 1,300 Boeing 737 flights between Atlanta and Dallas completely cutting their emissions [4].

The reason that this can reduce fuel consumption and emissions is because it removes the fuel burn aspect in a step-down approach. A step-down approach involves descending to a restricted altitude and maintaining that altitude before descending again to the next cleared

altitude and so on until landing. This is what is used by many commercial airlines, but despite descending, it continues to burn fuel. At each step in the approach, thrust will be required to maintain level altitude until the aircraft is cleared to descend to the next altitude or land; this results in fuel burn and emissions. CDAs allow for an aircraft to power down to idle or near idle and simply glide down to the runway in one smooth, continuous motion, eliminating the need for thrust input at each stage of the descent. The goal of this study is to create a model that allows us to quickly simulate a CDA and understand the parameters that define whether it is successful or not.

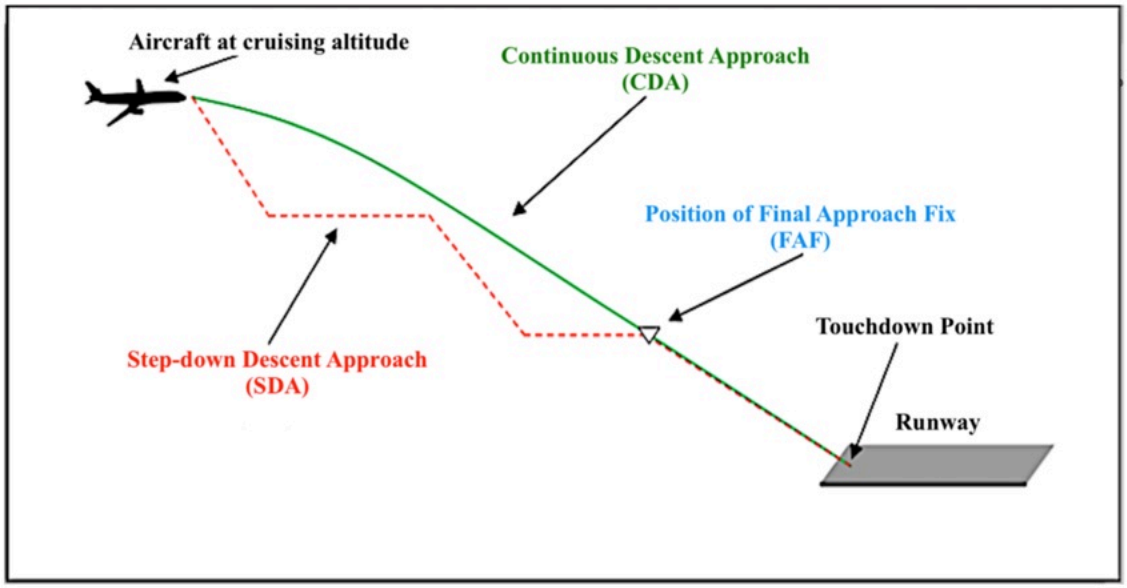



**Figure 1:** A visual representation of a CDA vs. a Step-Down Approach. The dashed red line represents the traditional step-down approach, whereas the green line represents the continuous descent approach as described in the introduction [5].

## Methodology

The method used for this paper was Monte Carlo simulations. A Monte Carlo simulation leverages statistical sampling to investigate different variables input into the simulations and creates a model of possible results, making it appropriate for this study on the parameters that define a successful CDA [6]. Various combinations of initial altitude, airspeed, and thrust input were sampled in these Monte Carlo simulations. First, a simulation loop that simulated a controlled descent arrival (CDA) of a Boeing 737 or an Airbus A320 was developed using Python. This involved establishing physical constants. These were the acceleration due to gravity (g), wing reference area (S), initial aircraft mass (m), sea-level air density ($\rho_0$), scale height for the atmospheric density exponential model (H_scale), mean aerodynamic chord (L), and the dynamic viscosity of air at sea level (μ), as well as establishing the value of thrust as zero. These values were set and would not change throughout the simulation. Acceleration due to gravity was set at the standard value of 9.81 m/s^2. Wing reference area was 122.6 m^2, and initial weight was set at 60,000 kg, about 5,000 kg below maximum landing weight [7]. Sea-level air density was set at 1.225 kg/m^3 [8]. The H scale was set at 8.5 km [9]. The mean

aerodynamic chord was set at 4.2 m [10]. The dynamic viscosity of air at sea level was set at 1.789e^-5 kg/m/s [8]. Thrust was set at zero because the CDA intends to eliminate the need for thrust and decrease descent emissions. These values are those for an A320 aircraft, which is widley considered to be similar to a B737. The initial states were set at a normal cruising altitude of 10,000 meters and a True Airspeed (TAS) of 425 knots (KTAS). The glide slope was initially set at 3 degrees but later changed to 2.6 degrees. This change was made to allow the simulated aircraft to slow below the threshold for flap deployment, which it was unable to do at the steeper 3-degree glide slope. These values were selected because they are accurate to actual initial circumstances for a B737's descent, although the 2.6-degree glide slope is a little steeper than the normal approach. After all, the aim is to eliminate the need for thrust input near the end of the descent to reach the runway.

Within the simulation loop itself, to have a calculation of air density at every point in the model, the Isothermal Scale Height Formula (Equation 1) was used, where ρ is air density, $\rho_0$ is sea-level air density, h is altitude above sea level (m), and H is the scale height for the atmospheric density exponential model [11].

$$\rho(h) = \rho_0 e^{-\frac{h}{H}} \quad (1)$$

This calculation of air density was done to allow for a calculation of the Reynolds Number during the simulation. The Reynolds Number is a dimensionless, unitless number that represents the ratio of inertial forces and viscous forces. This is important because air is a fluid and therefore contains viscous forces that will affect how an aircraft behaves. It is represented by Equation 2:

$$Re = \frac{\rho V L}{\mu} \quad (2)$$

Re is the Reynolds number, ρ is the air density, V is the true airspeed, L is the mean aerodynamic chord, and μ is the dynamic viscosity of air [12]. However, since the simulation is simply a model and not completely accurate to an actual airplane, a correction is required. For this, I used a logarithmic Reynolds number scaling factor derived from the Prandtl-Schlichting skin friction relation (Equation 3) [13]:

$$C_f = \frac{0.455}{(logRe)^{2.58}} \quad (3)$$

Scaling the Reynolds number to $10^6$ allows for a model to match full-scale conditions, as seen in Equation 4 [14].

$$\frac{C_D(Re)}{C_D(10^6)} = \left(\frac{Re}{10^6}\right)^{-n} \quad (4)$$

Applying a first-order Taylor polynomial expansion gives Equation 5:

$$(\frac{Re}{10^6})^{-n} = 1 - nln(10)log(\frac{Re}{10^6}) \quad (5)$$

Using an estimated value of n = 0.0087, $nln(10) = 0.02$, the final Reynolds number correction factor is given by Equation 6.

$$Re_{correction} = 1.0 - 0.02log(\frac{Re}{10^6}) \quad (6)$$

This factor was then used to calculate parasitic drag to be combined with the induced drag and calculate total drag as seen in Equation 7, where $C_L$ is the coefficient of lift and K is the induced drag factor.

$$C_D = (C_{D0} * Re_{correction}) + KC_L^2 \quad (7)$$

Since real airplanes deploy flaps before landing, flaps were added. Conditions for airspeed were added, stating that if indicated airspeed (IAS) was less than 108.033, flaps state = 1. This changed the coefficient of lift to 0.9, the coefficient of drag to 0.045, and K to 0.050. If IAS dropped below 92.6, flaps state = 2. Coefficient of lift was changed to 1.4, coefficient of drag was changed to 0.075, and K to 0.055. These values were estimates because we were not able to find specific data on the flap deployment schedule of commercial A320s. Then, a 2-minute three-degree turn was added to implement another factor that could impact the CDA. This is a standard rate turn, so the two-minute time means that the end heading of the aircraft will not be affected [15]. Since adding a turn greatly reduces the coefficient of lift by redirecting some of the lift into a horizontal component, we determined that it might be necessary to add some thrust at some point during the flight to maintain a consistent coefficient of lift and allow the airplane to reach the airport at an appropriate landing speed [16]. This is where the Monte Carlo simulation comes into play.

Since thrust was initially a set value in the original code, it was changed to be set as a random value between 10,000 and 100,000 newtons (N) for 30 seconds at a random time between 0 and 2000 seconds, the total time of the simulation. These values were recorded for each run. Preliminary values for the location of the “airport” were set at x = 0 nautical miles (NM) and y = 50 NM because the plane begins and ends on a heading of 90 degrees due north, or straight along the y-axis. The range of tolerance for the landing zone was set within one nautical mile of the airport. This value was chosen because it allows for the plane to get sufficiently close to the airport to consider a runway made, especially for such a preliminary study. Future studies should reduce this range of tolerance. The target landing speed was set at 150 KTAS (knots true airspeed), which is roughly equivalent to 150 KIAS (knots indicated airspeed), the actual value used by pilots. The range of tolerance for landing speed was set at 15 knots. This was chosen because it allows for some variation in landing speed without allowing the aircraft to be

excessively fast or slow. The conditions for success were set as being within the range of tolerance for both landing zone and landing speed. At first, only 1,000 runs were allowed, and the airport location was finalized to be at (0,107). After this, 10,000 runs were allowed. The results were graphed on two figures. The first, Figure 6 is a map of the landing zone showing the airport, area around the airport, and where each trial landed. Figure 7 is the operation window of thrust magnitude vs thrust input time. The code used in the Monte Carlo simulation can be found in Appendix B.

Lastly, a sensitivity study to determine how sensitive the success of the simulation was to different variables was executed. Using this with a Monte Carlo Method is ideal because it is probabilistic rather than deterministic and will allow us to see which variables affect the outcome the most [17]. The simulation was modified to test how sensitive the success rate was to initial altitude and flap speeds. Rather than having them be fixed values, the initial altitude and flap 1 and 2 deployment speeds were added to an array and varied. The initial altitudes tested were 8000, 9000, 10000, 11000, and 12000 meters. Each of the flap speeds had three settings: the original and plus or minus 10 m/s. These were plotted on a heat graph to provide an easy visualization of the differences in success rates based on these variables.

**Results and Discussion**

Figure 2 displays the top-down flight path of the simulation. The standard rate turn is visible and justifies the placement of the airport in the Monte Carlo.

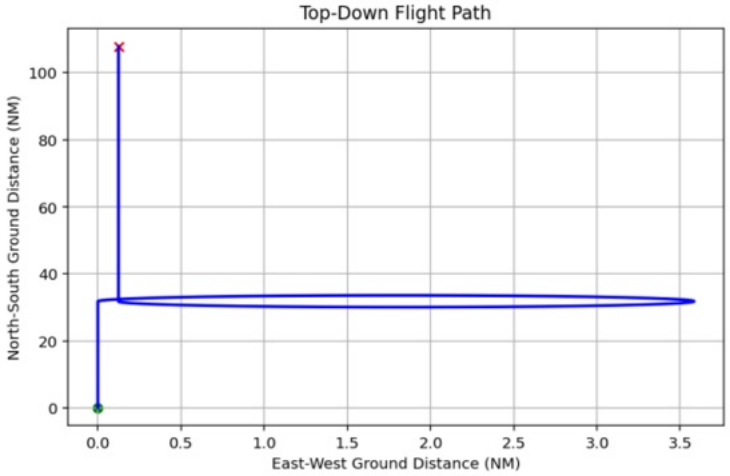



**Figure 2:** Top-down flight path of the simulated airplane; the x and y axes are in nautical miles, and the standard rate turn is displayed. North is the positive direction on the y-axis and south is the negative direction. East is the positive direction on the x-axis and west is the negative direction. (0, 0) is the initial location of the airplane.

Figure 2 displays a representation of the top-down flight path of the CDA simulation. The ellipse displayed is actually a circle but was left as an ellipse in the graph for readability. This is the result of a standard rate turn that happens at the time 300 s and lasts for 120 seconds with a 3-degree bank angle. A standard rate turn is expected to turn the airplane 360 degrees in approximately two minutes, so the final heading of the plane being the same as the initial heading, due north, is a promising indication that the simulation is accurate. Figures 3 and 4 are of the airspeed profile over time and the aerodynamic state transitions. The state transitions display the coefficients of lift and drag and how they change throughout flight.

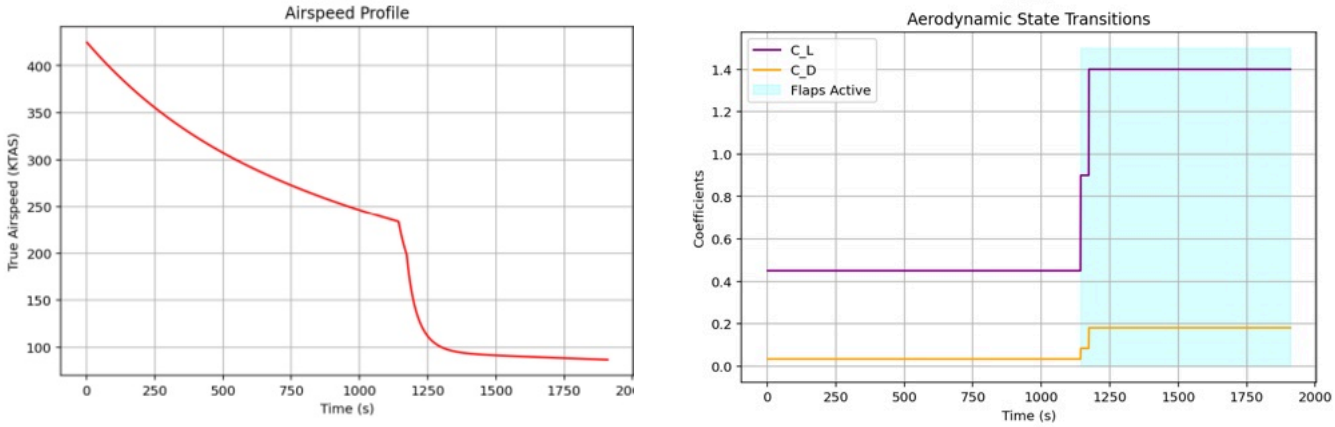



**Figure 3:** The airspeed profile of the simulation with True Airspeed (KTAS) vs Time (s).
**Figure 4:** Aerodynamic state transitions of the simulation. Coefficient of lift (C_L) and coefficient of drag (C_D) are displayed against time. The section highlighted blue is where flaps are deployed.

Figure 3 is the airspeed profile for the airplane. The shape of the curve makes sense because as the airplane slows, it eventually hits the flaps deployment speed at 108.033 knots indicated airspeed (KIAS). The deployment of the flaps significantly alters the aerodynamics of the airplane, increasing both lift and drag and slowing the plane. This results in the observed sharp decline in airspeed in Figure 3. There is a second, less significant decrease in rate of change of speed once the second flap deployment speed is reached at 92.6 KIAS. The axes on the graph are shown in knots true airspeed (KTAS) for simplicity's sake, since showing KIAS would require accounting for altitude in each calculation of speed. Figure 3 is shown alongside Figure 4 because this shows exactly which time flaps were deployed and how the two graphs line up. The two jumps in the coefficients of lift and drag are each flap deployment, and the section highlighted in blue is the area where flaps are deployed. These line up with the changes in the decrease of airspeed shown in Figure 4, demonstrating the relationship between flaps and airspeed.

Figure 5 is the altitude descent profile of the simulation. It shows the continuous descent of the aircraft as it approaches the airport.

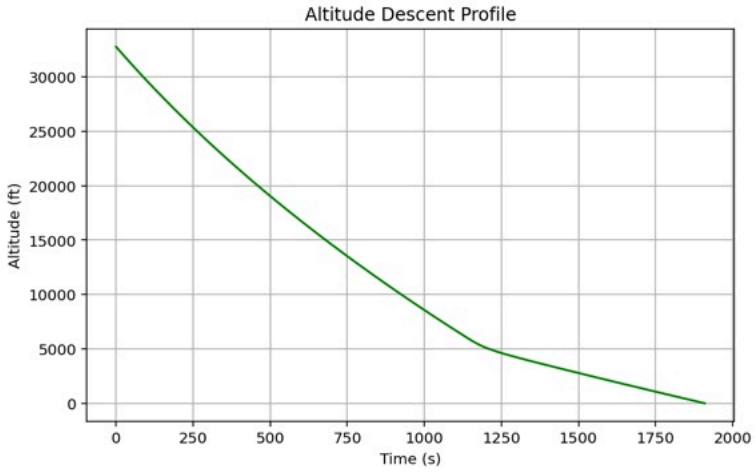
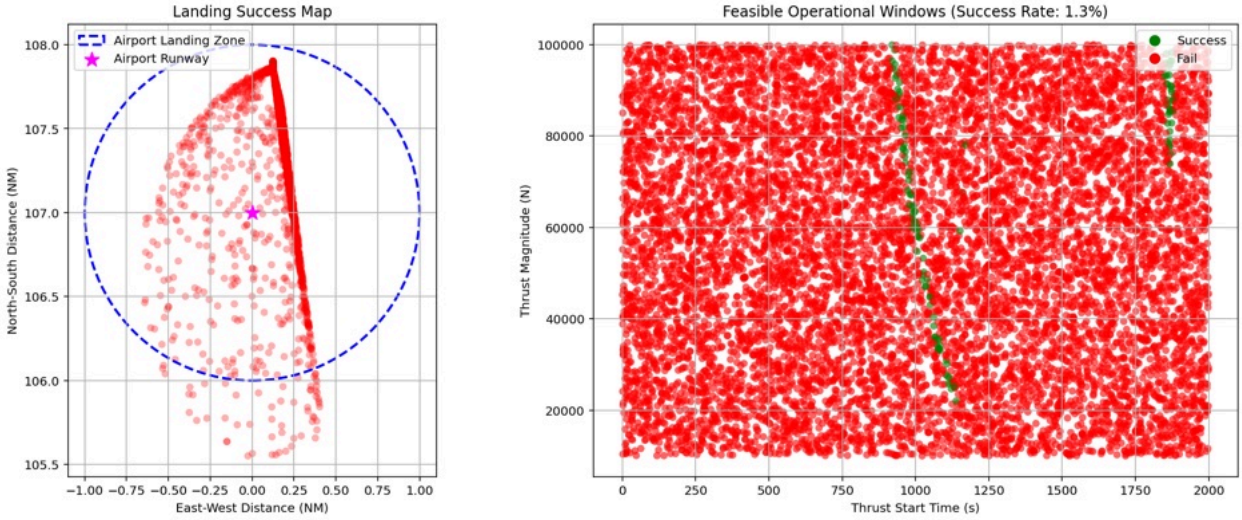



**Figure 5:** The altitude descent profile of the simulation. Altitude is displayed in feet and graphed vs time.

Figure 5 is the altitude descent profile of the CDA simulation. It shows the glide path of the airplane and is relatively consistent with the CDA depicted in Figure 1. However, there is some discrepancy. This inconsistency in the glide path lines up in time with the deployment of flaps. This makes sense because flaps greatly increase lift and the weight of the airplane does not change, outside of fuel burn, so the plane will descend at a slower rate and explains the change in the slope of the glide path.




Figures 6 and 7 are the visual representation of the results for the Monte Carlo simulation.

**Figure 6:** Landing success map of the Monte Carlo simulation. The pink star is the airport and the blue circle is the distance of tolerance for the airport. All axes are in NM. Each dot is a trial. North is the positive direction on the y-axis and south is the negative direction. East is the positive direction on the x-axis and west is the negative direction.
**Figure 7:** Feasible operation window of the Monte Carlo simulation. The trials are displayed against thrust magnitude and thrust start time. If a trial is green, it is successful in both final landing speed and landing location.

Figure 6 shows the landing zone of the Monte Carlo Simulation. It shows that the majority of simulated CDAs ended up within the 1 NM distance of tolerance from the airport. This is promising and shows that nearly every run will at least reach the airport. Variations in the location are caused by the point in the flight that thrust is added. If it is added during the turn, this will not only cause variation in the Y landing location but also in the X landing location since there will be a force vector in both directions. Any input of thrust throughout the flight will affect the Y landing location. The figure to the right of Figure 6 is Figure 7. This shows the thrust input amount graphed against the thrust input time for the Monte Carlo simulation. The green dots are successful trials, and the red are unsuccessful. The success criteria were defined as being within the 15 KTAS landing speed tolerance about the 150 KTAS target landing speed and being within the distance of tolerance for the airport. There was a 1.3% success rate after 10,000 runs. A majority, 91,  of the successful runs lie along the line in Equation 8:

$$y = 5.44 \cdot 10^{7} \cdot e^{-6.81 \cdot 10^{-3} x} \tag{8}$$

These successful trials all fell between 21,442 N and 100,000 N of thrust over the interval 919 to 1141 seconds when they were along this line. This shows a general decrease in the amount of thrust needed as the time into the flight increases; however, there is another grouping of successful trials at around 1850 seconds that all require very high thrust and do not follow the trend. This could be due to low speeds in a simulation later in the flight, requiring large amounts of thrust to reach landing speed or to avoid landing short of the airport.

Figures 8 through 10 represent the histograms of each variable recorded from the successful runs from the Monte Carlo simulation. They are represented to allow for the analysis of the conditions that resulted in successful runs.

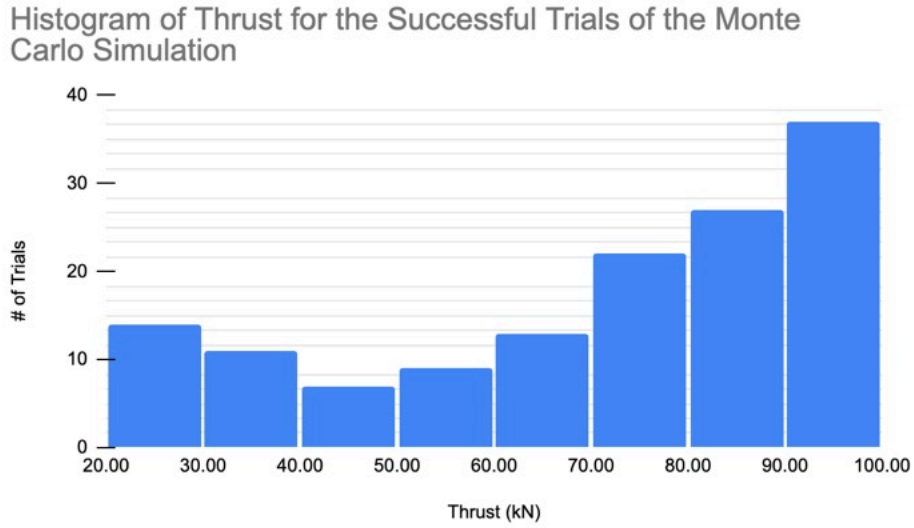



**Figure 8:** The histogram of the thrust in kilonewtons (kN), for all successful trials. The original data were adjusted to kN, the bin sizes changed to 10, and the start and end values adjusted for

easier interpretation of the data. The actual smallest value is 21,442 N or 21.442 kN and the actual largest value is 99,976 N or 99.976 kN.

Figure 8 is a histogram of the thrust imputed for the successful trials. Generally, the majority of successful trials imputed at least 70 kN of thrust, with the greater the thrust, the more successful trials being the common trend. When looking at the graph of Equation 8 as it relates to Figure 7, these higher thrust trials typically fall earlier in the flight than the lower thrust trials. This could be because as the aircraft descends it gains speed, requiring less thrust to reach the appropriate landing speed.

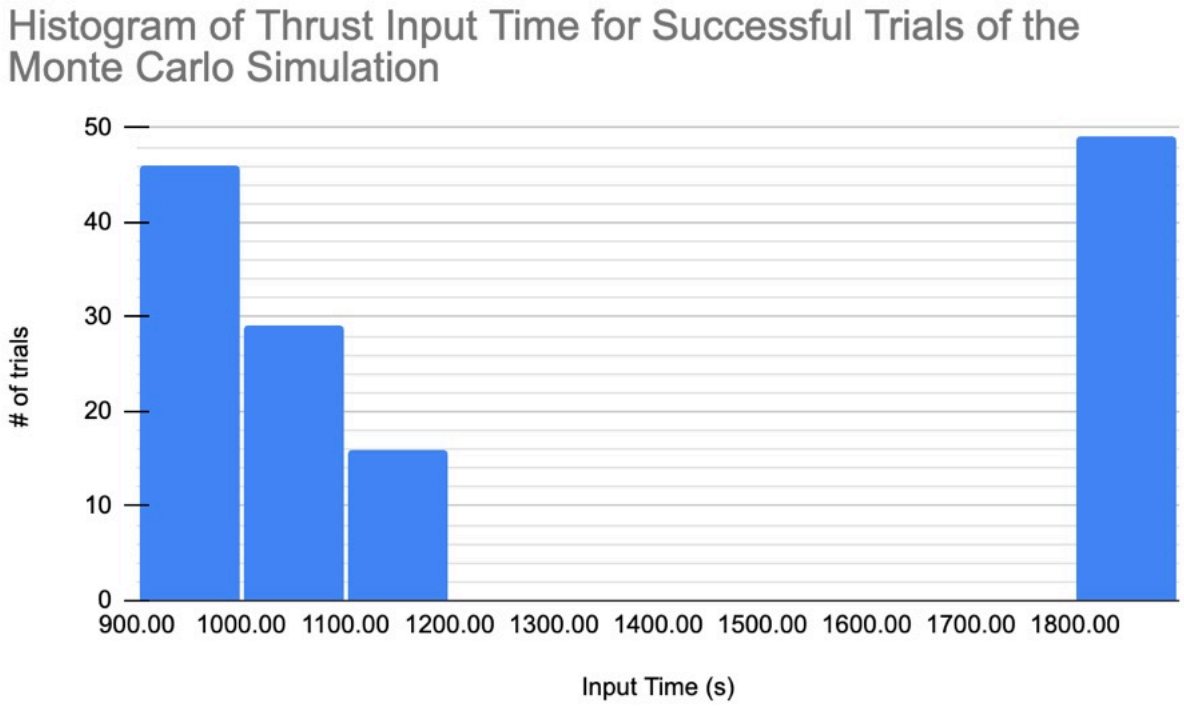



**Figure 9:** The histogram of the thrust input times for successful trials from the monte carlo simulation. The bin sizes were changed to 100 and the start and end values adjusted for easier interpretation of data. The actual smallest value is 919.27 seconds and the actual largest value is 1884.95 seconds.

Figure 9 is the histogram of thrust input time. It shows that the majority of successful trials happened earlier in the flight. However, there are 49 trials that happen between 1,800 seconds and 1,900 seconds. These are the ones that required very high thrust as shown in Figure 7. For the runs that input thrust earlier, there were more successful trials for the earlier times than the ones that followed. Using the line from Equation 8, these were all trials with higher thrust, supporting the trend observed in Figure 8.

The final x location of the airplane does not require a histogram because all values are 0.124 nautical miles (NM) away from the original x location of 0.

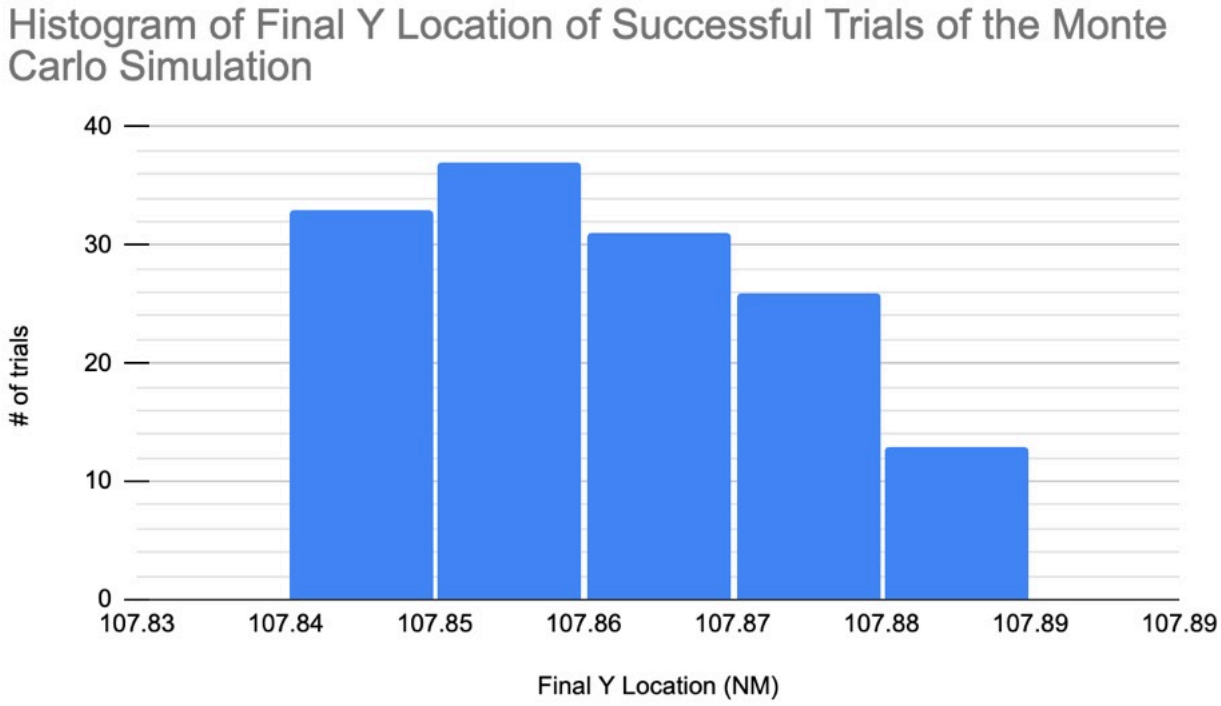



**Figure 10:** The histogram of the final y location of the airplane in successful Monte Carlo trials. The location represents NM away from the original y location at 0. The bin size was adjusted to 0.01 and the bounds of the histogram changed to make it easier to analyze the data. The actual smallest value is 107.84 NM and the actual largest value is 107.89 NM.

Figure 10 shows the final Y location of the successful trials. All of these trials fell within 0.05 NM of each other. This tight grouping of landing locations was about 0.8 NM North of the airport on the Y axis and can be seen in the tight cluster of trials seemingly approaching a point in Figure 6. This could be because of additional added thrust increasing the speed to an appropriate landing speed and causing the airplane to overshoot the airport. The data indicate a potential need to begin a CDA from a greater, though not too significant, distance. The final X location was not reported in a histogram because all X locations were the same. This is due to the standard rate turn and lack of wind putting the airplane back on the original heading for every trial. This resulted in the data seen in Figure 11, which displays the histogram of final distances from the airport. All values fell between 0.85 and 0.89 NM. This distance from the airport is mostly due to overshooting along the Y axis and again indicates a need to begin the descent sooner.

Figures 11 and 12 show the results from the sensitivity study.

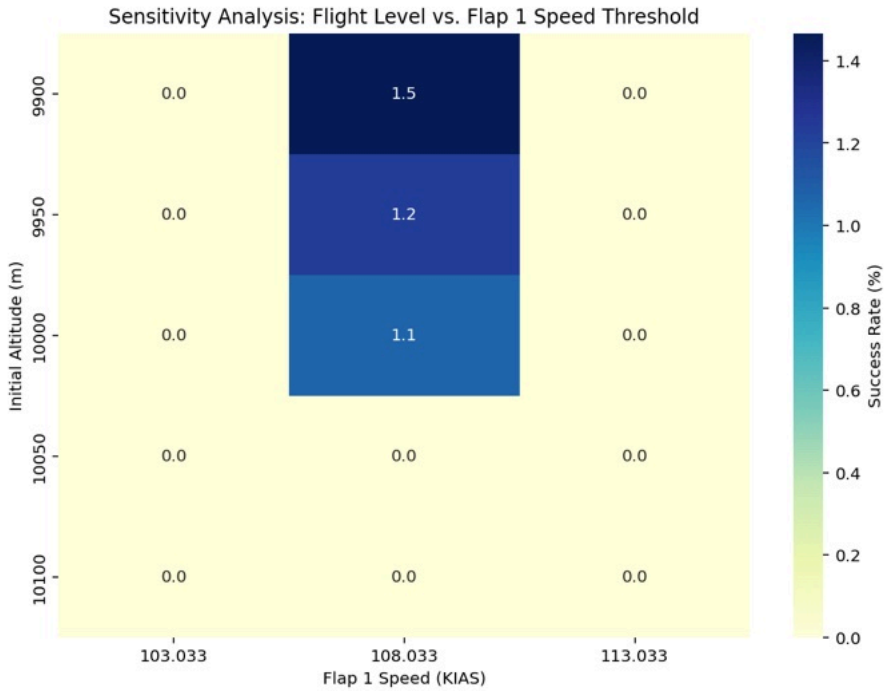



**Figure 11:** Heat map of successful trials based on initial altitude in meters and flap 1 extension speed in knots indicated airspeed (KIAS).

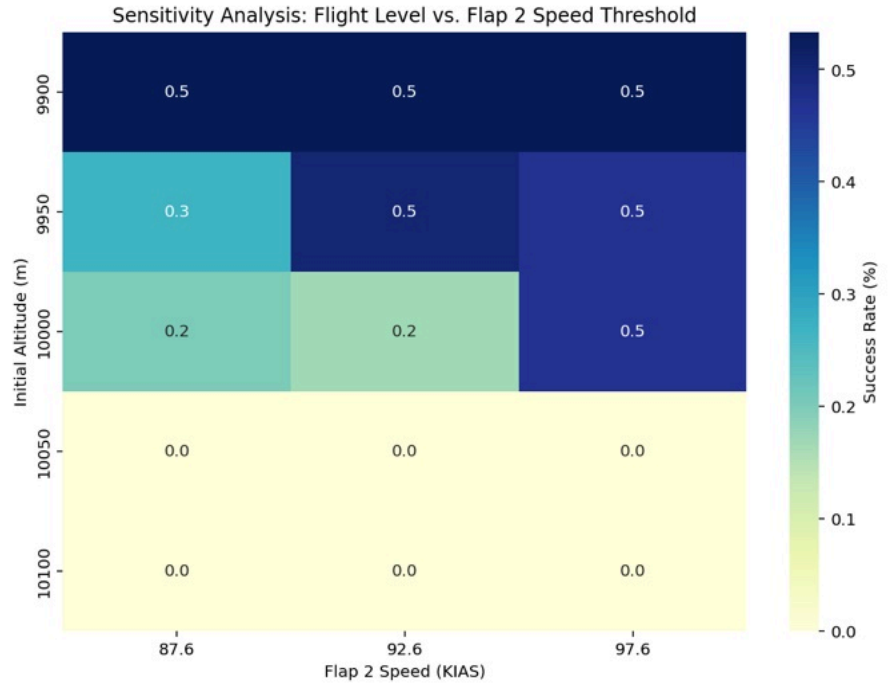



**Figure 12:** Heat map of successful trials based on initial altitude in meters and flap 2 extension speed in KIAS.

Figures 11 and 12 show the results from the sensitivity analysis. Initially, altitude steps of 1000 meters and flap extension speed steps of 10 KIAS were used, however these produced

almost no successful trials, defeating the purpose of using a heat map. Heat maps were chosen because the provide a nice visual representation of how different sets of conditions' success rates compared. Figure 11 shows the sensitivity based on the initial altitude and the flap 1 speed setting. The data show that successful trials only had success when flap 1 was extended at 108.033 KIAS. The success rate increased from 1.1 to 1.5 as initial altitude decreased from 10,000 m to 9,900 m. This indicates that the success of a landing is very sensitive to the speed that the first set of flaps is deployed at and that starting lower increases success rates. This could be due to the trend of the airplane overshooting the airport in the Monte Carlo simulation. Looking at Figure 12 reveals that the airplane had higher success rates at higher flap 2 extension speed and lower altitude. This supports the trend seen in the altitude success rates in Figure 11. It also indicates that the second set of flaps should be added sooner in the flight rather than later. This could be because of a need to increase the coefficient of lift more as the airplane slows down to land.

## Limitations

An important limitation of this study is that the methods were simply Python simulations; the data gathered from these were not from real flights, so they are only estimates. This was chosen for a reason though: it allows for quick estimates and calculations without significant time or monetary investment in research. It is suggested to gather real data from test aircraft performing CDAs in future studies so the data is from a real-world implementation of them rather than a model attempting to accurately model the diverse conditions that aircraft actually experience.

This study is also limited by trial number. Since this simulation was meant to be quick and efficient, the number of trials in the Monte Carlo was limited to 10,000 and the number of runs per cell in the sensitivity study were limited to 1,000. This increases the possibility for outliers to affect the data. However, more trials can be added at the cost of time. A 10,000-trial run in the Monte Carlo took about a minute and a half to run, and a 1,000 run per cell sensitivity study took about six mintues. If time allows, increase the number of trials. By the law of large numbers, this will reduce sampling error and result in more accurate and reliable results [18].

One more limitation of this study is the number of variables included in the simulation. Typical commercial airliners have more than two flap deployments, and the weight of the aircraft will change as the flight goes on and fuel is burned. In further studies, it is recommended to include variables like this for better accuracy. This could be done by adding a condition in the code that tracks fuel burn and subsequently decreases the weight of the aircraft as the flight progresses, and by simply adding more flap deployments.

Also, this study focuses on an airplane similar to a Boeing 737 or an Airbus A320. Looking into other types of aircraft and their behaviors in a CDA could reveal more about the implementation of CDAs in aviation. This can be done by changing the wing reference area, initial weight of the aircraft, and potentially their flap coefficients, depending on what type of flap system they use.

## Conclusion

The importance of decarbonizing aviation is unmistakable, but the way in which we achieve this is less clear. Implementing these CDAs prove to be viable options for this but still

require a lot of development in the strategy behind them before they can truly be implemented. The simulation successfully simulates the CDA of an airplane and can be run in seconds. The Monte Carlo simulation takes fewer than 2 minutes, and the sensitivity study takes . These allow us to quickly change conditions and get results for a CDA study. The initial simulation reveals the forces and factors that go into a CDA and how they look throughout a flight. The Monte Carlo then shows what kind of thrust will be needed to allow for the aircraft to reach the airport at an appropriate speed. The data show that larger amounts of thrust, 70 to 100 kN, about midway through the CDA resulted in the most successful trials. While the ultimate goal of CDA is to remove the emissions associated with thrust in a step-down approach, the reduction of thrust amount and input time in this simulation is a step in the right direction. The sensitivity study reveals important information about how different variables affect the CDA. Starting altitude is very important, if an airplane starts too high, it will need to begin its descent sooner, if it starts too low, the descent should begin later. Also, the flap 1 deployment speed was very important, adding flaps too soon will result in an airplane that cannot reach the airport and adding them too late will have a plane that is fast and lands past the airport. The second notch of flaps was less important for success; however, more successful trials happened when the second notch was deployed at lower speeds. Overall, this study provides important insight into the implementation of CDA as an approach to decarbonizing aviation.

**References**

1. Timperley, Jocelyn. “Should We Give up Flying for the Sake of the Climate?” *BBC News*, BBC, 24 Feb. 2022, www.bbc.com/future/article/20200218-climate-change-how-to-cut-your-carbon-emissions-when-flying.
2. *Aviation - IEA*, www.iea.org/energy-system/transport/aviation. Accessed 21 Aug. 2026.
3. Dabin Xue, Sen Du, Bing Wang, Wen-Long Shang, Nicolò Avogadro, Washington Yotto Ochieng, Low-carbon benefits of aircraft adopting continuous descent operations, Applied Energy, Volume 383, 2025, 125390, ISSN 0306-2619, https://doi.org/10.1016/j.apenergy.2025.125390.
4. *FAA implements more efficient descent procedures to reduce fuel burn, emissions*. FAA Implements More Efficient Descent Procedures to Reduce Fuel Burn, Emissions | Federal Aviation Administration. (n.d.). https://www.faa.gov/newsroom/faa-implements-more-efficient-descent-procedures-reduce-fuel-burn-emissions
5. Alharbi, Emad A., et al. “Analytical Model for Enhancing the Adoptability of Continuous Descent Approach at Airports.” *MDPI*, MDPI, 30 Jan. 2022, www.mdpi.com/2076-3417/12/3/1506.
6. Ibm. “What Is Monte Carlo Simulation?” *IBM*, 17 Dec. 2024, www.ibm.com/think/topics/monte-carlo-simulation.
7. “Airbus A320 Specifications Overview.” *Scribd*, Scribd, www.scribd.com/document/463906750/Aircraft-Family-A320-Specifications-pdf. Accessed 4 Sept. 2026.
8. “International Standard Atmosphere.” *Aerodynamics for Students*, University of Cambridge,

www-mdp.eng.cam.ac.uk/web/library/enginfo/aerothermal_dvd_only/aero/atmos/atmtab.html. Accessed 4 Sept. 2026.
9. “Earth Fact Sheet.” *AEROSPACE and ASTRONAUTICS*, 21 Oct. 2014, aerospaceastro.com/earth/earth-fact-sheet/.
10. “MAC Weight and Balance Manual.” *Scribd*, Scribd, www.scribd.com/document/891198620/MAC-wbm-59-61-1. Accessed 4 Sept. 2026.
11. “8.7: Scale Height in an Isothermal Atmosphere.” *Physics LibreTexts*, Libretexts, 18 Jan. 2026, phys.libretexts.org/Bookshelves/Thermodynamics_and_Statistical_Mechanics/Heat_and_Thermodynamics_(Tatum)/08%3A_Heat_Capacity_and_the_Expansion_of_Gases/8.07%3A_Scale_Height_in_an_Isothermal_Atmosphere.
12. “Reynolds Number.” *NASA*, NASA, www.grc.nasa.gov/www/k-12/airplane/reynolds.html. Accessed 21 Aug. 2026.
13. *Chapter 7.5 Turbulent Boundary Layer*, stern.lab.uiowa.edu/sites/stern.lab.uiowa.edu/files/2025-01/Chapter%207.5%20Turbulent%20Boundary%20Layer.pdf. Accessed 22 Aug. 2026.
14. Khaled, Md Faiaz, and Aly Mousaad Aly. “Assessing Aerodynamic Loads on Low-Rise Buildings Considering Reynolds Number and Turbulence Effects: A Review - Advances in Aerodynamics.” *SpringerLink*, Springer Nature Singapore, 1 June 2022, link.springer.com/article/10.1186/s42774-022-00114-0.
15. “Rate of Turn.” *Rate of Turn | SKYbrary Aviation Safety*, skybrary.aero/articles/rate-turn. Accessed 31 Aug. 2026.
16. Martin, Swayne. “The Aerodynamics of a Turn.” *Online Flight Training Courses and CFI Tools*, 29 Sept. 2015, www.boldmethod.com/learn-to-fly/aerodynamics/the-aerodynamics-of-a-turn-in-an-airplane/.
17. “The Use of the Monte Carlo Method in Sensitivity Analysis and Its Advantages.” *PCB Design & Analysis Resources*, resources.pcb.cadence.com/blog/2020-the-use-of-the-monte-carlo-method-in-sensitivity-analysis-and-its-advantages. Accessed 1 Sept. 2026.
18. Mascha, Edward J, and Thomas R Vetter. “Significance, Errors, Power, and Sample Size: The Blocking and Tackling of Statistics.” *National Center for Biotechnology Information*, U.S. National Library of Medicine, Feb. 2018, pubmed.ncbi.nlm.nih.gov/29346210/.

## Appendix A: Supplemental Histograms and Figures

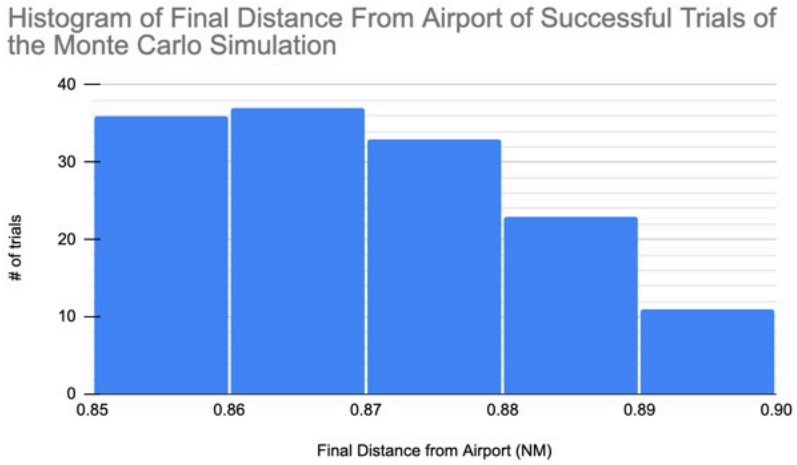



**Figure A1:** The histogram of the final distance from the airport in NM of the successful trials in the Monte Carlo simulation. The bin sizes and range were not adjusted from the original exported from Google Sheets. However, the actual smallest value was 0.853 NM and the actual largest value was 0.896 NM.

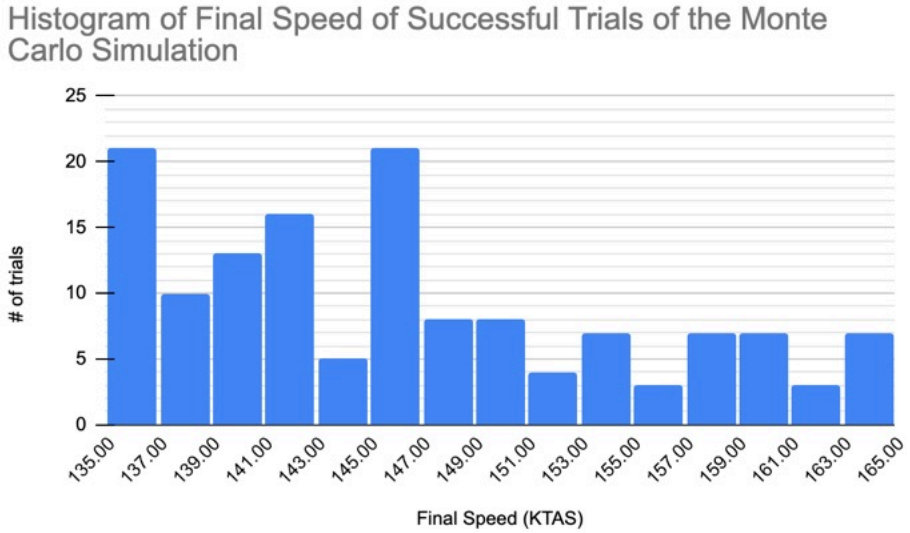



**Figure A2:** The histogram of the final speed of successful Monte Carlo trials in Knots True Airspeed (KTAS). The bin size was changed to 2 and the range was adjusted to make the data easier to interpret. The actual smallest value was 135.12 KTAS and the actual largest value was 164.25 KTAS.

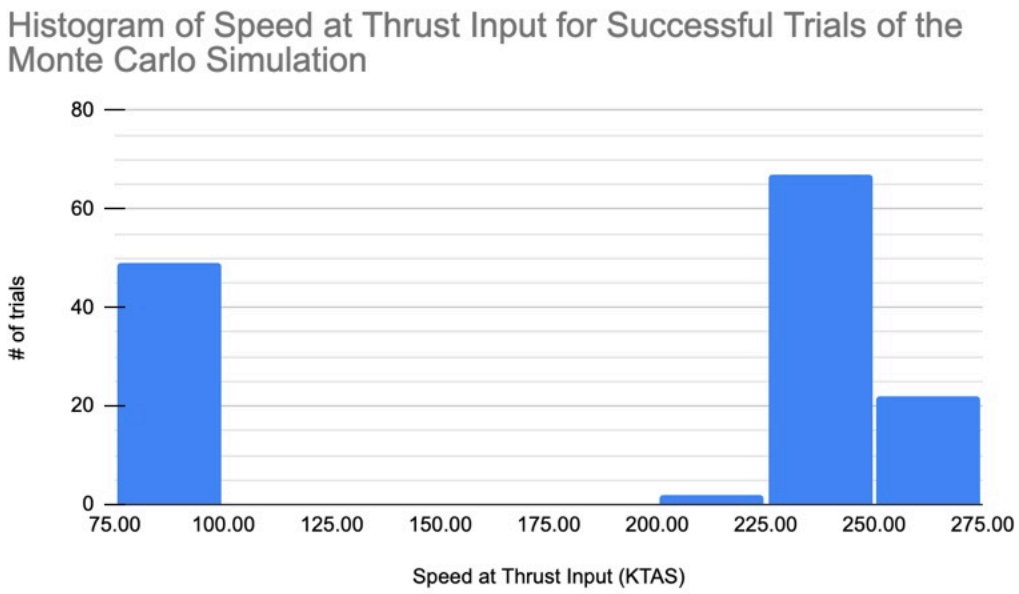



**Figure A3:** Histogram of the speed at the input of thrust in KTAS for successful Monte Carlo trials. The bin size was adjusted to 25 and the range adjusted to make the data easier to understand. The actual minimum was 86.89 KTAS and the actual maximum was 254 KTAS.

## Appendix B: Monte Carlo Code

```
import numpy as np
import matplotlib.pyplot as plt
import random
import pandas as pd

# CONSTANTS
g = 9.81
S = 122.6  # wing reference area
m = 60000.0  # initial aircraft mass (kg)
rho_0 = 1.225  # sea-level air density (kg/m^3)
H_scale = 8500.0  # scale height for atmospheric density exponential model (m)
L = 4.2  # mean aerodynamic chord (m)
mu = 1.789e-5  # dynamic viscosity of air (kg/(m*s), baseline sea-level value)

# Winds
W_x = 0.0  # East/West wind
W_y = 0.0  # North/South wind
W_h = 0.0  # - is a downdraft
# x is east west, y is north south

dt = 1.0
max_time = 2000

# Airport Location (NM)
Airport_x = 0.0
Airport_y = 107.0
dist_tolerance = 1

# Target Landing Speed (KTA)
Target_speed = 150.0
Speed_tolerance = 15.0


def run_simulation(thrust_val, thrust_start, thrust_duration=30.0):
    # Initial States
    t = 0.0
    h = 10000.0
    V = 218.639  # true airspeed (m/s)
    gamma = np.radians(-2.6)  # glide slope(radians)
    x = 0.0  # initial X position (m)
    y = 0.0  # initial Y position (m)
    heading = np.radians(90)  # 90 degrees = north

    speed_thrust_start = None

    while h > 0 and t < max_time:
        rho = rho_0 * np.exp(-h / H_scale)

        V_re = V - W_x

        Re = (rho * V_re * L) / mu

        IAS = V * np.sqrt(rho / rho_0)

        if IAS < 92.6:
```

```
            flaps_state = 2
            CL = 1.4
            CD0 = 0.075
            K = 0.055
        elif IAS < 108.033:
            flaps_state = 1
            CL = 0.9
            CD0 = 0.045
            K = 0.050
        else:
            flaps_state = 0
            CL = 0.45
            CD0 = 0.025
            K = 0.045

        re_correction = 1.0 - 0.02 * np.log10(Re / 1e6)
        CD = (CD0 * re_correction) + K * (CL ** 2)

        if thrust_start <= t < (thrust_start + thrust_duration):
            thrust = thrust_val
            if speed_thrust_start is None:
                speed_thrust_start = V * 1.94383
        else:
            thrust = 0.0

        Lift = 0.5 * rho * (V ** 2) * S * CL
        Drag = 0.5 * rho * (V ** 2) * S * CD

        dV_dt = (thrust - Drag) / m - g * np.sin(gamma)
        dh_dt = V * np.sin(gamma) + W_h

        if 300 <= t < 420:
            turn_rate = np.radians(-3.0)  # negative for right turn using radians
        else:
            turn_rate = 0.0

        heading += turn_rate * dt

        V_ground_no_wind = V * np.cos(gamma)
        dx_dt = V_ground_no_wind * np.cos(heading) + W_x
        dy_dt = V_ground_no_wind * np.sin(heading) + W_y

        V += dV_dt * dt
        h += dh_dt * dt
        x += dx_dt * dt
        y += dy_dt * dt
        t += dt

    final_x_nm = x / 1852.0
    final_y_nm = y / 1852.0
    final_speed_ktas = V * 1.94384

    distance_to_airport = np.sqrt((final_x_nm - Airport_x) ** 2 + (final_y_nm -
Airport_y) ** 2)
    speed_error = abs(final_speed_ktas - Target_speed)
```

```
    met_distance = distance_to_airport <= dist_tolerance
    met_speed = speed_error <= Speed_tolerance
    success = met_distance and met_speed

    return {
        "success": success,
        "met_distance": met_distance,
        "met_speed": met_speed,
        "final_x": final_x_nm,
        "final_y": final_y_nm,
        "final_speed": final_speed_ktas,
        "distance_to_airport": distance_to_airport,
        "speed_error": speed_error,
        "speed_thrust_start": speed_thrust_start
    }


num_runs = 10000

inputs_thrust = []
inputs_start_time = []
results_success = []
results_x = []
results_y = []
successful_trials = []
speed_log = []

success_count = 0

for run in range(num_runs):
    sim_thrust = random.uniform(10000, 100000)  # varies thrust between 10,000 and
100,000 N
    sim_start_time = random.uniform(0, 2000)

    res = run_simulation(sim_thrust, sim_start_time)

    inputs_thrust.append(sim_thrust)
    inputs_start_time.append(sim_start_time)
    results_success.append(res["success"])
    results_x.append(res["final_x"])
    results_y.append(res["final_y"])

    if res["success"]:
        success_count += 1
        successful_trials.append({"Run ID": run,
                                  "Thrust": sim_thrust,
                                  "Input Time": sim_start_time,
                                  "Final X": res["final_x"],
                                  "Final Y": res["final_y"],
                                  "Distance From Airport": res["distance_to_airport"],
                                  "Final Speed": res["final_speed"],
                                  "Speed at Thrust Start": res["speed_thrust_start"]
                                  })

df = pd.DataFrame(successful_trials)
df.to_csv("Total Successful Trials.csv", index=False)
```

```
print(f"Total Successful Runs: {len(successful_trials)}")

success_rate = (success_count / num_runs) * 100
print("Monte Carlo Simulation Completed.")
print(f"Overall Success Rate: {success_rate:0.2f}% ({success_count} / {num_runs}
runs)")

# PLOTS
figs, axs = plt.subplots(1, 2, figsize=(15, 6))

scatter = axs[0].scatter(results_x, results_y, c=['green' if s else 'red' for s in
results_success], alpha=0.3,
                         edgecolors='none')
airport_circle = plt.Circle((Airport_x, Airport_y), dist_tolerance, color='blue',
fill=False, linestyle='--',
                            linewidth=2, label='Airport Landing Zone')
axs[0].add_patch(airport_circle)
axs[0].scatter(Airport_x, Airport_y, color='magenta', marker='*', s=150, zorder=5,
label='Airport Runway')
axs[0].set_xlabel('East-West Distance (NM)')
axs[0].set_ylabel('North-South Distance (NM)')
axs[0].set_title('Landing Success Map')
axs[0].grid(True)
axs[0].legend()
axs[0].set_aspect('equal')

scatter = axs[1].scatter(inputs_start_time, inputs_thrust, c=['green' if s else 'red'
for s in results_success],
                         alpha=0.6, edgecolors='none')
axs[1].set_xlabel('Thrust Start Time (s)')
axs[1].set_ylabel('Thrust Magnitude (N)')
axs[1].set_title(f'Feasible Operational Windows (Success Rate: {success_rate:.1f}%)')
axs[1].grid(True)

from matplotlib.lines import Line2D

legend_elements = [Line2D([0], [0], marker='o', color='white', label='Success',
markerfacecolor='green', markersize=10),
                   Line2D([0], [0], marker='o', color='white', label='Fail',
markerfacecolor='red', markersize=10)]
axs[1].legend(handles=legend_elements)

plt.tight_layout()
plt.show()
```