ANALYSIS OF AIRCRAFT CONTROL INPUT TO PRODUCE 3D CONTINUOUS DESCENT APPROACH (CDA) TRAJECTORY WITH PY-FME

Ronald Andryando, Neno Ruseno
Aviation Engineering Department, Faculty of Engineering, International University Liaison Indonesia
BSD City, South Tangerang, Banten 15345, Indonesia
Corresponding Author: andryandoronald@gmail.com, neno.ruseno@iuli.ac.id

Abstract *– One of proposed solutions to decrease the fuel consumption of a flight is by optimizing descent trajectory using Continuous Descent Approach (CDA). The focus of this research is to analyze the aircraft inputs used in CDA with several types of trajectories (straight and turning) in 3D (Three Dimensional). The CDA concept used is based on Time and Energy Managed Operation concept where the use of idle thrust is the key point. The research will also analyze the fuel consumption of aircraft in CDA trajectory and compare it with conventional descent trajectory. The methodology on this research is simulation using a Python programming module called Py-FME with Cessna-172 aircraft data. The result concluded that thrust and elevator input have significant effect on aircraft controls to achieved CDA. The research also found that CDA could reduce the fuel consumption by 67.6%.*

Keywords*: 3D Trajectory, Continuous Descent Approach, Fuel consumption analysis, Descent Trajectory, PY-FME, Cessna 172*

I. INTRODUCTION

Air transportation is one of the important transportation modes nowadays. However, fuel cost of the flight is significant. There are some methods to reduce the fuel consumption including using a new aircraft that have a good efficiency engine or trajectory efficiency such as Continuous Descent Approach (CDA). It is an optimized landing trajectory that required minimum thrust idle around 0.5 thrust. The simulation results shown that 1853 kg fuel would have been saved when a B747-200 on 25 March 2006 had flown a CDA procedure. (Jin, Cao, & Sun, 2013)

Unlike the normal descending aircraft, the trajectory of Continuous Descent Approach has continuous negative gradient, while the normal descending aircraft trajectory is like a stair step (Prats, X., Bendris, B., Dalmau, R., Montolio, J., Labs, B. D., Lenz, H., & Kohrs, R., 2016).

The aim of this research is to generate a 3D Continuous Descent Approach trajectory and analyze which control inputs of the aircraft has significant effect in achieving the Continuous Descent Approach trajectory. It will also calculate the fuel saving of CDA compared with conventional descent trajectory.

This publication will consist of research methodology in section 2, result and analysis and section 3 and conclusion and recommendation in chapter 4.

II. METHODOLOGY

This section consists of concept of navigation and flight control, Continuous Descent Operation concept, Py-FME module and the scenario of the simulation.

2.1 Basic Concept of Navigation and Control

The basic concept of air navigation was the same as in the general navigation, where controlling vehicle movement from one point to the other point. However, there are some differences between air navigation and general navigation. The aircraft relatively travel at high speed, high altitude and without a clear visualization. It has a less time to calculate its position during en-route.

There are several techniques that can be used for the air navigation. However, it depends on the equipment available in the aircraft. Whether the aircraft flying under instrument flight rules (IFR) or visual flight rules (VFR).

The navigation system of an aircraft or air navigation was based on dead reckoning concept. Dead reckoning is the process of calculating current position by using its previous position. Though dead reckoning still being used, there is a new concept that implement dead reckoning, Inertia Navigation System (INS). The basic philosophy is to begin with a knowledge of initial position, keep track of speed and direction, and thus be able to determine position continually as time progresses. The INS based on the newton's second law:

$$\vec{F} = m \cdot \vec{a} \qquad (2.1)$$

Where: $\vec{F}$ is the force vector, m is the mass, and $\vec{a}$ is the acceleration vector. This equation can be integrated in order to determine the position.

Furthermore, in navigation, the aircraft needs to be controlled. Mainly, there are two control system in aircraft: open loop and close loop control system. In the open loop control system there is no feedback, while in close loop system there is a feedback. Feedback is when the output is sent back into the input to correct the system based upon the desired output as shown in Figure 2.1. While open loop does not have feedback, so whatever the result is it will be the output. It is shown in Figure 2.2.

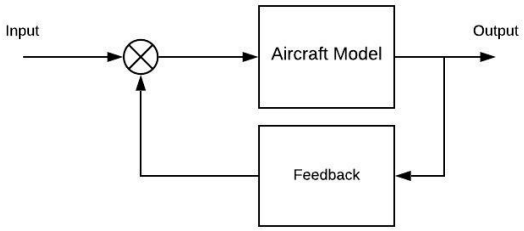


Figure 2.1: Close loop control system

Figure 2.2: Open loop control system

For the transfer function of close loop system is:

$$C(s) = \frac{G(s)}{1 + G(s)H(s)} R(s) \qquad (2.2)$$

Where $G(s)$ is the aircraft model, $H(s)$ feedback, $C(s)$ is output, and $R(s)$ is input. The input parameters $R(s)$: Delta elevator, delta aileron, delta rudder, and delta thrust. Output parameters$C(s)$: Height, x earth, y earth, phi, theta, and psi. Feedback parameters $H(s)$: phi, psi, theta. (Ogata, 2010)

Reference frame is one of important things in navigation system. It allows to know how to takes quantities expressed in one frame and convert them into another. Since velocity was typically expressed in a local reference frame, and position was frequently expressed in reference frame fixed to earth (latitude and longitude).

There are some types of reference frame, true inertial frame, earth-centered inertial frame (i-Frame), earth-centered earth-fixed frame (e-Frame), navigation frame, body frame, and

sensor frames as shown in figure 3.3. (Nebylov & Watson, 2016)

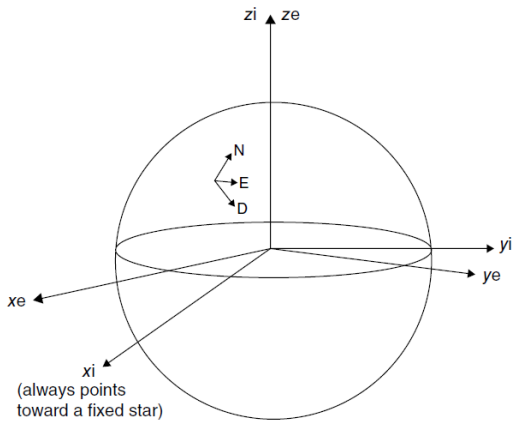


Figure 2.3: Reference frame (I-frame, e-frame, and n-frame)

True Inertial frame is more known as fixed star. This frame does not accelerate with respect to the star. Earth centered Inertial frame (I-frame) is centered in earth but not rotating with earth, it remains fixed respect to star. Earth centered earth fixed frame (e-frame) is also centered in earth but it rotates along with earth. Navigation frame (n-frame) is centered in vehicles, the most common things on n-frame is the x-axis is point to north and y-axis point to east, z-axis point down. Sometimes n-frame called NED frame.

2.2 Concept of CDA

Continuous Descent Approach is an optimized method where the descent profile is a smooth trajectory with constant descent angle instead of the conventional descent. Continuous descent approach designed to reduce fuel consumption and noise. This method uses minimal thrust to be able save the fuel consumption.

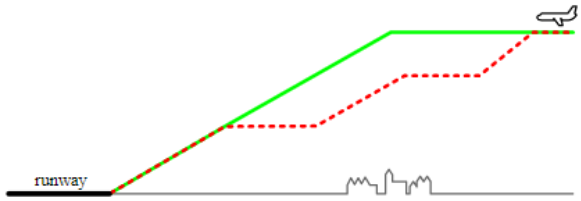


Figure 2.4: Descent Trajectory: CDA trajectory (green line), conventional trajectory (red line)

The Continuous Descent Approach trajectory is shown in Figure 2.4. The aircraft in CDA trajectory will descent continuously from the final approach point and go straight into the runway, while the conventional trajectory will descent into certain levels of constant altitude. The conventional descent could produce more noise that impact the object under the aircraft (Verhoeven, R., Bussink, F. J., Prats, X., & Dalmau, R., 2013) and (Lim, Y., Sabatini, R., & Gardi, A., 2018).

2.3 Py-FME Simulation Module

PY-FME is a program to simulate the aircraft motion. It stands for Python Flight Mechanics Engine. This program represents aircraft dynamics, its controls, navigation and also surrounding factors. It is available in github website with the link: https://github.com/AeroPython/PyFME.

The simulation will be done in Python programming language using the Py-FME module. This module has 4 major classes: aircraft, environment, models, and utils. Aircraft model consist of aircraft class, Cessna-172, and Cessna-310. However, the Cessna-310 is not complete (Saez, Aqreed, & Rodriguez, n.d.)

Aircraft class consist of mass and inertia, geometry, controls, coefficients, thrust, forces and moments, velocities, and attitude angles of aircraft. Environment class consist of atmosphere, environment, gravity, and wind. Models class consists of state, constants, dynamic system and Euler flat earth. Utils class consist of altimetry, anemometry, coordinates, input generator, and trimmer.

The output from PY-FME simulation are:

- Position: Xearth, Yearth, height
- Attitutde: psi, theta, phi
- Speed: Vnorth, Veast, Vdown
- Rotational rate: p, q, r
- Aerodynamic parameters: alpha, beta, TAS
- Forces: Fx, Fy, Fz

- Moments: Mx, My, Mz
- Control Inputs: elevator, aileron, rudder, thrust

In the original of Py-FME, there is no feedback controls implemented (open-loop system). For our research purpose, the author adding the feedback controls, therefore the system will be a close loop. For example, in a straight CDA, the pitch angle is used as the feedback with the threshold is 3 degrees. When the pitch angle is bigger than 3 degrees, then the system will lower the thrust input to maintain the pitch angle below the threshold.

The closed loop system is implemented in the function of _get_current_controls. The modified code for controlling pitch angle (theta) using thrust is as follow:

```
from pyfme.utils.input_generator import Doublet, Constant, Step, Ramp

 #Feedback control for Theta using Thrust
if state.attitude.theta > -0.050: # descent angle more than -3 deg
    self.controls['delta_t'] = Step(t_init=0, T=150, A=-0.55, offset=0.57799667)
else:
    self.controls['delta_t'] = Step(t_init=0, T=150, A=-0.5, offset=0.57799667)

c = {c_name: c_fun(time) for c_name, c_fun in self.controls.items()}
return c
```

2.4 Simulation Scenarios

There are 9 scenarios of this simulation: scenario 1 until scenario 6 is the straight CDO, scenario 7 is 60 degree turn CDO, scenario 8 is 30 degree turn CDO, and scenario 9 is the straight conventional descent.

Scenario 1 is straight CDA with thrust input and thrust feedback, scenario 2 is straight CDA with thrust input without feedback, scenario 3 is straight CDA with elevator input and elevator feedback, scenario 4 is straight CDA with elevator input without feedback, scenario 5 is straight CDA with thrust and elevator input and thrust feedback, scenario 6 is straight CDA with thrust and elevator input without feedback.

Scenario 7 is the 60 degree turn landing CDA, scenario 8 is the 30 degree turn landing CDA. Scenario 7 and 8 are used to evaluate control inputs in turning CDA. While scenario 9 is the conventional landing trajectory that will be used for comparison between CDA and non CDA trajectories. All the scenarios are shown in Figure 2.5.

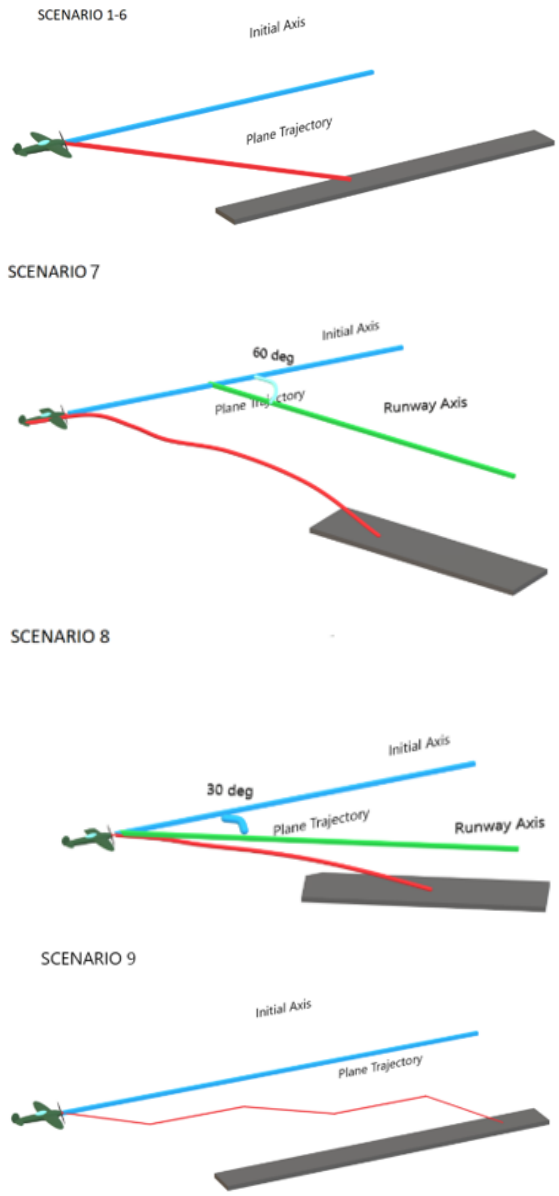


Figure 2.5: All scenarios of simulation

For this simulation aircraft used is Cessna 172. For all scenarios the starting altitude is 610 meters, atmosphere condition is ISA1976 (Troposphere) and no external disturbance. The duration of simulation is 160 seconds and the true airspeed used is 45 m/s.

III. RESULT AND ANALYSIS

The result of the simulation is varied, due to the combination of the input and the objective of each scenario. For each type of input gives unique result to the movement of the aircraft. Scenario 1 until scenario 6 is the straight CDA, Scenario 7 is 60 degree turning CDA, Scenario 8 is 30 degree turning CDA, and Scenario 9 is straight conventional trajectory.

In this section the result from scenario 1 (straight CDA) and scenario 7 (turning CDA) will be presented in detail. The result from other scenarios are shown in Appendix. Furthermore, the aggregate of all result will be analyzed in this section.

3.1 Straight CDA Result

In this scenario the profile will be straight Continuous Descent Approach, using Thrust Input with Thrust Feedback. This feedback is to prevent the aircraft to descent more than -3 degree with lowering the input of thrust.

The input changes only happen in thrust at the start of the simulation as shown in Figure 3.1. The duration of the input is 160 seconds with amplitude of the input is -0.5 radian, and the starting point of the thrust is from default of the aircraft.

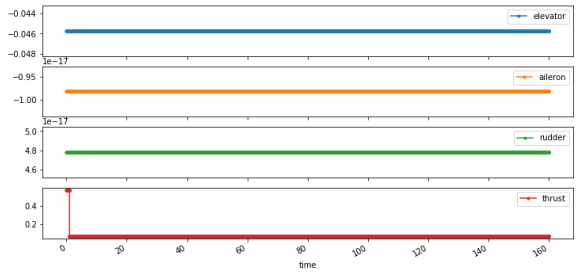


Figure 3.1: Input of the straight CDA trajectory

The output of the trajectory in 3D plot is shown in Figure 3.2. It is a straight CDA. However, at the beginning of the trajectory there are some oscillation in term of altitude, due to a response to the changes in thrust power.

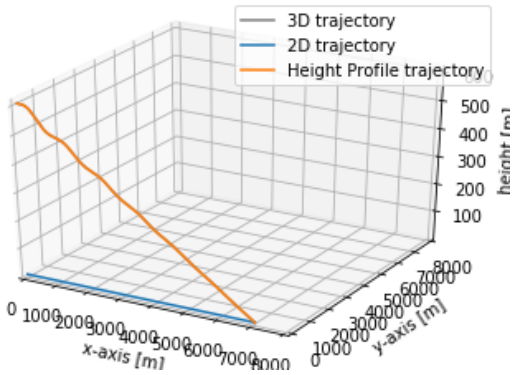


Figure 3.2: 3D trajectory of straight CDA

3.2 Turning CDA Result

In this scenario the trajectory of CDA will turn in 60 degrees. The inputs are elevator, aileron, rudder, and thrust. It is shown in Figure 3.3. The input changes occur at elevator, aileron and rudder to control the turning trajectory.

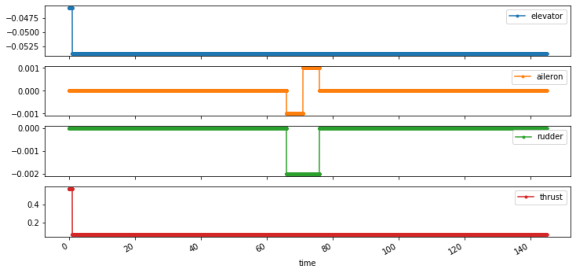


Figure 3.3: Input of the turning CDA trajectory

The result of 3D trajectory is shown in Figure 3.4. It able to turn in around 60 degrees with a small oscillation in vertical profile at the beginning.

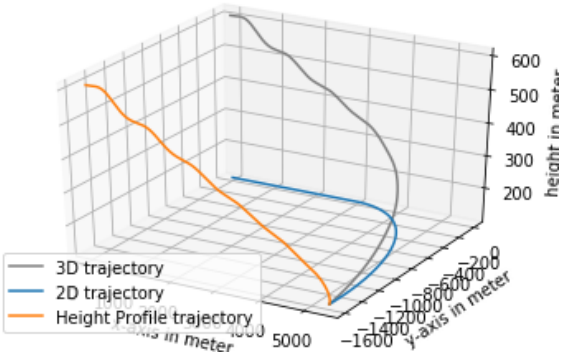


Figure 3.4: 3D trajectory of turning CDA

3.3 Analysis

Since CDA requires to have a constant descent rate, we will analyze the trajectory based on that parameter for each of the scenario. We use the box plot to represent its statistical parameters and compare each of the simulation result.

The result of scenario 1 to 6 which represent a straight CDA is plotted in Figure 3.5.

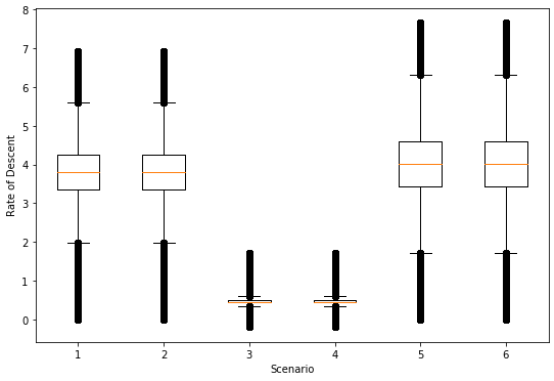


Figure 3.5: Boxplot of the descent rate scenario 1 until scenario 6

These boxplots have a normal distribution, not skewed left and not skewed right. It means median is equal to mean. The plot shows that scenario 3 and scenario 4 are more stable, while scenario 1, 2, 5, and 6 are fluctuating. It is shown by the data of scenario 1, 2, 5 and 6 are spreading, while the data of scenario 3 and 4 not spreading.

Since, scenario 5 and 6 have the biggest median of descent rate, it means scenario 5 and 6 is the most suitable for to be used for CDA. It happens because the input of scenario 5 and 6 is the combination of thrust and elevator.

For the turning trajectory, the descent rate result is shown in Figure 3.6.

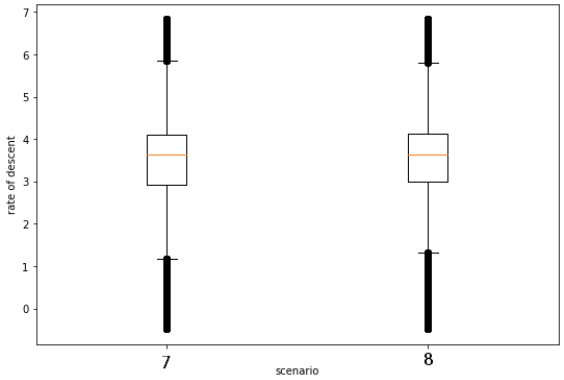


Figure 3.6: Boxplot of the descent rate of scenario 7 (turning 60 deg) and scenario 8 (turning 30 deg)

Boxplot of scenario 7 is a little bit skewed up, and boxplot of scenario 8 is normal distribution. The mean of scenario 7 is bigger than the median. Scenario 7 and 8 are fluctuating as shown by the spreading data.

The median of turning scenarios (60 and 30 degree) almost the same because of the same thrust and elevator input. The difference of those scenario is the effect of aileron and rudder input magnitude.

Scenario 9 is the conventional descent trajectory. it is used for the fuel consumption comparison. The fuel burn rate of the Cessna172 is 32.175 Liters per hour for 100% thrust, and 0.0089375 per seconds for 100% thrust. (Buscombe, n.d.) Total fuel usage can be calculated by:

$$FU = \sum Br.T \qquad (3.1)$$

Where FU is the Total Fuel Usage, Br is the Burn Rate, and T is Time.

Here, the total fuel usage will be compared between CDA trajectory (scenario 1-6) and conventional descent trajectory (scenario 9). The result of calculation is shown in Table 3.1. The average of CDA trajectory requires 0.1051 liter, while the conventional trajectory consumes 0.3248 liter. Thus, the CDA can save fuel by 0.2197 liter or 67.6% compare to the conventional descent trajectory.

Table 3.1 Total Fuel Usage each trajectory

Scenario	Time (second)	Thrust (%)	Total Fuel Usage (Liter)
1	1	57	0.1084
	159	7.3	
2	1	57	0.1084
	159	7.3	
5	1	57	0.1019
	159	7.3	
6	1	57	0.1019
	159	7.3	
CDA average	-	-	*0.1051*
9*	100	7.7	0.3248
	60	57.7	

*Non CDA Trajectory

IV. Conclusion and Recommendation

Base on the finding in previous section, we can conclude that:

- Every input (elevator, aileron, rudder and thrust) affect the CDA trajectory. However, thrust has the biggest effect on the trajectory.
- Control inputs for straight CDA are Elevator and Thrust. Combination input of elevator and thrust make a better CDA trajectory.
- Control inputs for turning CDA are elevator, aileron, rudder and thrust. Similar to the straight CDA, the combination input of elevator and thrust affect the vertical profile of the trajectory. While the turning trajectory affect significantly by the magnitude of the rudder and aileron input.
- By using CDA trajectory for straight scenario, the fuel can be saved by 67.6%.

In addition, we recommend:

- Adding other aircraft models to Py-FME, thus we can compare the result for other aircraft type.
- Using other feedback control such as PID to get better handling quality of the CDA

REFERENCES

Buscombe, J. (n.d.). *:Cessna 172 - The Most Popular Light Aircraft in History*. Retrieved from Flyairshare: https://flyairshare.com.au/blog/cessna-172/#:~:text=The%20burn%20rate%20of%20the,(32.176%20litres)%20per%20hour

Jin, L., Cao, Y., & Sun, D. (2013). *Investigation of Potential Fuel Savings Due to Continuous-Descent Approach.* Indiana: American Institute of Aeronautics and Astronautics.

Lim, Y., Sabatini, R., & Gardi, A. (2018). Energy Efficient 4D Trajectories for Terminal Descent Operations. *International Symposium on Sustainable Aviation.* Rome: Sustainable Aviation Research Society.

Nebylov, A. V., & Watson, J. (2016). *AEROSPACE NAVIGATION SYSTEM.* John Wiley & Sons.

Ogata, K. (2010). Modern Control Engineering. In K. Ogata, *Modern Control Engineering* (p. 894). Prentice Hall.

Prats, X., Bendris, B., Dalmau, R., Montolio, J., Labs, B. D., Lenz, H., & Kohrs, R. (2016). *4D Continuous Descent Operations Supported by an Electronic Flight Bag.* Barcelona: IEEE.

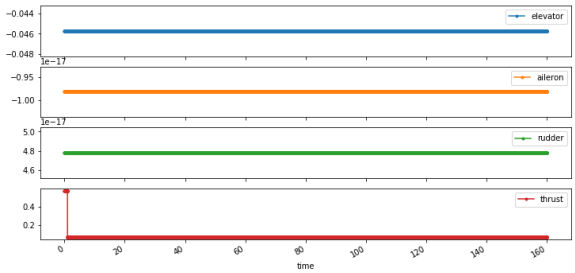
Saez, A., Aqreed, & Rodriguez, J. L. (n.d.). *AeroPython*. Retrieved from PyFME: https://github.com/AeroPython/PyFME/

Verhoeven, R., Bussink, F. J., Prats, X., & Dalmau, R. (2013). *TIME AND ENERGY MANAGED OPERATION (TEMO).* Amsterdam: IEEE.

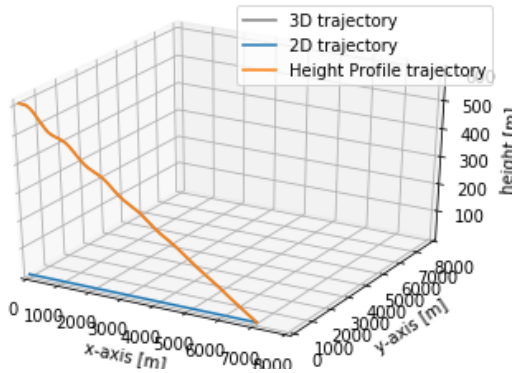
APPENDIX – Result of other scenarios

1. Scenario 2

   Input:

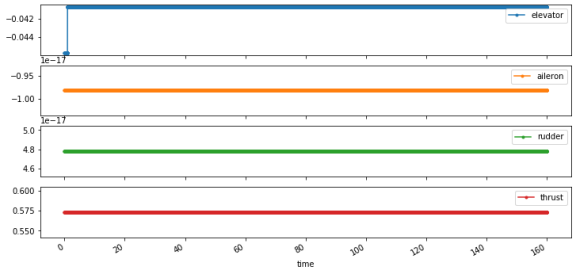


   Output:

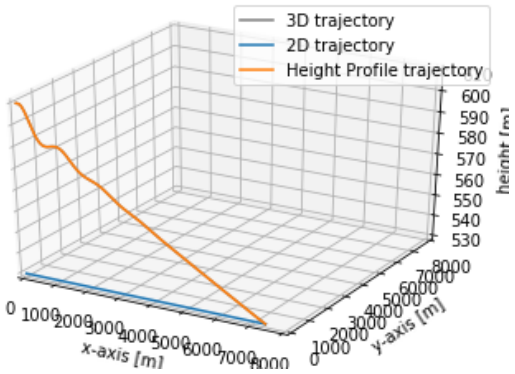


2. Scenario 3

   Input:

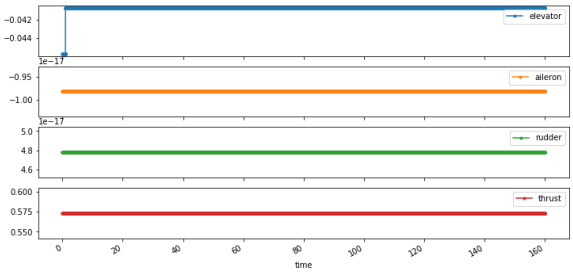


Output:

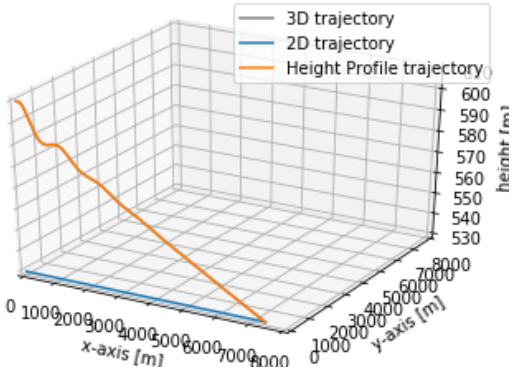


3. Scenario 4

Input:

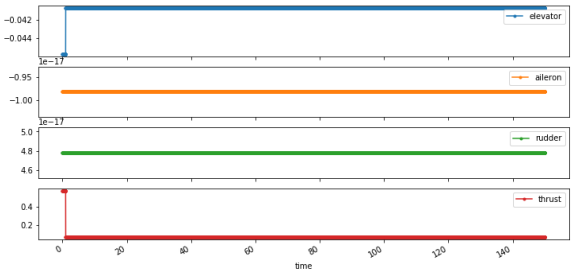


Output:



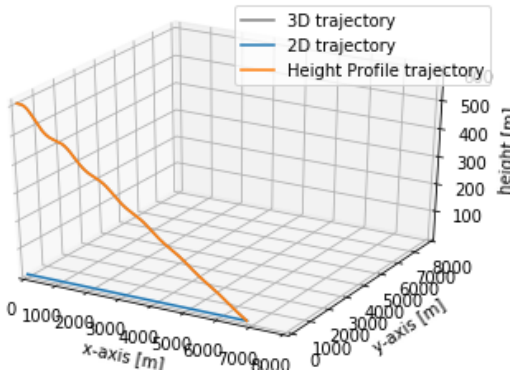
4. Scenario 5

Input:

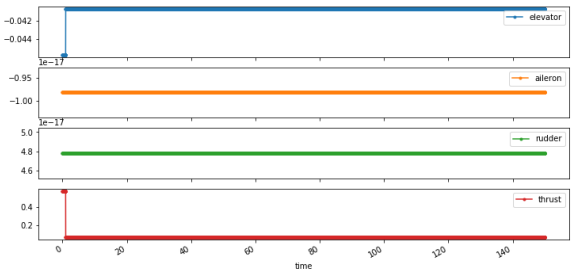


Output:

5. Scenario 6

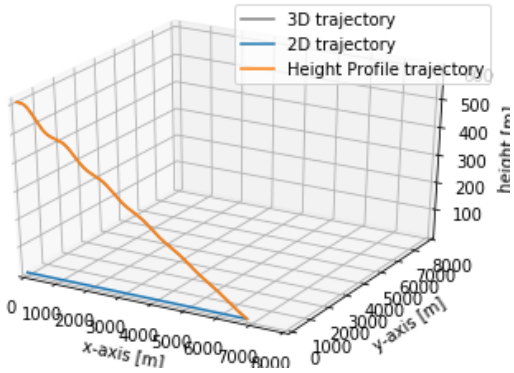


Input:

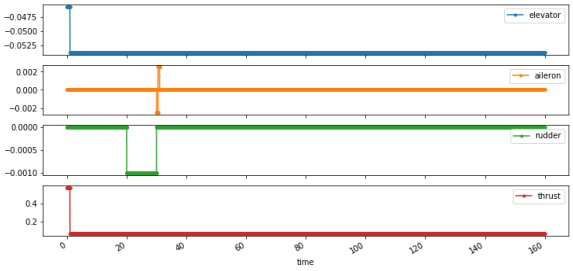


Output:

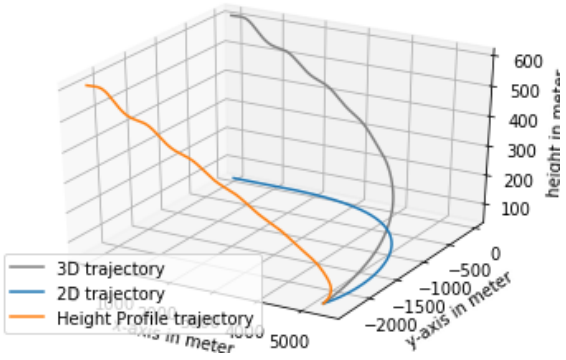
6. Scenario 8



Input:

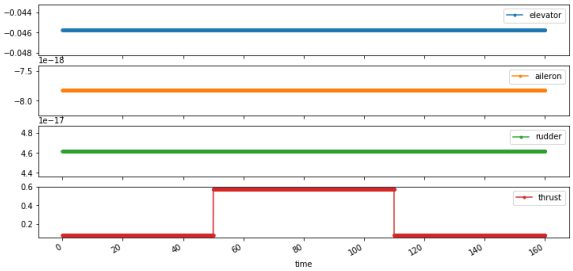


Output



7. Scenario 9

Input:



Output: