ISTANBUL TECHNICAL UNIVERSITY

FACULTY OF AERONAUTICS
AND ASTRONAUTICS

Intr. to Optimal Control

Term Project

LECTURER: Barış Başpınar

Mustafa Eren Asarkaya (110160011)

Hüseyin Bayar (110170028)

Muhammed İsmail Aydınlıoğlu(110170042)

Table 1. Division of Labor

	Literature Review	Part A Coding	Part A Creating Algorithm	Part B Coding	Part B Creating Algorithm	Reporting
Mustafa Eren Asarkaya	+	-	-	+	+	-
Hüseyin Bayar	+	-	+	-	+	+
Muhammed İsmail Aydınlıoğlu	+	+	+	+	-	-

Introduction

Aircraft trajectory analysis research has been conducted. The used method for each case explained at the beginning of each part. Then, the graphic results are shown for each flight condition and each part. Then, the results such as fuel consumption compared in conclusion section and general pros and cons of the used methods are shown.

Definition of the Problem

Regarding the improvement of the trajectory efficiency, its optimization implies a complete analysis in time and space. This approach can be obtained using optimal control theories which may provide the entire set of optimal parameters (including the attitude angles, velocities, consume, etc.) for a trajectory between two points, as a function of time and subjected to the particular restrictions involved. The variable to minimize might be the fuel consumption, the flight time and also functions of both, costs and others. This leads to not only economical savings but also big improvements against environmental impact. In order to define the flight plan, the aircraft dynamics and the elements affecting the flight have been faced by the development of a complete model that leads to realistic simulation. The model includes: dynamic non-linear equations of motion, different aerodynamic configurations, Earth model, aircraft model, wind and among others.

PART A

The most suitable approaches to solve this kind of optimal control problems belongs to the direct collocation methods. While indirect methods follow a” first optimize, then discretize” approach, direct methods use the opposite:” first discretize, then optimize” technique. In this way, the infinite dimensional problem is converted into a finite dimensional one. Among the direct methods family, it can be differenced some different approaches. Direct single shooting methods allow an easy transformation of the optimal control problem to a Non-Linear Programming (NLP) problem. However, they are very sensitive to initial guess perturbations. Some of the problems are reduced with direct multiple shooting methods, but when including inequality restrictions, the constrained curves must be specified in advance.

The problem is subjected to a set of equations of motion which are described by first order ODEs. These dynamic equations must be rewritten to the form:

$$\dot{x}(\tau) = \frac{t_f - t_i}{2} f(x(\tau), u(\tau), p)$$

Also path constraints and algebraic equations must be rewritten to:

$$\phi(\tau)[x(t), u(\tau), p] \leq 0;$$

$$0 = \mathrm{g}[x(\tau), u(\tau), p]$$

The derivative of the state vector $\tilde{x}(\tau_j)$ is approximated, where D is computed as a function of the number of nodes, N, of the phase. Consider now the following matrices;

- Matrix U with dimensions x_u × (N + 1) stores the trajectories of the controls

$$U = [u(\tau_0), u(t_1), \dots, u(t_n)]$$

- Matrices X and $\dot{X}$ with dimensions x_x ×(N + 1) stores the trajectories of the states and their approximate derivatives

$$X = [x(\tau_0), x(t_1), \dots, x(t_n)]$$

$$\dot{X} = [\tilde{x}(\tau_0), \tilde{x}(t_1), \dots, \tilde{x}(t_n)]$$

- Matrix F with dimensions x_x × (N + 1) contains the right hand side of differential constraints evaluated

$$F = \frac{t_f - t_i}{2} [f[x(\tau_0), x(t_0), x(t_0)], \dots, f[x(\tau_n), x(t_n), x(t_n)]]$$

Then, the differential equations are transformed into algebraic constraints expressed as:

$$\left[\dot{X} - F\right] = [XD^T - F] = 0$$

and the decision vector y as:

$$y = \begin{bmatrix} U \\ X \\ p \\ t_i \\ t_f \end{bmatrix}$$

The problem to solve is the NLP problem described in the introduction of part a.

Flight 1

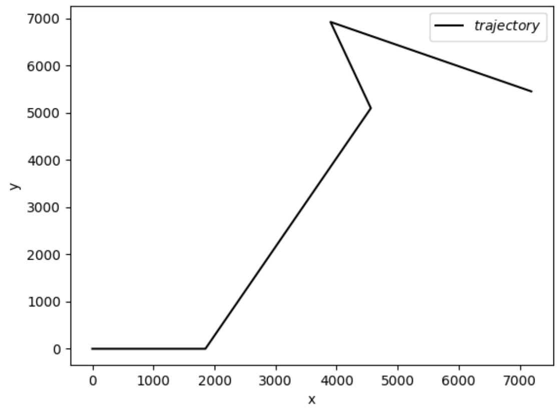

Figure 1. x-y graphic

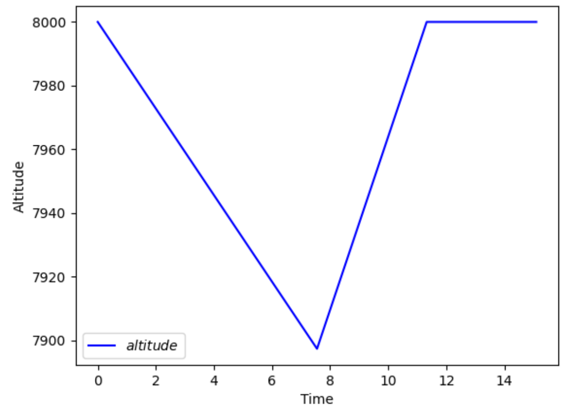

Figure 2. Altitude (h)-Time graphic

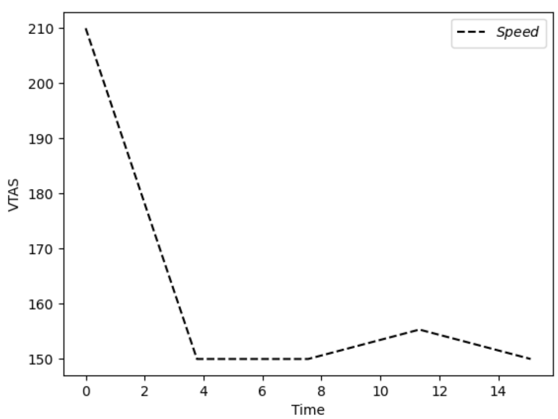

Figure 3. V_{tas} −Time graphic

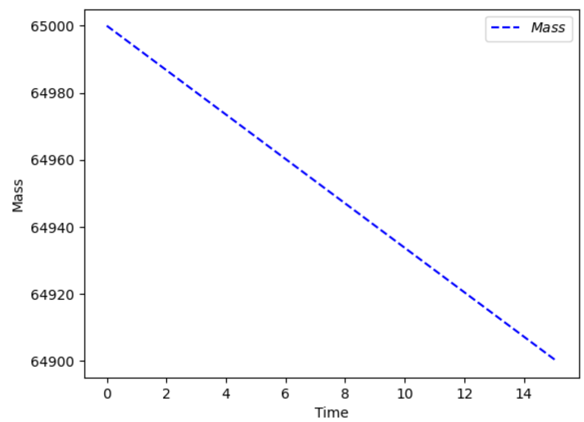

Figure 4. Mass (kg)-Time graphic

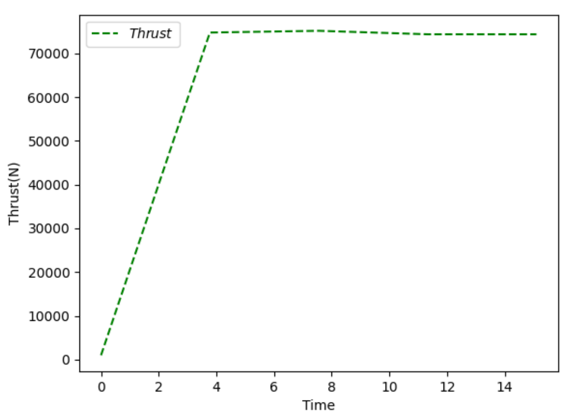

Figure 5. Thrust-Time graphic

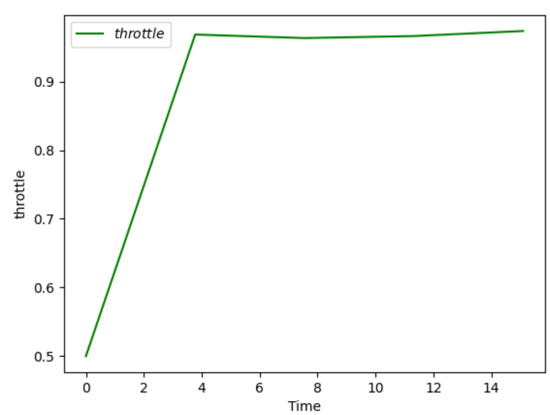

Figure 6. Throttle (δ)-Time graphic

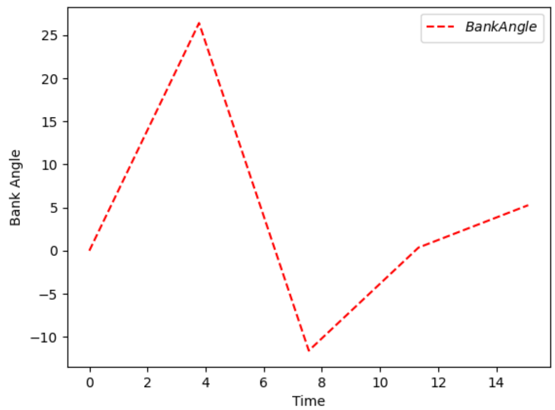

Figure 7. Bank Angle (μ)-Time graphic

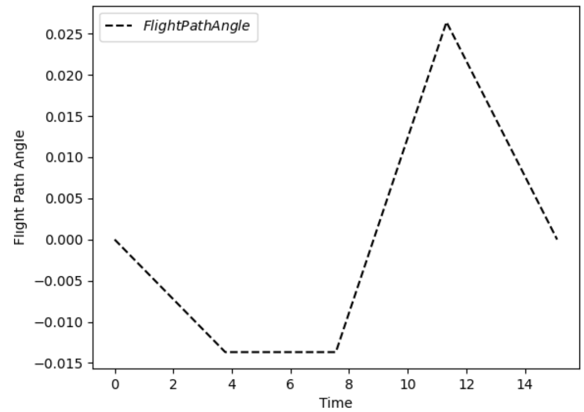

Figure 8. Flight Path Angle (γ)-Time graphic

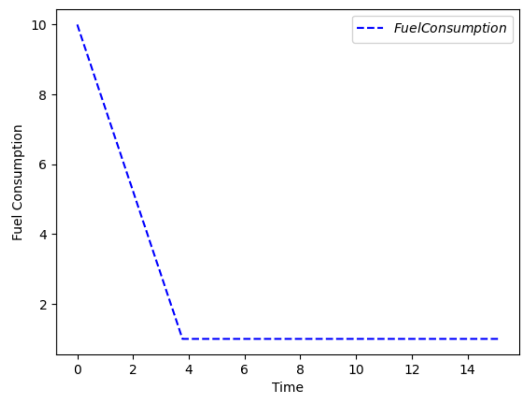

Figure 9. Fuel Consumption-Time graphic

Flight 2

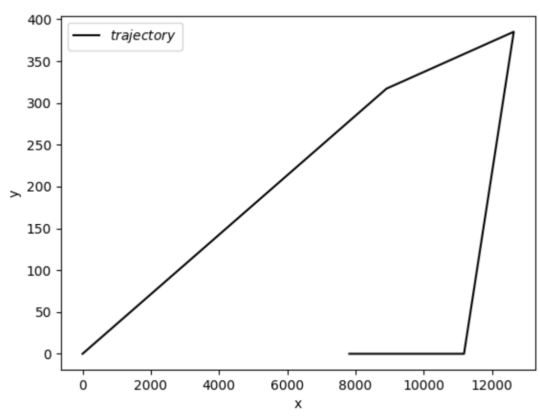

Figure 10. x-y graphic

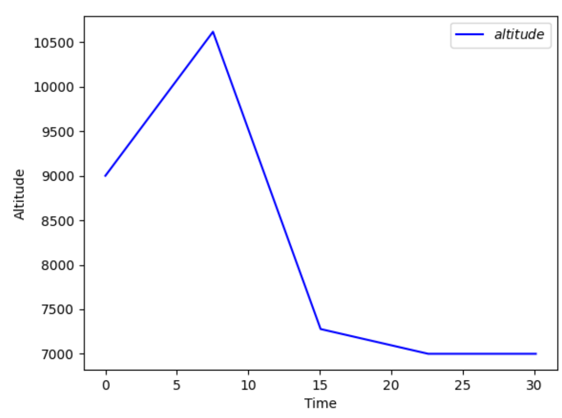

Figure 11. Altitude (h)-Time graphic

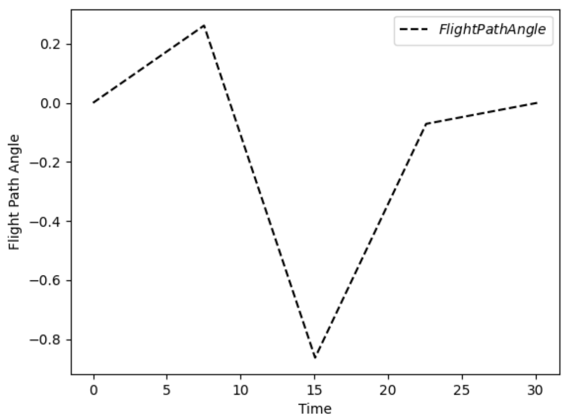

Figure 12. Flight Path Angle (γ)-Time graphic

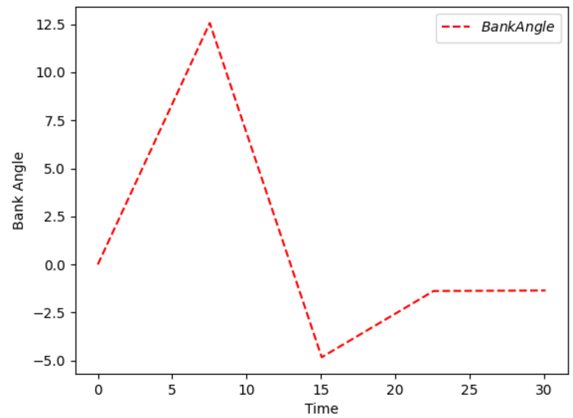

Figure 13. Bank Angle (μ) -Time graphic

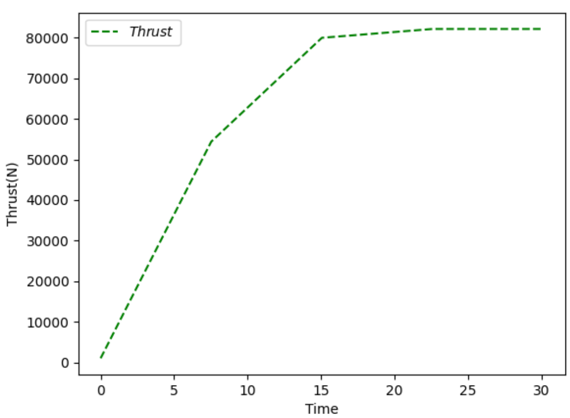

Figure 14. Thrust-Time graphic

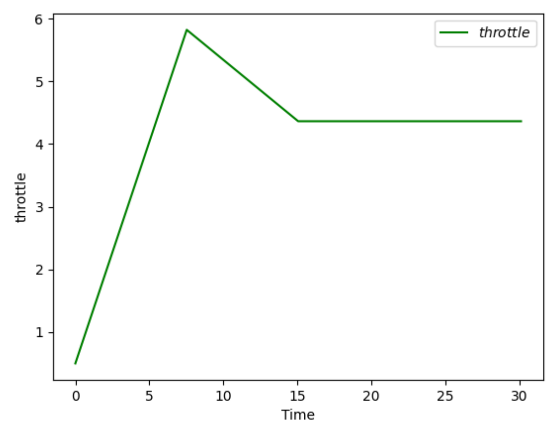

Figure 15. Throttle (δ)-Time graphic

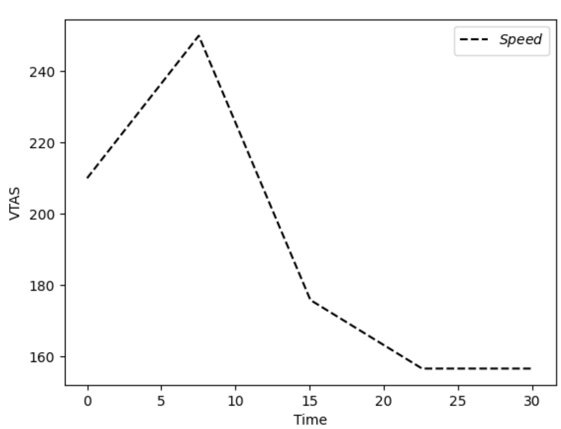

Figure 16. V_{tas} −Time graphic

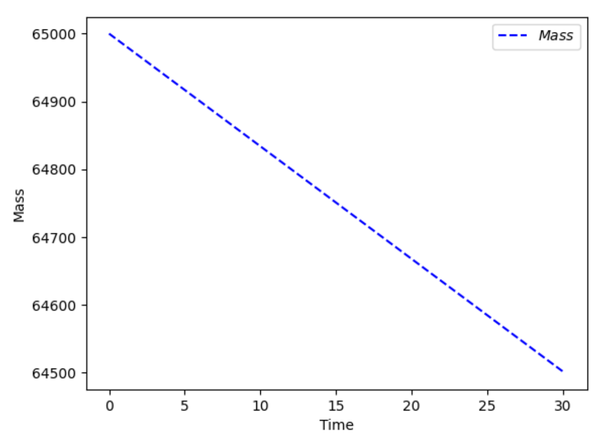

Figure 17. Mass(kg)-Time graphic

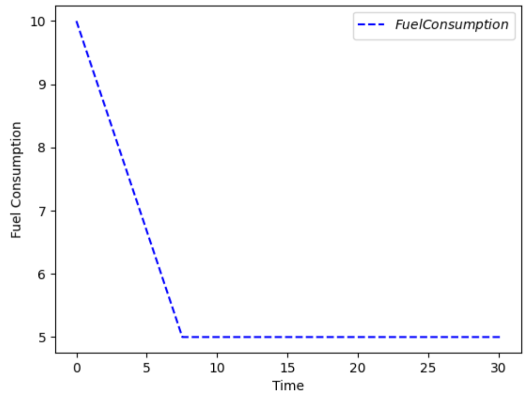

Figure 18. Fuel Consumption-Time graphic

Flight 3

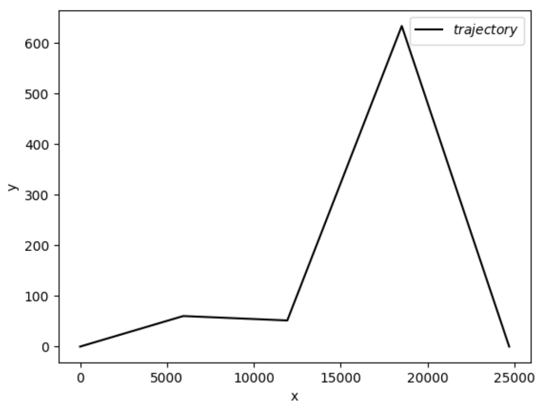

Figure 19. x-y graphic

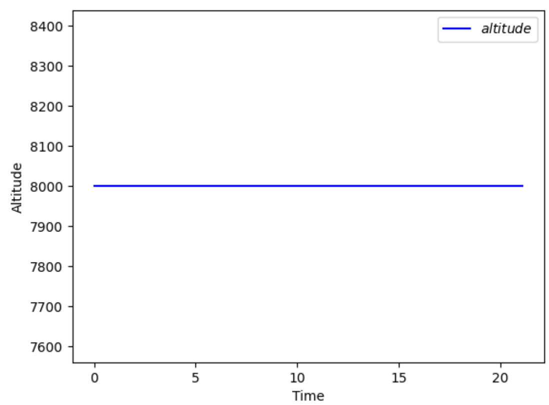

Figure 20. Altitude (h)-Time graphic

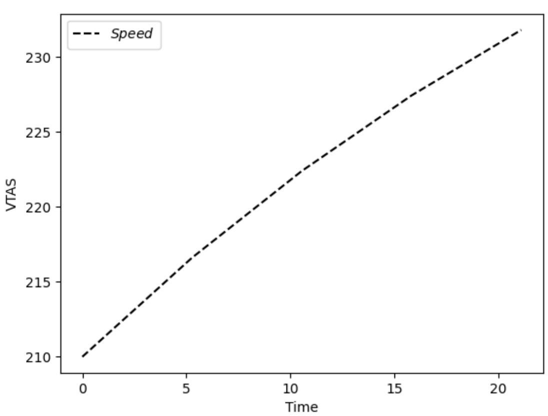

Figure 21. V_{tas} –Time graphic

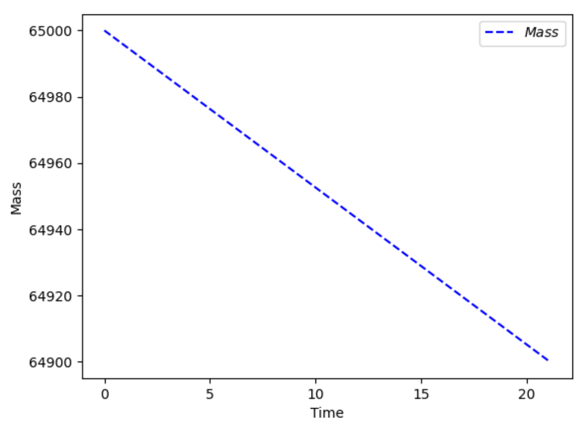

Figure 22. Mass(kg)-Time graphic

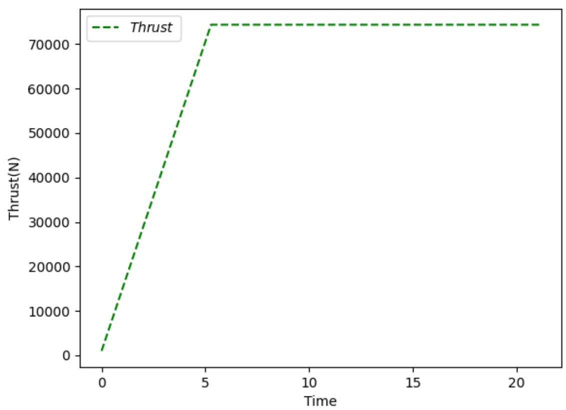

Figure 23. Thrust-Time graphic

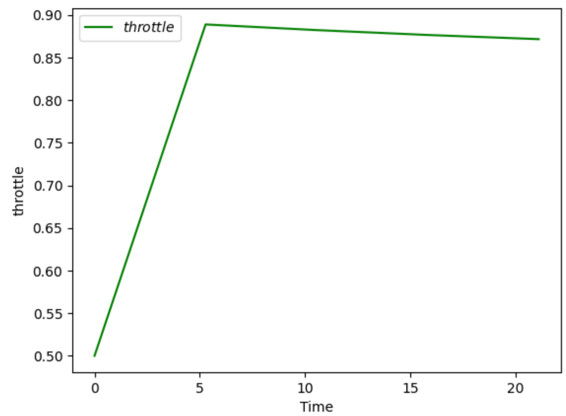

Figure 24. Throttle (δ)-Time graphic

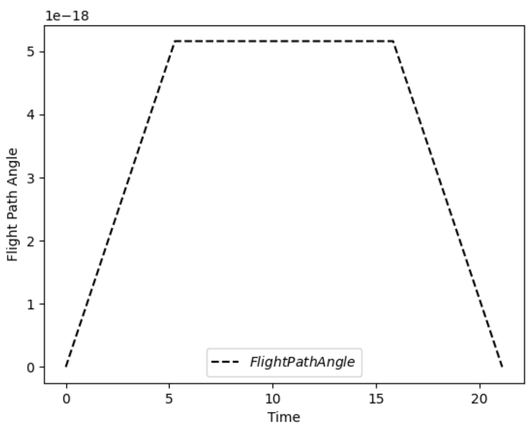

Figure 25. Flight Path Angle (γ)-Time graphic

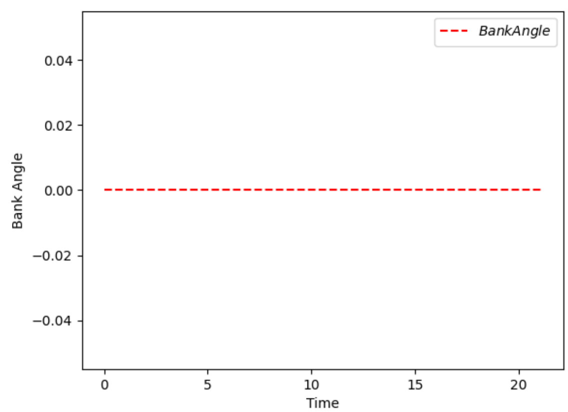

Figure 26. Bank Angle (μ) -Time graphic

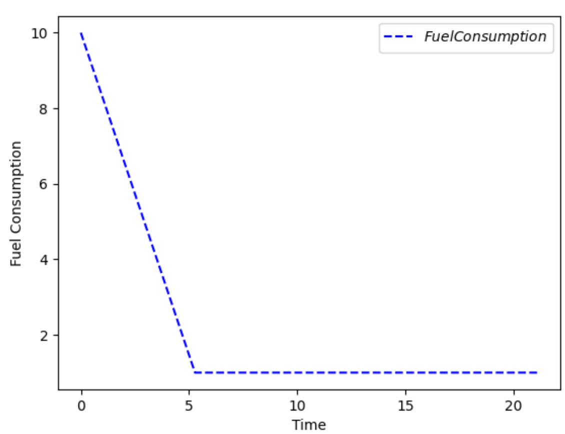

Figure 27. Fuel Consumption-Time graphic

PART B

Dynamic programming methods use Hamilton-Jacobi-Bellman equation to derive some optimality conditions such as linear systems with quadratic cost → Riccati Equation. The numerical framework of DP is very suitable to handle discrete-time dynamic systems with nonlinear characteristics. Moreover, the 3D waypoint representation of the flight trajectory is similar to the discretization of the states grid system; consequently, DP is a natural numerical method to deal with the 3D flight-trajectory optimization. Other great advantages of using DP are that it not only guarantees an absolute (global) optimum, but it also can easily handle equality and inequality constraints of the system. It is still not widely used in many practical applications due to the computational burden, known as the curse of dimensionality, and the interpolation problem (when the trajectory from a grid point does not reach exactly the next grid point), known as the menace of the expanding grid.

Considering a nonlinear system whose dynamics is modelled by a set of ordinary differential equations as:

$$\dot{X}(t) = f[t, X(t), U(t)]$$

this can be approximated by a set of difference equations as follows:

$$X_{k+1} = f[t_k, X_k, U_k] \tag{1}$$

where X_k and U_k are respectively the state and control vector with appropriate boundary conditions at any stage k with (k = 0, 1, . . . , N − 1).

The Performance Index (PI) of Equation $J = \Phi\left[t_f, X\left(t_f\right)\right] + \int_{t_0}^{t_f} L[t, X(t), U(t)]dt$ can be approximated by a summation as follows

$$J = \Phi[t_N, X_N] + \sum_{k=0}^{N-1} L[k, X_k, U_k] \tag{2}$$

This assumes that the optimal control, state, and cost are known from initial stage 0 to any stage k. Then, at any stage k + 1, the Principle of Optimality (PO) states that whatever the initial state and the initial decision, in this case, X_{k+1} and U_{k+1}, the remaining decision U_k must be optimal with regard to the state X_k that results from the first decision U_{k+1}. Thus, the PI can be rewritten as:

$$J^*_{k+1}(X_{k+1}) = \min_{U_{k+1}} [\, L[t_{k+1}, X_{k+1}, U_{k+1}] \; + \; J^*_k(X_k)]$$

This equation is the mathematical form of PO; it is also known as the functional equation of DP, where J^*_{k+1} represents the cost of the optimal path from initial stage 0 to any stage k + 1, and J^*_k is the optimal cost from initial stage 0 to any stage k.

Flight 1

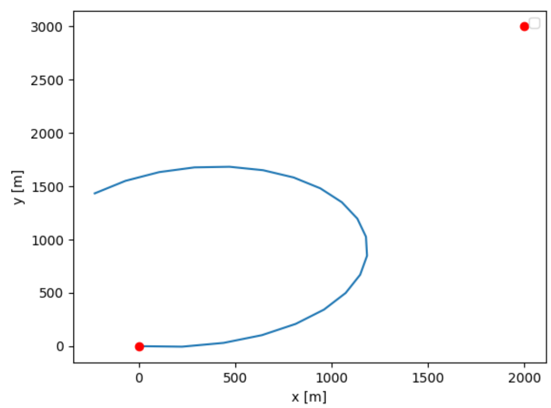

Figure 28. x-y graphic

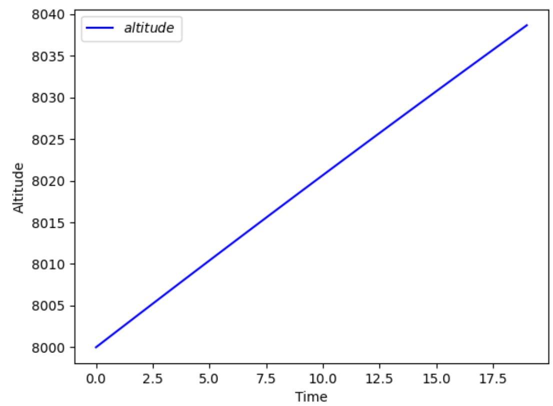

Figure 29. Altitude (h)-Time graphic

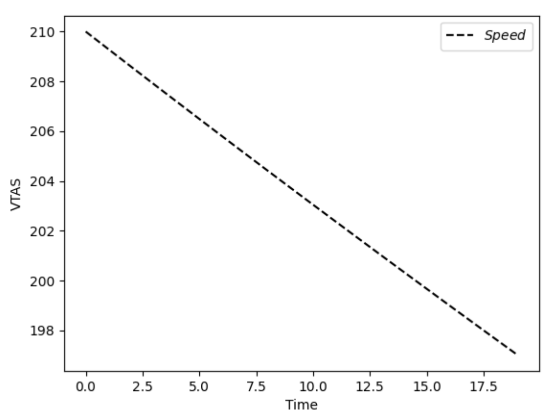

Figure 30. V_{tas} −Time graphic

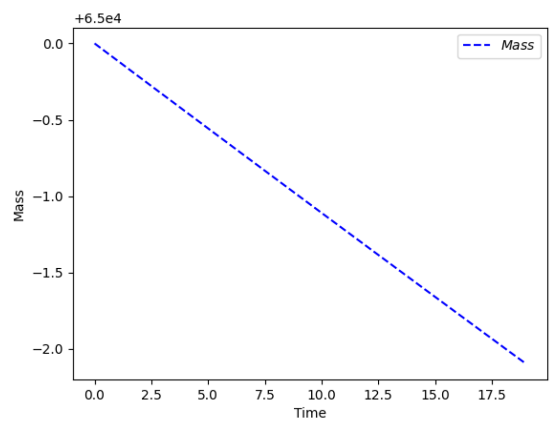

Figure 31. Mass(kg)-Time graphic

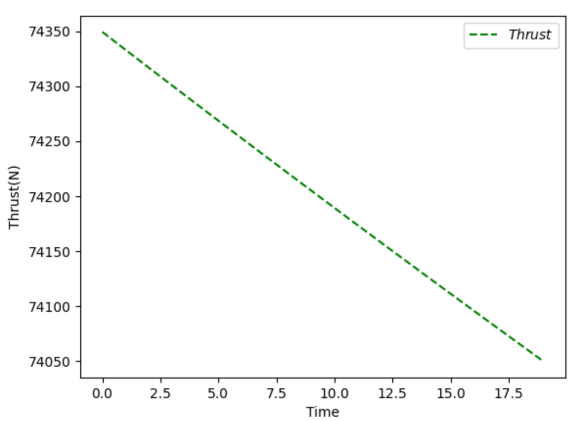

Figure 32. Thrust-Time graphic

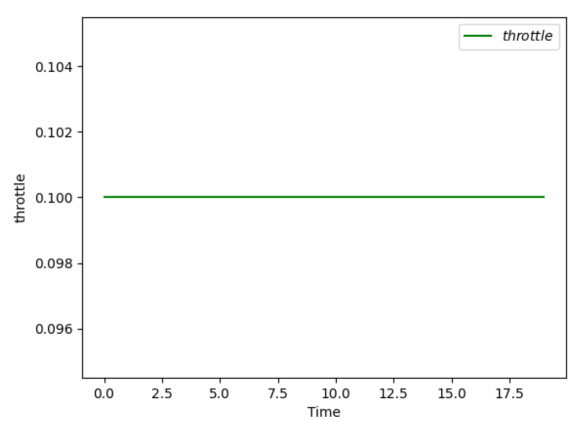

Figure 33. Throttle (δ)-Time graphic

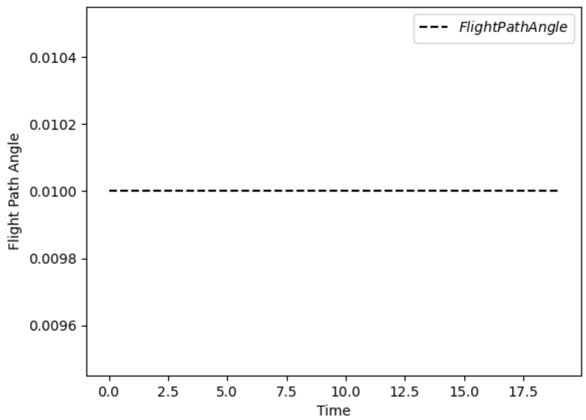

Figure 34. Flight Path Angle (γ)-Time graphic

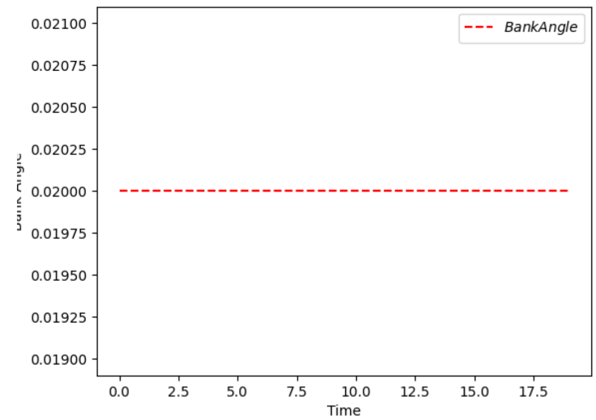

Figure 35. Bank Angle (μ) -Time graphic

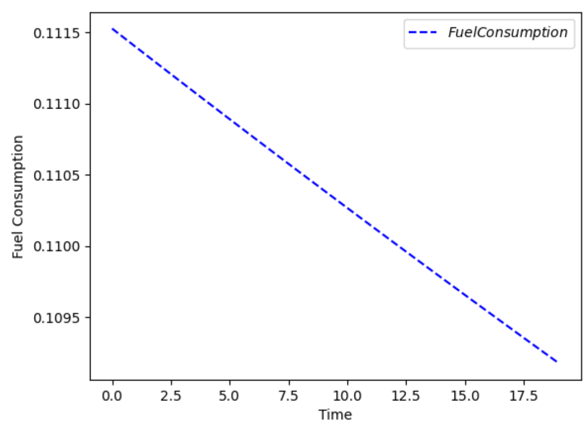

Figure 36. Fuel Consumption-Time graphic

Flight 2

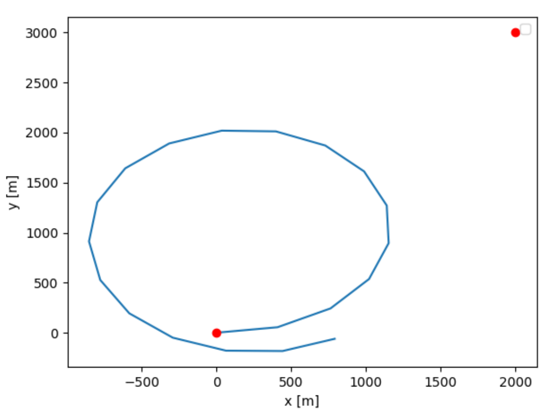

Figure 37. x-y graphic

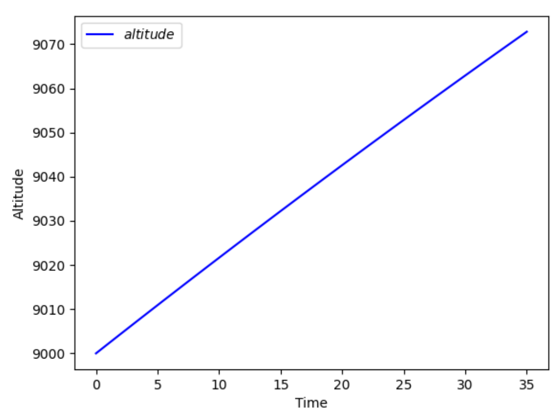

Figure 38. Altitude (h)-Time graphic

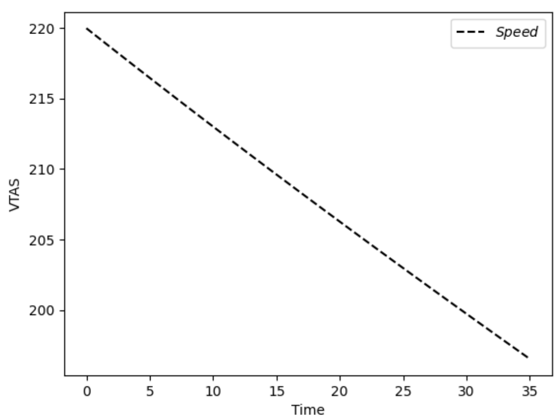

Figure 39. V_{tas} **−Time graphic**

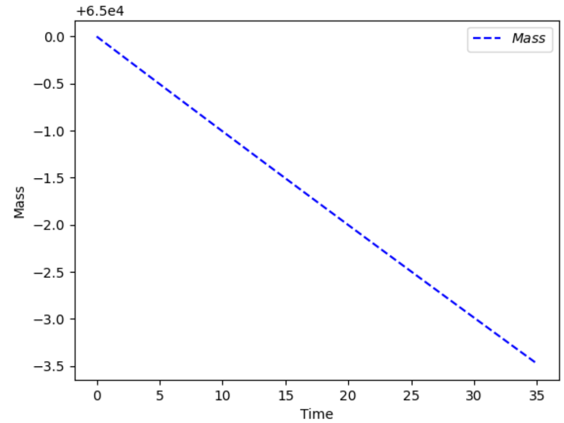

Figure 40. Mass(kg)-Time graphic

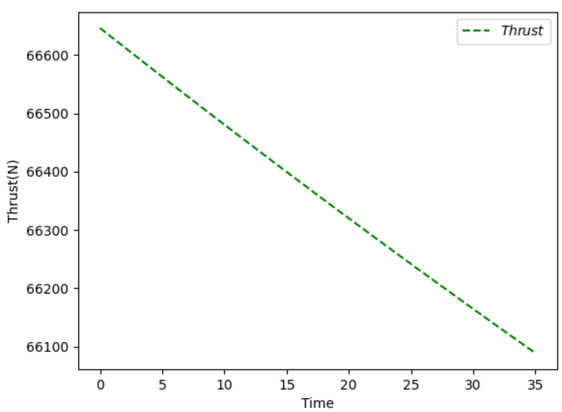

Figure 41. Thrust-Time graphic

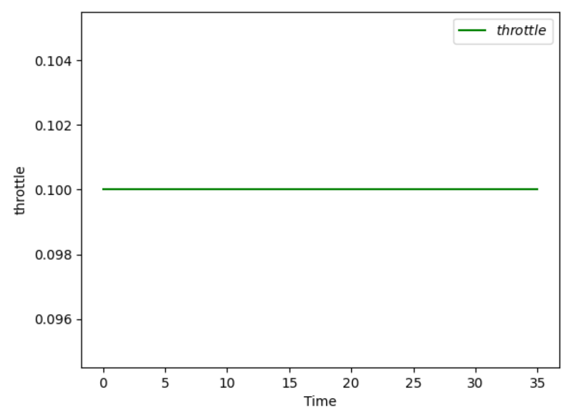

Figure 42. Throttle (δ)-Time graphic

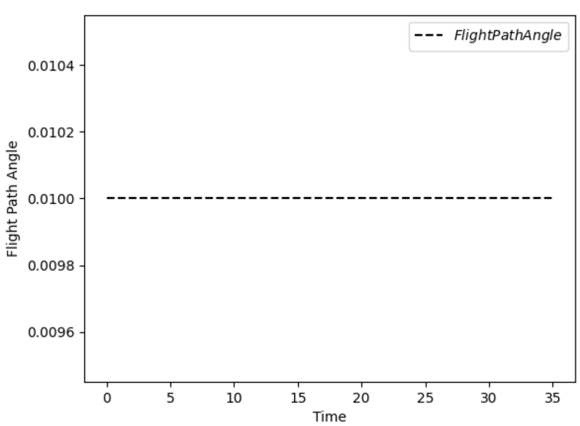

Figure 43. Flight Path Angle (γ)-Time graphic

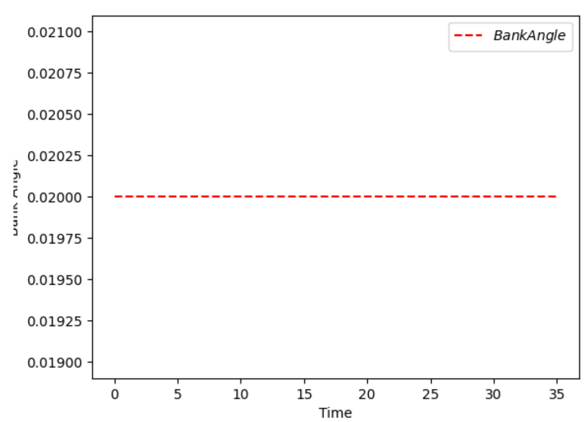

Figure 44. Bank Angle (μ) -Time graphic

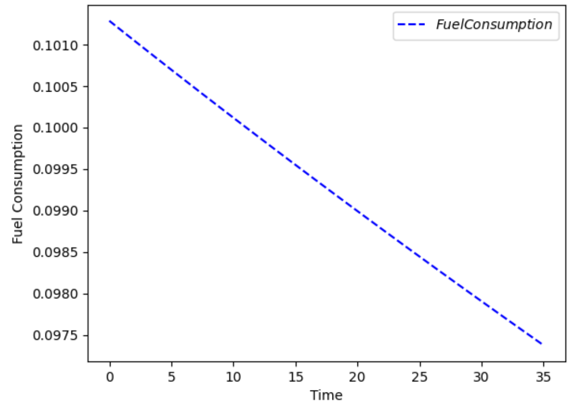

Figure 45. Fuel Consumption-Time graphic

Flight 3

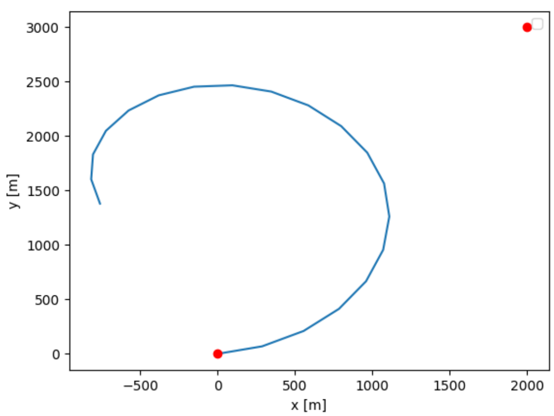

Figure 46. x-y graphic

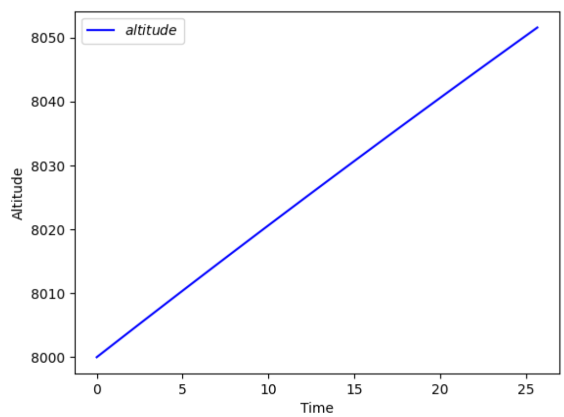

Figure 47. Altitude (h)-Time graphic

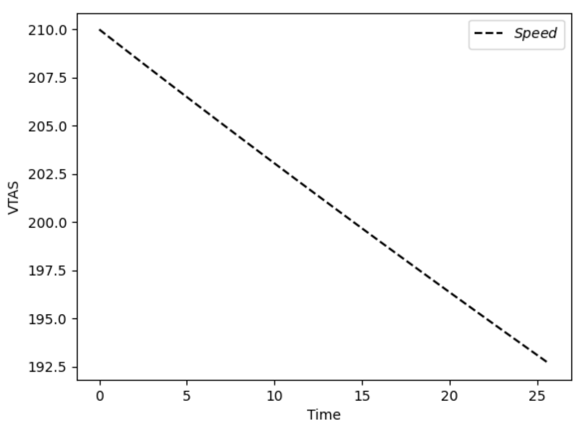

Figure 48. V_{tas} –Time graphic

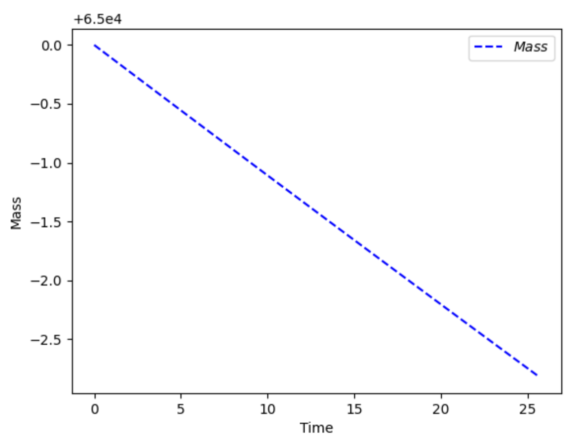

Figure 49. Mass(kg)-Time graphic

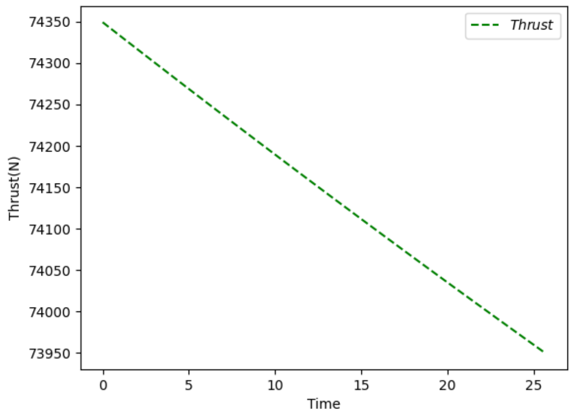

Figure 50. Thrust-Time graphic

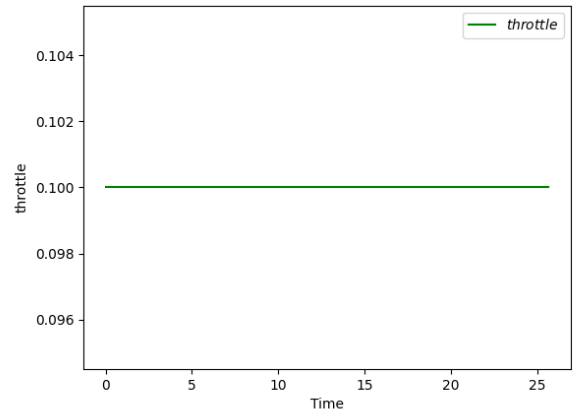

Figure 51. Throttle (δ)-Time graphic

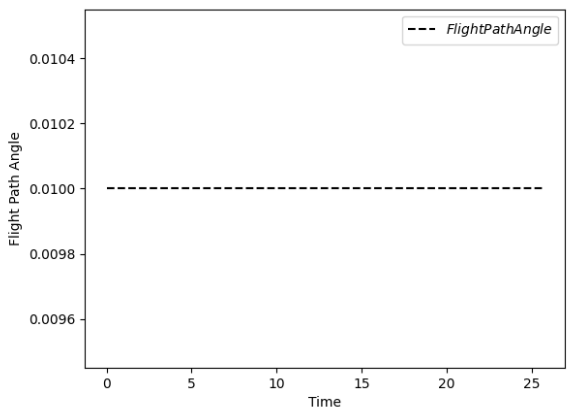

Figure 52. Flight Path Angle (γ)-Time graphic

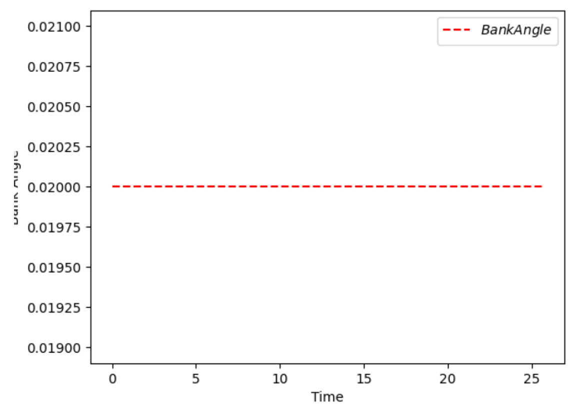

Figure 53. Bank Angle (μ) -Time graphic

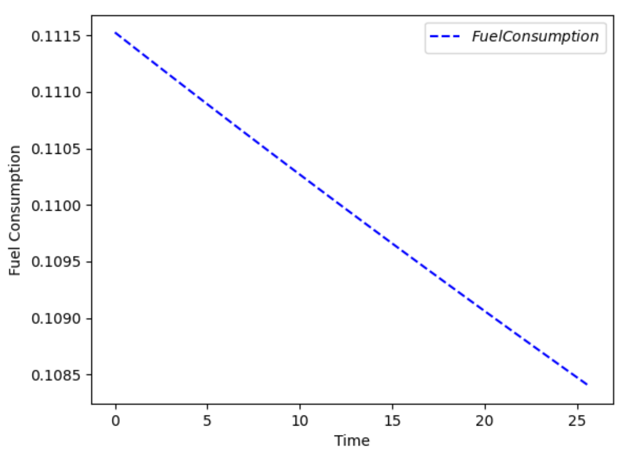

Figure 54. Fuel Consumption-Time graphic

CONCLUSION

Table 2. Fuel consumption comparison

Flight Number	Spent Fuel Calculated with Direct Method	Spent Fuel Calculated with Dynamic Programming
Flight 1	100 kg	2000kg
Flight 2	500 kg	3500kg
Flight 3	100 kg	2500kg

In this Project, trajectory optimization for a jet passenger aircraft (Boeing 737-800) was studied to minimize the fuel consumption. The calculation results for 3 flight cases were as follows:

- Only in flight 2, Aircraft used its max thrust.
- In flight condition 2, aircraft used most fuel.
- From the mathematical point of view, it can be noticed once again that the order of magnitude of the states affect to the error and the deviations. Moreover, greater fluctuations are found for the angles and V_{tas}.
- As seen in throttle,bank angle, flight path angle graphics dynamic programming code could not complete the missons. Therefore, it can be clearly said that direct method solution is more accurate.

PROS of the IMPLEMENTED METHODS

- A numerical optimization method which is easy to handle and provides an optimal solution in a short time is necessary in the design of guidance and control system for aerospace vehicles.
- Can use state-of-the-art methods for NLP solution.
- Can treat inequality constraints and multipoint constraints much easier.

→Dynamic Programming Searches whole state space, finds global optimum.

→Optimal feedback controls precomputed.

→Analytic solution to some problems possible (linear systems with quadratic cost → Riccati Equation)

CONS of the IMPLEMENTED METHODS

- This direct method has disadvantages, that is, the global optimality is not guaranteed and computational time is uncertain because of iterative calculations.
- This Direct Method obtains only suboptimal/approximate solution.

→ Dynamic Programming is, in general, intractable, because partial differential equation (PDE) in high dimensional state space: “curse of dimensionality”

→ Dynamic Programming finds the optimal solution from every combination of transitions of grid points in the state space. Accuracy of the solution depends on the grid, and it often needs some adjustment to make the solution accurate enough to be useful for practical purpose. Furthermore, if the number of control variables is not enough to realize the transition between the two grid points, it requires fine grid and large computational time.

References

[1] Ahmed, K., Bousson, K., & Coelho, M. de. (2021). A modified dynamic programming approach for 4D minimum fuel and emissions trajectory optimization. *Aerospace*, *8*(5), 135. https://doi.org/10.3390/aerospace8050135

[2] Guijarro, R. A. (2015). *Commercial Aircraft Trajectory Optimization Using Optimal Control* (thesis).

[3] https://syscop.de/files/2014ss/noc-summer-school/OptimalControl.pdf