UC Berkeley

CEE 295 : Energy Systems and Control

Project Report : Flight Path Optimization

Students :

Gaofeng Su, Pierre Melikov, Siyuan Feng, Deep Dayaramani, Jaewoong Lee

10th of May, 2019

1 Abstract

This project is based on a crucial issue in the aviation world : how to optimize the trajectory and controls given to the aircraft in order to optimize flight time and fuel consumption. This project aims to provide elements of a response to this problem and to define, under certain simplifying assumptions, an optimal response. using Model Predictive Control (MPC). Thus, the first step will be to define the dynamic model of the aircraft in accordance with the controllable inputs and wind disturbances. Finally, we will identify a precise objective in terms of optimization and implement an optimization program to solve it.

2 Introduction

2.1 Motivation Background

In today's world, we use multiple different ways of transportation to get from one place to the other. These methods of transportation generate a lot of emissions of CO_2 which lead to global warming and rising sea levels. Because of this, the transportation industry has seen a major shift in the technology used just to lower these emissions. But this hasn't been the same case across all different modes of transportation, especially in the airplane industry. Airplanes contribute to 12.5% of the global CO_2 pollution and 80% of the emissions are from flights of over 1500km. This is one of the main reasons this project focuses on flight path optimization. Flight paths contribute to the amount of fuel used, distance travelled which taking into account the wind speed, air pressure and distance between the two locations. They can provide a lot of advantages such as :

— By optimizing the flight path to consume the least fuel while finding the shortest path possible, we can decrease emissions by a large factor. This reduces the overall environmental impact of airplanes.
— A decrease in amount of fuel used saves the airlines a lot of money while benefiting the passengers by decreasing the cost of their overall journey.
— They can minimize the impact of the weather on the aircraft thus reducing the amount of maintenance required and saving the airlines more money.
— Better management of Air Traffic due to predetermined routes.

There are a lot of challenges that come with creating flight plans which include :

— Taking into account the weather over the whole journey. Weather here refers to pressure of air and wind speed at all points during the journey.
— The weight and amount of fuel being carried. This can also be related to the weight and size of the airplane as larger amounts of fuel are usually carried by larger planes

There are many ways to design and determine flight paths. The method that we chose is Model Predictive Control. A major reason for this was that there wasn't a lot of research about using Model Predictive Control (MPC) to design flight paths. One reason might be that MPC is mostly used in real-time optimum control problem (e.g. path following problem). Hence, it is interesting to discuss the possibility of using MPC for a nonlinear, large scale optimization problem, which we have further explored through this project.

Another reason we chose this topic was that our team had members taking CE263, Scalable Spatial Analysis where they were learning about data processing (e.g. data decoding, data arrangement, visualization), especially applied to the wind database we included in our model.

2.2 Relevant Literature

A MODEL PREDICTIVE CONTROL APPROACH TO AIRCRAFT MOTION CONTROL (Luca Deori, 2015–XXVIII November)

This goal of this paper is to develop a Model Predictive Control model that can be used in aircraft reference path following problem. This paper and our project are both try to establish a MPC optimization problem subject to the aircraft dynamics, aircraft performance constraints, wind disturbance. This can provide us some idea how to establish a appropriate aircraft model and relevant constraints.

However, there are some key difference between the paper and our project. This paper use MPC for a path following problem with a 3D aircraft model. The solution involve feedback control strategy and MPC was used to solve a moving finite horizon optimization problem at each time step/feedback loop until it reach the end of the reference path. In this case, the flight path was well-defined before using MPC.

Our project is to use MPC to find the optimum flight path (travel time focus or fuel cost focus), where the finite time horizon cover the whole flight trip, which make the scale of the problem extremely large.

2.3 Objective of this Study

Implement Model Predictive Control (MPC) to optimized the flight path with minimum travel time focus and minimum fuel cost focus from Chicago O'Hare International Airport (ORD) to San Francisco International Airport (SFO) taking forecast wind data into account.

3 Technical Description

The implementation and resolution of the problem is based on four parts. First of all, the dynamic model of the aircraft is configured and developed, according to the objectives set and the degree of precision required. In a second step, we do the same for the wind, and integrate this modeling with the aircraft one. Once the dynamic model is established, it must then be discretized. Indeed, given the complexity and non-linearities of the function describing the dynamic behavior of the system, an analytical solution to our problem is difficult to find. And so we choose to solve the problem numerically. Finally, the final optimization problem will be implemented in the last phase.

3.1 Aircraft model

A aircraft point mass model (PMM) will be introduced in the path planning optimization problem. The original model is in 3-dimension, however, to simplify the problem, the original model was reduced into a 2-dimension model.

$$\begin{aligned}
\dot{x} &= v\cos(\theta) + w_x \\
\dot{y} &= v\sin(\theta) + w_y \\
\dot{v} &= \frac{T - D}{m} = \frac{2T - C_d \rho A v^2}{2m} \\
\dot{m} &= -\eta T \\
\dot{\theta} &= \frac{g}{v} tan(\phi)
\end{aligned} \tag{1}$$

x : west-east distance [m]
y : north-south distance [m]
w_x : wind disturbance on x direction [m/s]
w_y : wind disturbance on y direction [m/s]
v : aircraft ground speed [m/s]
m : aircraft mass [kg]
θ : heading angle [rad]
C_d : drag coefficient [-]
ρ : air density [kg/m^3]
A : aircraft wing area [m^2]
η : thrust specific fuel consumption coefficient [kg/N·s]
T : aircraft thrust force [N]
ϕ :bank angle [rad]

In this model, $\dot{\theta}$ involves the aircraft speed, which appears in the denominator. This implies that v cannot be 0. For a path following problem, this is acceptable. However, in a path planning problem, nonzero speed restriction implies that the aircraft can only reach the destination at the end of the given time horizon (i.e. the aircraft cannot reach the destination with a minimum travel time and stop there). Also, for nonlinear programming, the nonlinear solver is very sensitive to the form of the optimization problem and the constraints parameters. Hence, it is better to avoid putting speed in the denominator.

Therefor, the expression of $\dot{\theta}$ was reformulated and replaced by a new input variable φ [rad/s], which represent the turning rate. The final aircraft point mass model that will be used in MPC path planing :

$$
\begin{aligned}
\dot{x} &= v\cos(\theta) + w_x \\
\dot{y} &= v\sin(\theta) + w_y \\
\dot{v} &= \frac{2T - C_d\rho A v^2}{2m} \\
\dot{m} &= -\eta T \\
\dot{\theta} &= \varphi
\end{aligned}
\tag{2}
$$

3.2 Wind model

Historical Track dataset includes 1336 flights between ORD and SFO from July,2013 to August,2013. Departure time, geographical coordinates of a time during the flight, ACID can be read from the dataset. Those flights were completed by two types of aircraft : Airbus 319 and Airbus 320.

After analysing all the 1336 paths, five ARTCC Zones (Air Route Traffic Control Centers), ZAU, ZDV, ZLC, ZMP and ZOA, are commonly crossed by flight departure from ORD to SFO. The boundaries of these five zones would be used to crop wind data. Figure displayed below shows an example of past tracks completed by VRD211.

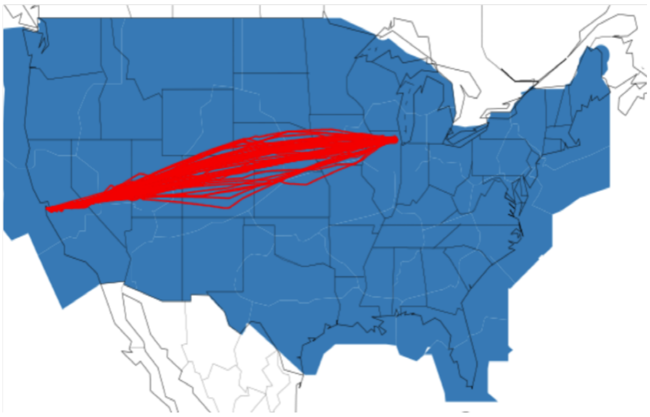

FIGURE 1 – Actual Flight Tracks from ORD to SFO

In our MPC model, forecasted wind data are used. The data are obtained from National Center for Environmental Prediction (NCEP). NCEP generates predicted wind data by Rapid Refresh (RAP) method, the continental-scale hourly-updated National Oceanic Atmospheric Administration (NOAA) assimilation/modeling system. RAP is comprised of a numerical forecast model and an analysis/assimilation system.

The data are given with locations and wind components in 2D-coordinates. The location data contain x and y coordinates in longitude and latitude, and the wind data contain wind speed in x and y direction at the corresponding location.

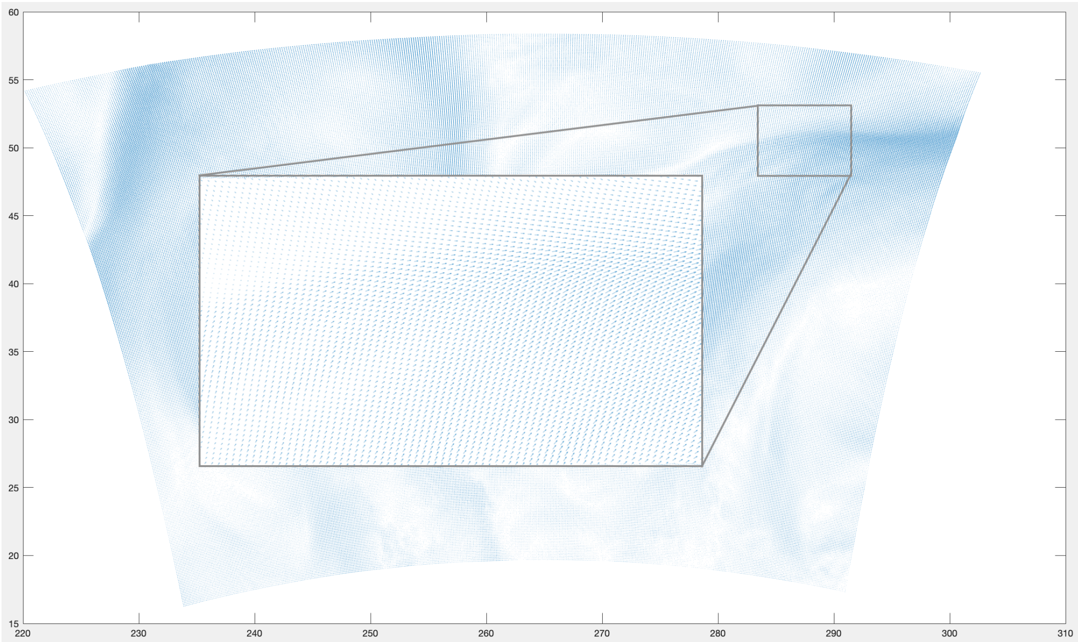

FIGURE 2 – Vector field generated with the raw wind data collected on 07/02/13 at 00 :00.

Wind components are added into our dynamics to create a more realistic MPC model.

$$\begin{aligned} \dot{x_k} &= v_k \cos(\theta_k) + w_{xk} \\ \dot{y_k} &= v_k \sin(\theta_k) + w_{yk} \end{aligned} \tag{3}$$

However, the raw wind data obtained from NCEP cannot be directly combined with our dynamics since the raw wind data components are not directly compatible with the velocity of the aircraft. To convert the raw wind data into the compatible form with the velocity dynamics of the aircraft, we developed a mathematical expression.

We first pick out the five regions defined by Federal Aviation Administration that all the real aircraft path goes through. The wind force are considered in two perpendicular directions u and v. Then we take the average of the wind force for all the 5 time slots that is accessible, and obtained the average wind force spatial distribution in the two directions u and v. We also transform the 2D coordinate systems from WGS84 to WGS 84/Pseudo-Mercator with python package pyproj, and then apply another linear transformation to set the origin of the coordinates system to the departure location. The unit is set to kilometer.

However, the discrete wind data will be hard to handled as input into the aircraft dynamics. Thus, we construct two polynomial functions to fit the two wind force spatial distribution.Here, we use x and y to represent coordinates along longitude and latitude directions. The trick is how to determine the degree and items of the polynomial functions.

From the plots above, we could find the wind force in u direction is high correlated in x and y, while the correlation is weak for the v direction. Based on this finding, we try different polynomial functions with degrees ranging from 3 to 9 to see their performance in fitting the real wind force distribution. A least square method is applied to specify the parameter of the polynomial functions and the best performances are found when the degree is set to 5. The results are shown below. Figure 2 to Figure 4 show decent fitting performance of the two designed polynomial functions.

$$w_{xk} = a_1 y_k^4 + a_2 y_k^3 + a_3 y_k^2 + a_4 y_k + a_5 + a_6 x_k^4 + a_7 x_k^3 + a_8 x_k^2 + a_9 x_k + a_{10} + a_{11} y_k^3 x_k + a_{12} y_k^2 x_k^2 + a_{13} y_k x_k^3$$

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \\ a_{11} \\ a_{12} \\ a_{13} \end{pmatrix} \approx \begin{pmatrix} 5.404 \cdot 10^{-12} \\ -7.525 \cdot 10^{-9} \\ -1.010 \cdot 10^{-5} \\ 1.8023 \cdot 10^{-3} \\ 3.054 \cdot 10^{-1} \\ 1.071 \cdot 10^{-12} \\ 8.131 \cdot 10^{-9} \\ 1.957 \cdot 10^{-5} \\ 1.360 \cdot 10^{-2} \\ 3.054 \cdot 10^{-1} \\ -4.493 \cdot 10^{-13} \\ 1.372 \cdot 10^{-12} \\ -1.971 \cdot 10^{-12} \end{pmatrix}$$

$$w_{yk} = b_1 y_k^4 + b_2 y_k^3 + b_3 y_k^2 + b_4 y_k + b_5 + b_6 x_k^4 + b_7 x_k^3 + b_8 x_k^2 + b_9 x_k + b_{10}$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \\ b_8 \\ b_9 \\ b_{10} \end{pmatrix} \approx \begin{pmatrix} 6.505 \cdot 10^{-12} \\ -2.358 \cdot 10^{-10} \\ 2.009 \cdot 10^{-6} \\ 8.207 \cdot 10^{-6} \\ 6.216 \\ -2.184 \cdot 10^{-12} \\ -1.574 \cdot 10^{-08} \\ -1.790 \cdot 10^{-5} \\ 3.587 \cdot 10^{-2} \\ 6.216 \end{pmatrix}$$

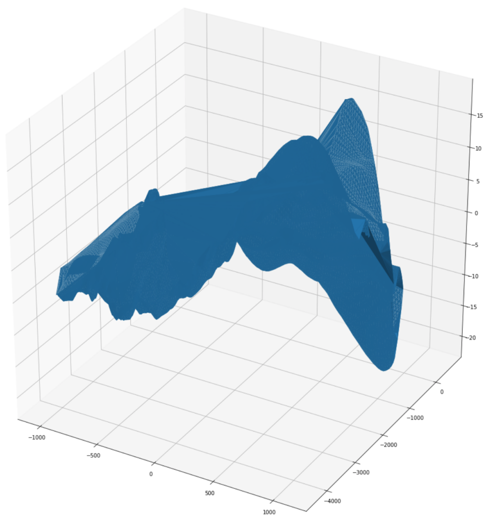

FIGURE 3 – Actual wind force distribution in u direction

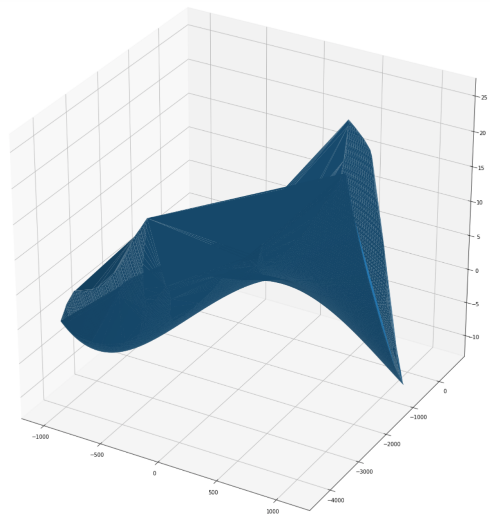

FIGURE 4 – Approximate function in u direction

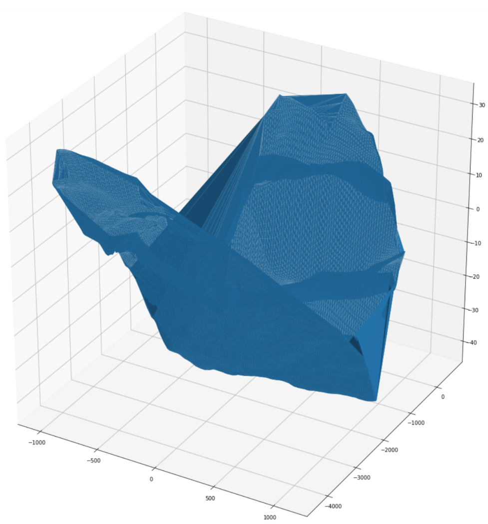

FIGURE 5 – Actual wind force distribution in v direction

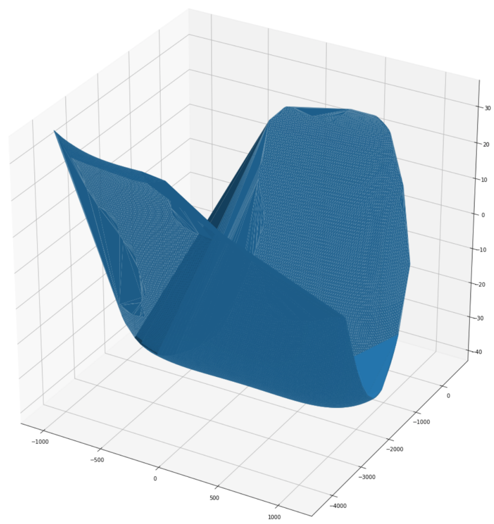

FIGURE 6 – Approximate function in v direction

3.3 Discretization

The continuous dynamic model established earlier is therefore discretized. The time derivative of a variable can be approximated by the relationship $x_{k+1} = x_k + \Delta T \cdot \dot{x_k}$. Thus, according to the model above, we finally have the following discretized dynamic relationship : $x_{k+1} = x_k + \Delta T \cdot f(x_k, u_k, w_k)$

3.4 Non-linear Programming

$$
\begin{aligned}
\underset{X,U}{\text{minimize}} \quad & \sum_{k=0}^{N-1} (X_k - X_f)^\top Q (X_k - X_f) + U_k^\top R U_k \\
\text{subject to} \quad & X_{lb} \le X_i \le X_{ub},\ i = 0, \ldots, N \\
& U_{lb} \le U_i \le U_{ub},\ i = 0, \ldots, N-1 \\
& X_{i+1} = X_i + \Delta T \cdot f(X_i, U_i, W_i),\ i = 0, \ldots, N-1 \\
& X_0 = \mathcal{X}_0 \\
& X_N = \mathcal{X}_f \\
& X_f = \mathcal{X}_f
\end{aligned}
$$

with :

$$
f(X_k, U_k, W_k) = \begin{pmatrix} \dot{x_k} \\ \dot{y_k} \\ \dot{v_k} \\ \dot{m_k} \\ \dot{\theta_k} \end{pmatrix} = \begin{pmatrix} v_k \cdot cos(\theta_k) + w_{xk} \\ v_k \cdot sin(\theta_k) + w_{yk} \\ \frac{2T_k - C_d \rho A v_k^2}{2m_k} \\ -\eta T_k \\ \varphi_k \end{pmatrix}
$$

and

$$
U_k = \begin{pmatrix} T_k \\ \varphi_k \end{pmatrix}
$$

4 Results

The MPC was programmed in MATLAB. The nonlinear solver are very sensitive to the scale of the problem (i.e. numbers of variables). Hence, the total time horizon, time step size, numbers of time step need to be carefully selected.

T = total time horizon [h]
N = total numbers of time step [-]
dT = time step size [s]

A nonlinear solver IPOPT, which is suitable for large-scale nonlinear optimization, was used. Firstly, we focus on optimizing the flight path without considering wind data and obtain the results as below. To do the optimization, we tried several time by decreasing the whole trip time from 4 hours to 3.6 hours. We wasn't able to make the aircraft arrive at SFO in 3.5 hours. Our optimal travel time is 3.6 hours in total.

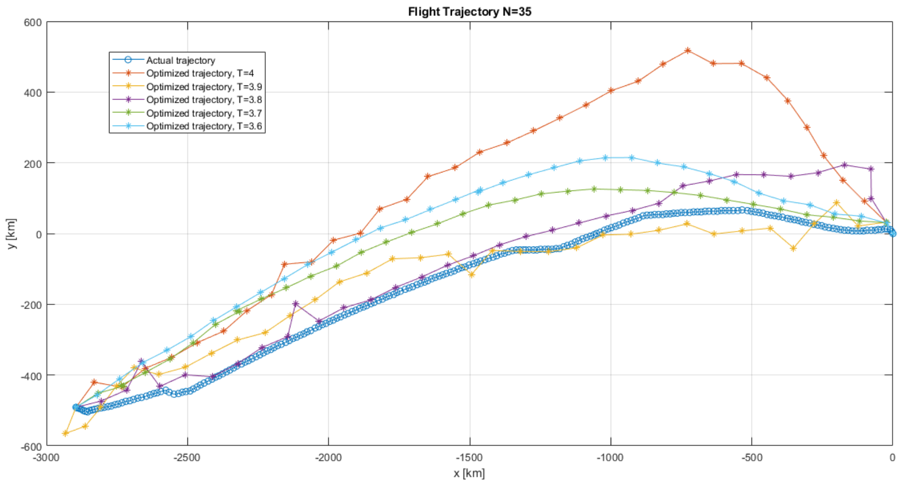

FIGURE 7 – Optimized travel time flight trajectories (no wind) with N=35

Then, based on the polynomial function we fitted for the actual wind data, we apply the function to the state matrix. And the optimal trajectory with wind data is shown as below. If we decrease the total travel time to 3.7 hours, the airplane cannot reach the destination SFO. This time, out optimal flight time is 3.8 hours.

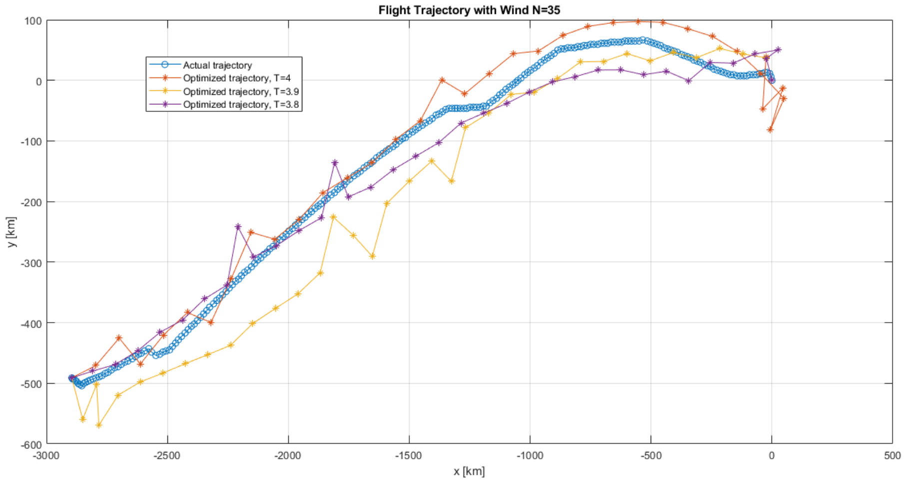

FIGURE 8 – Optimized travel time flight trajectories (with wind) with N=35

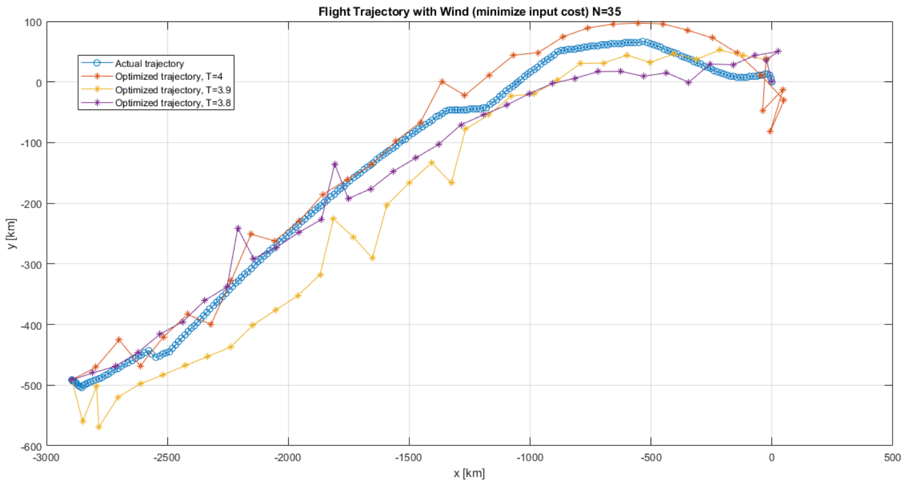

FIGURE 9 – Optimized fuel cost for the flight trajectories (with wind) with N=35

Compare the results without wind data and with wind data above, the optimal trajectories with the input of wind data are much closer to the actual flight path, which shows an acceptable result and the fitted wind function performs very well.

5 Discussion

Our project is aimed to minimize the total travel time and fuel cost to obtain the optimal flight path from Chicago O'Hare International Airport (ORD) to San Francisco International Airport (SFO) taking forecast wind data into account. And compare it to the actual flight truck from the existing sample data.

Since the optimization is a nonlinear, non-convex problem, the main challenge is to find a suitable way to solve this optimization problem. The other challenge is that in a nonlinear, non-convex optimization problem, any solution is a local optimum solution, and there is no grantee that it is a global optimum solution.

Although there exist some solvers that can solve nonlinear optimization problem, we found out that these solvers are very sensitive to the constraints (especially to the nonlinear aircraft dynamics), the scale of the problem (i.e. numbers of variables).

For example, for a given total time horizon T=5 [hour], time step =60[s], which lead to total numbers of step = 300. The solver (IPOPT) could not find a solution. However, when reduced the total numbers of step to 40 given T=4 [hour] (i.e. time step = 360[s]), the solver can find a solution.

Also, with different set up (i.e. different total time horizon, different numbers of step), the solution can change drastically. However, we can see that with certain set up, the optimized path is very similar to the actual path. For a travel time focus optimization, the path is reasonable, since not considering fuel cost, the path associate with the minimum travel time should be a straight line on the surface, which is an arc on a x-y plane.

While taking wind into consideration, the path is even much closer to the actual flight, which indicates that the aircraft model is reasonable and the aircraft performance set up is very close to reality.

There are also issues in the result. As mentioned before, the solver is very sensitive to the problem setup. For this optimization problem, the IPOPT solver will only give a unique solution, meaning that the solution is independent to the cost matrix (i.e. minimum travel time path and minimum fuel cost path are the same).

The reason might due to nonlinear programming and the defect of the nonlinear solver. Since the behavior of the nonlinear problem is very complicated and hard to handle. And how to handle the nonlinear problem and make a robust MPC model for this aircraft model should be considered in the future.

6 Summary

It is possible to use MPC method to do a flight path optimization with a nonlinear model. However, due to non-linearity of the problem and the defect of the nonlinear solver, the total number of variables needs to be carefully selected in order to obtain a reasonable result. Hence, the large scale nonlinear problem was reduced to a smaller scale by increase the time step to avoid the solver issue. And this can lead to a low resolution result, which is a drawback of the MPC method.

However, comparing the result and the actual flight trajectories and the, the solution is reasonable, which indicate that MPC method still have the potential to solve the flight path optimization problem. To obtain a more robust result, further improvement needs to be done in handling the non-linearity of the optimization problem.

7 References

[1] Simone Garatti ; Maria Prandini, Luca Deori. "A MODEL PREDICTIVE CONTROL APPROACH TO AIRCRAFT MOTION CONTROL".

[2] Majumder, Shibarchi, and Mani Shankar Prasad. "Flight Path Optimization Based on Obstacles and Weather Updates." 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), 2016. doi :10.1109/spin.2016.7566732.

[3] Khardi, Salah, and Lina Abdallah. "Optimization Approaches of Aircraft Flight Path Reducing Noise : Comparison of Modeling Methods." Applied Acoustics 73, no. 4 (2012) : 291-301. doi :10.1016/j.apacoust.2011.06.012.

[4] Yu, Bin, Zhen Guo, Sobhan Asian, Huaizhu Wang, and Gang Chen. "Flight Delay Prediction for Commercial Air Transport : A Deep Learning Approach." Transportation Research Part E : Logistics and Transportation Review125 (2019) : 203-21. doi :10.1016/j.tre.2019.03.013.

8 Appendixe

```
Cd= 0.04;            %Drag force coefficient[-]
rho= 0.38;           %[kg/m3] air density
eta= 0.01667*10^-3;  %[kg/(N*s)]
S= 122.6;            %[m2]...Aircraft wing S

%ORD=[272.093, 41.9742]; %start coord
%SFO=[237.617, 37.6211]; %final coord
ORD=[0, 0]; %start coord
SFO=[-2895204.87, -491111.25]; %final coord
```

Wind approximated funciton coefficient

```
a1=5.404039761756626e-12;
a2=-7.525095226657606e-09;
a3=-1.0097962636800737e-05;
a4=0.0018023200165759225;
a5=0.30541919780328985;
a6=1.0706181973188272e-12;
a7=8.130657646668963e-09;
a8=1.9566392595939457e-05;
a9=0.013598711210671936;
a10=0.3054111118706383;
a11=-4.4925592769414514e-13;
a12=1.3721355294688644e-12;
a13=-1.9705270758495876e-12;

b1=6.505580005194991e-12;
b2=-2.3582120860024345e-10;
b3=-2.0098332293991807e-06;
b4=-8.207601052061964e-06;
b5=6.216049027788961;
b6=-2.184365129705414e-12;
b7=-1.574583855440115e-08;
b8=-1.7909052661551478e-05;
b9=0.0358673394470828;
b10=6.216049027919514;
```

Simulation Parameters

```
    N=35;
    dT=4*3600/N;          %[s] time step
```

Dynamic model/Linearized model

```
%Aircraft dynamics
%x=[x;y;v;m;theta]      u=[T;yaw];

dyn=@(x,u) [x(1)+dT*(x(3)*cos(x(5))+a1*(x(2)^4/1000^4)+...
    a2*(x(2)^3/1000^3)+a3*(x(2)^2/1000^2)+a4*(x(2)/1000)+a5+...
    a6*(x(1)^4/1000^4)+a7*(x(1)^3/1000^3)+a8*(x(1)^2/1000^2)+...
    a9*(x(1)/1000)+a10+a11*(x(1)^3/1000^3)*(x(2)/1000)+...
    a12*(x(1)^2/1000^2)*(x(2)^2/1000^2)+a13*(x(1)/1000)*(x(2)^3/1000^3));
            x(2)+dT*(x(3)*sin(x(5))+b1*(x(2)^4/1000^4)+...
            b2*(x(2)^3/1000^3)+b3*(x(2)^2/1000^2)+b4*(x(2)/1000)+...
            b5+b6*(x(1)^4/1000^4)+b7*(x(1)^3/1000^3)+...
            b8*(x(1)^2/1000^2)+b9*(x(1)/1000)+b10);
            x(3)+dT*(2*u(1)-Cd*rho*S*(x(3)^2))/(2*x(4));
            x(4)-dT*eta*u(1);
            x(5)+dT*u(2)];
```

Cost Matrics

```
Q=diag([10^3,10^3,0,0,0]);    %state cost
R=diag([0,0]);              %input cost
```

Initial/Final States

```
%x=[x; y; v; m; theta]      u=[T; yaw];
x0=[ORD'; 100; 78000; deg2rad(127)];      %initial states
xf=[SFO'; 0; 0; 0];   %final states
```

State/Input Constraints

```
xlb=[-inf; -inf; 0; 37200; -inf];
xub=[inf; inf; 250; 78000; inf];
ulb=[0; -2.5*pi/180];
uub=[120000*2; 2.5*pi/180];
```

Path Planning

```
    tic
    [xopt,uopt,flag]=CFTOC2(dyn,Q,R,x0,xf,xlb,xub,ulb,uub,dT,N);
    toc

function [xopt,uopt,flag] = CFTOC2(dyn,Q,R,x0,xf,xlb,xub,ulb,uub,dT,N)
disp('initialize...')
%states:[x ;y ;v ;m ;theta]
```

```
%input:[T ;yaw]
x = sdpvar(5,N+1);
u = sdpvar(2,N);

%cost
cost=[];
%initial states constriant:
constraint = [x(:,1)==x0,x(1:3,end)==xf(1:3)];  %initial/final state

for k=1:N
    %states dynamics
    constraint = constraint + [x(:,k+1)==dyn(x(:,k),u(:,k))];

    %states constraints
    constraint = constraint + [xlb <= x(:,k)<= xub];
    constraint = constraint + [-0.6*dT <= (x(3,k+1)-x(3,k))<= 0.6*dT];

    %input constraints
    constraint = constraint + [ulb <= u(:,k)<= uub];
    %cost
    cost=cost+(x(:,k)-xf)'*Q*(x(:,k)-xf)+u(:,k)'*R*u(:,k);
end

%slove optimization problem
disp('optimizing...')
%choose solver
options=sdpsettings('solver','ipopt','verbose',0,'showprogress',1);
%options.ipopt.check_derivatives_for_naninf='yes';
result=optimize(constraint,cost,options);

%check feasibility
flag=result.problem;
disp(yalmiperror(flag));
if flag==0
    %solution found
    xopt=double(x);    uopt=double(u);
else
    %no feasible solution
    xopt=[];    uopt=[];
end
end
```