

Optimal control of a combat aircraft with thrust vectoring during take-off - 10th EUCASS - 9th CEAS

Juan Manzanero, PhD^{*†}, Diego Lodares^{*}, Miguel Sabaris Boullosa^{*}, Francisco Asensio Nieto^{*}

^{*}Airbus Defence and Space SAU

Airbus Defence And Space: Avda. de John Lennon, s/n, 28906 Getafe, Spain

juan.manzanero@airbus.com

[†]Corresponding author

Abstract

In this work we explore the benefits of thrust vectoring to reduce the take-off distance of a combat aircraft. To assess the aircraft's pure performance and handling qualities, the optimal control method is proposed herein to avoid the need to design preliminary Flight Control Laws (FCL) to simulate the aircraft. This becomes even more relevant for combat aircraft, usually open-loop longitudinally unstable, where a sub-optimal FCL might not fully exploit the capabilities of the plant and lead to misleading design decisions. We conclude that without thrust vectoring, the aircraft is limited by its capability to rotate early on and build up enough angle of attack, whereas with thrust vectoring it is limited by the maximum rotation angle due to tail strikes. Overall, we find that the take-off distance is reduced by 38%. The optimal control method helps to understand how thrust vectoring can be used efficiently, to introduce a pitch up moment that induces the rotation, followed by a pitch down moment that stops it and prevents impacts against the tarmac.

1. Introduction

The early design stages of a new aircraft are a multidisciplinary challenge in which a large number of configurations of different nature are compared.

To study the capabilities of a particular design, the optimal control method is a powerful tool. This method solves the vehicle's control time-histories that minimize a given functional. In this work, we study the aircraft's dynamics as it takes-off and climbs, but many other targets are possible: minimizing fuel consumption, performing exercise manoeuvres, following a given path, etc. The method has been widely used in the context of race-car simulation,¹² launch and reentry flight,^{2,9,11} robotics,^{2,14} orbital mechanics,² and tumoral studies.⁷

Amongst all the optimal control methods, we have implemented a collocation solver that uses the trapezoidal rule.⁵ Optimal control methods have been traditionally divided into direct and indirect methods. Their main difference is that direct methods first discretize, then optimize the problem, whereas indirect methods first optimize, and then discretize the problem. As a result, direct methods require to solve a nonlinear programming (NLP) problem using an optimization toolbox (e.g. Ipopt¹⁵ or SNOPT⁴), whereas indirect methods require the user to first write the maximum principle equations of their dynamic system,³ and then solve the resulting nonlinear equations. We have chosen the direct collocation method since it results in a much simpler implementation, scallable, easier to parallelize, and with the possibility to derive a generic optimal control toolbox that applies to any physical system (e.g. GPOPS-II¹¹).

This technique allows us to study the effect of thrust vectoring on take-off. The main benefit of the optimal control method is that it takes the handling of the aircraft out of the equation, since that responsibility falls on the optimization process, while retaining all the dynamic effects of the simulation. Thus, the controls are optimal in the sense that they are *the best possible* such that the aircraft best executes the desired manoeuvre, in this case, to maximize the take-off final altitude. Other methods include the development of a manned simulator, but this requires the design of *ad hoc* Flight Control Laws, which is not immediate for highly manoeuvrable combat aircraft; or the study of the pitch onset produced by thrust vectoring employing simplified models. The latter, however, does not consider all the dynamic effects and may lead to inaccurate conclusions.

The rest of this work is organized as follows: in Sec. 2 we present the 3dof pitch aircraft model, in Sec. 3 we describe the collocation method and its application to the take-off problem, in Sec. 4 we show and discuss the simulation results, and in Sec. 5 we draw the final conclusions.

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

2. Aircraft model

In this work we consider the equations that describe the pitch motion of an aircraft. The velocity of the aircraft is $\vec{v} = (v_x, 0, v_z)$, and its angular velocity is $\vec{\omega} = (0, q, 0)$, where the pitch rate q is the time derivative of the pitch angle θ .

The motion of the aircraft is described by the body-axes linear and angular momentum equations,

$$m\dot{v}_x + mqv_z = f_{x,aero} + f_{x,engine} + f_{x,lg} - mg \sin \theta, \quad (1)$$

$$m\dot{v}_z - mqv_x = f_{z,aero} + f_{z,engine} + f_{z,lg} + mg \cos \theta, \quad (2)$$

$$I_{yy}\dot{q} = m_{y,aero} + m_{y,engine} + m_{y,lg}, \quad (3)$$

where m and I_{yy} are the aircraft's mass and pitch inertia, \vec{f}_{aero} are the aerodynamic forces, \vec{f}_{engine} are the propulsive forces, \vec{f}_{lg} are the landing gear forces, and $g = 9.81 \text{ m/s}^2$ is the gravity acceleration.

2.1 Aerodynamic model

The aircraft model is a simplistic dataset that depends on the angle of attack $\alpha = \text{atan}(v_z/v_x)$, and the HTP δ_{htp} and flap δ_{flaps} deflections,

$$f_{x,aero} = \frac{1}{2}\rho V^2 S C_x(\alpha, \delta_{htp}, \delta_{flaps}), \quad (4)$$

$$f_{z,aero} = \frac{1}{2}\rho V^2 S C_z(\alpha, \delta_{htp}, \delta_{flaps}), \quad (5)$$

$$m_{y,aero} = \frac{1}{2}\rho V^2 S C_{my}(\alpha, \delta_{htp}, \delta_{flaps}), \quad (6)$$

An example of an aerodynamic dataset for a combat aircraft can be found in.¹⁰ No atmospheric wind was considered in the simulations presented herein.

2.2 Engine model

The engine model is also a simplified linear thrust model,

$$T = T_{\max} \delta_{pla} (1 - p_0 |\delta_{tv}|), \quad (7)$$

$$f_{x,engine} = T \cos \delta_{tv}, \quad (8)$$

$$f_{z,engine} = -T \sin \delta_{tv}, \quad (9)$$

$$m_{y,engine} = \vec{r}_{plume} \wedge \vec{f}_{engine}, \quad (10)$$

where δ_{pla} is the *power lever angle* in $[0, 1]$, δ_{tv} is the deflection angle of the thrust-vectoring-capable nozzle, and \vec{r}_{plume} is the position of the nozzle with respect to the center of gravity. Note that we have included an efficiency term on the thrust that accounts for losses due to the deflection of the nozzle.

The dynamics of the engine are also taken into account by adding a first order lag filter on the pilot command,

$$2\delta_{pla}^{\dot{}} + \delta_{pla} = \delta_{pla,cmd}. \quad (11)$$

2.3 Landing gear model

While on land, the aircraft uses the nose and main landing gear to move along the runway. From the position and orientation of the aircraft, we can compute the vertical position of the wheels' contact points, z_{nose} and z_{main} , and from its velocity and rates we can compute their derivatives, \dot{z}_{nose} and \dot{z}_{main} .

Thus, we compute the loads on each of the landing gears as those produced by the spring and damping of the suspension and tire carcass,

$$N_i = \max(0, k_i z_i + c_i \dot{z}_i) \geq 0, \quad (12)$$

where k_i is the spring stiffness and c_i is the damping constant.

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

2.4 Summary

We stack all the equations and write them as a system of first order ordinary differential equations,

$$\dot{\mathbf{q}} = \mathbf{f}(\mathbf{q}, \mathbf{u}), \quad (13)$$

where $\mathbf{q} = (z, v_x, v_z, \theta, q, \delta_{pla}) \in \mathbb{R}^{N_q=6}$ is the state vector and $\mathbf{u} = (\delta_{htp}, \delta_{flap}, \delta_{pla,cmd}, \delta_{tv}) \in \mathbb{R}^{N_u=4}$ is the control vector.

3. Optimal control

In this section we describe the optimal control methodology used throughout this work. Among the different techniques in the literature, we implement a first order collocation method known as the trapezoidal rule. This method is selected mainly due to its ease of implementation and for being plant independent, as opposed to indirect methods, which require to derive a set of mathematically intense equations from Pontryagin's principle.⁶ Ultimately, this enables one to implement an optimal control toolbox which can be applied to any dynamic system.¹¹

3.1 Dynamic equations

Since we are interested in controlling the aircraft during the take-off and climb phases, it is beneficial to use the position along the runway instead of the time as the independent variable. Thus, we rewrite the dynamic equations (13) using $\dot{x} = v_x$,

$$\frac{d\mathbf{q}}{dx} = \frac{1}{v_x} \mathbf{f}(\mathbf{q}, \mathbf{u}) - \mathbf{f}^*(\mathbf{q}, \mathbf{u}). \quad (14)$$

This is a common practice in other fields such as race-car simulation, where the vehicle must follow a curvilinear trajectory along a two-dimensional racetrack.¹²

Next, we discretise the runway into a mesh of $n + 1$ points, $x = (x_0, x_1, \dots, x_n)$, where $x_0 = 0$ and $x_n = L$ is the length of the runway. Then, we define $\mathbf{q}_i = \mathbf{q}(x_i)$ and $\mathbf{u}_i = \mathbf{u}(x_i)$ as the approximations of the states and controls at each point of the mesh.

We use the trapezoidal rule to numerically approximate the integral of the dynamic equations in the n elements $e_i = [x_i, x_{i+1}]$,

$$\mathbf{q}_{i+1} - \mathbf{q}_i = \frac{1}{2} \Delta x_i (\mathbf{f}_i^* + \mathbf{f}_{i+1}^*), \quad i = 0, \dots, n, \quad (15)$$

where $\mathbf{f}_i^* = \mathbf{f}^*(\mathbf{q}_i, \mathbf{u}_i)$, and the mesh spacing (which is allowed to spatially vary) is $\Delta x_i = x_{i+1} - x_i$. Note that the resulting scheme is implicit.

Usually, when simulating a dynamic system one knows the positions of the controls at each point u_i (either imposed if the plant is open loop, or derived from the control laws) and can sequentially solve the equations from the initial to the final positions. However, in the optimal control method, the controls are unknown and they result from the solution of an optimization problem. Thus, we recast (15) as a set of constraints for the optimization problem,

$$\mathbf{c}_i = \mathbf{q}_{i+1} - \mathbf{q}_i - \frac{1}{2} \Delta x_i (\mathbf{f}_i^* + \mathbf{f}_{i+1}^*) = 0, \quad (16)$$

note that $n \times N_q$ constraints are required to approximate the full aircraft dynamics in the complete mesh.

3.2 Fitness function

The dynamic equations impose N_q equations per computational point, but the N_u degrees of freedom of the control variables \mathbf{u}_i remain free. Thus, they are computed such that they maximize a given fitness function. In the take-off case, we maximise the altitude of the aircraft at the mesh's endpoint,

$$f_z = z(t_f) = \int_0^{t_f} \dot{z} dt = \int_0^L \frac{\dot{z}}{v_x} dx \approx \sum_{i=0}^n \frac{\Delta x_i}{2} \left(\left(\frac{\dot{z}}{v_x} \right)_i + \left(\frac{\dot{z}}{v_x} \right)_{i+1} \right). \quad (17)$$

Usually, the solution to optimal control problems yield unpractical solutions due to very high frequency components in the control variables. A common practice to limit these oscillations is to augment the fitness function with a term that penalises the control rates,

$$f_u = \sum_{c=1}^{N_u} \int_0^L \epsilon_c \dot{u}_c^2 dx \approx \sum_{i=0}^n \frac{\Delta x_i}{2} \sum_{c=1}^{N_u} \epsilon_c (\dot{u}_{c,i}^2 + \dot{u}_{c,i+1}^2), \quad (18)$$

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

Table 1: Variable bounds for the states and control positions and rates

State variables	z [m]	v_x [m/s]	v_z [m/s]	q [deg/s]	δ_{pla} [-]
	$(-\infty, -2.2]$	$[0.6, 200.0]$	$[-100.0, 100.0]$	$[-50, 50]$	$[0, 1]$
Control variables	δ_{htp} [deg]	δ_{flap} [deg]	$\delta_{\text{pla,cmd}}$ [-]	δ_{tv} [deg]	
position	$[-25, 25]$	$[-30, 30]$	$[0, 1]$	$[-10, 10]$	
rate	$[-70, 70]$	$(-70, 70)$	$[-100, 100]$	$[-40, 40]$	

where ϵ_c are N_u dissipative constants that can be tuned: small values lead to more oscillatory controls, while larger values lead to slower ones.

Thus, the fitness function to be minimized is

$$f = \sum_{i=0}^n \frac{\Delta x_i}{2} \left(\left(\frac{\dot{z}}{v_x} \right)_i + \left(\frac{\dot{z}}{v_x} \right)_{i+1} \right) + \sum_{i=0}^n \frac{\Delta x_i}{2} \sum_{c=0}^{N_u} \epsilon_c \left(\dot{u}_{c,i}^2 + \dot{u}_{c,i+1}^2 \right). \quad (19)$$

3.3 Time derivatives of the controls

Since we have involved the time derivatives of the controls in the fitness functions, we need $n \times N_u$ extra variables (the time derivatives of the controls along the mesh, $\dot{\mathbf{u}}_i$) and constraints in the problem to compute them,

$$\mathbf{u}_{i+1} - \mathbf{u}_i - \frac{\Delta x_i}{2} \left(\left(\frac{\dot{\mathbf{u}}}{v_x} \right)_i + \left(\frac{\dot{\mathbf{u}}}{v_x} \right)_{i+1} \right) = 0, \quad i = 0, \dots, n. \quad (20)$$

Including the time derivatives of the controls has an extra benefit, as it allows us to bound them, introducing actuator rate limit effects in the system.

3.4 Stability constraints

We add two extra constraints that represent restrictions to the optimal control solution on all mesh points:

1. The aircraft's angle of attack shall remain within $0^\circ \leq \alpha \leq 30^\circ$
2. The aircraft's rearmost point must always maintain a minimum clearance with the ground, $z_{\text{tail}} \leq -0.5$ m.

3.5 Variable bounds

As part of the optimization problem, we have specified bounds for the search space in all of the unknowns (state, controls, and control derivatives). The full set of bounds is listed in Table 1.

3.6 Summary: nonlinear programming (NLP)

The optimal control problem is casted in its NLP form,

$$\min f(\mathbf{x}), \text{ s.t.} \quad (21)$$

$$\mathbf{c}(\mathbf{x}) = 0 \quad (22)$$

$$\mathbf{d}_L \leq \mathbf{d}(\mathbf{x}) \leq \mathbf{d}_U \quad (23)$$

$$\mathbf{x}_L \leq \mathbf{x} \leq \mathbf{x}_U \quad (24)$$

where $\mathbf{x} = (\mathbf{q}_1, \mathbf{u}_1, \dot{\mathbf{u}}_1, \dots, \mathbf{q}_{n+1}, \mathbf{u}_{n+1}, \dot{\mathbf{u}}_{n+1}) \in \mathbb{R}^{n \times (N_q + 2N_u)}$, $\mathbf{c} \in \mathbb{R}^{n \times (N_q + N_u)}$ represents the concatenation of all the dynamic (15) and control derivatives (20) equations, $\mathbf{d} \in \mathbb{R}^{2n}$ represents the two stability constraints per point, and \mathbf{x}_L and \mathbf{x}_U are the variable bounds.

The NLP is solved using Ipopt,¹⁵ an open source package for large-scale optimization which implements the interior point method. Typically, this software is able to find solutions to the optimal control problem in less than 500 iterations.

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

3.7 Automatic differentiation (AD)

The interface of the NLP to Ipopt requires the evaluation of the gradient and Hessian of the fitness function and constraint function. Two common approaches are

1. the numerical evaluation of derivatives using finite differences,⁸
2. the exact calculation of derivatives (by hand or symbolic math toolboxes),
3. the evaluation of derivatives using automatic differentiation.¹³

The first approach is numerically expensive and inaccurate, and the second approach is tedious and error-prone. In this work, we use automatic differentiation to compute the first and second order derivatives. Automatic differentiation evaluates derivatives via operator overloading of the equations in the software. By *recording a tape* of the computations, the software is able to reproduce the steps to compute their first and second order derivatives. To compute the derivatives required to interface our NLP, we have used the open-source software CppAD.¹

Finally, some of the functions used in the model need to be sufficiently smooth so that their derivatives are well defined. Thus, we use the following approximations for the absolute value and max functions,

$$\max(0, x) \approx \frac{1}{2}(x + \sqrt{x^2 + \varepsilon^2}), \quad |x| \approx x \tanh(x/\varepsilon), \quad (25)$$

with ε being a sufficiently small constant number.

4. Simulations

In this section we show the capabilities of the optimal control solver to simulate a aircraft take-off, and the improvement in the take-off distance that the thrust vectoring attains. In all simulations, the thrust power lever angle is fixed to maximum reheat, i.e., not part of the optimal control solution. The simulation takes into account the engine dynamics from idle to maximum power by means of the engine dynamic model (11).

4.1 Take-off without thrust vectoring

We start by fixing the engine nozzle to its nominal position to study the aircraft without thrust vectoring. Results are shown in Fig. 1. We have solved the optimal control problem in a runway of 600 m discretized into 801 points, for which Ipopt took 250 iterations. The simulation includes two cases: without flaps (blue), and with the flaps fully deflected (red). In this simulation, only the HTP is used to control the aircraft. In the plots, the circle represents the instant at which the nose gear lifts and the aircraft starts its rotation, and the cross represents the instant when the main gear lifts and the aircraft takes off.

We see that the aircraft's take off is mostly limited by its capability to effectively rotate and build up a large angle of attack, necessary to lift the aircraft. The HTP deflects to its maximum pitch-up position as the speed increases. When the aircraft reaches a certain speed, the nose gear lifts and the aircraft starts the rotation, which increases the angle of attack and the lift of the aircraft. Then, when a sufficient angle of attack is reached, the HTP starts to reduce the pitch up since the aircraft is highly unstable longitudinally, and the clearance point constraint comes into play. Finally, the main gear separates from the ground and the aircraft lifts off. The HTP is used from then on to stabilise the aircraft, but no longer saturated in position nor rate.

We confirm that the aircraft with flaps is able to reduce the take-off distance, but not much, since the aircraft is limited by its capability to rotate and the control power of the HTP rather than the capability of the wings to generate lift. Thus, the use of thrust vectoring is expected to be very beneficial to reduce the take-off distance.

4.2 Take-off with thrust vectoring

We recompute the solutions to the optimal control problem, this time, using both the HTP and thrust vectoring to control the aircraft's attitude. Fig. 2 summarizes the results. In this case, we consider the aircraft with the flaps fully deflected to max. lift configuration. In blue, we have represented the solution without thrust vectoring (nozzle at 0 deg), and in red the solution with thrust vectoring in the loop. It is evident that thrust vectoring makes a difference since the take-off distance (the abscissa of the cross marker in the plot) is reduced from 360m to 225m (38%).

In the simulation with thrust vectoring, the sum of the HTP and the thrust vectoring can produce the required amount of torque to rotate the aircraft at any speed, but they must wait until the speed is large enough to produce the

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

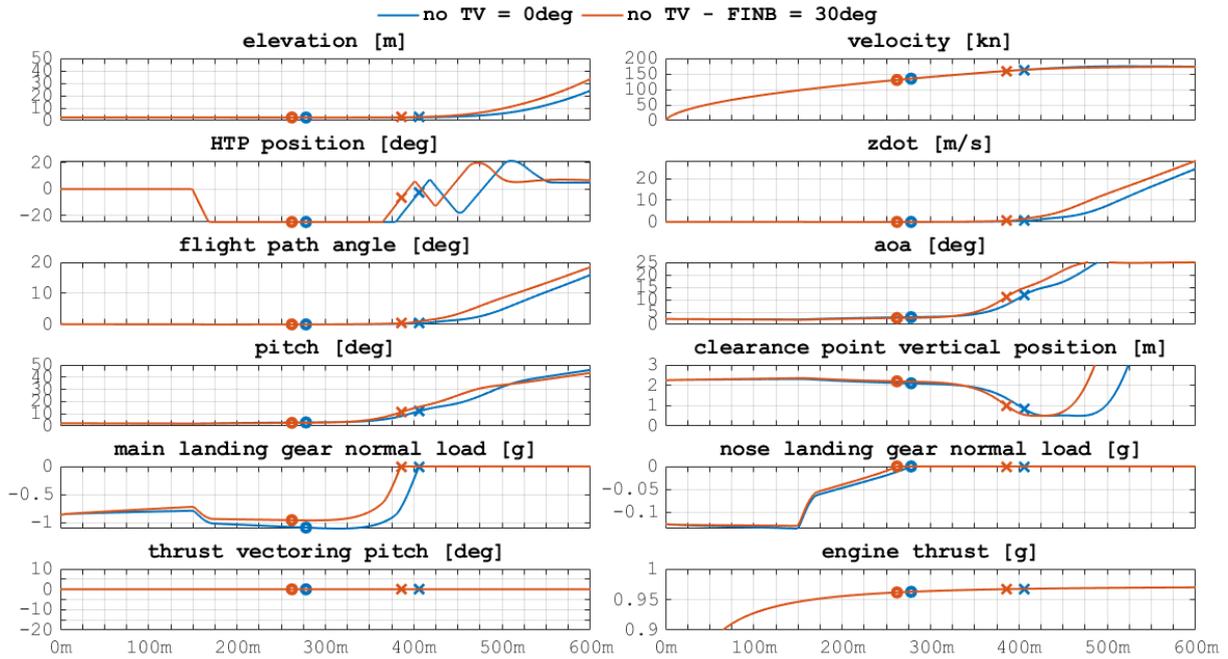


Figure 1: Optimal control simulation without thrust vectoring. We compare the aircraft's performance without (blue) and with (red) fully deflected flaps.

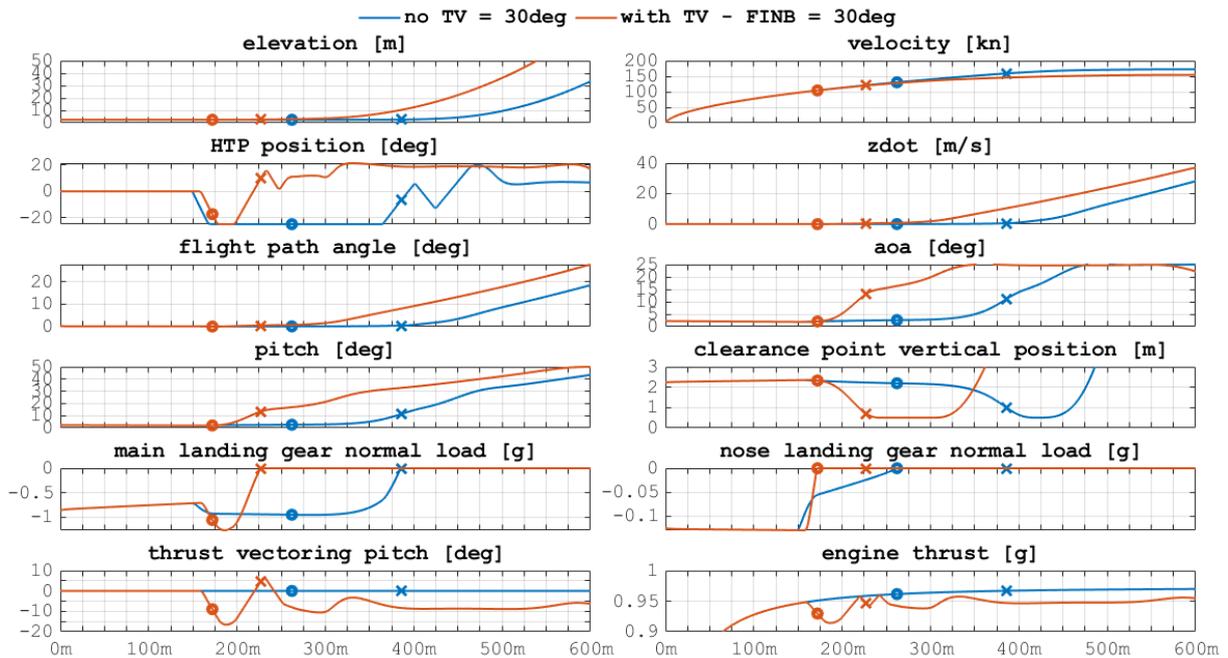


Figure 2: Optimal control simulation with thrust vectoring. We compare the aircraft's performance without (blue) and with (red) fully deflected flaps.

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

required lift once the aircraft reaches the maximum angle of attack. Recall that the maximum angle of attack on ground is limited by the minimum tail-ground clearance condition.

Thus, after the aircraft has started to rotate, the HTP and the thrust vectoring must reduce the pitch up to prevent the aircraft's tail hitting the ground. Consequently, as opposed to the case without thrust vectoring, where the take-off distance was limited by the capability of the aircraft to rotate, in the case with thrust vectoring the take-off distance is limited by the minimum speed at which the aircraft produces enough lift at the maximum angle of attack. We see that the aircraft with thrust vectoring accelerates less than the aircraft with a fixed nozzle, due to the reduction in engine efficiency caused by the nozzle deflection.

5. Conclusions

In this work we have studied the benefits of a thrust vectoring control system to reduce the take-off distance of a combat aircraft. To do so, we have studied solutions to an optimal control problem applied to the vehicle's longitudinal dynamics, as opposed to other traditional techniques which require the development of *ad hoc* Flight Control Laws to stabilise aircraft that may have relaxed open-loop stability characteristics, or the study of pitch onset and acceleration metrics employing reduced models that don't make use of the available aerodynamic dataset in its entirety.

This methodology was applied to compare several optimal take-off manoeuvres with and without thrust vectoring, to conclude that the take-off distance is reduced by 38% in the former case. We have seen that deflecting the flaps accelerates the take-off but the gain is not remarkable, since the aircraft is limited by its capability to rotate early on due to the control power of its HTP surface. The thrust vectoring control can overcome this difficulty as the aircraft can rotate and reach its maximum angle of attack earlier, thus decreasing the take-off distance.

References

- [1] BM Bell. Automatic differentiation software cppad.; 2010. URL <http://www.coinor.org/CppAD>.
- [2] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010.
- [3] Thomas J Böhme, Benjamin Frank, Thomas J Böhme, and Benjamin Frank. Indirect methods for optimal control. *Hybrid systems, optimal control and hybrid vehicles: Theory, methods and applications*, pages 215–231, 2017.
- [4] Philip E Gill, Walter Murray, and Michael A Saunders. Snopt: An sqp algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.
- [5] Matthew Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, 2017.
- [6] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.
- [7] Urszula Ledzewicz and Heinz Schättler. Analysis of optimal controls for a mathematical model of tumour anti-angiogenesis. *Optimal Control Applications and Methods*, 29(1):41–57, 2008.
- [8] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.
- [9] J Meditch. On the problem of optimal thrust programming for a lunar soft landing. *IEEE Transactions on Automatic Control*, 9(4):477–484, 1964.
- [10] Luat T Nguyen. *Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability*, volume 12854. National Aeronautics and Space Administration, 1979.
- [11] Michael A Patterson and Anil V Rao. Gpops-ii: A matlab software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming. *ACM Transactions on Mathematical Software (TOMS)*, 41(1):1–37, 2014.
- [12] Giacomo Perantoni and David JN Limebeer. Optimal control for a formula one car with variable parameters. *Vehicle System Dynamics*, 52(5):653–678, 2014.
- [13] Louis B Rall and George F Corliss. An introduction to automatic differentiation. *Computational Differentiation: Techniques, Applications, and Tools*, 89:1–18, 1996.

OPTIMAL CONTROL OF A COMBAT AIRCRAFT WITH THRUST VECTORING DURING TAKE-OFF

- [14] Yoshiyuki Sakawa. Trajectory planning of a free-flying robot by using the optimal control. *Optimal Control Applications and Methods*, 20(5):235–248, 1999.
- [15] Andreas Wächter and Lorenz T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical programming*, 106:25–57, 2006.