JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 34, No. 1, January–February 2011

Trajectory Generation for Aircraft Avoidance Maneuvers Using Online Optimization

Rushen B. Patel* and Paul J. Goulart†
Imperial College London, London, England SW7 2AZ, United Kingdom

DOI: 10.2514/1.49518

This paper presents an aircraft trajectory generation scheme for use as a part of an autonomous counterhijack control system for civilian aircraft. In this scheme, buildings and other critical infrastructure and landmarks are modeled as constraint objects to be avoided in the aircraft flight path. A three degree-of-freedom nonlinear model and a direct multiple shooting method are employed to generate finite horizon avoidance trajectories for this system. A novel method for modeling nondifferentiable constraint obstacles (e.g., polytopes) is developed, employing dualization of the state exclusion regions to maintain continuity, thus allowing the use of a gradient-based optimization algorithm. The dualization method is further extended to construct positively invariant target sets that ensure the terminal state of each of the finite horizon trajectories generated remains feasible when extended over an infinite horizon. These conditions, when combined with a warm-start method based on shift initialization of prior solutions, ensure the optimization is initialized close to a feasible solution for the nonconvex problem. The results show, via a selection of simulation cases and for various classes of constraint objects, that the proposed strategy produces feasible avoidance trajectories with computation times viable for real-time applications.

Nomenclature

$\mathcal{B}_p$	=	p-norm unit ball
C°	=	polar set of C
C_j	=	state exclusion set (prohibited volume)
D	=	drag force
d^s	=	s search direction in optimization
d^z	=	z search direction in optimization
d^λ	=	λ search direction in optimization
$f(\cdot)$	=	system dynamics
$f^*(\cdot)$	=	Fenchel conjugate
$J(\cdot)$	=	objective function
$J_N(\cdot)$	=	objective function with horizon N
L	=	lift force
M	=	number of state exclusion sets
m	=	aircraft mass
N	=	number of multiple shooting time steps
P	=	terminal cost matrix (state)
$\mathcal{P}(\cdot)$	=	optimal control problem
$p(x)$	=	position vector at a given state
Q	=	stage cost matrix (state)
R	=	stage cost matrix (input)
r_j	=	barrier parameter
s	=	Lagrange multiplier for bound constraint
T	=	engine thrust
t_f	=	time horizon for optimization
t_p	=	throttle setting
$\mathbf{u}$	=	control sequence
u_i	=	input vector at time step i
V	=	true airspeed
$v(x)$	=	velocity vector at a given state
$\mathcal{X}_\mathcal{N}$	=	set of feasible initial conditions
$\mathcal{X}_f$	=	target set
$\mathbf{x}$	=	state sequence
$x^\top y$	=	x transpose multiplied by y
x_c	=	x-position of aircraft in Cartesian coordinates
x_i	=	state vector at time step i
y_c	=	y-position of aircraft in Cartesian coordinates
z_c	=	z-position of aircraft in Cartesian coordinates
α	=	angle of attack
γ	=	flight-path angle
$\gamma_C(\cdot)$	=	gauge function of C
η	=	dual variable for target set condition
κ_f	=	terminal control law
λ	=	Lagrange multiplier for equality constraints
μ	=	roll angle
$\sigma_C(\cdot)$	=	support function of C
$\phi(\cdot)$	=	propagation of system dynamics
χ	=	heading angle
$\|(\cdot)\|_Q$	=	weighted 2-norm

I. Introduction

SINCE 11 September 2001, a number of measures have been implemented to prevent further aircraft hijackings. These include additional safety measures on the ground, improved cockpit security and the presence of air marshals onboard aircraft.[‡] However, no system has yet been implemented to prevent a similar scenario occurring in the event that a malicious pilot has gained control of an aircraft. This paper develops the trajectory generation part of a system that prevents aircraft from being flown into buildings, or other strategic targets such as power stations, via direct override of the aircraft controls. These targets are encapsulated in predetermined virtual no-go zones described as prohibited volumes (PVs) throughout the paper. The system analyzes the trajectory of the aircraft in real-time and recognizes when this trajectory could potentially penetrate a prohibited volume. Once the system has recognized a flight path as threatening, and after audio, visual and/or haptic warnings to the pilot, the system will override the pilot input and automatically fly an avoidance trajectory (see Fig. 1). The results presented in this paper focus on the avoidance trajectory generation element, i.e., the outer-loop controller, of the overall system and algorithmic measures to ensure a safe trajectory is determined.

Assuming a nonhostile pilot, Kuchar and Yang produced a comprehensive review of current methods for conflict detection and

Received 21 February 2010; revision received 10 October 2010; accepted for publication 14 October 2010.

*Ph.D. Student, Department of Aeronautics; r.patel07@imperial.ac.uk. Student Member AIAA.

†Lecturer, Department of Aeronautics; p.goulart@imperial.ac.uk. Member AIAA.

‡107th Congress. Aviation and Transportation Security Act, November 2001.

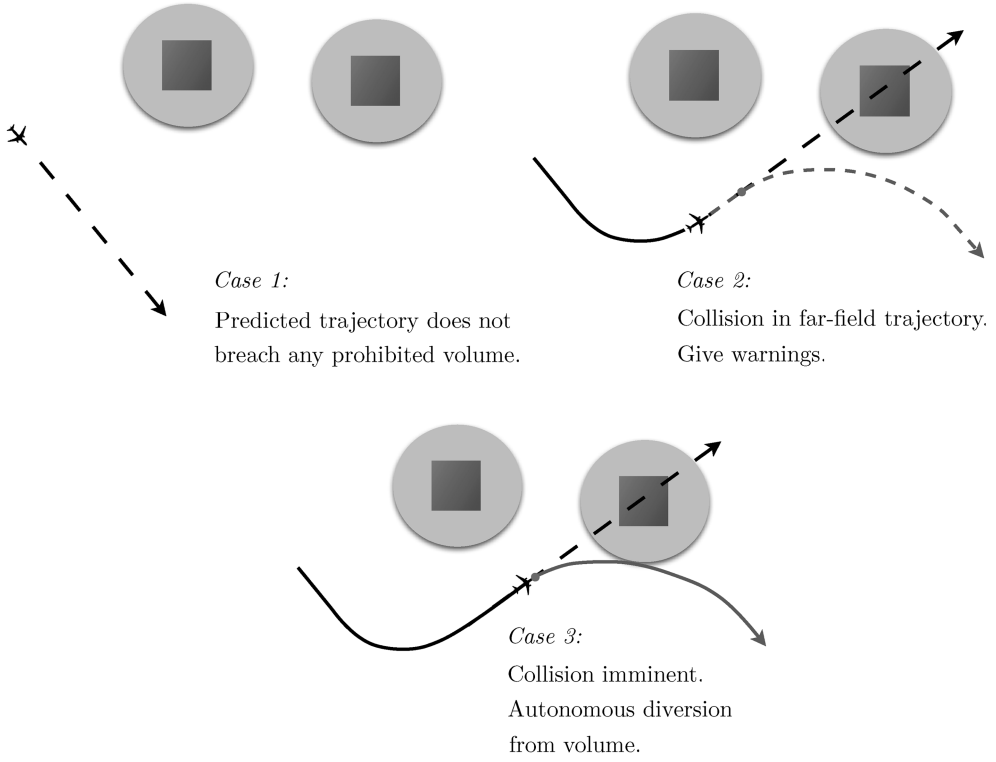


Fig. 1 Counterhijack system.

resolution (CDR) systems [1]. A CDR system should predict when a conflict will occur in the future, communicate the detected conflict to a human operator and, in some cases, assist in the resolution of the conflict situation. A number of the models reviewed by Kuchar and Yang [1] use optimization approaches which combine kinematic aircraft models with a set of cost functionals that penalize unfavorable behavior. An optimal resolution strategy is then determined by solving for the trajectories with the lowest cost. The Traffic Collision Avoidance System (TCAS), for example, searches through a set of potential climb or descent maneuvers and selects the least aggressive maneuver that still provides adequate protection [2]. For CDR systems where the pilot is expected to carry out the corrective action, economic costs or operator workload are often incorporated as design considerations. For the system proposed here, the cost functional to be minimized will depend on the type of trajectory the aircraft should follow and the desired terminal state for the maneuver. To satisfy the specifications of the system, a receding horizon approach is used similar to model predictive control (MPC) [3], with some notable differences that are highlighted in the paper.

Sampling-based trajectory planning methods have been investigated in path planning problems for autonomous vehicles. Involving a search over the entire state space, these methods include gradient descent over a navigation function [4] and rapidly expanding random tree (RRT) algorithms [5–7]. Frazzoli et al. extend the RRT method to include system dynamics by using a finite state motion model [8]. Dever et al. suggested the use of parameterized maneuver classes for nonlinear trajectory generation where online trajectories are generated through interpolation from predetermined maneuver classes [9]. Although model constraints are incorporated, direct inclusion of obstacles was not considered. These methods have been demonstrated as effective and computationally efficient for online motion planning, however, in certain cases convergence can be slow and the trajectories generated far from optimal.

Path planning with MPC techniques for trajectory generation has received a great deal of attention, particularly for unmanned aircraft [10,11]. Keviczky and Balas compared MPC techniques for single axis control and found scheduled MPC provided an attractive alternative to full nonlinear control [12]. Kuwata et al. investigated robust MPC using constraint tightening to design trajectories for an aerial vehicle operating in an environment with disturbances and constraints such as no-fly zones [13]. The computations, performed online, demonstrated feasibility, but only for a simplified vehicle model operating in a 2-D environment. A closed form solution of the nonlinear aircraft model using Taylor series expansions was investigated by Slegers et al. but the inclusion of constraints using this method would prove difficult [14]. Another approach explored by Cowling et al. is to use the differential flatness property of the equations of motion to express the state and control vectors as a function of the output vector [15]. Trajectories are parameterized using polynomials and the inverse dynamics used to determine states and controls at each discretized time point. Optimization then takes place over the output space allowing constraints such as obstacles to be dealt with more readily, but in turn, this results in more difficulty incorporating input constraints. Lee proposed a counterhijack system known as *Soft Walls* [16] based on a game-theoretic control approach. The scheme was demonstrated using a simplified vehicle model in two dimensions, but extension to a full 3-D aircraft model would require a prohibitively large amount of computational power.

The literature demonstrates that efficient algorithms exist that are well characterized for simpler subproblems of the aircraft path planning problem. However, they are frequently too computationally expensive to be used for real-time applications and, when they are tractable, are not optimal or guaranteed to find a feasible solution if one exists. The scheme presented in this paper attempts to address these issues.

The approach to the counterhijack problem adopted in this work involves a complete control override with all computations performed onboard the aircraft. Figure 2 shows a system flow diagram for the complete system. A scheme for the outer-loop controller is developed, which generates finite horizon avoidance trajectories using a nonlinear 3 deg-of-freedom (DOF) aircraft model and direct multiple shooting optimization method. This method of trajectory generation allows the incorporation of realistic aircraft behavior including constraints into the trajectory generation algorithm. It also allows a wide class of obstacle types to be placed dynamically into the aircraft environment. It has the flexibility to be modified with relative ease for use in other areas such as control of unmanned aircraft.

The paper is an extension of the work previously presented by Patel et al. [17] and is organized as follows. Section II describes the

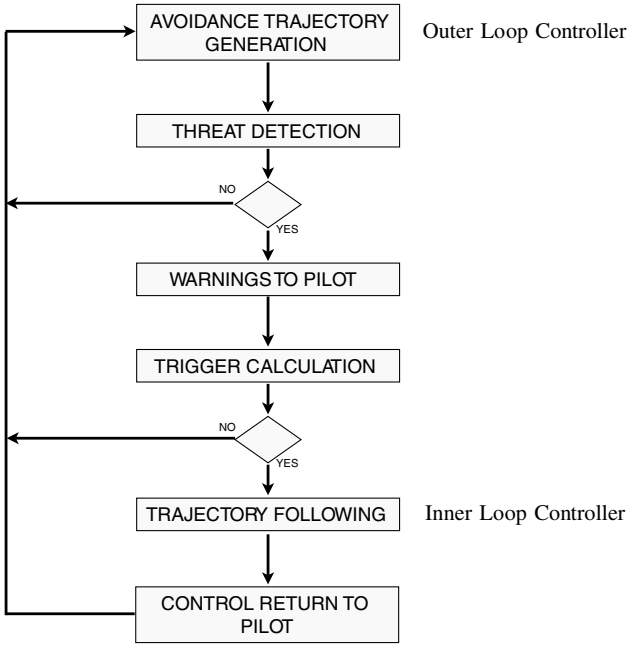


Fig. 2 System flow diagram.

aircraft model and direct multiple shooting control strategy, and gives an outline of the interior-point optimization method.

Section III describes a method for constructing positively invariant end conditions for this problem based on dualization of each obstacle constraint. This extends the original finite horizon problem to the infinite horizon. A method of algorithmic warm starting is also developed based on shift initialization of prior solutions to ensure the optimization is initialized close to a feasible solution.

In Sec. IV, the scheme is evaluated using several simulation results based on a typical civilian aircraft. Finally, Sec. V gives conclusions and outlines future work required to exploit the present research in a complete counterhijack system.

II. Trajectory Generation Algorithm

A. Aircraft Model

The aircraft dynamics are modeled using a nonlinear 3-DOF model (Fig. 3) which excludes side slip. The lower-order 3-DOF point mass model is used, as opposed to a more comprehensive 6-DOF aircraft model, because the outer-loop controller is only required to generate avoidance trajectories flyable via coordinated turns. Appropriate constraints are included in this model to ensure it

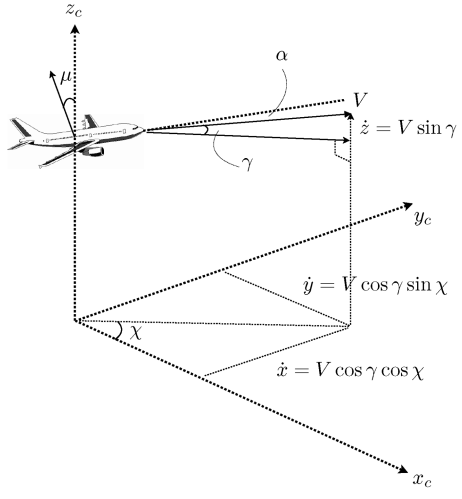


Fig. 3 Coordinate system for the 3-DOF model.

maneuvers within the envelope of a real aircraft. The states of the model are the horizontal position (x_c and y_c) and altitude z_c of the aircraft, the true airspeed V, the flight-path angle γ, and the heading angle χ. The control inputs to the model are the engine thrust T, the angle of attack α, and the roll angle μ. The state vector x is defined as $x \triangleq [x_c, y_c, z_c, \gamma, \chi, V]$ and the input vector u as $u \triangleq [\alpha, \mu, T]$. Instead of using the thrust as a direct input to the model, a more realistic approach is to use a fractional throttle as an input and derive a suitable throttle to thrust relationship ($T(t_p, z_c, V)$) where $t_p \in [0, 1]$ is the throttle setting. This is described in Sec. IV.

Using simple Newtonian dynamics, and assuming there are no disturbances due to wind, the equations of motion are:

$$\dot{x_c} = V \cos\gamma \cos\chi \tag{1a}$$

$$\dot{y_c} = V \cos\gamma \sin\chi \tag{1b}$$

$$\dot{z_c} = V \sin\gamma \tag{1c}$$

$$\dot{\gamma} = \frac{1}{mV}[(L + T\sin\alpha)\cos\mu - mg\cos\gamma] \tag{1d}$$

$$\dot{\chi} = \frac{1}{mV\cos\gamma}[(L + T\sin\alpha)\sin\mu] \tag{1e}$$

$$\dot{V} = \frac{1}{m}[(T\cos\alpha - D) - mg\sin\gamma] \tag{1f}$$

The aerodynamic forces of lift and drag are denoted L and D, respectively, and the mass of the aircraft (assumed constant over the duration of the trajectory) by m. The aircraft moment of inertia and rotational dynamics are not included in Eq. (1) because it is a point mass model. For the purposes of this model the aircraft can be controlled directly via changes to the angle of attack and the roll angle. Sideslip is assumed to be negligible and is therefore omitted. Because of the relatively short distances and time intervals considered in this study, the Earth is regarded as ideally flat and nonrotating.

The aircraft model is constrained to incorporate the performance and structural limitations that are present on a real aircraft. Any avoidance trajectories that are generated can also include additional constraints for the safety of any passengers on board. These input constraints are modeled as lower and upper bounds

$$\alpha \in [\alpha_{\min}, \alpha_{\max}] \tag{2a}$$

$$\mu \in [\mu_{\min}, \mu_{\max}] \tag{2b}$$

$$T \in [0, T_{\max}] \tag{2c}$$

The limits on the angle of attack are chosen to remain within the vehicle stall limits, the roll is bounded by a suitable value for structural integrity and the time varying maximum thrust is determined by the engine specifications at a given altitude and Mach number. The state constraints are likewise modeled as

$$z_c \in [0, z_{\max}] \tag{3a}$$

$$V \in [V_{\min}, V_{\max}] \tag{3b}$$

$$\gamma \in [\gamma_{\min}, \gamma_{\max}] \tag{3c}$$

The maximum altitude achievable is set in Eq. (3a) as the performance ceiling of the aircraft. The velocity bounds can be a (time varying) function of the aircraft state, with the stall velocity and

maximum operating velocity V_{MO} as the bounding limits. Alternatively, the most conservative values for the given aircraft could be used as fixed constraints. The flight-path angle is bounded by a suitable value for both passenger safety and aircraft structural integrity. Rate constraints on the inputs $(\dot{\alpha}, \dot{\mu}, \dot{T})$ and also on the flight-path angle $\dot{\gamma}$ are incorporated to model the delay between the desired control input being applied and achieved.

It is possible for the system to be triggered with an initial state lying outside these bounds, e.g., if a hostile pilot adopts an extreme attitude in an attempt to crash. In these cases it is possible to initially extend the envelope to contain the extreme initial point, then reduce it in size at an achievable rate back to the intended envelope of the aircraft.

B. Multiple Shooting Optimal Control Problem

There are a number of strategies for solving nonlinear optimal control problems depending on the class of problem. Most practical problems are solved using nonlinear programming methods based on some discretization of the original continuous control problem. The controls u and states x can be discretized in time to form a finite dimensional approximation to the original state and input trajectory. This approximated problem can then be solved using one of many numerical optimization methods available.

Multiple shooting is one such method used frequently in the chemical engineering industry where large-scale control problems often arise [18]. It was first introduced by Plitt and Bock [19], and its use in aerospace applications suggested by Betts [20]. In multiple shooting the control horizon is divided into N intervals, with states variables introduced at the end of each interval as additional decision parameters. This is more advantageous than single shooting (e.g., optimization over the control inputs only, without modelling intermediate states) due to improved convergence and enhanced robustness to poor initial data, discretization or round-off errors [19]. Figure 4 shows a graphical interpretation of the multiple shooting strategy. The inputs are held constant over each interval and the states are decoupled.

1. *Discretizing the Optimal Control Problem*

To find a constraint-admissible trajectory for the aircraft trajectory generation problem, the aircraft dynamics Eq. (1) are incorporated into a single expression

$$\dot{x} = f(x, u) \tag{4}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the combined state and input vectors, respectively. Because the computational optimization method requires a finite dimensional decision space, the dynamics are then discretized by integration over fixed intervals ΔT using an ordinary differential equation solver. The state at the end of a multiple shooting segment is

$$\phi(x_i, u_i) \triangleq x_i + \int_t^{t+\Delta T} f(x(\tau), u(\tau))\, \mathrm{d}\tau \tag{5}$$

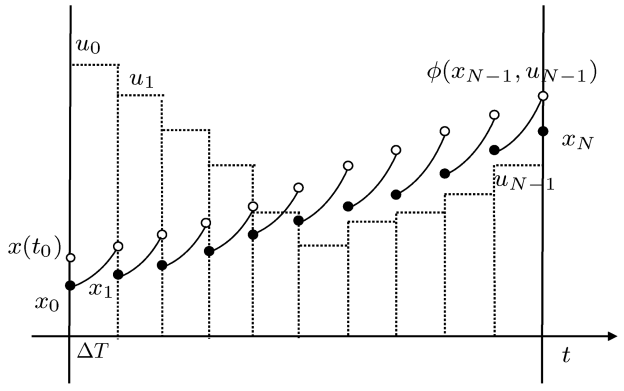


Fig. 4 Multiple shooting.

Alternatively, a simpler integration method (e.g., a first-order Euler method) can be adopted if the time step is small enough to ensure an acceptable level of accuracy. In either case, the function ϕ: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the dynamics of the discretized system, with $x_i \in \mathbb{R}^n$ and $u_i \in \mathbb{R}^m$, representing the system states and inputs at discrete time intervals, treated as decision variables. Continuity of the system state trajectory is ensured via introduction of equality constraints in the form

$$x_0 - x(t_0) = 0 \tag{6a}$$

$$\phi(x_i, u_i) - x_{i+1} = 0, \quad \forall\ i \in \{0, \ldots, N-1\} \tag{6b}$$

where the initial state $x_0 := x(t_0)$ is assumed known. For this system, we wish to determine a constraint-admissible state and input sequence that minimizes

$$J_N(\mathbf{x}, \mathbf{u}) = F(x_N) + \sum_{i=0}^{N-1} \ell_i(x_i, u_i) \tag{7}$$

where $\mathbf{u} \triangleq \{u_0, \ldots, u_{N-1}\}$ and $\mathbf{x} \triangleq \{x_0, \ldots, x_N\}$. The stage costs ℓ_i: $\mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ and terminal cost F: $\mathbb{R}^n \to \mathbb{R}_+$ are assumed to be twice continuously differentiable. Additional rate constraints on the states and inputs can be posed in the form

$$\dot{u}_{\min} \leq \frac{u_0 - u(t_0)}{\Delta T} \leq \dot{u}_{\max} \tag{8a}$$

$$\dot{u}_{\min} \leq \frac{u_{i+1} - u_i}{\Delta T} \leq \dot{u}_{\max}, \quad \forall\ i \in \{0, \ldots, N-2\} \tag{8b}$$

$$\dot{x}_{\min} \leq \frac{x_{i+1} - x_i}{\Delta T} \leq \dot{x}_{\max}, \quad \forall\ i \in \{0, \ldots, N-1\} \tag{8c}$$

where the condition Eq. (8a) ensures that the first input of the control sequence u_0 can feasibly be reached from the initial condition $u(t_0)$. In practical terms this ensures that the generated control sequence starts close to the pilot's current selection.

2. *Modeling State Exclusion Regions*

In addition to constraints based on vehicle performance limitations, there exists a collection of regions in space (*prohibited volumes*) from which the vehicle trajectory is to remain excluded. Therefore, a collection of M sets $\mathcal{C}_j \subseteq \mathbb{R}^3$ are defined with associated reference points r_j, such that the aircraft trajectory is to be excluded from each of the sets $\mathcal{C}_j \oplus r_j$, where the $\oplus$ operator denotes translation of the set by r_j. Each of the sets $\mathcal{C}_j$ is assumed to be closed and convex with nonempty interior. Formally, the following set exclusion condition is imposed:

$$p(x_i) \notin \bigcup_{j=1}^{M} (\mathrm{int}\mathcal{C}_j \oplus r_j), \quad \forall\ i \in \{0, \ldots, N\} \tag{9}$$

where int $\mathcal{C}_j$ denotes the interior of the set $\mathcal{C}_j$, and $p(\cdot)$ is a function extracting the position vector from the full state vector. It is possible to model a very wide variety of constraint sets in this way, including nonconvex regions of space, due to the presence of the union operation. The constraint Eq. (9) dictates that the system state remain outside of a (collection of) convex set(s), and is therefore nonconvex.

The optimal control problem we wish to solve can be written

$$\min_{\mathbf{x},\mathbf{u}} J_N(\mathbf{x}, \mathbf{u})$$

subject to: (2), (3), (6), (8) $\quad (\mathcal{P})$

$$p(x_i) \notin \bigcup_{j=1}^{M} (\mathrm{int}\mathcal{C}_j \oplus r_j), \quad \forall\ i \in \{0, \ldots, N\}, \quad \forall\ j \in \{1, \ldots, M\}$$

A difficulty in solving this problem is the presence of the nonconvex and nondifferentiable state exclusion regions. State exclusion conditions (obstacle constraints) of this type often appear in path planning problems. Earlier work considered the obstacle avoidance problem in a continuous time framework using dynamic optimization [21–23]. The set of states that can be steered to a target set, while satisfying bound constraints and avoiding obstacles, is a level set of the value function of the dynamic optimization problem, obtained by solving a Hamilton–Jacobi–Bellman equation. Another approach is to include an additional potential function for avoiding obstacles in the cost function, thus converting the collision avoidance problem into an unconstrained optimization allowing gradient-based solvers [24]. However, the construction of potential functions in a general framework is difficult. Polytopic constaints are often dealt with via the use of additional binary variables in mixed integer optimization, each half-space representing a polytope face is a linear constraint, and satisfaction of at least one constraint ensures avoidance [25,26]. A new approach to modeling polytopic constraints is developed in this paper, that avoids the use of integer variables by preserving continuity and is amenable for use with a gradient-based optimization strategy.

As stated, the constraint Eq. (9) can be nondifferentiable in general, e.g., in the case where the sets $\mathcal{C}_j$ are polytopic. Therefore, these constraints are remodeled in such a way that continuity and differentiability are preserved, avoiding the need for integer variables. To do this, the following preliminary definitions and results are required:

Definition 1 (Polar Sets): *Given a set* $C \subseteq \mathbb{R}^n$ with $0 \in C$, the polar of C is defined as:

$$C^\circ \triangleq \{v | \langle v, x\rangle \leq 1, \quad \forall\ x \in C\}$$

Lemma 1 (Properties of Polar Sets): The following properties hold [27]:

1) If C is closed and convex with $0 \in \text{int}C$, then C° is compact and convex with $0 \in \text{int}C^\circ$.

2) If the set C is polyhedral with

$$C = \{x | Hx \leq 1\}$$

then

$$C^\circ = \{y | \exists z, y = H^\top z, z \geq 0, z^\top \mathbf{1} = 1\}$$

3) If C is a unit ball for some norm $\|\cdot\|$, i.e., $C = \{x | \|x\| \leq 1\}$, the polar set is $C^\circ = \{x | \|x\|_* \leq 1\}$ where $\|\cdot\|_*$ is the associated dual norm. In particular, for the p–norm unit ball $\mathcal{B}_p \triangleq \{x \| \ \|x\|_p \leq 1\}$

$$(\mathcal{B}_p)^\circ = \mathcal{B}_q, \qquad 1 < p < \infty, \qquad p^{-1} + q^{-1} = 1$$
$$(\mathcal{B}_1)^\circ = \mathcal{B}_\infty, \qquad (\mathcal{B}_\infty)^\circ = \mathcal{B}_1$$

4) If $C = \{x | \|Hx\| \leq 1\}$ for some matrix H, then

$$C^\circ = \{x | \exists y, x = H^\top y, \|y\|_* \leq 1\}$$

Definition 2 (Support and Gauge Functions): Given a convex set $C \subseteq \mathbb{R}^n$, the support function σ_C: $\mathbb{R}^n \to \bar{\mathbb{R}}$ is defined as:

$$\sigma_C(x) \triangleq \sup_{y \in C}(x^\top y)$$

If $0 \in C$ the gauge function γ_C: $\mathbb{R}^n \to \bar{\mathbb{R}}$ is defined as:

$$\gamma_C(x) \triangleq \inf\{\lambda \geq 0 | x \in \lambda C\}$$

Remark 1: The support and gauge functions have straightforward geometric interpretations; the set $\{y | x^\top y = \sigma_C(x)\}$ defines a plane tangent to C with normal vector x, while $\lambda = \gamma_C(x)$ is the least amount by which C can be scaled while guaranteeing that $x \in \lambda C$.

The support and gauge functions of a set C have several properties that will be useful in subsequent sections; principal among these properties is their relation to one another through the polar set C°:

Lemma 2 (Properties of Support and Gauge Functions): If C *is a closed and convex set with* $0 \in C$, then the following properties hold:

1) $\sigma_C(\cdot) = \gamma_{C^\circ}(\cdot)$ and $\gamma_C(\cdot) = \sigma_{C^\circ}(\cdot)$ [27].

2) The Fenchel conjugate of the gauge function [28]

$$\gamma_C^*(y) \triangleq \sup_x\{x^\top y - \gamma_C(x)\}$$

is

$$\gamma_C^*(y) = \begin{cases} 0 & \text{if } y \in C^\circ \\ \infty & \text{otherwise} \end{cases}$$

i.e., the conjugate of the gauge function is the indicator function of the associated polar set.

Because the gauge function is the minimum scaling required for a set $\mathcal{C}_j$ centered at r_j to include the point $p(x_i) - r_j$, the set exclusion condition Eq. (9) can be remodeled as

$$\gamma_{\mathcal{C}_j}(p(x_i) - r_j) \geq 1, \quad \forall\ i \in \{0, \ldots, N\}, \quad \forall\ j \in \{1, \ldots, M\}$$

Using Lemma 2.1, this can be rewritten as

$$\sigma_{\mathcal{C}_j^\circ}(p(x_i) - r_j) \geq 1$$

From Definition 2 for the support function, this in turn can be written as

$$\sup_{y \in \mathcal{C}_j^\circ}((p(x_i) - r_j)^\top y) \geq 1$$

which is equivalent to

$$\begin{aligned} p(x_i) - r_j \in \{\tilde{p}(x) | \exists y, \tilde{p}(x)^\top y \geq 1, y \in \mathcal{C}_j^\circ\}, \\ \forall\ i \in \{0, \ldots, N\}, \quad \forall\ j \in \{1, \ldots, M\} \end{aligned} \tag{10}$$

Remark 2: Note that any vector y satisfying the conditions of Eq. (10) provides a certificate verifying the condition $p(x_i) - r_j \notin \text{int}\mathcal{C}_j$, because the hyperplane $\{p(x) | p(x)^\top y = 1\}$ contains the point $p(x_i)$ and is disjoint from the set $(\text{int}\mathcal{C}_j \oplus r_j)$.

3. Complete Control Problem

The cost function for the problem is defined as

$$J_N(\mathbf{x}, \mathbf{u}) = \|(x_N - x_N^r)\|_P^2 + \sum_{i=0}^{N-1}(\|x_i - x_i^r\|_Q^2 + \|u_i - u_i^r\|_R^2) \tag{11}$$

where $P \in \mathbb{R}^{6\times6}$, $Q \in \mathbb{R}^{6\times6}$, and $R \in \mathbb{R}^{3\times3}$ are positive semidefinite, symmetric matrices adjusted to penalize the states or inputs as required. The calculation is initialized with a reference trajectory, $\{x_i^r\}$ and $\{u_i^r\}$. The reference trajectory is generated by time integration of the model from the initial state x_0, with the inputs fixed as $u(t_0)$ over the time horizon. Deviations from this trajectory on some or all of the states and inputs can then be penalized within Eq. (11).

The aircraft-specific optimal control problem $\mathcal{P}(x(t_0), u(t_0))$, where $x(t_0)$ and $u(t_0)$ are the state and input at initial time t_0, can now be written as

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{u}} J_N(\mathbf{x}, \mathbf{u}) \\ &\quad \text{subject to: } (2), (3), (6), (8) \qquad \mathcal{P}' \\ &\quad \left.\begin{aligned} &(p(x_i) - r_j)^\top y_{i,j} \geq 1 \qquad y_{i,j} \in \mathcal{C}_j^\circ \\ &\qquad \forall\ i \in \{0, \ldots, N\}, \quad \forall\ j \in \{1, \ldots, M\} \end{aligned}\right\} \end{aligned}$$

The set of feasible initial conditions is

$$\mathcal{X}_{\mathcal{N}} \triangleq \{(x(t_0), u(t_0) | \mathcal{P}' \text{ is feasible with initial state } x(t_0), \text{ initial input } u(t_0) \text{ and horizon } N\} \tag{12}$$

4. Geometric Interpretation and Implementation of $\mathcal{P}'$

The geometric interpretation of the remodeling of problem $\mathcal{P}$ into the form $\mathcal{P}'$ via introduction of polar sets and dual variables is straightforward. The polar sets C_j° are large in the directions for which the original constraint sets C_j are small, and vice versa. If one is able to find a multiplier $y_{i,j} \in C_j^\circ$ such that the second to last constraint in $\mathcal{P}'$ is satisfied, then that multiplier must be aligned sufficiently well with the relative aircraft position vector $(p(x_i) - r_j)$ to ensure that the aircraft position at time i lies outside of the original constraint set $(\text{int}C_j \oplus r_j)$.

This remodeling ensures that the problem constraints (and in particular polytopic constraints) are representable in terms of continuously differentiable functions, rather than via recourse to integer variables. This has considerable practical advantages for numerical implementation, because it allows the problem $\mathcal{P}'$ to be solved using an existing gradient-based solver suitable for continuous optimization problems. This is achieved at the cost of increased problem dimensionality due to the appearance of the new dual variables $y_{i,j}$ at each time step. The nonconvexity of the control problem is of course unchanged by this remodeling.

For the most common constraint types in practical applications (e.g., polytopes, scaled norm balls for different norms), the polar sets C_j° are not inherently more complex to represent than the original sets C_j. From Lemma 1, polar sets of polytopes are themselves polytopes (with face normals exchanged with vertices), and scaled balls in any norm are replaced with scaled balls in the associated dual norm.

In practice this approach is only required for polytopic constraints, because norm ball constraints are differentiable in all cases except the infinity norm, which is dealt with in Remark 3. In Sec. III, however, extension of the problem to the infinite horizon is addressed by employing the same dualization methods. In this case dualization is required for both the norm ball and polytopic constraints.

Remark 3: Note that the constraints of the polar set C° (*Lemma 1.4*) are differentiable for the 2-norm case. The infinity norm case $C = \{x | \|Hx\|_\infty \leq 1\}$ requires an extra step to preserve continuity. Because the norm $\|\cdot\|_\infty$ has the dual norm $\|\cdot\|_1$, then

$$C^\circ = \{x | \exists y, x = H^\top y, \|y\|_1 \leq 1\} \tag{13}$$

The variable y can be further partitioned into positive and negative components, respectively, to give

$$C^\circ = \{x | \exists (y^+, y^-) \geq 0, x = H^\top (y^+ - y^-), \mathbf{1}^\top (y^+ + y^-) \leq 1\} \tag{14}$$

where the constraints are now differentiable at the cost of additional variables (y^+, y^-).

In those cases where the constraint sets $\mathcal{C}_j$ are in the forms outlined in Lemma 1, the problem $\mathcal{P}'$ can be composed (possibly after the inclusion of ancillary variables) entirely in terms of twice continuously differentiable functions, thereby allowing the use of one of the many standard gradient-based optimization algorithms that are available.

C. Optimization Strategy

The problem of direct aircraft trajectory generation is constrained, nonlinear, continuous, and multivariate. There are a number of optimization packages available for solving these types of problems. A gradient-based interior-point approach is chosen to solve this problem because these methods are typically efficient for large-scale problems, particularly when there are a number of active constraints [29]. The advantage of using a multiple shooting strategy with a gradient-based optimization is the sparse bounded Jacobian matrix resulting from the problem structure.

The algorithm chosen, IPOPT [30], uses a barrier method which prevents iterates becoming infeasible. Consider the optimization problem:

$$\min_z J(z) \qquad \text{subject to: } g(z) = 0, \quad z \geq 0 \tag{15}$$

with decision variable $z \in \mathbb{R}^n$. The objective function J: $\mathbb{R}^n \to \mathbb{R}$ and the equality constraints g: $\mathbb{R}^n \to \mathbb{R}^m$ are assumed to be twice continuously differentiable with $m \leq n$. Problems with nonlinear inequality constraints, $h(z) \leq 0$ are easily recast in the form Eq. (15) via introduction of additional slack variables. Interior point methods typically compute solutions for a sequence of barrier problems

$$\min_{z \in \mathbb{R}^n} J(z) - r_j \sum_{i=1}^{n} \log(z^{(i)}) \quad \text{subject to: } g(z) = 0, \quad z \geq 0 \tag{16}$$

where $z^{(i)}$ represents the i-th component of z, for a decreasing series of barrier parameters r_j converging to zero, to solve the original problem Eq. (15) [31]. The logarithmic term ensures the solution stays within the feasible region by penalizing solutions close to the boundaries. As $r_j \downarrow 0$ the penalties disappear allowing for solutions that converge to the constraint boundaries. Solutions to each barrier problem are found by solving the Karush–Kuhn–Tucker (KKT) conditions

$$\begin{aligned} &\nabla J(z) + \nabla g(z)\lambda = 0 \qquad g(z) = 0 \qquad ZSe - re = 0 \\ &(Z, S) > 0 \end{aligned} \tag{17}$$

where $Z \triangleq \text{diag}(z)$, $S \triangleq \text{diag}(s)$, $e = (1, 1, \ldots, 1)$, and $\lambda \in \mathbb{R}^m$ and $s \in \mathbb{R}^n$ are the Lagrange multipliers for the equality and bound constraints of Eq. (16), respectively. For nonlinear problems the KKT conditions are also nonlinear and solved via a line search. To solve a single barrier problem in the series (fixed value of r_j), a damped Newton's method is applied to solve Eq. (17). Given an iterate (z_k, λ_k, s_k) with $(z_k, s_k) > 0$, search directions $(d_k^z, d_k^\lambda, d_k^s)$ are obtained from the linearization of (17) at (z_k, λ_k, s_k), given by

$$\begin{aligned} &\begin{bmatrix} \nabla^2_{zz}\mathcal{L}(z_k, \lambda_k, s_k) & \nabla g(z_k) & -I \\ \nabla g(z_k)^T & 0 & 0 \\ S_k & 0 & Z_k \end{bmatrix} \begin{pmatrix} d_k^z \\ d_k^\lambda \\ d_k^s \end{pmatrix} \\ &\quad = -\begin{pmatrix} \nabla J(z_k) + \nabla g(z_k)\lambda - s_k \\ g(z_k) \\ Z_k S_k e - r_j e \end{pmatrix} \end{aligned} \tag{18}$$

where

$$\mathcal{L}(z, \lambda, s) = J(z) + g(z)^T \lambda - s \tag{19}$$

is the Lagrangian function of the original problem Eq. (15).

D. Outer-Loop Controller

Given the multiple shooting optimal control problem $\mathcal{P}(x(t_0), u(t_0))$, a control sequence $\mathbf{u}$ is calculated that would produce a state sequence $\mathbf{x}$ assuming the system dynamics $f(x, u)$ are a perfect match to the real aircraft. In reality, model disturbances would produce a state sequence $\mathbf{x} + \Delta\mathbf{x}$ for the outer-loop controller. To mitigate this state error in the trajectory generation part of the scheme a suitably robust inner-loop controller is required for the trajectory following. For the purposes of this paper it is assumed that the inner-loop controller allows accurate tracking of $\mathbf{x}$.

The outer-loop controller runs continuously in the background as the pilot flies the aircraft. Once the override trigger is activated at time t_0 after a threat of PV violation is detected, control is removed from the pilot and the inner-loop controller flies the sequence $\mathbf{x}$ from the

solution of $\mathcal{P}(x(t_0 - t_d), u(t_0 - t_d))$ where t_d is the time delay between trigger activation and the optimization calculation start time. At time t_f the inner-loop controller is disengaged and control of the aircraft is returned to the pilot. The proceeding section of the paper is concerned with finding a solution to $\mathcal{P}(x(t_0), u(t_0))$ such that a solution to $\mathcal{P}(x(t_f), u(t_f))$ exits and more generally $\mathcal{P}(x(t), u(t))$ exists where $t \in [t_0, \infty]$.

III. Extension of the Optimization Problem $\mathcal{P}'$ to the Infinite Horizon

In this section a positively invariant terminal set is constructed. The motivation for doing so is to ensure that a trajectory cannot end with a PV obstacle directly in the subsequent flight path. This standard method in predictive control is achieved by once again employing dualization. The dualization approach here is required for both differentiable and nondifferentiable obstacle constraints as it is deemed difficult to achieve feasibility over an infinite horizon otherwise. The construction of the invariant set also delivers an initial iterate for the next optimization via shift initialization, therefore ensuring that the optimization starts near a feasible solution.

It is important to note that while the terminal invariance technique of predictive control is being employed, the problem is not recursively feasible. Because the control input is not actually applied to the aircraft while the pilot is still in command, the pilot can continue to fly until inside a PV. At some point close to the PV the problem becomes infeasible (no control action is available to avoid the PV). The aim of the overall system is to remove control authority from the pilot before this situation can occur. The methods for doing so are presented in [32].

A. Terminal Invariance Condition

An obvious difficulty with the problem formulation $\mathcal{P}'$ is that it generates a trajectory to avoid prohibited volumes over a finite time interval only, and does not therefore provide any assurance of obstacle avoidance over an infinite horizon [10].

The core difficulty is that the feasible set Eq. (12) does not satisfy the nesting condition

$$\cdots \subseteq \mathcal{X}_{i-1} \subseteq \mathcal{X}_i \subseteq \mathcal{X}_{i+1} \subseteq \cdots$$

i.e., the set of feasible initial conditions for $\mathcal{P}'$ is not guaranteed to be monotonically nondecreasing with increasing horizon length. A guarantee such as this can be provided by augmenting $\mathcal{P}'$ with a suitable target/terminal set $\mathcal{X}_f \subseteq \mathbb{R}^n$, along with an appropriate terminal control law $\kappa_f \colon \mathbb{R}^n \to \mathbb{R}^m$. For simplicity the terminal control law is assumed to be constant, i.e., $\kappa_f(x) = u_f$ for all x. The following requirements are imposed on the terminal conditions:

Assumption 1: The set $\mathcal{X}_f \subseteq \mathbb{R}^n$ and constant control law $u_f = \kappa_f(x)$ satisfy the following conditions:

1) $\mathcal{X}_f$ satisfies the constraints Eq. (3) for all $x \in \mathcal{X}_f$.

2) The control input u_f satisfies the constraints Eq. (2) for all $x \in \mathcal{X}_f$.

3) The set $\mathcal{X}_f$ is positively invariant under the control law κ_f, i.e., $\phi(x, \kappa_f(x)) \in \mathcal{X}_f$ *for all* $x \in \mathcal{X}_f$.

4) The position components of $\mathcal{X}_f$ do not intersect any prohibited volume, that is

$$\mathbb{P}_3 \mathcal{X}_f \bigcap (\text{int}\mathcal{C}_j \oplus r_j) = \emptyset, \quad \forall\ j \in \{1, \ldots, M\} \tag{20}$$

where $\mathbb{P}_3 \mathcal{X}_f$ is the projection of the set $\mathcal{X}_f$ onto the position states.

A set satisfying the above assumptions can be constructed by imposing conditions to ensure that the terminal set $\mathcal{X}_f$ and input u_f represent a level flight condition or steady climb, and that the straight-line extrapolation of the aircraft terminal state does not intersect any prohibited volumes.

The first of these conditions can be satisfied by ensuring that the terminal control input u_f satisfies a null space condition

$$Gf(x, u_f) = 0 \tag{21}$$

for $x \in \mathcal{X}_f$, where $G \in \mathbb{R}^{6\times 6}$ is a diagonal matrix selecting the three states $\dot{\gamma}$, $\dot{\chi}$ and $\dot{v}$. Denoting the vehicle velocity vector at some position state $p(x)$ as $v(x)$, and defining the straight line projection of a given state as

$$R(p(x), v(x)) \triangleq \{p(x) + \beta v(x) | \beta \geq 0\}$$

the condition Eq. (20) can be satisfied by ensuring that the ray $R(p(x), v(x))$ does not intersect any prohibited volume for any $x \in \mathcal{X}_f$. A set $\mathcal{X}_f$ can now be constructed satisfying all of the conditions of Assumption 1 by defining it as

$$\begin{aligned}\mathcal{X}_f &\triangleq \{x | \exists u,\ Gf(x,u) \\ &= 0,\ (x,u) \text{ satisfy constraints (2), (3) } R(p(x), v(x)) \bigcap (\text{int}\mathcal{C}_j \oplus r_j) \\ &= \emptyset, \quad \forall\ j \in \{1, \ldots, M\}\end{aligned}$$

The following analysis provides a result that allows the treatment of the set exclusion condition (4) entirely in terms of continuously differentiable functions. This is achieved by once again exploiting the results of Sec. II to remodel the problem in terms of the polar sets $\mathcal{C}_j$.

Proposition 1: Suppose that the vectors $(p(x), v(x)) \in \mathbb{R}^3$ parameterize a ray

$$R(p(x), v(x)) \triangleq \{p(x) + \beta v(x) | \beta \geq 0\}$$

and C is a closed and convex set containing the origin in its interior, and $r \in \mathbb{R}^3$. Then the condition

$$R(p(x), v(x)) \bigcap (\text{int}C \oplus r) = \emptyset$$

is satisfied if and only if

$$\{y | (p(x) - r)^\top y \geq 1, v(x)^\top y \geq 0, y \in C^\circ\} \neq \emptyset \tag{22}$$

Proof: Rewriting in terms of the gauge function of C, the first statement is equivalent to

$$\inf\{\gamma_C(z) | \beta \geq 0, z = (p(x) - r) + \beta v(x)\} \geq 1 \tag{23}$$

The Lagrange dual function for this optimization problem is

$$g(\lambda, y) = \inf_{\beta, z} (\gamma_C(z) + ((p(x) - r) + \beta v(x) - z)^\top y - \beta\lambda)$$

with multipliers (λ, y). This can be rewritten as

$$\begin{aligned} g(\lambda, y) &= \inf_z (\gamma_C(z) - z^\top y) + \inf_\beta (\beta(v(x)^\top y - \lambda)) + (p(x) - r)^\top y \\ &= -\sup_z (z^\top y - \gamma_C(z)) + \inf_\beta (\beta(v(x)^\top y - \lambda)) + (p(x) - r)^\top y \\ &= -\gamma_C^*(y) + \inf_\beta (\beta(v(x)^\top y - \lambda)) + (p(x) - r)^\top y \end{aligned}$$

Recalling from Lemma 2.2 that the gauge function's conjugate γ_C^* is equivalent to the indicator function of the polar set C°, the above simplifies to

$$g(\lambda, y) = \begin{cases} (p(x) - r)^\top y & \text{if } y \in C^\circ \quad \text{and} \quad v(x)^\top y = \lambda \\ -\infty & \text{otherwise} \end{cases}$$

Because the minimization problem Eq. (23) is convex and contains only linear constraints, Slater's condition is satisfied and its dual maximization problem

$$\sup_{(y, \lambda \geq 0)} g(\lambda, y) = \sup\{(p(x) - r)^\top y | v(x)^\top y \geq 0, y \in C^\circ\}$$

obtains the minimum value of Eq. (23). Because the polar set C° is compact (Lemma 1.1), a maximizer for this problem always exists.

Therefore the set Eq. (22) is guaranteed to be nonempty if and only if Eq. (23) holds. □

The generalized condition Eq. (22) for a closed and convex set C can be used to derive constraints for polyhedral and p-norm ball prohibited volumes using Lemma 1 (see Fig. 5).

Corollary 1: (polyhedral prohibited volume) The condition

$$R(p(x), v(x)) \bigcap (\text{int}C \oplus r) = \emptyset$$

with

$$C = \{p(x) | Hp(x) \leq 1\}$$

is satisfied if and only if

$$\{\eta | (p(x) - r)^\top H^\top \eta \geq 1, v(x)^\top H^\top \eta \geq 0, \eta^\top \mathbf{1} = 1, \eta \geq 0\} \neq \emptyset \tag{24}$$

Proof: From Lemma 1.2, the condition $y \in C^\circ$ is equivalent to

$$C^\circ = \{y | \exists \eta, y = H^\top \eta, \eta \geq 0, \eta^\top \mathbf{1} = 1\}$$

Substituting into Eq. (22) obtains the condition Eq. (24) above. □

Corollary 2: (p-norm ball prohibited volume) The condition

$$R(p(x), v(x)) \bigcap (\text{int}C \oplus r) = \emptyset$$

with

$$C = \{p(x) | \|Hp(x)\| \leq 1\}$$

is satisfied if and only if

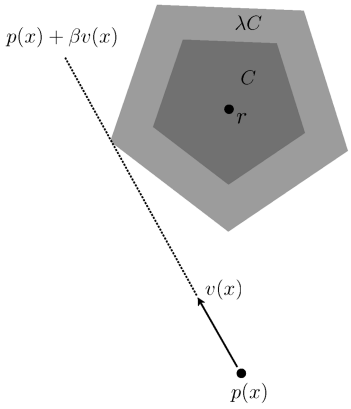


Fig. 5 Polyhedral PV and scaling via the gauge function.

$$\{\eta | (p(x) - r)^\top H^\top \eta \geq 1, v(x)^\top H^\top \eta \geq 0, \|\eta\|_* \leq 1\} \neq \emptyset \tag{25}$$

Proof: From Lemma 1.4, the condition $y \in C^\circ$ is equivalent to

$$C^\circ = \{y | \exists \eta, y = H^\top \eta, \|\eta\|_* \leq 1\}$$

Substituting into Eq. (22) obtains the condition Eq. (25) above. □

Using the duality property of the Lagrange dual function [33], the original inner minimization of Eq. (23) is converted into a maximization where only the existence of a solution has to be proved for the constraint to be satisfied. The constraints can be interpreted geometrically in two dimensions as shown in Fig. 6. In these diagrams the PV is shown centered at the origin ($r = 0$) and the shaded half-spaces are regions from which η is excluded. As well as defining the bounding region shown in the diagrams (offset from the center), the constraint $p(x)^\top H^\top \eta \geq 1$ also ensures that the starting point is not already inside the PV. The permissible region for the η vector narrows as the projected velocity passes closer to the PV. If the projected velocity passes through the PV, the set Eq. (25) is empty.

The construction of a positively invariant target set $\mathcal{X}_f$ has been demonstrated. The additional constraints are easy to incorporate into the original problem structure of $\mathcal{P}'$:

$$\begin{aligned}
&\min_{\mathbf{x},\mathbf{u}} J_N(\mathbf{x}, \mathbf{u}) \qquad \text{subject to: (2), (3), (6), (8)} \\
&(p(x_i) - r_j)^\top y_{i,j} \geq 1, \quad \forall\ i \in \{0, \ldots, N\} \\
&y_{i,j} \in C_j{}^\circ, \quad \forall\ i \in \{0, \ldots, N\} \qquad (p(x_N) - r_j)^\top \eta_j \geq 1 \\
&v(x_N)^\top \eta_j \geq 0 \qquad \eta_j \in C_j{}^\circ \Big\} \quad \forall\ j \in \{1, \ldots, M\}
\end{aligned}$$

where $y_{i,j}$ and η_j are auxiliary variables in the problem. Note that the constraints of this problem are differentiable and, in the case of polytopic obstacles, can be solved without the introduction of integer variables. This preserves the overall continuity of the problem, ensuring that its solution is amenable to the use of standard solvers for continuous nonlinear optimization.

B. Warm Starting

The proposed scheme has similarities with a traditional MPC formulation. However, two key differences exist. Firstly, in normal operation before the override is triggered, the control generated by the system is not applied because the pilot retains control of the aircraft. Therefore, it is possible for the problem to become infeasible close to a PV. This is mitigated by ensuring the override is triggered before this can happen (see [32]). Secondly, when the override has been triggered the control input generated by the reduced-order model optimization is not explicitly applied as there is no obvious mapping of inputs from the 3DOF model to the real aircraft. Instead, an inner-loop controller is used to track the trajectory. Once the override is activated and the aircraft is being flown by the controller,

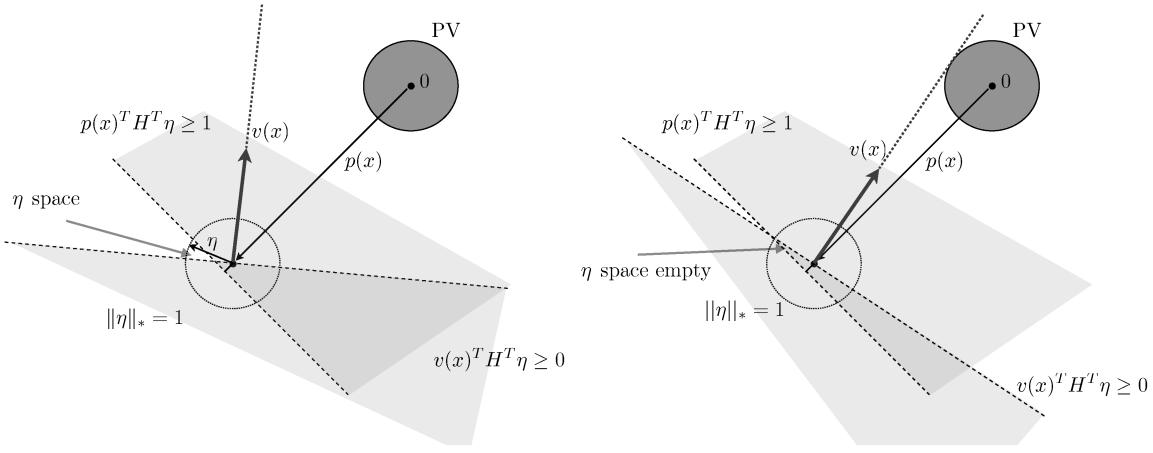


Fig. 6 Left: projected velocity far from PV. Right: projected velocity touching PV.

the optimization is not recalculated at each time step as is required for traditional MPC. Instead the last avoidance trajectory generated is flown to completion via the inner-loop controllers.

There is a choice available in how to initialize the optimization, both for the decision variables and the multipliers. Warm starting is commonly used to decrease the computation time for the optimization but can have an effect on the algorithm's ability to find a solution as shown in Sec. IV. In this paper only the primal (decision) variables in the optimization are investigated (note that auxiliary variables in the problem are always initialized to zero). Assuming an optimal solution $(x_0^*; u_0^*; x_1^*; u_1^*; \ldots; u_{N-1}^*; x_N^*)$ to the original problem $\mathcal{P}'$ has been computed, in the finite horizon case the next calculation at future time t_1 can be initialized with $(x_1^*; u_1^*; \ldots; u_{N-1}^*; x_N^*)$. From the dynamic programming principle, this solution will be optimal if the initial state at t_1 is x_1^*. For the moving horizon case the new variables at the end of the time horizon u_N, x_{N+1}, also need to be initialized. This is often done using shift initialization, where the state x_{N+1} is obtained by forward simulation holding the final input constant, i.e., $u_N := u_{N+1}$ [34]. The positive invariance aspect of the formulation serves two purposes. Given the assumption of perfect tracking of $\mathbf{x}$, a solution to $\mathcal{P}(x(t_f), u(t_f))$ is known to exist and therefore the aircraft is not left in a trap situation at the end of an avoidance trajectory. The solution of $\mathcal{P}(x(t_0), u(t_0))$ can also be used before the override is triggered to initialize the problem for the next time step using shift initialization, as opposed to initialization using the reference trajectory. This prevents the decision variable becoming stuck in a locally suboptimal region and ensures that a solution can be found when one is available, as demonstrated in Sec. IV.

IV. Results

The first part of this section describes some scenarios for which the optimization problem (including end constraints) has been run with various PV constraints. The second part looks at the effect of warm starting on computation time and demonstrates how the combination of warm starting and positive invariance provides an initialization point close to the feasible region. The aircraft data for a Boeing 767 is used to conduct the simulations [35]. The implementation contains an additional throttle to thrust model based on the engine type of the aircraft. In this case the Pratt and Whitney 4056 installed engine thrust $(T(z, v))$ is modeled using data from the design operating points and Mattingly's code used to deduce the off design data [36]. The relationship between throttle and thrust is then assumed to be linear. This would not be the case in reality, however, in an actual implementation the data would be available. In the optimization an explicit second order Runge–Kutta integration scheme is used to propagate the model dynamics. The number of multiple shooting segments is $N = 50$ and the time horizon $t_f = 20$ s. In all the simulations shown the cost matrices $P = Q = I$ and $R = I$, i.e., deviations of all the states and inputs away from their reference values are penalized. Angular states in the model are measured in radians and position states in kilometers.

A. Algorithm Performance

The graphs of the results are shown in Figs. 7–10. In the state/input graphs, on the right of each figure, the aircraft states are given in the top two rows and the inputs in the final row. In the trajectory graphs the reference trajectory (the current predicted trajectory x^r with inputs u^r held fixed) is shown in white, and the problem solution (avoidance trajectory) shown as the darker line. The dashed line appended to the avoidance trajectory shows the projected final trajectory with the final input u_N held fixed. The PVs are placed with their centers at $z_c = 0$ (ground level). The simulation is started with the aircraft traveling along the x_c axis. The optimization is initialized with the calculated reference trajectory, and the additional variables $u_N = u_{N-1}$ and $\eta_j = 0$. The number of iterations to convergence and time taken by the algorithm are given in the figure captions. The algorithm is implemented in C++ and simulations performed on a 2.4 GHz Intel iMac.

Conservative bounds on the max/min velocity are calculated from the aircraft's initial state. The aircraft is assumed to be flying in a clean configuration (i.e., without flaps and landing gear), however, different models could incorporate these conditions. A ground constraint such that $z > 0$ for all t is imposed so that the aircraft cannot be flown into the ground. As well as the rate constraints on all the inputs and flight-path angle $\dot{\gamma}$, bound constraints are applied to the absolute values ($|\mu| \leq 20^\circ$ and $|\gamma| \leq 20^\circ$). A minimum final height of 50 m is imposed to ensure safe separation from the ground. The flight-path angle at the end of the trajectory is constrained to be $0^\circ \leq \gamma \leq 5^\circ$ to ensure the aircraft is not pointed nose down or in an excessive climb. The terminal invariance constraint on the heading angle rate $\dot{\chi} = 0$ ensures that the roll angle μ is zero at the end of the trajectory giving straight flight.

1. Single Hemispherical PV

The first case shown in Fig. 7 is a single hemispherical PV with a radius of 250 m. The aircraft is started on a banked trajectory shown in white, passing through the side of the PV. The avoidance trajectory climbs above the PV and leaves the aircraft in a steady climb. The α profile shows a drop after 10 s as the solution tries to move back to the reference trajectory (recall that deviations from this line are penalized). There is a subsequent increase to ensure the γ end condition is satisfied. The μ profile shows a drop to zero so that the aircraft is wings level at the end of the trajectory. The solve time for the 13 IPOPT iterations is 0.24 s indicating that the solution method is viable for a real-time application.

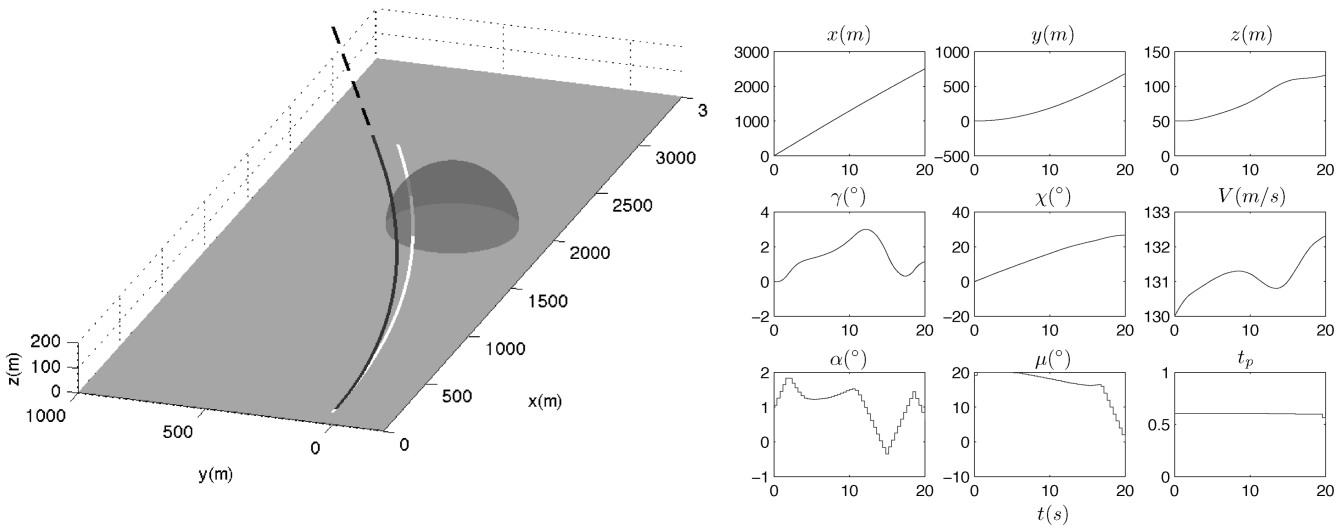


Fig. 7 Solve time = 0.24 s, 13 IPOPT iterations.

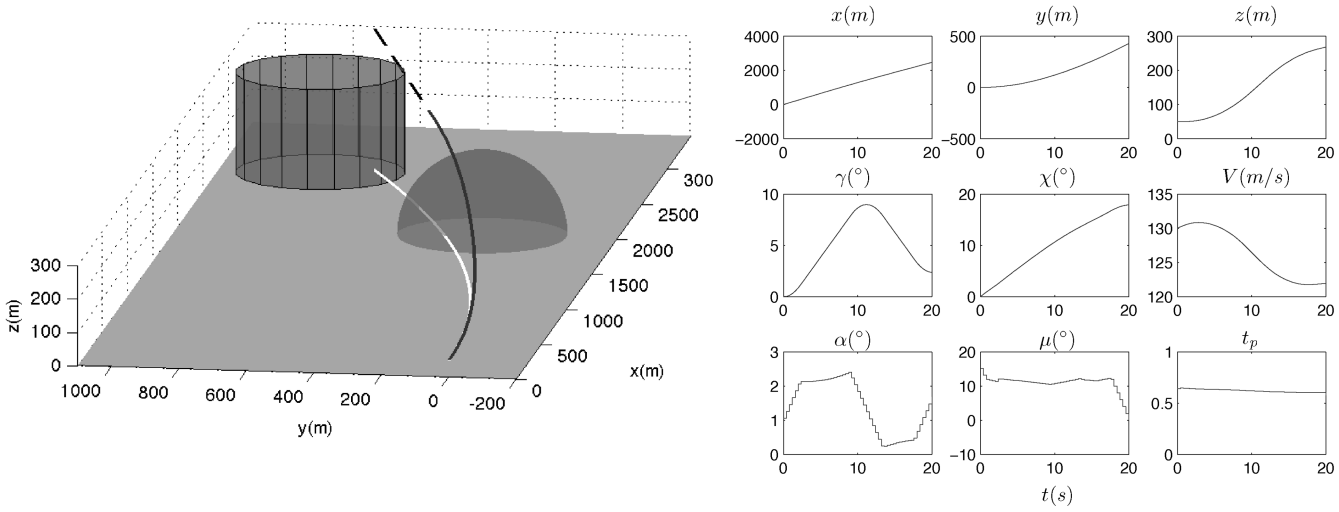


Fig. 8 Solve time = 0.44 s, 21 IPOPT iterations.

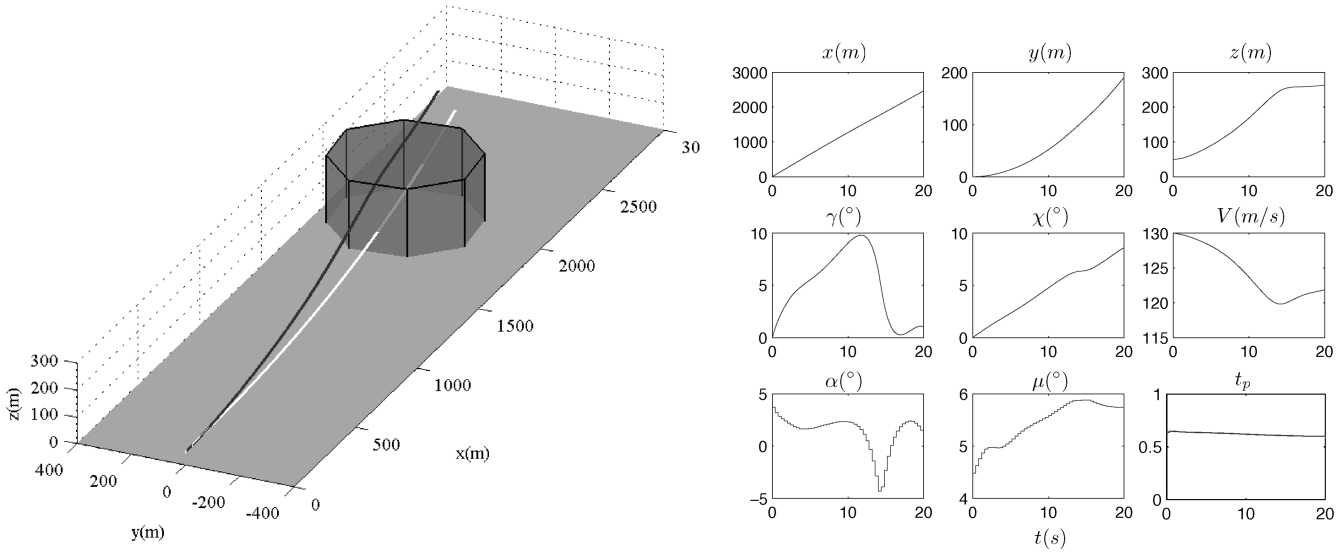


Fig. 9 Solve time = 0.65 s, 27 IPOPT iterations.

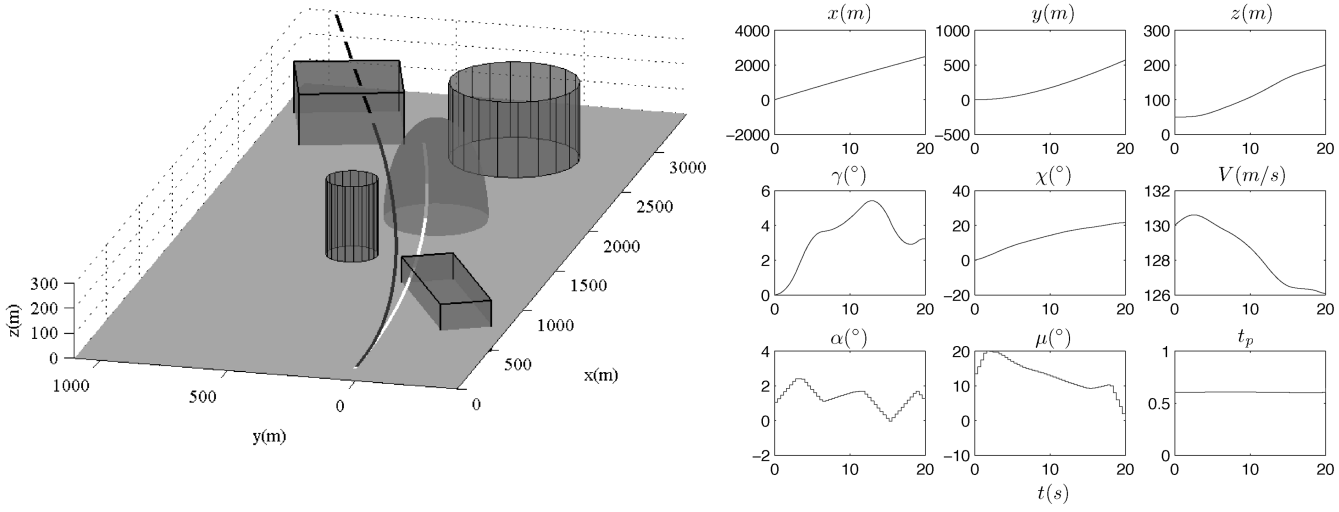


Fig. 10 Solve time = 0.51 s, 22 IPOPT iterations.

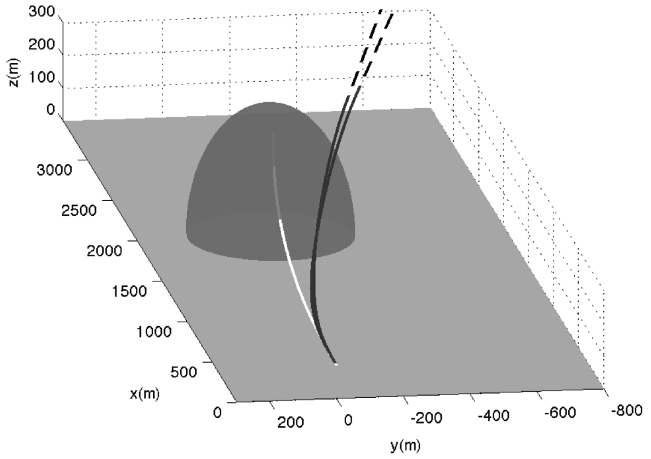


Fig. 11 Two avoidance trajectories with a 1.2 s time interval.

2. Hemispherical and Cylindrical PVs

The same situation as the previous case is considered but with a cylindrical PV of infinite height placed on the projected path of the previous solution. Figure 8 shows the additional end constraints in the optimization have caused a shift in the avoidance trajectory so that the projected path passes to the side of the cylindrical PV. A similar drop in the α profile is encountered as the previous case, again because the solution remains as close to the reference trajectory as possible. The number of iterations required to find the optimal solution in this case has increased resulting in a greater computation time of 0.44 s.

3. Polyhedral PV

Avoidance of a polyhedral PV is shown in Fig. 9. The additional constraints in the optimization due to the polyhedral obstacle result in a longer computation time per iteration. There are also more iterations before convergence when compared with a hemispherical PV.

4. Urban Environment Simulation

A number of PVs of varying types are placed in a representative urban environment scenario. Figure 10 shows the generated avoidance trajectory. The algorithm deals with the increased number of constraints and is still able to find a solution within a reasonable time.

Table 1 Computation times for forward problem

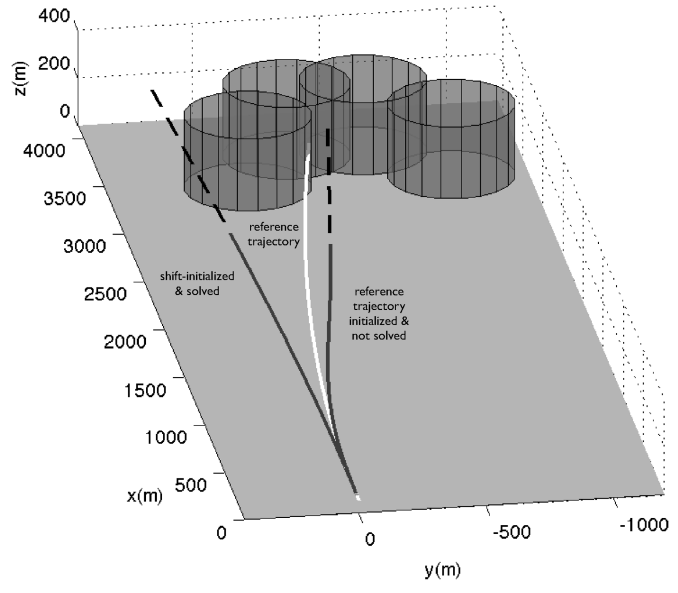
Initialization Point	Starting point	Solution time
Zero-Initalized	Pretrigger	0.78 s
Reference trajectory initialized	Pretrigger	0.24 s
Shift-Initialized	Pretrigger	0.26 s
Zero-Initialized	Posttrigger	0.75 s
Reference trajectory initialized	Posttrigger	0.20 s
Shift-Initialized	Posttrigger	0.20 s

B. Warm Starting

Warm starting the optimization with different initialization points affects the number of iterations required, and therefore the solution time for producing the avoidance trajectory. In the cases shown previously, the solution was initialized with the reference trajectory and the additional variables $u_N = u_{N-1}$ and $\eta_j = 0$. For the exact same problem of the single hemispherical PV, but now zero-initialized, the same solution (within the optimizer tolerance limits) is obtained after 39 iterations with a computation time of 0.68 s. This is almost 3 times the computation time compared with the reference trajectory initialized case and demonstrates the impact the initialization point can have on the efficiency of the optimization.

Assuming the outer-loop controller runs once every 1.2 s (to allow convergence of the optimization), the affect of initializing with a previous solution on the computation time can be investigated. Figure 11 shows a solution at t_0 of $\mathcal{P}(x(t_0), u(t_0))$. A second solution is shown for the forward problem $\mathcal{P}(x(1.2), u(1.2))$ and Table 1 shows the computation times for different initialization points. The pretrigger starting point refers to the case where the override has not been activated and the pilot is assumed to still be following the reference trajectory, i.e., the solution of $\mathcal{P}(x^r(1.2), u^r(1.2))$. The posttrigger case assumes that the avoidance trajectory generated at t_0 is being followed.

The results show the shift initialization does not necessarily reduce the computation time compared with the reference trajectory initialized case. This is because the optimal point is usually at the boundary of the feasible set, which is not an ideal initialization point for the next problem [37]. Strategies are available for perturbing the initialization point or taking a suboptimal solution to the original problem as the initializer, which is on the interior of the feasible set [38]. Given the computation time is reasonably low in any case, a greater concern is the ability of the algorithm to find any feasible solution if one is available. Figure 12 shows a scenario where four infinite height cylindrical PVs are present. The gradients of this



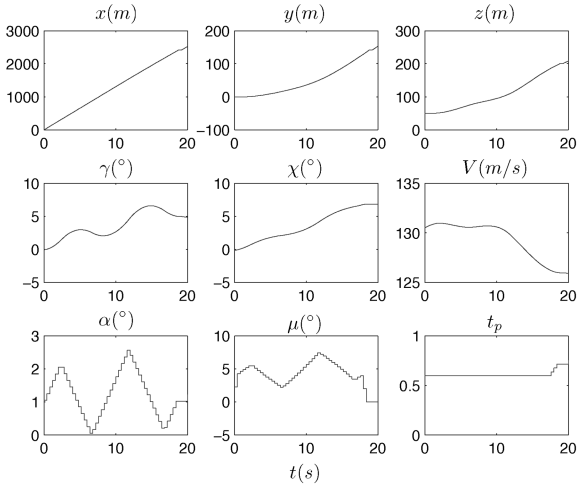


Fig. 12 Comparison of reference trajectory initialized and shift-initialized problems. The trajectory used for the initialization is shown on the right.

arrangement cause the reference trajectory initialized problem to fail (not converged after 500 iterations of the solver). In contrast the shift-initialized solution, which has been initialized from previous solutions progressively approaching this initial condition, has solved demonstrating the advantages of shift initialization for the scheme.

V. Conclusions

The paper presents a direct multiple shooting scheme for producing avoidance trajectories for aircraft with computation times sufficiently fast for the real-time system. Nondifferentiable polytopic obstacle constraints are dealt with via a dualization method that preserves continuity and allows the use of a gradient-based optimization method. The paper has also described how dual variables can be used to ensure positive invariance of the algorithm. The combination of the positively invariant terminal condition and warm starting via shift initialization has been used to demonstrate how to prevent the aircraft entering a trap situation where a solution is no longer available when generating avoidance maneuvers. The direct multiple shooting scheme has the potential to be applied to a broad class of autonomous systems including unmanned aircraft. It provides advantages over past work in this application, allowing a realistic vehicle model, flexible object constraints which can be time varying (i.e., moving) and dynamic model constraints based on the current vehicle state.

Discussions throughout this paper have concentrated on the outer-loop trajectory generation part of the overall counterhijack system. Readers will note that a simplified 3 deg-of-freedom model has been used to generate these trajectories and inner-loop controllers are required for robust tracking. Currently, for demonstration purposes, proportional-integral-derivative controllers are used to follow the desired trajectories and ongoing work involves investigations of trigger mechanisms for the override system and conducting experiments with airline pilots in a full motion simulator for the complete proof-of-concept counterhijack system.

Acknowledgments

Funding for this research has been provided by QinetiQ and the UK Engineering and Physical Sciences Research Council. The authors also wish to thank Rod Angel of QinetiQ for valuable discussions.

References

[1] Kuchar, J. K., and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 179–189. doi:10.1109/6979.898217

[2] Moir, I., and Seabridge, A., *Civil Avionics Systems*, Professional Engineering Publishing, London, 2003.

[3] Rawlings, J. B., and Mayne, D. Q., *Model Predictive Control Theory and Design*, Nob Hill Publishing, Madison, WI, 2009.

[4] LaValle, S. M., and Konkimalla, P., "Algorithms for Computing Numerical Optimal Feedback Motion Strategies," *International Journal of Robotics Research*, Vol. 20, 2001, pp. 729–752. doi:10.1177/02783640122067633

[5] Redding, J., Amin, J. N., and Boskovic, J. D., "A Real-Time Obstacle Detection and Reactive Path Planning System for Autonomous Small-Scale Helicopters," AIAA 2007-6413, 2007.

[6] LaValle, S. M., "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Iowa State Univ. Technical Rept. TR 98-11, 1998.

[7] LaValle, S. M., and Kuner, J. J., Jr., "Randomized Kinodynamic Programming," *The International Journal of Robotics Research*, Vol. 20, No. 5, 2001, pp. 378–400. doi:10.1177/02783640122067453

[8] Frazzoli, E., Dahleh, M. A., and Feron, E., "Real-Time Motion Planning for Agile Autonomous Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 1, 2002, pp. 116–129. doi:10.2514/2.4856

[9] Dever, C., Mettler, B., Feron, E., Popovic, J., and McConley, M., "Nonlinear Trajectory Generation for Autonomous Vehicles via Parameterized Maneuver Classes," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 2, 2006, pp. 289–302. doi:10.2514/1.13400

[10] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained Model Predictive Control: Stability and Optimality," *Automatica*, Vol. 36, No. 6, 2000, pp. 789–814. doi:10.1016/S0005-1098(99)00214-9

[11] Singh, L., and Fuller, J., "Trajectory Generation for a UAV in Urban Terrain, Using Nonlinear MPC," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 2001, pp. 2301–2308.

[12] Keviczky, T., and Balas, G. J., "Receding Horizon Control of an F-16 Aircraft: A Comparative Study," *Control Engineering Practice*, Vol. 14, No. 9, 2006, pp. 1023–1033. doi:10.1016/j.conengprac.2005.06.003

[13] Kuwata, Y., Schouwenaars, T., Richards, A., and How, J., "Robust Constrained Receding Horizon Control for Trajectroy Planning," , AIAA 2005-6079, 2005.

[14] Slegers, N., Kyle, J., and Costello, M., "Nonlinear Model Predictive Control Technique for Unmanned Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 5, 2006, pp. 1179–1188. doi:10.2514/1.21531

[15] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., and Cook, A. K., "A Prototype of an Autonomous Controller for a Quadrotor UAV," *Proceedings of the European Control Conference*, European Union Control Association, Kos, Greece, 2007.

[16] Lee, E. A., "Soft Walls: Modifying Flight Control Systems to Limit the Flight Space of Commercial Aircraft," Univ. of CA Berkeley Technical Rept. UCB/ERL MO1/31, 2001.

[17] Patel, R. B., Goulart, P. J., and Serghides, V., "Real-Time Trajectory Generation for Aircraft Avoidance Maneuvers," AIAA 2009-5623, 2009.

[18] Diehl, M., Bock, H. G., and Schlöder, J. P., "A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control," *SIAM Journal on Control and Optimization*, Vol. 43, No. 5, 2005, pp. 1714–1736. doi:10.1137/S0363012902400713

[19] Plitt, K. J., and Bock, H. G., "A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems," *International Federation of Automatic Control*, Vol. 4, 1984, pp. 242–247.

[20] Betts, J. T., "Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, Vol. 21, No. 2, 1998, pp. 193–207. doi:10.2514/2.4231

[21] Sundar, S., and Schiller, Z., "Time-Optimal Obstacle Avoidance," *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE Publications, Piscataway, NJ, 1995, pp. 3075–3080.

[22] Schiller, Z., "On-Line Suboptimal Obstacle Avoidance," *Proceedings of the IEEE International Conference on Robotics and Automation*, IEEE Publications, Piscataway, NJ, 1999, pp. 335–340.

[23] Hagenaars, H. L., Imura, J., and Nijmeijer, H., "Approximate Continuous-Time Optimal Control in Obstacle Avoidance by Time/Space Discretization of Nonconvex State Constraints," *Proceedings of the IEEE International Conference on Control Applications*, IEEE Publications, Piscataway, NJ, 2004, pp. 878–883.

[24] Kim, H. J., Shim, D. H., and Sastry, S., "Nonlinear Model Predictive Tracking Control for Rotorcraft-Based Unmanned Aerial Vehicles," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 2002, pp. 3576–3080.

[25] Richards, A. G., and How, J. P., "Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 2002, pp. 1936–1941.

[26] Rakovic, S. V., and Mayne, D. Q., "Robust Time Optimal Obstacle Avoidance Problem for Constrained Discrete Time Systems," *Proceedings of the 44th IEEE Conference on Decision and Control*, IEEE Publications, Piscataway, NJ, 2005.

[27] Rockafellar, R. T., and Wets, R. J. B., *Variational Analysis*, Springer–Verlag, New York, 2004.

[28] Wanka, G., Bot, R. I., and Vargyas, E., "Duality for Location Problems with Unbounded Unit Balls," *European Journal of Operational Research*, Vol. 179, 2007, pp. 1252–1265. doi:10.1016/j.ejor.2005.09.048

[29] Bartlett, R. A., Wächter, A., and Biegler, L. T., "Active Set vs Interior Point Strategies for Model Predictive Control," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 2000, pp. 4229–4233.

[30] Wächter, A., and Biegler, L. T., "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming," *Mathematical Programming*, Vol. 106, Springer, Berlin, 2006, pp. 25–57.

doi:10.1007/s10107-004-0559-y
[31] Wright, S., and Nocedal, J., *Numerical Optimization*, Springer, New York, 1999.
[32] Patel, R. B., and Goulart, P. J., "The Design of Trigger Mechanisms for Aircraft Collision Avoidance Maneuvers," AIAA 2010-8167, 2010.
[33] Boyd, S., and Vandenberghe, L., *Convex Optimization*, Cambridge Univ. Press, Cambridge, England, 2004.
[34] Diehl, M., Ferreau, H. J., and Haverbeke, N., "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation," *International Workshop on Assessment and Future Directions of NMPC*, Pavia, Italy, 2008.
[35] Jackson, P. A., editor, *Jane's All the World's Aircraft*, Jane's Information Group, Alexandria, VA, 2006.
[36] Pratt, D. T., Heiser, W. H., and Mattingly, J. D., *Aircraft Engine Design*, AIAA, Reston, VA, 2002.
[37] Gondzio, J., "Warm Start of the Primal-Dual Method Applied in the Cutting-Plane Scheme," *Mathematical Programming*, Vol. 83, No. 1, 1998, pp. 125–143.
doi:10.1007/BF02680554
[38] Wu, Y. C., and Debs, A. S., "Initialization, Decoupling, Hot Start, and Warm Start in Direct Nonlinear Interior Point Algorithm for Optimal Power Flows," *IEEE Proceedings: Generation, Transmission and Distribution*, Vol. 148, IEEE, Piscataway, NJ, 2001, pp. 67–75.