A Continuous Optimization Approach to Drift Counteraction Optimal Control*

Sunbochen Tang[1], Nan Li[1], Ilya Kolmanovsky[1], Robert Zidek[1]

***Abstract*— Drift counteraction optimal control (DCOC) aims at optimizing control to maximize the time (or a yield) until the system trajectory exits a prescribed set, which may represent safety constraints, operating limits, and/or efficiency requirements. To DCOC problems formulated in discrete time, conventional approaches were based on dynamic programming (DP) or mixed-integer programming (MIP), which could become computationally prohibitive for higher-order systems. In this paper, we propose a novel approach to discrete-time DCOC problems based on a nonlinear programming formulation with purely continuous variables. We show that this new continuous optimization-based approach leads to the same exit time as the conventional MIP-based approach, while being computationally more efficient than the latter. This is also illustrated by numerical examples representing the drift counteraction control for an indoor airship.**

I. Introduction

During the operation of many systems, it is desired to maintain the system state to stay within a prescribed operating region for safety and/or efficiency reasons. However, due to a cumulative effect of persistent disturbances and/or limited control authority and resources, the system state may drift and eventually get outside the desired operating region. The purpose of drift counteraction optimal control (DCOC) is to maximize the time (or, more generally, a functional representing the total yield) until the system trajectory exits the prescribed region [1], which is referred to as the *time-before-exit* in this paper.

The use of DCOC has been considered in a variety of applications, including for hybrid electric vehicle powertrain management [2], for satellite life extension [3], and for automated driving [4], [5]. As an example, a satellite operating in a low Earth orbit is subject to persistent atmospheric drag. Because the satellite has a finite amount of fuel on board, it will eventually fall out of orbit. In [3], DCOC is considered to maximize the time for the satellite to stay in its orbit (within a deviation tolerance).

DCOC problems can be formulated either in continuous time or in discrete time. Solving a continuous-time DCOC problem typically involves solving the Hamilton-Jacobi-Bellman (HJB) equation [6], [7], [8], which is in general very difficult. Therefore, recent studies have been focused on DCOC problems formulated in discrete time, where a larger set of optimization algorithms and computation tools are available. Approaches based on dynamic programming (DP) to discrete-time DCOC problems are proposed in [2], [9], [10]. Although these approaches can treat quite general DCOC problems in theory, they become computationally prohibitive when the system order is high, due to the *curse of dimensionality* associated with DP iterations [11]. It is shown in [12] that a discrete-time DCOC problem can be equivalently expressed as a mixed-integer programming (MIP) problem. In this MIP formulation, the number of integer variables grows linearly with the length of the planning horizon. Consequently, for problems requiring a longer planning horizon, such as in the case of higher-order systems, the resulting MIP problem is still quite difficult to handle due to the combinatorial worst-case complexity for optimizing over the integer variables [13]. Therefore, to reduce computational complexity, a continuous relaxation of this MIP problem is proposed in [12]. In particular, for linear systems with prescribed operating regions defined by affine constraints, the relaxed problem is a linear programming (LP) problem and hence can be efficiently solved. However, an optimal solution to this relaxed problem may only be a sub-optimal solution to the original MIP problem, i.e., not leading to the maximum time-before-exit that can be possibly achieved.

In this paper, we present a novel approach to discrete-time DCOC problems based on a nonlinear programming (NLP) formulation with purely continuous variables. The proposed NLP formulation is essentially an improved version of continuous relaxation of the MIP formulation introduced in [12]. However, we show that an optimal solution to this new relaxation guarantees to be an optimal solution to the original MIP problem, and hence achieves the maximum time-before-exit. The proposed approach is motivated by the NLP formulation for minimum-time optimal control problems recently introduced in [14]. In particular, the above optimality guarantee relies on a cost function with exponential weighting and is established by a sensitivity analysis. Furthermore, using numerical examples representing the drift counteraction control for an indoor airship, we show that solving for a maximum time-before-exit solution with this new NLP-based approach takes considerably less computation time than through solving the original MIP problem.

This paper is organized as follows. In Section II, we introduce the DCOC problem and present an equivalent MIP formulation developed in previous literature. In Section III, we propose a novel NLP formulation of the DCOC problem and analyze the relationship between the original DCOC problem and the new NLP problem. In Section IV, we use examples representing the drift counteraction control for an indoor airship to illustrate the proposed approach, where the

*This work has been supported by the National Science Foundation Award Number EECS 1931738.

[1]Sunbochen Tang, Nan Li, Ilya Kolmanovsky, and Robert Zidek are with the Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA. Emails: {tangsun, nanli, ilya, robzidek}@umich.edu

drift is caused by wind and temperature disturbances. The paper is concluded in Section V.

II. Drift Counteraction Optimal Control

Consider a system represented by the following discrete-time model,

$$x_{k+1} = f_d(x_k, u_k), \tag{1}$$

where $x_k \in \mathbb{R}^{n_x}$ denotes the state vector at the discrete time instant k, $u_k \in \mathbb{R}^{n_u}$ denotes the control input vector at k, and $f_d : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ is assumed to be a twice continuously differentiable function (i.e., $f_d \in C^2(\mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x})$).

Consider the following prescribed set for the state,

$$X = \{x \in \mathbb{R}^{n_x} : H(x) \le h\}, \tag{2}$$

where $H \in C^2(\mathbb{R}^{n_x} \to \mathbb{R}^{n_h})$ and $h \in \mathbb{R}^{n_h}$. Assume that the initial state x_0 belongs to X. The objective of drift counteraction optimal control (DCOC) is to maximize the time, k^*, before the system trajectory exits X, with control inputs u_k, $k = 0, 1, ...$, taking values in an admissible set $U \subset \mathbb{R}^{n_u}$, which is assumed to be compact and convex.

Formally, the DCOC problem can be formulated as follows,

$$\max_{u_0, u_1, ..., u_{N-1}} \kappa(x_0, \{u_k\}_{k=0}^{N-1}) \tag{3a}$$

$$\text{subject to } x_{k+1} = f_d(x_k, u_k), \tag{3b}$$

$$u_k \in U, \quad k = 0, 1, ..., N-1, \tag{3c}$$

where x_0 is a given initial condition, and $\kappa(x_0, \{u_k\}_{k=0}^{N-1})$ is defined as

$$\kappa(x_0, \{u_k\}_{k=0}^{N-1}) = \max\left\{k \in \mathbb{Z}_{[0,N]} : x_i \in X, i = 0, 1, ..., k\right\}. \tag{4}$$

It was shown in [12] that the above DCOC problem can be equivalently expressed as the following mixed-integer programming (MIP) problem,

$$\min_{\substack{u_0, u_1, ..., u_{N-1} \\ \delta_0, \delta_1, ..., \delta_N}} \sum_{k=0}^{N} \delta_k \tag{5a}$$

$$\text{subject to } x_{k+1} = f_d(x_k, u_k), \tag{5b}$$

$$u_k \in U, \tag{5c}$$

$$\delta_k \le \delta_{k+1}, \quad k = 0, 1, ..., N-1, \tag{5d}$$

$$H(x_k) \le h + \mathbf{1} M \delta_k, \tag{5e}$$

$$\delta_k \in \{0, 1\}, \quad k = 0, 1, ..., N, \tag{5f}$$

where x_0 is the given initial condition, $\mathbf{1}$ denotes the n_h-dimensional vector of ones, and $M > 0$ is a sufficiently large positive number. Specifically, a global optimizer $(\{u_k^*\}_{k=0}^{N-1}, \{\delta_k^*\}_{k=0}^{N})$ of the MIP problem (5) provides a global optimizer $\{u_k^*\}_{k=0}^{N-1}$ of (3), and furthermore, the maximum time-before-exit, $\kappa^*(x_0) = \kappa(x_0, \{u_k^*\}_{k=0}^{N-1})$, satisfies $\kappa^*(x_0) = \max\{k : \delta_k^* = 0\}$ (see Theorem 1, [12]).

Solving mixed-integer problems is computationally challenging [13]. Therefore, it was proposed in [12] to approximately solve the MIP problem (5) through solving a relaxed version of (5), which dropped the integer constraints (5f) and was thereby a continuous optimization problem. Although such a technique demonstrated reasonably good performance in simulation examples, there was no guarantee that the control input sequence $\{u_k\}_{k=0}^{N-1}$ obtained from the relaxed problem necessarily lead to the maximum time-before-exit κ^* of the original DCOC problem (3). The major contribution of this paper is to introduce an improved version of the relaxed problem such that the above guarantee is achieved with rigorous proof.

III. Continuous Optimization Approach to DCOC

Consider the following problem,

$$\min_{\substack{u_0, u_1, ..., u_{N-1} \\ \varepsilon_0, \varepsilon_1, ..., \varepsilon_N}} J_{(6)} = \sum_{k=0}^{N} \theta^{N-k} \varepsilon_k, \tag{6a}$$

$$\text{subject to } x_{k+1} = f_d(x_k, u_k), \tag{6b}$$

$$u_k \in U, \tag{6c}$$

$$0 \le \varepsilon_k \le \varepsilon_{k+1}, \quad k = 0, 1, ..., N-1, \tag{6d}$$

$$H(x_k) \le h + \mathbf{1} M \varepsilon_k, \quad k = 0, 1, ..., N, \tag{6e}$$

where $\varepsilon_k \in \mathbb{R}$, and $\theta > 1$ is a weighting parameter which is chosen to be sufficiently large.

The problem (6) is a nonlinear programming (NLP) problem with purely continuous variables. A connection between problems (5) and (6) is stated in the following theorem.

Theorem 1: Given $x_0 \in X$, there exists a number $\theta_0 > 1$ such that if $\theta \ge \theta_0$, then any global minimizer of (6), $(\{u_k^*\}_{k=0}^{N-1}, \{\varepsilon_k^*\}_{k=0}^{N})$, satisfies $\varepsilon_k^* = 0$ for all $k = 0, ..., \kappa^*(x_0)$, where $\kappa^*(x_0)$ is the maximum time-before-exit introduced below (5). In turn, the control input sequence $\{u_k^*\}_{k=0}^{N-1}$ is a global optimizer of the DCOC problem (3).

To prove Theorem 1, we introduce a cost function $\phi(\varepsilon_{\kappa^*(x_0)+1}, ..., \varepsilon_N) \triangleq \sum_{k=\kappa^*(x_0)+1}^{N} \theta^{N-k} \varepsilon_k$ and start with the following related problem,

$$\min_{\substack{u_0, u_1, ..., u_{N-1} \\ \varepsilon_0, \varepsilon_1, ..., \varepsilon_N}} \phi(\varepsilon_{\kappa^*(x_0)+1}, ..., \varepsilon_N) = \sum_{k=\kappa^*(x_0)+1}^{N} \theta^{N-k} \varepsilon_k \tag{7a}$$

$$\text{subject to } x_{k+1} = f_d(x_k, u_k), \tag{7b}$$

$$u_k \in U, \tag{7c}$$

$$\varepsilon_k \le \varepsilon_{k+1}, \quad k = \kappa^*(x_0), ..., N-1, \tag{7d}$$

$$H(x_k) \le h + \mathbf{1} M \varepsilon_k, \quad k = 0, 1, ..., N, \tag{7e}$$

$$\varepsilon_k = \eta_k, \quad k = 0, 1, ..., \kappa^*(x_0), \tag{7f}$$

where η_k, $k = 0, 1, ..., \kappa^*(x_0)$, are parameters. For convenience, we let $\varepsilon = [\varepsilon_{\kappa^*(x_0)+1}, ..., \varepsilon_N]^\top$, $\eta = [\eta_0, \eta_1, ..., \eta_{\kappa^*(x_0)}]^\top$, and $\eta^{(k)} = \eta_k e_k$, where e_k, $k = 0, 1, ..., \kappa^*(x_0)$, are the standard basis vectors of $\mathbb{R}^{\kappa^*(x_0)+1}$. It is clear that the minimizers of (7) depend parametrically on η.

We now make the following assumption:

Assumption 1: For $\eta = \mathbf{0}$, we let $z(\mathbf{0}) = \left(\{u_k(\mathbf{0})\}_{k=0}^{N-1}, \{\varepsilon_k(\mathbf{0})\}_{k=0}^{N}\right) \in \mathbb{R}^{n_z}$ denote a minimizer of (7) and let $\lambda(\mathbf{0}) \in \mathbb{R}^{n_\lambda}$ denote its associated Lagrange multiplier vector. The pair $(z(\mathbf{0}), \lambda(\mathbf{0}))$ is assumed to satisfy the strong second-order sufficient conditions (see Theorem 2 of [15]).

Under Assumption 1 and according to Theorem 3 of [15], there exists a neighborhood of $\mathbf{0}$, $V \subset \mathbb{R}^{\kappa^*(x_0)+1}$, and

continuously differentiable functions $z: V \to \mathbb{R}^{n_z}$ and $\lambda: V \to \mathbb{R}^{n_\lambda}$ such that for all $\eta \in V$, the pair $(z(\eta), \lambda(\eta))$ satisfies the strong second-order sufficient conditions. In this case, the following sensitivity result holds [15]: For $k = 0, 1, ..., \kappa^*(x_0)$,

$$\lim_{\eta_k \to 0^+} \frac{\phi(\varepsilon(\eta^{(k)})) - \phi(\varepsilon(\mathbf{0}))}{\eta_k - 0} = -\lambda_k(\mathbf{0}), \tag{8}$$

where $\lambda_k(\mathbf{0})$ is the Lagrange multiplier associated with the equality constraint $\varepsilon_k = \eta_k = 0$.

Furthermore, we make the following assumption on $\varepsilon(\eta)$:

Assumption 2: There exists $L > 0$ such that,

$$\left| \lim_{\eta_k \to 0^+} \frac{\varepsilon_i(\eta^{(k)}) - \varepsilon_i(\mathbf{0})}{\eta_k - 0} \right| \leq L, \tag{9}$$

for all $i = \kappa^*(x_0) + 1, ..., N$, all $k = 0, 1, ..., \kappa^*(x_0)$, and all $\theta > 1$.

We now consider the following problem, which replaces the constraints $\varepsilon_k = \eta_k = 0$, $k = 0, 1, ..., \kappa^*(x_0)$, in (7) with the penalties $\theta^{N-k}|\varepsilon_k|$ in the cost function,

$$\min_{\substack{u_0, u_1, ..., u_{N-1} \\ \varepsilon_0, \varepsilon_1, ..., \varepsilon_N}} J_{(10)} = \sum_{k=0}^{\kappa^*(x_0)} \theta^{N-k}|\varepsilon_k| + \phi(\varepsilon_{\kappa^*(x_0)+1}, ..., \varepsilon_N) \tag{10a}$$

$$\text{subject to } x_{k+1} = f_d(x_k, u_k), \tag{10b}$$

$$u_k \in U, \tag{10c}$$

$$\varepsilon_k \leq \varepsilon_{k+1}, \quad k = \kappa^*(x_0), ..., N-1, \tag{10d}$$

$$H(x_k) \leq h + \mathbf{1}M\varepsilon_k, \quad k = 0, 1, ..., N. \tag{10e}$$

The relationship between the problems (7) and (10) is stated in the following lemma,

Lemma 1: Under Assumptions 1 and 2, there exists $\theta_0 > 1$ such that if $\theta \geq \theta_0$, then (7) with $\eta = \mathbf{0}$ and (10) are equivalent, i.e., the minimizers of (7) with $\eta = \mathbf{0}$ are all minimizers of (10), and vice versa.

Proof: For each $k = 0, 1, ..., \kappa^*(x_0)$, the combination of (8) and (9) yields the following bound on $\lambda_k(\mathbf{0})$,

$$\begin{aligned}
|\lambda_k(\mathbf{0})| &= \left| \lim_{\eta_k \to 0^+} \frac{\phi(\varepsilon(\eta^{(k)})) - \phi(\varepsilon(\mathbf{0}))}{\eta_k - 0} \right| \\
&= \left| \lim_{\eta_k \to 0^+} \frac{\sum_{i=\kappa^*(x_0)+1}^{N} \theta^{N-i}\left(\varepsilon_i(\eta^{(k)}) - \varepsilon_i(\mathbf{0})\right)}{\eta_k - 0} \right| \\
&\leq \sum_{i=\kappa^*(x_0)+1}^{N} \theta^{N-i} \left| \lim_{\eta_k \to 0^+} \frac{\varepsilon_i(\eta^{(k)}) - \varepsilon_i(\mathbf{0})}{\eta_k - 0} \right| \\
&\leq \frac{\theta^{N-\kappa^*(x_0)} - 1}{\theta - 1} L < \frac{L}{\theta - 1} \theta^{N-\kappa^*(x_0)}.
\end{aligned}$$

Now let $\theta_0 = L + 1 > 1$. If $\theta \geq \theta_0$, then for each $k = 0, 1, ..., \kappa^*(x_0)$, the weight associated with $|\varepsilon_k|$ in the cost function (10a), θ^{N-k}, satisfies $\theta^{N-k} \geq \theta^{N-\kappa^*(x_0)} \geq \frac{L}{\theta-1}\theta^{N-\kappa^*(x_0)} > |\lambda_k(\mathbf{0})|$. This means $\theta^{N-k}|\varepsilon_k|$ is an exact penalty function [16] for the constraint $\varepsilon_k = \eta_k = 0$. Since (10) treats the constraints $\varepsilon_k = \eta_k = 0$, $k = 0, 1, ..., \kappa^*(x_0)$, in (7) with exact penalties, according to Theorem 14.3.1 of [16], the second-order sufficient conditions of (7) and (10) are equivalent. ∎

Now we are ready to prove Theorem 1. We will show that any global minimizer of (6) must also be a global minimizer of (10). Then, according to the equivalence relation between (7) with $\eta = \mathbf{0}$ and (10), such a global minimizer of (6) must satisfy the conditions $\varepsilon_k = \eta_k = 0$ for $k = 0, 1, ..., \kappa^*(x_0)$ in (7f). Hence, Theorem 1 is proven.

Proof: Let $z^* = (\{u_k^*\}_{k=0}^{N-1}, \{\varepsilon_k^*\}_{k=0}^{N})$ be a global minimizer of (6). Since the set of constraints of (10) is a subset of the constraints of (6), z^* is a feasible point of (10). Let $z' = (\{u_k'\}_{k=0}^{N-1}, \{\varepsilon_k'\}_{k=0}^{N})$ be a global minimizer of (10). By Lemma 1, z' is at least a local minimizer of (7) with $\eta = \mathbf{0}$, and hence satisfies $\varepsilon_k' = \eta_k = 0$ for $k = 0, 1, ..., \kappa^*(x_0)$. In turn, z' is a feasible point of (6). Since z^* and z' are global minimizers of (6) and (10), respectively, and are feasible points of (10) and (6), respectively, we have $J_{(6)}(z^*) \leq J_{(6)}(z')$ and $J_{(10)}(z') \leq J_{(10)}(z^*)$. However, for any point z that is feasible to (6) (and hence is also feasible to (10)), we have

$$\begin{aligned}
J_{(6)}(z) &= \sum_{k=0}^{\kappa^*(x_0)} \theta^{N-k}\varepsilon_k + \sum_{k=\kappa^*(x_0)+1}^{N} \theta^{N-k}\varepsilon_k \\
&= \sum_{k=0}^{\kappa^*(x_0)} \theta^{N-k}|\varepsilon_k| + \sum_{k=\kappa^*(x_0)+1}^{N} \theta^{N-k}\varepsilon_k = J_{(10)}(z).
\end{aligned}$$

Note that $\varepsilon_k \geq 0$ due to the constraints in (6d). This implies $J_{(6)}(z^*) \leq J_{(6)}(z') = J_{(10)}(z') \leq J_{(10)}(z^*) = J_{(6)}(z^*)$. Therefore, we have $J_{(6)}(z^*) = J_{(6)}(z')$ and $J_{(10)}(z') = J_{(10)}(z^*)$, which implies z' and z^* are not only feasible points but indeed also global minimizers of (6) and (10), respectively. This proves Theorem 1. ∎

Theorem 1 guarantees that a global minimizer of the NLP problem (6) provides a control input sequence $\{u_k^*\}_{k=0}^{N-1}$ that maximizes the time-before-exit (4), i.e., solves the DCOC problem (3). In practice, NLP problems are typically solved using gradient-based algorithms such as the interior-point method and the sequential quadratic programming method, which converge to only local minimizers. In what follows we extend Theorem 1 to results of local minimizers. This extension relies on the following assumption:

Assumption 3: Let $Z \subset \mathbb{R}^{n_z}$ denote the feasible region of (6), characterized by the constraints (6b)-(6e), and let $z^* = (\{u_k^*\}_{k=0}^{N-1}, \{\varepsilon_k^*\}_{k=0}^{N})$ be a global minimizer of (6). It is assumed that for any $z_0 \in Z$, there exists $r_0(z_0) > 0$ such that $z_0 + r(z^* - z_0) \in Z$ for all $r \in [0, r_0(z_0)]$.

Assumption 3 holds for many cases. For instance, if f_d represents a linear system and all components of H are convex functions, then the feasible region Z is convex, and in this case Assumption 3 holds true. More generally, if Z is star-shaped with z^* as a star center, then Assumption 3 also holds true.

With Assumption 3, we have the following result:

Theorem 2: Under Assumption 3, all local minimizers of (6) are indeed global minimizers.

Proof: Let z' be a local minimizer of (6). Then, there exists $r_1 \in (0, r_0(z')]$ such that $J_{(6)}(z') \leq J_{(6)}(z)$ for all $z \in Z$ with $\|z - z'\|_2 \leq r_1\|z^* - z'\|_2$, where $\|\cdot\|_2$ refers to the Euclidean norm. Let $z'' = z' + r_1(z^* - z')$, which satisfies $z'' \in Z$, by Assumption 3, and $\|z'' - z'\|_2 = r_1\|z^* - z'\|_2$. Thus,

$J_{(6)}(z') \leq J_{(6)}(z'')$. However, because the cost function $J_{(6)}$ is linear in z, we also have $J_{(6)}(z'') = J_{(6)}(z' + r_1(z^* - z')) = r_1 J_{(6)}(z^*) + (1 - r_1) J_{(6)}(z') \leq r_1 J_{(6)}(z') + (1 - r_1) J_{(6)}(z') = J_{(6)}(z')$, where the inequality in the middle is due to the fact that z^* is a global minimizer. In this case, it must hold that $J_{(6)}(z') = J_{(6)}(z'') = J_{(6)}(z^*)$. This proves that z' is indeed a global minimizer of (6). ■

Theorem 2 guarantees that, under Assumption 3, any local minimizer of the NLP problem (6), obtained using gradient-based algorithms, solves the DCOC problem (3).

IV. Numerical Examples

Numerical examples are developed based on a drift counteraction control problem of an indoor airship to illustrate the proposed NLP-based approach (6), and also to compare this new approach with the previous MIP-based approach (5).

A. Dynamics Model of an Indoor Airship

The model representing the indoor airship dynamics is derived based on the one developed in [17]. The airship is depicted in Fig. 1, where two motors are attached to a horizontal bar below the ellipsoidal hull, and one motor is mounted on the vertical fin at the back. The center of gravity is assumed to be below the center of buoyancy. As a result, most of the pitch and roll motion will be filtered out, i.e., the angular velocities about the x- and y-axes of the body-fixed frame are assumed to be 0.

Under the assumption $\omega_x^b = \omega_y^b = 0$, the transformation from the ground-fixed frame to the body-fixed frame reduces to a rotation ϕ about the z-axis. Define the state and control input vectors as follows,

$$x = [x^f, y^f, z^f, \phi, v_x^b, v_y^b, v_z^b, \omega_z]^\top, \quad u = [F_1, F_2, F_3]^\top, \quad (11)$$

where (x^f, y^f, z^f) represents the position of the airship in the ground-fixed frame, and (v_x^b, v_y^b, v_z^b) represents the velocity of the airship in its body-fixed frame.

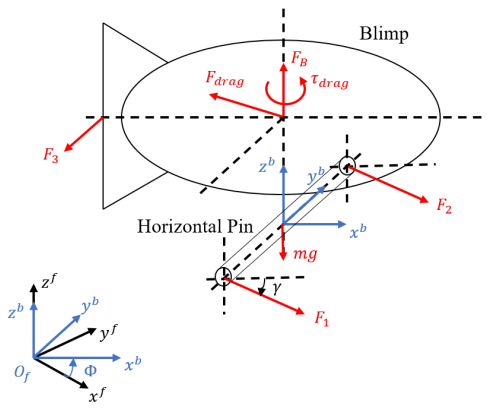

Fig. 1: A diagram of the airship model.

Then, the airship dynamics can be described by the following set of equations of motion,

$$\begin{cases} \dot{x}^f = v_x^b \cos\phi - v_y^b \sin\phi \\ \dot{y}^f = v_x^b \sin\phi + v_y^b \cos\phi \\ \dot{z}^f = v_z^b \\ \dot{\phi} = \omega_z \\ \dot{v}_x^b = \frac{1}{m_x}\left((F_1 + F_2)\cos\gamma + F_{drag,xb}\right) \\ \dot{v}_y^b = \frac{1}{m_y}\left(F_3 + m_x v_x^b \omega_z + F_{drag,yb}\right) \\ \dot{v}_z^b = \frac{1}{m_z}\left((F_1 + F_2)\sin\gamma + \rho V g - mg + F_{drag,zb}\right) \\ \dot{\omega}_z = \frac{(F_1 - F_2) l_y \cos\gamma - F_3 l_x + (m_x - m_y) v_x^b v_y^b + \tau_{drag,zb}}{J_z}, \end{cases} \quad (12)$$

where m_x, m_y, m_z, J_z are the augmented masses and inertia [18], γ is the tilt angle of the lower-mounted propellers, $F_{drag,ib}, \tau_{drag,ib}, i = x, y, z$ are the drag forces acting along each axis of the body-fixed frame, F_1, F_2, F_3 are the propeller thrusts, and l_x, l_y are the acting lengths of the propellers. Note that the air in the room is assumed to be dry air with constant atmosphere pressure. By the ideal gas law, the air density ρ can be calculated as follows,

$$\rho = p/(R_{dry\,air} T), \quad p = 1[atm] = 101.325[kPa], \quad (13)$$

where T represents the temperature, and $R_{dry\,air}$ is the specific gas constant for dry air.

Since it is unclear whether laminar or turbulent boundary layers are dominant at the airship surfaces in an indoor environment, we model the drag force as a second order Taylor series of the airship's relative speed to the wind [18]. Suppose the ambient wind has a spatially distributed velocity profile $v_w(x, y, z) = [w_x, w_y, w_z]^\top$ in the ground-fixed frame, the drag forces can be modeled as follows,

$$\begin{aligned} F_{drag,xb} &= B_x(v_x^b - w_x\cos\phi - w_y\sin\phi) \\ &\quad + D_x(v_x^b - w_x\cos\phi - w_y\sin\phi)^2, \\ F_{drag,yb} &= B_y(v_y^b + w_x\sin\phi - w_y\cos\phi) \\ &\quad + D_y(v_y^b + w_x\sin\phi - w_y\cos\phi)^2, \\ F_{drag,zb} &= B_z(v_z^b - w_z) + D_z(v_z^b - w_z)^2, \\ \tau_{drag,zb} &= B_\tau\omega_z + D_\tau\omega_z^2, \end{aligned} \quad (14)$$

where $B_x, B_y, B_z, B_\tau, D_x, D_y, D_z, D_\tau$ are negative constants proportional to their corresponding cross-section areas.

B. Counteraction of Spatially Distributed Wind Disturbance

The first example considers the case where the airship is operating in a hallway where wind disturbance is present. Due to the friction caused by the side walls, the wind velocity is assumed to be distributed linearly from the center to both sides, as described by the following function,

$$v_w(x^f, y^f, z^f) = \left[0, -6 + |x^f|, 0\right]^\top [m/s]. \quad (15)$$

The prescribed set for the state vector and the admissible set for the control input are defined in (16). Note that the maximum propeller thrust cannot counteract the wind drift, and therefore, the state vector will eventually exit the prescribed set X.

$$X = \{x : |x(1)| \leq 3, |x(2)| \leq 3\}, \quad U = \{u : ||u||_\infty \leq 2\}. \quad (16)$$

We consider an initial condition $x(0) = [0,0,2,\pi/2, 0,0,0,0]^\top$, with which the airship is heading toward the wind. To solve the DCOC problem, we first discretize the continuous-time model (12) using the solver "CVODE" [19] with a sampling period $\Delta T = 0.06[s]$ to get the function f_d. Then, when formulating the problem (6), we choose $N = 50$ and $\theta = 1.6$.

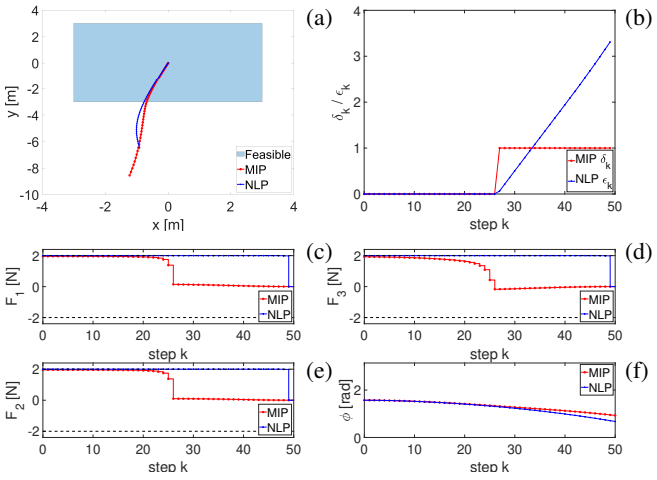

Fig. 2: Counteraction of wind drift.

The simulation results are presented in Fig. 2. The projections of the airship trajectories onto the $x^f - y^f$ plane are shown in (a). It can be observed that the trajectories given by the MIP (red line) and the NLP (blue line) formulations are almost identical before exiting the prescribed set X (light blue region). The time histories of the obtained δ_k from (5) and the obtained ε_k from (6) are shown in (b). It can be observed that both formulations result in the same time-before-exit $\kappa^*(x_0) = 26$. The control input trajectories and the airship heading angle trajectory ϕ_k are shown in (c, d, e) and (f), respectively. It can be observed from these control and state trajectories that the lower-mounted propeller pair is used to push the airship against the wind drift, while the back-mounted propeller is used to turn the airship and drive it to the region where the wind speed is lower. As a result, the controls represent a strategy to counteract wind disturbance by both pushing against head wind and moving to a position where the disturbance is smaller. Note also that because the MIP formulation (5) concerns itself only with whether the constraints are violated or not, while the NLP formulation (6) also penalizes the extents of constraint violation, their resulting state and control trajectories are not exactly the same, especially after the time-before-exit $\kappa^*(x_0) = 26$.

C. Counteraction of Temperature Disturbance

The second example considers the case where the airship is operated in a ventilated room. During a summer day, part of the room under direct sunlight can be heated up, where high temperature results in low air density, leading to a lack of buoyancy for the airship. Typically the temperature gradient in a ventilated room is around $2[^\circ C/m]$, along with low speed air flow, around $0.3[m/s]$ [20]. We consider the following spatial temperature distribution and the air flow profile in the room,

$$v_w(x^f, y^f, z^f) = [-0.3, 0, 0]^\top [m/s],$$
$$T(x^f, y^f, z^f) = 273 + 25 - 3x^f + 2z^f [K]. \quad (17)$$

The prescribed set for the state vector and the admissible set for the control input are defined as follows,

$$X = \{x : |x(1) + 2| \le 1, |x(2)| \le 2, |x(3) - 2| \le 0.1\},$$
$$U = \{u : ||u||_\infty \le 2\}. \quad (18)$$

We consider an initial condition $x(0) = [-2.5, 0.5, 2, 0,0,0,0,0]^\top$, with which the airship is heading toward the wind. After discretizing the model (12) with a sampling period $\Delta T = 0.1[s]$, we choose $N = 60$ and $\theta = 1.6$ to formulate the problem (6).

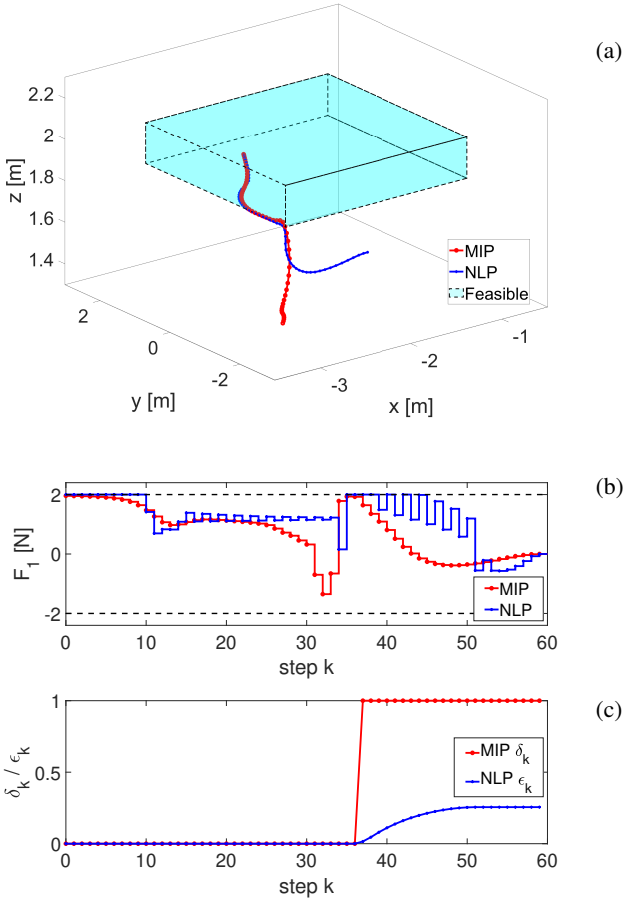

Fig. 3: Counteraction of spatial temperature disturbance.

The simulation results are shown in Fig. 3. The red and blue lines represent trajectories given by the MIP and the NLP formulations, respectively. The (x^f, y^f, z^f) trajectories, control input $F_1(k)$ time histories, and the time histories of δ_k and ε_k obtained by the two formulations are presented in (a), (b), and (c), respectively. The control input trajectories generated by both formulations result in the same time-before-exit $\kappa^*(x_0) = 36$. For both formulations, system states x^f, y^f are maintained within their prescribed ranges over the horizon while the range $|z^f - 2| \le 0.1$ is violated at $k = 37$. Although the state trajectories of both simulations are quite

similar before they exit the prescribed set X, significant discrepancies between the control input trajectories of the MIP and NLP formulations can be observed in (b) after $k = 26$. Note that the MIP formulation (5) typically has many distinct solutions for the continuous variables $\{u_k\}_{k=0}^{N-1}$, since the cost function is determined only by the integer variables $\{\delta_k\}_{k=0}^{N}$. For instance, if a solution for $\{u_k\}_{k=0}^{N-1}$ is perturbed by a little bit, the cost value will be very likely unchanged. Depending on the numerical solver used to solve (5) as well as the initial guess for the solution, the MIP formulation could lead to different solutions, which is a reason for the discrepancies observed in (b).

D. Computation Time Comparison

The computation times required to solve one instance of the MIP and the NLP formulations are reported in Table I. The numerical computations are implemented using the open source tool "CasADi" [21]. We use the solvers "bonmin" and "ipopt" embedded in CasADi to solve the MIP and the NLP problems. All results reported here correspond to the MATLAB 2019b platform on a Windows 10 PC with an i5-7400 CPU and 16GB RAM.

Example	MIP Time [s]	NLP Time [s]
B	91	6
C	1190	11

TABLE I: Computation time for the numerical examples.

For each example, the computation time for solving the NLP problem (6) is significantly less than that for solving the MIP problem (5). Furthermore, the computation time for solving the MIP problem of Example C is 12 times longer than that of Example B, while this ratio corresponding to the NLP formulation is below 2. These results illustrate that the MIP formulation can be computationally challenging, especially when treating complex systems with a long horizon; while the NLP formulation is much more computationally scalable.

V. Conclusions

A novel continuous optimization-based approach to drift counteraction optimal control (DCOC) problems has been proposed in this paper. By solving a nonlinear programming problem with an exponentially weighted cost function, the approach guarantees to find a control input sequence that maximizes the time-before-exit. Compared to a previous approach based on mixed-integer programming (MIP), the new approach has much lower computational footprint.

The new approach has been illustrated based on numerical examples representing the drift counteraction control for an indoor airship. Simulation results validated the effectiveness of this approach, also demonstrated its improvement in computational efficiency over the previous MIP-based approach.

Integrating the proposed approach in a model predictive control framework to achieve closed-loop drift counteraction control represents a natural extension and will be investigated in detail in our future work.

Acknowledgement

The authors would like to express their gratitude to Professor Moritz Diehl for pointing out the reference [14] and suggesting that a similar approach of weighted stage costs could be developed for treating DCOC problems through NLP.

References

[1] I. Kolmanovsky and R. A. E. Zidek, "Drift counteraction and control of downsized and underactuated systems: What MPC has to offer?" *IFAC-PapersOnLine*, vol. 51, no. 20, pp. 175–190, 2018.

[2] I. V. Kolmanovsky, L. Lezhnev, and T. L. Maizenberg, "Discrete-time drift counteraction stochastic optimal control: Theory and application-motivated examples," *Automatica*, vol. 44, no. 1, pp. 177–184, 2008.

[3] R. A. E. Zidek and I. V. Kolmanovsky, "Deterministic drift counteraction optimal control and its application to satellite life extension," in *Conference on Decision and Control*. IEEE, 2015, pp. 3397–3402.

[4] ——, "Optimal driving policies for autonomous vehicles based on stochastic drift counteraction," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 290–296, 2017.

[5] Z. Li, T. Chu, I. V. Kolmanovsky, and X. Yin, "Training drift counteraction optimal control policies using reinforcement learning: An adaptive cruise control example," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2903–2912, 2017.

[6] G. Barles and B. Perthame, "Exit time problems in optimal control and vanishing viscosity method," *SIAM Journal on Control and Optimization*, vol. 26, no. 5, pp. 1133–1148, 1988.

[7] A.-P. Blanc, "Deterministic exit time control problems with discontinuous exit costs," *SIAM Journal on Control and Optimization*, vol. 35, no. 2, pp. 399–434, 1997.

[8] M. Malisoff, "Viscosity solutions of the Bellman equation for exit time optimal control problems with vanishing Lagrangians," *SIAM Journal on Control and Optimization*, vol. 40, no. 5, pp. 1358–1383, 2002.

[9] R. A. E. Zidek and I. V. Kolmanovsky, "Stochastic drift counteraction optimal control and enhancing convergence of value iteration," in *Conference on Decision and Control*. IEEE, 2016, pp. 1119–1124.

[10] A. A. Menezes, D. D. Shah, and I. V. Kolmanovsky, "An evaluation of stochastic model-dependent and model-independent glider flight management," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1040–1056, 2017.

[11] D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[12] R. A. E. Zidek, A. Bemporad, and I. V. Kolmanovsky, "Optimal and receding horizon drift counteraction control: Linear programming approaches," in *American Control Conference*. IEEE, 2017, pp. 2636–2641.

[13] A. Richards and J. How, "Mixed-integer programming for control," in *American Control Conference*. IEEE, 2005, pp. 2676–2683.

[14] R. Verschueren, H. J. Ferreau, A. Zanarini, M. Mercangöz, and M. Diehl, "A stabilizing nonlinear model predictive control scheme for time-optimal point-to-point motions," in *Conference on Decision and Control*. IEEE, 2017, pp. 2525–2530.

[15] C. Büskens and H. Maurer, "Sensitivity analysis and real-time optimization of parametric nonlinear programming problems," in *Online Optimization of Large Scale Systems*. Springer, 2001, pp. 3–16.

[16] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 2013.

[17] H. Zhang and J. P. Ostrowski, "Visual servoing with dynamics: Control of an unmanned blimp," in *International Conference on Robotics and Automation*, vol. 1. IEEE, 1999, pp. 618–623.

[18] J.-C. Zufferey, A. Guanella, A. Beyeler, and D. Floreano, "Flying over the reality gap: From simulated to real indoor airships," *Autonomous Robots*, vol. 21, no. 3, pp. 243–254, 2006.

[19] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, and C. S. Woodward, "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[20] L. Magnier, R. Zmeureanu, and D. Derome, "Experimental assessment of the velocity and temperature distribution in an indoor displacement ventilation jet," *Building and Environment*, vol. 47, pp. 150–160, 2012.

[21] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.