

Hybrid optimal control with mixed-integer Lagrangian methods

Viktoriya Nikitina* Alberto De Marchi* Matthias Gerdts*

* *University of the Bundeswehr Munich, Department of Aerospace Engineering, Institute of Applied Mathematics and Scientific Computing, 85577 Neubiberg, Germany.*

Abstract: Models involving hybrid systems are versatile in their application but difficult to optimize efficiently due to their combinatorial nature. This work presents a method to cope with hybrid optimal control problems which, in contrast to decomposition techniques, does not require relaxing the integrality constraints. Based on the discretize-then-optimize approach, our scheme addresses mixed-integer nonlinear problems under mild assumptions. The proposed numerical algorithm builds upon the augmented Lagrangian framework, whose subproblems are handled using successive mixed-integer linearizations with trust regions. We validate the performance of the numerical routine with extensive investigations using hybrid optimal control problems from different fields of application. Promising preliminary results are presented for a motion planning task with hysteresis and a Lotka-Volterra fishing problem with total variation.

Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

Keywords: Hybrid dynamics, Optimal control, Mixed-integer programming, Constrained optimization, Augmented Lagrangian.

1. INTRODUCTION

We consider hybrid optimal control problems (OCPs), that is, optimization problems involving possibly nonsmooth dynamics, mixed-integer states or controls, or logical constraints. Hybrid OCPs were investigated, e.g., by Oldenburg and Marquardt (2008); Grossmann and Trespalacios (2013) where hybrid dynamics are modeled with disjunctive programming. An important subclass of hybrid OCPs are mixed-integer OCPs with discrete-valued controls. Numerical methods often use variable time transformations, relaxation, or decomposition techniques; compare Gerdts (2006); Sager et al. (2011); Bürger et al. (2020); Plate et al. (2024) for important developments. Such approaches are often tailored to discrete-valued controls and exploit the time structure.

In this work, we permit more general hybrid OCPs and apply a direct discretization method to transform the dynamic optimization problem into a finite dimensional mixed-integer nonlinear program (MINLP). Our focus is not on the direct discretization method itself, which exists in various versions, compare Gerdts (2023) for an overview, nor on taking advantage of the time structure. We investigate instead the numerical viability of an optimization scheme that handles hybrid features as they stand, namely without relaxing integrality constraints. Moreover, since solving these problems to global optimality is in general computationally intractable, we do not seek global solutions and are content with suboptimal, yet practical

ones, as in Exler and Schittkowski (2007); Quirynen and Di Cairano (2021).

We study a method for the numerical solution of a class of MINLPs of the following type:

minimize $f(x)$ subject to $x \in \mathcal{X}$, $c(x) \in \mathcal{C}$ (P)
with $f: \mathcal{X} \rightarrow \mathbb{R}$ and $c: \mathcal{X} \rightarrow \mathbb{R}^m$ differentiable functions, $\mathcal{C} \subset \mathbb{R}^m$ a nonempty closed *convex* set (projection-friendly in practice), and \mathcal{X} a nonempty closed set with a mixed-integer linear (MIL) structure. In particular, we assume \mathcal{X} to be described by linear inequalities and integrality constraints on some variables, that is,

$$\mathcal{X} := \left\{ x \in \mathbb{R}^n \mid \begin{array}{l} Ax \leq b, \\ x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \subseteq \{1, \dots, n\} \end{array} \right\} \quad (1)$$

for some matrix A , vector b and index set \mathcal{I} . The general template (P) covers hybrid OCPs discretized with arbitrary schemes, and can capture various features such as mixed-integer control inputs, nonsmooth dynamics and state jumps.

In the following, we will partition the decision variables x into a real-valued vector u and an integer-valued vector z , according to their admissible values for \mathcal{X} . Furthermore, we consider the following blanket assumptions.

Assumption 1.1. With regard to (P),

- functions f and c are continuously differentiable with locally Lipschitz derivatives;
- admissible values for the integer-valued decision variables in z lie in a bounded set.

This work proposes a solution technique for (discretized) hybrid OCPs based on a safeguarded augmented Lagrangian (AL) approach, traditionally confined to nonlinear programming. Adopting the AL framework of Birgin and Martínez (2014), the constrained problem (P) is

This research was partially funded by dtec.bw – Digitalization and Technology Research Center of the Bundeswehr [EMERGENCY-VRD, MissionLab, MORE, SeRANIS]. dtec.bw is funded by the European Union – NextGenerationEU. Corresponding author: Alberto De Marchi, ORCID 0000-0002-3545-6898.

solved through a sequence of simpler subproblems. This methodology is motivated by sequential (partially) unconstrained minimization schemes, including penalty and barrier methods; see [Fiacco and McCormick \(1968\)](#). The AL subproblems are tackled by the tailored solver proposed in [De Marchi \(2025\)](#). Such a practical AL algorithm for MINLPs is our main contribution, intended to seek good local, possibly non-global, solutions to (P).

The remainder of this work is structured as follows. A summary of the optimality concept is described in Section 2, while algorithmic details are provided in Section 3. Hybrid OCPs are addressed with our method in Section 4, where numerical results demonstrate its modelling versatility and efficacy.

2. OPTIMALITY CONDITIONS

This section is dedicated to developing suitable solution concepts for (P). First of all, we seek *feasible* points, namely some $\bar{x} \in \mathcal{X}$ such that $c(\bar{x}) \in \mathcal{C}$. Then, a *global* minimizer \bar{x} for (P) can be readily characterized by

$$\bar{x} \in \mathcal{X}, \quad c(\bar{x}) \in \mathcal{C}, \quad \forall x \in \mathcal{X}, c(x) \in \mathcal{C}: f(\bar{x}) \leq f(x).$$

Since we aim at developing an affordable numerical method, possibly at the price of global optimality, we are interested in *local* optimality notions too. However, as these are sensitive to how neighborhoods are defined, local notions can be weak and fragile in the mixed-integer setting considered here. Indeed, considering simple balls, sufficiently small neighborhoods of feasible points may contain only one integer configuration, making it locally optimal, in a weak sense. To avoid this naive declaration of local optimality, we develop some stronger conditions by constructing neighborhoods based on a polyhedral norm applied to the real-valued components only, denoted $\|\cdot\|_{\text{PL}}$ as in ([De Marchi, 2025](#), Sec. 2). For instance, it can take on the forms

$$\begin{aligned} \|x\|_{\text{PL}} &:= \sum \{|x_i| \mid i \notin \mathcal{I}\} && \text{for } \ell_1\text{-type,} \\ \text{or } \|x\|_{\text{PL}} &:= \max \{|x_i| \mid i \notin \mathcal{I}\} && \text{for } \ell_\infty\text{-type.} \end{aligned}$$

Although not a norm, $\|\cdot\|_{\text{PL}}$ induces compact neighborhoods according to

$$\mathbb{B}_{\text{PL}}(\bar{x}, \Delta) := \{x \in \mathbb{R}^n \mid \|x - \bar{x}\|_{\text{PL}} \leq \Delta\},$$

owing to the boundedness of integer-valued variables by Assumption 1.1. Now, analogously to the “unconstrained” case, local minimizers for (P) can be defined as follows.

Definition 2.1. A point $\bar{x} \in \mathbb{R}^n$ is called a **LOCAL MINIMIZER** for (P) if it is feasible and there exists $\Delta > 0$ such that $f(\bar{x}) \leq f(x)$ for all feasible $x \in \mathbb{B}_{\text{PL}}(\bar{x}, \Delta)$. If the latter property additionally holds for all $\Delta > 0$, then \bar{x} is called a **GLOBAL MINIMIZER** for (P).

Notice that non-polyhedral norms for $\|\cdot\|_{\text{PL}}$ are deliberately avoided so that, in practice, local models of (P) take the form of mixed-integer *linear* programs. However, this specification is not necessary to the definition of suitable neighborhoods nor to identify stronger local minimizers.

We now focus on first-order necessary optimality conditions for (P). These should be useful and practical in order to characterize and numerically detect points that are (at least) candidates for global (or local) minimizers ([Birgin and Martínez, 2014](#), Chapter 3).

2.1 Mixed-integer polyhedral constraints

Let us first consider the minimization of some smooth function $\varphi: \mathcal{X} \rightarrow \mathbb{R}$ over \mathcal{X} :

$$\text{minimize } \varphi(x) \quad \text{over } x \in \mathcal{X}. \quad (2)$$

Recalling ([De Marchi, 2025](#), Def. 2.3, Prop. 2.4), a first-order optimality measure associated to (2) is defined for all $\bar{x} \in \mathcal{X}$ and $\Delta \geq 0$ by

$$\Psi_{\varphi, \mathcal{X}}(\bar{x}, \Delta) := \max_{\substack{x \in \mathcal{X} \\ \|x - \bar{x}\|_{\text{PL}} \leq \Delta}} \langle \nabla \varphi(\bar{x}), \bar{x} - x \rangle \geq 0. \quad (3)$$

Then, a first-order optimality concept for (2) is that in the following definition, which also provides an approximate counterpart thereof.

Definition 2.2. Given some $\varepsilon \geq 0$, a point $\bar{x} \in \mathcal{X}$ is said ε -**CRITICAL** for (2) if there exists some $\Delta > 0$ such that $\Psi_{\varphi, \mathcal{X}}(\bar{x}, \Delta) \leq \varepsilon$. A 0-critical point is simply called **CRITICAL**.

As discussed in ([De Marchi, 2025](#), Sec. 2.2), criticality is a necessary optimality condition for (2) and, as such, plays a crucial role when it comes to designing an affordable numerical method for solving (2). The denomination *criticality*, in contrast to *stationarity*, emphasizes that in general the former notion implies (hence, is stronger than) the latter.

2.2 Nonlinear constraints

The “unconstrained” notion of Definition 2.2 is useful to characterize solutions of a problem with simple constraints. But what is a “critical point” for a problem such as (P)? A stationarity characterization that resembles, at least in spirit, the classical Karush-Kuhn-Tucker (KKT) conditions in nonlinear programming has been proposed and analysed by [De Marchi \(2024a\)](#), Sec. 2.2).

Let the Lagrangian function $\mathcal{L}: \mathcal{X} \times \mathbb{R}^m \rightarrow \mathbb{R}$ associated to (P) be defined by

$$\mathcal{L}(x, y) := f(x) + \langle y, c(x) \rangle \quad (4)$$

where y denotes a Lagrange multiplier. From the viewpoint of nonlinear programming, where local minima correspond to saddle points of the Lagrangian functions, we consider the following notion for KKT-like points of (P), building upon Definition 2.2.

Definition 2.3. A point $\bar{x} \in \mathbb{R}^n$ is said **KKT-CRITICAL** for (P) if $\bar{x} \in \mathcal{X}$ and there exists a multiplier $y \in \mathbb{R}^m$ and some $\Delta > 0$ such that

$$\Psi_{\mathcal{L}(\cdot, y), \mathcal{X}}(\bar{x}, \Delta) = 0 \quad \text{and} \quad y \in \mathcal{N}_{\mathcal{C}}(c(\bar{x})).$$

Notice that KKT criticality demands feasibility since the normal cone $\mathcal{N}_{\mathcal{C}}(c(\bar{x}))$ to \mathcal{C} at $c(\bar{x})$ would be empty otherwise. By (3)–(4), the first condition in Definition 2.3 can be rewritten as

$$\min_{\substack{x \in \mathcal{X} \\ \|x - \bar{x}\|_{\text{PL}} \leq \Delta}} \langle \nabla f(\bar{x}) + c'(\bar{x})^\top y, x - \bar{x} \rangle = 0,$$

meaning that the Lagrangian function at \bar{x} cannot be (locally) further minimized with respect to x while maintaining MIL feasibility, in the sense of Definition 2.2. Thanks to the $\|\cdot\|_{\text{PL}}$ -localization, this condition is in general stronger than the inclusion $0 \in \nabla_x \mathcal{L}(\bar{x}, y) + \mathcal{N}_{\mathcal{X}}^{\text{lim}}(\bar{x})$, corresponding to mere stationarity, where $\mathcal{N}_{\mathcal{X}}^{\text{lim}}(\bar{x})$ denotes the limiting

normal cone to \mathcal{X} at \bar{x} . Equivalence holds when the set \mathcal{X} is convex, which is not the case as soon as $\mathcal{I} \neq \emptyset$.

A criterion for numerical termination arises from relaxing criticality and feasibility requirements in Definition 2.3. Analogously to ε -criticality of Definition 2.2, we consider the following concept, akin to (De Marchi, 2024b, Def. 3.2).

Definition 2.4. Given some $\varepsilon \geq 0$, a point $\bar{x} \in \mathbb{R}^n$ is said ε -KKT-CRITICAL for (P) if $\bar{x} \in \mathcal{X}$ and there exists a multiplier $y \in \mathbb{R}^m$, a vector $z \in \mathcal{C}$, and some $\Delta > 0$ such that

$$\Psi_{\mathcal{L}(\cdot, y), \mathcal{X}}(\bar{x}, \Delta) \leq \varepsilon, \quad y \in \mathcal{N}_{\mathcal{C}}(z), \quad \|c(\bar{x}) - z\| \leq \varepsilon.$$

Consistently with Definition 2.3, a 0-KKT-critical point is KKT-critical. Possibly under some constraint qualifications (e.g., LICQ), KKT-criticality provides a necessary optimality condition for (P). However, regardless of additional regularity, each local minimizer of (P) is *asymptotically* KKT-critical, cf. De Marchi (2024a, Thm 2.6).

3. ALGORITHMIC FRAMEWORK

We employ an augmented Lagrangian (AL) scheme to seek a numerical solution for (P) under Assumption 1.1. This design choice is to exploit the fact that the AL framework maintains the original problem's structure and it does not rely on smoothness of the objective nor on convexity of the feasible set (Birgin and Martínez, 2014). Building upon a sequence of shifted-penalty subproblems, AL algorithms can be described with a nested structure. An outer loop generates a sequence of primal-dual estimates, adapts the penalty parameter, and monitors convergence toward KKT-critical points. Each subproblem associated to a set of parameters is addressed by an iterative procedure, effectively an inner loop. Although the AL framework comprises several algorithms and variants, all these share a core element: the successive parametric minimization of an AL function. The safeguarded AL scheme for (P) stated in Algorithm 3.1 patterns that of Birgin and Martínez (2014, Sec. 4.1) and proceeds with a sequence of subproblems of the form

$$\text{minimize } \mathcal{L}_{\mu}(x, \hat{y}) \quad \text{over } x \in \mathcal{X} \quad (5)$$

for some given penalty parameter $\mu > 0$ and multiplier estimate $\hat{y} \in \mathbb{R}^m$. Herein, the MIL constraints $x \in \mathcal{X}$ are kept implicit, and the AL function is defined by

$$\mathcal{L}_{\mu}(x, \hat{y}) := f(x) + \frac{1}{2\mu} \text{dist}_{\mathcal{C}}^2(c(x) + \mu\hat{y}) - \frac{\mu}{2} \|\hat{y}\|^2, \quad (6)$$

cf. De Marchi (2024b, Sec. 4). Feasibility of (5) follows from \mathcal{X} being nonempty whereas well-posedness is due to continuity of $\mathcal{L}_{\mu}(\cdot, \hat{y})$ and can be guaranteed, e.g., by coercivity or (level) boundedness arguments. Notice that \mathcal{L} and \mathcal{L}_{μ} are differentiable with respect to both primal and dual variables, thanks to convexity of \mathcal{C} , with derivatives

$$\nabla_x \mathcal{L}_{\mu}(x, \hat{y}) = \nabla f(x) + c'(x)^{\top} y_{\mu}(x, \hat{y}), \quad (7a)$$

$$\nabla_y \mathcal{L}_{\mu}(x, \hat{y}) = c(x) - s_{\mu}(x, \hat{y}) \quad (7b)$$

where

$$s_{\mu}(x, \hat{y}) := \text{proj}_{\mathcal{C}}(c(x) + \mu\hat{y}), \quad (8a)$$

$$y_{\mu}(x, \hat{y}) := \hat{y} + \frac{c(x) - s_{\mu}(x, \hat{y})}{\mu}. \quad (8b)$$

Furthermore, note that the input requirement in Algorithm 3.1 is not a hard restriction: given some $x^0 \notin \mathcal{X}$,

Algorithm 3.1: Safeguarded augmented Lagrangian routine for (P)

Input: $x^0 \in \mathcal{X}$, $y^0 \in \mathbb{R}^m$, $\epsilon_p, \epsilon_d > 0$, $\mu_1, \varepsilon_1 > 0$,
 $\kappa_{\mu}, \theta_{\mu}, \kappa_{\varepsilon} \in (0, 1)$, $\mathcal{Y} \subset \mathbb{R}^m$ compact

```

1 for  $j = 1, 2, \dots$  do
2    $\hat{y}^j \leftarrow \text{proj}_{\mathcal{Y}}(y^{j-1})$ 
3    $x^j \leftarrow$  compute an  $\varepsilon_j$ -critical point for  $\mathcal{L}_{\mu_j}(\cdot, \hat{y}^j)$ 
      over  $\mathcal{X}$  starting from  $x^{j-1}$ 
4    $s^j \leftarrow \text{proj}_{\mathcal{C}}(c(x^j) + \mu_j \hat{y}^j)$  and  $v^j \leftarrow c(x^j) - s^j$ 
5    $y^j \leftarrow \hat{y}^j + v^j / \mu_j$ 
6   if  $\varepsilon_j \leq \epsilon_d$  and  $\|v^j\| \leq \epsilon_p$  then return  $(x^j, y^j)$ 
7   if  $j = 1$  or  $\|v^j\| \leq \max\{\epsilon_p, \theta_{\mu} \|v^{j-1}\|\}$  then
       $\mu_{j+1} \leftarrow \mu_j$ , else  $\mu_{j+1} \leftarrow \kappa_{\mu} \mu_j$ 
8    $\varepsilon_{j+1} \leftarrow \max\{\epsilon_d, \kappa_{\varepsilon} \varepsilon_j\}$ 

```

it is enough to project it onto \mathcal{X} by solving a MILP, for instance $\min_{x \in \mathcal{X}} \|x - x^0\|_1$.

The major computational toll is taken by Step 3, which aims to minimize $\mathcal{L}_{\mu_j}(\cdot, \hat{y}^j)$ over \mathcal{X} . In fact, only approximate criticality is required, and the sequential mixed-integer linearization scheme of De Marchi (2025, Alg. 3.1) can be readily adopted to compute a suitable ε_j -critical point x^j , in the sense of Definition 2.2. Although the performance of the algorithm might be improved by warm-starting the inner solver and guaranteeing descent behaviour, these properties are not strictly required in the convergence analysis (De Marchi, 2024a). The classical first-order dual estimate update takes place at Steps 4 and 5. Given a (possibly inexact, first-order) solution x to (5), the dual update rule in Step 5 is designed toward the identity

$$\nabla_x \mathcal{L}_{\mu}(x, \hat{y}) = \nabla f(x) + c'(x)^{\top} y = \nabla_x \mathcal{L}(x, y),$$

as usual in AL methods. This relationship allows to monitor the “outer” convergence for (P) with the “inner” subproblem tolerance for (5). Since the inclusion $y \in \mathcal{N}_{\mathcal{C}}(s)$ in Definition 2.4 is always satisfied by construction of s^j and y^j in Steps 4 and 5, approximate KKT criticality can be directly detected in Step 6 returning a pair (x^j, y^j) . The dual safeguard takes place in Step 2, where the compact set \mathcal{Y} can be a generic hyperbox or tailored to the constraint set \mathcal{C} to exploit additional known structures. Finally, penalty parameter and inner tolerance are adapted in Steps 7 and 8 following classical update rules (Birgin and Martínez, 2014, Sec. 12.4). Initial values for steering primal-dual tolerance sequences can be user-specified or adaptively selected based on the initial infeasibility and criticality measure. We shall mention also that, when a feasible initial point is available, reset schemes are applicable and provide asymptotic feasibility guarantees hardly attainable otherwise.

In analogy to magical steps (Birgin and Martínez, 2014, Sec. 8.2), (De Marchi, 2024b, Sec. 4.1) and feasibility pumps (D’Ambrosio et al., 2012), additional steps can be optionally included in Algorithm 3.1 to ease or accelerate the solution process. Such routines could pursue feasibility (switching to the minimization of an infeasibility measure) or refine the incumbent solution by solving a nonlinear program with fixed integer-valued variables (De Marchi, 2025, Rem. 3.1), (Kirches et al., 2022, Sec. 5).

Remark 3.1. A favorable attribute of AL methods is that if the AL subproblems are solved to approximate global optimality, then the iterates accumulate at feasible, globally optimal points (Birgin and Martínez, 2014, Thm 5.2). Thus, whenever additional knowledge or problem structure allow to solve the AL subproblems to approximate global optimality, this general property of AL methods automatically provides global guarantees for (P).

4. NUMERICAL RESULTS

In this section, we present and compare the results achieved by the proposed numerical approach for two hybrid optimal control problems. Before considering the simulations, we highlight some aspects regarding the computational procedure behind Algorithm 3.1 (ALM).

ALM is initialized with zero dual estimate y^0 , penalty parameter $\mu_1 = 10^{-2}$ and inner tolerance $\varepsilon_1 = 10^{-2}$. The choice of x^0 is based on a more sophisticated approach. Admissible points for the set \mathcal{X} are computed as projections onto it, in the ℓ_1 -norm sense. Another routine was implemented to generate reasonable starting points and to refine solutions provided by ALM. While integer variables are relaxed in the former case, they are fixed (to an admissible value) in the latter case so that only real-valued variables are further optimized. In both scenarios, one ends up with a nonlinear problem, which is then tackled by Ipopt (Wächter and Biegler, 2006). We use the numerical routine of (De Marchi, 2025, Sec. 4) for executing Step 3, with Gurobi for solving the MIL subproblems there.

4.1 Car with hysteretic turbo charger and drag

The first problem handled by the proposed method is inspired by one from De Marchi (2025, Sec. 4.1). The aim is to perform point-to-point one-dimensional motion planning of a car with hybrid nonlinear dynamics, extending the previous model to accommodate a drag force term. The underlying double-integrator point-mass model is equipped with a hysteretic turbo accelerator. The car is described by its time-dependent position $x(t)$, velocity $v(t)$ and a turbo state $w(t) \in \{0, 1\}$ for each time point $t \in [0, T]$, $T > 0$. The control variables correspond to the inputs to the acceleration $a(t)$ and brake $b(t)$ pedals. The turbo mode becomes active or inactive once the velocity exceeds $v^+ := 10$ or falls below the threshold $v^- := 5$, respectively. An activated turbo mode entails an effective increase of the nominal thrust by three times but leaves the braking force unaffected. These dynamics read

$$\dot{x}(t) = v(t), \quad \dot{v}(t) = \tau(w(t), a(t)) - b(t) - c_d v(t)^2$$

where $c_d \geq 0$ denotes the drag coefficient, and the thrust τ has two modes of operation depending on the turbo state: $\tau(w, a) = a$ if $w = 0$, and $\tau(w, a) = 3a$ if $w = 1$. The velocity is limited by $|v(t)| \leq v_{\max} := 25$. Boundary conditions, state and control bounds, and the minimum-control objective function are chosen following De Marchi (2025, Sec. 4.1). The same holds for the hysteresis characteristic describing the turbo behaviour, whose logical implications are encoded as MIL constraints.

Simulations Considering the final time $T = 10$, the problem is fully discretized using the explicit Euler scheme

on a uniform time grid with $N = 100$ intervals. The resulting problem in the form (P) has 100 binary-valued optimization variables and 101 nonlinear constraints.

ALM is initialized with an all-zero primal input. The solutions returned for different values of the drag coefficient $c_d \in \{10^{-3}, 10^{-2}\}$ are displayed in Figure 1, together with the solution obtained with dynamic programming (DP). For DP, we used uniform discretization grids with 100 values for each (real-valued) state and control. In all tests, boundary conditions and path constraints are approximately satisfied. As expected, all solutions resemble the one presented in De Marchi (2025) for the linear case with $c_d = 0$, whereas more acceleration and less braking are demanded with stronger drag. The DP solution exhibits some chattering behaviour in the braking phase—this result may be an artifact due to discretization. However, finer discretizations appeared impractical in terms of runtime (on a standard laptop): for each call, DP took minutes to compute, whereas ALM was in the order of seconds.

For $c_d = 10^{-3}$, the turbo car problem was in addition solved by ALM combined with a subsequent fixed-integer NLP refinement to examine the impact of the problem size on the computational effort. As illustrated in Figure 2, the NLP runtime grows faster but is lower than ALM's, which however increases only linearly with N .

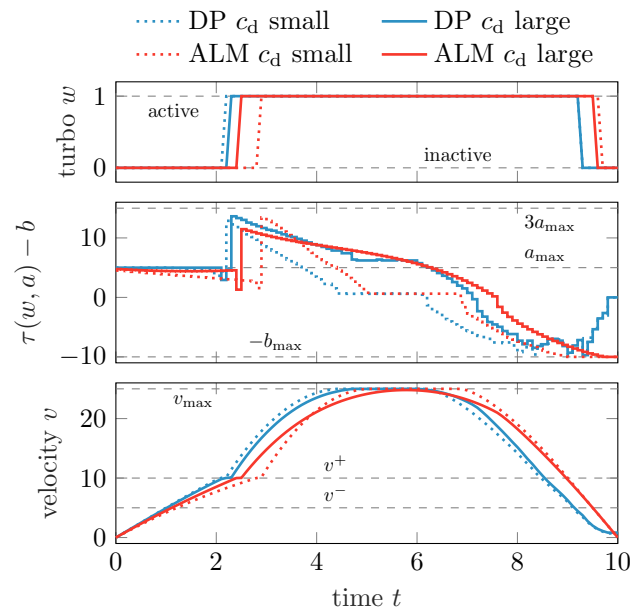


Fig. 1. Solutions for the turbo car problem with different values of drag coefficient $c_d \in \{10^{-3}, 10^{-2}\}$. Comparison of Algorithm 3.1 (ALM), starting from an all-zero initial guess, and dynamic programming (DP).

4.2 Lotka-Volterra fishing with total variation

Let us now consider the classical Lotka-Volterra fishing problem complemented with a total variation (TV) term (Sager, 2012). For two states x_1 and x_2 , which can be interpreted as the biomasses of prey and predator populations, respectively, a binary optimal control is sought. The dynamics are described with the model

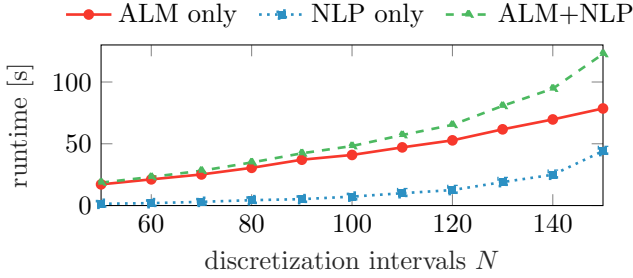


Fig. 2. Runtimes for the turbo car problem with drag $c_d = 10^{-3}$ for different number N of discretization intervals. Solutions obtained with ALM and subsequent fixed-integer NLP refinement.

$$\begin{aligned}\dot{x}_1(t) &= x_1(t) - x_1(t)x_2(t) - c_1^\top w(t), \\ \dot{x}_2(t) &= x_1(t)x_2(t) - x_2(t) - c_2^\top w(t)\end{aligned}$$

for some given parameters $c_1, c_2 \in \mathbb{R}^5$ and control inputs $w(t) \in \{0,1\}^5$. For every time $t \in \mathbb{R}$, the pointwise constraint $\sum_{i=1}^5 w_i(t) = 1$ encodes the choice of one among five control options. The primary objective is formulated as

$$J(x) := \min \int_0^T \|x(t) - x_r\|^2 dt$$

for tracking the reference value $x_r = (1,1)$, starting from the initial state $x_1(0) = 0.5$, $x_2(0) = 0.7$ and with final time $T = 12$ (Sager et al., 2011, Sec. 5).

To limit the well-known chattering behaviour of the optimal control, we monitor the total variation (in fact, its discrete counterpart, after time discretization)

$$\text{TV}(w) := \frac{1}{2} \sum_{i=1}^5 \sum_{k=1}^{N-1} |w_{i,k+1} - w_{i,k}|,$$

possibly imposing an upper bound $\text{TV}(w) \leq U_{\text{TV}}$ or including a penalty term $\alpha_{\text{TV}} \text{TV}(w)$ in the objective. By including additional $(N-1)$ real-valued variables and $4(N-1)$ linear inequality constraints for each binary-valued control, the nonsmooth TV term can be expressed as a linear function of the unknowns.

Simulations The mixed-integer OCP is fully discretized using the explicit Euler scheme over a uniform time grid with $N = 50$ intervals. The resulting discretized problem contains 250 integer-valued optimization variables and 51 nonlinear constraints.

The initial guess values are set to 1 for all variables and the problem (with no TV cost nor limit) is first considered with relaxed integrality and solved by Ipopt. Such solution is then used as input for the combinatorial integral approximation (CIA) (Sager et al., 2011), computed via `pycombina` (Bürger et al., 2020), and for warm-starting ALM. The refined solutions are displayed in Figure 3 and summarized in Table 1, together with the globally optimal solution obtained with DP (on a uniform grid with 100 values for each state). We report the effective control $\Omega := \sum_{i=1}^5 iw_i$, instead of the vector-valued w , to better display the selected fishing option. When warm-started from the NLP relaxation, the infeasible initial guess is first projected onto \mathcal{X} before running ALM; the returned

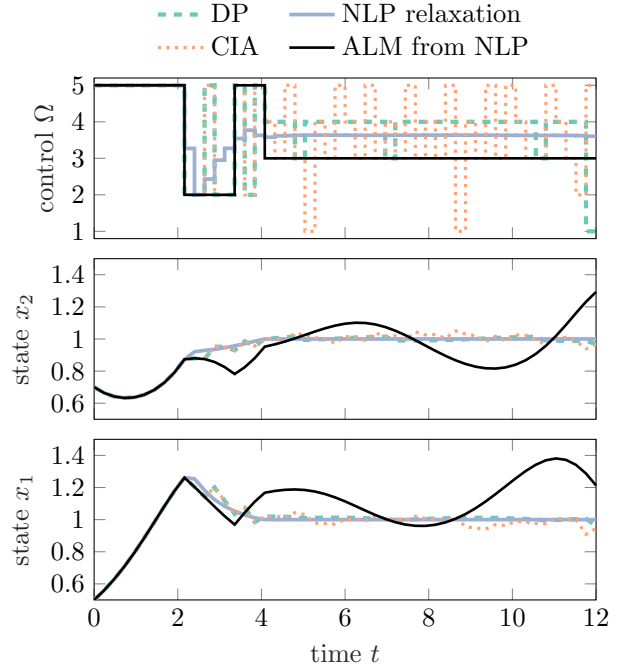


Fig. 3. Solutions for the fishing problem without TV, starting CIA and ALM from the NLP relaxation.

solution corresponds to such projection and does not provide good tracking performance. Similarly, when warm-started from the DP or CIA solution, ALM returns the same control sequence, validating their local optimality.

Table 1. Results for the fishing problem with and without TV: upper bound U_{TV} on the total variation TV, tracking cost J . When not indicated, ALM starts from the CIA solution.

solver	U_{TV}	$\text{TV}(w)$	$J(x)$
DP	∞	14	0.388
ALM from DP	∞	14	0.388
NLP relaxation	∞	2.6	0.381
ALM from NLP	∞	3	0.834
CIA from NLP	∞	38	0.398
ALM	∞	38	0.398
ALM	38	38	0.398
ALM	33	33	0.415
ALM	28	28	0.825
ALM	23	23	0.549
ALM	18	18	1.206
ALM	13	13	0.601

We now consider the performance of ALM when limiting the control TV. Always starting from the CIA solution, ALM is executed with decreasing values U_{TV} of maximum admissible TV. The corresponding results are shown in Figure 4 and collected in Table 1. Note that since the ALM solution coincides with the CIA solution for $U_{\text{TV}} = 38$, this case is omitted in the plots from Figure 4.

Expectedly, limiting the (discretized) TV helps avoiding a chattering behaviour of the optimal solution but at the cost of impairing the tracking performance. In all cases, the returned solution had the maximum TV allowed. Moreover, Table 1 reports that the solution for $U_{\text{TV}} = 23$ (resp. $U_{\text{TV}} = 13$) had a better tracking objective than

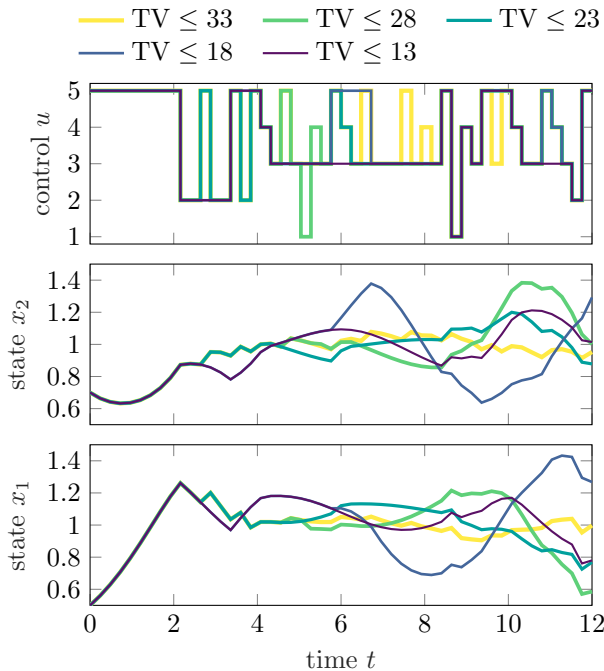


Fig. 4. Solutions for the fishing problem with limited TV, starting ALM from the CIA solution.

the one for $U_{TV} = 28$ ($U_{TV} = 18$), testifying the local nature of ALM when coupled with the local inner solver of De Marchi (2025).

5. CONCLUSIONS

This work presents an efficient method to address hybrid optimal control problems without any integrality relaxations. The proposed approach is based on the augmented Lagrangian framework combined with a routine for nonlinear optimization with mixed-integer linear constraints. The robustness and performance of the algorithm were assessed on two problem examples from different fields of application, validating the proposed strategy. Future research should consider the integration of other techniques for handling nonlinear constraints.

ACKNOWLEDGEMENTS

We thank Rebecca Richter for providing the dynamic programming model and implementation.

REFERENCES

- Birgin, E.G. and Martínez, J.M. (2014). *Practical Augmented Lagrangian Methods for Constrained Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA.
- Bürger, A., Zeile, C., Hahn, M., Altmann-Dieses, A., Sager, S., and Diehl, M. (2020). pycombina: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. *IFAC-PapersOnLine*, 53(2), 6502–6508. doi:10.1016/j.ifacol.2020.12.1799. 21st IFAC World Congress.
- D'Ambrosio, C., Frangioni, A., Liberti, L., and Lodi, A. (2012). A storm of feasibility pumps for nonconvex MINLP. *Mathematical Programming*, 136(2), 375–402. doi:10.1007/s10107-012-0608-x.

- De Marchi, A. (2024a). Affordable mixed-integer Lagrangian methods: optimality conditions and convergence analysis. *arXiv:2406.12436*.
- De Marchi, A. (2024b). Implicit augmented Lagrangian and generalized optimization. *Journal of Applied and Numerical Optimization*, 6(2), 291–320. doi:10.23952/jano.6.2024.2.08.
- De Marchi, A. (2025). Mixed-integer linearity in nonlinear optimization: a trust region approach. *Optimization Letters*. doi:10.1007/s11590-025-02190-9.
- Exler, O. and Schittkowski, K. (2007). A trust region SQP algorithm for mixed-integer nonlinear programming. *Optimization Letters*, 1(3), 269–280. doi:10.1007/s11590-006-0026-1.
- Fiacco, A.V. and McCormick, G.P. (1968). *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*. Wiley, New York.
- Gerdts, M. (2006). A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 27(3), 169–182. doi:10.1002/oca.778.
- Gerdts, M. (2023). *Optimal Control of ODEs and DAEs*. De Gruyter Oldenbourg. doi:10.1515/9783110797893. 2nd Edition.
- Grossmann, I.E. and Trespalacios, F. (2013). Systematic modeling of discrete-continuous optimization models through generalized disjunctive programming. *AIChE Journal*, 59(9), 3276–3295. doi:10.1002/aic.14088.
- Kirches, C., Larson, J., Leyffer, S., and Manns, P. (2022). Sequential linearization method for bound-constrained mathematical programs with complementarity constraints. *SIAM Journal on Optimization*, 32(1), 75–99. doi:10.1137/20M1370501.
- Oldenburg, J. and Marquardt, W. (2008). Disjunctive modeling for optimal control of hybrid systems. *Computers & Chemical Engineering*, 32(10), 2346–2364. doi:10.1016/j.compchemeng.2007.12.002.
- Plate, C., Sager, S., Stoll, M., and Tetschke, M. (2024). Second-order partial outer convexification for switched dynamical systems. *IEEE Transactions on Automatic Control*, 69(7), 4643–4656. doi:10.1109/TAC.2024.3351567.
- Quirynen, R. and Di Cairano, S. (2021). Sequential quadratic programming algorithm for real-time mixed-integer nonlinear MPC. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 993–999. IEEE Press. doi:10.1109/CDC45484.2021.9683714.
- Sager, S. (2012). A benchmark library of mixed-integer optimal control problems. In J. Lee and S. Leyffer (eds.), *Mixed Integer Nonlinear Programming*, 631–670. doi:10.1007/978-1-4614-1927-3_22. https://mintoc.de/index.php/Lotka_Volterra_absolute_fishing_problem, accessed: 2025-04-03.
- Sager, S., Jung, M., and Kirches, C. (2011). Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3), 363–380. doi:10.1007/s00186-011-0355-4.
- Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1), 25–57. doi:10.1007/s10107-004-0559-y.