Time-Varying Optimization for the Control of Differentially Flat Systems With Input Constraints

Adam W. Hall[1] †, John W. Simpson-Porco[2], and Angela P. Schoellig[3] †

***Abstract*—We present a continuous-time framework for constrained control of differentially flat nonlinear systems using cascaded time-varying optimization (TVO). Flatness-based controllers can exploit exact nonlinear dynamics through convex optimization in flat-output coordinates, but enforcing actuator limits remains challenging because the flat-to-physical input map is nonlinear. To address this, we cascade two TVO layers: an output-level TVO that drives constrained tracking in flat-output space, and an input-level TVO that continuously projects the resulting command onto time-varying actuator limits. Both layers are implemented as coupled ordinary differential equations integrated online, avoiding repeated solution of optimization problems. We detail the problem formulation and controller construction, and validate the approach on a wheeled mobile robot tracking task where the method matches the tracking performance of nonlinear model predictive control while strictly respecting constraints, using only an integrator at deployment rather than an online optimization solver.**

***Index Terms*—time-varying optimization, constrained nonlinear control, differential flatness, input constraints, mobile robots**

I. Introduction

Achieving optimal real-time nonlinear robot control on small mobile platforms demands computationally efficient methods, particularly when managing state and input constraints. Nonlinear model predictive control (NMPC) is the most widely used framework for constrained nonlinear control. Modern implementations solve the nonlinear program online to local optimality via sequential quadratic program (SQP) [1]–[3]; the cost is structural, requiring an embedded solver and a discretized model at every timestep.

Differential flatness offers a path to computational efficiency: if a system is differentially flat, its dynamics can be exactly linearized through a nonlinear input transformation, enabling convex optimization in flat-output coordinates [4]. Flat model predictive control (FMPC) exploits this structure for efficient constrained tracking [5]–[7]. However, because the flat-to-physical input map is nonlinear, imposing convex actuator limits in the original input space yields *nonlinear* constraints in flat coordinates, reversing the efficiency gains of flatness. Existing workarounds based on convex approximations or projection stages can be difficult to construct and often lead to conservative performance [8]–[10].

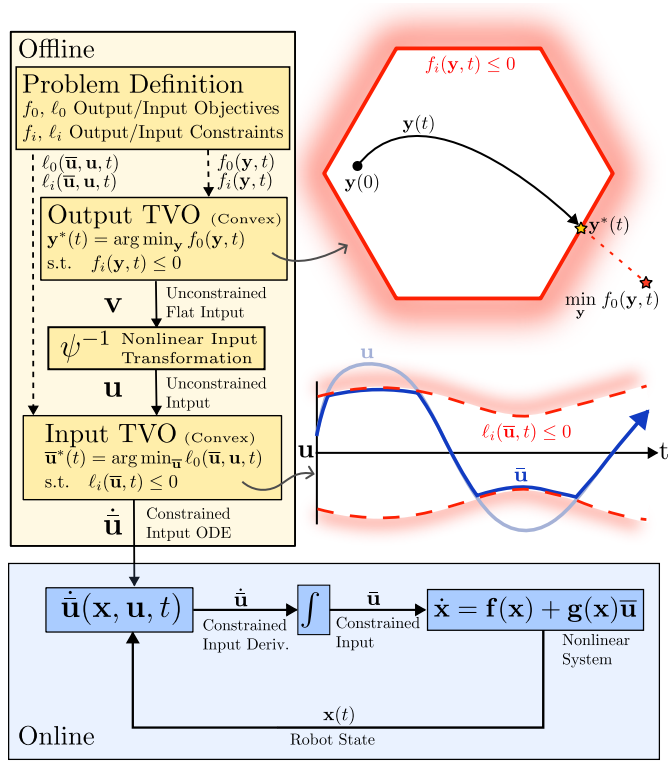

Fig. 1. Overview of the cascaded IC-TVO architecture. The output-level TVO produces a nominal flatness-based input, and the input-level TVO enforces actuator constraints while preserving the tracking objective.

Continuous-time time-varying optimization (TVO) methods offer a structurally different route to the same goal: rather than discretizing time and solving an optimization at each timestep, they track the time-varying optimum directly through a predictor–corrector ordinary differential equation (ODE) [11], [12]. When combined with differential flatness, TVO-based controllers achieve constrained tracking in flat-output space with convergence guarantees [13], [14]. However, these methods do not explicitly enforce actuator bounds.

To address this gap, we propose input-constrained, time-varying optimization (IC-TVO), a cascaded architecture that pairs an output-level TVO for constrained tracking with a second input-level TVO that projects the nominal flatness-based command onto time-varying actuator limits (Figure 1). Both

[1]Adam W. Hall is with the Learning Systems and Robotics Lab (www.learnsyslab.org) at the University of Toronto Institute for Aerospace Studies (UTIAS), Toronto, Canada. Email: adam.hall@robotics.utias.utoronto.ca

[2]J. W. Simpson-Porco is with the Department of Electrical and Computer Engineering, University of Toronto, Canada. Email: jwsimpson@ece.utoronto.ca

[3]Angela P. Schoellig is with the Learning Systems and Robotics Lab (www.learnsyslab.org) at the Technical University of Munich and the University of Toronto, and the Munich Institute for Robotics and Machine Intelligence (MIRMI), Munich, Germany.

†These authors are with the Vector Institute for Artificial Intelligence, Toronto, Canada.

layers are realized as coupled ODEs and integrated online, avoiding the repeated solution of optimization problems. We present the formulation and controller construction, provide high-level stability arguments based on composite Lyapunov analysis, and validate the approach on a wheeled mobile robot (WMR) example where IC-TVO matches a state-of-the-art NMPC implementation in tracking while using only an ODE integrator at deployment. In doing so, we extend the TVO literature by cascading two convex TVO problems through a nonlinear relation.

The manuscript emphasizes formulation and empirical behavior, with concise proof sketches to support the main results. Section II defines the problem, Section III summarizes output-level TVO, Section IV introduces the input-constraining cascade, and Section V presents numerical results.

A. Notation

Scalars are denoted by lowercase characters and vectors/matrices by bold symbols. Time is $t \in \mathbb{R}_{\geq 0}$, Euclidean norm is $\|\cdot\|$, and full time derivatives use superscripts, e.g., $\mathbf{x}^{(k)}$. For smooth $f(\mathbf{x}, t)$, we use $\nabla_{\mathbf{x}} f$, $\nabla_{\mathbf{xx}} f$, and $\nabla_{\mathbf{x}t} f$ for first/second/state-time partial derivatives.

II. Problem Statement

We consider a control-affine nonlinear system

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) + \mathbf{g}(\mathbf{x}(t))\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)), \tag{1}$$

with state $\mathbf{x} \in \mathbb{R}^n$, input $\mathbf{u} \in \mathbb{R}^m$, and output $\mathbf{y} \in \mathbb{R}^m$. We assume (1) is *0-flat* with flat output $\mathbf{y} = \mathbf{h}(\mathbf{x})$ [4], [15], meaning that states and inputs can be recovered from the flat output and its derivatives alone—without requiring knowledge of $\mathbf{u}$ or its derivatives—through smooth maps $\mathbf{x} = \boldsymbol{\phi}^{-1}(\mathbf{y}, \ldots, \mathbf{y}^{(k-1)})$, $\mathbf{u} = \boldsymbol{\psi}^{-1}(\mathbf{y}, \ldots, \mathbf{y}^{(k-1)}, \mathbf{y}^{(k)})$. With the flat state $\mathbf{z} = \operatorname{col}\left(\mathbf{y}, \dot{\mathbf{y}}, \ldots, \mathbf{y}^{(k-1)}\right)$ and flat input $\mathbf{v} := \mathbf{y}^{(k)}$, the dynamics take the linear Brunovský form $\dot{\mathbf{z}} = \mathbf{A}\,\mathbf{z} + \mathbf{B}\,\mathbf{v}$, and the physical input is obtained through the nonlinear map $\mathbf{u} = \boldsymbol{\psi}^{-1}(\mathbf{z}, \mathbf{v})$.

The control objective is to minimize a strongly convex, time-varying cost $f_0(\mathbf{y}, t)$ while respecting time-varying output constraints $\mathbf{y}(t) \in \mathcal{Y}(t)$ and input constraints $\mathbf{u}(t) \in \mathcal{U}(t)$, where both constraint sets are defined by convex inequalities:

$$\mathcal{Y}(t) = \{\mathbf{y} \mid f_i(\mathbf{y}, t) \leq 0,\ i = 1, \ldots, n_f\},$$
$$\mathcal{U}(t) = \{\mathbf{u} \mid \ell_i(\mathbf{u}, t) \leq 0,\ i = 1, \ldots, n_u\}.$$

We assume standard regularity conditions (strong convexity/smoothness of the objective, constraint qualification, and dynamic feasibility) hold throughout; see [13] for precise statements. This formulation subsumes the common case of tracking a reference $\mathbf{y}^{\text{ref}}(t)$ with $f_0(\mathbf{y}, t) = \|\mathbf{y} - \mathbf{y}^{\text{ref}}(t)\|^2$ under time-invariant box input constraints.

III. Time-Varying Optimization-Based Control

This section reviews the TVO-based controller from [13], [14], which our input-constraining extension in Section IV builds upon.

A. Output-Level TVO

The goal is to design a feedback law such that $\mathbf{y}(t)$ tracks the time-varying minimizer of the constrained problem

$$\mathbf{y}^*(t) := \underset{\mathbf{y} \in \mathbb{R}^m}{\arg\min}\, f_0(\mathbf{y}, t) \quad \text{s.t.} \quad f_i(\mathbf{y}, t) \leq 0,\ i \in \{1, \ldots, n_f\}. \tag{2}$$

Rather than solving (2) from scratch at each timestep, TVO methods track $\mathbf{y}^*(t)$ by designing continuous-time dynamics that mimic the behaviour of an optimization algorithm.

a) Barrier approximation: The constraints are incorporated via a log-barrier, yielding the smooth unconstrained surrogate

$$\hat{\mathbf{F}}(\mathbf{y}, t) = f_0(\mathbf{y}, t) - \sum_{i=1}^{n_f} \frac{1}{c(t)} \log(-f_i(\mathbf{y}, t)),$$

with an exponentially growing penalty $c(t) = c_0 e^{\alpha_c t}$ that drives $\hat{\mathbf{y}}^*(t) = \arg\min_{\mathbf{y}} \hat{\mathbf{F}}(\mathbf{y}, t) \to \mathbf{y}^*(t)$ as $t \to \infty$ [13], [16].

b) Target system design: The key idea is to treat the objective gradient $\nabla_{\mathbf{y}} \hat{\mathbf{F}}(\mathbf{y}, t)$ and its first $k-1$ time derivatives as a state vector

$$\boldsymbol{\eta}(\mathbf{z}, t) := \operatorname{col}\left(\nabla_{\mathbf{y}} \hat{\mathbf{F}}(\mathbf{y}, t),\ \nabla_{\mathbf{y}}^{(1)} \hat{\mathbf{F}}(\mathbf{y}, t),\ \ldots,\ \nabla_{\mathbf{y}}^{(k-1)} \hat{\mathbf{F}}(\mathbf{y}, t)\right),$$

and to design the controller so that $\boldsymbol{\eta}$ evolves according to a stable *target system*

$$\dot{\boldsymbol{\eta}} = \mathbf{H}_y\, \boldsymbol{\eta}, \tag{3}$$

where $\mathbf{H}_y = \hat{\mathbf{H}}_y \otimes \mathbf{I}_m$ is a Hurwitz matrix, with $\hat{\mathbf{H}}_y \in \mathbb{R}^{k \times k}$ in canonical controllable form with coefficients $a_0, \ldots, a_{k-1}$. Because $\mathbf{H}_y$ is Hurwitz, $\boldsymbol{\eta} \to 0$ as $t \to \infty$, which in turn implies $\nabla_{\mathbf{y}} \hat{\mathbf{F}}(\mathbf{y}, t) \to 0$ and hence $\mathbf{y}(t) \to \hat{\mathbf{y}}^*(t)$. In other words, we are *embedding* the convergence dynamics of an optimization algorithm directly into the closed-loop system.

c) Deriving the control input: By expanding each total time derivative $\nabla_{\mathbf{y}}^{(i)} \hat{\mathbf{F}}(\mathbf{y}, t)$ via the chain rule, one finds that the highest flat-output derivative $\mathbf{v} = \mathbf{y}^{(k)}$ enters affinely through the Hessian $\boldsymbol{\Gamma}_0 = \nabla_{\mathbf{yy}} \hat{\mathbf{F}}(\mathbf{y}, t)$. Enforcing the last row of (3) and solving for $\mathbf{v}$ gives

$$\mathbf{v}(\mathbf{z}, t) = -\boldsymbol{\Gamma}_0^{-1} \left[\sum_{i=0}^{k-1} a_i\, \boldsymbol{\eta}_i + \frac{\partial \boldsymbol{\eta}_{k-1}}{\partial t} + \sum_{j=1}^{k-1} \binom{k-1}{j} \boldsymbol{\Gamma}_j\, \mathbf{z}_{k-j} \right], \tag{4}$$

where $\boldsymbol{\Gamma}_j = \nabla_{\mathbf{yy}}^{(j)} \hat{\mathbf{F}}(\mathbf{y}, t)$ and a_i are the coefficients of $\mathbf{H}_y$. The first sum in (4) is the *corrector* that drives $\boldsymbol{\eta} \to 0$; the remaining terms form the *predictor* that compensates for the time variation of the cost and constraints. Mapping back through the flatness relation yields the physical input

$$\mathbf{u}(\mathbf{x}, t) = \boldsymbol{\psi}^{-1}(\boldsymbol{\phi}(\mathbf{x}),\ \mathbf{v}(\boldsymbol{\phi}(\mathbf{x}), t)). \tag{5}$$

d) Stability intuition: Under feedback (5), $\boldsymbol{\eta}$ satisfies (3) by construction. One can construct the Lyapunov-like function $V_{\mathbf{x}} = \frac{1}{2} \boldsymbol{\eta}^T \mathbf{P}_y\, \boldsymbol{\eta}$, where $\mathbf{P}_y$ is positive definite and satisfies $\mathbf{P}_y \mathbf{H}_y + \mathbf{H}_y^T \mathbf{P}_y \preceq -\epsilon_y \mathbf{I}$, which gives $\dot{V}_{\mathbf{x}} \leq -\epsilon_y \|\boldsymbol{\eta}\|^2$.

Coercivity of $V_{\mathbf{x}}$ over the barrier-induced domain (the log-barrier terms force $V_{\mathbf{x}} \to \infty$ as $\mathbf{y}$ approaches a constraint boundary) ensures that sub-level sets are compact, solutions remain feasible, and $\mathbf{y}(t) \to \mathbf{y}^*(t)$ asymptotically. The main technical difficulty is that the barrier terms make the feedback non-Lipschitz uniformly in time; this is handled by stitching together solutions over finite intervals, with the full proof deferred to subsequent work.

IV. Input-Constrained TVO

The output-level controller (5) does not respect actuator limits, and can demand arbitrarily large inputs when constraints are active. We now introduce a second TVO layer that projects the nominal command onto the feasible input set, and analyze the resulting interconnected system.

A. Input-Constraining TVO

Given the nominal input $\mathbf{u}_r = \mathbf{u}(\mathbf{x}, t)$ from (5), we seek the bounded input $\overline{\mathbf{u}}(t) \in \mathcal{U}(t)$ that stays as close to $\mathbf{u}_r$ as possible, in the sense of minimizing a strongly convex input cost $\ell_0(\overline{\mathbf{u}}, \mathbf{u}_r, t)$, which we express as

$$\overline{\mathbf{u}}^*(t) = \underset{\overline{\mathbf{u}} \in \mathbb{R}^m}{\arg\min}\ \ell_0(\overline{\mathbf{u}}, \mathbf{u}_r, t) \text{ s.t. } \ell_i(\overline{\mathbf{u}}, t) \le 0,\ i = 1, \dots, n_u.$$

The cost ℓ_0 is designed such that when no input constraint is active, $\overline{\mathbf{u}}^* = \mathbf{u}_r$ and the nominal controller is unmodified; otherwise, $\overline{\mathbf{u}}^*$ is the ℓ_0-optimal projection of $\mathbf{u}_r$ onto $\mathcal{U}(t)$ (e.g., $\ell_0(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \|\overline{\mathbf{u}} - \mathbf{u}_r\|^2$).

Following the same barrier strategy as in Section III, the constraints are absorbed into a smooth surrogate

$$\hat{L}(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \ell_0(\overline{\mathbf{u}}, \mathbf{u}_r, t) - \sum_{i=1}^{n_u} \frac{1}{c_u(t)} \log(-\ell_i(\overline{\mathbf{u}}, t)),$$

with $c_u(t) = c_0 e^{\alpha_c t}$ as before. Defining the gradient map $\boldsymbol{\sigma}(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \nabla_{\overline{\mathbf{u}}} \hat{L}(\overline{\mathbf{u}}, \mathbf{u}_r, t)$, we design a Hurwitz target system $\dot{\xi} = \mathbf{H}_u \xi$ with $\mathbf{H}_u = \mathrm{diag}(-b_0, \dots, -b_{m-1})$, and require $\boldsymbol{\sigma}$ to be a trajectory of this target system.

A critical modelling point is that $\mathbf{u}_r = \mathbf{u}(\mathbf{x}, t)$, where $\mathbf{x}$ itself evolves under the constrained input $\overline{\mathbf{u}}$ via $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\overline{\mathbf{u}}$. Thus, expanding $\dot{\boldsymbol{\sigma}}$ via the chain rule gives

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{\Lambda}_{\overline{\mathbf{u}}} \dot{\overline{\mathbf{u}}} + \boldsymbol{\Lambda}_{\mathbf{u}_r} \left(\frac{\partial \mathbf{u}_r}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \mathbf{u}_r}{\partial t} \right) + \frac{\partial \boldsymbol{\sigma}}{\partial t}, \tag{6}$$

where $\boldsymbol{\Lambda}_{\overline{\mathbf{u}},}(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \nabla_{\overline{\mathbf{u}}\overline{\mathbf{u}}} \hat{L}(\overline{\mathbf{u}}, \mathbf{u}_r, t)$ and $\boldsymbol{\Lambda}_{\mathbf{u}_r}(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \nabla_{\overline{\mathbf{u}}\mathbf{u}_r} \hat{L}(\overline{\mathbf{u}}, \mathbf{u}_r, t)$. Equating (6) with $\mathbf{H}_u \boldsymbol{\sigma}$ and solving for $\dot{\overline{\mathbf{u}}}$ yields the input-layer ODE

$$\dot{\overline{\mathbf{u}}} = -\boldsymbol{\Lambda}_{\overline{\mathbf{u}}}^{-1} \left(\frac{\partial \boldsymbol{\sigma}}{\partial t} + \boldsymbol{\Lambda}_{\mathbf{u}_r} \left(\frac{\partial \mathbf{u}_r}{\partial \mathbf{x}} \dot{\mathbf{x}} + \frac{\partial \mathbf{u}_r}{\partial t} \right) - \mathbf{H}_u \boldsymbol{\sigma} \right). \tag{7}$$

We write this compactly as $\dot{\overline{\mathbf{u}}} = \mathbf{f}_{\overline{\mathbf{u}}}(\overline{\mathbf{u}}, \mathbf{x}, t)$, making the dependence on the plant state explicit. As in the output TVO, the $\mathbf{H}_u \boldsymbol{\sigma}$ term is the corrector that drives $\boldsymbol{\sigma} \to 0$ (and hence $\overline{\mathbf{u}} \to \overline{\mathbf{u}}^*$), while the remaining terms predict the time variation of the reference input, constraints, and—crucially—the plant state. This can be shown through a similar stability proof using a Lyapunov-like function $V_{\mathbf{u}}(\overline{\mathbf{u}}, \mathbf{u}_r, t) = \frac{1}{2}\boldsymbol{\sigma}(\overline{\mathbf{u}}, \mathbf{u}_r, t)^T \mathbf{P}_u \boldsymbol{\sigma}(\overline{\mathbf{u}}, \mathbf{u}_r, t)$, with $\mathbf{P}_u$ positive definite and $\mathbf{P}_u \mathbf{H}_u + \mathbf{H}_u^T \mathbf{P}_u \preceq -\epsilon_u \mathbf{I}$.

B. Interconnected System

Applying $\overline{\mathbf{u}}$ to the plant and coupling with (7) gives the interconnected closed-loop system. Defining error coordinates $\mathbf{x}_e = \mathbf{x} - \mathbf{x}^*$ and $\overline{\mathbf{u}}_e = \overline{\mathbf{u}} - \overline{\mathbf{u}}^*$, this takes the form

$$\begin{aligned} \dot{\mathbf{x}}_e &= \mathbf{f}(\mathbf{x}_e + \mathbf{x}^*) + \mathbf{g}(\mathbf{x}_e + \mathbf{x}^*)(\overline{\mathbf{u}}_e + \overline{\mathbf{u}}^*) - \dot{\mathbf{x}}^*, \\ \dot{\overline{\mathbf{u}}}_e &= \mathbf{f}_{\overline{\mathbf{u}}}(\mathbf{x}_e,\ \overline{\mathbf{u}}_e,\ t). \end{aligned}$$

The two subsystems are coupled in both directions: the plant state affects the nominal reference $\mathbf{u}_r$, and the constrained input $\overline{\mathbf{u}}$ drives the plant. We seek $(\mathbf{x}_e, \overline{\mathbf{u}}_e) \to (0, 0)$ as $t \to \infty$. Although each TVO layer is individually asymptotically stable (Sections III and IV-A), stability of the interconnection is not automatic [17] and requires a joint analysis.

C. Stability of the Closed-Loop System

Here, we sketch the key ideas of the stability proof. The analysis uses a composite Lyapunov function $V = V_{\mathbf{x}} + V_{\mathbf{u}}$, combining the output-layer and input-layer Lyapunov functions from Sections III and IV-A. Both terms are positive definite and coercive over their barrier-induced domains, so V has compact sub-level sets within the feasible domain.

The main challenge is that applying the constrained input $\overline{\mathbf{u}}$ rather than the nominal $\mathbf{u}$ introduces a coupling term in $\dot{V}_{\mathbf{x}}$ proportional to the input mismatch $\Delta\mathbf{u} = \overline{\mathbf{u}}^* - \mathbf{u}(\mathbf{x}, t)$. Combining the bounds on $\dot{V}_{\mathbf{x}}$ and $\dot{V}_{\mathbf{u}}$ via Young's inequality yields a scalar differential inequality of the form $\dot{V} \le -k_v V + \alpha_r \|\Delta\mathbf{u}\|^2$, provided the input-layer gain $\mathbf{H}_u$ is chosen sufficiently fast relative to the output-layer gain $\mathbf{H}_y$ (a small-gain condition). A Grönwall argument then establishes practical asymptotic stability: the combined error converges to a neighborhood whose size is governed by $\Delta\mathbf{u}$. When the optimal trajectory is dynamically feasible under the input constraints ($\Delta\mathbf{u} \to 0$), the cascade is asymptotically stable. Finally, the coercivity of the barrier terms in V prevents trajectories from reaching constraint boundaries, ensuring $\mathbf{y}(t) \in \mathcal{Y}(t)$ and $\overline{\mathbf{u}}(t) \in \mathcal{U}(t)$ for all time. The exact details of this proof are still a work-in-progress and will be detailed in subsequent work.

V. Numerical Examples

We validate IC-TVO on a WMR and compare with NMPC. The IC-TVO ODEs use `CasADi` [18] with the adaptive-step `CVODES` [19] solver, and the NMPC baseline uses `acados` [2].

a) WMR dynamics: Consider the WMR $\dot{\mathbf{x}} = [u_1 \cos\theta,\ u_1 \sin\theta,\ u_2]^T$ with state $\mathbf{x} = [x, y, \theta]^T$ and input $\mathbf{u} = [u_1, u_2]^T$. This system is 0-flat in $\mathbf{y} = [x, y]^T$ [13], with $u_1 = \sqrt{\dot{x}^2 + \dot{y}^2}$ and $u_2 = (\dot{x}\ddot{y} - \dot{y}\ddot{x})/(\dot{x}^2 + \dot{y}^2)$. The output TVO (4) determines $\ddot{\mathbf{y}}$, from which $\mathbf{u}$ is computed and passed to the input TVO (7), yielding $\overline{\mathbf{u}}$.

b) Constrained tracking: We task the WMR with tracking a reference $\mathbf{y}^{\mathrm{ref}}(t)$ (a cubic spline over $T = 5\,\mathrm{s}$) subject to a time-varying affine output constraint $\mathbf{a}^T\mathbf{y} + b(t) - r_c \le 0$ with $\mathbf{a} = [1, 2]^T$, $b(t) = 0.2t$, and robot radius $r_c = 0.1$. The input TVO enforces time-varying box constraints $\mathbf{u}^{\min}(t) \le \overline{\mathbf{u}} \le \mathbf{u}^{\max}(t)$ by solving (7) with cost $\ell_0 = \|\overline{\mathbf{u}} - \mathbf{u}\|^2$. For

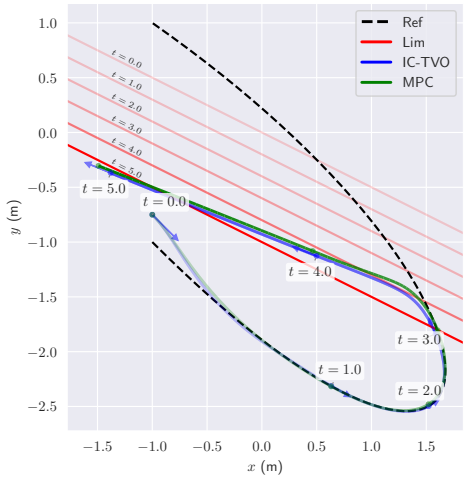

Fig. 2. Trajectories of IC-TVO (blue) and NMPC (green) tracking the reference (black dashed) near the time-varying output constraint (red). Higher transparency indicates earlier time.

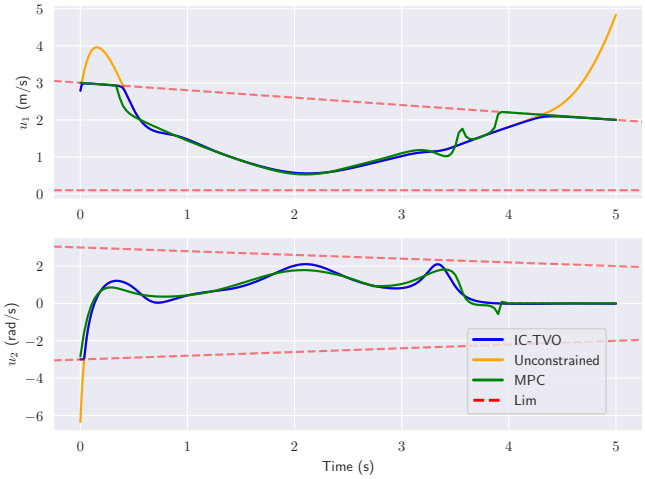

Fig. 3. Unconstrained input $\mathbf{u}$ (orange), constrained input $\overline{\mathbf{u}}$ (blue), and NMPC input (green). The time-varying input bounds (red dashed) are respected by both IC-TVO and NMPC.

comparison, we implement NMPC with `acados` [2], with constraints matched to the TVO problem and cost tuned for best performance with a horizon $N = 30$ at 30 Hz.

c) Results: Figure 2 shows that both IC-TVO and NMPC track the reference while respecting the constraint. The trajectories are very similar; IC-TVO achieves a root-mean-squared error (RMSE) of 0.503 m compared to 0.497 m for the NMPC, the small gap arising from the barrier keeping IC-TVO slightly inside the constraint. Computation is also comparable, at 0.27 s versus 0.43 s, for IC-TVO and NMPC respectively. Of course, this is a simple qualitative comparison dependent on many parameters, but it highlights that even an off the shelf integrator is able to achieve comparable performance to a highly optimized NMPC solver. Future work will explore how to optimize integration methods to achieve even better performance.

Figure 3 compares the inputs: the unconstrained TVO input $\mathbf{u}$ from (5) violates the bounds, while the constrained input $\overline{\mathbf{u}}$ from integrating (7) respects all time-varying input limits. The constrained inputs are comparable to—and at times smoother than—those produced by NMPC.

VI. Discussion and Conclusion

We have presented IC-TVO, a cascaded time-varying optimization framework for constrained control of differentially flat systems, producing constrained inputs through ODE integration rather than repeated online optimization. The numerical results demonstrate tracking comparable to a state-of-the-art NMPC implementation at comparable computational cost, using only an ODE integrator and no online solver at deployment.

A key practical advantage is that the adaptive-step integrator yields a continuous-time input signal by construction: dense-output interpolation supplies high-frequency (>1 kHz) actuation between state updates at negligible cost. Sampled-data NMPC can approximate this by holding its open-loop plan, but IC-TVO provides the inter-sample signal natively. This is attractive for embedded platforms where estimation rates are slower than desired actuation rates. In the extended version we further apply IC-TVO to multi-agent coordination and moving-obstacle avoidance with input bounds, where it performs well even when standing assumptions (e.g., dynamic feasibility under input constraints) are not strictly met.

Several limitations warrant discussion. First, our formulation assumes the system is 0-flat [15] (i.e., the flat output depends only on the state); extending IC-TVO to general flat systems where the flat output depends on inputs and their derivatives remains open. Second, the log-barrier is in effect an interior-point relaxation, so the resulting ODE stiffens near active constraints as the barrier sharpens; integrating it is not inherently cheaper than a small convex projection. The advantage of IC-TVO is thus architectural—no embedded solver, a continuous-time signal—rather than a guaranteed speedup. Adaptive-step solvers handle this in simulation, but the fixed-step integrators favored in robotics are less suited to it, and dedicated solvers or barrier relaxation [20], [21] may be needed for embedded deployment. Third, as discussed in Section IV-C, the cascade analysis relies on a small-gain condition and bounded input mismatch; a complete proof with precise convergence rates is ongoing.

Future work will explore relaxed barrier approximations for better conditioning, Lyapunov redesign (analogous to anti-windup) to tighten the cascade bounds, and experimental validation. A full treatment of the stability theory and additional examples will appear in the extended version.

References

[1] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge: Cambridge University Press, 2017.

[2] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. v. Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, vol. 14, no. 1, pp. 147–183, 2022.

[3] K. Nguyen, S. Schoedel, A. Alavilli, B. Plancher, and Z. Manchester, "Tinympc: Model-predictive control on resource-constrained microcontrollers," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 1–7.

[4] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: Introductory theory and examples," *Int. J. Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[5] M. Greeff and A. P. Schoellig, "Flatness-based model predictive control for quadrotor trajectory tracking," in *2018 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, Madrid, Spain, 2018.

[6] ——, "Exploiting differential flatness for robust learning-based tracking control using gaussian processes," *IEEE Control Syst. Lett.*, vol. 5, no. 4, pp. 1121–1126, 2021.

[7] A. W. Hall, M. Greeff, and A. P. Schoellig, "Differentially flat learning-based model predictive control using a stability, state, and input constraining safety filter," *IEEE Control Syst. Lett.*, vol. 7, pp. 2191–2196, 2023.

[8] D. Simon, J. Lofberg, and T. Glad, "Nonlinear model predictive control using Feedback Linearization and local inner convex constraint approximations," *2013 European Control Conference, ECC 2013*, no. 3, pp. 2056–2061, 2013, iSBN: 9783033039629.

[9] J. Deng, V. Becerra, and R. Stobart, "Input constraints handling in an MPC/feedback linearization scheme," *International Journal of Applied Mathematics and Computer Science*, vol. 19, no. 2, pp. 219–232, 2009.

[10] M. Greeff, A. W. Hall, and A. P. Schoellig, "Learning a stability filter for uncertain differentially flat systems using Gaussian processes," in *2021 60th IEEE Conf. Decision and Control (CDC)*. IEEE, 2021, pp. 789–794.

[11] A. Simonetto, E. Dall'Anese, S. Paternain, G. Leus, and G. B. Giannakis, "Time-varying convex optimization: Time-structured algorithms and applications," *Proceedings of the IEEE*, vol. 108, no. 11, pp. 2032–2048, 2020.

[12] M. Fazlyab, S. Paternain, V. M. Preciado, and A. Ribeiro, "Prediction-Correction Interior-Point Method for Time-Varying Convex Optimization," *IEEE Transactions on Automatic Control*, vol. 63, no. 7, pp. 1973–1986, July 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8062794/

[13] T. Zheng, J. W. Simpson-Porco, and E. Mallada, "Closed-loop motion planning for differentially flat systems: A time-varying optimization framework," *arXiv preprint arXiv:2310.13090*, 2023.

[14] T. Zheng, J. Simpson-Porco, and E. Mallada, "Implicit Trajectory Planning for Feedback Linearizable Systems: A Time-varying Optimization Approach," in *2020 American Control Conference (ACC)*. Denver, CO, USA: IEEE, July 2020, pp. 4677–4682.

[15] P. Martin, R. M. Murray, and P. Rouchon, "Flat systems, equivalence and trajectory generation," California Institute of Technology, Pasadena, CA, USA, Tech. Rep. mmr03-cds, 2003. [Online]. Available: http://www.cds.caltech.edu/~murray/preprints/mmr03-cds.pdf

[16] S. P. Boyd and Lieven. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[17] M. Krstić, I. Kanellakopoulos, and P. V. Kokotović, *Nonlinear and Adaptive Control Design*, ser. Adaptive and Learning Systems for Signal Processing, Communications, and Control. New York: Wiley, 1995.

[18] 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.

[19] A. C. Hindmarsh, R. Serban, C. J. Balos, D. J. Gardner, D. R. Reynolds, and C. S. Woodward, "User documentation for cvodes," urlhttps://sundials.readthedocs.io/en/latest/cvodes, 2024, v7.0.0. [Online]. Available: https://sundials.readthedocs.io/en/latest/cvodes

[20] J. Hauser and A. Saccon, "A Barrier Function Method for the Optimization of Trajectory Functionals with Constraints," in *Proceedings of the 45th IEEE Conference on Decision and Control*. San Diego, CA, USA: IEEE, 2006, pp. 864–869.

[21] C. Feller and C. Ebenbauer, "Relaxed logarithmic barrier function based model predictive control of linear systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1223–1238, 2017.