A New Semidefinite Relaxation for Linear and Piecewise-Affine Optimal Control with Time Scaling

Lujie Yang, Tobia Marcucci, Pablo A. Parrilo, and Russ Tedrake
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology

Abstract— We introduce a semidefinite relaxation for optimal control of linear systems with time scaling. These problems are inherently nonconvex, since the system dynamics involves bilinear products between the discretization time step and the system state and controls. The proposed relaxation is closely related to the standard second-order semidefinite relaxation for quadratic constraints, but we carefully select a subset of the possible bilinear terms and apply a change of variables to achieve empirically tight relaxations while keeping the computational load relatively light. We further extend our method to handle piecewise-affine (PWA) systems by formulating the PWA optimal-control problem (OCP) as a shortest-path problem in a graph of convex sets (GCS). For instance, when a PWA system is used to model a hybrid dynamical system, different paths in this graph represent different mode sequences, and the convex sets model the relaxed dynamics within each mode. By combining a tight convex relaxation of the GCS problem with our time-flexible relaxation, we can solve PWA OCPs through a single semidefinite program.
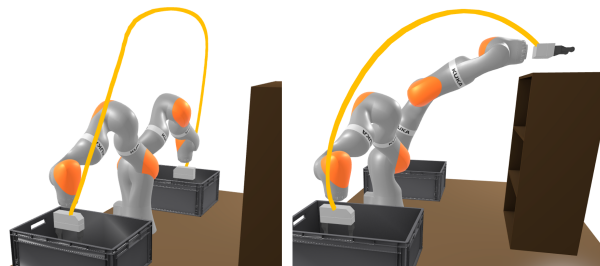
Fig. 1: Minimum-time trajectories for a 7D double integrator moving among obstacles with robot arm geometry. The collision-free configuration space is approximately decomposed into convex polytopes. Our algorithm efficiently navigates these polytopes while respecting input limits. Videos of the trajectories can be found at (left) and (right).

I. INTRODUCTION

We consider the optimal-control problem (OCP) for discrete-time dynamical systems obtained by discretizing a continuous-time system with time step $h > 0$. Consider, for example, a system with continuous-time linear dynamics approximated with the Euler method:

$$x_{k+1} = x_k + h(Ax_k + Bu_k). \tag{1}$$

where x_k and u_k are the state and control at discrete time k, and A and B are the matrices of the continuous-time system. When the time scaling h is fixed, optimizing the trajectory can be efficiently approached through convex optimization. Similarly, if the system trajectory is predetermined, the time-scaling problem can also be addressed as a convex program. However, if the trajectory and time scaling need to be optimized jointly, then the problem becomes nonconvex and considerably more complex to solve. In fact, most common approaches rely on nonconvex trajectory optimization and are prone to converging to local minima.

Many existing works on trajectory optimization [1], [2] generally require a predefined final time [3], [4], thereby limiting their ability to simultaneously optimize both the time scaling and trajectories. This restriction often overlooks low-cost trajectories that could emerge from varying terminal times, leading to less efficient or unsuitable solutions for time-critical applications [5], [6]. Additionally, the fixed-duration requirement is impractical for systems that can only reach the final state within a specific time frame. To remedy this inflexibility, some methods adopt a bilevel optimization approach [7]–[10], alternating between optimizing the time scaling on a fixed path and refining the path for a given time scaling. However, these algorithms are sensitive to the initial guess and easily get stuck in local optima. Direct-collocation techniques [2], [11] offer an alternative by parameterizing trajectories with splines and support incorporating time as a decision variable, but they rely on nonlinear programming (NLP) solvers that may also struggle with local optima or fail to identify feasible trajectories.

Mixed-integer formulations for minimum-time control have been studied over the past decades [12]–[15]. By overcoming the localness of nonlinear programming, mixed-integer convex programming (MICP) has applied to motion planning for autonomous vehicles [16]–[18], locomotion [19]–[21], and manipulation [22], [23]. MICP approaches often handle the originally nonconvex constraints with piecewise convex approximations, and can be solved to global optimum given enough time budget. Additionally, MICP methods are naturally capable of generating optimal trajectories or feedback controllers for piecewise-affine (PWA) dynamical systems with contacts [24]–[26]. These strategies typically adopt integer variables to assign system modes, enabling the simultaneous planning of discrete mode sequences and continuous motions. Nevertheless, these methods typically require integer variables to decide the system mode at every time step for a predefined horizon, therefore imposing considerable computational challenges.

Recently, the work [27] has proposed an efficient MICP formulation of hybrid OCPs as finding the shortest paths

in a Graph of Convex Sets (GCS) (see also [28, Ch. 10]). The GCS approach enjoys a tight convex relaxation of the nonconvex control problem. By addressing the planning problem through a single convex program, GCS empowers a range of effective planners [29]–[31] capable of handling high-dimensional manipulation and contact problems.

In this paper, we leverage semidefinite relaxation (SDR) [32]–[35] to convexify the nonconvex dynamical constraints and incorporate them in the GCS framework. We additionally allow the time step as decision variables to jointly optimize trajectories and time scaling for PWA systems. Recent advances in SDRs have enabled applications to collision-free motion planning [36]–[38] and optimal control [39]–[42]. However, these methods have mostly been limited to path planning in low-dimensional spaces or smooth dynamics without mode switches. Moreover, previous studies [37], [42] found that a computationally expensive complete higher-order SDR is necessary to tighten the relaxation with dynamical constraints. We show that by leveraging the specific structure of the system dynamics, our new formulation achieves empirical tightness with a smaller, more efficient relaxation, removing redundant terms and significantly improving computational efficiency.

SDRs also play an important role in other robotics problems, including inverse kinematics [43], [44] and certifiable perception algorithms [45], [46]. Similar to our work, [31] uses SDR with GCS, but to relax the nonconvex quasi-static dynamics and solves a planar pushing task for fixed final times. In our work, we exploit the structure of the OCP with time scaling, applying and adapting the SDR technique to generate globally optimal trajectories for linear systems. We then further explore such relaxation's potential in trajectory optimization for PWA dynamical systems using GCS.

Our key contributions are summarized as follows:

1) We propose an SDR for nonconvex OCPs for linear systems with time scaling. This SDR exploits the problem structure to be both empirically tight and lightweight.
2) We extend the SDR to handle PWA systems by transcribing the mode optimization problem as a shortest-path problem in GCS. Our method simultaneously optimizes the mode sequence and system trajectory, within a unified convex-optimization framework.

We validate our approach on an inverted pendulum with contact and a 7D double integrator approximation of the dynamics of a robot arm (see Fig. 1). By decomposing the nonconvex OCP into convex components, we are able to jointly optimize trajectories and time scaling for PWA systems that are otherwise challenging for traditional NLP or MICP solvers.

II. PRELIMINARIES ON SEMIDEFINITE RELAXATION

In this section, we provide the necessary background on quadratically constrained quadratic programs (QCQP) and their SDR. The interested readers can refer to [35], as well as [32]–[34], for more details.

Given the variable $z \in \mathbb{R}^n$, we consider the following QCQP (in homogeneous form in terms of $x = (1, z)$):

$$\text{minimize} \quad x^\top A_0 x \tag{2a}$$
$$\text{subject to} \quad x^\top A_i x \geq 0, \quad \forall i = 1, \ldots, l \tag{2b}$$
$$Bx \geq 0 \tag{2c}$$
$$x_1 = 1, \tag{2d}$$

where $A_0, \ldots, A_l \in \mathbb{R}^{(n+1)\times(n+1)}$ are symmetric matrices, $B \in \mathbb{R}^{m\times(n+1)}$, and x_1 denotes the first entry of x. To handle the nonconvex QCQP (2), the standard SDR introduces a positive semidefinite (PSD) matrix variable X, which serves as a relaxation of $xx^\top$. By dropping the nonconvex rank-1 constraint $X = xx^\top$, we arrive to the semidefinite program

$$\text{minimize} \quad \text{tr}(A_0 X) \tag{3a}$$
$$\text{subject to} \quad \text{tr}(A_i X) \geq 0, \quad \forall i = 1, \ldots, l \tag{3b}$$
$$BXe_1 \geq 0 \tag{3c}$$
$$BXB^\top \geq 0 \tag{3d}$$
$$e_1^\top Xe_1 = 1 \tag{3e}$$
$$X \succeq 0, \tag{3f}$$

where $e_1 = (1, 0, \ldots, 0) \in \mathbb{R}^{n+1}$. The optimal value of the SDR (3) provides a lower bound for the original QCQP (2). If the relaxation were exact (i.e., if X is rank-1), the optimal solution of (2) could be directly obtained by factorizing X as $X = xx^\top$. However, in general, the SDR may yield a higher-rank solution, and additional steps may be needed to recover a feasible solution for the original nonconvex QCQP.

III. PROBLEM STATEMENT

We consider a discrete-time dynamical system obtained by discretizing a continuous-time system with time step $h > 0$. The system state $x_k \in \mathbb{R}^{n_x}$ and control input $u_k \in \mathbb{R}^{n_u}$ at each discrete time k are constrained to lie in the polytopes $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{U} \subseteq \mathbb{R}^{n_u}$. We aim to design optimal state trajectories $\{x_k\}_{k=0}^K$ and control inputs $\{u_k\}_{k=0}^{K-1}$, alongside the time step h. Note that the time step h directly parameterizes the time scaling of the entire trajectory. The objective function balances the total time Kh and system energy with the weight $\eta \geq 0$. The initial state is given by x_{start}, and the final state is required to be x_{goal}. Overall, the OCP is

$$\text{minimize} \quad \eta Kh + h \sum_{k=0}^{K-1} (x_k^\top Q x_k + u_k^\top R u_k) \tag{4a}$$
$$\text{subject to} \quad x_{k+1} = f(x_k, u_k, h), \quad \forall k = 0, \ldots, K-1 \tag{4b}$$
$$x_k \in \mathcal{X}, \ u_k \in \mathcal{U}, \quad \forall k = 0, \ldots, K-1 \tag{4c}$$
$$x_0 = x_{\text{start}}, \ x_K = x_{\text{goal}}, \tag{4d}$$

where the matrices Q, R are PSD, and the function f represents the discrete-time system dynamics.

IV. CONVEX RELAXATION WITH TIME SCALING

Even for LTI systems, the OCP (4) is nonconvex due to the biconvex interaction between trajectory parameters $\{x_k\}_{k=0}^K$, $\{u_k\}_{k=0}^{K-1}$, and the time step h (see the objective (4a) and the dynamics (1)). To tackle this challenge, we propose a

convex relaxation of the nonconvex program that leverages the problem structure. We group the bilinear terms to construct an expressive SDR that aligns with the OCP structure. Our approach is first demonstrated for LTI systems, which represent the simplest framework for understanding the logic behind our method. Then, in Section V, we expand this strategy to handle PWA systems with a fixed mode schedule. Finally, in Section VI, we discuss how to also optimize the mode sequence for PWA systems.

A. Time-Flexible Relaxation

The ordinary SDR for nonconvex QCQPs [35] is inadequate for addressing (4) since the stage cost (4a) is not quadratic but rather cubic in the decision variables. To handle the polynomial terms $hx_k^\top x_k, hu_k^\top u_k, hx_k, hu_k$ and obtain a tight convex relaxation of the nonconvex program, we define $r = (x_0, \ldots, x_K, u_0, \ldots, u_{K-1})$. We then introduce the following vector y and its corresponding PSD matrix Y:

$$y = \begin{bmatrix} 1 \\ h \\ r \\ hr \end{bmatrix}, \quad Y = \frac{1}{h}yy^\top = \begin{bmatrix} \frac{1}{h} & 1 & \frac{r^\top}{h} & r^\top \\ 1 & h & r^\top & hr^\top \\ \frac{r}{h} & r & \frac{rr^\top}{h} & rr^\top \\ r & hr & rr^\top & hrr^\top \end{bmatrix}. \quad (5)$$

The construction of Y exploits the fact that the time step h is positive, and is tailored to the bilinear structure of the OCP. Notice that y includes the second-order term hr but excludes h^2 and $r \otimes r$, which would be part of an ordinary second-order SDR [33], [34]. The multiplication by $1/h$ in the matrix Y transforms the terms $h^2 rr^\top$ and h^2, which do not appear in the OCP (4), into $hrr^\top$ and h, which are necessary to linearize the objective (4a).

We report the detailed time-flexible relaxation (TFR) program in Appendix , see (13). There, additional constraints obtained by multiplying linear constraints of the original QCQP are added. While these constraints are redundant for the original nonconvex formulation, they play a crucial role in tightening the SDR.

B. Sparse Time-Flexible Relaxation

Although the TFR provides empirically tight solutions to all the control problems considered in this paper, it can be computationally expensive. Specifically, the number of variables in y is doubled compared to r, which consequently squares the number of variables in Y. Therefore this dense SDR can present significant scalability challenges for more complex systems. To address these challenges, we leverage the sparsity in the OCP and alleviate the computational demands. In particular, we exploit the fact that only the specific group of variables (x_k, x_{k+1}, u_k) are coupled in the costs and constraints. This allows us to build an SDR using the variables

$$r_k = \begin{bmatrix} x_k \\ x_{k+1} \\ u_k \end{bmatrix}, \quad y_k = \begin{bmatrix} 1 \\ h \\ r_k \\ hr_k \end{bmatrix}, \quad Y_k = \frac{1}{h}y_k y_k^\top. \quad (6)$$
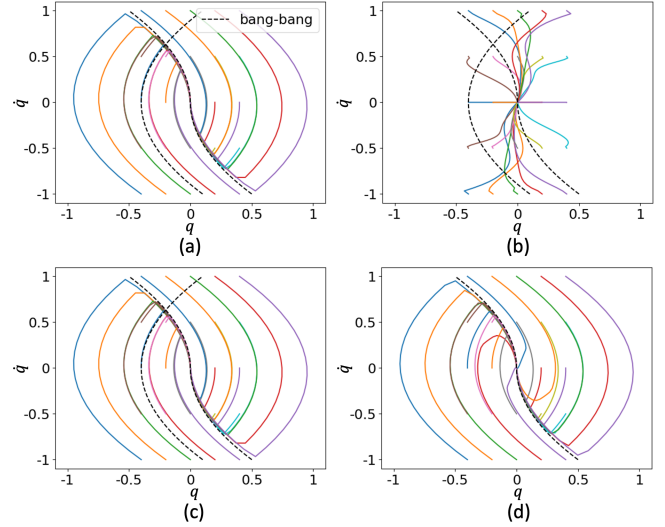

Fig. 2: Comparison of different SDRs for the minimum-time double integrator. (a) Ground truth state-space trajectories. (b) Trajectories from standard SDR. (c) Trajectories from dense TFR. (d) Trajectories from sparse TFR.

To ensure state variables are consistent across time steps, we impose the equality constraints

$$r_k[n_x : 2n_x] = r_{k+1}[: n_x], \quad (7)$$

where the notation follows the Python indexing rules. Additional constraints that further tighten the relaxation are detailed in Appendix , see (14).

Example IV.1 (Minimum-time double integrator). *Fig. 2 illustrates the tightness of different SDRs on a minimum-time double integrator problem, whose optimal solution is shown in Fig. 2(a). In this example, K is set to 30, and the minimum-time objective excludes the quadratic state and control penalties ($Q = R = 0$), making it amenable to the standard first-order SDR. Fig. 2(b) shows that the standard dense first-order SDR results in a loose approximation of the closed-loop trajectories. Although a dense second-order SDR could potentially provide tighter relaxations, it runs out of memory for this problem due to the large value of K. In contrast, Fig. 2(c) shows that our dense TFR precisely recovers the optimal controller. Fig. 2(d) illustrates that the sparse TFR produces trajectories that closely match the optimal bang-bang policy, while being computationally more efficient than the dense formulation.*

V. PIECEWISE-AFFINE SYSTEMS WITH A FIXED MODE SEQUENCE

We now extend our TFR to handle optimal control of discrete-time PWA systems of the form

$$x_{k+1} = x_k + h(A_i x_k + B_i u_k + c_i)$$
$$\text{if } (x_k, u_k) \in \mathcal{X}_i \times \mathcal{U}_i, \quad (8)$$

with the system mode $i \in \{1, \ldots, s\}$ and polytopic domains $\mathcal{X}_i \in \mathbb{R}^{n_x}, \mathcal{U}_i \in \mathbb{R}^{n_u}$. Here, we assume that we are given the number of mode switches N and the fixed mode sequence

$i_1, \ldots, i_N$ traversed by the system (the optimization of the mode sequence will be discussed in Section VI). We divide the trajectory into N segments. In each segment, the system evolves for K_n time steps under a specific mode. The objective is to find the optimal trajectory $\{x_k^n\}_{k=0}^{K_n}, \{u_k^n\}_{k=0}^{K_n-1}$ and time duration h_n in each mode $n \in \{1, \ldots, N\}$. This results in the optimization problem

$$\text{minimize} \quad \eta \sum_{n=1}^{N} K_n h_n + \sum_{n=1}^{N} h_n \sum_{k=0}^{K_n-1} (x_k^{n\top} Q x_k^n + u_k^{n\top} R u_k^n) \tag{9a}$$

$$\text{subject to} \quad \forall n = 1, \ldots, N, \ \forall k = 0, \ldots, K_n - 1 :$$

$$x_{k+1}^n = x_k^n + h_n(A_{i_n} x_k^n + B_{i_n} u_k^n + c_{i_n}) \tag{9b}$$

$$x_k^n \in \mathcal{X}_{i_n}, \ u_k^n \in \mathcal{U}_{i_n} \tag{9c}$$

$$x_0^n = x_{K_{n-1}}^{n-1} \tag{9d}$$

$$x_0^1 = x_{\text{start}}, \ x_{K_N}^N = x_{\text{goal}}. \tag{9e}$$

Constraint (9d) ensures that the state trajectory is continuous during the mode transition. Instead of applying the TFR to the entire program (9), we modularize the computation and take the TFR of (9a)–(9c) in each mode independently. This TFR is denoted as S_n, and has variables $Y^n = y^n y^{n\top}/h_n$ where $r^n = (x_0^n, \ldots, x_K^n, u_0^n, \ldots, u_{K-1}^n)$ and $y^n = (1, h_n, r^n, h_n r^n)$. The boundary conditions (9e) are imposed in the initial and terminal mode, respectively, as in (13d)–(13f).

We highlight that this modular relaxation significantly reduces the number of binary variables that parameterize a mode switch, from being proportional to the total number of time steps (as in traditional MICP formulations) to being proportional to the number of mode switches. This significantly reduces the computational cost of the MICP transcription.

A. Continuity Constraints with Coupling Matrix

To effectively enforce the state continuity constraints (9d), we introduce a vector $\bar{y}$ that couples the time steps and equivalent states in consecutive modes. For all $n = 1, \ldots, N - 1$, we define

$$\bar{x} = x_{K_n}^n = x_0^{n+1}, \ \bar{Y} = \bar{y}\bar{y}^\top$$
$$\bar{y} = \left(1, h_n, h_{n+1}, \bar{x}, h_n \bar{x}, h_{n+1}\bar{x}, \frac{\bar{x}}{h_n}, \frac{\bar{x}}{h_{n+1}}\right). \tag{10}$$

The coupling PSD matrix $\bar{Y}$ includes terms such as:

$$\bar{x}\bar{x}^\top, \ h_n \bar{x}, \ h_{n+1}\bar{x}, \ \frac{\bar{x}}{h_n}, \ \frac{\bar{x}}{h_{n+1}}, \ h_n \bar{x}\bar{x}^\top, \ h_{n+1}\bar{x}\bar{x}^\top, \ \frac{\bar{x}\bar{x}^\top}{h_n}, \ \frac{\bar{x}\bar{x}^\top}{h_{n+1}},$$

which are designed to precisely match the corresponding entries in Y^n and Y^{n+1} of consecutive modes. This ensures that both the state variables and the time-dependent elements of the system's dynamics are accurately aligned across the mode transitions, effectively tightening the SDR. The coupling matrix $\bar{Y}$ allows us to enforce crucial additional constraints on the PSD matrices Y^n paired with consecutive modes, especially those involving h_n, which are critical for maintaining system continuity and optimizing the overall trajectory.
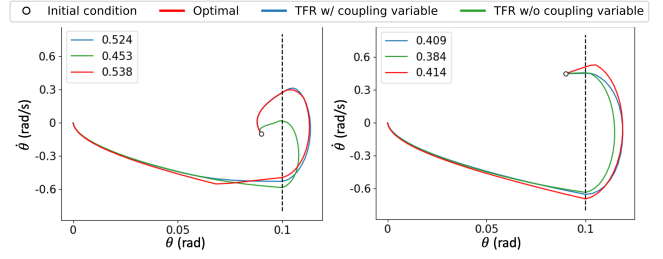

Fig. 3: Effect of the coupling variable on balancing the inverted pendulum with wall in minimum time for different initial conditions. The vertical dashed line indicates the angular limit beyond which the pendulum collides with the wall. The legend indicates the time duration for each trajectory.

Example V.1 (Minimum-time Inverted Pendulum with Wall). *Fig. 3 illustrates the effectiveness of the coupling matrix $\bar{Y}$ in adding useful continuity constraints. The task is to balance a linearized inverted pendulum around the upright equilibrium between elastic walls in minimum time. The system parameters are adopted from [24], with the state defined as $x = (\theta, \dot{\theta})$. For different initial conditions, the TFR trajectory that incorporates the coupling variable consistently provides a closer approximation to the optimal solution.*

VI. MODE OPTIMIZATION FOR PIECEWISE-AFFINE SYSTEMS

In the previous section, we discussed how to generate optimal trajectories for PWA systems with a predefined mode sequence. Here, we show how to optimize the system trajectory and the mode sequence jointly, while respecting the dynamics in each mode. We think of the different mode sequences as different paths in a graph of modes, and we formulate the OCP as a shortest-path problem in GCS [27]. The convex sets and functions in this GCS will be constructed leveraging our TFR.

A. Shortest Paths in Graphs of Convex Sets

A GCS is a directed graph $G = (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Each vertex $v \in \mathcal{V}$ of a GCS is associated with a convex compact set $\mathcal{X}_v$ and a continuous variable x_v inside it. Instead of being a constant as in an ordinary weighted graph, the length of an edge $e = (u, v) \in \mathcal{E}$ in a GCS is a nonnegative convex function $l_e(x_u, x_v)$ of the continuous variables paired with the edge endpoints. Each edge is also paired with a convex constraint of the form $(x_u, x_v) \in \mathcal{X}_e$.

The shortest-path problem in a GCS is formulated as follows:

$$\text{minimize} \quad \sum_{e=(u,v)\in\mathcal{E}_p} l_e(x_u, x_v) \tag{11a}$$

$$\text{subject to} \quad p \in \mathcal{P} \tag{11b}$$

$$x_v \in \mathcal{X}_v, \qquad \forall v \in \mathcal{V} \tag{11c}$$

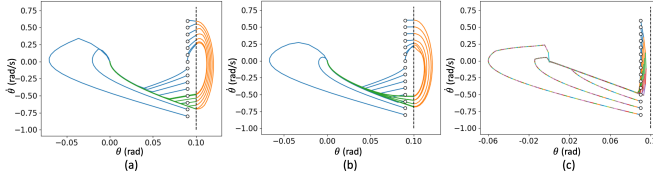$$(x_u, x_v) \in \mathcal{X}_e, \qquad \forall e = (u, v) \in \mathcal{E}_p, \tag{11d}$$

Fig. 4: Time-optimal trajectories for the inverted pendulum with wall. (a) GCS+TFR+NLP solutions. The vertical dashed line represents the angle at which the pendulum makes contact with the wall. (b) GCS+TFR solutions. (c) Convex relaxation of MICP formulation with fixed h. The different segment colors indicate mode transitions in (a) and (b), and illustrate linear interpolations between two sample points in (c).

where p is a path between two pre-specified vertices in G, $\mathcal{P}$ is the family of such paths, and $\mathcal{E}_p$ denotes the set of edges traversed by p. The decision variables are both the discrete path p and the continuous variables x_v for all $v \in \mathcal{V}$. The shortest-path problem in GCS is NP-hard [27, Theorem 1]. However, the techniques proposed in [27] allow us to derive an empirically tight convex relaxation of this problem.

We will model our PWA OCP as a shortest-path problem in GCS, where different paths represent different mode sequences. Then we will use our TFR, concurrently with the techniques from [27], to derive a unified convex relaxation of the PWA control problem.

B. Spectrahedra as Convex Sets $\mathcal{X}_v$

We associate each mode i with a vertex $v \in \mathcal{V}$ in GCS and construct a spectrahedron (see, e.g., [47, Sec. 2.1.2]) as the feasible set of each semidefinite program S_i. These spectrahedra encode the relaxed dynamical constraints, and serve as convex sets $\mathcal{X}_v$ in the GCS. Similarly, their corresponding PSD matrices Y^i serve as the continuous variables, x_v.

C. Optimal Control of PWA Systems with Time Scaling

We now outline the workflow of convex optimal control with time scaling for PWA systems. For each mode i, we construct the spectrahedron from S_i as the convex set and solve GCS to determine the mode sequence $p = (i_1, \ldots, i_N)$ with an approximate optimal trajectory $(r_{i_1}, \ldots, r_{i_N})$ satisfying relaxed dynamical constraints. We then refine the trajectory for the mode sequence p using nonlinear programming solvers for the original bilinear program (9) (not the SDR) to obtain an optimal and dynamically-consistent trajectory from the good initial guess given by the combination of GCS and our TFR (henceforth GCS+TFR).

VII. Additional Experimental Results

We demonstrate the effectiveness of our approach on an inverted pendulum with contact and a 7DoF robot arm. By convexifying the system dynamics and the mode scheduling problem, while enabling flexible time duration, we are able to generate optimal trajectories that are challenging for traditional NLP or MICP methods.
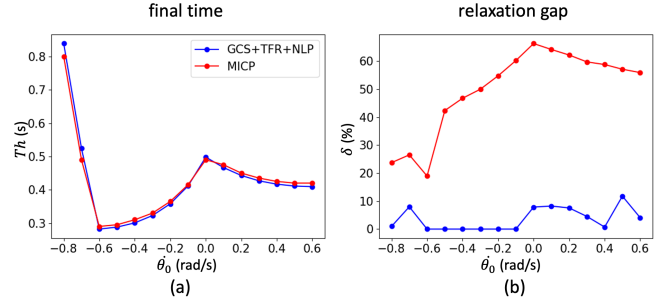

Fig. 5: Performance comparison for balancing the inverted pendulum with contact in minimum time. The pendulum starts from the initial conditions $(0.09, \dot{\theta}_0)$. (a) Final time. (b) Relaxation gap. The proposed GCS+TFR formulation is much tighter than the relaxation of the standard MICP.
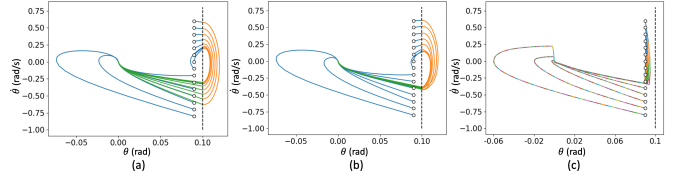

Fig. 6: Trajectories for balancing the inverted pendulum with the time and control-effort cost. (a) GCS+TFR+NLP solutions. (b) GCS+TFR solutions. (c) Convex relaxation of MICP.

A. Inverted Pendulum with Elastic Wall

We apply our method to balance an inverted pendulum around the upright equilibrium between elastic walls as in Example V.1. We compare TFR with the fixed-time-step mixed-integer convex formulation for PWA systems proposed in [26], [48]. The MICP approach uses a time step of $h = 0.005$ and $K = 160$ for a comparable discretization resolution, and can only find final times that are multiple of the fixed time step. In contrast, our method uses $K = 20$ per mode and allows for finer, floating-point time steps. Fig. 4 shows the minimum-time trajectories for different initial conditions. The GCS+TFR solution in Fig. 4(b) closely approximates the globally optimal solution refined by the NLP solver in Fig. 4(a). In contrast, the convex relaxation of MICP in Fig. 4(c) is very loose, resulting in trajectories that consistently miss contacts with the wall. Fig. 5(a) shows the minimum time required to balance the inverted pendulum using both methods. We compare the tightness of GCS+TFR with the convex relaxation of MICP in Fig. 5(b). The relaxation gap for our method is defined as $\delta = (C_{\text{NLP}} - C_{\text{GCS}})/C_{\text{NLP}}$, where C_{NLP} and C_{GCS} are the costs of GCS+TFR+NLP and GCS+TFR respectively. For MICP, $\delta = (C_{\text{opt}} - C_{\text{relax}})/C_{\text{opt}}$, where C_{opt} is the optimal cost and C_{relax} is the cost of the relaxed program.

Our approach is not limited to minimum-time problems. To show this, in Fig. 6 we report the trajectories generated to stabilize the pendulum with both time and control-effort cost ($\eta = R = 1$ and $Q = 0$). Fig. 7 reports the corresponding relaxation performance.
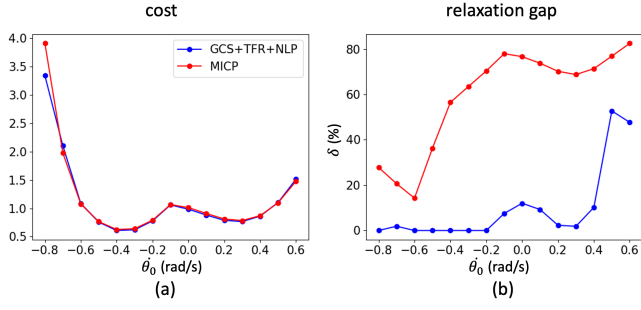
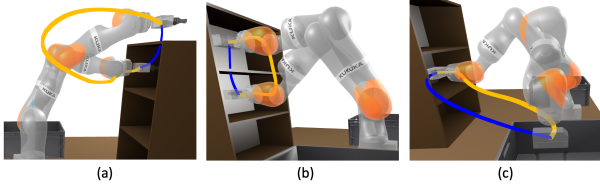Fig. 7: Performance comparison for balancing the inverted pendulum with a time and a control effort penalty.

Fig. 8: Minimum-time trajectories for the robot arm. Videos of the trajectories can be found at (a), (b), (c).

B. Robot Arm

We consider optimizing trajectores and time scaling of a 7D double integrator operating in an environment containing a bookshelf and two bins Fig. 1, with LBR iiwa robot arm geometry. We consider this a step towards reasoning about the full manipulator dynamics, and plan to address the remaining challenges in future work.

The obstacle-free configuration space for the 7 DoF LBR iiwa arm is approximately decomposed into 10 convex regions using the IRIS-NP algorithm [49]. Our method determines the optimal sequence of convex regions to traverse, as well as the optimal trajectories within each region. We compare TFR with direct collocation [2] on 5 tasks for moving from different initial to final configurations. While direct collocation imposes obstacle avoidance as nonlinear constraints, our approach inherently handles these constraints by operating within the convex regions. Fig. 8 visualizes the robot arm's time-optimal trajectories for moving (a) from above the shelf to the top shelf, (b) from the top shelf to the middle shelf, (c) from the middle shelf to the left bin. In all tasks, direct collocation fails to find collision-free paths, even when initialized with the trajectories produced by TFR. Previous GCS and sampling-based methods were also unable to find optimal trajectories that strictly adhere to input limits with a time penalty. Our proposed formulation consistently produces empirically tight solutions as reported in Tab. I, leading to fast, smooth, and collision-free trajectories that respect the dynamical constraints in complex environments.

VIII. Conclusion

In this paper, we propose a novel SDR for the optimal control of linear and PWA systems with time scaling. This framework utilizes techniques from semidefinite programming, exploits the structure of the OCP, and enables the efficient computation of globally optimal solutions. Our

Task	0	1	2	3	4
Relaxation gap δ (%)	0.9	2.1	14.3	24.4	5.8

TABLE I: Relaxation gap of the TFR for moving the robot arm to different configurations in minimum time.

formulation allows optimizing over the time scaling and the state and control trajectories jointly, in contrast to common bilevel methods. In addition, our TFR can be integrated into the GCS framework for mode optimization. This enables the joint optimization of the mode sequence and continuous system trajectories within a unified convex program.

Appendix

This appendix details the proposed semidefinite program with the additional constraints that tighten our TFR. We consider the following abstract representation for the nonconvex optimal control program with time scaling for linear systems (1):

$$\text{minimize } \eta \cdot Kh + \text{tr}((Q_x + R_u) \cdot hrr^\top) \tag{12a}$$

$$\text{subject to } Fr = Ghr \quad \text{(dynamical constraints)} \tag{12b}$$

$$Cr + d \geq 0 \quad \text{(input and state constraints)} \tag{12c}$$

$$Hr = b \quad \text{(terminal conditions)} \tag{12d}$$

$$h \geq 0. \tag{12e}$$

Our SDR reads

$$\text{minimize } \eta \cdot Kh + \text{tr}((Q_x + R_u) \cdot hrr^\top) \tag{13a}$$

$$\text{subject to } Fr = Ghr, \ Fr/h = Gr \tag{13b}$$

$$Frr^\top = Ghrr^\top, \ Frr\top/h = Grr^\top \tag{13c}$$

$$Hr = b, \ Hhr = bh \tag{13d}$$

$$Hr/h = b/h, \ Hrr^\top = br^\top \tag{13e}$$

$$Hhrr^\top = bhr^\top, \ Hrr\top/h = br^\top/h \tag{13f}$$

$$Cr + d \geq 0, \ (Cr + d)(Cr + d)^\top \geq 0 \tag{13g}$$

$$h(Cr + d) \geq 0, \ h(Cr + d)(Cr + d)^\top \geq 0 \tag{13h}$$

$$(Cr + d)/h \geq 0, \ (Cr + d)(Cr + d)^\top/h \geq 0 \tag{13i}$$

$$h \geq 0, \ 1/h \geq 0, \tag{13j}$$

where the products of variables are replaced by the corresponding entries from the matrix Y in (5). The additional equality constraints necessary for the sparse TFR are

$$hr_k [n_x : 2n_x] = hr_{k+1}[: n_x]$$

$$\frac{r_k}{h} [n_x : 2n_x] = \frac{r_{k+1}}{h}[: n_x]$$

$$r_k r_k^\top [n_x : 2n_x, n_x : 2n_x] = r_{k+1} r_{k+1}^\top[: n_x, : n_x]$$

$$hr_k r_k^\top [n_x : 2n_x, n_x : 2n_x] = hr_{k+1} r_{k+1}^\top[: n_x, : n_x] \tag{14}$$

$$\frac{r_k r_k^\top}{h} [n_x : 2n_x, n_x : 2n_x] = \frac{r_{k+1} r_{k+1}^\top}{h}[: n_x, : n_x].$$

References

[1] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[2] M. Kelly, "An introduction to trajectory optimization: How to do your own direct collocation," *SIAM Review*, vol. 59, no. 4, pp. 849–904, 2017.

[3] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.

[4] E. F. Camacho and C. Bordons, *Model Predictive Control*. Springer, 2007.

[5] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Science Robotics*, vol. 6, no. 56, p. eabh1221, 2021.

[6] D. Dong, H. Berger, and I. Abraham, "Time optimal ergodic search," *arXiv preprint arXiv:2305.11643*, 2023.

[7] G. Tang, W. Sun, and K. Hauser, "Time-optimal trajectory generation for dynamic vehicles: A bilevel optimization approach," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 7644–7650.

[8] T. Marcucci, P. Nobel, R. Tedrake, and S. Boyd, "Fast path planning through large collections of safe boxes," *IEEE Transactions on Robotics*, 2024.

[9] N. R. Kapania, J. Subosits, and J. Christian Gerdes, "A sequential two-step algorithm for fast generation of vehicle racing trajectories," *Journal of Dynamic Systems, Measurement, and Control*, vol. 138, no. 9, p. 091005, 2016.

[10] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics Research: The 16th International Symposium ISRR*. Springer, 2016, pp. 649–666.

[11] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *Journal of Guidance, Control, and Dynamics*, vol. 10, no. 4, pp. 338–342, 1987.

[12] A. Richards and J. P. How, "Robust variable horizon model predictive control for vehicle maneuvering," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 16, no. 7, pp. 333–351, 2006.

[13] F. D. Carvallo, A. W. Westerberg, and M. Morari, "Milp formulation for solving minimum time optimal control problems," *International Journal of Control*, vol. 51, no. 4, pp. 943–947, 1990.

[14] H. P. Rothwangl, "Numerical synthesis of the time optimal nonlinear state controller via mixed integer programming," in *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, vol. 4. IEEE, 2001, pp. 3201–3205.

[15] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No. CH37301)*, vol. 3. IEEE, 2002, pp. 1936–1941.

[16] T. Schouwenaars, B. De Moor, E. Feron, and J. How, "Mixed integer programming for multi-vehicle path planning," in *2001 European Control Conference (ECC)*. IEEE, 2001, pp. 2603–2608.

[17] R. Deits and R. Tedrake, "Efficient mixed-integer planning for uavs in cluttered environments," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 42–49.

[18] B. Landry, R. Deits, P. R. Florence, and R. Tedrake, "Aggressive quadrotor flight through cluttered environments using mixed integer programming," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1469–1475.

[19] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 279–286.

[20] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, 2016.

[21] B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernández-López, and C. Semini, "Simultaneous contact, gait, and motion planning for robust multilegged locomotion via mixed-integer convex optimization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2531–2538, 2017.

[22] L. Blackmore and B. Williams, "Optimal manipulator path planning with obstacles using disjunctive programming," in *2006 American Control Conference*. IEEE, 2006, pp. 3–pp.

[23] H. Ding, G. Reißig, D. Groß, and O. Stursberg, "Mixed-integer programming for optimal path planning of robotic manipulators," in *2011 IEEE International Conference on Automation Science and Engineering*. IEEE, 2011, pp. 133–138.

[24] T. Marcucci, R. Deits, M. Gabiccini, A. Bicchi, and R. Tedrake, "Approximate hybrid model predictive control for multi-contact push recovery in complex environments," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 31–38.

[25] F. R. Hogan and A. Rodriguez, "Feedback control of the pusher-slider system: A story of hybrid and underactuated contact dynamics," *arXiv preprint arXiv:1611.08268*, 2016.

[26] T. Marcucci and R. Tedrake, "Mixed-integer formulations for optimal control of piecewise-affine systems," in *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, 2019, pp. 230–239.

[27] T. Marcucci, J. Umenberger, P. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 507–532, 2024.

[28] T. Marcucci, "Graphs of convex sets with applications to optimal control and motion planning," Ph.D. dissertation, Massachusetts Institute of Technology, 2024.

[29] T. Marcucci, M. Petersen, D. von Wrangel, and R. Tedrake, "Motion planning around obstacles with convex optimization," *Science Robotics*, vol. 8, no. 84, p. eadf7843, 2023.

[30] T. Cohn, M. Petersen, M. Simchowitz, and R. Tedrake, "Non-euclidean motion planning with graphs of geodesically-convex sets," in *Robotics: Science and Systems (RSS)*, 2023.

[31] B. P. Graesdal, S. Y. Chia, T. Marcucci, S. Morozov, A. Amice, P. A. Parrilo, and R. Tedrake, "Towards tight convex relaxations for contact-rich manipulation," in *Robotics: Science and Systems (RSS)*, 2024.

[32] P. A. Parrilo, *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[33] ——, "Semidefinite programming relaxations for semialgebraic problems," *Mathematical Programming*, vol. 96, pp. 293–320, 2003.

[34] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[35] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 20–34, 2010.

[36] B. El Khadir, J. B. Lasserre, and V. Sindhwani, "Piecewise-linear motion planning amidst static, moving, or morphing obstacles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7802–7808.

[37] S. Teng, A. Jasour, R. Vasudevan, and M. Ghaffari, "Convex geometric motion planning on lie groups via moment relaxation," *arXiv preprint arXiv:2305.13565*, 2023.

[38] X. Zhang, Z. Cheng, J. Ma, S. Huang, F. L. Lewis, and T. H. Lee, "Semi-definite relaxation-based admm for cooperative planning and control of connected autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 9240–9251, 2021.

[39] J. B. Lasserre, D. Henrion, C. Prieur, and E. Trélat, "Nonlinear optimal control via occupation measures and lmi-relaxations," *SIAM Journal on Control and Optimization*, vol. 47, no. 4, pp. 1643–1666, 2008.

[40] Y. Jiang and Z.-P. Jiang, "Global adaptive dynamic programming for continuous-time nonlinear systems," *IEEE Transactions on Automatic Control*, vol. 60, no. 11, pp. 2917–2929, 2015.

[41] L. Yang, H. Dai, A. Amice, and R. Tedrake, "Approximate optimal controller synthesis for cart-poles and quadrotors via sums-of-squares," *IEEE Robotics and Automation Letters*, 2023.

[42] S. Kang, X. Xu, J. Sarva, L. Liang, and H. Yang, "Fast and certifiable trajectory optimization," *arXiv preprint arXiv:2406.05846*, 2024.

[43] P. Trutman, M. S. El Din, D. Henrion, and T. Pajdla, "Globally optimal solution to inverse kinematics of 7dof serial manipulator," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6012–6019, 2022.

[44] F. Marić, M. Giamou, S. Khoubyarian, I. Petrović, and J. Kelly, "Inverse kinematics for serial kinematic chains via sum of squares optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7101–7107.

[45] F. Kahl and D. Henrion, "Globally optimal estimates for geometric reconstruction problems," *International Journal of Computer Vision*, vol. 74, pp. 3–15, 2007.

[46] H. Yang and L. Carlone, "Certifiably optimal outlier-robust geometric perception: Semidefinite relaxations and scalable global optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 2816–2834, 2022.

[47] G. Blekherman, P. A. Parrilo, and R. R. Thomas, *Semidefinite optimization and convex algebraic geometry*. SIAM, 2012.

[48] N. Moehle and S. Boyd, "A perspective-based convex relaxation for switched-affine optimal control," *Systems & Control Letters*, vol. 86, pp. 34–40, 2015.

[49] M. Petersen and R. Tedrake, "Growing convex collision-free regions in configuration space using nonlinear programming," *arXiv preprint arXiv:2303.14737*, 2023.