Title
Efficient Global Solutions to Single-Input Optimal Control Problems via Approximation by Sum-of-Squares Polynomials

Journal
IEEE Transactions on Automatic Control, 67(9)

ISSN
0018-9286

Authors
Rodrigues, Diogo
Mesbah, Ali

Publication Date
2022-09-01

DOI
10.1109/tac.2022.3165481

Peer reviewed

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

Efficient Global Solutions to Single-Input Optimal Control Problems via Approximation by Sum-of-Squares Polynomials

Diogo Rodrigues and Ali Mesbah

***Abstract*—Optimal control problems are prevalent in model-based control, state and parameter estimation, and experimental design for complex dynamical systems. An approach for obtaining solutions to these problems is based on the notion of parsimonious input parameterization and comprises two tasks: the enumeration of arc sequences followed by the computation of optimal values of a small number of decision variables for each sequence. This paper proposes an efficient global solution method for single-input optimal control problems for nonlinear dynamical systems with a potentially large number of states or complex dynamics via sum-of-squares polynomials and parallel computing. The method approximates the problem for a given arc sequence as a polynomial optimization problem that can be efficiently solved to global optimality via semidefinite programming. It is established that the difference between the cost obtained by the proposed method and the globally optimal cost of the original problem is bounded and depends on the polynomial approximation error. The method is illustrated by simulation examples of a reaction system and a rocket.**

***Index Terms*—Optimal control; Global optimization; Polynomial optimization; Sum-of-squares polynomials; Semidefinite programming; Nonlinear systems**

I. Introduction

Optimal control problems (OCPs) are extensively applied for optimal design, analysis, and operation of a wide range of complex dynamical systems. Efficient solution methods for OCPs are useful for optimization-based state and parameter estimation, experimental design, and model-based control, among other tasks in engineering applications. In OCPs, the selected decision variables represent time-varying functions over a time interval such that a cost is optimized subject to constraints. OCPs are generally complex to solve since they involve infinitely many decision variables, and typically there exist not only terminal constraints at the end of the time interval but also path constraints along the trajectory [1].

Direct methods are a popular solution approach for OCPs wherein the original infinite-dimensional problem is approximated as a finite-dimensional one via discretization of the time-varying functions [2], [3]. However, direct methods only seek local optimality, rather than global optimality. The local optima attained by these optimization algorithms may be suboptimal with respect to the global optimum by a significant margin [4]. In contrast, indirect methods seek a solution to the necessary conditions of optimality by using Pontryagin's maximum principle or the Hamilton-Jacobi-Bellman equation [5], [6]. Although methods based on the Hamilton-Jacobi-Bellman equation provide global solutions, indirect methods lack scalability with respect to the number of states and can be very sensitive to the choice of initial guess. Alternatively, global optimization algorithms can be used. To this end, two approaches can be highlighted: branch-and-bound approaches and reformulation as a convex problem. Branch-and-bound approaches locate the optimum by dividing the space of decision variables into subsets until the global optimum is found [7]. The alternative is to reformulate the original nonconvex problem as a convex problem with a single local optimum that corresponds to the global optimum of the original problem. For example, if the cost and constraints are explicitly written as polynomial functions, one can express the problem as a polynomial optimization problem (POP), which can be reformulated as a convex semidefinite program (SDP) via the concept of sum-of-squares polynomials [8], [9]. However, with either approach to global optimization, the worst-case complexity scales exponentially with the number of decision variables [4]. Global optimality via direct methods is further complicated by the fact that the number of decision variables is large even after the use of typical discretization methods [10].

The parsimonious input parameterization approach has been proposed to reduce the number of decision variables in OCPs [11], which is useful for attaining global optimality. This approach (i) identifies all the arcs that can occur in the solution to an OCP, (ii) generates a finite set of plausible arc sequences, and (iii) describes each sequence by a small number of decision variables. Then, for a given arc sequence, one can compute the optimal values of these decision variables via numerical optimization. It has been demonstrated that a parsimonious input parameterization reduces the number of decision variables without causing any loss of optimality [11]. This approach can be considered to be similar to the widely known technique of switching point optimization [12]–[15]. However, switching point optimization relies on relatively strong assumptions with respect to the arcs in the solution to the OCP. More specifically, switching point optimization assumes that the arcs are either bang-bang arcs or singular arcs with an analytical expression for the input. This in turn

This work was supported by the Swiss National Science Foundation, project number 184521.

Diogo Rodrigues was and Ali Mesbah is with the Department of Chemical and Biomolecular Engineering, University of California, Berkeley, CA 94720, USA. mesbah@berkeley.edu

Diogo Rodrigues is with Centro de Química Estrutural, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal. dfmr@tecnico.ulisboa.pt

implies that the OCP is not input-affine or is not related to complex nonlinear systems, that is, nonlinear systems with a large number of states or complex dynamics, otherwise analytical expressions for the optimal input in singular arcs are unknown. Input-affine OCPs may require an even number of differentiations (that is, at least two differentiations) of the switching function to obtain the optimal input in singular arcs. This procedure is impractical or even intractable in the case of complex nonlinear systems if the aim is to obtain an analytical expression for the optimal input in singular arcs. In contrast, with the more general formulation of parsimonious input parameterization, any type of singular arcs can be considered. A main advantage of parsimonious input parameterization, which was suggested but not further investigated in [11], is that the small number of decision variables provided by this approach enables (i) accurate approximations of the terminal cost and constraints as explicit polynomial functions of the decision variables and (ii) efficient computation of globally optimal values of the decision variables for each arc sequence via polynomial optimization. The use of parsimonious input parameterization for computation of global solutions via polynomial optimization is investigated in the current paper. This goal motivates the approximation of the terminal cost and constraints as explicit polynomial functions of the decision variables since that yields a POP for each arc sequence. This procedure only requires numerical integration of the dynamic equations of the states and of the adjoint variables for each evaluated value of the decision variables, which addresses the issue of scalability with respect to the number of states. Then, one can compute the global solution to the POP for each arc sequence via reformulation as an SDP, which enables efficient global solutions to OCPs via parallel computing.

Hence, this paper aims to extend the parsimonious input parameterization approach for efficient global solutions to approximations of OCPs for complex nonlinear dynamical systems, that is, with a large number of states or complex dynamics. Particular emphasis is given to the case of single-input OCPs with solutions that can be accurately described by a relatively small number of arcs. For a general OCP formulation, the paper first recalls how to formulate the problem for a given arc sequence in terms of a reduced number of decision variables. The first main contribution of this paper is a method for approximating the latter problem as a POP via multivariate Hermite interpolation and the establishment of a quantifiable bound for the error between the solutions to both problems. As the second main contribution, we demonstrate that the POP can be solved efficiently to global optimality via the concept of sum-of-squares polynomials. Finally, the methods are illustrated via simulation examples.

II. Problem Statement

Consider the general class of OCPs formulated as

$$\min_{\mathbf{u}(\cdot),\mathbf{x}(\cdot),t_f} \quad \mathscr{J}\big(\mathbf{u}(\cdot),t_f\big)=\phi\big(\mathbf{x}(t_1),\ldots,\mathbf{x}(t_T),t_f\big), \tag{1a}$$

$$\text{s.t.} \quad \mathscr{T}\big(\mathbf{u}(\cdot),t_f\big)=\boldsymbol{\psi}\big(\mathbf{x}(t_1),\ldots,\mathbf{x}(t_T),t_f\big)\leq\mathbf{0}_{n_\psi}, \tag{1b}$$

$$\dot{\mathbf{x}}(t)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big), \quad \mathbf{x}(t_0)=\mathbf{x}_0, \tag{1c}$$

$$\underline{\mathbf{u}}\leq\mathbf{u}(t)\leq\overline{\mathbf{u}}, \quad \mathbf{h}\big(\mathbf{x}(t)\big)\leq\mathbf{0}_{n_h}, \tag{1d}$$

where t_0 is the initial time, $t_1<\ldots<t_T$ are T times, $t_f=t_T\in[t_0,t_{max}]$ is the finite final time with upper bound t_{max}; $\mathbf{u}(t)$ is the n_u-dimensional vector of piecewise-continuous inputs for all $t\in\big[t_0,t_f\big)$ with n_u-dimensional vectors of lower and upper bounds $\underline{\mathbf{u}}$ and $\overline{\mathbf{u}}$; $\mathbf{x}(t)$ is the n_x-dimensional vector of piecewise-continuously differentiable states for all $t\in\big[t_0,t_f\big)$; $\mathbf{f}(\mathbf{x},\mathbf{u})$ is an n_x-dimensional vector function, smooth for all $(\mathbf{x},\mathbf{u})\in\mathbb{R}^{n_x}\times\mathbb{R}^{n_u}$; $\mathbf{h}(\mathbf{x})$ is an n_h-dimensional vector function, smooth for all $\mathbf{x}\in\mathbb{R}^{n_x}$; $\phi(\mathbf{X},t)$, $\boldsymbol{\psi}(\mathbf{X},t)$ are a scalar function and an n_ψ-dimensional vector function, respectively, smooth for all $(\mathbf{X},t)\in\mathbb{R}^{Tn_x}\times[t_0,t_{max}]$. We assume that $\mathbf{h}^{(1)}(\mathbf{x},\mathbf{u}):=\frac{\partial\mathbf{h}}{\partial\mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x},\mathbf{u})$ depends explicitly on $\mathbf{u}$.

The inputs that represent the solution to Problem (1) are composed of several arcs. For each input u_j, each arc can be of type 1) bang-bang, such that it is determined by an equality $u_j=\underline{u}_j$ or $u_j=\overline{u}_j$, 2) active-state constraint, such that it is determined by an equality $h_k^{(1)}(\mathbf{x},\mathbf{u})=0$ for some $k=1,\ldots,n_h$, or 3) free, such that it is determined by an equality that stems from the dynamics given by $\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)$, also labeled as singular in the relevant case of input-affine OCPs with $\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)$ affine in $\mathbf{u}(t)$ [11], [16]. Hence, there is a finite number of arc types from which arc sequences can be formed. If we consider as plausible arc sequences only sequences with a number of arcs no larger than some upper bound $\bar{n}_a$ and without consecutive arcs of the same type, it follows that the number of plausible sequences is also finite.

Remark 1. *The effect of non-optimal inputs is different for arcs of different types: a non-optimal input in bang-bang and active-state constraint arcs has an important effect on the cost, while a non-optimal input in free/singular arcs has a negligible effect on the cost [16]. This difference is explained next. We denote the adjoint variables that represent the sensitivity of the Lagrangian* $\Phi\big(\mathbf{x}(t_1),\ldots,\mathbf{x}(t_T),t_f\big):=\phi\big(\mathbf{x}(t_1),\ldots,\mathbf{x}(t_T),t_f\big)+\boldsymbol{\nu}^{\mathrm{T}}\boldsymbol{\psi}\big(\mathbf{x}(t_1),\ldots,\mathbf{x}(t_T),t_f\big)$ *with Lagrange multipliers* $\boldsymbol{\nu}$ *with respect to the states* $\mathbf{x}(t)$ *as* $\boldsymbol{\lambda}(t)$*, the Lagrange multipliers that correspond to the constraints* $\underline{\mathbf{u}}\leq\mathbf{u}(t)$, $\mathbf{u}(t)\leq\overline{\mathbf{u}}$, $\mathbf{h}\big(\mathbf{x}(t)\big)\leq\mathbf{0}_{n_h}$ *as* $\underline{\boldsymbol{\mu}}(t)$, $\overline{\boldsymbol{\mu}}(t)$, $\boldsymbol{\pi}(t)$*, respectively, and the Hamiltonian function as* $H\big(\mathbf{x}(t),\mathbf{u}(t),\boldsymbol{\lambda}(t)\big)=\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)^{\mathrm{T}}\boldsymbol{\lambda}(t)$*. From the interpretation of* $\boldsymbol{\lambda}(t)$ *as state sensitivities, it is possible to conclude that the sensitivity of the Lagrangian with respect to the inputs* $\mathbf{u}(\xi)$ *over an interval* $\xi\in[t,t+\delta t]$ *is* $\int_t^{t+\delta t}\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\big(\mathbf{x}(\xi),\mathbf{u}(\xi)\big)^{\mathrm{T}}\boldsymbol{\lambda}(\xi)\mathrm{d}\xi$*. If this interval is included in a free/singular arc, it is known from Pontryagin's maximum principle that, at the optimal solution, the integrand is also the switching function* $\frac{\partial\mathbf{f}}{\partial\mathbf{u}}\big(\mathbf{x}(\xi),\mathbf{u}(\xi)\big)^{\mathrm{T}}\boldsymbol{\lambda}(\xi)=\frac{\partial H}{\partial\mathbf{u}}\big(\mathbf{x}(\xi),\mathbf{u}(\xi),\boldsymbol{\lambda}(\xi)\big)^{\mathrm{T}}=\mathbf{0}_{n_u}$*. This implies that the first-order approximation of the variation of the Lagrangian due to the variation of an input* $u_j(\xi)$ *over an interval* $\xi\in[t,t+\delta t]$ *in a free/singular arc is equal to zero around the optimal trajectory. On the other hand, at the optimal solution, the sensitivity of the Lagrangian with respect to relaxations of the input constraints* $\underline{\mathbf{u}}\leq\mathbf{u}(\xi)$ *and* $\mathbf{u}(\xi)\leq\overline{\mathbf{u}}$ *over an interval* $\xi\in[t,t+\delta t]$ *where these constraints are active is* $\int_t^{t+\delta t}-\big(\underline{\boldsymbol{\mu}}(\xi)+\overline{\boldsymbol{\mu}}(\xi)\big)\mathrm{d}\xi$*, which is different from zero. Similarly, at the optimal solution, the sensitivity of the Lagrangian with respect to relaxations of active constraints* $\mathbf{h}\big(\mathbf{x}(t)\big)\leq\mathbf{0}_{n_h}$ *is nonzero and depends on*

$\boldsymbol{\pi}(t)$. *This implies that the first-order approximation of the variation of the Lagrangian due to the variation of an input* $u_j(\xi)$ *over an interval* $\xi \in [t, t+\delta t]$ *in a bang-bang or active-state constraint arc is different from zero around the optimal trajectory. Hence, we use the fact that a non-optimal input in free/singular arcs has a negligible effect on the cost to assume that free/singular arcs are approximated by linear functions of time throughout the paper. Even if the true optimal input in a free/singular arc is not a linear function, its approximation by a linear function does not have a significant effect on the cost and leads to a small loss of optimality. On the other hand, since a non-optimal input in bang-bang and active-state constraint arcs has an important effect on the cost, constraint handling is emphasized in the paper.*

Parsimonious input parameterization is an effective approach for describing the optimal inputs using only a few decision variables, in contrast to infinite-dimensional variables in the original OCP [11], [17]. For a given plausible arc sequence composed of n_s+1 bang-bang and free/singular arcs, the inputs are defined by the following decision variables: the switching times $\bar{t}_1, \ldots, \bar{t}_{n_s}$ to arcs of types 1 and 3, the final time $\bar{t}_{n_s+1} = t_f$, and the initial conditions of the free/singular arcs. The difference with respect to switching point optimization is precisely the inclusion of the initial conditions of the free/singular arcs as decision variables, which allow representing singular arcs in input-affine OCPs related to complex nonlinear systems [13]. The entry points in arcs of type 2 are given by the n_η-dimensional vector $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_{n_\eta})$, but the switching to these arcs cannot occur at arbitrary times since it depends on the states $\mathbf{x}$. In this paper, we assume that $\mathbf{h}^{(1)}(\mathbf{x}, \mathbf{u})$ explicitly depends on $\mathbf{u}$ because otherwise it would be impossible to ensure that the state constraint $h_k\big(\mathbf{x}(t)\big) \leq 0$ remains active for $t > \eta$ once an entry point η is reached such that $h_k\big(\mathbf{x}(\eta)\big) = 0$ for some $k = 1, \ldots, n_h$. For example, suppose that $\mathbf{h}^{(1)}(\mathbf{x})$ does not explicitly depend on $\mathbf{u}$ but $\mathbf{h}^{(2)}(\mathbf{x}, \mathbf{u}) := \frac{\partial \mathbf{h}^{(1)}}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x}, \mathbf{u})$ explicitly depends on $\mathbf{u}$. Then, once an entry point η is reached such that $h_k\big(\mathbf{x}(\eta)\big) = 0$ and $h_k^{(1)}\big(\mathbf{x}(\eta)\big) > 0$ for some $k = 1, \ldots, n_h$, there exists no $\mathbf{u}(t)$ that guarantees that $h_k\big(\mathbf{x}(t)\big) \leq 0$ for $t > \eta$. In contrast, if $\mathbf{h}^{(1)}(\mathbf{x}, \mathbf{u})$ explicitly depends on $\mathbf{u}$ as assumed, once an entry point η is reached such that $h_k\big(\mathbf{x}(\eta)\big) = 0$ for some $k = 1, \ldots, n_h$, it is possible to choose $\mathbf{u}(t)$ such that $h_k^{(1)}\big(\mathbf{x}(t), \mathbf{u}(t)\big) = 0$, which ensures that the state constraint $h_k\big(\mathbf{x}(t)\big) \leq 0$ remains active for $t > \eta$. Also, we assume that the optimal sequence of arcs of types 1, 2, and 3 is known for each given sequence of arcs of types 1 and 3 for clarity and convenience, that is, to simplify the exposition in the remainder of the paper, although this assumption is not a requirement.

The goal of this paper is to extend the parsimonious input parameterization approach and show how OCPs formulated as (1) can be solved efficiently to global optimality. Although the methods in this paper can be generalized to OCPs with multiple inputs $\mathbf{u}(t)$, these methods may be sensitive to the number of inputs because it affects the number of arc sequences. For this reason, in the remainder we consider, without loss of generality, OCPs with a single input $u(t)$, that is, $n_u = 1$, not only for the sake of clarity, but also because the proposed methods are most efficient in this case. Hence, this paper presents a method for efficiently solving polynomial approximations of single-input OCPs to global optimality, in particular for OCPs with solutions that can be accurately described by a relatively small number of arcs. The proposed approach for global optimality relies on using the dynamics to determine: (i) when and how the globally optimal switching between arcs takes place for a given plausible arc sequence; and (ii) which sequence provides the globally optimal solution. The plausible arc sequences for which global solutions are found can be chosen as shown in Section III. Then, addressing question (i) consists in computing the globally optimal values of the decision variables for the given arc sequence. For this, we represent the cost and constraints of the OCP as explicit polynomial functions since that converts the OCP into a set of POPs, one for each arc sequence, as shown in Section IV. Then, each POP is solved to global optimality as shown in Section V. Once question (i) is addressed for each sequence via parallel computing, it is trivial to answer question (ii) efficiently.

III. Choice of Arc Sequences

Now we describe how one can choose the arc sequences for which global solutions are found. As mentioned in Section II, the plausible arc sequences under consideration are the sequences with a number of arcs no larger than some upper bound $\bar{n}_a$ and without consecutive arcs of the same type. Suppose that we denote the bang-bang arcs as 1L or 1U, depending on whether they are determined by $u = \underline{u}$ or $u = \overline{u}$. Indeed, it would be implausible to have consecutive arcs of the same type: (i) two consecutive arcs of type 1L would be equivalent to a single arc of type 1L, and the same is valid for two consecutive arcs of type 1U; (ii) two consecutive arcs of type 3 would be represented by two consecutive linear functions, which would imply that an arc of type 3 needs to be approximated by a piecewise-linear function with two pieces and would contradict the assumption that an arc of type 3 can be approximated by a linear function. In addition, note that: (i) sequences with fewer than $\bar{n}_a$ arcs are particular cases of the sequences with $\bar{n}_a$ arcs where some arcs vanish, and (ii) the sequences that end with an arc of type 3 are not plausible in input-affine OCPs since they imply that the states $\mathbf{x}(t_f)$ at the final time must lie in a set of measure zero where $0 = \frac{\partial \mathbf{f}}{\partial u}\big(\mathbf{x}(t_f), u(t_f)\big)^{\mathrm{T}} \boldsymbol{\lambda}(t_f) = \frac{\partial \mathbf{f}}{\partial u}\big(\mathbf{x}(t_f), u(t_f)\big)^{\mathrm{T}} \frac{\partial \Phi}{\partial \mathbf{x}(t_T)}\big(\mathbf{x}(t_1), \ldots, \mathbf{x}(t_T), t_f\big)^{\mathrm{T}}$ as given by Pontryagin's maximum principle [5], [18]. Hence, by recalling that plausible arc sequences do not have consecutive arcs of the same type, all these sequences can be determined as follows: the final arc of an input-affine OCP can only be of type 1L or 1U; and each one of the previous arcs can be of type 1L, 1U, or 3, but not of the same type as its successor. This implies that the branching factor is 2 for each arc in a plausible sequence, and the number of plausible sequences is equal to $2^{\bar{n}_a}$ for an input-affine OCP or $\frac{3}{2}2^{\bar{n}_a}$ otherwise.

However, in many cases, it is reasonable to assume that one seeks a solution that includes a relatively small number

of arcs with an upper bound such as $\bar{n}_a$ between 3 and 5 since in practice one is not interested in chattering solutions with arcs of infinitesimal duration even if these solutions are globally optimal from a mathematical point of view [16], [17]. Note that such solutions are difficult to describe accurately by any input parameterization, including the prevalent piecewise-constant parameterization, unless a very large number of decision variables is used. Hence, while the proposed method may not be applicable to such problems, it is also true that efficient global solutions to problems of this type seem to be currently out of reach with any known method. For this reason, this paper focuses on problems with solutions that can be accurately described by a relatively small number of arcs. However, while the approach in this paper can guarantee global optimality for the OCP only if its globally optimal solution does not include more than $\bar{n}_a$ arcs, global optimality among all plausible arc sequences can be guaranteed in any case, which approximates the globally optimal cost even if the globally optimal solution includes more than $\bar{n}_a$ arcs. If there exists a single-input OCP for which the globally optimal cost with many arcs is significantly better than the globally optimal cost among all plausible arc sequences, then that OCP would always need to be described by a very large number of decision variables for numerical optimization, which would make that problem intractable with any existing method for global optimization. On the other hand, the proposed approach is useful for global optimization of all the remaining single-input OCPs.

In addition, one can note that the arcs of types 1L and 1U can be seen as particular cases of arcs of type 3 since in arcs of types 1L and 1U the input is constant, while in arcs of type 3 the input is assumed to be approximated by a linear function. This implies that certain arc sequences do not need to be considered. For example, for $\bar{n}_a = 3$, the sequence 1U-1L-1U does not need to considered because it is a particular case of the sequences 1U-3-1U and 3-1L-1U. However, it is not recommendable to express all the arc sequences as particular cases of a hypothetical sequence with $\bar{n}_a$ arcs of type 3 since the number of decision variables for that sequence would be excessive.

Hence, the algorithm that chooses and exhaustively encompasses all the plausible arc sequences in input-affine OCPs with a number of arcs no larger than $\bar{n}_a$ is as follows:

1) If $\bar{n}_a = 2$, start only with an empty sequence; if $\bar{n}_a > 2$, start with an empty sequence, a sequence 1L, and a sequence 1U. For each sequence, go to the next step.
2) Create two copies of the sequence and append 3-1L to one and 3-1U to the other one. For each sequence, go to the next step.
3) If the number of arcs is equal to $\bar{n}_a$, return the sequence; if the number of arcs is equal to $\bar{n}_a - 1$, go to the next step; if the number of arcs is equal to $\bar{n}_a - 2$, go to step 2; otherwise create two copies of the sequence and go to step 2 with one and to the next step with the other one.
4) Append 1L if the last arc was 1U or append 1U if the last arc was 1L. If the number of arcs is equal to $\bar{n}_a$, return the sequence; otherwise go to step 2.

By applying this algorithm, one can obtain fewer than $2^{\bar{n}_a}$ arc sequences with $\bar{n}_a$ arcs. For example, for $\bar{n}_a = 3$, the 6 sequences are 3-1L-1U, 3-1U-1L, 1L-3-1L, 1L-3-1U, 1U-3-1L, 1U-3-1U, that is, fewer than $2^{\bar{n}_a} = 8$ sequences. For $\bar{n}_a = 4$, the 8 sequences are 3-1L-3-1L, 3-1L-3-1U, 3-1U-3-1L, 3-1U-3-1U, 1L-3-1L-1U, 1L-3-1U-1L, 1U-3-1L-1U, 1U-3-1U-1L, that is, fewer than $2^{\bar{n}_a} = 16$ sequences.

IV. Reformulation of the OCP as Polynomial Optimization Problems

This section shows how to reformulate the OCP (1) as a set of POPs, one for each plausible arc sequence.

A. OCP with new decision variables

For a given arc sequence, we describe the input in the ith time interval $[\bar{t}_{i-1}, \bar{t}_i)$, for $i = 1, \ldots, n_s + 1$, by defining $n_{z,i}$ new states and initial conditions for this interval as $\mathbf{z}_i(t)$ and $\mathbf{z}_{i,0}$. One can then combine all the states into vectors with a dimension $n_z := n_x + n_{z,1} + \ldots + n_{z,n_s+1}$

$$\mathbf{z}(t) := \left[\mathbf{x}(t)^{\mathrm{T}}\ \mathbf{z}_1(t)^{\mathrm{T}}\ \cdots\ \mathbf{z}_{n_s+1}(t)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{2}$$

with corresponding initial conditions $\mathbf{z}_0$.

The arc type determines the dimension and meaning of the elements of $\mathbf{z}_i(t)$, $\mathbf{z}_{i,0}$ and their effect on the input $u(t)$ given by the control law $u(t) = \tilde{c}\big(\mathbf{z}(t)\big)$ and on the dynamics of $\mathbf{z}_i(t)$ given by $\dot{\mathbf{z}}_i(t) = \mathbf{q}_i\big(\mathbf{x}(t), \mathbf{z}_i(t)\big)$. For bang-bang arcs, $\mathbf{z}_i(t)$, $\mathbf{z}_{i,0}$ are of dimension 0 and $\tilde{c}\big(\mathbf{z}(t)\big) := \underline{u}$ or $\tilde{c}\big(\mathbf{z}(t)\big) := \overline{u}$. For active-state constraint arcs, $\mathbf{z}_i(t)$, $\mathbf{z}_{i,0}$ are not needed and $\tilde{c}\big(\mathbf{z}(t)\big)$ is such that $h_k^{(1)}\big(\mathbf{x}(t), \tilde{c}\big(\mathbf{z}(t)\big)\big) = 0$ for some $k = 1, \ldots, n_h$. For free/singular arcs, since we assume that the input is approximated by a linear function, $\mathbf{z}_i(t) = \begin{bmatrix} \tilde{u}_i(t) \\ \tilde{p}_i(t) \end{bmatrix}$ and $\mathbf{z}_{i,0} = \begin{bmatrix} u_i^0 \\ p_i \end{bmatrix}$ are of dimension 2, where u_i^0 and p_i are the initial value and derivative of the input and $\tilde{u}_i(t)$ is its value at time t, which implies that $\tilde{c}\big(\mathbf{z}(t)\big) := \tilde{u}_i(t)$ and $\mathbf{q}_i\big(\mathbf{x}(t), \mathbf{z}_i(t)\big) := \begin{bmatrix} \tilde{p}_i(t) \\ 0 \end{bmatrix}$. The set $\{i : i\text{th arc of } u(\cdot) \text{ is of type } 3\}$ is denoted as $\mathscr{S}$, which implies that $n_{z,1} + \ldots + n_{z,n_s+1} = 2|\mathscr{S}|$.

Then, upon eliminating input dependencies and rewriting Problem (1) in terms of the extended states $\mathbf{z}$, one obtains

$$\tilde{\boldsymbol{\chi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) := \begin{bmatrix} \tilde{\phi}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \\ \tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \end{bmatrix}, \tag{3}$$

with $\tilde{\phi}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) := \phi\big(\mathbf{x}(t_1), \ldots, \mathbf{x}(t_T), t_f\big)$ and a similar definition of $\tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$, and the dynamics

$$\tilde{\mathbf{f}}\big(\mathbf{z}(t)\big) := \left[\mathbf{f}\big(\mathbf{x}(t), \tilde{c}(\mathbf{z}(t))\big)^{\mathrm{T}}\ \mathbf{q}_1\big(\mathbf{x}(t), \mathbf{z}_1(t)\big)^{\mathrm{T}}\ \cdots\ \mathbf{q}_{n_s+1}\big(\mathbf{x}(t), \mathbf{z}_{n_s+1}(t)\big)^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{4}$$

Since the input parameters for the given arc sequence are $\boldsymbol{\tau} := \big(\bar{t}_1, \ldots, \bar{t}_{n_s}, t_f, \mathbf{z}_{1,0}, \ldots, \mathbf{z}_{n_s+1,0}\big)$ and it is assumed that any differential equations are solved for each $\boldsymbol{\tau}$, Problem (1) can be reformulated in terms of these new decision variables as

$$\min_{\boldsymbol{\tau}} \quad \hat{\phi}(\boldsymbol{\tau}) := \tilde{\phi}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big), \tag{5a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\psi}}(\boldsymbol{\tau}) := \tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \leq \mathbf{0}_{n_\psi}, \tag{5b}$$

$$\bar{t}_{i-1} \leq \bar{t}_i, \quad i = 1, \ldots, n_s + 1, \tag{5c}$$

$$\underline{u} \leq u_s^0 \leq \overline{u}, \quad \underline{u} \leq u_s^0 + p_s\,(\bar{t}_s - \bar{t}_{s-1}) \leq \overline{u}, \quad s \in \mathscr{S}, \tag{5d}$$

$$\dot{\mathbf{z}}(t) = \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big), \quad \mathbf{z}(t_0) = \mathbf{z}_0, \tag{5e}$$

which is convenient for numerical optimization since there are only $N := n_s + 1 + n_{z,1} + \ldots + n_{z,n_s+1}$ decision variables.

For each entry point $\hat{\eta}_j(\boldsymbol{\tau}) := \eta_j$, there exists $k = 1, \ldots, n_h$ such that $\tilde{h}_k\big(\mathbf{z}(\hat{\eta}_j(\boldsymbol{\tau})^-)\big) < 0$, $\tilde{h}_k\big(\mathbf{z}(\hat{\eta}_j(\boldsymbol{\tau}))\big) = 0$, which means that $\tilde{h}_k\big(\mathbf{z}(t)\big) := h_k\big(\mathbf{x}(t)\big) \leq 0$ becomes active at $t = \hat{\eta}_j(\boldsymbol{\tau})$.

B. Reformulation as polynomial optimization problems

We aim to reformulate the OCP for each arc sequence as a POP that is amenable to global optimization. This entails expressing $\hat{\boldsymbol{\chi}}(\boldsymbol{\tau}) := \begin{bmatrix}\hat{\phi}(\boldsymbol{\tau}) & \hat{\boldsymbol{\psi}}(\boldsymbol{\tau})^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}$ as a polynomial function [19], [20]. To this end, we compute each function $\hat{\chi}(\boldsymbol{\tau})$ and its first-order partial derivatives with respect to $\boldsymbol{\tau}$.

For this, it is essential to consider not only the extended states $\mathbf{z}(t)$ and the extended adjoint variables

$$\boldsymbol{\zeta}(t) := \begin{bmatrix}\boldsymbol{\lambda}(t)^{\mathrm{T}} & \boldsymbol{\zeta}_1(t)^{\mathrm{T}} & \cdots & \boldsymbol{\zeta}_{n_s+1}(t)^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}}, \tag{6}$$

but also the concept of modified Hamiltonian function $\tilde{H}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t)\big) = \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big)^{\mathrm{T}}\boldsymbol{\zeta}(t)$. As shown in (5), the extended states $\mathbf{z}(t)$ are described by the differential equations

$$\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t) = \frac{\partial \tilde{H}}{\partial \boldsymbol{\zeta}}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t)\big)^{\mathrm{T}} = \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big), \quad \mathbf{z}(t_0) = \mathbf{z}_0. \tag{7}$$

Likewise, the extended adjoint variables $\boldsymbol{\zeta}(t)$ are described by the differential equations

$$\begin{aligned}&\frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t) = -\frac{\partial \tilde{H}}{\partial \mathbf{z}}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t)\big)^{\mathrm{T}} = -\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(t)\big)^{\mathrm{T}}\boldsymbol{\zeta}(t), \quad \boldsymbol{\zeta}(t_T) = \mathbf{0}_{n_z},\\ &\boldsymbol{\zeta}(t_k^-) = \boldsymbol{\zeta}(t_k) + \frac{\partial \tilde{\chi}}{\partial \mathbf{z}(t_k)}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)^{\mathrm{T}}, \quad k = 1, \ldots, T,\end{aligned} \tag{8}$$

and in addition, for each entry point η such that $\tilde{h}_k\big(\mathbf{z}(t)\big) \leq 0$ becomes active at $t = \eta$ for some $k = 1, \ldots, n_h$, it holds that

$$\boldsymbol{\zeta}(\eta^-) = \boldsymbol{\zeta}(\eta) - \frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)^{\mathrm{T}} \frac{\big(\tilde{\mathbf{f}}\big(\mathbf{z}(\eta^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\eta)\big)\big)^{\mathrm{T}}\boldsymbol{\zeta}(\eta)}{\tilde{h}_k^{(1)}\big(\mathbf{z}(\eta^-)\big)}, \tag{9}$$

where the last expression is known for the case of state constraints of first order, that is, if $\mathbf{h}^{(1)}(\mathbf{x}, \mathbf{u}) := \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x}, \mathbf{u})$ depends explicitly on $\mathbf{u}$ as assumed, but is unknown for state constraints of higher order, to the best of our knowledge.

With these results, one can obtain the first-order partial derivatives of $\hat{\chi}(\boldsymbol{\tau})$ with respect to $\boldsymbol{\tau}$

$$\begin{aligned}\frac{\partial \hat{\chi}}{\partial \bar{t}_i}(\boldsymbol{\tau}) &= \tilde{H}\big(\mathbf{z}(\bar{t}_i^-), \boldsymbol{\zeta}(\bar{t}_i^-)\big) - \tilde{H}\big(\mathbf{z}(\bar{t}_i), \boldsymbol{\zeta}(\bar{t}_i)\big)\\ &= \big(\tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i)\big)\big)^{\mathrm{T}}\boldsymbol{\zeta}(\bar{t}_i), \quad i = 1, \ldots, n_s,\end{aligned} \tag{10}$$

$$\begin{aligned}\frac{\partial \hat{\chi}}{\partial t_f}(\boldsymbol{\tau}) &= \frac{\partial \tilde{\chi}}{\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) + \tilde{H}\big(\mathbf{z}(t_f^-), \boldsymbol{\zeta}(t_f^-)\big)\\ &= \frac{\partial \tilde{\chi}}{\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) + \tilde{\mathbf{f}}\big(\mathbf{z}(t_f^-)\big)^{\mathrm{T}}\boldsymbol{\zeta}(t_f^-),\end{aligned} \tag{11}$$

$$\frac{\partial \hat{\chi}}{\partial \mathbf{z}_{i,0}}(\boldsymbol{\tau}) = \boldsymbol{\zeta}_i(t_0)^{\mathrm{T}}, \quad i = 1, \ldots, n_s + 1. \tag{12}$$

Then, suppose that there exists $\bar{\boldsymbol{\tau}}$ such that, for all $\Delta\boldsymbol{\tau} \in \mathscr{R}$,

$$\hat{\chi}(\boldsymbol{\tau}) = \textstyle\sum_{\mathbf{k} \in \mathscr{K}_n^N} \big(\mathbf{c}_{\hat{\chi}}\big)_{\mathbf{k}} \Delta\boldsymbol{\tau}^{\mathbf{k}} + R_{\hat{\chi}}(\boldsymbol{\tau}), \tag{13}$$

where $\mathbf{c}_{\hat{\chi}}$ is the vector of polynomial coefficients of $\hat{\chi}(\boldsymbol{\tau})$, with $\big(\mathbf{c}_{\hat{\chi}}\big)_{\mathbf{k}} := \frac{1}{\mathbf{k}!}\frac{\partial^{\mathbf{k}}\hat{\chi}}{\partial \boldsymbol{\tau}^{\mathbf{k}}}(\bar{\boldsymbol{\tau}})$, $\mathbf{k}$ the vector of monomial powers in the set $\mathscr{K}_n^N := \big\{(k_1, \ldots, k_N) \in \mathbb{N}_0^N : 0 \leq k_1 + \ldots + k_N \leq n\big\}$ in the case of a polynomial of degree n, $\Delta\boldsymbol{\tau} := \boldsymbol{\tau} - \bar{\boldsymbol{\tau}}$ the deviation of $\boldsymbol{\tau}$ around $\bar{\boldsymbol{\tau}}$, $\mathbf{k}! := k_1! \ldots k_N!$, $\Delta\boldsymbol{\tau}^{\mathbf{k}} := (\tau_1 - \bar{\tau}_1)^{k_1} \ldots (\tau_N - \bar{\tau}_N)^{k_N}$, $\frac{\partial^{\mathbf{k}}}{\partial \boldsymbol{\tau}^{\mathbf{k}}} := \frac{\partial^{k_1+\ldots+k_N}}{\partial \tau_1^{k_1} \ldots \partial \tau_N^{k_N}}$, and $R_{\hat{\chi}}(\boldsymbol{\tau})$ is the orthogonal part with respect to the polynomial basis.

An efficient approach to approximating $\hat{\boldsymbol{\chi}}(\boldsymbol{\tau})$ as a polynomial function consists in (i) computing the partial derivatives of each one of the n_χ functions $\hat{\chi}(\boldsymbol{\tau})$ up to first order with respect to $\boldsymbol{\tau}$ and (ii) using multivariate Hermite interpolation to obtain a polynomial of degree $n > 1$ such as the one in (13) that fits the value $\hat{\chi}(\boldsymbol{\tau}_l)$ and the partial derivatives $\frac{\partial \hat{\chi}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}_l)$ at the sample points $\boldsymbol{\tau}_l$, for $l = 1, \ldots, m_\tau$ [21]. Note that this requires no more than computing the extended states $\mathbf{z}(t)$ and adjoint variables $\boldsymbol{\zeta}(t)$ for every $\hat{\chi}(\boldsymbol{\tau})$ that correspond to each point $\boldsymbol{\tau}_l$, which amounts to solving $n_\chi + 1$ systems of n_z differential equations for each $l = 1, \ldots, m_\tau$.

Remark 2. *One could also avoid computing the partial derivatives $\frac{\partial \hat{\chi}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}_l)$ and obtain a polynomial that fits only the value $\hat{\chi}(\boldsymbol{\tau}_l)$ at the sample points $\boldsymbol{\tau}_l$, for $l = 1, \ldots, m_\tau$. This would require no more than computing the extended states $\mathbf{z}(t)$ that correspond to each point $\boldsymbol{\tau}_l$, which would amount to solving one system of n_z differential equations for each $l = 1, \ldots, m_\tau$. Hence, this would entail solving m_τ systems of n_z differential equations to obtain m_τ values for interpolation. In contrast, the approach proposed in this paper requires solving only $(n_\chi + 1)m_\tau$ systems of n_z differential equations to obtain $(N+1)m_\tau$ values and partial derivatives for interpolation, owing to the computation of the extended adjoint variables $\boldsymbol{\zeta}(t)$. Since n_χ is expected to be smaller than N, the latter approach was chosen.*

Hence, one can compute the coefficient vector $\hat{\mathbf{c}}_{\hat{\chi}}$ that minimizes $\sum_{\boldsymbol{\kappa} \in \mathscr{K}_1^N} ||\mathbf{p}_{\hat{\chi},\boldsymbol{\kappa}} - \mathbf{A}_{\tau,\boldsymbol{\kappa}}\hat{\mathbf{c}}_{\hat{\chi}}||^2$, where $\big(\hat{\mathbf{c}}_{\hat{\chi}}\big)_{\mathbf{k}}$ is an approximation of $\big(\mathbf{c}_{\hat{\chi}}\big)_{\mathbf{k}}$, for all $\mathbf{k} \in \mathscr{K}_n^N$, and

$$\big(\mathbf{p}_{\hat{\chi},\boldsymbol{\kappa}}\big)_l = \frac{\partial^{\boldsymbol{\kappa}}\hat{\chi}}{\partial \boldsymbol{\tau}^{\boldsymbol{\kappa}}}(\boldsymbol{\tau}_l), \quad \boldsymbol{\kappa} \in \mathscr{K}_1^N, \quad l = 1, \ldots, m_\tau, \tag{14}$$

$$\begin{aligned}(\mathbf{A}_{\tau,\boldsymbol{\kappa}})_{l,\mathbf{k}} &= \begin{cases}\frac{\mathbf{k}!}{(\mathbf{k}-\boldsymbol{\kappa})!}\Delta\boldsymbol{\tau}_l^{\mathbf{k}-\boldsymbol{\kappa}}, & \mathbf{k} \geq \boldsymbol{\kappa}\\ 0, & \text{otherwise}\end{cases}, \quad \boldsymbol{\kappa} \in \mathscr{K}_1^N,\\ &\qquad l = 1, \ldots, m_\tau, \quad \mathbf{k} \in \mathscr{K}_n^N.\end{aligned} \tag{15}$$

The vector of polynomial coefficients $\hat{\mathbf{c}}_{\hat{\chi}}$ is of dimension $\binom{N+n}{N}$, while the number of value vectors $\mathbf{p}_{\hat{\chi},\boldsymbol{\kappa}}$ of dimension m_τ is $N+1$. This means that the number m_τ of sample points must be at least $\frac{(N+n)!}{n!(N+1)!}$, which is polynomial in N since n is typically bounded to avoid an overfitting polynomial. In addition, recall that N is typically small owing to the parsimonious nature of the input parameterization.

This yields the polynomial representation of $\hat{\chi}(\boldsymbol{\tau})$

$$p_{\hat{\chi}}(\boldsymbol{\tau}) = \textstyle\sum_{\mathbf{k} \in \mathscr{K}_n^N} \big(\hat{\mathbf{c}}_{\hat{\chi}}\big)_{\mathbf{k}} \Delta\boldsymbol{\tau}^{\mathbf{k}}. \tag{16}$$

Remark 3. *The polynomial functions $p_{\hat{\chi}}(\boldsymbol{\tau})$ are used to approximate mappings between the decision variables $\boldsymbol{\tau}$ and functions of the states $\mathbf{x}(t)$ at a finite number of times $t_1, \ldots, t_T$ that do not include the switching times $\bar{t}_1, \ldots, \bar{t}_{n_s}$ in $\boldsymbol{\tau}$. In other words, no switching time $\bar{t}_i$ is simultaneously related to the inputs and outputs of the mappings that are approximated by the polynomial functions $p_{\hat{\chi}}(\boldsymbol{\tau})$. Hence, the polynomial functions do not approximate the dependence of any functions of the states $\mathbf{x}(t)$ on the generic time $t < t_f$.*

Remark 4. *To avoid non-smoothness of $\hat{\chi}(\boldsymbol{\tau})$ due to the existence of different sequences of arcs of types 1, 2, and 3*

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

for the given sequence of arcs of types 1 and 3, the sample points $\boldsymbol{\tau}_l$ must be restricted to the ones that correspond to the optimal sequence of arcs of types 1, 2, and 3. The procedure is as follows. For the given sequence of arcs of types 1 and 3, sample points $\boldsymbol{\tau}_l$ are chosen. Different points $\boldsymbol{\tau}_l$ will lead to different sequences of arcs of types 1, 2, and 3, depending on which state constraints become active and result in arcs of type 2 and on the order of these arcs of type 2 with respect to the arcs of types 1 and 3. For example, suppose that, for a given sequence 1U-3-1U of arcs of types 1 and 3, it is known that the optimal sequence of arcs of types 1, 2, and 3 is 1U-3-1U-2, where the arc of type 2 is an arc with an active state constraint. In this example, some points $\boldsymbol{\tau}_l$ lead to the optimal sequence of arcs 1U-3-1U-2, while other points may lead to other sequences such as 1U-3-1U (without active state constraints) or 1U-2-3-1U (with a different order of the arcs of type 2 with respect to the arcs of types 1 and 3), among others. Then, only the points $\boldsymbol{\tau}_l$ that correspond to the optimal sequence of arcs of types 1, 2, and 3 (the sequence 1U-3-1U-2 in the previous example) are used for the computation of the polynomial approximation $p_{\hat{\chi}}(\boldsymbol{\tau})$ in (16)*. This is done to avoid the non-smoothness of $\hat{\chi}(\boldsymbol{\tau})$ that would occur if all the points $\boldsymbol{\tau}_l$ were used to construct the polynomial approximation regardless of their sequences of arcs of types 1, 2, and 3. Hence, we consider the problem only for the optimal sequence of arcs of types 1, 2, and 3. To this end, we use a support vector machine $p_{\hat{\eta}_j}(\boldsymbol{\tau})$ with polynomial kernel to decide whether the points $\boldsymbol{\tau}$ are such that each entry point $\hat{\eta}_j(\boldsymbol{\tau})$ in arcs of type 2 is placed with respect to $\bar{t}_1,\ldots,\bar{t}_{n_s}$ according to the optimal sequence of arcs of types 1, 2, and 3. To construct the support vector machine, the points $\boldsymbol{\tau}_l$ are classified in two groups: the points $\boldsymbol{\tau}_l$ that correspond to the optimal sequence of arcs of types 1, 2, and 3 (the sequence 1U-3-1U-2 in the previous example) are labeled with the value 1, and the remaining points are labeled with the value -1. This is done to prevent the POP from searching values of $\boldsymbol{\tau}$ for which the corresponding sequence of arcs of types 1, 2, and 3 is not the optimal one.*

Hence, when each function $\hat{\chi}(\boldsymbol{\tau})$ is expressed as a polynomial $p_{\hat{\chi}}(\boldsymbol{\tau})$ in the variables $\boldsymbol{\tau}$ for a given arc sequence, the OCP for that arc sequence is reformulated as the POP

$$\min_{\boldsymbol{\tau}} \quad p_{\hat{\phi}}(\boldsymbol{\tau}), \quad \text{s.t.} \quad \mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}) \geq \mathbf{0}_{n_\gamma},$$

$$\big(\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau})\big)_j := \begin{cases} h^{\psi}_j(\boldsymbol{\tau}),\ j{=}1,\ldots,n_\psi, \\ h^{\eta}_{j-n_\psi}(\boldsymbol{\tau}),\ j{=}n_\psi{+}1,\ldots,n_\psi{+}n_\eta, \\ h^{t}_{j-n_\psi-n_\eta}(\boldsymbol{\tau}),\ j{=}n_\psi{+}n_\eta{+}1,\ldots,\bar{n}_\psi, \\ \underline{h}^{b}_{j-\bar{n}_\psi}(\boldsymbol{\tau}),\ j{=}\bar{n}_\psi{+}1,\ldots,\bar{n}_\psi{+}|\mathscr{S}|, \\ \overline{h}^{b}_{j-\bar{n}_\psi-|\mathscr{S}|}(\boldsymbol{\tau}),\ j{=}\bar{n}_\psi{+}|\mathscr{S}|{+}1,\ldots,\bar{n}_\psi{+}2|\mathscr{S}|, \\ \underline{h}^{e}_{j-\bar{n}_\psi-2|\mathscr{S}|}(\boldsymbol{\tau}),\ j{=}\bar{n}_\psi{+}2|\mathscr{S}|{+}1,\ldots,\bar{n}_\psi{+}3|\mathscr{S}|, \\ \overline{h}^{e}_{j-\bar{n}_\psi-3|\mathscr{S}|}(\boldsymbol{\tau}),\ j{=}\bar{n}_\psi{+}3|\mathscr{S}|{+}1,\ldots,n_\gamma, \end{cases} \tag{17}$$

where $\bar{n}_\psi := n_\psi + n_\eta + n_s + 1$, $n_\gamma := \bar{n}_\psi + 4|\mathscr{S}|$,

$$h^{\psi}_j(\boldsymbol{\tau}) := -p_{\hat{\psi}_j}(\boldsymbol{\tau}), \quad j = 1,\ldots,n_\psi, \tag{18a}$$
$$h^{\eta}_j(\boldsymbol{\tau}) := p_{\hat{\eta}_j}(\boldsymbol{\tau}), \quad j = 1,\ldots,n_\eta, \tag{18b}$$
$$h^{t}_i(\boldsymbol{\tau}) := \bar{t}_i - \bar{t}_{i-1}, \quad i = 1,\ldots,n_s+1, \tag{18c}$$
$$\begin{aligned} &\underline{h}^{b}_i(\boldsymbol{\tau}) := u^0_s - \underline{u}, \quad \overline{h}^{b}_i(\boldsymbol{\tau}) := \overline{u} - u^0_s, \\ &\underline{h}^{e}_i(\boldsymbol{\tau}) := u^0_s + p_s(\bar{t}_s - \bar{t}_{s-1}) - \underline{u}, \\ &\overline{h}^{e}_i(\boldsymbol{\tau}) := \overline{u} - u^0_s - p_s(\bar{t}_s - \bar{t}_{s-1}), \quad s = \mathscr{S}_i,\ s \in \mathscr{S}, \end{aligned} \tag{18d}$$

and $\hat{\boldsymbol{\gamma}}(\boldsymbol{\tau})$ is defined as $\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau})$ using $\hat{\psi}_j(\boldsymbol{\tau})$ instead of $p_{\hat{\psi}_j}(\boldsymbol{\tau})$.

Remark 5. *The differential equations and initial conditions in* (5e) *are removed from* (17) *since the approximated functions $\hat{\phi}(\boldsymbol{\tau})$ and $\hat{\boldsymbol{\psi}}(\boldsymbol{\tau})$ are replaced by their polynomial approximations $p_{\hat{\phi}}(\boldsymbol{\tau})$ and $\mathbf{p}_{\hat{\boldsymbol{\psi}}}(\boldsymbol{\tau})$, which no longer depend on any differential equations or initial conditions.*

The complexity of the proposed approach is relatively insensitive to the number of states, which only changes the number of dynamic equations to be integrated. This means that the proposed approach deals efficiently with complex nonlinear dynamical systems since it scales well with a large number of states. In contrast, the approaches that use indirect methods related to the Hamilton-Jacobi-Bellman equation to solve OCPs to global optimality are limited by the number of states that they can handle, even when these methods involve the formulation of POPs (see [22] and Chapter 10 in [9]). However, as mentioned in Section II, the proposed approach would be sensitive to a larger number of inputs because it would affect the number of arc sequences. For this reason, OCPs with a single input $u(t)$ have been considered because the proposed methods are most efficient in this case.

Section V shows that the POP (17) is solved efficiently to global optimality via reformulation as a hierarchy of convex SDPs using the concept of sum-of-squares polynomials.

C. Error due to polynomial approximation

Since the solution to the POP (17) is not exactly the same as the solution to Problem (5) due to the fact that the polynomial functions $p_{\hat{\phi}}(\boldsymbol{\tau})$, $\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau})$ are approximations of the cost and constraint functions $\hat{\phi}(\boldsymbol{\tau})$, $\hat{\boldsymbol{\gamma}}(\boldsymbol{\tau})$, the question arises as to whether one can quantify the error in the optimal solution and the optimal value of the cost of the POP.

Suppose that the global solution to the POP (17) is $\boldsymbol{\tau}^*_p$, for which n_a constraints $-\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}) \leq \mathbf{0}_{n_\gamma}$ given by a selection matrix $\mathbf{S}_a$ are active with Lagrange multipliers $\boldsymbol{\nu}^*_p \geq \mathbf{0}_{n_a}$. The Karush-Kuhn-Tucker (KKT) conditions for $\boldsymbol{\tau}^*_p$ are

$$\frac{\partial p_{\hat{\phi}}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}^*_p)^{\mathrm{T}} - \frac{\partial \mathbf{p}_{\hat{\boldsymbol{\gamma}}}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}^*_p)^{\mathrm{T}}\mathbf{S}_a^{\mathrm{T}}\boldsymbol{\nu}^*_p = \mathbf{0}_N, \tag{19a}$$
$$-\mathbf{S}_a\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}^*_p) = \mathbf{0}_{n_a}. \tag{19b}$$

We aim to obtain explicit expressions for (i) the difference $\boldsymbol{\delta\tau}$ between $\boldsymbol{\tau}^*_p$ and $\boldsymbol{\tau}^*$, the KKT point of Problem (5) that corresponds to $\boldsymbol{\tau}^*_p$, and (ii) the difference $\delta\hat{\phi}$ between $p_{\hat{\phi}}(\boldsymbol{\tau}^*_p)$ and $\hat{\phi}(\boldsymbol{\tau}^*)$, the cost of Problem (5) at $\boldsymbol{\tau}^*$. It is impossible to obtain exact and explicit expressions for these differences since that would involve infinite series expansions around $\boldsymbol{\tau}^*_p$ and would imply explicit solutions to high-degree polynomials for $\boldsymbol{\delta\tau}$ and $\delta\hat{\phi}$ and the Abel-Ruffini theorem states that there is no closed-form algebraic expression for the solution to general polynomial equations of degree five or higher with arbitrary coefficients [23]. However, one can obtain explicit expressions for the approximations of $\boldsymbol{\delta\tau}$ and $\delta\hat{\phi}$, as well as exact and

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

implicit expressions that consider the variations of the second-order derivatives of the cost and Lagrangian functions and of the first-order derivatives of the constraint functions, which is done in the following theorem.

Theorem 1. *For a first-order approximation of the KKT conditions for Problem* (5) *and a second-order approximation of its cost, the explicit difference between the KKT points is*

$$\begin{aligned}\boldsymbol{\delta\tau} \simeq &\left(\mathbf{L}_p - \mathbf{L}_p\mathbf{Z}_p^{\mathrm{T}}\left(\mathbf{Z}_p\mathbf{L}_p\mathbf{Z}_p^{\mathrm{T}}\right)^{-1}\mathbf{Z}_p\mathbf{L}_p\right)\tfrac{\partial \varepsilon_{\hat{\mathscr{L}}}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}}\\ &-\mathbf{L}_p\mathbf{Z}_p^{\mathrm{T}}\left(\mathbf{Z}_p\mathbf{L}_p\mathbf{Z}_p^{\mathrm{T}}\right)^{-1}\mathbf{S}_a\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}_p^*),\end{aligned} \tag{20}$$

while the explicit difference between the costs is

$$\delta\hat{\phi} \simeq \varepsilon_{\hat{\phi}}(\boldsymbol{\tau}_p^*) + \tfrac{\partial\hat{\phi}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*)\boldsymbol{\delta\tau} - \tfrac{1}{2}\boldsymbol{\delta\tau}^{\mathrm{T}}\mathbf{H}_p\boldsymbol{\delta\tau}, \tag{21}$$

with the Lagrangian $\hat{\mathscr{L}}(\boldsymbol{\tau},\boldsymbol{\nu}) := \hat{\phi}(\boldsymbol{\tau}) - \boldsymbol{\nu}^{\mathrm{T}}\mathbf{S}_a\hat{\boldsymbol{\gamma}}(\boldsymbol{\tau})$*, the approximation errors* $\varepsilon_{\hat{\phi}}(\boldsymbol{\tau}) := p_{\hat{\phi}}(\boldsymbol{\tau}) - \hat{\phi}(\boldsymbol{\tau})$*,* $\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}) := \mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}) - \hat{\boldsymbol{\gamma}}(\boldsymbol{\tau})$*, and* $\varepsilon_{\hat{\mathscr{L}}}(\boldsymbol{\tau},\boldsymbol{\nu}) := \varepsilon_{\hat{\phi}}(\boldsymbol{\tau}) - \boldsymbol{\nu}^{\mathrm{T}}\mathbf{S}_a\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau})$ *for the cost, the constraints, and the Lagrangian, and the definitions* $\mathbf{L}_p := -\frac{\partial^2\hat{\mathscr{L}}}{\partial\boldsymbol{\tau}^2}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{-1}$*,* $\mathbf{Z}_p := \mathbf{S}_a\frac{\partial\hat{\boldsymbol{\gamma}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*)$*, and* $\mathbf{H}_p := \frac{\partial^2\hat{\phi}}{\partial\boldsymbol{\tau}^2}(\boldsymbol{\tau}_p^*)$*.*

Implicitly, the exact difference between the KKT points is

$$\begin{aligned}\boldsymbol{\delta\tau} = &\left(\mathbf{L} - \mathbf{L}\mathbf{Z}^{\mathrm{T}}\left(\mathbf{Z}\mathbf{L}\mathbf{Z}^{\mathrm{T}}\right)^{-1}\mathbf{Z}\mathbf{L}\right)\tfrac{\partial \varepsilon_{\hat{\mathscr{L}}}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}}\\ &-\mathbf{L}\mathbf{Z}^{\mathrm{T}}\left(\mathbf{Z}\mathbf{L}\mathbf{Z}^{\mathrm{T}}\right)^{-1}\mathbf{S}_a\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}_p^*),\end{aligned} \tag{22}$$

while the exact difference between the costs is

$$\delta\hat{\phi} = \varepsilon_{\hat{\phi}}(\boldsymbol{\tau}_p^*) + \tfrac{\partial\hat{\phi}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*)\boldsymbol{\delta\tau} - \tfrac{1}{2}\boldsymbol{\delta\tau}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta\tau}, \tag{23}$$

with the definitions

$$\mathbf{L} := -\left(\textstyle\int_0^1 \frac{\partial^2\hat{\mathscr{L}}}{\partial\boldsymbol{\tau}^2}(\boldsymbol{\tau}_p^* - \xi\boldsymbol{\delta\tau},\boldsymbol{\nu}_p^* - \xi\boldsymbol{\delta\nu})\mathrm{d}\xi\right)^{-1}, \tag{24}$$

$$\mathbf{Z} := \textstyle\int_0^1 \mathbf{S}_a\frac{\partial\hat{\boldsymbol{\gamma}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^* - \xi\boldsymbol{\delta\tau})\mathrm{d}\xi, \tag{25}$$

$$\mathbf{H} := \textstyle\int_0^1 2(1-\xi)\frac{\partial^2\hat{\phi}}{\partial\boldsymbol{\tau}^2}(\boldsymbol{\tau}_p^* - \xi\boldsymbol{\delta\tau})\mathrm{d}\xi. \tag{26}$$

Proof. The KKT conditions for the solution $\boldsymbol{\tau}_p^* - \boldsymbol{\delta\tau}$ to Problem (5) are given by

$$\frac{\partial\hat{\mathscr{L}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}} + \mathbf{L}^{-1}\boldsymbol{\delta\tau} + \mathbf{Z}^{\mathrm{T}}\boldsymbol{\delta\nu} = \mathbf{0}_N, \tag{27a}$$

$$-\mathbf{S}_a\hat{\boldsymbol{\gamma}}(\boldsymbol{\tau}_p^*) + \mathbf{Z}\boldsymbol{\delta\tau} = \mathbf{0}_{n_a}. \tag{27b}$$

Upon using the first-order approximation of these KKT conditions, they become

$$\frac{\partial\hat{\mathscr{L}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}} + \mathbf{L}_p^{-1}\boldsymbol{\delta\tau} + \mathbf{Z}_p^{\mathrm{T}}\boldsymbol{\delta\nu} \simeq \mathbf{0}_N, \tag{28a}$$

$$-\mathbf{S}_a\hat{\boldsymbol{\gamma}}(\boldsymbol{\tau}_p^*) + \mathbf{Z}_p\boldsymbol{\delta\tau} \simeq \mathbf{0}_{n_a}. \tag{28b}$$

Hence, from (19), it holds that

$$\mathbf{L}^{-1}\boldsymbol{\delta\tau} + \mathbf{Z}^{\mathrm{T}}\boldsymbol{\delta\nu} = \tfrac{\partial\varepsilon_{\hat{\mathscr{L}}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}}, \tag{29a}$$

$$\mathbf{Z}\boldsymbol{\delta\tau} = -\mathbf{S}_a\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}_p^*), \tag{29b}$$

which yields (22) by using the blockwise inversion formula, while the approximation

$$\mathbf{L}_p^{-1}\boldsymbol{\delta\tau} + \mathbf{Z}_p^{\mathrm{T}}\boldsymbol{\delta\nu} \simeq \tfrac{\partial\varepsilon_{\hat{\mathscr{L}}}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*,\boldsymbol{\nu}_p^*)^{\mathrm{T}}, \tag{30a}$$

$$\mathbf{Z}_p\boldsymbol{\delta\tau} \simeq -\mathbf{S}_a\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau}_p^*), \tag{30b}$$

yields the explicit expression for $\boldsymbol{\delta\tau}$ in (20) by using the blockwise inversion formula.

Then, since the cost of Problem (5) at $\boldsymbol{\tau}^*$ is given by

$$\hat{\phi}(\boldsymbol{\tau}_p^* - \boldsymbol{\delta\tau}) = \hat{\phi}(\boldsymbol{\tau}_p^*) - \tfrac{\partial\hat{\phi}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*)\boldsymbol{\delta\tau} + \tfrac{1}{2}\boldsymbol{\delta\tau}^{\mathrm{T}}\mathbf{H}\boldsymbol{\delta\tau}, \tag{31}$$

(23) holds and one can use the second-order approximation

$$\hat{\phi}(\boldsymbol{\tau}_p^* - \boldsymbol{\delta\tau}) \simeq \hat{\phi}(\boldsymbol{\tau}_p^*) - \tfrac{\partial\hat{\phi}}{\partial\boldsymbol{\tau}}(\boldsymbol{\tau}_p^*)\boldsymbol{\delta\tau} + \tfrac{1}{2}\boldsymbol{\delta\tau}^{\mathrm{T}}\mathbf{H}_p\boldsymbol{\delta\tau} \tag{32}$$

to obtain the explicit expression for $\delta\hat{\phi}$ in (21). □

Remark 6. *Theorem 1 only provides an explicit expression for the first-order approximation of the difference* $\boldsymbol{\delta\tau}$ *between* $\boldsymbol{\tau}_p^*$*, the global solution to the POP* (17)*, and* $\boldsymbol{\tau}^*$*, the KKT point of Problem* (5) *that corresponds to* $\boldsymbol{\tau}_p^*$*. This means that* $\boldsymbol{\tau}_p^* - \boldsymbol{\delta\tau}$ *is a good approximation for* $\boldsymbol{\tau}^*$ *with an explicit expression. One can obtain the exact KKT point* $\boldsymbol{\tau}^*$ *of Problem* (5) *that corresponds to* $\boldsymbol{\tau}_p^*$ *via local optimization of Problem* (5) *with initial guess* $\boldsymbol{\tau}_p^* - \boldsymbol{\delta\tau}$*.*

Moreover, one can assess the quality of the solution $\boldsymbol{\tau}^*$ obtained by solving the POP (17) to global optimality followed by local optimization of Problem (5). To this end, the following theorem shows that the difference between the cost $\hat{\phi}(\boldsymbol{\tau}^*)$ obtained from (17) and the globally optimal cost of (5) is bounded and depends on the polynomial approximation errors $\varepsilon_{\hat{\phi}}$, $\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}$, $\frac{\partial\varepsilon_{\hat{\mathscr{L}}}}{\partial\boldsymbol{\tau}}$ defined in Theorem 1.

Theorem 2. *If* $\delta\hat{\phi}_{max}^{\mathrm{KKT}}$ *is the maximum difference between the costs of any KKT point of the POP* (17) *and any corresponding KKT point of Problem* (5)*, then the difference between* $\hat{\phi}(\boldsymbol{\tau}^*)$ *and the cost of Problem* (5) *at its globally optimal solution is at most* $\delta\hat{\phi}_{max}^{\mathrm{KKT}} - \delta\hat{\phi}$ *and is bounded if the errors* $\varepsilon_{\hat{\phi}}$*,* $\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}$*,* $\frac{\partial\varepsilon_{\hat{\mathscr{L}}}}{\partial\boldsymbol{\tau}}$ *are bounded.*

Proof. The globally optimal solution to Problem (5) is a KKT point $\boldsymbol{\tau}^{\mathrm{KKT}}$ of Problem (5) that corresponds to some KKT point $\boldsymbol{\tau}_p^{\mathrm{KKT}}$ of the POP (17), and $\hat{\phi}(\boldsymbol{\tau}^*) - \hat{\phi}(\boldsymbol{\tau}^{\mathrm{KKT}}) - \delta\hat{\phi}_{max}^{\mathrm{KKT}} + \delta\hat{\phi} \leq p_{\hat{\phi}}(\boldsymbol{\tau}_p^*) - p_{\hat{\phi}}(\boldsymbol{\tau}_p^{\mathrm{KKT}}) \leq 0$ since $\boldsymbol{\tau}_p^*$ is the globally optimal solution to the POP (17). In addition, from Theorem 1, $\delta\hat{\phi}_{max}^{\mathrm{KKT}}$ and $\delta\hat{\phi}$ depend on $\varepsilon_{\hat{\phi}}$, $\boldsymbol{\varepsilon}_{\hat{\boldsymbol{\gamma}}}$, $\frac{\partial\varepsilon_{\hat{\mathscr{L}}}}{\partial\boldsymbol{\tau}}$. □

V. Global Solution to Each POP

This section first summarizes the concept of sum-of-squares (SOS) polynomials and its application to global optimization of POPs via semidefinite programming based on a more detailed discussion in [24], [25]. Then, the concept of SOS polynomials is used to obtain efficient global solutions to the OCP (1) reformulated as the POPs (17) via SDPs.

A. Sum-of-squares polynomials for global optimization

A polynomial $p(\boldsymbol{\tau})$ of degree $2d$ in the N variables $\boldsymbol{\tau}$ is an SOS polynomial if it can be written as a sum of squares of polynomials, and $p(\boldsymbol{\tau})$ is an SOS polynomial if and only if there exists a positive semidefinite matrix $\mathbf{Q}$ such that $p(\boldsymbol{\tau}) = \mathbf{v}_d(\boldsymbol{\tau})^{\mathrm{T}}\mathbf{Q}\mathbf{v}_d(\boldsymbol{\tau})$, where $\mathbf{v}_d(\boldsymbol{\tau})$ is the $s(N,d)$-dimensional vector of monomials of degree up to d in the N variables $\boldsymbol{\tau}$, with $s(N,d) := \binom{N+d}{N}$ [9]. Hence, constraining $p(\boldsymbol{\tau})$ to the set of SOS polynomials amounts to satisfying the linear matrix inequality (LMI) $\mathbf{Q} \succeq \mathbf{0}_{s(N,d)\times s(N,d)}$, which can be done via a convex SDP [26].

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

If $\varphi(\boldsymbol{\tau})$ is strictly positive on a compact basic semi-algebraic set $\mathbb{K}$ specified by some polynomials $g_j(\boldsymbol{\tau})$, that is, if $\varphi(\boldsymbol{\tau}) > 0$ $\forall \boldsymbol{\tau} \in \mathbb{K} = \{\boldsymbol{\tau} : g_j(\boldsymbol{\tau}) \geq 0, \forall j = 1,\ldots,n_c\}$ and $\mathbb{K}$ satisfies some technical assumptions, then $\varphi(\boldsymbol{\tau})$ can be represented as a combination of SOS polynomials. This important result is summarized in the following theorem [27].

Theorem 3. *Assume that there exists $q \in \{1,\ldots,n_c\}$ such that $\{\boldsymbol{\tau} : g_q(\boldsymbol{\tau}) \geq 0\}$ is compact. If $\varphi(\boldsymbol{\tau}) > 0$ $\forall \boldsymbol{\tau} \in \mathbb{K}$, then*

$$\varphi(\boldsymbol{\tau}) = p_0(\boldsymbol{\tau}) + \textstyle\sum_{j=1}^{n_c} g_j(\boldsymbol{\tau}) p_j(\boldsymbol{\tau}) \tag{33}$$

for some SOS polynomials $p_0(\boldsymbol{\tau})$ and $p_1(\boldsymbol{\tau}),\ldots,p_{n_c}(\boldsymbol{\tau})$.

Proof. See [27] for the proof. □

Remark 7. *If Theorem 3 applies and $\varphi(\boldsymbol{\tau}) > 0$ $\forall \boldsymbol{\tau} \in \mathbb{K}$, then there exists a positive integer d such that $\forall \boldsymbol{\alpha} \in \mathscr{X}_d$*

$$\varphi_{\boldsymbol{\alpha}} = \operatorname{tr}(\mathbf{R}_{0,\boldsymbol{\alpha}}\mathbf{Q}_0) + \textstyle\sum_{j=1}^{n_c} \sum_{\substack{\boldsymbol{\beta}\in\mathscr{X}_{d-v_j} \\ \boldsymbol{\alpha}-\boldsymbol{\beta}\in\mathscr{X}_{v_j}}} g_{j,\boldsymbol{\alpha}-\boldsymbol{\beta}} \operatorname{tr}\big(\mathbf{R}_{v_j,\boldsymbol{\beta}}\mathbf{Q}_j\big) \tag{34a}$$

and

$$\mathbf{Q}_0 \succeq \mathbf{0}_{s(N,d)\times s(N,d)}, \tag{34b}$$

$$\mathbf{Q}_j \succeq \mathbf{0}_{s(N,d-v_j)\times s(N,d-v_j)}, \quad j = 1,\ldots,n_c, \tag{34c}$$

where the matrices $\mathbf{R}_{v,\boldsymbol{\alpha}}$ are such that $\sum_{\boldsymbol{\alpha}\in\mathscr{X}_{d-v}} \mathbf{R}_{v,\boldsymbol{\alpha}}\boldsymbol{\tau}^{\boldsymbol{\alpha}} = \mathbf{v}_{d-v}(\boldsymbol{\tau})\mathbf{v}_{d-v}(\boldsymbol{\tau})^{\mathrm{T}}$, for $v = 0,\ldots,d$, the coefficients of $\varphi(\boldsymbol{\tau})$ of degree $2v_0$ or $2v_0-1$ and $g_j(\boldsymbol{\tau})$ of degree $2v_j$ or $2v_j-1$ are denoted as $\varphi_{\boldsymbol{\alpha}}$ and $g_{j,\boldsymbol{\alpha}}$, with $c_d := \max_{j=1,\ldots,n_c} v_j$, such that $\varphi(\boldsymbol{\tau}) = \sum_{\boldsymbol{\alpha}\in\mathscr{X}_d} \varphi_{\boldsymbol{\alpha}}\boldsymbol{\tau}^{\boldsymbol{\alpha}}$ and $g_j(\boldsymbol{\tau}) = \sum_{\boldsymbol{\alpha}\in\mathscr{X}_{v_j}} g_{j,\boldsymbol{\alpha}}\boldsymbol{\tau}^{\boldsymbol{\alpha}}$, for $j = 1,\ldots,n_c$, and the monomials $\boldsymbol{\tau}^{\boldsymbol{\alpha}} := \tau_1^{\alpha_1}\ldots\tau_N^{\alpha_N}$ of degree up to $2d$ in the variables $\boldsymbol{\tau}$ involve powers $\boldsymbol{\alpha} := (\alpha_1,\ldots,\alpha_N)$ in the set $\mathscr{X}_d := \{(\alpha_1,\ldots,\alpha_N) \in \mathbb{N}_0^N : 0 \leq \alpha_1 + \ldots + \alpha_N \leq 2d\}$, with the relaxation order $d \geq v := \max_{j=0,1,\ldots,n_c} v_j$ [8].

This result is very useful for the problem of computing J^*, an accurate approximation of the global minimum of $J(\boldsymbol{\tau})$ subject to the constraints $g_j(\boldsymbol{\tau}) \geq 0$, for $j = 1,\ldots,n_c$. Equivalently, one computes the maximum value ξ such that $\varphi(\boldsymbol{\tau}) = J(\boldsymbol{\tau}) - \xi$ is strictly positive $\forall \boldsymbol{\tau} \in \mathbb{K} = \{\boldsymbol{\tau} : g_j(\boldsymbol{\tau}) \geq 0, \forall j = 1,\ldots,n_c\}$. Such a problem can be formulated as the SDP $\min_{\xi,\mathbf{Q}_0,\mathbf{Q}_1,\ldots,\mathbf{Q}_{n_c}} -\xi$, s.t. (34) [24], [25].

Hence, if N and the maximum degree v of the polynomials are relatively small, the SDP can be solved efficiently since the relaxation order d that provides a representation in terms of SOS polynomials is usually not much larger than v. If this representation exists for some order d, a certificate can be obtained upon convergence of the SDP. The result about the representation for the order d is stated as follows [28].

Theorem 4. *Denote the optimal values of the dual variables for the constraints* (34a) *as $\mu_{\boldsymbol{\alpha}}^*$ $\forall \boldsymbol{\alpha} \in \mathscr{X}_d$ and of the dual variables for the LMI* (34b) *as $\mathbf{L}_0^*$. If $\exists G : G = \operatorname{rank}(\mathbf{L}_0^*) = \operatorname{rank}\Big(\sum_{\boldsymbol{\alpha}\in\mathscr{X}_{d-c_d}} \mathbf{R}_{c_d,\boldsymbol{\alpha}}\mu_{\boldsymbol{\alpha}}^*\Big)$, then $\varphi(\boldsymbol{\tau}) = J(\boldsymbol{\tau}) - J^*$ can be represented as in* (33) *with $p_0(\boldsymbol{\tau})$ of degree $2d$ and $p_j(\boldsymbol{\tau})$ of degree $2(d-v_j)$, for $j = 1,\ldots,n_c$. In addition, the global minimum $J^* = \xi^*$ and G global minimizers $\boldsymbol{\tau}_p^*$ can be computed using the fact that $\mathbf{v}_d(\boldsymbol{\tau}_p^*)$ lie both in the null space of $\mathbf{Q}_0^*$ and in the row space of $\mathbf{L}_0^*$.*

Proof. See [28] for the proof. □

B. Application to an OCP reformulated as POPs

This section shows how to apply the concept of SOS polynomials in Section V-A to obtain efficient global solutions to the POPs (17) that stem from the OCP (1) via SDPs.

In Section IV, it is shown that the OCP (1) can be reformulated as the POP (17) for each arc sequence. In terms of the notation in Section V-A, the POP (17) involves N decision variables, and each polynomial in the problem, both in the cost function and the constraints, is at most of degree n, which means that $v = \lceil n/2 \rceil$. Then, each relaxation order d in the hierarchy of semidefinite relaxations requires solving one LMI of size $\binom{N+d}{N}$, with $d \geq v$.

Hence, we introduce the following definitions:

$$\varphi(\boldsymbol{\tau}) := J(\boldsymbol{\tau}) - \xi, \quad J(\boldsymbol{\tau}) := p_{\hat{\phi}}(\boldsymbol{\tau}), \tag{35a}$$

$$g_j(\boldsymbol{\tau}) := \big(\mathbf{p}_{\hat{\boldsymbol{\gamma}}}(\boldsymbol{\tau})\big)_j, \quad j = 1,\ldots,n_\gamma. \tag{35b}$$

Then, the POP (17) is equivalent to the problem of computing the maximum ξ such that $\varphi(\boldsymbol{\tau})$ is strictly positive $\forall \boldsymbol{\tau} \in \mathbb{K} = \{\boldsymbol{\tau} : g_j(\boldsymbol{\tau}) \geq 0, \forall j = 1,\ldots,n_\gamma\}$.

However, we still need to add a new constraint to ensure that the condition of Theorem 3 is satisfied. To this end, we redefine $\mathbb{K} = \{\boldsymbol{\tau} : g_j(\boldsymbol{\tau}) \geq 0, \forall j = 1,\ldots,n_c\}$, with $n_c := n_\gamma + 1$, by adding the polynomial

$$g_{n_c}(\boldsymbol{\tau}) := r^{2v} - \textstyle\sum_{k=1}^{N} (\tau_k - \bar{\tau}_k)^{2v}, \tag{35c}$$

where r is a constant. If r is finite but sufficiently large to ensure that the minimizers $\boldsymbol{\tau}_p^*$ of the POP (17) are such that the $2v$-norm of $\Delta\boldsymbol{\tau}_p^*$ is bounded by r, then adding the new constraint does not change the minimizers. Moreover, the polynomial (35c) is of degree $2v$ since the polynomials with compact superlevel sets are at least of degree 2 and the polynomials that specify the other constraints are at most of degree $2v$ or $2v-1$. Now, the condition in Theorem 3 is satisfied since the superlevel set $\{\boldsymbol{\tau} : g_q(\boldsymbol{\tau}) \geq 0\}$ is compact for $q = n_c$. The boundedness of the $2v$-norm of $\Delta\boldsymbol{\tau}_p^*$ implies that $\bar{t}_1,\ldots,\bar{t}_{n_s}$, t_f, $\mathbf{z}_{1,0},\ldots,\mathbf{z}_{n_s+1,0}$ are bounded. Since the polynomials $p_{\hat{\phi}}(\boldsymbol{\tau})$, $p_{\hat{\psi}_j}(\boldsymbol{\tau})$, $p_{\hat{\eta}_j}(\boldsymbol{\tau})$ are obtained from points $\boldsymbol{\tau}_l$ such that $\Delta\boldsymbol{\tau}_l \in \mathscr{R}$, it can be assumed that $\Delta\boldsymbol{\tau}_p^*$ is bounded.

Since the condition in Theorem 3 is satisfied, the problem of computing the global minimum of $J(\boldsymbol{\tau})$ subject to $g_j(\boldsymbol{\tau}) \geq 0$, for $j = 1,\ldots,n_c$, can be formulated as the SDP described in Section V-A for some relaxation order $d \geq v = \lceil n/2 \rceil$ [24], [25]. A certificate of the representation in terms of SOS polynomials for the order d is obtained upon convergence of the SDP as shown in Theorem 4, which is a certificate of global optimality of the solution $\boldsymbol{\tau}_p^*$ and the cost $\xi^* = J^*$.

Suppose that $c_d \geq 1$ and a global optimum is computed and certified for the relaxation order $d = 5$ as in the examples of Section VI. This implies that the SDP in Section V-A has been solved for $d = 5$, which is an SDP with $\binom{N+2d}{N} = \frac{(N+10)\ldots(N+1)}{3628800}$ equality constraints, one LMI of size $\binom{N+d}{N} = \frac{(N+5)(N+4)(N+3)(N+2)(N+1)}{120}$, and $n_c = n_\gamma + 1$ LMIs of size $\binom{N+d-v_j}{N} \leq \frac{(N+4)(N+3)(N+2)(N+1)}{24}$. Since the complexity of SDPs is polynomial in their input size, that is, the number of constraints and the size of the LMIs, it means that a global solution $\boldsymbol{\tau}_p^*$ is computed and certified in polynomial time.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

The solution $\boldsymbol{\tau}_p^*$ to each POP (17) allows us to compute the global solution $\boldsymbol{\tau}^*$ for a given arc sequence. Hence, when the globally optimal cost is known for each arc sequence, one can check which sequence is the best one. As mentioned in Section III, the number of plausible arc sequences in input-affine OCPs is less than $2^{\bar{n}_a}$. Regarding the number of decision variables for each arc sequence, it is $N = n_s + 1 + 2|\mathscr{S}| \leq 2\bar{n}_a$. This means that, even for a relatively large upper bound $\bar{n}_a = 5$, less than $2^{\bar{n}_a} = 32$ arc sequences need to be considered, and the problem for each sequence can be solved in parallel and involves only $N \leq 2\bar{n}_a = 10$ decision variables. Despite the limitations in the size of the SDPs stemming from reformulation of POPs that the current SDP solvers can handle, POPs with about 10 decision variables have been characterized as problems that can be solved efficiently [29]. In contrast, if the arc sequences were not enumerated, we would need to use an input parameterization such as the piecewise-constant parameterization that is typically used for numerical solution of OCPs, which would require a large number of decision variables. In summary, since the problem for each sequence can be solved in parallel and involves only a reduced number of decision variables, the proposed approach provides a logical framework for efficient parallel computation of the global solution to single-input OCPs.

Remark 8. *It might be possible to improve the efficiency of the proposed scheme based on exhaustive enumeration of plausible arc sequences by (i) formulating POPs for computing global upper and lower bounds on the optimal solutions for comprehensive sets of arc sequences and (ii) enumerating only the parts of the search tree that cannot be eliminated by bounding steps. However, the details of a method that would allow efficient implementation of this idea are currently unclear. Hence, the implementation of this idea represents a possible direction for future work that is considered out of the scope of this paper.*

VI. Simulation Examples

This section illustrates the proposed method to efficiently compute the global solution to OCPs via two simulation examples: production maximization in a chemical reaction system and maximization of the peak altitude of a rocket.

A. Production maximization in a chemical reaction system

This simulation example corresponds to a problem of production maximization in an acetoacetylation reaction system with the species A, B, C, D, E [11]. This OCP is formulated with the states $\mathbf{x}(t) := \begin{bmatrix}\mathbf{x}_r(t)^{\mathrm{T}} & x_{in}(t)\end{bmatrix}^{\mathrm{T}}$ that represent the extents of reaction and inlet as:

$$\max_{u_{in}(\cdot),t_f} \ \mathscr{J}\big(u_{in}(\cdot),t_f\big) = n_{\mathrm{C}}(t_f), \tag{36a}$$

$$\text{s.t. } \mathscr{T}\big(u_{in}(\cdot),t_f\big) = \begin{bmatrix} n_{\mathrm{B}}(t_f) - c_{\mathrm{B},max}V(t_f) \\ n_{\mathrm{D}}(t_f) - c_{\mathrm{D},max}V(t_f) \\ t_f - t_{f,max} \end{bmatrix} \leq \mathbf{0}_3, \tag{36b}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t),u_{in}(t)\big) = \begin{bmatrix} \mathbf{r}_v(t) \\ \frac{u_{in}(t)}{1000} \end{bmatrix}, \quad \mathbf{x}(t_0) = \mathbf{0}_{R+1}, \tag{36c}$$

$$\begin{bmatrix} u_{in}(t) - \overline{u}_{in} & \underline{u}_{in} - u_{in}(t) \end{bmatrix}^{\mathrm{T}} \leq \mathbf{0}_2, \tag{36d}$$

where $\mathbf{x}_r(t) := \begin{bmatrix} x_{r,1}(t) & x_{r,2}(t) & x_{r,3}(t) \end{bmatrix}^{\mathrm{T}}$, $\underline{u}_{in} = 0$, $\overline{u}_{in} = 2$ mL min^{-1}, $t_{f,max} = 250$ min, $c_{\mathrm{B},max} = 0.025$ mol L^{-1}, and $c_{\mathrm{D},max} = 0.15$ mol L^{-1}, the $R = 3$ reaction rates $\mathbf{r}_v(t) := \begin{bmatrix} r_{v,1}(t) & r_{v,2}(t) & r_{v,3}(t) \end{bmatrix}^{\mathrm{T}}$ are given by $r_{v,1}(t) = k_1 \frac{n_{\mathrm{A}}(t) n_{\mathrm{B}}(t)}{V(t)}$, $r_{v,2}(t) = k_2 \frac{n_{\mathrm{B}}^2(t)}{V(t)}$, $r_{v,3}(t) = k_3 n_{\mathrm{B}}(t)$, with the rate constants $k_1 = 0.053$ L mol^{-1} min^{-1}, $k_2 = 0.128$ L mol^{-1} min^{-1}, and $k_3 = 0.028$ min^{-1}, the volume is given by $V(t) = V_0 + x_{in}(t)$, with $V_0 = 1$ L, and the numbers of moles $\mathbf{n}(t) := \begin{bmatrix} n_{\mathrm{A}}(t) & n_{\mathrm{B}}(t) & n_{\mathrm{C}}(t) & n_{\mathrm{D}}(t) & n_{\mathrm{E}}(t) \end{bmatrix}^{\mathrm{T}}$ are given by $\mathbf{n}(t) = \mathbf{N}^{\mathrm{T}}\mathbf{x}_r(t) + \mathbf{c}_{in}x_{in}(t) + \mathbf{n}_0$, with $\mathbf{n}_1 = \begin{bmatrix} -1 & -1 & 1 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{n}_2 = \begin{bmatrix} 0 & -2 & 0 & 1 & 0 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{n}_3 = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{N} = \begin{bmatrix} \mathbf{n}_1 & \mathbf{n}_2 & \mathbf{n}_3 \end{bmatrix}^{\mathrm{T}}$, $\mathbf{c}_{in} = \begin{bmatrix} 0 & 5 & 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ mol L^{-1}, as well as $\mathbf{n}_0 = \begin{bmatrix} 0.72 & 0.05 & 0.08 & 0.01 & 0 \end{bmatrix}^{\mathrm{T}}$ mol.

It was shown by [11] that, when linear functions are used to approximate free/singular arcs, a locally optimal solution consists of 3 arcs: in the first arc, $u_{in}^*(t) = \overline{u}_{in}$; the second arc is free/singular with $\underline{u}_{in} < u_{in}^*(t) < \overline{u}_{in}$, for which an approximation by a linear function is used; and in the third arc, $u_{in}^*(t) = \underline{u}_{in}$. This results in an input trajectory described by the 5 decision variables $\bar{t}_1$, $\bar{t}_2$, t_f, u_2^0, p_2. The optimal switching times are $\bar{t}_1^* = 5.96$ min, $\bar{t}_2^* = 230.26$ min, and the optimal final time is $t_f^* = 250$ min. The optimal initial conditions for the second arc are the initial value and the constant derivative of the linear function that describes $u_{in}^*(t)$ in this arc: $u_2^{0*} = 1.262$ mL min^{-1}, $p_2^* = -1.13 \times 10^{-3}$ mL min^{-2}. The optimal cost is $n_{\mathrm{C}}^*(t_f^*) = 0.51373$ mol, and all the terminal constraints are active. The local optimality is indicated by the fact that the gradients (10), (11), (12) are equal to zero and the solution satisfies the necessary conditions given by Pontryagin's maximum principle [5].

The proposed approach for obtaining global solutions to OCPs is applied by investigating all the 6 plausible arc sequences with a number of arcs no larger than $\bar{n}_a = 3$. Table I reports the execution time of the procedure on an Intel Core i7 3.4 GHz processor, the optimal cost $\hat{\phi}(\boldsymbol{\tau}^*)$, and the optimal values of the decision variables for these plausible arc sequences. The execution time includes the evaluation of $m_\tau = 1000$ sample points to obtain the polynomial representations $p_{\hat{\phi}}(\boldsymbol{\tau})$, $p_{\hat{\psi}_j}(\boldsymbol{\tau})$ of degree $n = 6$ and the local optimization needed to compute $\boldsymbol{\tau}^*$ for each arc sequence. For all the arc sequences, it is possible to extract the unique solution $\boldsymbol{\tau}_p^*$ to the POP (17) from the solution to the SDP for the relaxation order $d = 5$ and certify the global optimality of $\boldsymbol{\tau}_p^*$ with the cost J^*. The duration of the formulation of the SDP and the extraction and certification of the global solution is much smaller than the execution time of the SDP solver MOSEK 9.2. One can observe that the execution time is below 40 s for all arc sequences and the sequence with the best optimal cost is 1U-3-1L, that is, the sequence of the locally optimal solution. In addition, the globally optimal values $\bar{t}_1^*$, $\bar{t}_2^*$, t_f^*, u_2^{0*}, p_2^* of the decision variables for that arc sequence also correspond to the optimal values given by the locally optimal solution. For this problem, the solution of which has been accurately described by 5 decision variables in this paper by enumerating $6 < 2^3$ arc sequences, Figure 2 in [11] had shown that 25 decision variables are necessary to describe the optimal solution with

TABLE I
EXECUTION TIME, OPTIMAL COST $\hat{\phi}(\boldsymbol{\tau}^*)$, AND OPTIMAL VALUES $\bar{t}_1^*, \bar{t}_2^*, t_f^*, u_i^{0*}, p_i^*$ OF THE DECISION VARIABLES FOR THE GLOBAL SOLUTION TO THE OCP (36) FOR DIFFERENT PLAUSIBLE ARC SEQUENCES.

Arc sequence	Execution time (s)	$\hat{\phi}(\boldsymbol{\tau}^*)$ (mol)	$\bar{t}_1^*$ (min)	$\bar{t}_2^*$ (min)	t_f^* (min)	u_i^{0*} (mL min^{-1})	p_i^* (mL min^{-2})
3-1L-1U	35.0	-0.51350	230.52	250.00	250.00	1.328	-1.49 $\times 10^{-3}$ ($i=1$)
3-1U-1L	34.0	-0.51350	228.31	229.46	250.00	1.329	-1.50 $\times 10^{-3}$ ($i=1$)
1L-3-1L	25.1	-0.51350	0.00	230.52	250.00	1.328	-1.49 $\times 10^{-3}$ ($i=2$)
1L-3-1U	37.6	-0.50476	0.00	250.00	250.00	1.909	-6.92 $\times 10^{-3}$ ($i=2$)
1U-3-1L	24.9	-0.51373	5.96	230.26	250.00	1.262	-1.13 $\times 10^{-3}$ ($i=2$)
1U-3-1U	34.4	-0.50476	1.62	250.00	250.00	1.896	-6.92 $\times 10^{-3}$ ($i=2$)

the same accuracy in terms of cost when a piecewise-constant input parameterization is used. Hence, if it is assumed that the worst-case complexity of global optimization is given by $O(2^N)$, where N is the number of continuous decision variables, branch-and-bound with input parameters as decision variables may entail a complexity of $O(2^{25})$ for this example, while exhaustive enumeration of arc sequences only requires a complexity of $O(2^5)$ for less than 2^3 arc sequences, where these arc sequences can be evaluated in parallel.

In summary, one can show that the locally optimal solution to the OCP (36) shown in Fig. 1 is also the globally optimal solution with no more than $\bar{n}_a = 3$ arcs, and this only requires solving 6 problems in parallel in less than 40 s. We repeated the same procedure with a value $\bar{n}_a = 4$. Since we obtained the same three-arc solution as with $\bar{n}_a = 3$, it is reasonable to conclude that the three-arc solution is indeed the globally optimal solution to the OCP (36). As predicted, when the optimal solution for some sequences of $\bar{n}_a = 4$ arcs corresponds to the three-arc solution, some switching points coincide. We did not encounter any issues with the polynomial approximation or non-smoothness in the feasible region of the POP (17) due to the coincidence of switching points since all the functions in the OCP formulation (36) are smooth. Recall that, if we had only used local optimization to compute a local solution to the OCP (36), we could have obtained a local solution worse than $\boldsymbol{\tau}^*$ and it would not have been possible to provide any guarantee that the local solution is in any way close to the globally optimal solution.

For this problem in particular, the analytical expression for the time derivative of the optimal input in free/singular arcs is known (see (39) in [17]). Hence, it is possible to compare the costs that are achieved by considering (i) the analytical expression for the input in the free/singular arc and (ii) the approximation of the input in the free/singular arc by a linear function. The optimal input trajectories for both (i) and (ii) are presented in Fig. 1, which shows that, even though the input trajectories are similar, they are not coincident. Despite this difference between the input trajectories for (i) and (ii), the optimal costs are very similar: the cost is 0.51374 mol for (i), which is better than the cost for (ii) by a margin of only 6×10^{-6} mol. This shows that the difference in cost is negligible, which confirms that the approximation by a linear function is sufficiently accurate. At the same time, the proposed approach allows determining a global solution for the approximated problem, which is an accurate approximation of the original problem as shown. Note that, although the use of analytical expressions to represent the free/singular is possible for this illustrative problem, in general it is not possible to use such analytical expressions for every problem since they are typically unknown. This justifies the need for numerical approaches that are generally applicable to any single-input OCP.

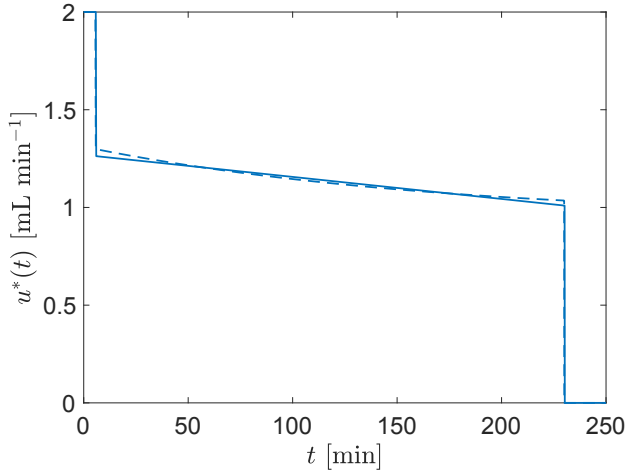


Fig. 1. Globally optimal input trajectory for the OCP (36) with the approximation of the input in the free/singular arc by a linear function (solid line) and the analytical expression for the input in the free/singular arc (dashed line).

B. Maximization of the peak altitude of a rocket

This simulation example corresponds to a problem of maximization of the peak altitude of a rocket, known as the Goddard problem, in a dimensionless formulation [30]. This OCP is formulated with the states $\mathbf{x}(t) := \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix}^{\mathrm{T}}$ that represent the altitude, velocity, and mass as:

$$\max_{u(\cdot),t_f} \; \mathscr{J}\left(u(\cdot),t_f\right) = x_1(t_f), \tag{37a}$$

$$\text{s.t. } \mathscr{T}\left(u(\cdot),t_f\right) = 0.6 - x_3(t_f) \leq 0, \tag{37b}$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t),u(t)\big) = \begin{bmatrix} x_2(t) \\ \frac{u(t)-d(t)}{x_3(t)} - \frac{1}{x_1(t)^2} \\ -\frac{u(t)}{c} \end{bmatrix}, \; \mathbf{x}(t_0) = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \tag{37c}$$

$$\begin{bmatrix} u(t) - \overline{u} & \underline{u} - u(t) \end{bmatrix}^{\mathrm{T}} \leq \mathbf{0}_2, \tag{37d}$$

where $\underline{u} = 0$, $\overline{u} = 3.5$, and $d(t) := a x_2(t)^2 \exp\big(b - b x_1(t)\big)$, with $a = 310$, $b = 500$, $c = 0.5$.

It can be shown that, when linear functions are used to approximate free/singular arcs, a locally optimal solution consists of 3 arcs: in the first arc, $u^*(t) = \overline{u}$; the second arc is free/singular with $\underline{u} < u^*(t) < \overline{u}$, for which an approximation by a linear function is used; and in the third arc, $u^*(t) = \underline{u}$.

TABLE II
EXECUTION TIME, OPTIMAL COST $\hat{\phi}(\boldsymbol{\tau}^*)$, AND OPTIMAL VALUES $\bar{t}_1^*$, $\bar{t}_2^*$, t_f^*, u_i^{0*}, p_i^* OF THE DECISION VARIABLES FOR THE GLOBAL SOLUTION TO THE OCP (37) FOR DIFFERENT PLAUSIBLE ARC SEQUENCES.

Arc sequence	Execution time (s)	$\hat{\phi}(\boldsymbol{\tau}^*)$ (-)	$\bar{t}_1^*$ (-)	$\bar{t}_2^*$ (-)	t_f^* (-)	u_i^{0*} (-)	p_i^* (-)
3-1L-1U	36.3	-1.012752	0.074949	0.197437	0.197437	3.5000	-2.2189 $\times 10$ ($i=1$)
3-1U-1L	25.5	-1.012760	0.064014	0.071770	0.198229	3.5000	-2.4986 $\times 10$ ($i=1$)
1L-3-1L	16.7	-1.012752	0.000000	0.074949	0.197437	3.5000	-2.2189 $\times 10$ ($i=2$)
1L-3-1U	31.7	-1.009223	0.000000	0.157158	0.157158	2.5452	-1.6195 $\times 10$ ($i=2$)
1U-3-1L	24.1	-1.012837	0.023998	0.073133	0.198847	1.9766	1.5644 $\times 10$ ($i=2$)
1U-3-1U	27.2	-1.012505	0.057143	0.187327	0.187327	0.0000	0.0000 $\times 10$ ($i=2$)

This results in an input trajectory described by the 5 decision variables $\bar{t}_1$, $\bar{t}_2$, t_f, u_2^0, p_2. The optimal switching times are $\bar{t}_1^* = 0.023998$, $\bar{t}_2^* = 0.073133$, and the optimal final time is $t_f^* = 0.198847$. The optimal initial conditions for the second arc are the initial value and the constant derivative of the linear function that describes $u^*(t)$ in this arc: $u_2^{0*} = 1.9766$, $p_2^* = 1.5644 \times 10$. The optimal cost is $x_1^*(t_f^*) = 1.012837$, and the terminal constraint is active. The local optimality is indicated by the fact that the gradients (10), (11), (12) are equal to zero and the solution satisfies the necessary conditions given by Pontryagin's maximum principle [5].

The proposed approach for obtaining global solutions to OCPs is applied by investigating all the 6 plausible arc sequences with a number of arcs no larger than $\bar{n}_a = 3$. Table II reports the execution time of the procedure on an Intel Core i7 3.4 GHz processor, the optimal cost $\hat{\phi}(\boldsymbol{\tau}^*)$, and the optimal values of the decision variables for these plausible arc sequences. The execution time includes the evaluation of $m_\tau = 1000$ sample points to obtain the polynomial representations $p_{\hat{\phi}}(\boldsymbol{\tau})$, $p_{\dot{\psi}_j}(\boldsymbol{\tau})$ of degree $n = 6$ and the local optimization needed to compute $\boldsymbol{\tau}^*$ for each arc sequence. For all the arc sequences, it is possible to extract the unique solution $\boldsymbol{\tau}_p^*$ to the POP (17) from the solution to the SDP for the relaxation order $d = 5$ and certify the global optimality of $\boldsymbol{\tau}_p^*$ with the cost J^*. The duration of the formulation of the SDP and the extraction and certification of the global solution is much smaller than the execution time of the SDP solver MOSEK 9.2. One can observe that the execution time is below 40 s for all arc sequences and the sequence with the best optimal cost is 1U-3-1L, that is, the sequence of the locally optimal solution. In addition, the globally optimal values $\bar{t}_1^*$, $\bar{t}_2^*$, t_f^*, u_2^{0*}, p_2^* of the decision variables for that arc sequence also correspond to the optimal values given by the locally optimal solution.

In summary, one can show that the locally optimal solution to the OCP (37) shown in Fig. 2 is also the globally optimal solution with no more than $\bar{n}_a = 3$ arcs, and this only requires solving 6 problems in parallel in less than 40 s. We repeated the same procedure with a value $\bar{n}_a = 4$. Since we obtained the same three-arc solution as with $\bar{n}_a = 3$, it is reasonable to conclude that the three-arc solution is indeed the globally optimal solution to the OCP (37). As predicted, when the optimal solution for some sequences of $\bar{n}_a = 4$ arcs corresponds to the three-arc solution, some switching points coincide. We did not encounter any issues with the polynomial approximation or non-smoothness in the feasible region of the POP (17) due to the coincidence of switching points since all the functions in the OCP formulation (37) are smooth. Recall that, if we had

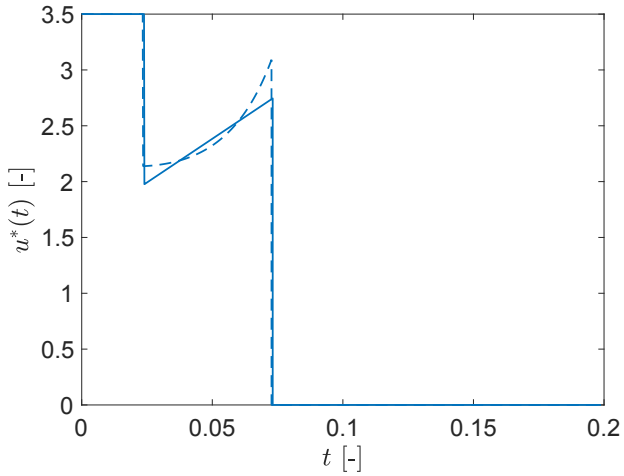


Fig. 2. Globally optimal input trajectory for the OCP (37) with the approximation of the input in the free/singular arc by a linear function (solid line) and the analytical expression for the input in the free/singular arc (dashed line).

only used local optimization to compute a local solution to the OCP (37), we could have obtained a local solution worse than $\boldsymbol{\tau}^*$ and it would not have been possible to provide any guarantee that the local solution is in any way close to the globally optimal solution.

For this problem in particular, the analytical expression for the optimal input in free/singular arcs is known [30]:

$$u^* = a x_2^2 \exp(b - b x_1) + \frac{c(1+x_2/c)x_3(b x_2^2 + 2/x_1^2)}{2c + 4x_2 + x_2^2/c} - \frac{2c x_3^2}{a \exp(b - b x_1) x_1^3 (2c + 4x_2 + x_2^2/c)}. \tag{38}$$

Hence, it is possible to compare the costs that are achieved by considering (i) the analytical expression for the input in the free/singular arc and (ii) the approximation of the input in the free/singular arc by a linear function. The optimal input trajectories for both (i) and (ii) are presented in Fig. 2, which shows that, even though the input trajectories are similar, they are not coincident. Despite this difference between the input trajectories for (i) and (ii), the optimal costs are very similar: the cost is 1.012837 for (i), which is better than the cost for (ii) by a margin of only 2×10^{-7}. This shows that the difference in cost is negligible, which confirms that the approximation by a linear function is sufficiently accurate. At the same time, the proposed approach allows determining a global solution for the approximated problem, which is an accurate approximation of the original problem as shown. Note that, although the use of analytical expressions to represent the free/singular is possible for this illustrative problem, in general it is not possible to use such analytical expressions for every problem since they

are typically unknown. This justifies the need for numerical approaches that are generally applicable to any single-input OCP.

VII. Conclusions

This paper presented an efficient global solution method for single-input OCPs that relies on the enumeration of plausible arc sequences. It was shown that the cost and constraints for a given arc sequence can be approximated as explicit polynomial functions of the decision variables, which in turn allows for reformulation of OCPs as a set of polynomial optimization problems. The latter problems can then be reformulated as a hierarchy of convex semidefinite programs and efficiently solved to global optimality via the concept of sum-of-squares polynomials. The paper showed that the difference between the cost obtained by the proposed approach and the globally optimal cost of the original problem is bounded and depends on the polynomial approximation errors. The proposed approach can deal efficiently with nonlinear dynamical systems with a large number of states or complex dynamics.

In future work, it would be useful to extend the proposed method to OCPs with multiple inputs and stochastic disturbances. In addition, enumeration of all the plausible arc sequences can be a limitation of the approach proposed in this paper. Although this limitation does not make the proposed approach less efficient than existing alternative methods based on branch-and-bound, it would be beneficial to address this limitation in future work.

References

[1] A. E. Bryson and Y. C. Ho, *Applied Optimal Control: Optimization, Estimation, and Control*. Washington DC: Hemisphere, 1975.

[2] K. L. Teo, C. J. Goh, and K. H. Wong, *A Unified Computational Approach to Optimal Control Problems*. New York: Longman Scientific & Technical, 1991.

[3] L. T. Biegler, A. M. Cervantes, and A. Wächter, "Advances in simultaneous strategies for dynamic process optimization," *Chem. Eng. Sci.*, vol. 57, no. 4, pp. 575–593, 2002.

[4] B. Houska and B. Chachuat, "Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control," *J. Optim. Theory Appl.*, vol. 162, no. 1, pp. 208–248, 2014.

[5] R. F. Hartl, S. P. Sethi, and R. G. Vickson, "A survey of the maximum principles for optimal control problems with state constraints," *SIAM Rev.*, vol. 37, no. 2, pp. 181–218, 1995.

[6] R. Luus, *Iterative Dynamic Programming*. Boca Raton, FL: Chapman & Hall, 2000.

[7] B. Chachuat, A. B. Singer, and P. I. Barton, "Global methods for dynamic optimization and mixed-integer dynamic optimization," *Ind. Eng. Chem. Res.*, vol. 45, no. 25, pp. 8373–8392, 2006.

[8] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM J. Optim.*, vol. 11, pp. 796–817, 2001.

[9] ——, *Moments, Positive Polynomials and Their Applications*. London, UK: Imperial College Press, 2010.

[10] B. Houska, H. J. Ferreau, and M. Diehl, "ACADO toolkit – An open-source framework for automatic control and dynamic optimization," *Optim. Control Appl. Methods*, vol. 32, no. 3, pp. 298–312, 2011.

[11] D. Rodrigues and D. Bonvin, "On reducing the number of decision variables for dynamic optimization," *Optim. Control Appl. Meth.*, vol. 41, pp. 292–311, 2020.

[12] C. Y. Kaya and J. L. Noakes, "Computational method for time-optimal switching control," *J. Optim. Theory Appl.*, vol. 117, pp. 69–92, 2003.

[13] X. Xu and P. J. Antsaklis, "Optimal control of switched systems based on parameterization of the switching instants," *IEEE Trans. Autom. Contr.*, vol. 49, no. 1, pp. 2–16, 2004.

[14] H. Maurer, C. Büskens, J.-H. R. Kim, and C. Y. Kaya, "Optimization methods for the verification of second order sufficient conditions for bang-bang controls," *Optim. Control Appl. Methods*, vol. 26, no. 3, pp. 129–156, 2005.

[15] G. Vossen, "Switching time optimization for bang-bang and singular controls," *J. Optim. Theory Appl.*, vol. 144, pp. 409–429, 2010.

[16] B. Srinivasan, S. Palanki, and D. Bonvin, "Dynamic optimization of batch processes: I. Characterization of the nominal solution," *Comput. Chem. Eng.*, vol. 27, no. 1, pp. 1–26, 2003.

[17] D. Rodrigues and D. Bonvin, "Dynamic optimization of reaction systems via exact parsimonious input parameterization," *Ind. Eng. Chem. Res.*, vol. 58, no. 26, pp. 11 199–11 212, 2019.

[18] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko, *The Mathematical Theory of Optimal Processes*. New York: Interscience, 1962.

[19] D. Henrion and J. B. Lasserre, "GloptiPoly: Global optimization over polynomials with Matlab and SeDuMi," *ACM Trans. Math. Softw.*, vol. 29, no. 2, pp. 165–194, 2003.

[20] H. Waki, S. Kim, M. Kojima, and M. Muramatsu, "Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity," *SIAM J. Optim.*, vol. 17, no. 1, pp. 218–242, 2006.

[21] R. A. Lorentz, "Multivariate Hermite interpolation by algebraic polynomials: A survey," *J. Comput. Appl. Math.*, vol. 122, no. 1-2, pp. 167–201, 2000.

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

[23] V. B. Alekseev, *Abel's Theorem in Problems and Solutions*. Springer, 2004.

[24] D. Rodrigues, M. R. Abdalmoaty, and H. Hjalmarsson, "Toward tractable global solutions to maximum-likelihood estimation problems via sparse sum-of-squares relaxations," in *Proc. 58th IEEE Conference on Decision and Control (CDC)*, Nice, France, 2019, pp. 3184–3189.

[25] ——, "Toward tractable global solutions to Bayesian point estimation problems via sparse sum-of-squares relaxations," in *Proc. 2020 American Control Conference (ACC)*, Denver, CO, 2020, pp. 1501–1506.

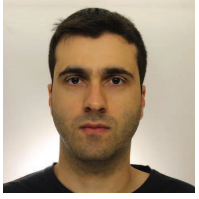
[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[27] M. Putinar, "Positive polynomials on compact semi-algebraic sets," *Ind. Univ. Math. J.*, vol. 42, pp. 969–984, 1993.

[28] J. B. Lasserre, "A semidefinite programming approach to the generalized problem of moments," *Math. Program.*, vol. 112, pp. 65–92, 2008.

[29] A. A. Ahmadi, G. Hall, A. Papachristodoulou, J. Saunderson, and Y. Zheng, "Improving efficiency and scalability of sum of squares optimization: Recent advances and limitations," in *Proc. 56th IEEE Conference on Decision and Control (CDC)*, Melbourne, VIC, Australia, 2017, pp. 453–462.

[30] H. Seywald and E. M. Cliff, "Goddard problem in presence of a dynamic pressure limit," *J. Guid. Control Dyn.*, vol. 16, no. 4, pp. 776–781, 1993.



Diogo Rodrigues received a PhD degree in Chemistry and Chemical Engineering with a distinction for a remarkable PhD thesis from the Swiss Federal Institute of Technology Lausanne (EPFL) in 2018. Between 2018 and 2019, he was a postdoctoral researcher at the Division of Decision and Control Systems, KTH Royal Institute of Technology. Between 2019 and 2021, Dr. Rodrigues was a postdoctoral scholar at the Department of Chemical and Biomolecular Engineering, University of California, Berkeley, with the support of a fellowship grant from the Swiss National Science Foundation. Since 2021, he is a postdoctoral researcher at Centro de Química Estrutural, Instituto Superior Técnico. His research interests include process control and optimization, system identification, and optimal control.

This article has been accepted for publication in IEEE Transactions on Automatic Control. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TAC.2022.3165481

Ali Mesbah is Associate Professor of Chemical and Biomolecular Engineering at the University of California at Berkeley. Before joining UC Berkeley, Dr. Mesbah was a senior postdoctoral associate at MIT. He holds a Ph.D. degree in Systems and Control from Delft University of Technology. Dr. Mesbah is a senior member of the IEEE Control Systems Society and AIChE. He serves on the IEEE Control Systems Society Conference Editorial Board and IEEE Control Systems Society Technology Conference Editorial Board, and is a subject editor of Optimal Control Applications and Methods and IEEE Transactions on Radiation and Plasma Medical Sciences. Dr. Mesbah is recipient of the Best Application Paper Award of the IFAC World Congress in 2020, the AIChE's 35 Under 35 Award in 2017, the IEEE Control Systems Outstanding Paper Award in 2017, and the AIChE CAST W. David Smith, Jr. Publication Award in 2015. His research interests lie at the intersection of optimal control, machine learning, and applied mathematics, with applications to learning-based analysis, diagnosis, and predictive control of manufacturing systems.