Toward Efficient Global Solutions to Optimal Control Problems via Second-Order Polynomial Approximations ⋆

Diogo Rodrigues *

* *Laboratory of Separation and Reaction Engineering - Laboratory of Catalysis and Materials, Associate Laboratory in Chemical Engineering, Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, 4200-465 Porto, Portugal (dfrodrigues@fe.up.pt)*

Abstract: Optimal control problems are used for many tasks such as model-based control, state and parameter estimation, and experimental design for complex dynamic systems. The solution to these problems can be divided into two tasks, where the first corresponds to the enumeration of different arc sequences and the second is the computation of the optimal values of the decision variables for each arc sequence. For the latter task, this paper proposes a method to approximate the cost and constraints of the problem as polynomial functions of the decision variables via computation of partial derivatives up to second order and multivariate Hermite interpolation. This method allows reformulating the problem for an arc sequence as a polynomial optimization problem, which is expected to enable efficiently solving optimal control problems to global optimality. The method is illustrated by a simulation example of a reaction system.

Keywords: optimal control, polynomial optimization, global optimality

1. INTRODUCTION

Optimal control problems (OCPs) are widely applied to design, analysis, and operation of many complex dynamic systems. Efficient solution methods for OCPs are vital for tasks such as state and parameter estimation, experimental design, and model-based control. OCPs involve selecting optimal time-varying functions over a time interval to optimize a cost subject to constraints. OCPs are complex since they involve infinitely many decision variables, and there exist not only (terminal) constraints at the end of the time interval but also (path) constraints along the trajectory (Bryson and Ho, 1975). Specialized techniques for solving OCPs have been developed, such as direct methods that reformulate the original infinite-dimensional problem as a finite-dimensional one via discretization, by dividing the time interval into subintervals where the solution description is simpler (Teo et al., 1991; Biegler et al., 2002).

Traditional direct methods for OCPs only seek local optimality, but not global optimality. The local optima attained by these algorithms may be suboptimal with respect to the global optimum by a significant margin (Houska and Chachuat, 2014). Alternatively, global optimization algorithms can be used. Two approaches can be highlighted: branch-and-bound approaches and reformulation as a convex problem. Branch-and-bound approaches estimate bounds of the cost and constraints and divide the space of decision variables until the global optimum is found (Chachuat et al., 2006). The alternative is to reformulate the original nonconvex problem, which can possess several local optima, 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 written in terms of polynomials, one can express the problem as a polynomial optimization problem (POP), which can be reformulated as a hierarchy of convex semidefinite programs (SDPs) via the concept of sum-of-squares (SOS) polynomials (Lasserre, 2001; Parrilo, 2003). However, both approaches to global optimization have exponential worst-case complexity with respect to the number of decision variables (Houska and Chachuat, 2014). In addition, convergence to global optimality is challenging because the number of decision variables in these problems is large even after discretization via direct methods.

To reduce the number of decision variables for global optimization, the parsimonious input parameterization approach has been proposed (Rodrigues and Bonvin, 2020), which consists of: (i) identifying all the arcs that can occur in the solution, (ii) generating a finite set of arc sequences, and (iii) describing each sequence by a small number of decision variables. Then, for a given arc sequence, the optimal values of the decision variables of the resulting problem can be computed. This differs from switching time optimization, which assumes that an analytical expression for each input is available and is challenging for input-affine OCPs related to complex nonlinear systems. The main advantage is that the small number of decision variables enables efficient global optimization for each arc sequence. This motivates approximating the terminal cost and constraints as explicit polynomial functions of the decision variables, resulting in a POP for each arc sequence, for which the global solution can be computed via reformulation as an SDP.

Previous work showed how to extend the parsimonious input parameterization approach for efficient global solutions

⋆ This work was supported by national funds through FCT/MCTES (PIDDAC): LSRE-LCM, UIDB/50020/2020 (DOI: 10.54499/UIDB/50020/2020) and UIDP/50020/2020 (DOI: 10.54499/UIDP/50020/2020); and ALiCE, LA/P/0045/2020 (DOI: 10.54499/LA/P/0045/2020).

to approximations of OCPs. A method for approximating the problem for a given arc sequence as a POP via multivariate Hermite interpolation was presented, and it was shown that the POP can be solved efficiently to global optimality via the concept of SOS polynomials (Rodrigues and Mesbah, 2022). However, this work focused on single-input OCPs, approximated free/singular arcs as linear functions of time, and used partial derivatives of the cost and constraints with respect to the decision variables only up to first order and multivariate Hermite interpolation to approximate the problem for a given arc sequence as a POP.

The goal of this paper is to show how to represent the cost and constraints of an OCP as explicit polynomial functions of the decision variables for each arc sequence, by taking advantage of the partial derivatives of the cost and constraints with respect to the decision variables up to second order. This way, the OCP is reformulated as a set of POPs, one for each arc sequence. The paper shows the OCP formulation, the structure of its solution, and how to reduce the number of decision variables in the problem for a given arc sequence by approximating the inputs in free/singular arcs with low-degree polynomials. Then, it presents a method to reformulate that problem as a POP using partial derivatives of the cost and constraints up to second order and multivariate Hermite interpolation. Finally, the method is illustrated via a simulation example.

2. OPTIMAL CONTROL PROBLEM AND SOLUTION

2.1 Problem formulation

This paper concerns the solution to OCPs formulated as

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

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

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

$$\mathbf{g}\big(\mathbf{x}(t),\mathbf{u}(t)\big) \leq \mathbf{0}_{n_g}, \qquad \mathbf{h}\big(\mathbf{x}(t)\big) \leq \mathbf{0}_{n_h}, \tag{1d}$$

where: t_0 is the initial time, $t_1 < \cdots < t_T$ are T times and $t_f = t_T \in [t_0, t_{max}]$ is the final time, with $t_0, \ldots, t_{T-1}$ fixed and t_{max} a finite upper bound; $\mathbf{u}(t)$ is the n_u-dimensional vector of piecewise-continuous inputs for all $t \in [t_0, t_f)$; $\mathbf{x}(t)$ is the n_x-dimensional vector of piecewise-continuously differentiable states for all $t \in [t_0, t_f)$; $\mathbf{f}(\mathbf{x},\mathbf{u})$ and $\mathbf{g}(\mathbf{x},\mathbf{u})$ are n_x-dimensional and n_g-dimensional vector functions, 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 $\mathbf{x} \in \mathbb{R}^{n_x}$; and $\phi(\mathbf{X},t)$ and $\boldsymbol{\psi}(\mathbf{X},t)$ are a scalar function and an n_ψ-dimensional vector function, smooth for all $(\mathbf{X},t) \in \mathbb{R}^{Tn_x} \times [t_0, t_{max}]$. It is assumed that $\mathbf{g}(\mathbf{x},\mathbf{u})$ and $\mathbf{h}^{(1)}(\mathbf{x},\mathbf{u}) := \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x})\mathbf{f}(\mathbf{x},\mathbf{u})$ depend explicitly on $\mathbf{u}$ and $\mathbf{h}(\mathbf{x})$ is linear in $\mathbf{x}$.

2.2 Solution structure

The solution to Problem (1) involves inputs composed of several arcs. For each input u_j, each arc is 1) bang-bang, such that it is determined by an equality $g_k(\mathbf{x},\mathbf{u}) = 0$ for some $k = 1,\ldots,n_g$, 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 system dynamics given by $\mathbf{f}\big(\mathbf{x}(t),\mathbf{u}(t)\big)$, also labeled as singular in the case of input-affine OCPs (Rodrigues and Bonvin, 2020). Hence, arc sequences can be formed from a finite number of arc types. If we consider as plausible only arc sequences with an upper bound on the number of arcs, the number of plausible sequences is also finite. The solution to the OCP relies on determining (i) when and how the optimal switching between arcs takes place for a given arc sequence, and (ii) which sequence provides the optimal solution. This paper focuses on (i) and assumes that the inputs in free/singular arcs are approximated by low-degree polynomials described by only a few parameters that correspond to the initial conditions of these arcs. Once (i) is addressed for each sequence via parallel computing, (ii) becomes straightforward.

The solution to (i) involves computing the optimal values of the decision variables for a given plausible arc sequence. It is then helpful to express the cost and constraints of the OCP as explicit polynomial functions of the decision variables, converting the OCP into a set of POPs, one for each arc sequence. Hence, for a plausible arc sequence with n_s switching times to arcs of types 1 and 3 (excluding t_0 as a switching time), the inputs $\mathbf{u}(\cdot)$ are fully defined by the decision variables, which are the switching times $\bar{t}_1,\ldots,\bar{t}_{n_s}$, the final time t_f, and the initial conditions of the free/singular arcs. In contrast to multi-point boundary value problems and switching time optimization, both the switching times and those initial conditions are decision variables, which allows representing singular arcs in input-affine OCPs related to complex nonlinear systems. The entry points in arcs of type 2 are given by the n_η-dimensional vector $\boldsymbol{\eta}$, but the switching to these arcs cannot occur at arbitrary times since it depends on the states $\mathbf{x}$. 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 simplicity, as explained by Rodrigues and Mesbah (2022).

Next, the inputs for a given arc sequence are described in the i^{th} time interval $[\bar{t}_{i-1},\bar{t}_i)$, for $i = 1,\ldots,n_s+1$, with $\bar{t}_0 = t_0$, $\bar{t}_{n_s+1} = t_f$. Then, suppose that, in the interval $[\bar{t}_{i-1},\bar{t}_i)$, there are $\tilde{n}_i$ switchings to different arcs of type 2, and the arcs are described in the r_ith interval $[\bar{t}_{i,r_i-1},\bar{t}_{i,r_i})$, for $r_i = 1,\ldots,\tilde{n}_i+1$, with $\bar{t}_{i,0} = \bar{t}_{i-1}$, $\bar{t}_{i,\tilde{n}_i+1} = \bar{t}_i$. For each input u_j, with $j = 1,\ldots,n_u$, there is a degree $\xi_{j,i} \geq 0$ for which a feedback law gives $u_j^{(\xi_{j,i})}(t)$ at the beginning of the interval $[\bar{t}_{i-1},\bar{t}_i)$ and a degree $\xi_{j,i} \geq \xi_{j,i,r_i} \geq 0$ for which a feedback law gives $u_j^{(\xi_{j,i,r_i})}(t)$ in the interval $[\bar{t}_{i,r_i-1},\bar{t}_{i,r_i})$ as an explicit function of the states, the inputs, and an optional parameter $p_{j,i}$ in the case of free/singular arcs.

For a bang-bang or active-state constraint arc for input u_j, $\xi_{j,i,r_i} = 0$ since this arc is determined by an equality

$$u_j(t) = \bar{c}_{j,i}^{r_i}\big(\mathbf{x}(t),u_1(t),\ldots,u_{j-1}(t),u_{j+1}(t),\ldots,u_{n_u}(t)\big). \tag{2}$$

For a free/singular arc for input u_j, $\xi_{j,i,r_i} \geq 0$ since $u_j(t)$ is described in this arc by the differential equation

$$u_j^{(\xi_{j,i,r_i})}(t) = p_{j,i}, \tag{3}$$

with initial conditions $u_j^{(\xi_{j,i,r_i}-1)}(\bar{t}_{i,r_i-1}),\ldots,u_j(\bar{t}_{i,r_i-1})$.

Hence, the parameter $p_{j,i}$ is of dimension $b_{j,i}$, where $b_{j,i}$ is a binary constant that specifies whether $u_j(t)$ is free/singular at the beginning of the i^{th} time interval, for $t \in [\bar{t}_{i-1},\bar{t}_i)$.

3. OCP WITH NEW DECISION VARIABLES

The dynamic feedback in the case $\xi_{j,i} > 0$ can be handled by defining the following $n_{z,i} := \xi_{1,i}+b_{1,i}+\ldots+\xi_{n_u,i}+b_{n_u,i}$ new states and initial conditions:

$$\mathbf{z}_i(t) := \begin{bmatrix} \left[\, \tilde{u}^0_{1,i}(t) \,\cdots\, \tilde{u}^{\xi_{1,i}-1}_{1,i}(t) \; \tilde{p}_{1,i}(t) \,\right]^{\mathrm{T}} \\ \vdots \\ \left[\, \tilde{u}^0_{n_u,i}(t) \,\cdots\, \tilde{u}^{\xi_{n_u,i}-1}_{n_u,i}(t) \; \tilde{p}_{n_u,i}(t) \,\right]^{\mathrm{T}} \end{bmatrix}, \tag{4a}$$

$$\mathbf{z}_{i,0} := \begin{bmatrix} \left[\, u^0_{1,i} \,\cdots\, u^{\xi_{1,i}-1}_{1,i} \; p_{1,i} \,\right]^{\mathrm{T}} \\ \vdots \\ \left[\, u^0_{n_u,i} \,\cdots\, u^{\xi_{n_u,i}-1}_{n_u,i} \; p_{n_u,i} \,\right]^{\mathrm{T}} \end{bmatrix}, \tag{4b}$$

which describes the inputs for a given arc sequence in the i^{th} time interval $[\bar{t}_{i-1}, \bar{t}_i)$, for $i = 1, \ldots, n_s + 1$. Then, one can combine all the states into augmented vectors with a dimension $n_z := n_x + n_{z,1} + \ldots + n_{z,n_s+1}$, as follows:

$$\mathbf{z}(t) := \left[\mathbf{x}(t)^{\mathrm{T}} \; \mathbf{z}_1(t)^{\mathrm{T}} \; \cdots \; \mathbf{z}_{n_s+1}(t)^{\mathrm{T}}\right]^{\mathrm{T}}. \tag{5}$$

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

The dynamics of the states $\mathbf{z}_i(t)$ are given by

$$\dot{\mathbf{z}}_i(t) = \mathbf{q}_i\big(\mathbf{z}(t)\big), \quad \mathbf{z}_i(\bar{t}_{i-1}) = \mathbf{z}_{i,0}, \tag{6}$$

where, for $t \in [\bar{t}_{i-1}, \bar{t}_i)$,

$$\mathbf{q}_i\big(\mathbf{z}(t)\big) = \Big[\mathbf{q}_{1,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big)^{\mathrm{T}} \; \cdots \; \mathbf{q}_{n_u,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big)^{\mathrm{T}}\Big]^{\mathrm{T}}, \tag{7}$$

with, for $j = 1, \ldots, n_u$ and for $t \in [\bar{t}_{i,r_i-1}, \bar{t}_{i,r_i})$,

$$\mathbf{q}_{j,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big) = \left[\, \tilde{u}^1_{j,i}(t) \,\cdots\, \tilde{u}^{\xi_{j,i}-1}_{j,i}(t) \; \tilde{p}_{j,i}(t) \; 0 \,\right]^{\mathrm{T}} \tag{8}$$

if $\xi_{j,i,r_i} > 0$, and

$$\mathbf{q}_{j,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big) = \mathbf{0}_{\xi_{j,i}+b_{j,i}} \tag{9}$$

if $\xi_{j,i,r_i} = 0$, and, for $t \notin [\bar{t}_{i-1}, \bar{t}_i)$,

$$\mathbf{q}_i\big(\mathbf{z}(t)\big) = \mathbf{0}_{n_{z,i}}. \tag{10}$$

This allows using the control laws $\mathbf{u}(t) = \tilde{\mathbf{c}}\big(\mathbf{z}(t)\big)$, where, for $j = 1, \ldots, n_u$ and for $t \in [\bar{t}_{i,r_i-1}, \bar{t}_{i,r_i})$,

$$\tilde{c}_j\big(\mathbf{z}(t)\big) = \begin{cases} \tilde{u}^0_{j,i}(t), & \xi_{j,i,r_i} > 0 \\ c^{r_i}_{j,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big), & \xi_{j,i,r_i} = 0 \end{cases}, \tag{11}$$

with $c^{r_i}_{j,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big) = \tilde{p}_{j,i}(t)$ if the arc is free/singular for u_j and $c^{r_i}_{j,i}\big(\mathbf{x}(t), \mathbf{z}_i(t)\big)$ obtained from (2) otherwise.

For example, if u_j is approximated by a linear function and the arcs are not free/singular for the other inputs, then $\mathbf{z}_i(t) = \left[\begin{smallmatrix} \tilde{u}^0_{j,i}(t) \\ \tilde{p}_{j,i}(t) \end{smallmatrix}\right]$ and $\mathbf{z}_{i,0} = \left[\begin{smallmatrix} u^0_{j,i} \\ p_{j,i} \end{smallmatrix}\right]$ are of dimension $\xi_{j,i} + 1 = 2$, where $u^0_{j,i}$ and $p_{j,i}$ are the initial value and derivative of the input u_j and $\tilde{u}^0_{j,i}(t)$ is its value at time t, which implies that $\tilde{c}_j\big(\mathbf{z}(t)\big) = \tilde{u}^0_{j,i}(t)$ and $\mathbf{q}_i\big(\mathbf{z}(t)\big) = \left[\begin{smallmatrix} \tilde{p}_{j,i}(t) \\ 0 \end{smallmatrix}\right]$.

Then, upon eliminating input dependencies and rewriting Problem (1) in terms of the extended states $\mathbf{z}$, one reformulates Problem (1) in terms of the new decision variables $\boldsymbol{\tau} := (\bar{t}_1, \ldots, \bar{t}_{n_s}, t_f, \mathbf{z}_{1,0}, \ldots, \mathbf{z}_{n_s+1,0})$ as

$$\min_{\boldsymbol{\tau}} \quad \hat{\phi}(\boldsymbol{\tau}) := \tilde{\phi}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big), \tag{12a}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\psi}}(\boldsymbol{\tau}) := \tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \leq \mathbf{0}_{n_\psi}, \tag{12b}$$

$$\dot{\mathbf{z}}(t) = \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big), \qquad \mathbf{z}(t_0) = \mathbf{z}_0, \tag{12c}$$

$$\bar{t}_{i-1} \leq \bar{t}_i, \quad i = 1, \ldots, n_s + 1, \tag{12d}$$

$$\tilde{\mathbf{g}}\big(\mathbf{z}(t)\big) \leq \mathbf{0}_{n_g}, \qquad \tilde{\mathbf{h}}\big(\mathbf{z}(t)\big) \leq \mathbf{0}_{n_h}, \tag{12e}$$

where

$$\tilde{\boldsymbol{\chi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) := \begin{bmatrix} \tilde{\phi}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \\ \tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \end{bmatrix}, \tag{13}$$

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 $\tilde{\boldsymbol{\psi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$ defined similarly, the dynamics

$$\tilde{\mathbf{f}}\big(\mathbf{z}(t)\big) := \begin{bmatrix} \mathbf{f}\big(\mathbf{x}(t), \tilde{\mathbf{c}}(\mathbf{z}(t))\big)^{\mathrm{T}} & \begin{bmatrix} \mathbf{q}_1\big(\mathbf{z}(t)\big) \\ \vdots \\ \mathbf{q}_{n_s+1}\big(\mathbf{z}(t)\big) \end{bmatrix}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{14}$$

$\tilde{\mathbf{g}}\big(\mathbf{z}(t)\big) := \mathbf{g}\big(\mathbf{x}(t), \tilde{\mathbf{c}}\big(\mathbf{z}(t)\big)\big)$, and $\tilde{\mathbf{h}}\big(\mathbf{z}(t)\big) := \mathbf{h}\big(\mathbf{x}(t)\big)$, which is more convenient for numerical optimization since there are only $N := n_s + 1 + n_{z,1} + \ldots + n_{z,n_s+1}$ decision variables, in contrast to infinite-dimensional variables in Problem (1). For this reason, the input parameterization is terms of $\boldsymbol{\tau}$ has been labeled parsimonious (Rodrigues and Bonvin, 2020). For any $\boldsymbol{\tau}$, the terminal cost and constraints $\hat{\boldsymbol{\chi}}(\boldsymbol{\tau}) := \left[\hat{\phi}(\boldsymbol{\tau}) \; \hat{\boldsymbol{\psi}}(\boldsymbol{\tau})^{\mathrm{T}}\right]^{\mathrm{T}}$ are computed via numerical integration and evaluation of $\tilde{\boldsymbol{\chi}}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$.

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 $\tilde{h}_k\big(\mathbf{z}(t)\big) \leq 0$ becomes active at $t = \hat{\eta}_j(\boldsymbol{\tau})$.

4. REFORMULATION AS A POLYNOMIAL OPTIMIZATION PROBLEM

We aim to reformulate the OCP for each arc sequence as a POP that is more amenable to global optimization, which entails expressing every function $\hat{\chi}(\boldsymbol{\tau})$ in the terminal cost and constraints and in the constraints that place each entry point $\hat{\eta}_j(\boldsymbol{\tau})$ in arcs of type 2 with respect to $\bar{t}_1, \ldots, \bar{t}_{n_s}$ according to the optimal sequence of arcs of types 1, 2, and 3 as a polynomial function. To this end, we can compute higher-order partial derivatives of that function $\hat{\chi}(\boldsymbol{\tau})$ with respect to $\boldsymbol{\tau}$ and use a multivariable series expansion to represent $\hat{\chi}(\boldsymbol{\tau})$ as a polynomial in $\boldsymbol{\tau}$, as shown next.

4.1 Computation of partial derivatives

Suppose that there exists $\bar{\boldsymbol{\tau}}$ such that, for all $\Delta\boldsymbol{\tau} \in \mathcal{R}$,

$$\hat{\chi}(\boldsymbol{\tau}) = \textstyle\sum_{\mathbf{k} \in \mathcal{K}^N_n} (\mathbf{c}_{\hat{\chi}})_{\mathbf{k}} \, \Delta\boldsymbol{\tau}^{\mathbf{k}} + R_{\hat{\chi}}(\boldsymbol{\tau}), \tag{15}$$

with $(\mathbf{c}_{\hat{\chi}})_{\mathbf{k}} := \frac{1}{\mathbf{k}!} \frac{\partial^{\mathbf{k}} \hat{\chi}}{\partial \boldsymbol{\tau}^{\mathbf{k}}}(\bar{\boldsymbol{\tau}})$, $\mathbf{k}$ the vector of monomial powers in the set $\mathcal{K}^N_n := \{(k_1, \ldots, k_N) \in \mathbb{N}^N_0 : 0 \leq k_1 + \ldots + k_N \leq n\}$ in the case of a polynomial of degree n, $\Delta\boldsymbol{\tau} := \boldsymbol{\tau} - \bar{\boldsymbol{\tau}}$ the deviation of $\boldsymbol{\tau}$ around $\bar{\boldsymbol{\tau}}$, $\mathbf{k}! := k_1! \ldots k_N!$, $\Delta\boldsymbol{\tau}^{\mathbf{k}} := (\tau_1 - \bar{\tau}_1)^{k_1} \ldots (\tau_N - \bar{\tau}_N)^{k_N}$, $\frac{\partial^{\mathbf{k}}}{\partial \boldsymbol{\tau}^{\mathbf{k}}} := \frac{\partial^{k_1 + \ldots + k_N}}{\partial \tau_1^{k_1} \ldots \partial \tau_N^{k_N}}$.

Next, we show how to compute the partial derivatives in (15). For this, it is essential to consider not only the extended states $\mathbf{z}(t)$ and the adjoint variables

$$\boldsymbol{\zeta}(t) := \left[\boldsymbol{\lambda}(t)^{\mathrm{T}} \; \boldsymbol{\zeta}_1(t)^{\mathrm{T}} \; \cdots \; \boldsymbol{\zeta}_{n_s+1}(t)^{\mathrm{T}}\right]^{\mathrm{T}}, \tag{16}$$

but also their partial derivatives with respect to $\boldsymbol{\tau}$, as well as the concept of modified Hamiltonian function

$$\tilde{H}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) = \frac{\partial \tilde{\chi}}{\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) + \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big)^{\mathrm{T}} \boldsymbol{\zeta}(t) \tag{17}$$

and its partial derivatives with respect to $\boldsymbol{\tau}$.

Appendix A shows the description of the states, adjoint variables, and their partial derivatives with respect to $\boldsymbol{\tau}$. All the equations in Appendix A can be obtained from the concept of states and adjoint variables and the application of the chain rule and the triple product rule.

Furthermore, the first-order partial derivatives of the modified Hamiltonian function with respect to $\boldsymbol{\tau}$ correspond to

$$\frac{\partial \tilde{H}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$$
$$= \frac{\partial^2 \tilde{\chi}}{\partial t_f^2}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\frac{\partial t_f}{\partial \boldsymbol{\tau}}$$
$$+ \sum_{l=1}^{T} \frac{\partial^2 \tilde{\chi}}{\partial t_f \partial \mathbf{z}(t_l)}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_l) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_l)\frac{\partial t_l}{\partial \boldsymbol{\tau}}\right)$$
$$+ \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big)^{\mathrm{T}}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t)\frac{\partial t}{\partial \boldsymbol{\tau}}\right)$$
$$+ \boldsymbol{\zeta}(t)^{\mathrm{T}}\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(t)\big)\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t)\frac{\partial t}{\partial \boldsymbol{\tau}}\right). \quad (18)$$

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

$$\frac{\partial \hat{\chi}}{\partial \bar{t}_i}(\boldsymbol{\tau}) = \tilde{H}\big(\mathbf{z}(\bar{t}_i^-), \boldsymbol{\zeta}(\bar{t}_i), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) - \tilde{H}\big(\mathbf{z}(\bar{t}_i), \boldsymbol{\zeta}(\bar{t}_i), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$$
$$= \big(\tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i)\big)\big)^{\mathrm{T}}\boldsymbol{\zeta}(\bar{t}_i),$$
$$i = 1, \ldots, n_s, \quad (19)$$
$$\frac{\partial \hat{\chi}}{\partial t_f}(\boldsymbol{\tau}) = \tilde{H}\big(\mathbf{z}(t_f^-), \boldsymbol{\zeta}(t_f), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$$
$$= \frac{\partial \tilde{\chi}}{\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) + \tilde{\mathbf{f}}\big(\mathbf{z}(t_f^-)\big)^{\mathrm{T}}\boldsymbol{\zeta}(t_f), \quad (20)$$
$$\frac{\partial \hat{\chi}}{\partial \mathbf{z}_{i,0}}(\boldsymbol{\tau})^{\mathrm{T}} = \boldsymbol{\zeta}_i(t_0), \quad i = 1, \ldots, n_s + 1, \quad (21)$$

and its second-order partial derivatives with respect to $\boldsymbol{\tau}$

$$\frac{\partial^2 \hat{\chi}}{\partial \bar{t}_i \partial \boldsymbol{\tau}}(\boldsymbol{\tau}) = \frac{\partial \tilde{H}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(\bar{t}_i^-), \boldsymbol{\zeta}(\bar{t}_i), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) - \frac{\partial \tilde{H}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(\bar{t}_i), \boldsymbol{\zeta}(\bar{t}_i), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$$
$$= \big(\tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\bar{t}_i)\big)\big)^{\mathrm{T}}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\bar{t}_i) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\bar{t}_i)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}}\right)$$
$$+ \boldsymbol{\zeta}(\bar{t}_i)^{\mathrm{T}}\left(\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(\bar{t}_i^-)\big) - \frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(\bar{t}_i)\big)\right)\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\bar{t}_i) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\bar{t}_i)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}}\right),$$
$$i = 1, \ldots, n_s, \quad (22)$$
$$\frac{\partial^2 \hat{\chi}}{\partial t_f \partial \boldsymbol{\tau}}(\boldsymbol{\tau}) = \frac{\partial \tilde{H}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(t_f^-), \boldsymbol{\zeta}(t_f), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$$
$$= \frac{\partial^2 \tilde{\chi}}{\partial t_f^2}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\frac{\partial t_f}{\partial \boldsymbol{\tau}}$$
$$+ \sum_{l=1}^{T} \frac{\partial^2 \tilde{\chi}}{\partial t_f \partial \mathbf{z}(t_l)}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_l) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_l)\frac{\partial t_l}{\partial \boldsymbol{\tau}}\right)$$
$$+ \tilde{\mathbf{f}}\big(\mathbf{z}(t_f^-)\big)^{\mathrm{T}}\left(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t_f) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t_f)\frac{\partial t_f}{\partial \boldsymbol{\tau}}\right)$$
$$+ \boldsymbol{\zeta}(t_f)^{\mathrm{T}}\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(t_f^-)\big)\left(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_f^-) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_f^-)\frac{\partial t_f}{\partial \boldsymbol{\tau}}\right), \quad (23)$$
$$\frac{\partial^2 \hat{\chi}}{\partial \mathbf{z}_{i,0} \partial \boldsymbol{\tau}}(\boldsymbol{\tau}) = \frac{\partial \boldsymbol{\zeta}_i}{\partial \boldsymbol{\tau}}(t_0) + \frac{\mathrm{d}\boldsymbol{\zeta}_i}{\mathrm{d}t}(t_0)\frac{\partial t_0}{\partial \boldsymbol{\tau}}, \quad i = 1, \ldots, n_s + 1. \quad (24)$$

These equations are useful to handle entry points in arcs of type 2, which is difficult if automatic differentiation is used.

In addition, the higher-order partial derivatives of $\mathbf{z}(t)$, $\boldsymbol{\zeta}(t)$, $\tilde{H}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)$, and $\hat{\chi}(\boldsymbol{\tau})$ with respect to $\boldsymbol{\tau}$ could be computed. However, the computation of higher-order partial derivatives with respect to $\boldsymbol{\tau}$ would become complex due to the need to perform consecutive symbolic differentiation, which would result in increasingly complex expressions for increasing differentiation order. Polynomial interpolation is used next to deal with this issue.

4.2 Polynomial interpolation

A more efficient approach consists of (i) computing the partial derivatives of every function $\hat{\chi}(\boldsymbol{\tau})$ in the cost and constraints up to second order with respect to $\boldsymbol{\tau}$ and (ii) using multivariate Hermite interpolation to obtain a polynomial of degree $n > 2$ such as the one in (15) that fits the value $\hat{\chi}(\boldsymbol{\tau}_l)$ and the first-order and second-order partial derivatives $\frac{\partial \hat{\chi}}{\partial \boldsymbol{\tau}}(\boldsymbol{\tau}_l)$, $\frac{\partial^2 \hat{\chi}}{\partial \boldsymbol{\tau}^2}(\boldsymbol{\tau}_l)$ at the sample points $\boldsymbol{\tau}_l$, for $l = 1, \ldots, m_\tau$ (Lorentz, 2000). Note that this requires no more than computing the extended states $\mathbf{z}(t)$ and adjoint variables $\boldsymbol{\zeta}(t)$ for every $\hat{\chi}(\boldsymbol{\tau})$ and their first-order partial derivatives $\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t)$, $\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t)$ that correspond to each point $\boldsymbol{\tau}_l$, which amounts to solving $n_\chi + 1$ systems of $n_z\,(N+1)$ differential equations for each $l = 1, \ldots, m_\tau$.

Hence, instead of computing the partial derivatives $\frac{\partial^{\mathbf{k}} \hat{\chi}}{\partial \boldsymbol{\tau}^{\mathbf{k}}}(\bar{\boldsymbol{\tau}})$ in (15) directly, one can compute the coefficient vector $\hat{\mathbf{c}}_{\hat{\chi}}$ that minimizes $\sum_{\boldsymbol{\kappa} \in \mathcal{K}_2^N} \left(\sigma^2_{\hat{\chi},\boldsymbol{\kappa}}\right)^{-1} ||\mathbf{y}_{\hat{\chi},\boldsymbol{\kappa}} - \mathbf{A}_{\tau,\boldsymbol{\kappa}}\hat{\mathbf{c}}_{\hat{\chi}}||^2$, where $(\hat{\mathbf{c}}_{\hat{\chi}})_{\mathbf{k}}$ is an approximation of $(\mathbf{c}_{\hat{\chi}})_{\mathbf{k}}\,\mathbf{s}^{\mathbf{k}}$, for all $\mathbf{k} \in \mathcal{K}_n^N$,

$$\left(\mathbf{y}_{\hat{\chi},\boldsymbol{\kappa}}\right)_l = \mathbf{s}^{\boldsymbol{\kappa}}\frac{\partial^{\boldsymbol{\kappa}} \hat{\chi}}{\partial \boldsymbol{\tau}^{\boldsymbol{\kappa}}}(\boldsymbol{\tau}_l), \quad \boldsymbol{\kappa} \in \mathcal{K}_2^N, \quad l = 1, \ldots, m_\tau, \quad (25)$$
$$(\mathbf{A}_{\tau,\boldsymbol{\kappa}})_{l,\mathbf{k}} = \begin{cases} \frac{\mathbf{k}!}{(\mathbf{k}-\boldsymbol{\kappa})!}\frac{\Delta\boldsymbol{\tau}_l^{\mathbf{k}-\boldsymbol{\kappa}}}{\mathbf{s}^{\mathbf{k}-\boldsymbol{\kappa}}}, & \mathbf{k} \geq \boldsymbol{\kappa} \\ 0, & \text{otherwise} \end{cases}, \quad \boldsymbol{\kappa} \in \mathcal{K}_2^N,$$
$$l = 1, \ldots, m_\tau, \quad \mathbf{k} \in \mathcal{K}_n^N, \quad (26)$$
$$\sigma^2_{\hat{\chi},\boldsymbol{\kappa}} = \sum_{l=1}^{m_\tau}\left(\left(\mathbf{y}_{\hat{\chi},\boldsymbol{\kappa}}\right)_l - \sum_{l'=1}^{m_\tau}\frac{\left(\mathbf{y}_{\hat{\chi},\boldsymbol{\kappa}}\right)_{l'}}{m}\right)^2, \quad \boldsymbol{\kappa} \in \mathcal{K}_2^N, \quad (27)$$

and $\mathbf{s}$ is a vector of scaling factors for $\boldsymbol{\tau}$.

Note that the coefficient vector $\hat{\mathbf{c}}_{\hat{\chi}}$ is of dimension $\binom{N+n}{N}$, while the number of value vectors $\mathbf{y}_{\hat{\chi},\boldsymbol{\kappa}}$ of dimension m_τ is $\binom{N+2}{N}$. This means that the number m_τ of sample points must be at least $\frac{2(N+n)!}{n!(N+2)!}$, 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 $\hat{\chi}(\boldsymbol{\tau})$

$$p_{\hat{\chi}}(\boldsymbol{\tau}) = \textstyle\sum_{\mathbf{k} \in \mathcal{K}_n^N} (\hat{\mathbf{c}}_{\hat{\chi}})_{\mathbf{k}}\frac{\Delta\boldsymbol{\tau}^{\mathbf{k}}}{\mathbf{s}^{\mathbf{k}}}. \quad (28)$$

Hence, when the cost and constraints are expressed as polynomials in $\boldsymbol{\tau}$ for a given arc sequence, the OCP for that arc sequence can be reformulated as a POP. This problem can then be solved efficiently to global optimality via reformulation as a hierarchy of convex SDPs using the concept of SOS polynomials if global optimality is certified for some small relaxation order (Lasserre, 2001; Rodrigues and Mesbah, 2022). However, methods to solve this problem to global optimality are out of the scope of this paper.

5. SIMULATION EXAMPLE

The simulation example corresponds to a problem of production maximization in an acetoacetylation reaction system with the species pyrrole A, diketene B, 2-acetoacetyl pyrrole C, dehydroacetic acid D, and oligomers E (Rodrigues and Bonvin, 2020). This OCP is formulated mathematically with the states $\mathbf{x}(t) := \left[\mathbf{x}_r(t)^{\mathrm{T}}\ x_{in}(t)\right]^{\mathrm{T}}$ as:

$$\max_{u_{in}(\cdot), t_f} \mathcal{J}(u_{in}(\cdot), t_f) = n_{\mathrm{C}}(t_f), \quad (29a)$$
$$\text{s.t. } \mathcal{T}\big(u_{in}(\cdot), t_f\big) = \begin{bmatrix} n_{\mathrm{B}}(t_f) - \bar{c}_{\mathrm{B}}V(t_f) \\ n_{\mathrm{D}}(t_f) - \bar{c}_{\mathrm{D}}V(t_f) \\ t_f - \bar{t}_f \end{bmatrix} \leq \mathbf{0}_3, \quad (29b)$$
$$\dot{\mathbf{x}}(t) = \mathbf{f}\big(\mathbf{x}(t), u_{in}(t)\big) = \begin{bmatrix} \mathbf{r}_v(t) \\ \frac{u_{in}(t)}{1000} \end{bmatrix},$$
$$\mathbf{x}(t_0) = \mathbf{0}_{R+1}, \quad (29c)$$
$$\mathbf{g}\big(\mathbf{x}(t), u_{in}(t)\big) = \begin{bmatrix} u_{in}(t) - \overline{u}_{in} \\ \underline{u}_{in} - u_{in}(t) \end{bmatrix} \leq \mathbf{0}_2, \quad (29d)$$

where $\underline{u}_{in} = 0$ and $\bar{u}_{in} = 2$ mL min^{-1}, $\bar{t}_f = 250$ min, $\bar{c}_{\mathrm{B}} = 0.025$ mol L^{-1}, and $\bar{c}_{\mathrm{D}} = 0.15$ mol L^{-1}, the $R = 3$ reaction rates 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}, and the volume is given by $V(t) = V_0 + x_{in}(t)$ and the numbers of moles by $\mathbf{n}(t) = \mathbf{N}^{\mathrm{T}} \mathbf{x}_r(t) + \mathbf{c}_{in} x_{in}(t) + \mathbf{n}_0$, with $V_0 = 1$ L, $\mathbf{N} = [\mathbf{n}_1 \ \mathbf{n}_2 \ \mathbf{n}_3]^{\mathrm{T}}$, $\mathbf{n}_1 = [-1 \ -1 \ 1 \ 0 \ 0]^{\mathrm{T}}$, $\mathbf{n}_2 = [0 \ -2 \ 0 \ 1 \ 0]^{\mathrm{T}}$, $\mathbf{n}_3 = [0 \ -1 \ 0 \ 0 \ 1]^{\mathrm{T}}$, $\mathbf{c}_{in} = [0 \ 5 \ 0 \ 0 \ 0]^{\mathrm{T}}$ mol L^{-1}, $\mathbf{n}_0 = [0.72 \ 0.05 \ 0.08 \ 0.01 \ 0]^{\mathrm{T}}$ mol. The numerical values for this example are listed in Table 1.

Table 1. Numerical values used in OCP (29).

Variable	Value	Units
k_1	0.053	L mol^{-1} min^{-1}
k_2	0.128	L mol^{-1} min^{-1}
k_3	0.028	min^{-1}
$c_{in,\mathrm{B}}$	5	mol L^{-1}
$n_{\mathrm{A},0}$	0.72	mol
$n_{\mathrm{B},0}$	0.05	mol
$n_{\mathrm{C},0}$	0.08	mol
$n_{\mathrm{D},0}$	0.01	mol
V_0	1	L
$\bar{c}_{\mathrm{B}}$	0.025	mol L^{-1}
$\bar{c}_{\mathrm{D}}$	0.15	mol L^{-1}
$\bar{t}_f$	250	min
$\underline{u}_{in}$	0	mL min^{-1}
$\bar{u}_{in}$	2	mL min^{-1}

It was shown by Rodrigues and Bonvin (2020) that, when linear polynomials are used to approximate free/singular arcs, a locally optimal solution consists of 3 arcs: in the first arc, $u_{in}^*(t) = \bar{u}_{in}$; the second arc is free/singular with $\underline{u}_{in} < u_{in}^*(t) < \bar{u}_{in}$, for which a linear function is used; and in the third arc, $u_{in}^*(t) = \underline{u}_{in}$. This results in the input trajectory shown in Fig. 1 that is described by the 5 decision variables $\bar{t}_1$, $\bar{t}_2$, $u_{1,2}^0$, $p_{1,2}$, t_f. 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 value and derivative of the linear function that describes $u_{in}^*(t)$ in the second arc are $u_{1,2}^{0*} = 1.262$ mL min^{-1}, $p_{1,2}^* = -1.13 \times 10^{-3}$ 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 of this solution is indicated by the fact that the gradients (19), (20), and (21) are equal to zero and the solution satisfies the necessary conditions given by Pontryagin's maximum principle (Pontryagin et al., 1962).

Next, the described arc sequence is investigated. The goal is not to show computational advantages over state-of-the-art methods for optimal control since these methods can only solve OCPs to local optimality. In contrast, the aim is to show that an accurate polynomial representation of the cost and constraints for a given arc sequence can be obtained since this is expected to be useful to solve OCPs to global optimality via reformulation as POPs. As shown by Rodrigues and Mesbah (2022), the difference between the resulting cost and the globally optimal cost of the original problem depends on the polynomial approximation errors.

Hence, we now describe how the 5 decision variables $\boldsymbol{\tau} = (\bar{t}_1, \bar{t}_2, t_f, u_{1,2}^0, p_{1,2})$ affect the cost and constraints via polynomial functions. To this end, we set $t_f = t_f^*$ and construct a polynomial $p_{\hat{\chi}}(\boldsymbol{\tau})$ of degree $n = 6$ in the $N = 4$ variables $\bar{t}_1, \bar{t}_2, u_{1,2}^0, p_{1,2}$ for every function $\hat{\chi}(\boldsymbol{\tau})$ in the cost and constraints via multivariate Hermite interpolation using partial derivatives up to second order at $m_\tau = 400$ points $\boldsymbol{\tau}_l$, for $l = 1, \ldots, m_\tau$, and the scaling factors $\mathbf{s} = (10, 10, 10, 1, 10^{-3})$, as in (28). Note that the number m_τ of points needs to be at least $\frac{2(N+n)!}{n!(N+2)!} = 14$. The points $\boldsymbol{\tau}_l$ are randomly chosen within the intervals $\bar{t}_1 \in [0, 250]$ min, $\bar{t}_2 \in [0, 250]$ min, $u_{1,2}^0 = [0.0, 2.0]$ mL min^{-1}, $p_{1,2} = [-10, 10] \times 10^{-3}$ mL min^{-2} and evaluated in 109.2 s on an Intel Core i7 3.4 GHz processor. While it is intractable to check if every function $p_{\hat{\chi}}(\boldsymbol{\tau})$ approximates $\hat{\chi}(\boldsymbol{\tau})$ accurately for all $\boldsymbol{\tau}$, one can verify it for a number of validation points, including the relevant case of $\boldsymbol{\tau}^* = (\bar{t}_1^*, \bar{t}_2^*, t_f^*, u_{1,2}^{0*}, p_{1,2}^*)$. Although the points $\boldsymbol{\tau}_l$ are sampled from large intervals for each variable, the polynomial representation of the cost and constraints predicts correctly their true value at $\boldsymbol{\tau}^*$. More precisely, $\hat{\phi}(\boldsymbol{\tau}^*) = 0.51373$ mol, $\hat{\psi}_1(\boldsymbol{\tau}^*) = \hat{\psi}_2(\boldsymbol{\tau}^*) = 0$ mol, while $p_{\hat{\phi}}(\boldsymbol{\tau}^*) = 0.51384$ mol, $p_{\hat{\psi}_1}(\boldsymbol{\tau}^*) = -5.9 \times 10^{-4}$ mol, $p_{\hat{\psi}_2}(\boldsymbol{\tau}^*) = -3.1 \times 10^{-4}$ mol. Accurate predictions (with an error in the order of 10^{-4} mol) are also obtained for other validation points around the points $\boldsymbol{\tau}_l$. Using partial derivatives up to first order as Rodrigues and Mesbah (2022) at $m_\tau = 5000$ points, which are evaluated in 107.2 s, one obtains $p_{\hat{\phi}}(\boldsymbol{\tau}^*) = 0.51393$ mol, $p_{\hat{\psi}_1}(\boldsymbol{\tau}^*) = -7.6 \times 10^{-4}$ mol, $p_{\hat{\psi}_2}(\boldsymbol{\tau}^*) = -3.1 \times 10^{-4}$ mol, which is generally worse.

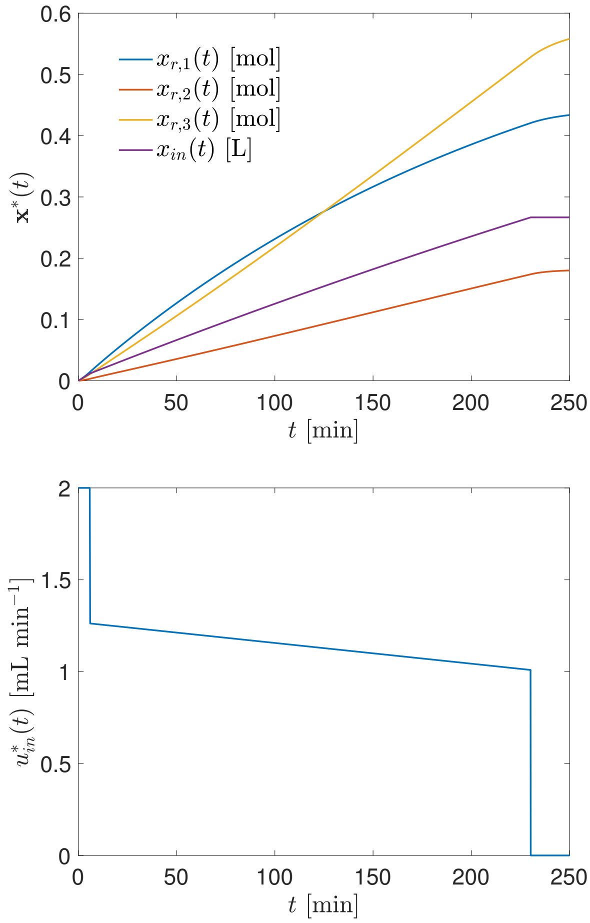

Fig. 1. Trajectories of the states and inputs for the optimal solution to OCP (29) with the parsimonious input parameterization and the approximation of the free/singular arc using linear functions.

6. CONCLUSIONS

This paper has presented a solution method for OCPs that relies on the enumeration of plausible arc sequences and has shown that the cost and constraints for a given arc sequence can be represented as explicit polynomial functions of the decision variables. These polynomial functions are computed via the knowledge of the partial derivatives of the cost and constraints with respect to the decision variables up to second order from the computed states, adjoint variables, and their partial derivatives.

In future work, it will be investigated whether the reformulation as POPs presented in this paper contributes to an efficient implementation of global solutions to OCPs. This is expected since any POP can be reformulated as a hierarchy of convex SDPs via the concept of SOS polynomials.

REFERENCES

Biegler, L.T., Cervantes, A.M., and Wächter, A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.*, 57(4), 575–593.

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

Chachuat, B., Singer, A.B., and Barton, P.I. (2006). Global methods for dynamic optimization and mixed-integer dynamic optimization. *Ind. Eng. Chem. Res.*, 45(25), 8373–8392.

Houska, B. and Chachuat, B. (2014). Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. *J. Optim. Theory Appl.*, 162(1), 208–248.

Lasserre, J.B. (2001). Global optimization with polynomials and the problem of moments. *SIAM J. Optim.*, 11, 796–817.

Lorentz, R.A. (2000). Multivariate Hermite interpolation by algebraic polynomials: A survey. *J. Comput. Appl. Math.*, 122(1-2), 167–201.

Parrilo, P.A. (2003). Semidefinite programming relaxations for semialgebraic problems. *Math. Program.*, 96(2), 293–320.

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

Rodrigues, D. and Bonvin, D. (2020). On reducing the number of decision variables for dynamic optimization. *Optim. Control Appl. Meth.*, 41(1), 292–311.

Rodrigues, D. and Mesbah, A. (2022). Efficient global solutions to single-input optimal control problems via approximation by sum-of-squares polynomials. *IEEE Trans. Automat. Contr.*, 67(9), 4674–4686.

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

Appendix A. DESCRIPTION OF THE STATES, ADJOINT VARIABLES, AND PARTIAL DERIVATIVES

As shown in Problem (12), the extended states $\mathbf{z}(t)$ are described by the differential equations

$$\begin{aligned}\frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t) &= \frac{\partial \tilde{H}}{\partial \boldsymbol{\zeta}}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)^{\mathrm{T}} \\ &= \tilde{\mathbf{f}}\big(\mathbf{z}(t)\big), \quad \mathbf{z}(t_0) = \mathbf{z}_0.\end{aligned}$$

This implies that the first-order partial derivatives of $\mathbf{z}(t)$ with respect to $\boldsymbol{\tau}$ correspond to

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t)\Big) = \frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(t)\big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t)\frac{\partial t}{\partial \boldsymbol{\tau}}\Big), \\ &\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\bar{t}_i) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\bar{t}_i)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}} = \frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\bar{t}_i^-) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\bar{t}_i^-)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}}, \quad i = 1, \ldots, n_s, \\ &\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_0) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_0)\frac{\partial t_0}{\partial \boldsymbol{\tau}} = \frac{\partial \mathbf{z}_0}{\partial \boldsymbol{\tau}},\end{aligned}$$

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

$$\begin{aligned}&\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\eta) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\eta)\frac{\partial \eta}{\partial \boldsymbol{\tau}} = \frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\eta^-) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\eta^-)\frac{\partial \eta}{\partial \boldsymbol{\tau}}, \\ &\frac{\partial \eta}{\partial \boldsymbol{\tau}} = -\Big(\tilde{h}_k^{(1)}\big(\mathbf{z}(\eta^-)\big)\Big)^{-1}\frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\eta^-).\end{aligned}$$

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

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

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

$$\begin{aligned}\boldsymbol{\zeta}(\eta^-) &= \boldsymbol{\zeta}(\eta) - \frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)^{\mathrm{T}}\Big(\tilde{h}_k^{(1)}\big(\mathbf{z}(\eta^-)\big)\Big)^{-1}\frac{\mathrm{d}\hat{\chi}}{\mathrm{d}\eta}(\boldsymbol{\tau}), \\ \frac{\mathrm{d}\hat{\chi}}{\mathrm{d}\eta}(\boldsymbol{\tau}) &= \tilde{H}\big(\mathbf{z}(\eta^-), \boldsymbol{\zeta}(\eta), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) \\ &\quad - \tilde{H}\big(\mathbf{z}(\eta), \boldsymbol{\zeta}(\eta), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big) + \frac{\partial}{\partial \eta}\big(\hat{\chi}(\boldsymbol{\tau})\big) \\ &= \Big(\tilde{\mathbf{f}}\big(\mathbf{z}(\eta^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\eta)\big)\Big)^{\mathrm{T}}\boldsymbol{\zeta}(\eta) + \frac{\partial}{\partial \eta}\big(\hat{\chi}(\boldsymbol{\tau})\big).\end{aligned}$$

This implies that the first-order partial derivatives of $\boldsymbol{\zeta}(t)$ with respect to $\boldsymbol{\tau}$ correspond to

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}t}\Big(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t)\Big) = -\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(t)\big)^{\mathrm{T}}\Big(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t)\frac{\partial t}{\partial \boldsymbol{\tau}}\Big) \\ &\quad - \frac{\partial^2 \tilde{H}}{\partial \mathbf{z}^2}\big(\mathbf{z}(t), \boldsymbol{\zeta}(t), \mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t)\frac{\partial t}{\partial \boldsymbol{\tau}}\Big), \\ &\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\bar{t}_i^-) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\bar{t}_i^-)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\bar{t}_i) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\bar{t}_i)\frac{\partial \bar{t}_i}{\partial \boldsymbol{\tau}}, \quad i = 1, \ldots, n_s, \\ &\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t_f) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t_f)\frac{\partial t_f}{\partial \boldsymbol{\tau}} = \frac{\partial^2 \tilde{\chi}}{\partial \mathbf{z}(t_f)\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\frac{\partial t_f}{\partial \boldsymbol{\tau}} \\ &\quad + \sum_{l=1}^{T}\frac{\partial^2 \tilde{\chi}}{\partial \mathbf{z}(t_f)\partial \mathbf{z}(t_l)}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_l) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_l)\frac{\partial t_l}{\partial \boldsymbol{\tau}}\Big), \\ &\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t_k) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t_k)\frac{\partial t_k}{\partial \boldsymbol{\tau}} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(t_k^+) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(t_k^+)\frac{\partial t_k}{\partial \boldsymbol{\tau}} \\ &\quad + \frac{\partial^2 \tilde{\chi}}{\partial \mathbf{z}(t_k)\partial t_f}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\frac{\partial t_f}{\partial \boldsymbol{\tau}} \\ &\quad + \sum_{l=1}^{T}\frac{\partial^2 \tilde{\chi}}{\partial \mathbf{z}(t_k)\partial \mathbf{z}(t_l)}\big(\mathbf{z}(t_1), \ldots, \mathbf{z}(t_T), t_f\big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(t_l) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(t_l)\frac{\partial t_l}{\partial \boldsymbol{\tau}}\Big), \\ &k = 1, \ldots, T-1,\end{aligned}$$

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

$$\begin{aligned}&\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\eta^-) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\eta^-)\frac{\partial \eta}{\partial \boldsymbol{\tau}} = \frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\eta) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\eta)\frac{\partial \eta}{\partial \boldsymbol{\tau}} \\ &\quad - \frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)^{\mathrm{T}}\Big(\tilde{h}_k^{(1)}\big(\mathbf{z}(\eta^-)\big)\Big)^{-1}\frac{\partial}{\partial \boldsymbol{\tau}}\Big(\frac{\mathrm{d}\hat{\chi}}{\mathrm{d}\eta}(\boldsymbol{\tau})\Big) \\ &\quad + \frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)^{\mathrm{T}}\Big(\tilde{h}_k^{(1)}\big(\mathbf{z}(\eta^-)\big)\Big)^{-2}\frac{\mathrm{d}\hat{\chi}}{\mathrm{d}\eta}(\boldsymbol{\tau})\frac{\partial \tilde{h}_k^{(1)}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(\eta^-)\big), \\ &\frac{\partial \tilde{h}_k^{(1)}}{\partial \boldsymbol{\tau}}\big(\mathbf{z}(\eta^-)\big) = \frac{\partial \tilde{h}_k}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\eta) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\eta)\frac{\partial \eta}{\partial \boldsymbol{\tau}}\Big), \\ &\frac{\partial}{\partial \boldsymbol{\tau}}\Big(\frac{\mathrm{d}\hat{\chi}}{\mathrm{d}\eta}(\boldsymbol{\tau})\Big) = \Big(\tilde{\mathbf{f}}\big(\mathbf{z}(\eta^-)\big) - \tilde{\mathbf{f}}\big(\mathbf{z}(\eta)\big)\Big)^{\mathrm{T}}\Big(\frac{\partial \boldsymbol{\zeta}}{\partial \boldsymbol{\tau}}(\eta) + \frac{\mathrm{d}\boldsymbol{\zeta}}{\mathrm{d}t}(\eta)\frac{\partial \eta}{\partial \boldsymbol{\tau}}\Big) \\ &\quad + \boldsymbol{\zeta}(\eta)^{\mathrm{T}}\Big(\frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(\eta^-)\big) - \frac{\partial \tilde{\mathbf{f}}}{\partial \mathbf{z}}\big(\mathbf{z}(\eta)\big)\Big)\Big(\frac{\partial \mathbf{z}}{\partial \boldsymbol{\tau}}(\eta) + \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}t}(\eta)\frac{\partial \eta}{\partial \boldsymbol{\tau}}\Big).\end{aligned}$$