


# Global optimal control with the direct multiple shooting method

H. Diedam | S. Sager 

Otto-von-Guericke-Universität Magdeburg,  
Magdeburg, Germany

## Correspondence

Sebastian Sager, Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany.

Email: sager@ovgu.de

## Funding information

European Research Council (ERC), Grant/Award Number: 647573; German BMBF, Grant/Award Number: 05M2013—GOSSIP

## Summary

We propose to solve global optimal control problems with a new algorithm that is based on Bock's direct multiple shooting method. We provide conditions and numerical evidence for a significant overall runtime reduction compared to the standard single shooting approach.

## KEYWORDS

global optimization, multiple shooting, optimal control

## 1 | INTRODUCTION

We are interested in globally optimal solutions of the control problem

**Definition 1.** (Basic control problem with linear objective and nonconvex dynamics)

$$\begin{aligned} \min_q \quad & c^T x(t) \\ \text{subject to} \quad & \dot{x}(t) = f(x(t), q), \quad t \in \mathcal{T}, \\ & x(0) = s_0, \\ & x(t) \in \mathcal{X}(t), \quad t \in \mathcal{T}, \\ & q \in \mathcal{Q}, \end{aligned} \quad (1)$$

on a fixed time horizon  $\mathcal{T} = [0, t_f]$  with differential states  $x : \mathcal{T} \mapsto \mathbb{R}^{n_x}$ , a finite number of controls  $q \in \mathcal{Q} \subseteq \mathbb{R}^{n_q}$ , fixed initial values  $s_0 \in \mathbb{R}^{n_x}$ , a linear objective function  $c^T x(t_f)$  of Mayer type with nonnegative  $c \in \mathbb{R}_+^{n_x}$ , and bounded convex feasible sets  $\mathcal{Q} = [\underline{q}, \bar{q}] \subseteq \mathbb{R}^{n_q}$  for controls and  $\mathcal{X}(t) = [\underline{x}(t), \bar{x}(t)] \subseteq \mathbb{R}^{n_x}$  for states.

**Assumption 2.** (Smoothness of the control problem)

We assume that the function  $f : \mathbb{R}^{n_x \times n_q} \mapsto \mathbb{R}^{n_x}$  is sufficiently smooth. Note that Lipschitz continuity guarantees the unique existence of a solution  $x(\cdot)$  of the differential equations for fixed  $q$  by virtue of the Picard-Lindelöf theorem. As we assume unique sensitivities in the following, also the partial derivative functions  $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial q}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial q^2}, \frac{\partial^2 f}{\partial x \partial q}$  are assumed to be Lipschitz continuous. Furthermore, we assume that there exists at least one feasible  $q^* \in \mathcal{Q}$  for Equation 1.

**Remark 3.** (Generality of the control problem)

Problem 1 is general in the sense that it allows arbitrary dynamics via  $f(\cdot)$  and both control values and discretized control functions via the finite-dimensional control vector  $q$ . Yet, in the interest of a simplified notation, we keep it as simple as possible. With little to no effort, all of the following results do also apply to more general classes of control problems. We discuss these extensions in Section 6 and concentrate first on the main issue, better relaxations of nonconvex dynamics.

There are at least 4 basic classes of algorithms to find optimal solutions to problem 1: first, the global approach to solve the Hamilton-Jacobi-Bellman equations, which corresponds to dynamic programming in a discrete setting; second, indirect approaches that solve boundary value problems resulting from necessary conditions of optimality, ie, Pontryagin's maximum principle; third, reformulation and relaxation of the control problem as a generalized moment problem, ie, a linear program defined on a measure space; and fourth, direct approaches that discretize control functions  $u : \mathcal{T} \mapsto \mathbb{R}^{n_u}$  to finite decision vectors  $q \in \mathbb{R}^{n_q}$ , as already done for problem 1, and relax possible path constraints  $x(t) \in \mathcal{X}(t)$  to finitely many points  $x(t_i) \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$ , and then solve the resulting nonlinear optimization problem (NLP) numerically. In most cases, the NLPs are solved locally. The aim of this paper is to develop a method that combines advantages of the direct methods with a certificate of global optimality.

Direct methods can be further distinguished on the basis of how the solution of the differential equations is included in the optimization process. *Direct single shooting* (DSS) solves problems of type 1 via an outer loop on the optimization variables  $q \in \mathbb{R}^{n_q}$ , where function and derivative evaluations are based on an integration of the differential equations on the time horizon  $\mathcal{T}$ . In this sense, the trajectories  $x(\cdot)$  are merely dependent variables and can be seen as part of the objective and constraint functions. *Direct collocation methods* discretize the differential equations and add variables and constraints to problem 1. This leads to a large-scale, but highly structured, NLP.

*Bock's direct multiple shooting method* can be seen as a hybrid between the 2. Additional variables and constraints are added, but adaptive external integrators can still be used to solve the differential equations on subintervals.

In the interest of comparison, we use the same time grid  $0 = t_0 < t_1 < \dots < t_{n_{ms}} = t_f$  for the evaluation of constraints in single and multiple shooting. In our implementation, we use the same grid for the discretization of control functions  $u(\cdot)$ . For our theoretic considerations and in the interest of a fair comparison, although, this is already implied by using a general control vector  $q$ , compare Equation 1.

**Definition 4.** (Solution trajectories)

For given  $q \in \mathbb{R}^{n_q}$ ,  $s_i \in \mathbb{R}^{n_x}$ , and  $\tau \in \mathcal{T}$ , we denote by  $x(\tau; q, s_i)$  the solution trajectory  $x(\cdot)$  of the differential equation  $\dot{x}(t) = f(x(t), q)$  with initial value  $x(t_i) = s_i$  on the time interval  $[t_i, \tau]$ , evaluated at time  $\tau$ . In slight abuse of notation, we infer the time  $t_i$  from the index of the argument  $s_i$ . The argument  $\tau$  in front of the semicolon refers to the evaluation time; the arguments after the semicolon list implicit dependencies of  $x(\cdot)$ . We also write  $x(t; q, s_0)$  with an implicit dependence on  $s_0 \in \mathbb{R}^{n_x}$ , although we assume  $s_0$  to be fixed, to highlight the difference between single and multiple shooting.

**Definition 5.** (Single and multiple shooting NLPs)

The single shooting NLP for problem 1 is defined as

$$\begin{aligned} \min_q \quad & c^T x^{ss}(t_f; q, s_0) \\ \text{subject to} \quad & x^{ss}(t_i; q, s_0) \in \mathcal{X}_i, \quad i = 1, \dots, n_{ms}, \\ & q \in \mathcal{Q}. \end{aligned} \tag{SS}^{\mathcal{Q} \times \mathcal{X}}$$

The  $n_{ms}$  node multiple shooting NLP for problem 1 is defined as

$$\begin{aligned} \min_{s_1, \dots, s_{n_{ms}}, q} \quad & c^T s_{n_{ms}} \\ \text{subject to} \quad & s_{i+1} = x^{ms,i}(t_{i+1}; q, s_i), \quad i = 0, \dots, n_{ms} - 1 \\ & s_i \in \mathcal{X}_i, \quad i = 1, \dots, n_{ms}, \\ & q \in \mathcal{Q}, \end{aligned} \tag{MS}^{\mathcal{Q} \times \mathcal{X}}$$

where variables  $(s_1, \dots, s_{n_{ms}}) \in \mathbb{R}^{n_{ms} n_x}$  and  $n_{ms} n_x$  matching constraints  $s_{i+1} = x^{ms,i}(t_{i+1}; q, s_i)$  have been added in comparison to problem (SS) $^{\mathcal{Q} \times \mathcal{X}}$ . Note that the evaluation of  $x^{ms,i}(t_{i+1}; q, s_i)$  implies that  $x(\cdot)$  is evaluated independently on the time intervals  $[t_i, t_{i+1}]$  with different initial values  $s_i$ . We write  $\mathcal{X} = \mathcal{X}_1 \times \dots \times \mathcal{X}_{n_{ms}} \subseteq \mathbb{R}^{n_x n_{ms}}$  as the Cartesian product of all  $\mathcal{X}$ .

We focus on complete and rigorous methods for global optimization.<sup>1</sup> The survey<sup>2</sup> gives a broad overview over the literature in global optimization in general. Underlying most deterministic global optimization algorithms are interval arithmetics,<sup>3</sup> a spatial Branch&Bound technique,<sup>4,5</sup> and convex relaxations of the original problem, such as McCormick relaxations<sup>6,7</sup> or polyhedral outer approximations.<sup>8</sup> We use the  $\alpha$ BB method.<sup>9-11</sup> The convex relaxations used in the  $\alpha$ BB algorithm are based on the eigenvalues of interval Hessians that are the bounds on the second derivatives. The authors in Skjäl and Westerlund<sup>12</sup> survey different methods to determine bounds on these eigenvalues. We use a method based on Gershgorin's circle theorem,<sup>13</sup> as suggested in Adjiman et al.<sup>11</sup>

The novelty compared to previous approaches to globally optimize optimal control problems with the  $\alpha$ BB method (see a number of studies<sup>14–17</sup>) is that we work with the multiple shooting discretization ( $MS^{Q \times \mathcal{X}}$ ). The discretizations ( $SS^{Q \times \mathcal{X}}$ ) and ( $MS^{Q \times \mathcal{X}}$ ) are obviously equivalent with respect to feasibility and optimality, but lead to different algorithmic behavior, both in local and in global optimizations. We show that the multiple shooting approach ( $MS^{Q \times \mathcal{X}}$ ) has advantages compared to the single shooting approach ( $SS^{Q \times \mathcal{X}}$ ) when used in a global optimization context. Advantages of course relate to computational time, which itself depends on 2 numbers:

- *The average computational time to solve a problem on a single node.* The advantages of direct multiple shooting in comparison to DSS have been discussed at length in a number of publications, eg, a few studies.<sup>18–21</sup> The most important advantages are the possibility to parallelize the evaluation of function and derivative evaluation, to provide initial values for the trajectory, improved convergence due to a lifting effect<sup>22</sup> and to high-rank updates,<sup>23</sup> and improved stability. There are several possibilities to exploit the sparsity of the Karush-Kuhn-Tucker matrix, such that the linear algebra is not significantly more expensive than in the single shooting case.<sup>24</sup> Note that the mentioned benefits arise in the context of global optimal control on the level of the single nodes that are solved locally. For example, an initialization of the additional variables ( $s_1, \dots, s_{n_{ms}}$ ) can be performed automatically by initializing them close to a reference trajectory or using the trajectory of the father node. We are not going to dwell further on these topics in this paper, but take it as one motivation to use the method also for global optimization.
- *The overall number of nodes in a spatial Branch&Bound tree.* This number is a result of the number of variables we are branching on and of the quality of the relaxations on a single node. We show in Corollary 14 that the right branching strategy guarantees that there is no increase in the number of nodes. We show in Theorem 16 under which assumptions the quality of the convex underestimations and concave overestimations, which can be expressed in terms of the relaxation factors  $\alpha$ , is strictly better. This surprising result is mainly due to the effect that the additional variables  $s$  reset the propagated bounds on the values of states and sensitivities and hence avoid an exponential growth in time. From a different point of view, we want to quantify the relation between the  $\alpha$ BB relaxation of a composite function and the composition of several  $\alpha$ BB relaxations, which in general, do not commute. For the case of an additional end point that can be seen as a 2-point multiple shooting, this was discussed in Little et al.<sup>5</sup>

Our approach is different from the so-called Branch-and-Lift algorithm of Houska and Chachuat,<sup>25</sup> in which the control parameterization is adaptively refined. Although we also lift to a higher dimension, our lifting concerns only the state variables. A combination of both approaches should be possible and desirable, but is beyond the scope of this paper.

The paper is organized as follows. In Section 2, we survey mathematical preliminaries, in particular, a basic  $\alpha$ BB algorithm and second-order derivatives of trajectories. In Section 3, we formulate the specific relaxations for the single and multiple shooting optimization problems ( $SS^{Q \times \mathcal{X}}$ ) and ( $MS^{Q \times \mathcal{X}}$ ). In Section 4, we prove the superiority of ( $MS^{Q \times \mathcal{X}}$ ) with respect to the tightness of relaxations and the non-increase of the number of Branch&Bound nodes despite the increase of variables. In Section 5, we show exemplarily the performance gain for a benchmark problem from the literature. In Section 6, we discuss applicability to more general optimal control problems than Equation 1 and algorithmic extensions. We conclude in Section 7 with a summary.

## 2 | PRELIMINARIES

For convenience of the reader, we shortly summarize useful notation, compare Table 1, and results from global optimization and discuss the particular case of optimal control, where derivatives of the functions with respect to the controls involve the dependent differential variables.

Every twice differentiable function  $\phi : [\underline{v}, \bar{v}] \subset \mathbb{R}^{n_v} \mapsto \mathbb{R}$  can be underestimated by a convex function

$$\phi_{cv}(v) := \phi(v) + \sum_{k=1}^{n_v} \alpha_{cv,k} (\bar{v}_k - v_k)(\underline{v}_k - v_k), \quad (2)$$

and overestimated by a concave function

$$\phi_{cc}(v) := \phi(v) - \sum_{k=1}^{n_v} \alpha_{cc,k} (\bar{v}_k - v_k)(\underline{v}_k - v_k), \quad (3)$$

for any  $v$  in the bounded box  $[\underline{v}, \bar{v}]$ , if only the  $\alpha_{cv,k} \in \mathbb{R}_+$  and  $\alpha_{cc,k} \in \mathbb{R}_+$  are chosen large enough. For notational convenience, we write

$$\psi(v_k) := (\bar{v}_k - v_k)(\underline{v}_k - v_k) \quad (4)$$

**TABLE 1** The notation used in this paper

Notation	Definition	Explanation
General notation		
$\mathcal{V}$	$[\underline{v}, \bar{v}]$	Interval between lower and upper bounds $\underline{v}, \bar{v} \in \mathbb{R}^{n_v}$
$ \mathcal{V} $	$\max( \underline{v} ,  \bar{v} )$	Absolute value of an interval
$(s_0, s_1, \dots, s_m)$	-	Concatenated vector in $\mathbb{R}^n$ with $n = n_{s_0} + \dots + n_{s_m}$
$\text{sol}(NLP)$	$v^*$	Optimal solution of (NLP)
$\text{val}(NLP)$	$\phi^{\text{obj}}(v^*)$	Optimal objective function value of (NLP)
$\psi(v_k)$	See Equation 4	Convexification term $(\bar{v}_k - v_k)(\underline{v}_k - v_k)$ for variable $v_k$
$\Phi(\mathcal{V})$	$\text{val}(NLP_{\text{cv}}^{\mathcal{V}})$	Lower bound on $\text{sol}(NLP)$
$\delta_{jl}$	-	Kronecker symbol, $\delta_{jl} = 1$ if $j = l$ , $\delta_{jl} = 0$ if $j \neq l$
Superscripts and subscripts		
cv, cc	See Equations 2 and 3	Convex underestimation/concave overestimation of a function
ss, ms	Definition 5	Single shooting or multiple shooting
$i$	-	Time index, $i = 0, \dots, n_{\text{ms}}$
$j$	-	Differential state index, $j = 1, \dots, n_x$
$k, l$	-	Indices of variables $q \in \mathbb{R}^{n_q}$ or $s_i \in \mathbb{R}^{n_s}$ , $i = 1, \dots, n_{\text{ms}}$
Exemplary trajectories and coefficients, similar for different superscripts and subscripts		
$x^{\text{ms},i}(\tau; q, s_i)$	Definition 4	Solution of $\dot{x}(t) = f(x(t), q)$ , $x(t_i) = s_i$ on $[t_i, \tau]$ at $\tau$ .
$x_{j,\text{cv}}^{\text{ms},i}(t; q, s_i)$	See Equations 20 and 21	Underestimation and overestimation of trajectories $x_j^{\text{ms},i}(t; q, s_i)$
$\mathcal{H}_{q_k, s_{i_l}}^{\text{ms},i,j}(t; \mathcal{Q}, S_i)$	$\frac{d^2 x_j^{\text{ms},i}(t; \mathcal{Q}, S_i)}{dq_k ds_{i_l}}$	Second-order sensitivity interval
$\alpha_{\text{cv},k}^{\text{ms},i,j}$	See Equations 22-25	Relaxation factors

Abbreviation: NLP, nonlinear optimization problem.

in the following. As the Hessians of the new functions need to be positive semi-definite for convexity (negative semi-definite for concavity), their eigenvalues need to be nonnegative (non-positive). Necessary conditions are hence

$$\alpha_{\text{cv},k} \geq -\frac{1}{2} \min(0, \lambda_{\min}(\nabla^2 \phi(v))) , \quad (5)$$

$$\alpha_{\text{cc},k} \geq -\frac{1}{2} \min(0, -\lambda_{\max}(\nabla^2 \phi(v))) . \quad (6)$$

One cheap, yet successful, way<sup>12</sup> to calculate sufficiently large  $\alpha_{\text{cv},k}$ ,  $\alpha_{\text{cc},k}$  is based on Gershgorin's circle theorem,<sup>13</sup> which implies as follows: if a matrix is strictly diagonally dominant and all its diagonal elements are positive, then the real parts of its eigenvalues are positive; if all its diagonal elements are negative, then the real parts of its eigenvalues are negative. Hence, if for  $k = 1, \dots, n_v$  and all  $v \in [\underline{v}, \bar{v}]$

$$\alpha_{\text{cv},k} \geq -\frac{1}{2} \min \left( 0, (\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} |(\nabla^2 \phi(v))_{kl}| \right) , \quad (7)$$

$$\alpha_{\text{cc},k} \geq -\frac{1}{2} \min \left( 0, -(\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} |(\nabla^2 \phi(v))_{kl}| \right) , \quad (8)$$

then  $\phi_{\text{cv}}$  is convex, and  $\phi_{\text{cc}}$  is concave on  $[\underline{v}, \bar{v}]$ . We are interested in factors  $\alpha$  that yield positive semi-definite (negative semi-definite) interval Hessians for all  $v \in [\underline{v}, \bar{v}]$ .

**Definition 6.** (Interval arithmetics)

We denote closed intervals by calligraphic uppercase characters,

$$\mathcal{V} := [\underline{v}, \bar{v}] = \{v \in \mathbb{R} : \underline{v} \leq v \leq \bar{v}\} \quad (9)$$

with lower bound  $\underline{v} \in \mathbb{R}$  and upper bound  $\bar{v} \in \mathbb{R}$ , indicated by underlined and overlined variables. The absolute value of an interval  $\mathcal{V}$  is  $|\mathcal{V}| := \max(|\underline{v}|, |\bar{v}|)$ , and the set of closed intervals is defined by  $[\mathbb{R}] := \{[\underline{v}, \bar{v}] : \underline{v} \leq \bar{v}, \underline{v} \in \mathbb{R}, \bar{v} \in \mathbb{R}\}$ . We use

the straightforward extensions to higher dimensions, ie, to interval matrices  $A \in [\mathbb{R}]^{m \times n}$ , and to functions  $\Phi : [\mathbb{R}] \mapsto [\mathbb{R}]$ , as an extension of  $\phi : \mathbb{R} \mapsto \mathbb{R}$ . All interval arithmetics are performed componentwise and result in interval components. We refer to the textbook<sup>26</sup> for an introduction to calculus with intervals.

Making use of the obvious relations

$$\begin{aligned} \min_{v \in [\underline{v}, \bar{v}]} \left( (\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} |(\nabla^2 \phi(v))_{kl}| \right) &\geq \min_{v \in [\underline{v}, \bar{v}]} (\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} \max_{v \in [\underline{v}, \bar{v}]} |(\nabla^2 \phi(v))_{kl}| \\ \min_{v \in [\underline{v}, \bar{v}]} \left( -(\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} |(\nabla^2 \phi(v))_{kl}| \right) &\geq -\max_{v \in [\underline{v}, \bar{v}]} (\nabla^2 \phi(v))_{kk} - \sum_{l \neq k} \max_{v \in [\underline{v}, \bar{v}]} |(\nabla^2 \phi(v))_{kl}| \end{aligned}$$

we overestimate Inequalities 7 and 8 further and get for  $\mathcal{V} = [\underline{v}, \bar{v}]$  the factors

$$\alpha_{cv,k} = -\frac{1}{2} \min \left( 0, \overline{(\nabla^2 \Phi(\mathcal{V}))_{kk}} - \sum_{l \neq k} |(\nabla^2 \Phi(\mathcal{V}))_{kl}| \right), \quad (10)$$

$$\alpha_{cc,k} = -\frac{1}{2} \min \left( 0, -\overline{(\nabla^2 \Phi(\mathcal{V}))_{kk}} - \sum_{l \neq k} |(\nabla^2 \Phi(\mathcal{V}))_{kl}| \right) \quad (11)$$

with the notation introduced in Definition 6, ie,

$$|(\nabla^2 \Phi(\mathcal{V}))_{kl}| = \max \left( |(\nabla^2 \Phi(\mathcal{V}))_{kl}|, |\overline{(\nabla^2 \Phi(\mathcal{V}))_{kl}}| \right).$$

Being able to underestimate and overestimate functions allows to relax optimization problems. The optimal objective function value of a general nonconvex NLP of the form

$$\begin{aligned} \min_v \quad &\phi^{\text{obj}}(v) \\ \text{subject to} \quad & \\ &0 \geq \phi^{\text{ineq}}(v), \\ &0 = \phi^{\text{eq}}(v), \\ &v \in \mathcal{V} = [\underline{v}, \bar{v}] \end{aligned} \quad (\text{NLP}^{\mathcal{V}})$$

is underestimated by the optimal objective function value of the convex NLP

$$\begin{aligned} \min_v \quad &\phi_{cv}^{\text{obj}}(v) \\ \text{subject to} \quad & \\ &0 \geq \phi_{cv}^{\text{ineq}}(v), \\ &0 \geq \phi_{cv}^{\text{eq}}(v), \\ &0 \geq -\phi_{cc}^{\text{eq}}(v), \\ &v \in \mathcal{V} = [\underline{v}, \bar{v}], \end{aligned} \quad (\text{NLP}_{cv}^{\mathcal{V}})$$

where the subscripts cv and cc refer to the  $\alpha$ BB relaxations (Equations 2 and (3)), assumed that for all functions, the corresponding values  $\alpha \in \mathbb{R}_+^n$  have been calculated via Equations 10 and 11 and are hence sufficiently large.

**Definition 7.** (Optimal solutions)

We define  $\text{sol}(\text{NLP}) := v^*$  as a (not necessarily unique and existent) locally optimal solution of a given (NLP). In addition, we define  $\text{val}(\text{NLP}) := \phi^{\text{obj}}(v^*)$  to be the corresponding optimal objective function value (possibly infinity).

The objective function value is underestimated, while the feasible region is overestimated, resulting in the desired lower bound on the optimal objective function value:

$$\text{val}(\text{NLP}_{cv}^{\mathcal{V}}) \leq \text{val}(\text{NLP}^{\mathcal{V}}).$$

Branch&Bound methods are described in detail in Scholz.<sup>27</sup> For an early application in global optimization, we refer to Falk and Soland,<sup>28</sup> whereas our implementation is based on Horst and Tuy.<sup>29,30</sup> The extension to nonconvex constraints is discussed in Stein et al.<sup>31</sup> A basic  $\alpha$ BB algorithm for a nonconvex (NLP <sup>$\mathcal{V}$</sup> ) is given in Algorithm 1.

<p><b>Algorithm 1:</b> The basic <math>\alpha</math>BB algorithm for (2).</p> <p><b>Initialize</b> counter <math>i := 0</math>, interval <math>\mathcal{V}_0 = \mathcal{V} = [y, \bar{v}]</math> and set of open problems <math>\mathcal{P} := \{(\text{NLP}^{\mathcal{V}_0})\}</math>.</p> <p><b>Initialize</b> lower bound <math>\underline{\Phi} := \text{val}(\text{NLP}_{\text{cv}}^{\mathcal{V}_0})</math> and upper bound <math>\bar{\Phi} := \text{val}(\text{NLP}^{\mathcal{V}_0})</math>.</p> <p><b>while</b> <math>\mathcal{P} \neq \emptyset</math> and <math>\bar{\Phi} - \underline{\Phi} &gt; \epsilon</math> <b>do</b></p> <p style="padding-left: 2em;"><b>Increment</b> iteration counter <math>i := i + 1</math>.</p> <p style="padding-left: 2em;"><b>Select</b> open problem <math>(\text{NLP}^{\mathcal{V}_i}) \in \mathcal{P}</math> with feasible domain <math>\mathcal{V}_i</math>.</p> <p style="padding-left: 2em;"><b>Branch</b> <math>\mathcal{V}_i</math> into a partition <math>\mathcal{V}_i = \bigcup_{k \in \{1, \dots, N\}} \mathcal{V}_{ik}</math>.</p> <p style="padding-left: 2em;"><b>for</b> <math>k = 1</math> <b>to</b> <math>N</math> <b>do</b></p> <p style="padding-left: 4em;"><b>Solve</b> <math>(\text{NLP}^{\mathcal{V}_{ik}})</math> locally.</p> <p style="padding-left: 4em;"><b>if</b> <math>\text{val}(\text{NLP}^{\mathcal{V}_{ik}}) &lt; \bar{\Phi}</math> <b>then</b></p> <p style="padding-left: 6em;"><b>Update</b> upper bound <math>\bar{\Phi} := \text{val}(\text{NLP}^{\mathcal{V}_{ik}})</math> and corresponding <math>v^* = \text{sol}(\text{NLP}^{\mathcal{V}_{ik}})</math>.</p> <p style="padding-left: 4em;"><b>end</b></p> <p style="padding-left: 4em;"><b>Construct</b> convex relaxation <math>(\text{NLP}_{\text{cv}}^{\mathcal{V}_{ik}})</math>.</p> <p style="padding-left: 4em;"><b>Solve</b> <math>(\text{NLP}_{\text{cv}}^{\mathcal{V}_{ik}})</math> locally (= globally).</p> <p style="padding-left: 2em;"><b>end</b></p> <p style="padding-left: 2em;"><b>Update</b> open problems</p> <p style="padding-left: 4em;"><math>\mathcal{P} := \mathcal{P} \setminus (\text{NLP}^{\mathcal{V}_i}) \bigcup_{k \in \{1, \dots, N\}} (\text{NLP}^{\mathcal{V}_{ik}})</math>.</p> <p style="padding-left: 2em;"><b>Update</b> open problems <math>\mathcal{P} := \mathcal{P} \setminus (\text{NLP}^{\bar{\mathcal{V}}})</math> for all <math>(\text{NLP}^{\bar{\mathcal{V}}}) \in \mathcal{P}</math> with <math>\bar{\Phi} \leq \text{val}(\text{NLP}_{\text{cv}}^{\bar{\mathcal{V}}})</math>.</p> <p style="padding-left: 2em;"><b>Update</b> global lower bound</p> <p style="padding-left: 4em;"><math>\underline{\Phi} := \min_{(\text{NLP}^{\bar{\mathcal{V}}}) \in \mathcal{P}} \text{val}(\text{NLP}^{\bar{\mathcal{V}}})</math>.</p> <p><b>end</b></p>
---

To construct the convex relaxations  $(\text{NLP}_{\text{cv}}^{\mathcal{V}_{ik}})$  for the special cases of single and multiple shooting, compare Section 3, we need the relaxation factors  $\alpha$ . To calculate them for the discretized control problems  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$  and  $(\text{MS}^{\mathcal{Q} \times \mathcal{X}})$ , we need the second derivatives of the occurring functions with respect to the decision variables, compare Equation 11, in particular, of the control trajectories  $x(\cdot)$  with respect to  $q$  for  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$  and with respect to  $(q, s)$  for  $(\text{MS}^{\mathcal{Q} \times \mathcal{X}})$ .

**Definition 8.** (Second-order sensitivities)

Let  $j \in \{1, \dots, n_x\}$  be fix and a bounded box  $\mathcal{Q} \subseteq \mathbb{R}^{n_q}$  be given. We define the single shooting second-order sensitivity interval of the trajectory  $x_j^{\text{ss}}(\cdot)$  with respect to  $q \in \mathbb{R}^{n_q}$ , scaled by the linear objective function vector  $c \in \mathbb{R}^{n_x}$ , as

$$\mathcal{H}_{q_k, q_l}^{\text{ss}, j}(t; \mathcal{Q}, s_0) := c_j \frac{d^2 x_j^{\text{ss}}(t; \mathcal{Q}, s_0)}{dq_k dq_l} \subseteq \mathbb{R}, \quad k, l \in \{1, \dots, n_q\}. \quad (12)$$

Let in addition  $i \in \{1, \dots, n_{\text{ms}}\}$  and a bounded box  $\mathcal{S}_i \subseteq \mathcal{X}_i \subseteq \mathbb{R}^{n_x}$  be given. We define the multiple shooting second-order sensitivity interval of the trajectory  $x_j^{\text{ms}, i}(\cdot)$  with respect to  $q \in \mathcal{Q}$  and  $s_i \in \mathcal{S}_i$  for  $t \in [t_i, t_{i+1}]$ , scaled by the linear objective function vector  $c \in \mathbb{R}^{n_x}$ , as

$$\mathcal{H}_{q_k, q_l}^{\text{ms}, i, j}(t; \mathcal{Q}, \mathcal{S}_i) := c_j \frac{d^2 x_j^{\text{ms}, i}(t; \mathcal{Q}, \mathcal{S}_i)}{dq_k dq_l} \subseteq \mathbb{R}, \quad k, l \in \{1, \dots, n_q\} \quad (13)$$

$$\mathcal{H}_{s_{ik}, s_{il}}^{\text{ms}, i, j}(t; \mathcal{Q}, \mathcal{S}_i) := c_j \frac{d^2 x_j^{\text{ms}, i}(t; \mathcal{Q}, \mathcal{S}_i)}{ds_{ik} ds_{il}} \subseteq \mathbb{R}, \quad k, l \in \{1, \dots, n_x\} \quad (14)$$

$$\mathcal{H}_{q_k, s_{il}}^{\text{ms}, ij}(t; \mathcal{Q}, \mathcal{S}_i) := c_j \frac{d^2 x_j^{\text{ms}, i}(t; \mathcal{Q}, \mathcal{S}_i)}{dq_k ds_{il}} \subseteq \mathbb{R}, \quad k \in \{1, \dots, n_q\}, \quad (15)$$

$$l \in \{1, \dots, n_x\}.$$

We obtain the functions  $h_{\cdot}^{ij}(\cdot; q, s_i)$  and their interval counterparts  $\mathcal{H}_{\cdot}^{ij}(\cdot; \mathcal{Q}, \mathcal{S}_i)$  as solution trajectories of a system of differential equations. See Lemma 19 in the Appendix.

### 3 | CONVEX RELAXATIONS OF SHOOTING PROBLEMS

In this section, we formulate relaxed, convex versions of  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$  and  $(\text{MS}^{\mathcal{Q} \times \mathcal{X}})$  in the spirit of  $(\text{NLP}_{\text{cv}}^{\mathcal{V}})$  and show how Algorithm 1 can be applied.

#### 3.1 | Single shooting

**Definition 9.** (Convexification of single shooting NLP)

For given data from problem  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$  as a discretization of problem 1, the convexified single shooting NLP is defined as

$$\begin{aligned} & \min_q \quad c^T x_{\text{cv}}^{\text{ss}}(t; q, s_0, \mathcal{Q}) \\ & \text{subject to} \\ & \quad x_{\text{cv}}^{\text{ss}}(t; q, s_0, \mathcal{Q}) \leq \bar{x}_i, \quad i = 1, \dots, n_{\text{ms}}, \\ & \quad x_{\text{cc}}^{\text{ss}}(t; q, s_0, \mathcal{Q}) \geq \underline{x}_i, \quad i = 1, \dots, n_{\text{ms}}, \\ & \quad q \in \underline{\mathcal{Q}}. \end{aligned} \quad (\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}})$$

The functions  $x_{\text{cv}}^{\text{ss}}(t; q, s_0, \mathcal{Q})$  and  $x_{\text{cc}}^{\text{ss}}(t; q, s_0, \mathcal{Q})$  are given as in Equations 2 and 3, more precisely for the only independent variables  $q \in \mathcal{Q}$  and for all entries  $j = 1, \dots, n_x$  as

$$x_{j, \text{cv}}^{\text{ss}}(t; q, s_0, \mathcal{Q}) := x_j^{\text{ss}}(t; q, s_0) + \sum_{k=1}^{n_q} \alpha_{\text{cv}, k}^{\text{ss}, ij} \psi(q_k), \quad (16)$$

$$x_{j, \text{cc}}^{\text{ss}}(t; q, s_0, \mathcal{Q}) := x_j^{\text{ss}}(t; q, s_0) - \sum_{k=1}^{n_q} \alpha_{\text{cc}, k}^{\text{ss}, ij} \psi(q_k), \quad (17)$$

where  $x^{\text{ss}}(\cdot; q, s_0)$  is the solution trajectory of  $\dot{x}(t) = f(x(t), q)$  with  $x(0) = s_0$  on  $[0, t_f]$ . The relaxation factors  $\alpha_{\text{cv}}^{\text{ss}}, \alpha_{\text{cc}}^{\text{ss}} \in \mathbb{R}^{n_{\text{ms}} n_x n_q}$  are determined on the basis of Equations 10 and 11 and Definition 8 as

$$\alpha_{\text{cv}, k}^{\text{ss}, ij} = -\frac{1}{2} \min \left( 0, \mathcal{H}_{q_k, q_k}^{\text{ss}, j}(t_i; \mathcal{Q}, s_0) - \sum_{l \neq k} \left| \mathcal{H}_{q_k, q_l}^{\text{ss}, j}(t_i; \mathcal{Q}, s_0) \right| \right), \quad (18)$$

$$\alpha_{\text{cc}, k}^{\text{ss}, ij} = -\frac{1}{2} \min \left( 0, -\overline{\mathcal{H}_{q_k, q_k}^{\text{ss}, j}(t_i; \mathcal{Q}, s_0)} + \sum_{l \neq k} \left| \mathcal{H}_{q_k, q_l}^{\text{ss}, j}(t_i; \mathcal{Q}, s_0) \right| \right), \quad (19)$$

for  $k = 1, \dots, n_q$  as functions of the box  $\mathcal{Q}$ .

**Lemma 10.** (Convex single shooting relaxation)

Problem  $(\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}})$  is a convex relaxation of problem  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$ , and for  $q^* = \text{sol}(\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}})$ , we have

$$\text{val}(\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}}) = c^T x^{\text{ss}}(t_i; q^*, s_0) + \sum_{j=1}^{n_x} \sum_{k=1}^{n_q} c_j \alpha_{\text{cv}, k}^{\text{ss}, ij} (\bar{q}_k - q_k^*)(q_k - q_k^*).$$

**Proof.** Constraints and objective function of problem  $(\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}})$  are convex by construction. Assume  $\hat{q} \in \mathbb{R}^{n_q}$  is feasible for problem  $(\text{SS}^{\mathcal{Q} \times \mathcal{X}})$ . Because of

$$\begin{aligned} x_{\text{cc}}^{\text{ss}}(t_i; \hat{q}, s_0, \mathcal{Q}) &\geq x^{\text{ss}}(t_i; \hat{q}, s_0) \geq \underline{x}_i \quad \text{and} \\ x_{\text{cv}}^{\text{ss}}(t_i; \hat{q}, s_0, \mathcal{Q}) &\leq x^{\text{ss}}(t_i; \hat{q}, s_0) \leq \bar{x}_i, \end{aligned}$$

it is also feasible for  $(\text{SS}_{\text{cv}}^{\mathcal{Q} \times \mathcal{X}})$ .



The objective function  $c^T x$  is assumed to be linear and is hence convex. Using the convex underestimation for the trajectory, and  $c \in \mathbb{R}_+^{n_x}$  nonnegative, we obtain

$$c^T x_{cv}^{ss}(t_f; \hat{q}, s_0, \mathcal{Q}) \leq c^T x^{ss}(t_f; \hat{q}, s_0)$$

for all feasible  $\hat{q} \in \mathbb{R}^{n_q}$ . Thus, problem  $(SS_{cv}^{Q \times X})$  is a convex relaxation of problem  $(SS^{Q \times X})$ .

The objective function value is bounded because of the boundedness of  $\mathcal{Q}$  and exists because of Assumption 2. It can be calculated as

$$\begin{aligned} & \text{val}(SS_{cv}^{Q \times X}) \\ &= c^T x_{cv}^{ss}(t_f; q^*, s_0, \mathcal{Q}) \\ &= \sum_{j=1}^{n_x} c_j \left( x_j^{ss}(t_f; q^*, s_0) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ss,i,j} (\bar{q}_k - q_k^*)(\underline{q}_k - q_k^*) \right), \end{aligned}$$

□

which concludes the proof.

We apply Algorithm 1 to  $(SS^{Q \times X})$  by taking  $\mathcal{V} = \mathcal{Q}$  and  $NLP^{\mathcal{V}} = (SS^{Q \times X})$ . The construction of the convex relaxation is described in Algorithm 2.

With *Update bounds*, we refer to taking the minimum of available upper bounds (maximum of lower bounds). For single shooting, there are 2 qualitatively different kinds of bounds  $\underline{x}_i$  and  $\bar{x}_i$ . First, there may be a priori bounds  $\underline{x}_i, \bar{x}_i$  specified in the original problem formulation  $(MS^{Q \times X})$  or passed on from a father node in a Branch&Bound tree (thus the recursive use). Second, the bounds  $\underline{x}_i, \bar{x}_i$  may result from a forward propagation of the set  $\mathcal{Q} = [\underline{q}, \bar{q}]$ , ie,  $\bar{x}_{ij} = \bar{x}_j^{ss}(t_i; \mathcal{Q}, s_0)$  or  $\underline{x}_{ij} = \underline{x}_j^{ss}(t_i; \mathcal{Q}, s_0)$ . There are different ways to calculate  $\underline{x}_j^{ss}(t_i; \mathcal{Q}, s_0)$  and  $\bar{x}_j^{ss}(t_i; \mathcal{Q}, s_0)$ , and it is well known that a priori bounds of the first type can be used to strengthen bounds of the second type, eg, Scott et al.<sup>7</sup> We use Taylor integration as a special case of validated integrators, see a number of studies.<sup>32-35</sup> We use it to calculate both, propagated states  $x_j^{ss}(t; \mathcal{Q}, s_0)$  and propagated second-order sensitivities  $\mathcal{H}_{q_k, q_l}^{ss,j}(t; \mathcal{Q}, s_0)$ . Therefore, there are no additional costs involved, as the propagated states are required for the sensitivities, anyways. Using bounds of the second type is not strictly necessary, but it may significantly speed up the solution process.

**Algorithm 2:** Construction of (9) for  $q^* = \text{sol}(5)$ .

**Solve**  $\dot{x}(t) = f(x(t), q^*)$  with  $x(0) = s_0$  on  $[0, t_f]$  to get  $x^{ss}(t_i; q^*, s_0)$  for  $i = 1, \dots, n_{ms}$ .

**Solve** differential equations and get all  $\mathcal{H}_{q_k, q_l}^{ss,j}(t_i; \mathcal{Q}, s_0)$  using (12) and Lemma 19.

**Calculate** relaxation factors  $\alpha_{cv}^{ss}, \alpha_{cc}^{ss} \in \mathbb{R}^{n_{ms} n_x n_q}$  using (18–19).

**Calculate** all  $x_{j,cv}^{ss}(t_i; q^*, s_0, \mathcal{Q})$  and  $x_{j,cc}^{ss}(t_i; q^*, s_0, \mathcal{Q})$  using (16–17).

**Update** all upper bounds:  $\bar{x}_{ij} = \min \left\{ \bar{x}_{ij}, \bar{x}_j^{ss}(t_i; \mathcal{Q}, s_0) \right\}$ .

**Update** all lower bounds:  $\underline{x}_{ij} = \max \left\{ \underline{x}_{ij}, \underline{x}_j^{ss}(t_i; \mathcal{Q}, s_0) \right\}$ .

### 3.2 | Multiple shooting

**Definition 11.** (Convexification of multiple shooting NLPs)

For given data from problem  $(MS^{Q \times X})$  as a discretization of problem 1, the convexified  $n_{ms}$  node multiple shooting NLP is defined as

$$\begin{aligned} & \min_{s_1, \dots, s_{n_{ms}}, q} c^T s_{n_{ms}} \\ & \text{subject to} \\ & s_i \geq x_{cv}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, \mathcal{S}_{i-1}), \quad i = 1, \dots, n_{ms} \\ & s_i \leq x_{cc}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, \mathcal{S}_{i-1}), \quad i = 1, \dots, n_{ms} \\ & s_i \in \mathcal{S}_i := [\underline{x}_i, \bar{x}_i], \quad i = 1, \dots, n_{ms}, \\ & q \in \mathcal{Q}, \end{aligned} \tag{MS_{cv}^{Q \times X}}$$



with  $S_0 = \{s_0\}$  for ease of notation. The trajectories  $x_{cv}^{ms,i}(\cdot; q, s_i, \mathcal{Q}, \mathcal{S}_i)$  and  $x_{cc}^{ms,i}(\cdot; q, s_i, \mathcal{Q}, \mathcal{S}_i)$  are again given as in Equations 2 and 3, now for the independent variables  $q \in \mathcal{Q}$  and the shooting variables  $s_i \in \mathcal{S}_i, i = 1, \dots, n_{ms}$ . For all entries  $j = 1, \dots, n_x$  and  $i = 0, \dots, n_{ms} - 1$ , we define

$$x_{j,cv}^{ms,i}(t_{i+1}; q, s_i, \mathcal{Q}, \mathcal{S}_i) := x_j^{ms,i}(t_{i+1}; q, s_i) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,i,j} \psi(q_k) + \sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,i,j} \psi(s_{il}), \quad (20)$$

$$x_{j,cc}^{ms,i}(t_{i+1}; q, s_i, \mathcal{Q}, \mathcal{S}_i) := x_j^{ms,i}(t_{i+1}; q, s_i) - \sum_{k=1}^{n_q} \alpha_{cc,k}^{ms,i,j} \psi(q_k) - \sum_{l=1}^{n_x} \alpha_{cc,n_q+l}^{ms,i,j} \psi(s_{il}), \quad (21)$$

where  $x^{ms,i}(\cdot; q, s_0)$  is the solution trajectory of  $\dot{x} = f(x(t), q)$  with  $x(t_i) = s_i$  on  $[t_i, t_{i+1}]$ . The relaxation factors  $\alpha_{cv}^{ms,i}, \alpha_{cc}^{ms,i} \in \mathbb{R}^{n_x(n_q+n_x)}$  are determined for  $i = 1, \dots, n_{ms} - 1$  on the basis of Equations 10 and 11 and Definition 8. Note that they are functions of  $\mathcal{Q}$  and  $\mathcal{S}_i$ , but not of  $q$  and  $s_i$ . This follows from the dependencies of the second-order sensitivities  $\mathcal{H}_{\cdot}^{ms,i,j}(t_{i+1}) := \mathcal{H}_{\cdot}^{ms,i,j}(t_{i+1}; \mathcal{Q}, \mathcal{S}_i)$ , although we omit this important implicit dependency on  $(\mathcal{Q}, \mathcal{S}_i)$  for a more compact notation:

$$\alpha_{cv,k}^{ms,i,j} := -\frac{1}{2} \min \left( 0, \overline{\mathcal{H}_{q_k, q_k}^{ms,i,j}(t_{i+1})} - \sum_{l \neq k} \left| \mathcal{H}_{q_k, q_l}^{ms,i,j}(t_{i+1}) \right| - \sum_{l=1}^{n_x} \left| \mathcal{H}_{q_k, s_{il}}^{ms,i,j}(t_{i+1}) \right| \right), \quad (22)$$

$$\alpha_{cc,k}^{ms,i,j} := -\frac{1}{2} \min \left( 0, -\overline{\mathcal{H}_{q_k, q_k}^{ms,i,j}(t_{i+1})} + \sum_{l \neq k} \left| \mathcal{H}_{q_k, q_l}^{ms,i,j}(t_{i+1}) \right| + \sum_{l=1}^{n_x} \left| \mathcal{H}_{q_k, s_{il}}^{ms,i,j}(t_{i+1}) \right| \right) \quad (23)$$

for  $k = 1, \dots, n_q$  and

$$\alpha_{cv,n_q+l}^{ms,i,j} := -\frac{1}{2} \min \left( 0, \overline{\mathcal{H}_{s_{il}, s_{il}}^{ms,i,j}(t_{i+1})} - \sum_{k=1}^{n_q} \left| \mathcal{H}_{q_k, s_{il}}^{ms,i,j}(t_{i+1}) \right| - \sum_{k \neq l} \left| \mathcal{H}_{s_{ik}, s_{il}}^{ms,i,j}(t_{i+1}) \right| \right), \quad (24)$$

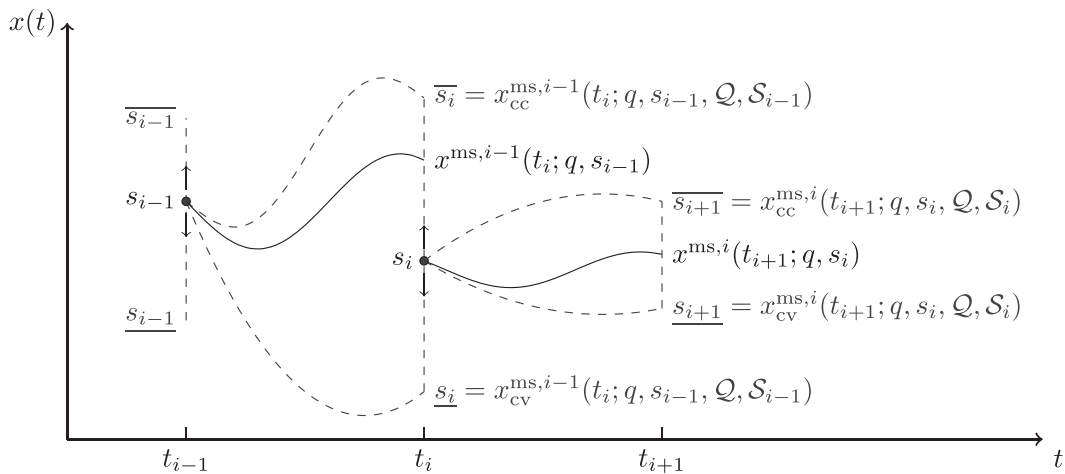
$$\alpha_{cc,n_q+l}^{ms,i,j} := -\frac{1}{2} \min \left( 0, -\overline{\mathcal{H}_{s_{il}, s_{il}}^{ms,i,j}(t_{i+1})} + \sum_{k=1}^{n_q} \left| \mathcal{H}_{q_k, s_{il}}^{ms,i,j}(t_{i+1}) \right| + \sum_{k \neq l} \left| \mathcal{H}_{s_{ik}, s_{il}}^{ms,i,j}(t_{i+1}) \right| \right) \quad (25)$$

for  $l = 1, \dots, n_x$ . We have  $\alpha_{cv,n_q+l}^{ms,i,j} = \alpha_{cc,n_q+l}^{ms,i,j} = 0$  for the special case  $i = 0$ , because  $s_0$  is assumed to be fixed and not a variable. Note that for the variable  $s_{n_{ms}}$ , no relaxation is defined, as it only enters linearly into the problem (it is not an initial value).

Figure 1 gives a 1-dimensional illustration of the relaxation concept. We show that in general, we obtain a convex relaxation.

**Lemma 12.** (Convex multiple shooting relaxation)

Problem  $(MS_{cv}^{Q \times X})$  is a convex relaxation of problem  $(MS^{Q \times X})$ .



**FIGURE 1** One-dimensional illustration of the relaxed matching conditions in problem  $(MS_{cv}^{Q \times X})$ . The multiple shooting variable  $s_i$  does not need to be equal to  $x^{ms,i-1}(t_i; q, s_{i-1})$ , but to be within  $[s_i, \bar{s}_i]$ . Note that for the nonlinear optimization problem, we only needed to define the underestimations and overestimations pointwise in time, but that a generalization to the visualized dashed trajectories is straightforward. They correspond to an evaluation of the values of  $\alpha$  (and hence of the second-order sensitivities) at times  $t \in [t_i, t_{i+1}]$ . By construction, the trajectory  $x_{cv}^{ms,i}(t; q, s_i, \mathcal{Q}, \mathcal{S}_i)$  gives a pointwise lower bound on  $x^{ms,i}(t; q, s_i)$  for all times  $t \in [t_i, t_{i+1}]$ , but is itself not necessarily monotone with respect to  $t$  [Colour figure can be viewed at [wileyonlinelibrary.com](http://wileyonlinelibrary.com)]

**Proof.** Convexity of problem  $(\text{MS}_{\text{cv}}^{\text{Q}\times\mathcal{X}})$  follows from the constructed convexity of the functions in the objective and in the lesser equal inequalities. Note that the equality constraint in  $(\text{MS}^{\text{Q}\times\mathcal{X}})$  is formulated as 2 inequalities in  $(\text{MS}_{\text{cv}}^{\text{Q}\times\mathcal{X}})$ .

However, a closer comparison between the specific relaxation Equations 20 and 21 and the general approach Equations 2 and 3 reveals that all terms involving the variables  $s_m$  with  $m = 1, \dots, n_{\text{ms}}, m \neq i$  are missing, which needs justification: The trajectories  $x_j^{\text{ms},i}(\cdot; q, s_i)$  do not depend on  $s_m$  (only the constraint depends linearly on  $s_{i+1}$ ), and hence, also the second-order derivatives are zero, which carries over to the corresponding  $\alpha$  values.

It is left to show that the convex problem  $(\text{MS}_{\text{cv}}^{\text{Q}\times\mathcal{X}})$  is a relaxation of  $(\text{MS}^{\text{Q}\times\mathcal{X}})$ . Assume  $(\hat{q}, \hat{s}) \in \mathbb{R}^{n_q + n_{\text{ms}} n_x}$  is a feasible solution of  $(\text{MS}^{\text{Q}\times\mathcal{X}})$ . Because of

$$\begin{aligned} x_{\text{cv}}^{\text{ms},i}(t_{i+1}; \hat{q}, \hat{s}_i, \mathcal{Q}, \mathcal{S}_i) &\leq x^{\text{ms},i}(t_{i+1}; \hat{q}, \hat{s}_i) = \hat{s}_{i+1}, \\ x_{\text{cc}}^{\text{ms},i}(t_{i+1}; \hat{q}, \hat{s}_i, \mathcal{Q}, \mathcal{S}_i) &\geq x^{\text{ms},i}(t_{i+1}; \hat{q}, \hat{s}_i) = \hat{s}_{i+1}, \end{aligned}$$

it is also feasible for  $(\text{MS}_{\text{cv}}^{\text{Q}\times\mathcal{X}})$ . The objective function  $c^T \hat{s}_{n_{\text{ms}}}$  is not affected by the convexification because of the assumed linearity.  $\square$

**Algorithm 3:** Construction of (11) for  $(q^*, s^*) = \text{sol}$  (5).

```

for  $i = 1$  to  $n_{\text{ms}}$  do
  Solve  $\dot{x}(t) = f(x(t), q^*), x(t_{i-1}) = s_{i-1}^*$  on  $[t_{i-1}, t_i]$  to
  get  $x^{\text{ms}}(t_i; q^*, s_{i-1}^*)$ .
  Solve differential equations and get all
   $\mathcal{H}_{\cdot, \cdot}^{\text{ms},j}(t_i; \mathcal{Q}, \mathcal{S}_{i-1})$  using (13–15) and Lemma 19.
  Calculate relaxation factors
   $\alpha_c v^{\text{ms},i}, \alpha_c c^{\text{ms},i} \in \mathbb{R}^{n_x(n_x + n_q)}$  using (22–23).
  Calculate all  $x_{j,\text{cv}}^{\text{ms}}(t_i; q^*, s_{i-1}^*, \mathcal{Q}, \mathcal{S}_{i-1})$  and
   $x_{j,\text{cc}}^{\text{ms}}(t_i; q^*, s_{i-1}^*, \mathcal{Q}, \mathcal{S}_{i-1})$  using (20–21).
  Update all upper bounds:  $\bar{x}_{ij} =$ 
   $\min \{ \bar{x}_{ij}, x_j^{\text{ms}}(t_i; \mathcal{Q}, \mathcal{S}_{i-1}), x_{j,\text{cc}}^{\text{ms}}(t_i; q^*, s_{i-1}^*, \mathcal{Q}, \mathcal{S}_{i-1}) \}$ .
  Update all lower bounds:  $\underline{x}_{ij} =$ 
   $\max \{ \underline{x}_{ij}, x_j^{\text{ms}}(t_i; \mathcal{Q}, \mathcal{S}_{i-1}), x_{j,\text{cv}}^{\text{ms}}(t_i; q^*, s_{i-1}^*, \mathcal{Q}, \mathcal{S}_{i-1}) \}$ .
  Update  $\mathcal{S}$  via  $\underline{s} = \underline{x}$  and  $\bar{s} = \bar{x}$ .
end

```

To apply Algorithm 1 to  $(\text{MS}^{\text{Q}\times\mathcal{X}})$ , we set  $\mathcal{V} = \mathcal{Q} \times \mathcal{S}$  and  $\text{NLP}^{\mathcal{V}} = (\text{MS}^{\text{Q}\times\mathcal{X}})$ . The construction of the convex relaxation in Algorithm 3 is similar to Algorithm 2, but the evaluations are performed per interval, and the bounds  $\underline{x}, \bar{x}$  on the multiple shooting variables  $s$  can potentially be additionally tightened by making use of the  $\alpha$ BB relaxations. Whereas the first 2 arguments of the upper bound minimization are identical to the single shooting case (a priori bounds from the original formulation or the father node, and set propagation), the third argument (use of  $\alpha$ BB relaxations) comes as an additional possible tightening. This definition makes the 2 inequalities in problem  $(\text{MS}_{\text{cv}}^{\text{Q}\times\mathcal{X}})$  redundant. We keep them anyhow as this facilitates the upcoming analysis.

## 4 | THEORETICAL RESULTS

It may seem at first sight that the introduction of additional variables in the multiple shooting approach is not a good idea, as it might increase the number of nodes in Algorithm 1. We give a counterargument to this misconception. Let us first look at the limit case of branching on  $\mathcal{Q}$ . Assume that we have  $\underline{q} = \bar{q} = \hat{q}$  for some  $\hat{q} \in \mathbb{R}^{n_q}$ . With the fixed initial value, we have  $\underline{s}_0 = \bar{s}_0 = s_0$ . Therefore,

$$x_{j,\text{cv}}^{\text{ms}}(t_1; \hat{q}, s_0, [\underline{q}, \bar{q}], [\underline{s}_0, \bar{s}_0]) = x_{j,\text{cc}}^{\text{ms}}(t_1; \hat{q}, s_0, [\underline{q}, \bar{q}], [\underline{s}_0, \bar{s}_0])$$

for all  $j = 1, \dots, n_x$ . Updating the bounds  $\underline{x}_1, \bar{x}_1$  in Algorithm 3 yields

$$\underline{s}_1 = \bar{s}_1 = x_{j,\text{cv}}^{\text{ms}}(t_1; \hat{q}, s_0, [\underline{q}, \bar{q}], [\underline{s}_0, \bar{s}_0]),$$

and the argument can be repeated for all  $s_i, i = 2, \dots, n_{ms}$ . With this limit case and the monotonicity of interval arithmetics, the standard convergence discussion for Algorithm 1 with respect to  $q$ , compare Papamichail and Adjiman,<sup>17</sup> carries over to convergence with respect to  $(q, s)$ . The convergence rate of the bound tightening can be estimated using the following theorem.

**Theorem 13.** (Fourth-order convergence)

There exists a constant  $C \in \mathbb{R}_+$  such that for all feasible solutions  $(q, s)$  of  $(MS_{cv}^{Q \times X})$  with given bounds  $\underline{q}, \bar{q}, \underline{x}, \bar{x}$ , to which Algorithm 3 is applied to calculate values of  $\alpha$  and updated bounds  $\underline{s}, \bar{s}$ ,

$$0 \geq \sum_{j=1}^{n_x} \psi(s_{ij}) \geq -\frac{1}{4} \|\bar{s}_i - \underline{s}_i\|^2 \geq -\mathcal{O}\left(\|\bar{q} - \underline{q}\|^4\right) \quad (26)$$

for all  $i = 1, \dots, n_{ms}$ .

**Proof.** We will use

$$\phi(v) = (\bar{v} - v)(\underline{v} - v) \geq -\frac{1}{4}(\bar{v} - \underline{v})^2,$$

the update in Algorithm 3 that gives bounds on  $\bar{s}_i$  and  $\underline{s}_i$ , the definitions (Equations 20 and (21)), and the fact that all  $\alpha$  are continuous functions of the bounds  $\underline{q}, \bar{q}, \underline{s}, \bar{s}$ , and hence have a maximum on this bounded set. We derive for arbitrary  $i = 1, \dots, n_{ms}$ ,

$$\begin{aligned} 0 &\geq \sum_{j=1}^{n_x} (\bar{s}_{ij} - s_{ij})(\underline{s}_{ij} - s_{ij}) \geq -\sum_{j=1}^{n_x} \frac{1}{4} (\bar{s}_{ij} - \underline{s}_{ij})^2 \\ &\geq -\frac{1}{4} \sum_{j=1}^{n_x} \left( x_{j,cc}^{ms}(t_i; q, s_{i-1}, Q, S_{i-1}) - x_{j,cv}^{ms}(t_i; q, s_{i-1}, Q, S_{i-1}) \right)^2 \\ &= -\frac{1}{4} \sum_{j=1}^{n_x} \left( \sum_{k=1}^{n_q} (\alpha_{cc,k}^{ms,i-1j} + \alpha_{cv,k}^{ms,i-1j}) \psi(q_k) + \sum_{l=1}^{n_x} (\alpha_{cc,n_q+l}^{ms,i-1j} + \alpha_{cv,n_q+l}^{ms,i-1j}) \psi(s_{i-1l}) \right)^2 \\ &\geq -C \left( \sum_{k=1}^{n_q} (\bar{q}_k - \underline{q}_k)^2 + \sum_{l=1}^{n_x} \psi(s_{i-1l}) \right)^2 \end{aligned}$$

We can use this for the induction base case ( $i = 1$ ), as from the fixed initial values  $s_0 = \underline{s}_0 = \bar{s}_0$  follows  $\psi(s_{0l}) = 0$  and thus the claim with the constant  $C = \frac{1}{4} n_x \max\{\alpha_{cc,k}^{ms,0j} + \alpha_{cv,k}^{ms,0j}\}$ , where the maximum is taken over the whole domain  $[q, \bar{q}]$  and for all  $j = 1, \dots, n_x, k = 1, \dots, n_q$ .

We can also use it for the inductive step by replacing  $\sum_{l=1}^{n_x} \psi(s_{i-1l})$  using Equation 26, which yields only more and more additional higher order terms as  $i$  increases.  $\square$

**Corollary 14.** (No need to branch on  $s$ )

Algorithm 1, using Algorithm 3 and branching exclusively on the set  $Q$  of  $\mathcal{V} = Q \times S$ , converges under the usual assumptions of the  $\alpha BB$  method.

This implies that no additional nodes in the Branch&Bound tree need to be considered because of the increased number of variables, if we only branch on  $q$ . It is an open question if it can even be beneficial to branch on the variables  $s$ .

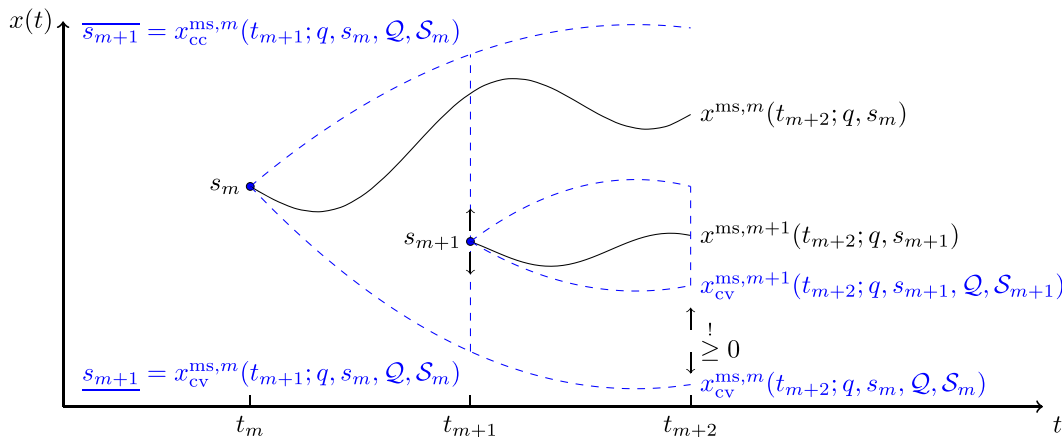
We now look at the question whether the quality of relaxations on a specific node is different between the single shooting and multiple shooting approach. We proceed as follows: we extract the main difficulty into the following assumption and then prove the main theorem of the paper in a rather straightforward way. Afterwards, we will discuss the following Assumption 15 that is conceptually visualized in Figure 2 in detail.

**Assumption 15.** (Basic assumption on dynamics and shooting grid)

Let  $(q^*, s^*)$  be an optimal solution for problem  $(MS_{cv}^{Q \times X})$ . Assume that for all  $0 \leq m < n_{ms}$  it holds

$$x_{cv}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, Q, S_{m+1}) \geq x_{cv}^{ms,m}(t_{m+2}; q^*, s_m^*, Q, S_m), \quad (27)$$

$$x_{cc}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, Q, S_{m+1}) \leq x_{cc}^{ms,m}(t_{m+2}; q^*, s_m^*, Q, S_m). \quad (28)$$



**FIGURE 2** One-dimensional illustration of Assumption 15 [Colour figure can be viewed at wileyonlinelibrary.com]

**Theorem 16.** (Multiple shooting gives better relaxations)

Let a control problem be given as in Definition 1 that obeys the smoothness Assumption 2, with convex single shooting and multiple shooting relaxations as in Definitions 9 and 11. Let for the multiple shooting grid, and for all subgrids that are obtained by removing an arbitrary number of points and thus reducing  $n_{ms}$ , Assumption 15 holds. Then multiple shooting provides a tighter relaxation, ie,

$$\text{val}(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}}) \geq \text{val}(\text{SS}_{cv}^{\mathcal{Q} \times \mathcal{X}}).$$

**Proof.** The proof is by induction over the number of multiple shooting nodes.

We consider first the case  $n_{ms} = 1$ . Let  $(q^*, s_1^*)$  be the optimal solution of  $(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}})$ . Because of

$$x_{cv}^{ss}(t_1; q^*, s_0, \mathcal{Q}) = x_{cv}^{ms,0}(t_1; q^*, s_0, \mathcal{Q}) \leq s_1^* \leq \bar{s}_1 = \bar{x}_1 \tag{29}$$

$$x_{cc}^{ss}(t_1; q^*, s_0, \mathcal{Q}) = x_{cc}^{ms,0}(t_1; q^*, s_0, \mathcal{Q}) \geq s_1^* \geq \underline{s}_1 = \underline{x}_1, \tag{30}$$

$q^*$  is a feasible solution of  $(\text{SS}_{cv}^{\mathcal{Q} \times \mathcal{X}})$ . The variable  $s_1$  enters the linear objective with  $c \in \mathbb{R}_+^{n_x}$  and is hence chosen componentwise as small as possible in an optimum,  $s_1^* = x_{cv}^{ms,0}(t_1; q^*, s_0, \mathcal{Q})$ . Therefore,

$$\text{val}(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}}) = c^T s_1^* = c^T x_{cv}^{ss}(t_1; q^*, s_0, \mathcal{Q}) = \text{val}(\text{SS}_{cv}^{\mathcal{Q} \times \mathcal{X}}).$$

Let now  $n_{ms} \geq 2$  and assume that the claim is true for all multiple shooting grids with  $n_{ms} - 1$  shooting nodes, where the node at time  $t_{m+1}$ ,  $0 \leq m \leq n_{ms} - 1$  has been removed. We denote

$$s' := (s_1, \dots, s_m, s_{m+2}, \dots, s_{n_{ms}}) \in \mathbb{R}^{n_x(n_{ms}-1)},$$

$$\mathcal{X}' := \mathcal{X}_1 \times \dots \times \mathcal{X}_m \times \mathcal{X}_{m+2} \times \dots \times \mathcal{X}_{n_{ms}}$$

and define the corresponding convexified  $n_{ms} - 1$  node multiple shooting NLP as

$$\begin{aligned} & \min_{q, s'} c^T s_{n_{ms}} \\ & \text{subject to} \\ & s_i \geq x_{cv}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, \mathcal{S}_{i-1}), \quad i = 1, \dots, m, m+3, \dots, n_{ms} \\ & s_{m+2} \geq x_{cv}^{ms,m}(t_{m+2}; q, s_m, \mathcal{Q}, \mathcal{S}_m), \\ & s_i \leq x_{cc}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, \mathcal{S}_{i-1}), \quad i = 1, \dots, m, m+3, \dots, n_{ms} \\ & s_{m+2} \leq x_{cc}^{ms,m}(t_{m+2}; q, s_m, \mathcal{Q}, \mathcal{S}_m), \\ & s_i \in \mathcal{S}_i := [\underline{x}_i, \bar{x}_i], \quad i = 1, \dots, m, m+2, \dots, n_{ms}, \\ & q \in \mathcal{Q}. \end{aligned} \tag{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}'}$$

This is obviously again a multiple shooting formulation like in Definition 11 and hence a convex relaxation of  $(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}'})$ . To finish the proof, we have to show that  $\text{val}(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}'}) \leq \text{val}(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}})$ , ie, that the underestimation of the objective function value is better (larger, we are minimizing) with an additional shooting node. Let  $(q^*, s^*)$  be a solution of  $(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}})$ . Then  $(q^*, s^{*'})$  is a feasible solution of  $(\text{MS}_{cv}^{\mathcal{Q} \times \mathcal{X}'})$ , if

$$x_{cv}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, \mathcal{Q}, \mathcal{S}_{m+1}) \geq x_{cv}^{ms,m}(t_{m+2}; q^*, s_m^*, \mathcal{Q}, \mathcal{S}_m), \tag{31}$$

$$x_{cc}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, Q, S_{m+1}) \leq x_{cc}^{ms,m}(t_{m+2}; q^*, s_m^*, Q, S_m), \quad (32)$$

because this implies a larger feasible region  $S_{m+2} = [s_{m+2}, \overline{s_{m+2}}]$  for the variable  $s_{m+2}$  in  $(MS_{cv}^{Q \times X'})$  when compared to  $(MS_{cv}^{Q \times X})$ , and by means of monotonicity of interval arithmetics and validated integration also for all subsequent  $S_i$ ,  $i = m+3, \dots, n_{ms}$ . The inequalities (Equations 31 and (32)) follow from Equations 27 and 28 in Assumption 15, concluding the proof.  $\square$

We now have a closer look at the crucial Assumption 15. We focus on the underestimation (Equation (27)). Making use of the definition (Equation (20)) of convex underestimation, the assumed inequalities are for triplets  $(t_m, t_{m+1}, t_{m+2})$  of neighboring points

$$\begin{aligned} & x_j^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m+1,j} \psi(q_k^*) + \sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,m+1,j} \psi(s_{m+1,l}^*) \\ & \geq x_j^{ms,m}(t_{m+2}; q^*, s_m^*) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m,j} \psi(q_k^*) + \sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,m,j} \psi(s_{m,l}^*) \end{aligned} \quad (33)$$

for all  $j = 1, \dots, n_x$  with solution trajectories on the time horizons  $[t_{m+1}, t_{m+2}]$  with initial value  $s_{m+1}$  and on  $[t_m, t_{m+2}]$  with initial value  $s_m$ , respectively.

The inequalities depend on  $q^*$ , on  $s^*$ , on the bounds in  $Q, S$ , on the choice of time points  $t_m, t_{m+1}, t_{m+2}$ , and on the particular dynamics that result in the trajectories  $x(\cdot)$  and the second-order derivatives from which the  $\alpha$  values are calculated. This makes it complicated to derive conditions for an optimal choice of a time grid, especially as it would probably be applied to several or all problems in the Branch&Bound tree, ie, for varying  $q^*, s^*, Q, S$ .

Still, the inequalities (Equation (33)) may give rise to automatic heuristic choices of efficient multiple shooting grids. Assuming that the optimal solution  $(q^*, s^*)$  is known for a grid that has been created on the basis of trial and error, a simulation may validate that the inequalities (Equation (33)) and their concave counterparts hold for all triples of neighboring time points in the multiple shooting grid. This implies that the chosen grid provides better relaxations than all subgrids with less points, including single shooting. As  $(q^*, s^*)$  is the result of Algorithm 1 and hence unknown and changing through the Branch&Bound tree, heuristic choices such as the solution of the root node problem of the Branch&Bound tree can be chosen. It is also promising to use different multiple shooting grids for different parts of the tree, using heuristic updates based on solutions of intermediate problems. We have not yet implemented such an algorithm, although.

We continue by motivating why Equation 33 is plausible to hold for many nodes even for random and for equidistant multiple shooting grids. As derived in the proof of Theorem 13, we know that the bounds of the multiple shooting nodes have a fourth-order convergence, ie,

$$\psi(s_{m+1,l}^*) \geq -\frac{1}{4} \sum_{j=1}^{n_x} \left( \sum_{k=1}^{n_q} (\alpha_{cc,k}^{ms,m,j} + \alpha_{cv,k}^{ms,m,j}) \psi(q_k^*) + \sum_{l=1}^{n_x} (\alpha_{cc,n_q+l}^{ms,m,j} + \alpha_{cv,n_q+l}^{ms,m,j}) \psi(s_{m,l}^*) \right)^2$$

This implies that at some part in the tree, the bounds  $\underline{s}, \bar{s}$  are so tight that

$$s_{m+1} \approx \underline{s_{m+1}} \approx \overline{s_{m+1}} \approx x_j^{ms,m}(t_{m+1}; q^*, s_m^*)$$

and therefore

$$x_j^{ms,m+1}(t_{m+2}; q^*, s_{m+1}) \approx x_j^{ms,m}(t_{m+2}; q^*, s_m^*). \quad (34)$$

With the same argument, the terms

$$\sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,m+1,j} \psi(s_{m+1,l}) \text{ and } \sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,m,j} \psi(s_{m,l}^*) \quad (35)$$

will be dominated by the relaxation terms with respect to  $q$ . Thus, we may neglect the first and the third terms in Equation 33 as we approach tighter bounds around the optimal solution, where it is well known that most of the computational workload is concentrated because of the clustering effect, compare a number of studies.<sup>36-38</sup> Hence, the most important part of Equation 33 is the relation

$$\sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m+1,j} \psi(q_k^*) \geq \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m,j} \psi(q_k^*), \quad (36)$$

and a sufficient condition for the dominance of the lower bounds would be that

$$\alpha_{cv,k}^{ms,m+1,j} \leq \alpha_{cv,k}^{ms,m,j} \quad (37)$$

for all  $k = 1, \dots, n_q, j = 1, \dots, n_x$  and the considered  $q^*, s^*, Q, S, t_m, t_{m+1}$ . These values  $\alpha_{cv,k}^{ms,m+1,j} \in \mathbb{R}$  are given by Equation 22 and Definition 8 as  $-\frac{1}{2}$  times the minimum between 0 and

$$c_j \frac{d^2 x_j^{ms,m+1}(t_{m+2})}{dq_k dq_k} - \sum_{l \neq k}^{n_q} \left| c_j \frac{d^2 x_j^{ms,m+1}(t_{m+2})}{dq_k dq_l} \right| - \sum_{l=1}^{n_x} \left| c_j \frac{d^2 x_j^{ms,m+1}(t_{m+2})}{dq_k ds_{m+1l}} \right|. \quad (38)$$

Because of the dependence of Equation 38 on  $q^*, s^*, Q, S, t_m, t_{m+1}$  and on the particular dynamics, it is still difficult to make a general statement on the circumstances under which Equation 33 holds. However, we note that a reduction of the absolute values of the second derivatives at time  $t_{m+2}$  is likely to result in Equation 37 that implies Equation 36 that makes Equation 33 very likely, especially close to the optimal solution. The main advantage of multiple shooting with one additional node at time  $t_{m+1}$  results from the fact that the initial values for the second-order sensitivities at time  $t_{m+1}$  are zero, compare Lemma 19. Although there may be counterexamples where this actually leads to an increase of the values in Equation 38, for most dynamics, a beneficial effect can be expected. This is illustrated exemplarily in Figures 4 and 5.

## 5 | NUMERICAL RESULTS

We illustrate our findings with numerical results for an established benchmark problem. It is commonly referred to as ‘‘singular control problem’’ and considered for global optimal control in a number of studies.<sup>14,39-42</sup> Using a reformulation of the Lagrange term and of time as additional differential states  $x_4$  and  $x_5$ , respectively, and  $t_f = 1$ , the problem can be stated as

$$\begin{aligned} \min_{u(\cdot)} \quad & x_4(t) \\ \text{subject to} \quad & \dot{x}_1(t) = x_2(t) & t \in [0, t_f] \\ & \dot{x}_2(t) = -x_3(t)u(t) + 16x_5(t) - 8 & t \in [0, t_f] \\ & \dot{x}_3(t) = u(t) & t \in [0, t_f] \\ & \dot{x}_4(t) = x_1(t)^2 + x_2(t)^2 + \frac{5}{10^4} \left( x_2(t) + 16x_5(t) - 8 - \frac{x_3(t)u(t)^2}{10} \right)^2 & t \in [0, t_f] \\ & \dot{x}_5(t) = 1 & t \in [0, t_f] \\ & x(0) = (0, -1, -\sqrt{5}, 0, 0) \\ & u(t) \in [-4, 10] & t \in [0, t_f]. \end{aligned} \quad (39)$$

All numerical calculations have been performed on a i7-5820 K processor with 16 GB RAM under Ubuntu 14.04. We use our C++ code `GloOptCon`<sup>43</sup> for the numerical study. It is based on external code in submodules, such as `CVODES` from the `SUNDIALS` software suite in version 2.6.0<sup>44</sup> for integration, `VSPODE` 1.4<sup>34</sup> for the validated integration for state bounds and the sensitivity Hessian, `Ipopt` 3.12.3<sup>45</sup> for the solution of NLPs, and `cppad`<sup>46</sup> for automatic differentiation. It is designed to allow a fair comparison between the single and multiple shooting versions of Algorithm 1. We define some algorithmic choices.

**Definition 17.** ( $n_q$  discretization of the control function)

We discretize the control function  $u : [0, t_f] \mapsto \mathbb{R}$  via piecewise constant functions on an equidistant grid, depending on the total number  $n_q$  of intervals, ie,

$$u(t) = q_k \quad \forall t \in [t_{k-1}, t_k), k = 1, \dots, n_q. \quad (40)$$

We consider optimal solutions for Equations 39 based on  $n_q$  discretizations with  $n_q = 1, \dots, 4$ . The optimal objective function values and control values found by our algorithms are listed in Table 2. They coincide with the values from Lin and Stadtherr.<sup>42</sup>

We shall compare single and multiple shooting algorithms with respect to their performance to obtain the results from Table 2. They all use an  $n_q$  discretization of the control function.

**TABLE 2** Globally optimal (for  $\varepsilon = 10^{-3}$ ) objective function value  $x_4^*(1)$  of problem 39 for an increasing number  $n_q$  of equidistant control discretization points together with the optimal controls  $q^*$

$n_q$	$x_4^*(1)$	$q^*$
1	0.4965	( 4.0709 )
2	0.2771	( 5.5748, -4.0000 )
3	0.1475	( 8.0015, -1.9438, 6.0420 )
4	0.1237	( 9.7890, -1.1997, 1.2566, 6.3558 )

**TABLE 3** Numerical results for different  $n_q$  control discretizations of problem 39 using the methods from Definition 18

$n_q$ Control Intervals	DSS		DMS1		DMS $n_q$	
	Iter	Time	Iter	Time	Iter	Time
1	1	0.44	0	0.08	0	0.49
2	62	61.87	59	66.6	63	121.6
3	1004	2702	966	2844	782	2196
4	13904	84161	13183	81937	5902	22650

Abbreviations: DMS, direct multiple shooting; DSS, direct single shooting; NLP, nonlinear optimization problem. Shown are the number of Branch&Bound nodes (Iter) and the CPU time in seconds. The CPU time is strongly dominated by the calculation of the  $\alpha$  values ( $\approx$  factor 200 compared to NLP solutions).

**Definition 18.** (Compared algorithms)

1. DSS uses Algorithms 1 and 2.

2. Direct multiple shooting with 1 shooting interval (DMS1) uses Algorithms 1 and 3 with a multiple shooting grid with

$n_{ms} = 1$  multiple shooting interval. This approach can also be seen as DSS with one extra variable vector  $s_{t_i}$  for  $x(t_i)$  and was proposed in Papamichail and Adjiman.<sup>16</sup>

3. Direct multiple shooting (DMS $n_q$ ) uses Algorithms 1 and 3 with a multiple shooting grid with  $n_{ms} = n_q$  equidistant multiple shooting intervals. In our comparison, the multiple shooting grid is chosen identical to the equidistant grid  $\{t_0, t_1, \dots, t_{n_q}\}$  of the control discretization.

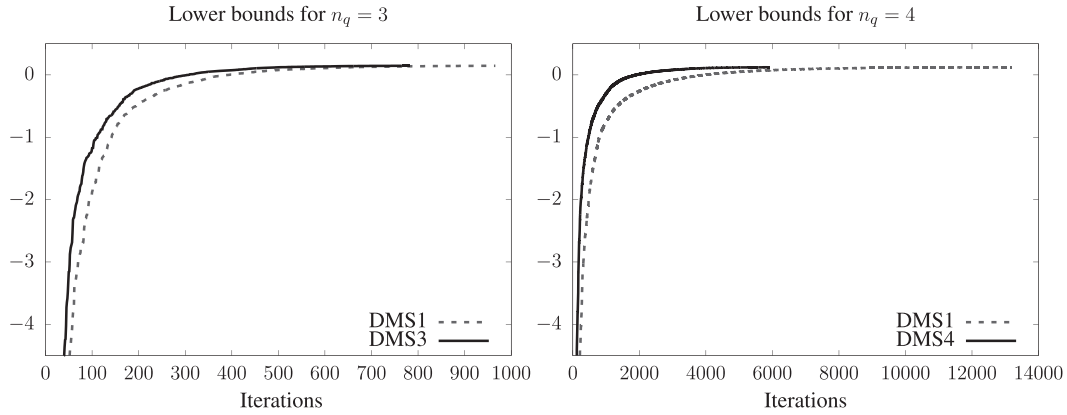
In the interest of a fair comparison, we do not use special branching rules or heuristics.

Table 3 shows computational results. The DSS and DMS1 show a similar behavior as can be expected, because the objective is linear and there are no end point constraints. In comparison, the number of iterations of DMS $n_q$  is significantly better as the numbers of control variables  $n_q$  and multiple shooting intervals  $n_{ms} = n_q$  increase. Note that the time difference between the equivalent approaches DMS1 and DMS $n_q$  for  $n_q = 1$  is due to a technical issue ( $s_0$  is included as a degree of freedom although it is fixed and causes thus computational overhead). The reduced number of iterations of multiple shooting compared to single shooting is due to the tighter relaxations of the nonconvex dynamics. Figure 3 shows the development of the lower bounds, highlighting the practical impact of Theorem 16.

To gain more insight, we compare the lower bounds for some selected choices of  $Q$  in Table 4. Corresponding intervals  $S_i$  are determined via the propagated updates in Algorithm 3. For the full domain  $[-4, 10]^{n_q}$ , DSS results in very weak lower bounds for all choices of  $n_q$ . This can be explained with the quadratic convergence behavior due to the  $\alpha$ BB terms  $\psi(q_i) \geq -\frac{1}{4}(\bar{q}_i - \underline{q}_i)^2$ . It is remarkable, although, how the introduction of additional multiple shooting nodes improves the bounds. Also for tighter domains, Table 4 reveals how much the multiple shooting approach DMS $n_q$  dominates single shooting in terms of the quality of relaxations as  $n_q = n_{ms}$  increases. It is especially encouraging that the bounds are significantly tighter close to the optimal solution  $q^*$  as prognosed in Section 4, because it is well known that the clustering effect causes many iterations in a neighborhood of the optimum.

As motivated above, this dominance of lower bounds is guaranteed if Equation 33 holds, and is more likely, if the relaxation factors are dominated for which the second-order sensitivities (Equation (38)) are decisive. The involved tensors are difficult to





**FIGURE 3** Evolution of lower bounds for the global optimum of problem 39 with 3 (left) and 4 (right) equidistant control discretizations between DMS1 and DMS $n_q$ . It becomes obvious how the multiple shooting approach with larger  $n_{ms}$  dominates the case with only one shooting interval. DMS indicates direct multiple shooting

**TABLE 4** Lower bounds on the solution of problem 39 for specific choices of  $Q$

$n_q$	$[q_i, \bar{q}_i] \forall i$	DSS	DMS1	DMS $n_q$
1	$[-4, 10]$	-0.735976	0.496546	0.496546
	$[-4, -3.95]$	128.981	128.981	128.981
	$[-9.95, 10]$	138.949	138.949	138.949
	$q_i^* + [-1; 1]$	0.496545	0.496546	0.496546
	$q_i^* + [-0.1; 0.1]$	0.496545	0.496546	0.496546
	$q_i^* + [-0.01; 0.01]$	0.496545	0.496545	0.496545
3	$[-4, 10]$	-906.238	-187.834	-187.834
	$[-4, -3.95]$	128.981	128.981	128.981
	$[-9.95, 10]$	138.949	138.949	138.949
	$q_i^* + [-1; 1]$	-0.767086	-0.319456	-0.312031
	$q_i^* + [-0.1; 0.1]$	0.146103	0.146104	0.146709
	$q_i^* + [-0.01; 0.01]$	0.147468	0.147469	0.147474
6	$[-4, 10]$	-993.86	-252.515	-169.889
	$[-4, -3.95]$	128.981	128.981	128.981
	$[-9.95, 10]$	138.949	138.949	138.949
	$q_i^* + [-1; 1]$	-0.605863	-0.250872	-0.053446
	$q_i^* + [-0.1; 0.1]$	0.120369	0.12037	0.12185
	$q_i^* + [-0.01; 0.01]$	0.122361	0.122362	0.122381

Abbreviations: DMS, direct multiple shooting; DSS, direct single shooting.

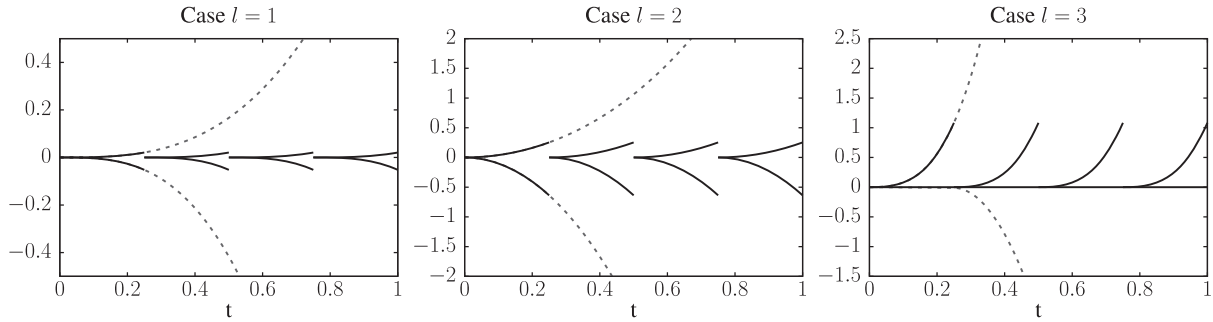
visualize, obviously. In Figures 4 and 5, we show selected trajectories for the choice of DMS $n_q$  with  $n_q = n_{ms} = 4$  and for the particular entry  $j = 4$  of the state vector (because the right-hand side of  $x_4(t)$  in problem 39 is the most interesting one).

In Figure 4, the shown trajectories for DMS1 and DMS4,

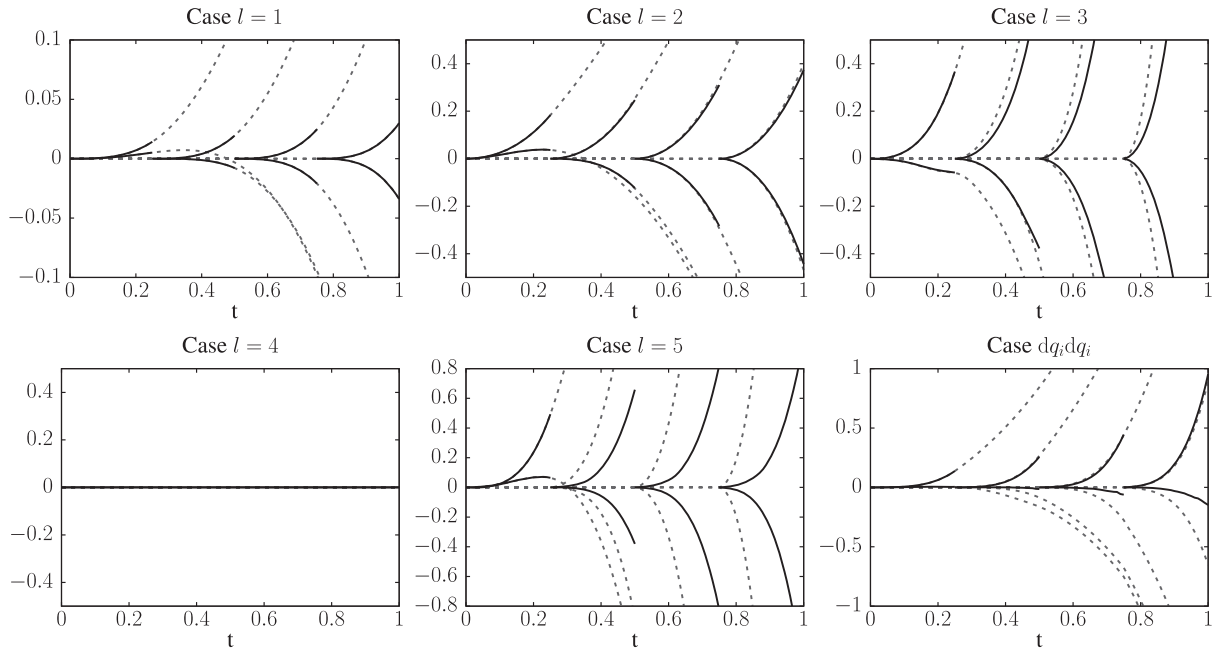
$$\frac{d^2 x_j^{ss,i}(t; Q)}{ds_{0k} ds_{0l}} \quad \text{and} \quad \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{ds_{ik} ds_{il}} \quad \text{for } i = 0, \dots, n_{ms} - 1, \tag{41}$$

illustrate exemplarily to what extent the “resetting” of the initial values

$$\frac{d^2 x_j^{ms,i}(t_i; Q, S_i)}{ds_{ik} ds_{il}} = 0$$



**FIGURE 4** Comparison between DMS1 (---) and DMS4 (—). Plot of selected trajectories (Equation (41)) for  $j = 4, k = 3$  and different values  $l = 1 \dots 3$  (from left to right). The single shooting trajectories grow exponentially over the whole time horizon, the multiple shooting trajectories on smaller subintervals. DMS indicates direct multiple shooting



**FIGURE 5** Comparison between DMS1 (---) and DMS4 (—) for selected second-order sensitivities (Equation (42)) for  $l = 1, \dots, 5$  (left to right) and of Equation 43 (bottom right), all for  $j = 4$ . Note that in each plot for both DMS1 and DMS4, 4 trajectories can be seen that correspond to derivatives with respect to the discretized control function  $u(t) = q_i \forall t \in [t_i, t_{i+1}]$ , with  $i = 0, \dots, n_{ms} - 1 = 3$ . This is different from Figure 4 where only derivatives with respect to  $s_0$  occur in DMS1, but with respect to  $(s_0, s_1, s_2, s_3)$  in DMS4. DMS indicates direct multiple shooting

at the multiple shooting times  $t_i$  reduces the amplitude of the decisive underestimations and overestimations at the end time  $t_f$ , compared to the exponentially growing single shooting sensitivities. In all plots, DMS1 corresponds to single shooting, DMS4 to multiple shooting.

In Figure 5, we have a look at one particular row of the tensor belonging to  $j = 4$  and the control discretization  $q$  with trajectories

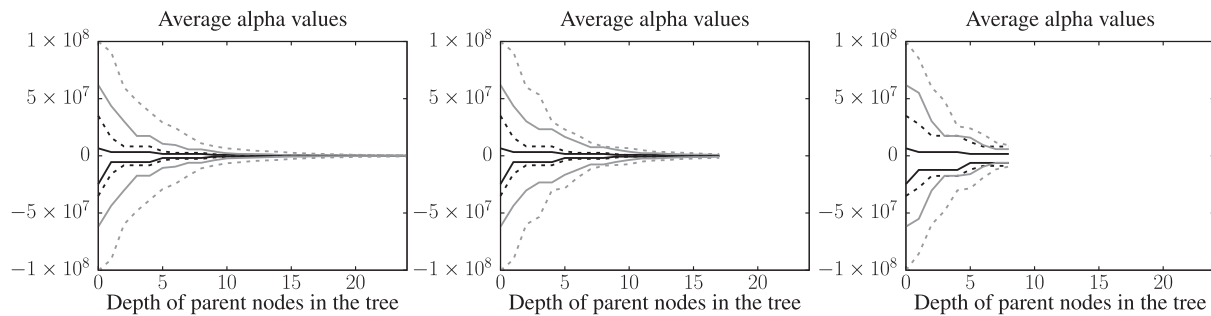
$$\frac{d^2 x_j^{ss,i}(t; Q)}{dq_i ds_{0l}} \text{ for } i = 0, \dots, n_{ms} - 1 \quad \text{and} \quad \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{dq_i ds_{il}} \text{ for } i = 0, \dots, n_{ms} - 1 \quad (42)$$

on the off-diagonals and with

$$\frac{d^2 x_j^{ss,i}(t; Q)}{dq_i dq_i} \text{ for } i = 0, \dots, n_{ms} - 1 \quad \text{and} \quad \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{dq_i dq_i} \text{ for } i = 0, \dots, n_{ms} - 1 \quad (43)$$

on the main diagonals.

Figure 5 reveals interesting properties. First and most important for the better performance, the magnitude of the second-order sensitivities with respect to  $(q_0, q_1, q_2)$  (not for  $q_3$ , obviously!) is drastically reduced, as in Figure 4 for the pure state derivatives.



**FIGURE 6** Prototypical evolution of  $\alpha$  values within the  $\alpha$ BB tree of DMS4. For different leaf nodes (ie, nodes that have been pruned), the averaged  $\alpha$  values are plotted for all parent nodes, starting with the root node with depth 0. Shown are  $\alpha_{cv}$  (positive) and  $\alpha_{cc}$  (negative) for the different shooting intervals, 1 (—), 2 (⋯), 3 (---), and 4 (-·-·-). One observes that in this example, the  $\alpha$  values are monotonically dominated by early time intervals. This monotonicity may be broken because of the branching, as in the rightmost plot. The behavior of the left plot is more representative for most nodes in the tree, although. The  $\alpha$  values are “monotonically constant,” ie, they do not change if the value of the next intervals does not change. This is due to the particular branching decision, because both  $S_i$  and  $Q_i$  only affect time intervals  $j \geq i$

For the case  $l = 4$ , all trajectories are identical zero, as the artificial state  $x_4(t)$  does not enter the function  $f(\cdot)$ . For  $l = 1$  at time  $t = 0.25$ , one sees that it is possible that the resetting of the initial value is outside of the bounds of the single shooting approach, which may theoretically result in a worse value for  $\alpha$  even for quasi-monotone functions. One also observes that the estimations for identical values of  $i$  differ, sometimes significantly. For example, for  $dq_2dq_2$ , single shooting shows poor performance with respect to the lower bound. These deviations between DMS1 and DMS4 are related to the performance of the validated integration, where wrapping effects and the impact of the system size on Taylor approximations play a crucial role. It would be interesting to study such differences also in the context of other approaches, such as Chebyshev enclosures.<sup>47</sup>

Figure 6 shows how the  $\alpha$  values change within the Branch&Bound tree for DMS4. One sees prototypically the monotonic decrease of the relaxation factors due to branching.

## 6 | EXTENSIONS

The applicability of the proposed multiple shooting approach is not restricted to the problem class 1, but can be further generalized. There are several straightforward extensions, ie, problem formulations to which our discussion of advantages of multiple shooting can be applied as well, such as

- an explicit dependence of the function  $f(\cdot)$  on time  $t$  that can be included via an artificial state  $x_{n_x+1}$  as done for problem 39;
- model parameters  $p$  that are either fixed and can then be regarded as part of the functions or degrees of freedom and can be included in the vector  $q$ ;
- continuous control functions  $u : \mathcal{T} \mapsto \mathbb{R}^{n_u}$  or integer control function  $v : \mathcal{T} \mapsto \{v^1, \dots, v^{n_v}\}$ , see one study,<sup>48</sup> which both can be approximated arbitrarily close using the finite-dimensional control vector  $q$  and basis functions; and
- more general boundary conditions and free initial values  $s_0$ , as well as multistage formulations.

An extension is also possible towards

- mixed path-control constraints  $g : \mathbb{R}^{n_x \times n_q} \mapsto \mathbb{R}^{n_s}$  via  $g(x(t), q) \leq 0$ ,
- bounded nonconvex sets  $\mathcal{Q}$  and  $\mathcal{X}$ , and
- nonlinear objective functionals of Bolza type.

Here, further convexifications of the occurring sets and functions in the spirit of the relaxed problem (NLP<sub>cv</sub><sup>v</sup>) are necessary, eg,  $g_{cv}(x(t), q) \leq 0$ . Even if it is possible to obtain an equivalent formulation of type 1, eg, by introducing an additional artificial state variable for the nonlinear Lagrange term, this should be done with special care, as function composition and relaxation do not commute. An important assumption for the applicability of our approach is the boundedness of  $\mathcal{Q}$  and  $\mathcal{X}$ .

In the interest of a clear presentation, we focused on the most important aspect of our novel approach, a multiple shooting-based convex relaxation of the nonconvex dynamics. In the PhD thesis,<sup>43</sup> further algorithmic extensions have been studied that we list here for completeness. They comprehend

- more efficient ways to calculate  $\alpha$  on the basis of an adaptively scaled Gershgorin approach;

- a motivation why it is beneficial to choose the multiple shooting time grid identical to the control discretization grid and a more detailed analysis that takes the local influence of control function variables  $q_i$  only on time intervals  $[t_i, t_{i+1}]$  into account;
- a reduced space heuristic, which neglects the terms  $\alpha_{.n_q+l}^{ms,i,j} \psi(s_{il})$  in Equations 20 and 21, motivated by the fourth-order convergence shown in Theorem 13 (this heuristic found the global optimum for all studied examples in a runtime orders of magnitude below the full space approaches);
- a fast bounds strategy, which keeps  $\alpha$  values constant for several iterations in the Branch&Bound tree: Updating  $\alpha$  based on the second-order interval sensitivities is significantly more expensive in global optimal control compared to calculating an interval Hessian in nondynamic global optimization. Therefore, it is even more tempting to speed up the runtime by infrequent updates of  $\alpha$ , relying on the fact that the underestimators and overestimators are still getting better in each iteration because of the tightened variable bounds. We observed a considerable speed up of the overall runtime at the expense of the number of iterations;
- pure constraint propagation for a lower bound and periodic local solution of the problem with the tightened bounds for an upper bound, as suggested in one study<sup>42</sup> as an alternative to  $\alpha$ BB. The method has the advantage to avoid the costly validated integration of second derivatives and the disadvantage of more iterations due to the weaker bounds. It is problem-dependent when this approach is competitive to  $\alpha$ BB (actually, it outperforms the  $\alpha$ BB approach for problem 39 concerning runtime), but obviously independent from our comparison of single vs multiple shooting.

For details, we refer to.<sup>43</sup>

## 7 | SUMMARY

We presented a novel method to globally solve a general class of optimal control problems in a rigorous way. We showed that no additional nodes occur despite the higher number of optimization variables, and under which conditions, the obtained lower bounds are better than for the standard approach. The main building blocks are the fourth-order convergence of the bounds of the introduced variables and a reset of the initial values on selected time points of the integration horizon. The superiority has been illustrated with a benchmark problem from the literature.

## ACKNOWLEDGMENTS

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 647573) and from the German BMBF under grant 05M2013—GOSSIP. The authors gratefully acknowledge this.

## REFERENCES

1. Neumaier A. *Complete Search in Continuous Global Optimization and Constraint Satisfaction*. Cambridge: University Press; Cambridge; 2004. 271-369.
2. Floudas CA, Gounaris CE. A review of recent advances in global optimization. *J Global Optim*. 2009;45(1):3-38.
3. Moore RE. *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall; 1966.
4. Land AH, Doig AG. An automatic method of solving discrete programming problems. *Econometrica*. 1960;28:497-520.
5. Little JDC, Murty KG, Sweeney DW, Karel C. An algorithm for the traveling Salesman problem. *Oper Res*. 1963;11(6):972-989.
6. Mitsos A, Chachuat B, Barton PI. McCormick-based relaxations of algorithms. *SIAM J Optim*. 2009;20(2):573-601.
7. Scott JK, Stuber MD, Barton PI. Generalized McCormick relaxations. *J Global Optim*. 2011;51(4):569-606.
8. Tawarmalani M, Sahinidis NV. A polyhedral branch-and-cut approach to global optimization. *Math Program*. 2005;103:225-249.
9. Androulakis IP, Maranas CD, Floudas CA.  $\alpha$ BB: a global optimization method for general constrained nonconvex problems. *J Global Optim*. 1995;7(4):337-363.
10. Adjiman CS, Floudas CA. Rigorous convex underestimators for general twice-differentiable problems. *J Global Optim*. 1996;9:23-40.
11. Adjiman CS, Androulakis IP, Floudas CA. A global optimization method,  $\alpha$ BB, for general twicedifferentiable constrained NLPs - II. Implementation and computational results. *Computers and Chemical Engineering*. 1998;22(9):1159-1179.
12. Skjäl A, Westerlund T. New methods for calculating  $\alpha$ BB-type underestimators. *J Global Optim*. 2014;58(3):411-427.
13. Gerschgorin S. Über die Abgrenzung der Eigenwerte einer Matrix. *Bull de l'Acad des Sci de l'URSS. Cl des sci math et na*. 1931;6:749-754.
14. Esposito WR, Floudas CA. Deterministic global optimization in nonlinear optimal control problems. *J Global Optim*. 2000;17(1-4):97-126.
15. Esposito WR, Floudas CA. Global optimization for the parameter estimation of differential- algebraic systems. *Ind Eng Chem Res*. 2000;39(5):1291-1310.

16. Papamichail I, Adjiman CS. A rigorous global optimization algorithm for problems with ordinary differential equations. *J Global Optim.* 2002;24(1):1-33.
17. Papamichail I, Adjiman CS. Proof of convergence for a global optimization algorithm for problems with ordinary differential equations. *J Global Optim.* 2005;33:83-107.
18. Bock HG, Plitt KJ. A multiple shooting algorithm for direct solution of optimal control problems. *Proceedings of the 9th IFAC World Congress.* Pergamon Press, Budapest; 1984:242-247.
19. Leineweber DB, Bauer I, Schäfer AAS, Bock HG, Schlöder JP. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (parts I and II). *Comput Chem Eng.* 2003;27:157-174.
20. Betts JT. *Practical Methods for Optimal Control Using Nonlinear Programming.* Philadelphia: SIAM Philadelphia; 2001.
21. Biegler LT. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*, Series on Optimization. Philadelphia: SIAM; 2010.
22. Albersmeyer J, Diehl M. The lifted Newton method and its application in optimization. *SIAM J Optim.* 2010;20(3):1655-1684.
23. Janka D, Kirches C, Sager S, Wächter A. An SR1/BFGS SQP algorithm for nonconvex nonlinear programs with block-diagonal Hessian matrix. *Math Prog Comput.* 2016;8(4):435-459.
24. Frasch J. Parallel algorithms for optimization of dynamic systems in real-time. *PhD thesis*, Otto-von- Guericke University Magdeburg, 2014.
25. Houska B, Chachuat B. Branch-and-lift algorithm for deterministic global optimization in nonlinear optimal control. *J Optim Theory Applic.* 2014;162(1):208-248.
26. Moore RE, Kearfott RB, Cloud MJ. *Introduction to Interval Analysis.* Philadelphia, PA, USA: SIAM; 2009.
27. Scholz D. *Deterministic Global Optimization: Geometric Branch-and-bound Methods and their Applications.* Springer, Heidelberg: Springer Optimization and Its Applications; 2011.
28. Falk JE, Soland RM. An algorithm for separable nonconvex programming problems. *Manage Sci.* 1969;15(9):550-569.
29. Horst R, Tuy H. *Global Optimization: Deterministic Approaches.* Berlin, Heidelberg: Springer; 1996.
30. Tuy H, Horst R. Convergence and restart in branch-and-bound algorithms for global optimization. Application to concave minimization and D.C. Optimization problems. *Math Program.* 1988;41(1-3):161-183.
31. Stein O, Kirst P, Steuermann P. An enhanced spatial branch-and-bound method in global optimization with nonconvex constraints. *Optimization Online.* 2013.
32. Lohner RJ. Computation of guaranteed enclosures for the solutions of ordinary initial and boundary value problems. In: Cash JR, Gladwell I, eds. *Computational Ordinary Differential Equations.* Oxford: Clarendon Press; 1992:425-436.
33. Nedialkov NS, Jackson KR, Pryce JD. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliable Comput.* 2001;7(6):449-465.
34. Lin Y, Stadtherr MA. Validated solutions of initial value problems for parametric ODEs. *Appl Numerical Math.* 2007;57(10):1145-1162.
35. Houska B, Villanueva ME, Chachuat B. A validated integration algorithm for nonlinear ODEs using Taylor models and ellipsoidal calculus. *IEEE 52nd Annual Conference on Decision and Control (CDC)*, Florence, Italy; December 2013:484-489.
36. Du K, Kearfott RB. The cluster problem in multivariate global optimization. *J Global Optim.* 1994;5(3):253-265.
37. Schichl H, Neumaier A. Exclusion regions for systems of equations. *SIAM J Numer Anal.* 2004;42(1):383-408.
38. Wechsung A, Schaber SD, Barton PI. The cluster problem revisited. *J Global Optim.* 2014;58(3):429-438.
39. Luus R. Optimal control by dynamic programming using systematic reduction in grid size. *Int J Control.* 1990;51(5):995-1013.
40. Chachuat B, Latifi MD. A new approach in deterministic global optimisation of problems with ordinary differential equations. In: Floudas CA, Pardalos PM, eds. *Frontiers in Global Optimization of Nonconvex Optimization and Its Applications*, vol. 74. Heidelberg: Springer; 2004:83-108.
41. Singer AB, Barton PI. Global optimization with nonlinear ordinary differential equations. *J Global Optim.* 2006;34(2):159-190.
42. Lin Y, Stadtherr MA. Deterministic global optimization of nonlinear dynamic systems. *AIChE Journal.* 2007;53(4):866-875.
43. Diedam H. Global optimal control using direct multiple shooting. *PhD thesis*, University of Heidelberg, 2015.
44. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS. SUNDIALS: Suite of Nonlinear and Differential/Algebraic Equation Solvers. *ACM Trans on Math Software.* 2005;31(3):363-396.
45. Wächter A, Biegler LT. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math Program.* 2006;106(1):25-57.
46. Bell BM. *CppAD: A Package for Differentiation of C++ Algorithms*: Computational Infrastructure for Operations Research; 2015.
47. Chachuat B, Houska B, Paulen R, Perić N, Rajyaguru J, Villanueva ME. Set-theoretic approaches in analysis, estimation and control of nonlinear systems. *IFAC-PapersOnLine.* 2015;48(8):981-995. 9th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2015, Whistler, Canada, 710 June 7 10, 2015.
48. Sager S, Bock HG, Diehl M. The integer approximation error in mixed-integer optimal control. *Math Program A.* 2012;133(1-2):1-23.
49. Bock HG. *Randwertproblemmethoden zur Parameteridentifizierung in Systemen nichtlinearer Differentialgleichungen* of *Bonner Mathematische Schriften*, vol. 183: Universität Bonn Bonn; 1987.
50. Albersmeyer J. Adjoint Based Algorithms and Numerical Methods for Sensitivity Generation and Optimization of Large Scale Dynamic Systems. *PhD thesis*, Ruprecht-Karls-Universität Heidelberg, 2010.

**How to cite this article:** Diedam H, Sager S. Global optimal control with the direct multiple shooting method. *Optim Control Appl Meth.* 2018;39:449-470. <https://doi.org/10.1002/oca.2324>

## APPENDIX

The following result is well known and used in the context of dynamic optimization, eg, Bock.<sup>49</sup>

**Lemma 19.** (First- and second-order sensitivities)

Let Assumption 2 holds, indices  $i \in \{0, \dots, n_{ms} - 1\}$  and  $j \in \{1, \dots, n_x\}$  be given, and let  $x(\cdot; q, s_i)$  be the solution trajectory to the differential equation

$$\dot{x}(t) = f(x(t), q), \quad x(t_i) = s_i$$

as specified in Definition 4. For  $k = 1, \dots, n_q$  and  $l = 1, \dots, n_x$  the trajectories  $\frac{dx_j(\cdot; q, s_i)}{dq_k} : \mathcal{T} \mapsto \mathbb{R}$  and  $\frac{dx_j(\cdot; q, s_i)}{ds_{il}} : \mathcal{T} \mapsto \mathbb{R}$  are solutions of the differential equations

$$\begin{aligned} \frac{d}{dt} \frac{dx_j(t; q, s_i)}{dq_k} &= \frac{\partial f_j(x(t), q)}{\partial x} \frac{dx(t; q, s_i)}{dq_k} \\ &+ \frac{\partial f_j(x(t), q)}{\partial q_k}, \quad \frac{dx_j(t_i; q, s_i)}{dq_k} = 0, \end{aligned} \quad (\text{A1})$$

$$\frac{d}{dt} \frac{dx_j(t; q, s_i)}{ds_{il}} = \frac{\partial f_j(x(t), q)}{\partial x} \frac{dx(t; q, s_i)}{ds_{il}}, \quad \frac{dx_j(t_i; q, s_i)}{ds_{il}} = \delta_{jl}, \quad (\text{A2})$$

respectively, and the trajectories  $h_{q_k s_{il}}^{ms, i, j}(\cdot; q, s_i) / c_j = \frac{d^2 x_j(\cdot; q, s_i)}{dq_k ds_{il}} : \mathcal{T} \mapsto \mathbb{R}$  are solutions of

$$\begin{aligned} \frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{dq_k ds_{il}} &= \frac{dx(t; q, s_i)^T \partial^2 f_j(x(t), q)}{dq_k} \frac{dx(t; q, s_i)}{ds_{il}} \\ &+ \frac{\partial^2 f_j(x(t), q)}{\partial q_k \partial x} \frac{dx(t; q, s_i)}{ds_{il}} \\ &+ \frac{\partial f_j(x(t), q)}{\partial x} \frac{d^2 x(t; q, s_i)}{dq_k ds_{il}}, \quad \frac{d^2 x_j(t; q, s_i)}{dq_k ds_{il}} = 0. \end{aligned} \quad (\text{A3})$$

For  $k, l = 1, \dots, n_x$  the trajectories  $h_{s_{ik} s_{il}}^{ms, i, j}(\cdot; q, s_i) / c_j = \frac{d^2 x_j(\cdot; q, s_i)}{ds_{ik} ds_{il}} : \mathcal{T} \mapsto \mathbb{R}$  are solutions of

$$\begin{aligned} \frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{ds_{ik} ds_{il}} &= \frac{dx(t; q, s_i)^T \partial^2 f_j(x(t), q)}{ds_{ik}} \frac{dx(t; q, s_i)}{ds_{il}} \\ &+ \frac{\partial f_j(x(t), q)}{\partial x} \frac{d^2 x(t; q, s_i)}{ds_{ik} ds_{il}}, \quad \frac{d^2 x_j(t; q, s_i)}{ds_{ik} ds_{il}} = 0. \end{aligned} \quad (\text{A4})$$

For  $k, l = 1, \dots, n_q$  the trajectories  $h_{q_k q_l}^{j}(\cdot; q, s_i) / c_j = \frac{d^2 x_j(\cdot; q, s_i)}{dq_k dq_l} : \mathcal{T} \mapsto \mathbb{R}$  are solutions of

$$\begin{aligned} \frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{dq_k dq_l} &= \frac{dx(t; q, s_i)^T \partial^2 f_j(x(t), q)}{dq_k} \frac{dx(t; q, s_i)}{dq_l} \\ &+ \frac{\partial^2 f_j(x(t), q)}{\partial q_k \partial q_l} + \frac{\partial^2 f_j(x(t), q)}{\partial q_l \partial x} \frac{dx(t; q, s_i)}{dq_k} \\ &+ \frac{\partial^2 f_j(x(t), q)}{\partial q_k \partial x} \frac{dx(t; q, s_i)}{dq_l} \\ &+ \frac{\partial f_j(x(t), q)}{\partial x} \frac{d^2 x(t; q, s_i)}{dq_k dq_l}, \quad \frac{d^2 x_j(t; q, s_i)}{dq_k dq_l} = 0. \end{aligned} \quad (\text{A5})$$

**Proof.** Follows directly from derivatives of the trajectory

$$x(t; q, s_i) = s_i + \int_{t_i}^t f(x(\tau), q) \, d\tau$$

with respect to time and to  $(q, s)$  and from inserting  $t = t_i$  in these expressions to determine the initial values.  $\square$

Note that a numerical integration of the differential equations must be carefully implemented, eg, obeying the principle of internal numerical differentiation.<sup>18,50</sup>