Global optimal control with the direct multiple shooting method

H. Diedam and S. Sager

Otto-von-Guericke-Universität Magdeburg, Germany

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. Copyright © 2016 John Wiley & Sons, Ltd.

1. INTRODUCTION

We are interested in globally optimal solutions of the control problem

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

\[ \min_q c^T x(t_f) \]

subject to

\[ \dot{x}(t) = f(x(t), q), \quad t \in T, \]
\[ x(0) = s_0, \]
\[ x(t) \in X(t), \quad t \in T, \]
\[ q \in Q, \]

on a fixed time horizon \( T = [0, t_f] \) with differential states \( x : T \mapsto \mathbb{R}^n_x \), a finite number of controls \( q \in Q \subseteq \mathbb{R}^n_q \), fixed initial values \( s_0 \in \mathbb{R}^n_0 \), a linear objective function \( c^T x(t_f) \) of Mayer type with non-negative \( c \in \mathbb{R}^n_x^+ \), and bounded convex feasible sets \( Q = [q, \tilde{q}] \subseteq \mathbb{R}^n_q \) for controls and \( X(t) = [x(t), \pi(t)] \subseteq \mathbb{R}^n_x \) 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 \( \frac{\partial f}{\partial x}, \frac{\partial f}{\partial q}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial x \partial q}, \frac{\partial^2 f}{\partial q^2} \) are assumed to be Lipschitz continuous. Furthermore, we assume that there exists at least one feasible \( q^* \in Q \) for (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 four 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, i.e., Pontryagin’s maximum principle. Third, reformulation and relaxation of the control problem as a Generalized Moment Problem, i.e., a linear program defined on a measure space. And fourth, direct approaches that discretize control functions $u : T \mapsto \mathbb{R}^{n_u}$ 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 $x(t_i) \in \mathcal{X}_i \subset \mathbb{R}^{n_x}$, 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 based on how the solution of the differential equations is included in the optimization process. **Direct single shooting** 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 $T$. 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 two. 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 $0 = t_0 < t_1 < \ldots < t_{n_{\text{ms}}} = t_T$ for the evaluation of constraints in single and in 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, though, this is already implied by using a general control vector $q$, compare (1).

**Definition 4 (Solution trajectories)**

For given $q \in \mathbb{R}^{n_q}$, $s_i \in \mathbb{R}^{n_s}$, and $\tau \in T$ 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 $s_0 \in \mathbb{R}^{n_s}$, 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

$$\min_{q} \quad e^{T} x_{\text{ss}}(t_i; q, s_0)$$

subject to

$$x_{\text{ss}}(t_i; q, s_0) \in \mathcal{X}_i, \quad i = 1, \ldots, n_{\text{ms}},$$

$$q \in \mathcal{Q}.$$  \hspace{1cm} \text{(SS)}$	ext{Q} \times \mathcal{X}$

The **$n_{\text{ms}}$ node multiple shooting NLP** for Problem (1) is defined as

$$\min_{s_1, \ldots, s_{n_{\text{ms}}}, q} \quad e^{T} s_{n_{\text{ms}}}$$

subject to

$$s_{i+1} = x_{\text{ms};i}(t_{i+1}; q, s_i), \quad i = 0, \ldots, n_{\text{ms}} - 1$$

$$s_i \in \mathcal{X}_i, \quad i = 1, \ldots, n_{\text{ms}},$$

$$q \in \mathcal{Q},$$

where variables $(s_1, \ldots, s_{n_{\text{ms}}}) \in \mathbb{R}^{n_{\text{ms}} n_s}$ and $n_{\text{ms}} n_s$ matching constraints $s_{i+1} = x_{\text{ms};i}(t_{i+1}; q, s_i)$ have been added in comparison to Problem (SS). Note that the evaluation of $x_{\text{ms};i}(t_{i+1}; q, s_i)$ implies that $x(\cdot)$ is evaluated independently on the time intervals $[t_i, t_{i+1}]$ with different initial values $s_i$. We write $\mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_{n_{\text{ms}}} \subset \mathbb{R}^{n_x n_{\text{ms}}}$ as the cartesian product of all $\mathcal{X}_i$. 
We focus on complete and rigorous methods for global optimization, [Neu04]. The survey [FG09] gives a broad overview over the literature in global optimization in general. Underlying most deterministic global optimization algorithms are interval arithmetics [Moo66], a spatial Branch&Bound technique [LD60, LMSK63], and convex relaxations of the original problem, such as McCormick relaxations [MCB09, SSB11] or polyhedral outer approximations [TS05]. We use the αBB method, [AMF95, AF96, AAF98]. The convex relaxations used in the αBB algorithm are based on the eigenvalues of interval Hessians that are the bounds on the second derivatives. The authors in [SW14] survey different methods to determine bounds on these eigenvalues. We use a method based on Gershgorin’s circle theorem, [Ger31], as suggested in [AAF98].

The novelty compared to previous approaches to globally optimize optimal control problems with the αBB method, see [EF00a, EF00b, PA02, PA05], is that we work with the multiple shooting discretization (MSQ×X). The discretizations (SSQ×X) and (MSQ×X) are obviously equivalent with respect to feasibility and optimality, but lead to different algorithmic behavior, both in local and in global optimization. We show that the multiple shooting approach (MSQ×X) has advantages compared to the single shooting approach (SSQ×X) when used in a global optimization context. Advantages of course relates to computational time, which itself depends on two numbers:

- **The average computational time to solve a problem on a single node.** The advantages of direct multiple shooting in comparison to direct single shooting have been discussed at length in a number of publications, e.g., [BP84, LBS+03, Bet01, Bie10]. 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 [AD10] and to high-rank updates [JKSW16], and improved stability. There are several possibilities to exploit the sparsity of the KKT matrix, such that the linear algebra is not significantly more expensive than in the single shooting case, [Fra14]. 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 15 that the right branching strategy guarantees that there is no increase in the number of nodes. We show in Theorem 17 under which assumptions the quality of the convex under- and concave overestimations, which can be expressed in terms of the relaxation factors α, 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 αBB relaxation of a composite function and the composition of several αBB relaxations, which in general do not commute. For the case of an additional end-point which can be seen as a two point multiple shooting this was discussed in [PA02].

The paper is organized as follows. In Section 2 we survey mathematical preliminaries, in particular a basic αBB algorithm and second order derivatives of trajectories. In Section 3 we formulate the specific relaxations for the single and multiple shooting optimization problems (SSQ×X) and (MSQ×X). In Section 4 we prove the superiority of (MSQ×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 examplarily the performance gain for a benchmark problem from the literature. In Section 6 we discuss applicability to more general optimal control problems than (1) and algorithmic extensions. We conclude in Section 7 with a summary.

## 2. PRELIMINARIES

For convenience of the reader, we shortly summarize useful notation 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 $\phi : [\underline{v}, \overline{v}] \subset \mathbb{R}^{n_v} \rightarrow \mathbb{R}$ can be underestimated by a convex function
\[
\phi_{cv}(v) := \phi(v) + \sum_{k=1}^{n_v} \alpha_{cv,k} (\overline{v}_k - v_k)(\underline{v}_k - v_k),
\]
and overestimated by a concave function
\[
\phi_{cc}(v) := \phi(v) - \sum_{k=1}^{n_v} \alpha_{cc,k} (\overline{v}_k - v_k)(\underline{v}_k - v_k),
\]
for any $v$ in the bounded box $[\underline{v}, \overline{v}]$, 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
\[
\psi(v_k) := (\overline{v}_k - v_k)(\underline{v}_k - v_k)
\]
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 non-negative (non-positive). Necessary conditions are hence
\[
\alpha_{cv,k} \geq -\frac{1}{2} \min \left( 0, \lambda_{\min}(\nabla^2 \phi(v)) \right),
\]
\[
\alpha_{cc,k} \geq -\frac{1}{2} \min \left( 0, -\lambda_{\max}(\nabla^2 \phi(v)) \right).
\]

One cheap, yet successful way [SW14] to calculate sufficiently large $\alpha_{cv,k}, \alpha_{cc,k}$ is based on Gershgorin’s circle theorem, [Ger31], which implies: 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

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{V}$</td>
<td>$[\underline{v}, \overline{v}]$</td>
<td>Interval between lower and upper bounds $\underline{v}, \overline{v} \in \mathbb{R}^{n_v}$</td>
</tr>
<tr>
<td>$</td>
<td>\mathcal{V}</td>
<td>$</td>
</tr>
<tr>
<td>$(s_0, s_1, \ldots, s_m)$</td>
<td>Concatenated vector in $\mathbb{R}^n$ with $n = n_{s_0} + \ldots + n_{s_m}$</td>
<td></td>
</tr>
<tr>
<td>sol(NLP)</td>
<td>$v^*$</td>
<td>Optimal solution of (NLP)</td>
</tr>
<tr>
<td>val(NLP)</td>
<td>$\phi^{obj}(v^*)$</td>
<td>Optimal objective function value of (NLP)</td>
</tr>
<tr>
<td>$\psi(v_k)$</td>
<td>See (4)</td>
<td>Convexification term $(\overline{v}_k - v_k)(\underline{v}_k - v_k)$ for variable $v_k$</td>
</tr>
<tr>
<td>$\Phi(\mathcal{V})$</td>
<td>val(NLP$^{cv}$)</td>
<td>Lower bound on sol(NLP)</td>
</tr>
<tr>
<td>$\delta_{jl}$</td>
<td>-</td>
<td>Kronecker symbol, $\delta_{jl} = 1$ if $j = l$, $\delta_{jl} = 0$ if $j \neq l$</td>
</tr>
</tbody>
</table>

Table I. The notation used in this paper.
\[ k = 1, \ldots, n_v \text{ and all } v \in [\underline{v}, \overline{v}] \]

\[ \alpha_{cv,k} \geq -\frac{1}{2} \min_{v \in [\underline{v}, \overline{v}]} \left( 0, \sum_{l \neq k} \left| \left( \nabla^2 \phi(v) \right)_{kl} \right| \right), \tag{7} \]

\[ \alpha_{cc,k} \geq -\frac{1}{2} \min_{v \in [\underline{v}, \overline{v}]} \left( 0, -\sum_{l \neq k} \left| \left( \nabla^2 \phi(v) \right)_{kl} \right| \right) \tag{8} \]

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

**Definition 6 (Interval arithmetics)**

We denote closed intervals by calligraphic uppercase characters,

\[ \mathcal{V} := [\underline{v}, \overline{v}] = \{ v \in \mathbb{R} : \underline{v} \leq v \leq \overline{v} \} \tag{9} \]

with lower bound \( \underline{v} \in \mathbb{R} \) and upper bound \( \overline{v} \in \mathbb{R} \), indicated by underlined and overlined variables. The absolute value of an interval \( \mathcal{V} \) is \( |\mathcal{V}| := \max(|\underline{v}|, |\overline{v}|) \) and the set of closed intervals is defined by \( [\mathbb{R}] := \{ [v, \overline{v}] : v \leq \overline{v}, v, \overline{v} \in \mathbb{R}, \overline{v} \in \mathbb{R} \} \). We use the straightforward extensions to higher dimensions, i.e., to interval matrices \( A \in [\mathbb{R}]^{m \times n} \), and to functions \( \Phi : [\mathbb{R}] \to [\mathbb{R}] \) as an extension of \( \Phi : \mathbb{R} \to \mathbb{R} \). All interval arithmetics are performed componentwise and result in interval components. We refer to the textbook [MKC09] for an introduction to calculus with intervals.

Making use of the obvious relations

\[
\min_{v \in [\underline{v}, \overline{v}]} \left( \sum_{l \neq k} \left| \left( \nabla^2 \phi(v) \right)_{kl} \right| \right) \leq \min_{v \in [\underline{v}, \overline{v}]} \left( \sum_{l \neq k} \left| \left( \nabla^2 \phi(v) \right)_{kl} \right| \right) - \min_{v \in [\underline{v}, \overline{v}]} \left( \sum_{l \neq k} \left| \left( \nabla^2 \phi(v) \right)_{kl} \right| \right)
\]

we overestimate (7-8) further and get for \( \mathcal{V} = [\underline{v}, \overline{v}] \) the factors

\[ \alpha_{cv,k} = \frac{1}{2} \min_{v \in [\underline{v}, \overline{v}]} \left( 0, \sum_{l \neq k} \left| \left( \nabla^2 \phi(\mathcal{V}) \right)_{kl} \right| \right), \tag{10} \]

\[ \alpha_{cc,k} = \frac{1}{2} \min_{v \in [\underline{v}, \overline{v}]} \left( 0, -\sum_{l \neq k} \left| \left( \nabla^2 \phi(\mathcal{V}) \right)_{kl} \right| \right) \tag{11} \]

with the notation introduced in Definition 6, i.e.,

\[ \left| \left( \nabla^2 \phi(\mathcal{V}) \right)_{kl} \right| = \max \left( \left| \left( \nabla^2 \phi(\mathcal{V}) \right)_{kl} \right|, \left| \left( \nabla^2 \phi(\mathcal{V}) \right)_{kl} \right| \right). \]

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

\[
\min_{v} \phi^{obj}(v) \\
\text{subject to} \quad 0 \geq \phi^{ineq}(v), \quad 0 = \phi^{eq}(v), \quad v \in \mathcal{V} = [\underline{v}, \overline{v}] \tag{NLP} \]

Copyright © 2016 John Wiley & Sons, Ltd.  \hspace{1cm}  Optim. Control Appl. Meth. (2016)  
Prepared using ocaauth.cls  \hspace{1cm}  DOI: 10.1002/oca
is underestimated by the optimal objective function value of the convex NLP

\[
\begin{align*}
\min_{v} & \quad \phi_{\text{obj}}(v) \\
\text{subject to} & \quad 0 \geq \phi_{\text{ineq}}^{\text{cv}}(v), \\
& \quad 0 \geq \phi_{\text{eq}}^{\text{cv}}(v), \\
& \quad 0 \geq -\phi_{\text{eq}}^{\text{cc}}(v), \\
& \quad v \in V = [\underline{v}, \overline{v}]
\end{align*}
\]

where the subscripts \(\text{cv}\) and \(\text{cc}\) refer to the \(\alpha\)BB relaxations (2-3), assumed that for all functions the corresponding values \(\alpha \in \mathbb{R}_{+}^{n_v}\) have been calculated via (10-11) 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:

\[\text{val}(\text{NLP}_{\text{cv}}) \leq \text{val}(\text{NLP}).\]

Branch&Bound methods are described in detail in [Sch11]. For an early application in global optimization, we refer to [FS69], whereas our implementation is based on [HT96, TH88]. The extension to nonconvex constraints is discussed in [SKS13]. A basic \(\alpha\)BB algorithm for a nonconvex (NLP) is given in Algorithm 1.

**Algorithm 1:** The basic \(\alpha\)BB algorithm for (NLP).

**Initialize** counter \(i := 0\), interval \(V_0 = V = [\underline{v}, \overline{v}]\) and set of open problems \(P := \{\text{NLP}_{V_0}\}\).

**Initialize** lower bound \(\Phi := \text{val}(\text{NLP}_{\text{cv}})\) and upper bound \(\overline{\Phi} := \text{val}(\text{NLP}_{V_0})\).

**while** \(P \neq \emptyset\) and \(\overline{\Phi} - \Phi > \epsilon\) **do**

**Increment** iteration counter \(i := i + 1\).

**Select** open problem \((\text{NLP}_{V_i}) \in P\) with feasible domain \(V_i\).

**Branch** \(V_i\) into a partition \(V_i = \bigcup_{k \in \{1, \ldots, N\}} V_{ik}\).

**for** \(k = 1\) **to** \(N\) **do**

\[\text{Solve} \ (\text{NLP}_{V_{ik}})\] locally.

\[\text{if} \ \text{val}(\text{NLP}_{V_{ik}}) < \overline{\Phi} \ \text{then}\]

\[\text{Update} \ \text{upper bound} \ \overline{\Phi} := \text{val}(\text{NLP}_{V_{ik}}) \text{ and corresponding } v^* = \text{sol}(\text{NLP}_{V_{ik}}).\]

**end**

**Construct** convex relaxation \((\text{NLP}_{V_{ik}})\).

**Solve** \((\text{NLP}_{V_{ik}})\) locally (= globally).

**end**

**Update** open problems \(P := P \setminus (\text{NLP}_{V_i}) \bigcup_{k \in \{1, \ldots, N\}} (\text{NLP}_{V_{ik}})\).

**Update** open problems \(P := P \setminus (\text{NLP})\) for all \((\text{NLP}) \in P\) with \(\overline{\Phi} \leq \text{val}(\text{NLP}_{V_{ik}})\).

**Update** global lower bound \(\Phi := \min_{(\text{NLP}) \in P} \text{val}(\text{NLP})\).

**end**

To construct the convex relaxations \((\text{NLP}_{V_{ik}})\) 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 \((\text{SS}_{Q \times X})\) and \((\text{MS}_{Q \times X})\), we need the second derivatives of the occurring functions with respect to the decision variables, compare (11), in particular of the control trajectories \(x(\cdot)\) with respect to \(q\) for \((\text{SS}_{Q \times X})\) and with respect to \((q, s)\) for \((\text{MS}_{Q \times X})\).
Definition 8 (Second order sensitivities)
Let \( j \in \{1, \ldots, n_x\} \) be fix and a bounded box \( Q \subseteq \mathbb{R}^{n_s} \) be given. We define the single shooting second order sensitivity interval of the trajectory \( x_j^q(t) \) with respect to \( q \in \mathbb{R}^{n_s} \), scaled by the linear objective function vector \( c \in \mathbb{R}^{n_r} \), as

\[
\mathcal{H}^{\text{ss}, i, j}_{q, s}(t; Q, s_0) := c_j \frac{d^2 x_j^q(t; Q, s_0)}{dq_k dq_l} \subseteq \mathbb{R}, \quad k, l \in \{1, \ldots, n_q\}. \tag{12}
\]

Let in addition \( i \in \{1, \ldots, n_{ms}\} \) and a bounded box \( S_i \subseteq X_i \subseteq \mathbb{R}^{n_s} \) be given. We define the multiple shooting second order sensitivity interval of the trajectory \( x_j^{ms,i}(t) \) with respect to \( q \in Q \) and \( s_i \in S_i \) for \( t \in [t_i, t_{i+1}] \), scaled by the linear objective function vector \( c \in \mathbb{R}^{n_r} \), as

\[
\mathcal{H}^{\text{ms}, i, j}_{q, s_i}(t; Q, S_i) := c_j \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{dq_k dq_l} \subseteq \mathbb{R}, \quad k, l \in \{1, \ldots, n_q\} \tag{13}
\]

\[
\mathcal{H}^{\text{ms}, i, j}_{s_i, k, s_i}(t; Q, S_i) := c_j \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{ds_k ds_l} \subseteq \mathbb{R}, \quad k, l \in \{1, \ldots, n_x\} \tag{14}
\]

\[
\mathcal{H}^{\text{ms}, i, j}_{s_i, k, s_i}(t; Q, S_i) := c_j \frac{d^2 x_j^{ms,i}(t; Q, S_i)}{dq_k ds_l} \subseteq \mathbb{R}, \quad k \in \{1, \ldots, n_q\}, l \in \{1, \ldots, n_x\} \tag{15}
\]

We obtain the functions \( h^{i, j}(\cdot; q, s_i) \) and their interval counterparts \( \mathcal{H}^{\cdot, i, \cdot}_{\cdot, \cdot}(\cdot; Q, S_i) \) as solution trajectories of a system of differential equations. The following result is well known and used in the context of dynamic optimization, e.g., [Boc87].

Lemma 9 (First and second order sensitivities)
Let Assumption 2 hold, indices \( i \in \{0, \ldots, n_{ms} - 1\} \) and \( j \in \{1, \ldots, n_x\} \) 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 \( k = 1, \ldots, n_q \) and \( l = 1, \ldots, n_x \) the trajectories \( \frac{dx_j^{i, q}(\cdot; q, s_i)}{dq_k} : \mathcal{T} \to \mathbb{R} \) and \( \frac{dx_j^{i, q}(\cdot; q, s_i)}{ds_{il}} : \mathcal{T} \to \mathbb{R} \) are solutions of the differential equations

\[
\frac{d}{dt} \frac{dx_j(t; q, s_i)}{dq_k} = \frac{\partial f_j(x(t), q)}{\partial x} \frac{dx(t; q, s_i)}{dq_k} + \frac{\partial f_j(x(t), q)}{\partial q_k} \frac{dx(t; q, s_i)}{dq_k}, \quad \frac{dx_j(t; q, s_i)}{dq_k} = 0, \quad k = 1, \ldots, n_q, l = 1, \ldots, n_x. \tag{16}
\]

\[
\frac{d}{dt} \frac{dx_j(t; q, s_i)}{ds_{il}} = \frac{\partial f_j(x(t), q)}{\partial x} \frac{dx(t; q, s_i)}{ds_{il}}, \quad \frac{dx_j(t; q, s_i)}{ds_{il}} = \delta_{ji}, \quad j = 1, \ldots, n_x. \tag{17}
\]

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

\[
\frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{dq_k ds_{il}} = \frac{dx(t; q, s_i)}{dq_k} \frac{\partial^2 f_j(x(t), q)}{\partial x^2} \frac{dx(t; q, s_i)}{ds_{il}} + \frac{\partial f_j(x(t), q)}{\partial x} \frac{d^2 x_j(t; q, s_i)}{dq_k ds_{il}}, \quad \frac{d^2 x_j(t; q, s_i)}{dq_k ds_{il}} = 0. \tag{18}
\]

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

\[
\frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{ds_{ik} ds_{il}} = \frac{dx(t; q, s_i)}{ds_{ik}} \frac{\partial^2 f_j(x(t), q)}{\partial x^2} \frac{dx(t; q, s_i)}{ds_{il}} + \frac{\partial f_j(x(t), q)}{\partial x} \frac{d^2 x_j(t; q, s_i)}{ds_{ik} ds_{il}}, \quad \frac{d^2 x_j(t; q, s_i)}{ds_{ik} ds_{il}} = 0. \tag{19}
\]
For \( k, l = 1, \ldots, n_q \) the trajectories \( \hat{h}^{s,j}_{q_k,q_l}(; q, s_i) / c_j = \frac{d^2x_j(q,s_i)}{dq_kdq_l} : \mathcal{T} \mapsto \mathbb{R} \) are solutions of

\[
\frac{d}{dt} \frac{d^2 x_j(t; q, s_i)}{dq_kdq_l} = \frac{dx(t; q, s_i)}{dt}^T \frac{\partial^2 f_j(x(t), q)}{\partial x \partial q} \frac{dx(t; q, s_i)}{dq_l} + \frac{\partial^2 f_j(x(t), q)}{\partial q_k \partial x} \frac{dx(t; q, s_i)}{dq_l} + \frac{\partial^2 f_j(x(t), q)}{\partial q_k \partial x} \frac{d^2x(t; q, s_i)}{dq_kdq_l},
\]

(20)

\[
\frac{d^2 x_j(t_i; q, s_i)}{dq_kdq_l} = 0.
\]

**Proof**

Follows directly from derivatives of the trajectory

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

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

\[
\square
\]

### 3. CONVEX RELAXATIONS OF SHOOTING PROBLEMS

In this section we formulate relaxed, convex versions of \((SS_{Q \times X}^3)\) and \((MS_{Q \times X}^3)\) in the spirit of \((NLP_{cv}^3)\) and show how Algorithm 1 can be applied.

**Definition 10** (Convexification of Single Shooting NLP)

For given data from Problem \((SS_{Q \times X}^3)\) as a discretization of Problem (1) the *convexified single shooting NLP* is defined as

\[
\begin{align*}
\min_q & \quad c^T x_{cv}^{ss}(t; q, s_0, Q) \\
\text{subject to} & \quad x_{cv}^{ss}(t_i; q, s_0, Q) \leq \mathcal{F}_i, \quad i = 1, \ldots, n_m, \\
& \quad x_{cv}^{ss}(t_i; q, s_0, Q) \geq \mathcal{G}_i, \quad i = 1, \ldots, n_m, \\
& \quad q \in Q.
\end{align*}
\]

(\(SS_{cv}^{Q \times X}\))

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

\[
x_{cv}^{ss}(t_i; q, s_0, Q) := x_{cv}^{ss}(t_i; q, s_0) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ss,i,j} \psi(q_k),
\]

(21)

\[
x_{cc}^{ss}(t_i; q, s_0, Q) := x_{cc}^{ss}(t_i; q, s_0) - \sum_{k=1}^{n_q} \alpha_{cc,k}^{ss,i,j} \psi(q_k),
\]

(22)

where \( x^{ss}(; q, s_0) \) is the solution trajectory of \( \dot{x}(t) = f(x(t), q) \) with \( x(0) = s_0 \) on \([0, t_i]\). The relaxation factors \( \alpha_{cv,k}^{ss}, \alpha_{cc,k}^{ss} \in \mathbb{R}^{n_m \times n_q} \) are determined based on (10-11) and Definition 8 as

\[
\alpha_{cv,k}^{ss,i,j} = -\frac{1}{2} \min \left( \frac{0, \mathcal{H}_{cv,k}^{ss,j}(t_i; Q, s_0)}{\sum_{l \neq k} \mathcal{H}_{cv,k}^{ss,j}(t_i; Q, s_0)} \right),
\]

(23)

\[
\alpha_{cc,k}^{ss,i,j} = -\frac{1}{2} \min \left( \frac{0, -\mathcal{H}_{cv,k}^{ss,j}(t_i; Q, s_0)}{\sum_{l \neq k} \mathcal{H}_{cv,k}^{ss,j}(t_i; Q, s_0)} \right),
\]

(24)

for \( k = 1, \ldots, n_q \) as functions of the box \( Q \).
Lemma 11 (Convex single shooting relaxation)
Problem \((SS_{cv}^{Q×X})\) is a convex relaxation of Problem \((SS^{Q×X})\) and for \(q^* = \text{sol}(SS_{cv}^{Q×X})\) we have

\[
\text{val}(SS_{cv}^{Q×X}) = c^T x_{ss}(t_i; q^*, s_0) + \sum_{j=1}^{n_x} \sum_{k=1}^{n_q} c_j \alpha_{cv,k}^{ss,i,j}(\eta_k - q_k^*)(q_k - \hat{q}_k)
\]

Proof
Constraints and objective function of Problem \((SS_{cv}^{Q×X})\) are convex by construction. Assume \(\hat{q} \in \mathbb{R}^{n_q}\) is feasible for Problem \((SS^{Q×X})\). Because of
\[
x_{cc}(t_i; \hat{q}, s_0, \mathcal{Q}) \geq x_{ss}(t_i; \hat{q}, s_0) \geq x_i \quad \text{and} \quad x_{cc}(t_i; \hat{q}, s_0, \mathcal{Q}) \leq x_{ss}(t_i; \hat{q}, s_0) \leq \bar{x}_i,
\]
it is also feasible for \((SS_{cv}^{Q×X})\).

The objective function \(c^T x\) is assumed to be linear, and is hence convex. Using the convex underestimation for the trajectory, and \(c \in \mathbb{R}^{n_x}_+\) non-negative, we obtain
\[
c^T x_{ss}(t_i; \hat{q}, s_0, \mathcal{Q}) \leq c^T x_{ss}(t_i; \hat{q}, s_0)
\]
for all feasible \(\hat{q} \in \mathbb{R}^{n_q}\). Thus Problem \((SS_{cv}^{Q×X})\) is a convex relaxation of Problem \((SS^{Q×X})\).

The objective function value is bounded because of the boundedness of \(\mathcal{Q}\) and exists because of Assumption 2. It can be calculated as
\[
\text{val}(SS_{cv}^{Q×X}) = c^T x_{ss}(t_i; q^*, s_0, \mathcal{Q}) = \sum_{j=1}^{n_x} c_j \left( x_{ss,j}(t_i; q^*, s_0) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ss,i,j} (\eta_k - q_k^*)(q_k - \hat{q}_k) \right)
\]
which concludes the proof. 

\(\square\)

Definition 12 (Convexification of Multiple Shooting NLPs)
For given data from Problem \((MS_{cv}^{Q×X})\) as a discretization of Problem \((1)\) the convexified \(n_{ms}\) node multiple shooting NLP is defined as

\[
\begin{align*}
\min_{{s_1,...,s_{n_{ms}},q}} & \quad c^T s_{ms} \\
\text{subject to} & \quad x_{ms,i}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, S_{i-1}), \; i = 1,\ldots,n_{ms} \\
& \quad x_{cc}^{ms,i-1}(t_i; q, s_{i-1}, \mathcal{Q}, S_{i-1}), \; i = 1,\ldots,n_{ms} \\
& \quad s_i \in S_i := [x_i, \bar{x}_i], \; i = 1,\ldots,n_{ms} \\
& \quad q \in \mathcal{Q},
\end{align*}
\]

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

\[
x_{ms,i}^{ms,i}(t_{i+1}; q, s_i, \mathcal{Q}, S_i) := x_{ms,i}^{ms,i}(t_{i+1}; q, s_i) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,i,j} \psi(q_k) + \sum_{l=1}^{n_x} \alpha_{cv,n_q+l}^{ms,i,j} \psi(s_{il}),
\]

\[
x_{cc}^{ms,i}(t_{i+1}; q, s_i, \mathcal{Q}, S_i) := x_{cc}^{ms,i}(t_{i+1}; q, s_i) - \sum_{k=1}^{n_q} \alpha_{cc,k}^{ms,i,j} \psi(q_k) - \sum_{l=1}^{n_x} \alpha_{cc,n_q+l}^{ms,i,j} \psi(s_{il}),
\]

where \(x_{ms,i}^{ms,i}(\cdot; q, s_0)\) 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 \(i = 1,\ldots,n_{ms} - 1\) based on \((10-11)\) and Definition 8. Note that they are functions of \(\mathcal{Q}\) and \(S_i\), not of \(q\) and \(s_i\). This follows from the dependencies of the second order sensitivities \(\mathcal{H}_{i}^{ms,i,j}(t_{i+1}) := \mathcal{H}_{i}^{ms,i,j}(t_{i+1}; \mathcal{Q}, S_i)\), although
we omit this important implicit dependency on \((Q, S_1)\) for a more compact notation:

\[
\alpha_{cv,k}^{ms,i,j} := -\frac{1}{2} \min_{0, t \neq k} \left(0, \frac{\partial H_{qs,qi}^{ms,i,j}(t_{i+1})}{\partial t} - \sum_{l \neq k} |H_{qs,qi}^{ms,i,j}(t_{i+1})| - \sum_{l=1}^{n_x} \left|H_{q_k, s_i}^{ms,i,j}(t_{i+1})\right| \right), \quad (27)
\]

\[
\alpha_{cc,k}^{ms,i,j} := -\frac{1}{2} \min_{0, t \neq k} \left(0, -\frac{\partial H_{qs,qi}^{ms,i,j}(t_{i+1})}{\partial t} + \sum_{l \neq k} |H_{qs,qi}^{ms,i,j}(t_{i+1})| + \sum_{l=1}^{n_x} \left|H_{q_k, s_i}^{ms,i,j}(t_{i+1})\right| \right), \quad (28)
\]

for \(k = 1, \ldots, n_q\) and

\[
\alpha_{cv,n_q+l}^{ms,i,j} := -\frac{1}{2} \min_{0, t \neq k} \left(0, H_{qs,qi}^{ms,i,j}(t_{i+1}) - \sum_{k=1}^{n_q} |H_{qs,qi}^{ms,i,j}(t_{i+1})| - \sum_{k \neq l}^{n_q} |H_{q_k, s_i}^{ms,i,j}(t_{i+1})| \right), \quad (29)
\]

\[
\alpha_{cc,n_q+l}^{ms,i,j} := -\frac{1}{2} \min_{0, t \neq k} \left(0, -H_{qs,qi}^{ms,i,j}(t_{i+1}) + \sum_{k=1}^{n_q} |H_{qs,qi}^{ms,i,j}(t_{i+1})| + \sum_{k \neq l}^{n_q} |H_{q_k, s_i}^{ms,i,j}(t_{i+1})| \right), \quad (30)
\]

for \(l = 1, \ldots, n_x\). 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 one-dimensional illustration of the relaxation concept. We show that in general we obtain a convex relaxation.

![Figure 1. One-dimensional illustration of the relaxed matching conditions in Problem (MS\(_Q^Q\times X\)).](image)

Note that for the NLP we only needed to define the under- 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 \(x_{cv}^{ms,i}(t; q, s_i, Q, S_i)\) gives a pointwise lower bound on \(x_{cv}^{ms,i}(t; q, s_i)\) for all times \(t \in [t_i, t_{i+1}]\), but is itself not necessarily monotone with respect to \(t\).

**Lemma 13 (Convex multiple shooting relaxation)**

Problem (MS\(_{cv}^QQ \times X\)) is a convex relaxation of Problem (MS\(_Q^Q \times X\)).

**Proof**

Convexity of Problem (MS\(_{cv}^QQ \times X\)) follows from the constructed convexity of the functions in the
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 \( Q \). Assume that we have \( \hat{q} = \hat{q} = \tilde{q} \) for some \( \tilde{q} \in \mathbb{R}^{n_q} \). With the fixed initial value we have \( s_0 = \tilde{s}_0 = s_0 \).

Therefore
\[
\sum_{j=1}^{n_s} \phi(s_{ij}) = \frac{1}{4} \sum_{j=1}^{n_s} (s_{ij})^2 \geq -C \left( \frac{1}{4} \| \tilde{q} - q \|^2 \right)
\]

for all \( i = 1, \ldots, n_{ms} \).

**Proof**

We will use
\[
\phi(v) = (v - v)(v - v) \geq -\frac{1}{4} (v - v)^2,
\]

the update in Algorithm 3 which gives bounds on \( \tilde{s}_1 \) and \( s_{ij} \), the definitions (25–26), and the fact that all \( \phi \) are continuous functions of the bounds \( q, \tilde{q}, \tilde{s}, \tilde{s} \), and hence have a maximum on this bounded set.

**Corollary 15 (No need to branch on \( s \))**

Algorithm 1, using Algorithm 3 and branching exclusively on the set \( \mathcal{Q} = \mathcal{Q} \times \mathcal{S} \), 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 the 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 16 that is conceptually visualized in Figure 2 in detail.

**Assumption 16 (Basic assumption on dynamics and shooting grid)**

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

\[
\begin{align*}
    x_{cv}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}, Q, S_{m+1}) &\geq x_{cv}^{ms,m}(t_{m+2}; q^*, s_{m}, Q, S_{m}), \\
    x_{cc}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}, Q, S_{m+1}) &\leq x_{cc}^{ms,m}(t_{m+2}; q^*, s_{m}, Q, S_{m}).
\end{align*}
\]

(32) \hspace{1cm} (33)

**Theorem 17 (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 10 and 12. 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 16 hold. Then multiple shooting provides a tighter relaxation, i.e.,

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

**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 \((\text{MS}^{Q \times X}_{cv})\). Because of

\[
\begin{align*}
    x_{cv}^{ss}(t_1; q^*, s_0, Q) &= x_{cv}^{ms,0}(t_1; q^*, s_0, Q) \leq s^*_1 \leq x_1 = x^*_1, \\
    x_{cc}^{ss}(t_1; q^*, s_0, Q) &= x_{cc}^{ms,0}(t_1; q^*, s_0, Q) \geq s^*_1 \geq s_1 \geq x_1,
\end{align*}
\]

(34) \hspace{1cm} (35)

\(q^*\) is a feasible solution of \((\text{SS}^{Q \times X}_{cv})\). 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, \(s^*_1 = x_{cv}^{ms,0}(t_1; q^*, s_0, Q)\). Therefore

\[
\text{val}(\text{MS}^{Q \times X}_{cv}) = c^T s^*_1 = c^T x_{cv}^{ms,0}(t_1; q^*, s_0, Q) = \text{val}(\text{SS}^{Q \times X}_{cv}).
\]

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

\[
s' := (s_1, \ldots, s_m, s_{m+2}, \ldots, s_{n_{ms}}) \in \mathbb{R}^{n_x(n_{ms}-1)}, \quad \mathcal{X}' := \mathcal{X}_1 \times \ldots \times \mathcal{X}_m \times \mathcal{X}_{m+2} \times \ldots \times \mathcal{X}_{n_{ms}}
\]

Figure 2. One-dimensional illustration of Assumption 16.
and define the corresponding convexified $n_{ms} - 1$ node multiple shooting NLP as

$$\min_{q, s^c} c^T s_{ms}$$

subject to

$$s_i \geq x_{cv}^{ms,i-1}(t_i; q, s_{i-1}, Q, S_{i-1}), \quad i = 1, \ldots, m, m + 3, \ldots, n_{ms}$$

$$(MS_{cv}^{Q \times X'})$$

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

$$s_i \leq x_{cc}^{ms,i-1}(t_i; q, s_{i-1}, Q, S_{i-1}), \quad i = 1, \ldots, m, m + 3, \ldots, n_{ms}$$

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

$$s_i \in S_i := [x_i, x'_i], \quad i = 1, \ldots, m, m + 2, \ldots, n_{ms},$$

$$q \in Q.$$  

This is obviously again a multiple shooting formulation like in Definition 12 and hence a convex relaxation of $(MS_{cv}^{Q \times X'})$. To finish the proof we have to show that $\text{val}(MS_{cv}^{Q \times X'}) \leq \text{val}(MS_{cv}^{Q \times X})$, i.e., 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 $(MS_{cv}^{Q \times X'})$. Then $(q^*, s^*)$ is a feasible solution of $(MS_{cv}^{Q \times X'})$, if

$$x_{cv}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, Q, S_{m+1}) \geq x_{cv}^{ms,m}(t_{m+2}; q^*, s_{m}^*, Q, S_{m}), \quad (36)$$

$$x_{cc}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*, Q, S_{m+1}) \leq x_{cc}^{ms,m}(t_{m+2}; q^*, s_{m}^*, Q, S_{m}), \quad (37)$$

because this implies a larger feasible region $S_{m+2} = [s_{m+2}^l, s_{m+2}^u]$ 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 $S_i, i = m + 3, \ldots, n_{ms}$. The inequalities (36-37) follow from (32-33) in Assumption 16, concluding the proof. \qed

We now have a closer look at the crucial Assumption 16. We focus on the underestimation (32). Making use of the definition (25) of convex underestimation the assumed inequalities are for triplets $(t_m, t_{m+1}, t_{m+2})$ of neighboring points

$$x_{j}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}^*) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m+1,j} \psi(q_k^*) + \sum_{l=1}^{n_s} \alpha_{cv,n_q+l}^{ms,m+1,j} \psi(s_{m+1+l}^*) \geq x_{j}^{ms,m}(t_{m+2}; q^*, s_{m}^*) + \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m,j} \psi(q_k^*) + \sum_{l=1}^{n_s} \alpha_{cv,n_q+l}^{ms,m,j} \psi(s_{m+l}^*) \quad (38)$$

for all $j = 1, \ldots, n_x$ 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, i.e., for varying $q^*, s^*, Q, S$.

Still, the inequalities (38) 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 based on trial and error, a simulation may validate that the inequalities (38) 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, though.
We continue by motivating why (38) is plausible to hold for many nodes even for random and for equidistant multiple shooting grids. As derived in the proof of Theorem 14 we know that the bounds of the multiple shooting nodes have a fourth order convergence, i.e.,

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

This implies that at some part in the tree the bounds $s, \pi$ are so tight that

\[
s_{m+1} \approx s_{m+1} \approx s_{m+1} \approx x_{j}^{ms,m}(t_{m+1}; q^*, s_m^*)
\]

and therefore

\[
x_{j}^{ms,m+1}(t_{m+2}; q^*, s_{m+1}) \approx x_{j}^{ms,m}(t_{m+2}; q^*, s_m^*). \tag{39}
\]

With the same argument, the terms

\[
\sum_{l=1}^{n_q} \alpha_{cc,m+1,l}^{ms,m+1,j} \psi(s_{m+l}) \quad \text{and} \quad \sum_{l=1}^{n_q} \alpha_{cv,m+1,l}^{ms,m+1,j} \psi(s_{m+l}) \tag{40}
\]

will be dominated by the relaxation terms with respect to $q$. Thus we may neglect the first and the third terms in (38) as we approach tighter bounds around the optimal solution, where it is well known that most of the computational workload is concentrated due to the clustering effect, compare [DK94, SN04, WSB14]. Hence the most important part of (38) is the relation

\[
\sum_{k=1}^{n_q} \alpha_{cc,k}^{ms,m+1,j} \psi(q_k^*) \geq \sum_{k=1}^{n_q} \alpha_{cv,k}^{ms,m,j} \psi(q_k^*). \tag{41}
\]

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

\[
\alpha_{cc,k}^{ms,m+1,j} \leq \alpha_{cv,k}^{ms,m,j} \tag{42}
\]

for all $k = 1, \ldots, n_q, j = 1, \ldots, n_x$ and the considered $q^*, s^*, Q, S, t_m, t_{m+1}$. These values $\alpha_{cc,k}^{ms,m+1,j} \in \mathbb{R}$ are given by (27) and Definition 8 as $-\frac{1}{2}$ times the minimum between 0 and

\[
c_j \frac{d^2 x_{j}^{ms,m+1}(t_{m+2})}{dq_i dq_k} - \sum_{l \neq j}^{n_q} c_j \frac{d^2 x_{j}^{ms,m+1}(t_{m+2})}{dq_i dq_l} - \sum_{l=1}^{n_q} c_l \frac{d^2 x_{j}^{ms,m+1}(t_{m+2})}{dq_l ds_{m+1}}. \tag{43}
\]

Due to the dependence of (43) 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 (38) hold. However, we note that a reduction of the absolute values of the second derivatives at time $t_{m+2}$ is likely to result in (42) which implies (41) which makes (38) 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 9. Although there may be counterexamples where this actually leads to an increase of the values in (43), 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 [Luu90, EF00a, CL04, SB06, LS07a]. 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

\[
\begin{align*}
\min_{u(t)} \quad & x_4(t_f) \\
\text{subject to} \quad & \dot{x}_1(t) = x_2(t) \quad t \in [0, t_f] \\
& \dot{x}_2(t) = -x_3(t)u(t) + 16t - 8 \quad t \in [0, t_f] \\
& \dot{x}_3(t) = u(t) \quad t \in [0, t_f] \\
& \dot{x}_4(t) = x_1(t)^2 + x_2(t)^2 + \frac{5}{18} \left( x_2(t) + 16x_5(t) - 8 - \frac{x_3(t)u(t)^2}{10} \right)^2 \quad t \in [0, t_f] \\
& \dot{x}_5(t) = 1 \quad t \in [0, t_f] \\
x(0) = (0, -1, -\sqrt{5}, 0, 0) \\
u(t) \in [-4, 10] \\
\end{align*}
\]

We discretize the control function.

**Definition 18** (discretization of the control function)

We discretize the control function \( u : [0, t_f] \to \mathbb{R} \) via piecewise constant functions on an equidistant grid, depending on the total number \( n_q \) of intervals, i.e.,

\[
u(t) = q_k \quad \forall \ t \in [t_{k-1}, t_k), \ k = 1, \ldots, n_q. \quad (45)\]

We consider optimal solutions for (44) based on \( n_q \) discretizations with \( n_q = 1, \ldots, 4 \). The optimal objective function values and control values found by our algorithms are listed in Table II. They coincide with the values from [LS07a].

<table>
<thead>
<tr>
<th>( n_q )</th>
<th>( x_4^*(1) )</th>
<th>( q^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4965</td>
<td>(4.0709)</td>
</tr>
<tr>
<td>2</td>
<td>0.2771</td>
<td>(5.5748, -4.0000)</td>
</tr>
<tr>
<td>3</td>
<td>0.1475</td>
<td>(8.0015, -1.9438, 6.0420)</td>
</tr>
<tr>
<td>4</td>
<td>0.1237</td>
<td>(9.7890, -1.1997, 1.2566, 6.3558)</td>
</tr>
</tbody>
</table>

Table II. Globally optimal (for \( \epsilon = 10^{-3} \)) objective function value \( x_4^*(1) \) of Problem (44) for an increasing number \( n_q \) of equidistant control discretization points together with the optimal controls \( q^* \).

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

**Definition 19** (Compared Algorithms)

1. **Direct single shooting (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 direct single shooting with one extra variable vector \( s_{t_f} \) for \( x(t_f) \) and was proposed in [PA02].
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 \( \{t_0, t_1, \ldots, t_{n_q}\} \) of the control discretization.

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

Table III shows computational results. 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 17.

To gain more insight, we compare the lower bounds for some selected choices of \(Q\) in Table IV. 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 \(\psi(q_i) \geq -\frac{1}{4}(q_i - q_{\text{ref}})^2\).

It is remarkable, though, how the introduction of additional multiple shooting nodes improves the bounds. Also for tighter domains Table IV reveals how much the multiple shooting approach with larger \(n_{ms}\) dominates the case with only one shooting interval.

As motivated above, this dominance of lower bounds is guaranteed if (38) holds, and is more likely, if the relaxation factors are dominated for which the second order sensitivities (43) are decisive. The involved tensors are difficult to 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 (44) is the most interesting one).
<table>
<thead>
<tr>
<th>$n_q$</th>
<th>$[q_i, \tilde{q}_i] \forall i$</th>
<th>DSS</th>
<th>DMS1</th>
<th>DMS2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$[-4, 10]$</td>
<td>-0.735976 0.496546 0.496546</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-4, -3.95]$</td>
<td>128.981 128.981 128.981</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-9.95, 10]$</td>
<td>138.949 138.949 138.949</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-1.0; 0.1]$</td>
<td>0.496545 0.496546 0.496546</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.1; 0.1]$</td>
<td>0.496545 0.496546 0.496546</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.01; 0.01]$</td>
<td>0.496545 0.496546 0.496546</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-4, -3.95]$</td>
<td>128.981 128.981 128.981</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-9.95, 10]$</td>
<td>138.949 138.949 138.949</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-1.0; 1]$</td>
<td>-0.767086 -0.319456 -0.312031</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.1; 0.1]$</td>
<td>0.146103 0.146104 0.146709</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.01; 0.01]$</td>
<td>0.120369 0.120370 0.121850</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>$[-4, 10]$</td>
<td>-993.86 -252.515 -169.889</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-4, -3.95]$</td>
<td>128.981 128.981 128.981</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$[-9.95, 10]$</td>
<td>138.949 138.949 138.949</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-1.0; 1]$</td>
<td>-0.605863 -0.250872 -0.053446</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.1; 0.1]$</td>
<td>0.120369 0.120370 0.121850</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$q_i^* + [-0.01; 0.01]$</td>
<td>0.122361 0.122362 0.122381</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table IV. Lower bounds on the solution of Problem (44) for specific choices of $Q$.

In Figure 4 the shown trajectories for DMS1 and DMS4,

\[
\frac{d^2 x_{j,i}^* (t; Q)}{ds_{ik} ds_{il}} \quad \text{and} \quad \frac{d^2 x_{j,i}^{\text{ms},i} (t; Q, S_i)}{ds_{ik} ds_{il}} \quad \text{for } i = 0, \ldots, n_{\text{ms}} - 1
\]

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

\[
\frac{d^2 x_{j,i}^{\text{ms},i} (t_i; Q, S_i)}{ds_{ik} ds_{il}} = 0
\]

at the multiple shooting times $t_i$ reduces the amplitude of the decisive under- 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.

![Figure 4. Comparison between DMS1 (-----) and DMS4 (-----). Plot of selected trajectories (46) for $j = 4, k = 3$ and different values $l = 1 \ldots 3$ (from left to right). The single shooting trajectories grow exponentially over the whole time horizon, the multiple shooting trajectories on smaller subintervals.](image)
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

$$\frac{d^2 x_{js}^{\text{ms},i}(t; Q)}{dq_i ds_i} \text{ for } i = 0, \ldots, n_{ms} - 1$$

and

$$\frac{d^2 x_{js}^{\text{ms},i}(t; Q, S_i)}{dq_i dq_i} \text{ for } i = 0, \ldots, n_{ms} - 1$$

on the off-diagonals and with

$$\frac{d^2 x_{js}^{\text{ms},i}(t; Q)}{dq_i dq_i} \text{ for } i = 0, \ldots, n_{ms} - 1$$

and

$$\frac{d^2 x_{js}^{\text{ms},i}(t; Q, S_i)}{dq_i dq_i} \text{ for } i = 0, \ldots, n_{ms} - 1$$

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.

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 $dq_2 dq_2$ 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, [CHP+ 15].

6. EXTENSIONS AND SUMMARY

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, i.e., 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 \( t \) which can be included via an artificial state \( x_{n_{x}+1} \) as done for problem (44),
- model parameters \( p \) which 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 \( u : \mathcal{T} \to \mathbb{R}^{n_{u}} \) or integer control function \( v : \mathcal{T} \to \{v^{1}, \ldots, v^{n_{v}}\}\), see [SBD12], 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 multi-stage formulations.

An extension is also possible towards

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

Here further convexifications of the occurring sets and functions in the spirit of the relaxed problem (NLP\textsuperscript{cv}) are necessary, e.g., \( g_{\text{cv}}(x(t), q) \leq 0 \). Even if it is possible to obtain an equivalent formulation of type (1), e.g., 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 \( Q \) and \( X \).

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 [Die15] further algorithmic extensions have been studied that we list here for completeness. They comprehend

- more efficient ways to calculate \( \alpha \) based on 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 \( \alpha_{\text{ms}, i,j} \cdot \psi(s_{i}) \) in (25,26), motivated by the fourth order convergence shown in Theorem 14 (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 non-dynamic global optimization. Therefore it is even more tempting to speed up the runtime by infrequent updates of \( \alpha \), relying on the fact that the under- and overestimators are still getting better in each iteration due to 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 [LS07a] as an alternative to \( \alpha \text{BB} \).

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 \( \alpha \text{BB} \) (actually, it outperforms the \( \alpha \text{BB} \) approach for problem (44) concerning runtime), but obviously independent from our comparison of single vs. multiple shooting.

For details, we refer to [Die15].

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.

ACKNOWLEDGEMENTS

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

REFERENCES


