

Global Optimization with Nonlinear Ordinary Differential Equations

ADAM B. SINGER and PAUL I. BARTON

Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, USA

(Received 21 June 2004; accepted in revised form 4 May 2005)

Abstract. This paper examines global optimization of an integral objective function subject to nonlinear ordinary differential equations. Theory is developed for deriving a convex relaxation for an integral by utilizing the composition result defined by McCormick (Mathematical Programming 10, 147–175, 1976) in conjunction with a technique for constructing convex and concave relaxations for the solution of a system of nonquasimonotone ordinary differential equations defined by Singer and Barton (SIAM Journal on Scientific Computing, Submitted). A fully automated implementation of the theory is briefly discussed, and several literature case study problems are examined illustrating the utility of the branch-and-bound algorithm based on these relaxations.

Key words: Convex relaxations, dynamic optimization, nonquasimonotone differential equations

1. Introduction

Many practically important physical systems are modeled abstractly by ordinary differential and integral equations. Classically, optimization of these models has been addressed in their native infinite dimensional spaces by the calculus of variations and optimal control; examples are found in Troutman (1996), Pontriagin (1962), and Bryson, Jr. and Ho (1975). The techniques developed for solving these problems are mainly based on solving necessary conditions for local minima. Except for problems possessing certain convexity properties, global solutions are not verifiable. Furthermore, even finding local minima is complicated by the fact that the necessary conditions consist of difficult to solve two-point boundary value problems, possibly in conjunction with application of the minimum principle. However, with the increasing efficiency of computational methods for solving initial value problems in large systems of differential equations, the advent of methods for approximating the decision space with finite dimension, and the increasing sophistication of techniques for robustly solving finite dimensional optimization problems to global optimality, the possibility of solving finite dimensional approximations of optimal control problems globally has become a reality.

In order to reduce an infinite dimensional optimization problem to a finite dimensional problem, the problem must be discretized. Two different discretization approaches exist: complete collocation (e.g., Neuman and Sen, 1973; Tsang et al., 1975), in which both states and controls are discretized, and control parameterization (e.g., Bruschi and Schappelle, 1973; Teo et al., 1991), in which only the controls are discretized. This paper exclusively considers control parameterization.

A popular method for deterministic global optimization is the combination of branch-and-bound and relaxation techniques to eliminate systematically portions of the decision space that theoretically cannot contain the minimum. If the relaxations are generated properly, these methods can be shown to have finite convergence to ε tolerance. Esposito, and Floudas (2000a,b), Papamichail and Adjiman (2002), and Singer and Barton (2004) have previously presented deterministic methods for globally solving optimization problems with differential equations embedded. These approaches are now considered in more detail.

In their papers, Esposito, and Floudas (2000a,b) present the first practical deterministic algorithm for finding the global optimum of an optimal control problem. Utilizing a branch-and-bound framework and building on the αBB relaxation technique presented in Adjiman et al., (1998a,b), they have shown that, under some mild differentiability assumptions, an NLP with a dynamic system embedded is twice-continuously differentiable. Hence, by computing an interval enclosure of the Hessian for any nonconvex terms in the objective and constraint functionals, a relaxation for the nonconvex terms can be derived by adding a suitable quadratic term to the original functional. A suitable quadratic term is one whose convexity overpowers the nonconvexity of the original functional; this is guaranteed by selecting β parameters for the quadratic term such that the Hessian for the combined quadratic and nonconvex function is positive-semidefinite. In the abstract sense, the theoretical construct proposed by Esposito and Floudas is rigorous. However, in practice, the algorithm as presented is pseudo-deterministic because only sampling approaches are proposed for determining the necessary β parameters. Moreover, not only is it very computationally expensive to sample a high dimension space in order to calculate β parameters, but in the worst case, incomplete sampling could yield a β parameter that would not overpower the nonconvex term hence yielding a nonconvex relaxation.

Recently, Papamichail and Adjiman (2002) have presented the first truly rigorous deterministic global optimization algorithm for problems with ordinary differential equations (ODEs) embedded. Again, a spatial branch-and-bound algorithm is considered. The uniqueness of the approach is in its rigorous construction of a convex relaxation. Rather than attempt to convexify every term in the objective and constraint functionals with

state variables participating, their method substitutes a new real valued decision variable for each state at a fixed time, relegating any nonconvexity arising from the state variables to equality constraints with both the new variables and the state variables appearing explicitly. This new equality constraint is then relaxed by deriving a convex lower bounding inequality and a concave upper bounding inequality. Any other nonconvexities in the resulting transformed lower bounding problem are also relaxed via standard convexification techniques. However, because a state variable's functional dependence on the optimization parameters is only known indirectly via the solution of a system of differential equations, two techniques based on differential inequalities are presented for rigorously bounding these solutions. The first technique utilizes differential inequalities to provide constant (relative to the embedded parameters) bounds on the states. The second technique utilizes an α shift parameter and an overpowering quadratic term to convexify or concavify a state while retaining a functional dependence on the embedded parameter. The α parameter is determined rigorously by a combination of interval arithmetic techniques and the simultaneous solution of second order sensitivity equations. This method may be computationally expensive, particularly if the number of optimization parameters and constraints grows large, because the computation of second order sensitivities grows quadratically in the number of optimization parameters. Moreover, the method is applicable only to objective functionals and constraints involving the state variables at fixed time points.

Singer and Barton (2004) have presented a rigorous global optimization technique applicable to optimal control problems with linear dynamic systems embedded. In their approach, a nonlinear objective functional is considered with particular emphasis on an integral formulation. Based on the monotonicity property of the Lebesgue integral, a convexity theory is presented enabling the direct relaxation of a nonconvex integral. Rather than adding new variables and new constraints to the problem, special structural properties of linear systems are exploited in a composition approach to construct convex relaxations of the original nonconvex problem. These relaxations are then used in a branch-and-bound framework to solve the original nonconvex problem. The advantages of this approach are that the optimization problems can be solved in their original integral form; new variables, new constraints, and second order sensitivities are not needed; and the relaxations derived by composition are relatively tight. The disadvantage, of course, is that the technique is restricted to problems with embedded linear dynamic systems.

In this paper, we extend the ideas from Singer and Barton (2004) on linear dynamics to encompass the solution of optimization problems with integral objective functions and nonlinear, nonquasimonotone dynamic systems.

A deterministic branch-and-bound algorithm with finite ε convergence to a global solution is developed. In general, any rigorous method for constructing convex relaxations for an integrand is suitable for constructing a convex relaxation for an integral. However, in this paper, we emphasize utilizing the relaxation scheme of McCormick (1976) combined with our technique for constructing convex relaxations for the solutions of parameter dependent, nonlinear, nonquasimonotone ordinary differential equations as presented in Singer and Barton (Submitted). Several small examples are included to illustrate the complete construction of the relaxations, and larger case studies are included to demonstrate the efficiency of the algorithm. For the larger problems, the relaxations are generated automatically.

2. Problem Formulation

This section details the abstract mathematical formulation of the problems considered in this paper. Throughout the paper, states are denoted by x , parameters are denoted by p , and the independent variable is denoted by t . Underlined variables denote that the variable is held constant. Bold type is used to indicate vector-valued quantities. The general problem is now stated as follows:

PROBLEM 2.1. Let P be a nonempty compact convex subset of \mathbb{R}^{n_p} , $X \subseteq \mathbb{R}^{n_x}$ such that $\mathbf{x}(t, \mathbf{p}) \in X \forall (t, \mathbf{p}) \in [t_0, t_f] \times P$. Consider the following problem:

$$\min_{\mathbf{p}} J(\mathbf{p}) = \phi(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} \ell(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt$$

subject to

$$g_i(\mathbf{x}(t_f, \mathbf{p}), \mathbf{p}) + \int_{t_0}^{t_f} h_i(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt \leq \mathbf{0}, \quad i = 1, \dots, nc$$

where $\mathbf{x}(t, \mathbf{p})$ is given by

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \\ \mathbf{x}(t_0, \mathbf{p}) &= \mathbf{x}_0(\mathbf{p}) \end{aligned}$$

and $\mathbf{p} \in P$; ϕ is a continuous mapping $\phi: X \times P \rightarrow \mathbb{R}$; ℓ is a Lebesgue integrable mapping $\ell: (t_0, t_f] \times X \times P \rightarrow \mathbb{R}$; f_i is a Lebesgue integrable mapping $f_i: (t_0, t_f] \times X \times P \rightarrow \mathbb{R}$, $i = 1, \dots, n_x$; x_{i0} is a continuous mapping $x_{i0}: P \rightarrow \mathbb{R}$, $i = 1, \dots, n_x$; g_i is a continuous mapping $g_i: X \times P \rightarrow \mathbb{R}$, $i = 1, \dots, nc$; and h_i is a Lebesgue integrable mapping $h_i: (t_0, t_f] \times X \times P \rightarrow \mathbb{R}$,

$i = 1, \dots, nc$. nc is the number of point and/or isoperimetric constraints. Furthermore, the functions $\ell, \mathbf{f}, \mathbf{h}$ are only permitted a finite number of stationary simple discontinuities in t (Definition 2.1 in Singer and Barton (2004)).

In addressing Problem 2.1, we speak of a solution to the differential equations defined in Problem 2.1 in the sense of Carathéodory. That is, a solution is considered admissible if the differential equations are satisfied almost everywhere. By asserting that the differential equations satisfy an existence and uniqueness condition (Equations 12.2 and 14.2, respectively, in Walter, 1970) and the set of feasible points is not empty, the existence of a minimum to Problem 2.1 can be proven. The proof of this assertion is a trivial extension of Proposition 2.1 and Corollary 2.1 in Singer and Barton (2004) and is thus omitted here.

3. Development of Convex Relaxations

The algorithm developed for solving Problem 2.1 employs a standard branch-and-bound algorithm to guarantee convergence to a global solution in the Euclidean space P . Within this framework, an upper bounding problem and a lower bounding relaxation must be derived and solved at each node in the branch-and-bound tree. In our algorithm, an upper bound at each node is typically computed by solving Problem 2.1 locally. Determining a rigorous lower bound at each node is much more complicated and is thus explained in detail below.

As is typical with branch-and-bound algorithms, a convex relaxation is chosen for the lower bounding problem. Choosing a convex relaxation for the lower bounding problem implies that a local solution of the lower bounding problem is a global solution of the relaxation and is therefore theoretically guaranteed to be a valid lower bound. The idea behind deriving a convex relaxation for Problem 2.1 is that integrating a pointwise in time convex underestimator for an integrand provides a convex underestimator for an integral. While this result is generally valid, its use in a practical algorithm is predicated on the ability to find efficiently a convex relaxation of an integrand defined by the composition of functions known only via the numerical solution of a system of ODEs. This section develops a method for constructing a convex relaxation for Problem 2.1.

The analysis begins by deriving the result that a pointwise in time convex relaxation for a parameter dependent integrand implies a convex relaxation for the integral on the parameter space. We begin with the following lemma on the monotonicity of a parameter dependent integral. The notation \underline{t} is utilized to represent any fixed time in $(t_0, t_f]$.

LEMMA 3.1. Let $\mathbf{p} \in P$, $t \in (t_0, t_f]$, and $\ell_1, \ell_2 : (t_0, t_f] \times P \rightarrow \mathbb{R}$ such that ℓ_1, ℓ_2 are Lebesgue integrable. If

$$\ell_1(\underline{t}, \mathbf{p}) \leq \ell_2(\underline{t}, \mathbf{p}), \quad \forall (\underline{t}, \mathbf{p}) \in (t_0, t_f] \times P$$

then

$$L_1(\mathbf{p}) = \int_{t_0}^{t_f} \ell_1(t, \mathbf{p}) dt \leq \int_{t_0}^{t_f} \ell_2(t, \mathbf{p}) dt = L_2(\mathbf{p}), \quad \forall \mathbf{p} \in P.$$

Proof. See Singer and Barton (2004) □

The following theorem shows that if a parameter dependent integrand is convex on P for each fixed $\underline{t} \in (t_0, t_f]$, the integral is convex on P . Immediately following this theorem is a corollary that illustrates the construction of a convex relaxation for an integral from a convex relaxation for the integrand.

THEOREM 3.2. Let $\mathbf{p} \in P$, $t \in (t_0, t_f]$, and $\ell : (t_0, t_f] \times P \rightarrow \mathbb{R}$ be Lebesgue integrable. If $\ell(\underline{t}, \mathbf{p})$ is convex on P for each fixed $\underline{t} \in (t_0, t_f]$, then

$$L(\mathbf{p}) = \int_{t_0}^{t_f} \ell(t, \mathbf{p}) dt$$

is convex on P .

Proof. By hypothesis, $\ell(\underline{t}, \mathbf{p})$ is convex on P , $\forall \underline{t} \in (t_0, t_f]$. Therefore, given $\lambda \in (0, 1)$, we have

$$\ell(\underline{t}, \lambda \mathbf{p} + (1 - \lambda) \mathbf{p}_0) \leq \lambda \ell(\underline{t}, \mathbf{p}) + (1 - \lambda) \ell(\underline{t}, \mathbf{p}_0), \quad \forall (\underline{t}, \mathbf{p}), (\underline{t}, \mathbf{p}_0) \in (t_0, t_f] \times P.$$

By Lemma 3.1, the above equation implies

$$\begin{aligned} \int_{t_0}^{t_f} \ell(t, \lambda \mathbf{p} + (1 - \lambda) \mathbf{p}_0) dt &\leq \int_{t_0}^{t_f} \lambda \ell(t, \mathbf{p}) dt \\ &\quad + \int_{t_0}^{t_f} (1 - \lambda) \ell(t, \mathbf{p}_0) dt, \quad \forall \mathbf{p}, \mathbf{p}_0 \in P, \end{aligned}$$

which by inspection equals

$$L(\lambda \mathbf{p} + (1 - \lambda) \mathbf{p}_0) \leq \lambda L(\mathbf{p}) + (1 - \lambda) L(\mathbf{p}_0), \quad \forall \mathbf{p}, \mathbf{p}_0 \in P.$$

By definition, $L(\mathbf{p})$ is convex on P . □

An interesting implication of requiring only partial convexity of the integrand in Theorem 3.2 is that even if an integrand is nonconvex on $(t_0, t_f] \times P$, its integral may still be convex. The following example illustrates this principle.

EXAMPLE 3.3. Consider $p \in [0, 3]$ and the following integral:

$$J(p) = \int_0^1 -\sin(pt) \, dt.$$

The integrand is nonconvex on $[0, 1] \times [0, 3]$; however, the integrand is partially convex on $[0, 3]$ for each fixed $t \in [0, 1]$. By Theorem 3.2, $J(p)$ is convex on $[0, 3]$. Figure 1 below depicts the partially convex integrand, while Figure 2 illustrates the convex integral.

The following corollary combines Lemma 3.1 with Theorem 3.2 to yield a result enabling the construction of a convex (and a concave) relaxation for an integral.

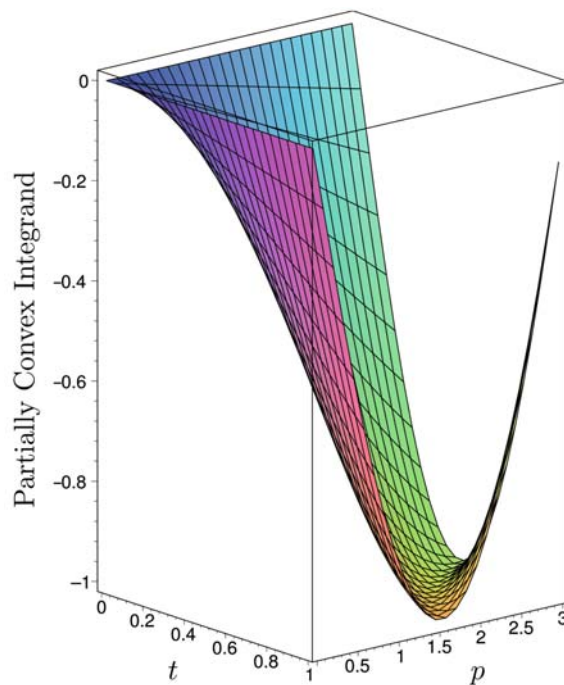


Figure 1. Convex integral from Example 3.3.

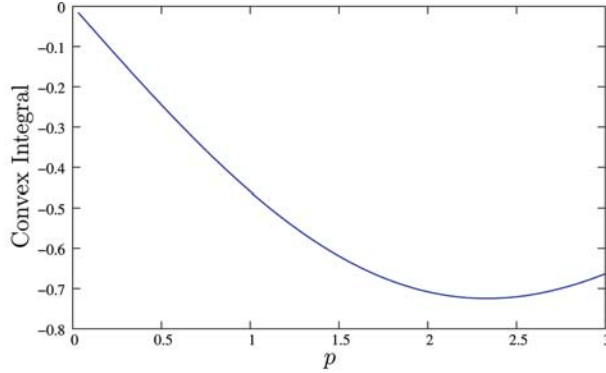


Figure 2. Partially convex integrand from Example 3.3.

COROLLARY 3.4. Let $\mathbf{p} \in P$, $t \in (t_0, t_f]$, and $\mathcal{U}, \ell : (t_0, t_f] \times P \rightarrow \mathbb{R}$ be Lebesgue integrable. If $\mathcal{U}(\underline{t}, \mathbf{p})$ is convex on P , $\forall \underline{t} \in (t_0, t_f]$ and

$$\mathcal{U}(\underline{t}, \mathbf{p}) \leq \ell(\underline{t}, \mathbf{p}), \quad \forall (\underline{t}, \mathbf{p}) \in (t_0, t_f] \times P,$$

then $U(\mathbf{p})$ is convex on P and $U(\mathbf{p}) \leq L(\mathbf{p})$, where

$$U(\mathbf{p}) = \int_{t_0}^{t_f} \mathcal{U}(t, \mathbf{p}) dt.$$

Proof. The proof is immediately evident from Lemma 3.1 and Theorem 3.2. \square

Remark 3.5. An analogous result applies for concave overestimators.

Corollary 3.4 enables one to calculate a convex underestimator for an integral by integrating a relaxation of the integrand, provided this relaxation is convex for each fixed $\underline{t} \in (t_0, t_f]$. Because a continuous set contains an uncountable number of points, computing an underestimator at each fixed $\underline{t} \in (t_0, t_f]$ is clearly impossible numerically. Obviously, in order to solve problems practically, numerical integration methods are required; fixed time quantities are simply computed at each integration time step.

As presented, Theorem 3.2, and consequently Corollary 3.4, apply to an integrand for which convexity is known on the parameter space P . If the functional dependence on \mathbf{p} for an integrand is explicit, computing a convex relaxation for this integrand would follow directly from standard techniques. However, Problem 2.1 contains an integrand that may have dependence on parameters via composition with the state variables. Therefore, in order to apply Corollary 3.4 to Problem 2.1, we must first

introduce a method for determining convexity of a composite function. The theorem presented below is a classical result first presented by McCormick (1976) for relaxing factorable functions. The discussion following the theorem justifies its use in the context of Problem 2.1.

THEOREM 3.6. *Let $S \subset \mathbb{R}^{n_x}$ be a nonempty convex set. Consider the function $V(v(\mathbf{p}))$ where $v: S \rightarrow \mathbb{R}$ is continuous, and let $S \subset \{\mathbf{p} \mid v(\mathbf{p}) \in [a, b]\}$. Suppose that a convex function $c(\mathbf{p})$ and a concave function $C(\mathbf{p})$ satisfying*

$$c(\mathbf{p}) \leq v(\mathbf{p}) \leq C(\mathbf{p}), \quad \forall \mathbf{p} \in S$$

are available. Denote a convex relaxation of $V(\cdot)$ on $[a, b]$ by $e_V(\cdot)$, denote a concave relaxation of $V(\cdot)$ on $[a, b]$ by $E_V(\cdot)$, let z_{min} be a point at which $V(\cdot)$ attains its infimum on $[a, b]$, and let z_{max} be a point at which $V(\cdot)$ attains its supremum on $[a, b]$. If the above conditions are satisfied, then

$$u(\mathbf{p}) = e_V(\text{mid}\{c(\mathbf{p}), C(\mathbf{p}), z_{min}\})$$

is a convex underestimator for $V(v(\mathbf{p}))$ on S , and

$$o(\mathbf{x}) = E_V(\text{mid}\{c(\mathbf{p}), C(\mathbf{p}), z_{max}\})$$

is a concave overestimator for $V(v(\mathbf{p}))$ on S , where the mid function selects the middle value of three scalars.

Proof. We prove the theorem only for $u(\mathbf{p})$; the proof for $o(\mathbf{p})$ is analogous. First, the convex relaxation $e_V(\cdot)$ is broken up into the sum of three terms: the convex nonincreasing part, the convex nondecreasing part, and the minimum value. The convex nonincreasing part is defined as

$$e_V^D(v) = e_V(\min\{v, z_{min}\}) - A$$

and the convex nondecreasing part by

$$e_V^I(v) = e_V(\max\{v, z_{min}\}) - A,$$

where $A = e_V(z_{min})$. Obviously, then,

$$e_V(v) = e_V^D(v) + e_V^I(v) + A.$$

Consider the chain of inequalities

$$\begin{aligned}
V(v(\mathbf{x})) &\geq e_V(v(\mathbf{p})) = e_V^D(v(\mathbf{p})) + e_V^I(v(\mathbf{p})) + A \\
&\geq e_V^D(\min\{b, C(\mathbf{p})\}) + e_V^I(\max\{a, c(\mathbf{p})\}) + A \\
&= e_V(\min\{b, C(\mathbf{p}), z_{min}\}) + e_V(\max\{a, c(\mathbf{p}), z_{min}\}) - A \\
&= e_V(\min\{C(\mathbf{p}), z_{min}\}) + e_V(\max\{c(\mathbf{p}), z_{min}\}) - A \\
&= e_V(\text{mid}\{c(\mathbf{p}), C(\mathbf{p}), z_{min}\}).
\end{aligned}$$

The convexity of the underestimating function follows because the minimum of two concave functions is concave, the maximum of two convex functions is convex, a convex nonincreasing function of a concave function is convex, and a convex nondecreasing function of a convex function is convex. \square

Remark 3.7. In his seminal paper, McCormick (1976) generalized the results of Theorem 3.6 to terms of the general form $T(t(\mathbf{x})) + U(u(\mathbf{x})) \cdot V(v(\mathbf{x}))$. Furthermore, the paper explains how Theorem 3.6 may be applied recursively to generate a convex underestimator for any factorable program.

In this paper, Theorem 3.6 is utilized recursively in order to derive a convex relaxation for the integrand in the objective function; the integrand is defined by a composition with $\mathbf{x}(t, \mathbf{p})$, which is itself known only indirectly via the solution of the differential equations. As presented, however, Theorem 3.6 applies to Euclidean optimization problems where the inner function of the composition is bounded in range. Utilizing any suitable state bounding technique (see Section 4), time varying enclosures of the solution of the differential equations for all \mathbf{p} in a hyperrectangle P may be obtained. We label this interval $\mathcal{X}(t) = [\mathbf{x}^L(t), \mathbf{x}^U(t)]$ and also make extensive use of its pointwise in time analog, $\mathcal{X}(t)$. A suitable state bounding technique is defined as any method such that as P becomes degenerate, $\mathcal{X}(t)$ becomes degenerate as well. In this context, the term degenerate refers to an interval whose lower bound equals its upper bound. For example, if Corollary 2.5 in Singer and Barton (Submitted) is satisfied via interval arithmetic techniques, the suitability of the bounding interval becomes obvious, for at degeneracy of the parameter interval, the bounding differential system equals the original system of differential equations. Moreover, because the interval $\mathcal{X}(t)$ (and hence $\mathcal{X}(t)$) rigorously bounds the range of $\mathbf{x}(t, \mathbf{p})$ and because $\mathcal{X}(t)$ shrinks to degeneracy as P shrinks to degeneracy, we have a range interval $[a, b]$ satisfying Theorem 3.6. In the case where recursion is required to generate an underestimator for the integrand, recursive interval extensions on $\mathcal{X}(t)$ provide the additional range

bounding sets; interval monotonicity ensures that the recursively derived range enclosures also approach degeneracy. To complete the justification for using Theorem 3.6 for dynamic problems, we must show that the composition is performed on a Euclidean space rather than on the function space \mathcal{X} . Clearly, for each fixed $\underline{t} \in (t_0, t_f]$, $\mathcal{X}(\underline{t}) \subset \mathbb{R}^{n_x}$ and is therefore a Euclidean space. Additionally, any fixed time set created by an interval extensions on $\mathcal{X}(\underline{t})$ is also a subset of a Euclidean space. This justifies utilizing Theorem 3.6 in conjunction with Corollary 3.4 to derive convex relaxations for Problem 2.1.

The only remaining issue in creating convex relaxations for Problem 2.1 is the derivation of convex underestimators $\mathbf{c}(t, \mathbf{p})$ and concave overestimators $\mathbf{C}(t, \mathbf{p})$ for the state variables. In general, computing $\mathbf{c}(t, \mathbf{p})$ and $\mathbf{C}(t, \mathbf{p})$ is a very difficult task. In Singer and Barton (Submitted), we have developed a general method for computing $\mathbf{c}(t, \mathbf{p})$ and $\mathbf{C}(t, \mathbf{p})$ for nonquasimonotone ODEs. Assuming the ability to generate state bounds and the ability to generate convex and concave relaxations on $\mathcal{X}(\underline{t}) \times P$ of the right hand sides of a parameter dependent system of ODEs, the technique constructs systems of linear bounding differential equations. By the properties of linear differential equations, the solutions of these bounding differential equations are affine in the parameter space for each fixed time and are therefore both convex and concave. Differential inequalities are utilized to demonstrate that the bounding differential equations rigorously relax the state variables. Furthermore, by construction, one can show that degeneracy of the parameter space implies that $\mathbf{c}(t, \mathbf{p})$ and $\mathbf{C}(t, \mathbf{p})$ both equal $\mathbf{x}(t, \mathbf{p})$. Because manual application of the technique is highly error prone, we have written a program to implement automatically the derivation of $\mathbf{c}(t, \mathbf{p})$ and $\mathbf{C}(t, \mathbf{p})$. The convex state bounding theorem is presented below without proof. First, however, we define the notion of a linearization for a function.

DEFINITION 3.8. Let $\mathcal{L}_{f_i}(t, \mathbf{y}, \mathbf{q})|_{\mathbf{y}^*(t), \mathbf{q}^*}$ be a linearization of $\mathbf{f}(t, \mathbf{x}, \mathbf{p})$ at $(\mathbf{y}^*(t), \mathbf{q}^*)$. The linearization is given for $i = 1, \dots, n_x$ by the following:

$$\begin{aligned} \mathcal{L}_{f_i}(t, \mathbf{y}, \mathbf{q})|_{\mathbf{y}^*(t), \mathbf{q}^*} &= f_i(t, \mathbf{y}^*(t), \mathbf{q}^*) + \sum_{j=1}^{n_x} \left. \frac{\partial f_i}{\partial x_j} \right|_{\mathbf{y}^*(t), \mathbf{q}^*} (y_j - y_j^*(t)) \\ &\quad + \sum_{j=1}^{n_p} \left. \frac{\partial f_i}{\partial p_j} \right|_{\mathbf{y}^*(t), \mathbf{q}^*} (q_j - q_j^*). \end{aligned} \quad (1)$$

We now present the following theorem utilized for constructing convex underestimators and concave overestimators for the solution of a parameter dependent ODE.

THEOREM 3.9. *Consider*

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \\ \mathbf{x}(0, \mathbf{p}) &= \mathbf{x}_0(\mathbf{p})\end{aligned}$$

whose solution is bounded by the set $\mathcal{X}(t)$ of time varying state bounds (Section 4 provides a method for computing $\mathcal{X}(t)$). Let P be convex. For $i = 1, \dots, n_x$ and each fixed $\underline{t} \in (t_0, t_f]$, let $u_i(\underline{t}, \mathbf{x}, \mathbf{p})$ be a convex underestimator for f_i on $\mathcal{X}(\underline{t}) \times P$ and let $o_i(\underline{t}, \mathbf{x}, \mathbf{p})$ be a concave overestimator for f_i on $\mathcal{X}(\underline{t}) \times P$. For $i = 1, \dots, n_x$, consider the differential equations

$$\begin{aligned}\dot{c}_i &= g_{c,i}(t, \mathbf{c}, \mathbf{C}, \mathbf{p}) = \inf_{\substack{\mathbf{z} \in \mathcal{C}(t, \mathbf{p}) \\ z_i = c_i(t)}} \mathcal{L}_{u_i}(t, \mathbf{z}, \mathbf{p}) \Big|_{\mathbf{x}^*(t), \mathbf{p}^*} \\ \dot{C}_i &= g_{C,i}(t, \mathbf{c}, \mathbf{C}, \mathbf{p}) = \sup_{\substack{\mathbf{z} \in \mathcal{C}(t, \mathbf{p}) \\ z_i = C_i(t)}} \mathcal{L}_{o_i}(t, \mathbf{z}, \mathbf{p}) \Big|_{\mathbf{x}^*(t), \mathbf{p}^*}\end{aligned}$$

for some reference trajectory $(\mathbf{x}^*(\underline{t}), \mathbf{p}^*) \in \mathcal{X}(\underline{t}) \times P$ (differentiability of u_i and o_i is assumed along the reference trajectory) with initial conditions

$$\mathbf{c}(t_0) \leq \mathbf{x}(t_0, \mathbf{p}) \leq \mathbf{C}(t_0) \quad \forall \mathbf{p} \in P, \quad (2)$$

where $\mathcal{C}(t, \mathbf{p}) = \{\mathbf{z} \mid \mathbf{c}(t, \mathbf{p}) \leq \mathbf{z} \leq \mathbf{C}(t, \mathbf{p})\}$. Then for each fixed $\underline{t} \in (t_0, t_f]$, $\mathbf{c}(\underline{t}, \mathbf{p})$ is a convex underestimator for $\mathbf{x}(\underline{t}, \mathbf{p})$ on P and $\mathbf{C}(\underline{t}, \mathbf{p})$ is a concave overestimator for $\mathbf{x}(\underline{t}, \mathbf{p})$ on P .

Proof. See Singer and Barton (2004a) for proof and discussion. \square

Remark 3.10. Differentiability of u_i and o_i along the reference trajectory is assumed in order to employ Definition 3.8 to establish a formulaic representation of the linearization. Theorem 3.9 still holds for nondifferentiable convex functions u_i and nondifferentiable concave functions o_i provided any suitable subgradient is chosen for construction. That is, the linearization employed may be any valid hyperplane supporting the relaxation along the reference trajectory. In practice, nondifferentiability typically arises from taking the maximum or minimum of two valid supporting hyperplanes (e.g., the relaxation of a bilinear function). In this case, arbitrarily choosing either of the two hyperplanes yields a valid relaxation; this selection process is discussed in detail in Singer (2004a).

Thus far, only abstract theory necessary for relaxing Problem 2.1 has been presented. A very simple example is now presented to illustrate the practical computation of a convex underestimator for an integral with an embedded nonlinear dynamic system. In order to focus entirely on the

derivation of a convex relaxation for the integral, an analytically solvable differential equation has been selected for the example. Detailed examples for deriving $\mathcal{X}(t)$, $\mathbf{c}(t, \mathbf{p})$, and $\mathbf{C}(t, \mathbf{p})$ where an analytic solution to the ODE is not available are found in Singer and Barton (Submitted).

EXAMPLE 3.11. Consider the optimization problem

$$\min_{p \in [-3, 3]} L(p) = \int_0^1 -[x(t, p)]^2 dt,$$

where $x(t, p)$ is given by

$$\begin{aligned} \dot{x} &= px \\ x(0) &= -1 \\ p &\in [-3, 3]. \end{aligned}$$

This example develops a convex relaxation for $L(p)$ at the root node. The analytical solution to the differential equation is

$$x(t, p) = -\exp(pt), \quad (3)$$

and this allows us to compute explicitly the objective function:

$$L(p) = \begin{cases} -1 & \text{if } p = 0 \\ [1 - \exp(2p)] / (2p) & \text{otherwise.} \end{cases}$$

Because the objective function is nonconvex, a global optimization technique is warranted; hence, a convex relaxation at the root node is required. From Corollary 3.4, we know that to generate a convex underestimator for the integral requires a convex underestimator for the integrand. First, the composition for the integrand is identified. In this case, we trivially have

$$\ell(z) = -z^2,$$

where $z = x(t, p)$ is given by the solution of the differential equation (in this case given explicitly by Equation (3)). From Theorem 3.6, a convex underestimator for the integrand is given pointwise in time by

$$\mathcal{U}(\underline{t}, p) = e(\text{mid}\{c(\underline{t}, p), C(\underline{t}, p), z_{\min}(\underline{t})\}), \quad \forall \underline{t} \in [0, 1].$$

Therefore, we must identify $e(\cdot)$, compute $\mathcal{X}(t)$ and $z_{\min}(\underline{t})$, and derive functions $c(\underline{t}, p)$ and $C(\underline{t}, p)$. Because t is always positive, for any fixed $\underline{t} \in [0, 1]$, $x(\underline{t}, p)$ is monotonically decreasing. This fact enables us to trivially calculate $\mathcal{X}(t)$ by interval extensions to yield

$$\mathcal{X}(t) = [-\exp(p^U t), -\exp(p^L t)] = [-\exp(3t), -\exp(-3t)].$$

For each $\underline{t} \in [0, 1]$, ℓ is concave on $Z = \mathcal{X}(\underline{t})$; therefore, $e(z)$ is simply the secant given by

$$e(\underline{t}, z) = [x^U(\underline{t}) + x^L(\underline{t})][x^L(\underline{t}) - z] - [x^L(\underline{t})]^2, \quad \forall \underline{t} \in [0, 1].$$

Because $[x^U(\underline{t}) + x^L(\underline{t})] < 0$, $\forall \underline{t} \in [0, 1]$, we know that $z_{\min}(\underline{t}) = x^L(\underline{t})$, $\forall \underline{t} \in [0, 1]$. To complete the analysis, we must derive the convex underestimator $c(\underline{t}, p)$ and the concave overestimator $C(\underline{t}, p)$. The second partial derivative of Equation (3) with respect to p is less than or equal to zero for all $(\underline{t}, p) \in [0, 1] \times P$; thus, the function is always concave. Trivially, we have

$$C(\underline{t}, p) = -\exp(p\underline{t}), \quad \forall \underline{t} \in [0, 1]$$

and

$$c(\underline{t}, p) = [\exp(p^L\underline{t}) - \exp(p^U\underline{t})](p - p^L)/(p^U - p^L) - \exp(p^L\underline{t}), \quad \forall \underline{t} \in [0, 1].$$

Combining, the pointwise in time underestimator for the integrand is given by

$$\begin{aligned} \mathcal{U}(\underline{t}, p) = & [x^U(\underline{t}) + x^L(\underline{t})][x^L(\underline{t}) - \text{mid}\{c(\underline{t}, p), C(\underline{t}, p), z_{\min}(\underline{t})\}] \\ & - [x^L(\underline{t})]^2, \quad \forall \underline{t} \in [0, 1]. \end{aligned}$$

By Corollary 3.4, we have that a relaxation for the integral is given by

$$U(p) = \int_0^1 \mathcal{U}(t, p) dt.$$

4. Nonquasimonotone Systems and Exploding Bounds

One of the most challenging aspects of solving Problem 2.1 globally is the presence of nonquasimonotone differential equations in the embedded dynamics. Applying standard techniques for generating state bounds (see Walter, 1970 and Harrison, 1977) to nonquasimonotone differential equations leads to bounds that rapidly explode on short time scales. That is, on typical time scales of interest, the state bounds generated by standard techniques exponentially approach infinity. Because generating convex relaxations for Problem 2.1 requires bounds on the states, state bounds approaching infinity lead to convex relaxations approaching negative infinity. Although one can prove that the state bounds for nonquasimonotone systems approach degeneracy as the bounds on P approach degeneracy, a numerical implementation for solving Problem 2.1 requires reasonable bounds at all branch-and-bound nodes.

Often for engineering problems, bounding information about the solution of the differential equations is known independently from the statement of the differential equations themselves. For example, suppose a system of differential equations models the concentrations of chemical species undergoing chemical reaction. By conservation of mass, both upper and lower bounds can be obtained for the states. The set of natural bounds for a system is labeled $\bar{\mathbf{X}}(t, \mathbf{p})$. The following theorem, taken from Singer and Barton (Submitted), utilizes $\bar{\mathbf{X}}(t)$ to derive nonexploding state bounds for systems of nonquasimonotone differential equations.

THEOREM 4.1. *Let $\mathbf{x}(t, \mathbf{p})$ be a solution of*

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \\ \mathbf{x}(t_0) &= \mathbf{x}_0(\mathbf{p})\end{aligned}$$

and let $\mathbf{x}(t, \mathbf{p}) \in \bar{\mathbf{X}}(t, \mathbf{p})$ for each $\mathbf{p} \in P$, where $\bar{\mathbf{X}}(t, \mathbf{p})$ is known independently from the solution of the system of differential equations. Furthermore, let $\bar{\mathbf{X}}(t)$ be defined pointwise in time by

$$\bar{\mathbf{X}}(t) = [\inf_{\mathbf{q} \in P} \bar{\mathbf{X}}(t, \mathbf{q}), \sup_{\mathbf{q} \in P} \bar{\mathbf{X}}(t, \mathbf{q})].$$

If for $i = 1, \dots, n_x$,

$$\begin{aligned}(i) \quad v_i(t_0) &\leq \inf_{\mathbf{q} \in P} x_i(t_0, \mathbf{q}) \\ (ii) \quad w_i(t_0) &\geq \sup_{\mathbf{q} \in P} x_i(t_0, \mathbf{q})\end{aligned}$$

and if $\forall \mathbf{v}(t), \mathbf{w}(t) \in G(t)$

$$\begin{aligned}(iii) \quad \dot{v}_i = g_i(t, \mathbf{v}, \mathbf{w}) &\leq \inf_{\substack{\mathbf{z} \in \bar{\mathbf{X}}(t) \cap G(t) \\ z_i = v_i(t)}, \mathbf{q} \in P} f_i(t, \mathbf{z}, \mathbf{q}) \\ (iv) \quad \dot{w}_i = h_i(t, \mathbf{v}, \mathbf{w}) &\geq \sup_{\substack{\mathbf{z} \in \bar{\mathbf{X}}(t) \cap G(t) \\ z_i = w_i(t)}, \mathbf{q} \in P} f_i(t, \mathbf{z}, \mathbf{q}),\end{aligned}$$

where $G(t) = \{\mathbf{z} \mid \mathbf{v}(t) \leq \mathbf{z} \leq \mathbf{w}(t)\}$, then

$$\mathbf{v}(t) \leq \mathbf{x}(t, \mathbf{p}) \leq \mathbf{w}(t), \quad \forall (t, \mathbf{p}) \in (t_0, t_f] \times P.$$

Proof. See Singer and Barton (Submitted) for proof and discussion. \square

In general, solving the parametric optimization problems defining the right-hand sides of the state bounding differential inequalities is prohibitively expensive. Instead, the solution to the optimization problems is estimated via interval extensions of the right hand sides of the differential equations. For the lower [upper] bounding differential equation i , the

equality constraint $z_i = x_i^L$ [$z_i = x_i^U$] is enforced. For all other state variables $j \neq i$, natural interval arithmetic is applied, where the bounds for variable j are given pointwise in time by

$$\mathcal{X}_j(t) \cap \bar{\mathcal{X}}_j(t) = [\max\{\bar{x}_j^L(t), x_j^L(t)\}, \min\{\bar{x}_j^U(t), x_j^U(t)\}], \quad \forall t \in (t_0, t_f].$$

Additionally, the derivative of the state bound is set to zero when the state bound exceeds its natural bound. Therefore, by construction, $\mathcal{X} \subseteq \bar{\mathcal{X}}$. The set $\bar{\mathcal{X}}$ is defined for each of the case studies for which this technique is utilized. The authors refer the interested reader to Singer and Barton (2004a) for a detailed discussion of state bounding.

5. Infinite Convergence of the Branch-and-bound Algorithm

In order to validate the proposed algorithm, the algorithm must be at least infinitely convergent thus guaranteeing ε convergence to the global solution in finite time. A branch-and-bound algorithm is said to be at least infinitely convergent if the selection operation is bound improving and the bounding operation is consistent (Theorem IV.3 in Horst and Tuy, 1993). Additionally, convergence requires the ability to delete partitions for which the intersection of this partition and the feasible space is empty. By definition, a selection operation is bound improving if at least after a finite number of steps, at least one partition element where the actual lower bound is attained is selected for further partitioning (Definition IV.6 in Horst and Tuy, 1993). By Definition IV.4 in Horst and Tuy (1993), a bounding operation is consistent if, at every step, any unfathomed partition element can be further refined, and if any infinite sequence of nested partitions has the property that the lower bound in any partition converges to the upper bound of this same partition. In other words, not only must every unfathomed partition be refineable, but as any infinite sequence of nested partitions approaches its limit set, its lower bound must converge to the upper bound. In global NLP branch-and-bound algorithms, fathoming of a partition occurs only when its lower bound is greater than the current best upper bound (or within an ε tolerance). Therefore, partitions containing the global minimum are fathomed only at termination. By construction, the branching strategy employed is exhaustive. In an exhaustive search, such as bisecting the variable satisfying

$$\arg \min_i |p_i^U - p_i^L|,$$

by the finite dimensionality of the problem, every unfathomed partition is selected for further partitioning after a finite number of steps. Thus, the retention of partitions on which the global minimum is attained and the utilization of an exhaustive search together imply a bound improving selection operation. Furthermore, because P is a subset of a Euclidean space,

by the Archimedean property of the real numbers, any unfathomed partition can always be refined (by bisection on a coordinate axis, for example); therefore, the first criterion of a consistent bounding operation is trivially satisfied. However, it is not immediately obvious whether or not the integral underestimators obey the second property for consistent bounding. Proving convergence of a branch-and-bound algorithm for solving Problem 2.1 reduces to proving that any infinite sequence of nested partitions has the property that the lower bound in any partition converges to the upper bound of this same partition. Equivalently, convergence follows if the lower bound in any partition converges pointwise to the objective function in this same partition as the Euclidean norm of the partition approaches zero (the interval approaches degeneracy). The proof of this statement requires the assumption that the relaxation for the integrand of $U(\mathbf{p})$ itself possesses a consistent bounding operation with monotonic pointwise convergence. This assumption is justified because the convex underestimators utilized in standard optimization problems possess the property that as the interval decreases, the convex underestimators become higher with monotonic pointwise convergence (see for examples McCormick, 1976 or Maranas and Floudas, 1994). Problem 2.1 considers problems $J(\mathbf{p}) = \phi(t_f, \mathbf{p}) + L(\mathbf{p})$. The relaxation selected for ϕ must possess a consistent bounding operation. Therefore, it is sufficient to consider only the consistent bounding operation of $L(\mathbf{p})$. The following theorem demonstrates this principle.

THEOREM 5.1. *Consider the optimization problem given by*

$$\min_{\mathbf{p} \in P} L(\mathbf{p}) = \int_{t_0}^{t_f} \ell(t, \mathbf{x}(t, \mathbf{p}), \mathbf{p}) dt$$

subject to

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(t, \mathbf{x}, \mathbf{p}) \\ \mathbf{x}(t_0, \mathbf{p}) &= \mathbf{x}_0(\mathbf{p}) \end{aligned}$$

and the relaxation $U(\mathbf{p})$ defined by Corollary 3.4. Let the integrand $\mathcal{U}(t, \mathbf{p})$ be defined by Theorem 3.6, let $\mathbf{c}(t, \mathbf{p})$ and $\mathbf{C}(t, \mathbf{p})$ be defined by Theorem 3.2 in Singer and Barton (2004a), and let the state bounds be defined by Corollary 2.5 in Singer and Barton (Submitted). If the interval in any partition approaches degeneracy, then the lower bound in this partition ($U(\mathbf{p})$) converges pointwise to the upper bound in this partition ($L(\mathbf{p})$).

Proof. Choose any partition and any fixed $\underline{t} \in (t_0, t_f]$ and let $[\mathbf{p}^L, \mathbf{p}^U]$ be the bounds on the parameter in this partition. From Corollary 2.5 in Singer and Barton (Submitted), as the interval $[\mathbf{p}^L, \mathbf{p}^U]$ approaches the degener-

ate value \mathbf{p}^d , the interval $[\mathbf{x}^L(\underline{t}), \mathbf{x}^U(\underline{t})]$ approaches the degenerate value of $\mathbf{x}^d(\underline{t})$. To be valid, the convex underestimator u_i and the concave overestimator o_i from Theorem 3.2 in Singer and Barton (Submitted) must themselves possess a consistent bounding operation and hence as $\overline{\mathcal{X}}(\underline{t}) \times P$ shrinks to degeneracy, $u_i \uparrow f_i$ and $o_i \downarrow f_i$ for $i = 1, \dots, n_x$. The right-hand sides of the equations defining \dot{c}_i and \dot{C}_i are linearizations on u_i and o_i , respectively. Since u_i and o_i are each approaching f_i , choosing $(\mathbf{x}^*(t), \mathbf{p}^*) = (\mathbf{x}^d(t), \mathbf{p}^d)$, we have that $g_{c,i}(\mathbf{c}, \mathbf{C}, \mathbf{p}) \uparrow f_i(\mathbf{x}, \mathbf{p})$ and $g_{C,i}(\mathbf{c}, \mathbf{C}, \mathbf{p}) \downarrow f_i(\mathbf{x}, \mathbf{p})$ since the linearization approaches the value of the function it approximates at the point of linearization. Now, suppose that at each step k , the interval $[\mathbf{p}^L, \mathbf{p}^U]_i$ is bisected (or reduced in some other manner) such that as $k \rightarrow \infty$, $[\mathbf{p}^L, \mathbf{p}^U]_k \rightarrow \mathbf{p}^d$ (the reduction is possible because the subdivision rule is exhaustive). By construction, we have the following sequence:

$$\mathcal{U}_k \uparrow \ell \quad \text{as } k \rightarrow \infty \quad \text{for } \underline{t} \in [t_0, t_f],$$

where the convergence arises because the McCormick underestimator \mathcal{U} possesses a consistent bounding operation (with monotonic convergence) as $[\mathbf{p}^L, \mathbf{p}^U]$ approaches degeneracy. Because \underline{t} was fixed arbitrarily, the convergence is true for all $t \in [t_0, t_f]$. By the monotone convergence theorem (Theorem 1, Section 2.3 in Adams and Guillemin, 1996),

$$L(\mathbf{p}^d) = \int_{t_0}^{t_f} \ell \, dt = \lim_{k \rightarrow \infty} \int_{t_0}^{t_f} \mathcal{U}_k \, dt = U(\mathbf{p}^d).$$

Because the partition was arbitrarily chosen, the convergence is applicable to any partition. \square

Remark 5.2. Strictly speaking, the monotone convergence theorem only applies to positive sequences. Of course, if \mathcal{U} is not positive over all of $[t_0, t_f]$, then \mathcal{U} can be divided into \mathcal{U}_+ and \mathcal{U}_- , and the integral can be written as the difference of integrals possessing only positive integrands. The monotone convergence theorem is then applied piecewise.

Theorem 5.1 illustrates that the convex relaxations constructed in this paper indeed possess a consistent bounding operation. The arguments immediately preceding Theorem 5.1 already justify that the selection operation is bound improving. Therefore, by Theorem IV.3 in Horst and Toy (1993), the branch-and-bound algorithm considered in this paper has infinite convergence to the global solution of Problem 2.1 thus validating the approach presented for solving Problem 2.1.

6. Numerical implementation

In order to solve Problem 2.1 for practical problems, a general purpose, numerical program, named GDOC, has been written. The program itself is divisible into four modules: branch-and-bound, numerical optimization, numerical integration with parametric sensitivity, and residual evaluation. Because solving Problem 2.1 is computationally expensive, a compiled code approach was selected in preference to an interpreted approach. Where applicable, publicly available libraries were utilized, and when necessary, custom code was designed and written. The development platform was SuSE Linux using gcc version 3.3.1. The remainder of this section discusses the four modules individually with reference to the specific routines utilized for solving the case studies in this paper. However, due to the orthogonality of the design, the individual modules are easily replaceable with analogous routines.

6.1. BRANCH-AND-BOUND

At the inception of this project, the authors were unaware of any publicly available branch-and-bound libraries providing source code. Therefore, a custom branch-and-bound library was written in C++, libBandB; the current version of the code is Version 3.2, Singer (2004b). LibBandB provides several run-time configurable user options and heuristics. Unless otherwise noted, for comparison purposes, the case studies examined in this paper were performed without using any branch-and-bound heuristics. The method for selecting the next node on which to branch was to select the node in the branch-and-bound tree with the least lower bound. At each node, the variable with the greatest distance between its upper bound and lower bound was selected for branching, and the branching was performed by bisection. The tolerance to which each case study was solved is noted in the Case Studies Section.

6.2. LOCAL OPTIMIZATION

As previously mentioned, local optimization of the original problem was utilized to provide upper bounds at each node and local optimization of a convex relaxation was utilized to provide lower bounds at each node. In general, the most computationally expensive aspect to solving Problem 2.1 is the repeated numerical integrations required to provide both objective function values and gradients for the optimization. Thus, a sequential quadratic programming routine was chosen in order to limit the number of integration calls required to perform the optimization. Because most of the case studies examined are small, dense problems, NPSOL version 5.0 (Gill et al., 1998) was utilized. NPSOL's optimization tolerance was always set

at least two orders of magnitude smaller than the branch-and-bound tolerance and at least two orders of magnitude greater than the integration tolerance for each specific problem.

6.3. NUMERICAL INTEGRATION

In order to compute both the objective function and the gradient of the objective function, the integral and the associated ODEs for Problem 2.1 must be numerically solved. Because many of the problems addressed in the Case Studies Section are stiff and dense, CVODES (Hindmarsh and Serban, 2002) was chosen for numerical integration, and the sensitivities were computed via the staggered corrector option (Feehery et al., 1997). By design, CVODES is capable of numerically solving any system of nonlinear ODEs possessing Lipschitz continuity of the right hand side of the ODEs. For the problems considered in this paper, the upper bounding problem obeys this assumption at least between points of explicit time discontinuities; CVODES is reset at time events. However, by construction, the ODEs in the lower bounding problem do not necessarily possess Lipschitz continuous right hand sides between explicit time discontinuities, for the lower bound possesses state events. This condition arises because the convex and concave relaxations of the right-hand sides of the ODEs defining $\dot{\mathbf{c}}$ and $\dot{\mathbf{C}}$ may not be smooth. If this condition occurs, then the derivatives of the relaxations may be discontinuous causing nonsmooth shifts through time in the linearizations of the relaxation equations. Although this nonsmoothness does not pose any theoretical limitations (see Singer and Barton, Submitted), this nonsmoothness often causes CVODES to fail. Therefore, a discontinuity locking layer with bisection event detection was written around CVODES. Because any valid support linearization yields valid relaxations \mathbf{c} and \mathbf{C} , the mode actually selected at an event is arbitrary. Because the selection of the integration mode at an event is arbitrary, chattering in the numerical integration is prevented by locking the residual routine into an arbitrary mode while chattering ensues. A complete discussion of discontinuity locking and its impact on the performance of the numerical integration is beyond the scope of this paper; the interested reader is referred to Singer (2004a). The authors note that chattering bears no implication on the quality of the integration, for locking in any arbitrary mode has only the effect of isolating one of the distinct linearizations. However, the selection of the linearization can affect the tightness of the resulting relaxation. Unfortunately, no *a priori* method exists to indicate which linearization provides a tighter relaxation. Both the absolute and relative tolerances for the integrator were set to 10^{-8} for all of the case studies; this value was always at least two orders of magnitude smaller than the tolerance set for the optimization routine.

As previously mentioned, numerically solving the ODEs is the most computationally expensive aspect of solving Problem 2.1. Clearly, integrating the lower bounding problem is more expensive than integrating the upper bounding problem, for the lower bounding problem is both four times larger than the upper bounding problem (in number of state equations) and requires discontinuity locking. In general, each major iteration in the optimization routine requires an integration call because the lower bound is parameter dependent. However, for problems only requiring the objective function at fixed time points (i.e., the objective function does not contain an integral), the affine structure of \mathbf{c} and \mathbf{C} may be exploited to reduce the number of required integrations. In order to compute the lower bound at a fixed time \underline{t} , we must be able to compute $\mathcal{X}(\underline{t})$, $\mathbf{c}(\underline{t}, \mathbf{p})$, and $\mathbf{C}(\underline{t}, \mathbf{p})$. By definition, the state bounds are parameter independent; therefore, they need not be recomputed for each major iteration of the optimizer. Furthermore, because \mathbf{c} and \mathbf{C} are both affine, from Singer and Barton (2004), we know they have the following structure:

$$\begin{aligned}\mathbf{c}(\underline{t}, \mathbf{p}) &= \mathbf{M}_c(\underline{t})\mathbf{p} + \mathbf{n}_c(\underline{t}) \\ \mathbf{C}(\underline{t}, \mathbf{p}) &= \mathbf{M}_C(\underline{t})\mathbf{p} + \mathbf{n}_C(\underline{t}).\end{aligned}$$

Thus, by integrating once and subsequently storing the state bounds, $\mathbf{M}_c(\underline{t})$, $\mathbf{M}_C(\underline{t})$, $\mathbf{n}_c(\underline{t})$, and $\mathbf{n}_C(\underline{t})$, $\mathbf{c}(\underline{t}, \mathbf{p})$ and $\mathbf{C}(\underline{t}, \mathbf{p})$ can be computed via linear algebra instead of numerical integration on future major iterations of the optimizer. The computation of $\mathbf{M}_c(\underline{t})$ and $\mathbf{M}_C(\underline{t})$ is effectively free, for they are simply the parametric sensitivities, which are already computed for the calculation of the gradient of the lower bound. The values $\mathbf{n}_c(\underline{t})$ and $\mathbf{n}_C(\underline{t})$ can either be computed by augmenting the original system with an additional set of differential equations as shown in Singer and Barton (2004) or via linear algebra. The error in $\mathbf{c}(\underline{t}, \mathbf{p})$ and $\mathbf{C}(\underline{t}, \mathbf{p})$ introduced by the linear algebra calculation is negligible if the problem is scaled such that $p_i^U - p_i^L$ is order one for $i = 1, \dots, n_p$.

An interesting aspect of exploiting the linear algebra for solving the lower bounding problem is that this technique also enables one to compute an upper bound for the problem without performing any integration. From Theorem 3.6, once $\mathbf{c}(\underline{t}, \mathbf{p})$ and $\mathbf{C}(\underline{t}, \mathbf{p})$ have been computed, the only additional information required for constructing a concave overestimator for a state composition is determining z_{max} . A rigorous upper bound could then be computed by maximizing the concave overestimator for the objective function, and we note simply that the upper bound converges to the objective function value in an analogous manner to the convergence of the lower bound to the objective function. Furthermore, because the state relaxations are affine, depending on the structure of the objective function,

Problem 2.1 may be completely reducible to solving a sequence of converging upper and lower bounding linear programs requiring only one integration per branch-and-bound node. However, because solving the original problem locally is relatively inexpensive computationally, periodically solving the original problem locally provides a better incumbent by which to fathom in the branch-and-bound algorithm hence accelerating the convergence of the algorithm.

6.4. RESIDUAL EVALUATION

The residual evaluation is actually the most complicated of the four modules of the code. While the upper bounding problem is easily coded manually, the generation of code for the lower bounding problem requires the application of Theorem 3.6, the computation of the state bounds (Corollary 2.5 in Singer and Barton, Submitted) and the application of the linear bounding theorem (Theorem 3.2 in Singer and Barton, Submitted). To complicate matters further, this analysis is unique for each individual problem. Because the manual application of the requisite theory would be both tedious and error prone, a compiler was written to automatically generate the residual routines for both the upper and lower bounding problems. The user writes the dynamics of the problem in a simple input language, and the compiler generates a series of discontinuity locked Fortran 77 subroutines defining the residuals for both the upper and lower bounding problems. This residual file is simply compiled and linked to the numerical ODE solver, the local optimizer, and the branch-and-bound routine.

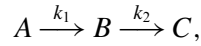
7. Case Studies

This section examines several literature example problems. As previously stated, unless otherwise noted, no branch-and-bound heuristics are utilized to accelerate convergence. For point constrained problems, the linear structure of the convex relaxation is always exploited, and the lower bounding integration is only performed for the first optimization function call at a given node. The set \bar{X} is defined for each of the case studies for which this technique is utilized. Finally, the notation x^* represents the reference trajectory for variable x ; the reference trajectories for both states and parameters are given for each problem. The computations were all performed on an AMD Athlon XP2000+ operating at 1667 MHz running SuSE linux kernel 2.4.

7.1. FIRST-ORDER IRREVERSIBLE SERIES REACTION

The first example is a simple parameter estimation problem first proposed by Tjoa and Biegler (1991). As with all parameter estimation problems, the

objective is to minimize the error between “experimental” data and the prediction of the model. For this problem, the kinetics are given by



and the data were taken from Floudas et al. (1999). The problem is mathematically formulated below.

$$\min_{\mathbf{k}} \sum_{\mu=1}^{10} \sum_{i=1}^2 (\hat{x}_{\mu,i} - x_{\mu,i})^2,$$

where $\hat{\mathbf{x}}_i(t, \mathbf{k})$ is given by

$$\begin{aligned} \frac{d\hat{x}_1}{dt} &= -k_1 \hat{x}_1 \\ \frac{d\hat{x}_2}{dt} &= k_1 \hat{x}_1 - k_2 \hat{x}_2 \end{aligned}$$

$$\hat{\mathbf{x}}(0) = (1, 0)$$

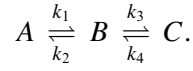
$$\mathbf{k} \in [0, 10]^2$$

$$(\mathbf{k}^*, \mathbf{x}^*(t)) = (\mathbf{k}^L, \mathbf{x}^L(t)) \quad \forall t \in (t_0, t_f].$$

GDOC was used to solve this problem to a global minimum within an absolute branch-and-bound tolerance of 10^{-4} . GDOC terminated in 0.036 CPU seconds with an objective function value of 1.22×10^{-6} at the point (5.0, 1.0), but the absolute value of the objective function at termination should be viewed with some skepticism considering the tolerance of the branch-and-bound code. Because the data in Floudas et al. (1999) differs from the exact prediction of the model only by roundoff error, the global minimum value of this problem is effectively zero (within the propagation of the roundoff error) since the model exactly fits the data. Furthermore, although the problem is nonconvex, the objective function possesses only one local minimum. By construction, our algorithm guarantees that the convex relaxation for this problem is nonnegative. As expected, utilizing a local solution for the upper bound, the problem was trivially solved at the root node.

7.2. FIRST-ORDER REVERSIBLE SERIES REACTION

The second example problem presented, also attributed to Tjoa and Beigler (1991), extends the first example by allowing reversibility of the kinetic equation:



The data for this problem were also obtained from Floudas et al. (1999). Aside from simply possessing more degrees of freedom, solving the problem with reversible kinetics is slightly more difficult than solving the problem with irreversible kinetics because the dynamics are nonquasimonotone. However, this difficulty is overcome by utilizing Theorem 4.1 to generate state bounds; the state bounding equations utilized in this case study are found in Appendix A. The problem is mathematically stated as

$$\min_{\mathbf{k}} \sum_{\mu=1}^{20} \sum_{i=1}^3 (\hat{x}_{\mu,i} - x_{\mu,i})^2$$

where $\hat{\mathbf{x}}_i(t, \mathbf{k})$ is given by

$$\begin{aligned} \dot{\hat{x}}_1 &= -k_1 \hat{x}_1 + k_2 \hat{x}_2 \\ \dot{\hat{x}}_2 &= k_1 \hat{x}_1 - (k_2 + k_3) \hat{x}_2 + k_4 \hat{x}_3 \\ \dot{\hat{x}}_3 &= k_3 \hat{x}_2 - k_4 \hat{x}_3 \end{aligned}$$

$$\hat{\mathbf{x}}(0) = (1, 0, 0)$$

$$\bar{\mathbf{x}} = [0, 1]^3$$

$$\mathbf{k} \in [0, 10]^2 \times [10, 50]^2$$

$$(\mathbf{k}^*, \mathbf{x}^*(t)) = (\mathbf{k}^L, \mathbf{x}^L(t)) \quad \forall t \in (t_0, t_f].$$

As with the previous problem, using a branch-and-bound tolerance of 10^{-4} , this problem also solved at the root node because the literature data for this problem also contains no error. GDOC terminated in 0.044 CPU s with an objective function value of 7.69×10^{-5} at the point (4.00, 2.00, 40.0, 20.0).

7.3. CATALYTIC CRACKING OF GAS OIL

The following problem is another parameter estimation problem attributed to Tjoa and Biegler (1991). The problem is stated as

$$\min_{\mathbf{k}} \sum_{\mu=1}^{20} \sum_{i=1}^2 (\hat{x}_{\mu,i} - x_{\mu,i})^2,$$

where $\hat{\mathbf{x}}_i(t, \mathbf{k})$ is given by

$$\dot{\hat{x}}_1 = -(k_1 + k_3)\hat{x}_1^2$$

$$\dot{\hat{x}}_2 = k_1\hat{x}_1^2 - k_2\hat{x}_2$$

$$\hat{\mathbf{x}}(0) = (1, 0)$$

$$\bar{\mathcal{X}} = [0, 1]^2$$

$$\mathbf{k} \in [0, 20]^2$$

$$(\mathbf{k}^*, \mathbf{x}^*) = (\mathbf{k}^{mid}, \mathbf{x}^{mid}(\underline{t})) \quad \forall \underline{t} \in (t_0, t_f].$$

Unlike the previous two problems, the fitted data for this problem at the global minimum do not exactly fit the “experimental” data. Because the objective function value at termination for this problem is close to zero, the authors believe that solving the problem to an absolute error tolerance is the most sensible approach. Table 1 presents the results for solving this problem.

7.4. SINGULAR CONTROL

The next problem we examine is the singular control problem originally formulated in Luus (1990). The problem is given by

$$\min_{u(t)} \int_{t_0}^{t_f} x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3u^2)^2 dt$$

subject to

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_3u + 16t - 8$$

$$\dot{x}_3 = u$$

$$\mathbf{x}_0 = (0, -1, -\sqrt{5})$$

$$t \in (0, 1] \quad -4 \leq u(t) \leq 10$$

$$(\mathbf{p}^*, \mathbf{x}^*(\underline{t})) = (\mathbf{p}^U, \mathbf{x}^U(\underline{t})) \quad \forall \underline{t} \in (t_0, t_f],$$

Table 1. Results for catalytic cracking of gas oil problem

Objective function	Location	CPUs	Nodes	Absolute tolerance
2.66×10^{-3}	(12.2, 7.98, 2.22)	0.18	1	10^{-2}
2.66×10^{-3}	(12.2, 7.98, 2.22)	5.78	83	10^{-3}
2.66×10^{-3}	(12.2, 7.98, 2.22)	12.23	189	10^{-4}
2.66×10^{-3}	(12.2, 7.98, 2.22)	19.40	295	10^{-5}

where the variable \mathbf{p} in the reference trajectory derives from a piecewise constant control parameterization. Because this problem does not derive from a physical system, no natural bounding set exists. Studying the singular control problem enables us to examine a very important tradeoff in implementing a solution strategy for problems with integral objective functions. The theoretical presentation of the algorithm has emphasized relaxing Problem 2.1 utilizing Corollary 3.4 to relax the integral directly. As previously stated, in applying Corollary 3.4, the relaxation for the integrand must be constructed for each fixed $\underline{t} \in (t_0, t_f]$. Under this requirement, the affine nature of the relaxations $\mathbf{c}(\underline{t}, \mathbf{p})$ and $\mathbf{C}(\underline{t}, \mathbf{p})$ cannot be exploited, and an integration must be performed for each major iteration of the optimizer. Trivially, however, the singular control problem can be reformulated as

$$\min_{u(t)} z(t_f)$$

subject to

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_3 u + 16t - 8 \\ \dot{x}_3 &= u \\ \dot{z} &= x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3 u^2)^2 \\ z(0) &= 0 \\ \mathbf{x}_0 &= (0, -1, -\sqrt{5}) \\ t \in (0, 1] \quad &-4 \leq u(t) \leq 10 \\ (\mathbf{p}^*, \mathbf{x}^*(\underline{t})) &= (\mathbf{p}^U, \mathbf{x}^U(\underline{t})) \quad \forall \underline{t} \in (t_0, t_f], \end{aligned}$$

where the integrand has simply been replaced by a quadrature variable; this transformation is quite common in the literature and is always possible. Now, the affine nature of the state relaxations can be exploited, for we only need relaxations at a fixed time. The problem was solved by GDOC in both the original formulation and the quadrature variable reformulation; the results are presented in Tables 2 and 3. For each formulation, the problem was solved to an absolute tolerance of 10^{-3} with piecewise constant control profiles on equally spaced meshes with a varying number of time intervals. Additionally, each problem was solved with and without post-processing, Lagrangian reduce heuristics as defined in Sahinidis (1995). The authors note that the utilization of reduce heuristics does not affect the quality of the solution but only enhance the performance of the algorithm.

As expected, the number of nodes required to solve the integral formulation is less than the number of nodes required to solve the quadrature

formulation. This is a common feature for problems with nonlinear dynamics, particularly in cases where the integrand is highly nonlinear, for adding additional nonlinearity in the dynamics decreases the tightness of the state bounds and the tightness of the state relaxations. For the problem considered, the CPU time required for solution is also less for the integral formulation than for the quadrature formulation. However, one quickly notes that the cost per node for the quadrature formulation is smaller than the cost per node for the integral formulation; furthermore, the relative cost per node for the quadrature formulation decreases with the increasing number of parameters because the cost of the integration increases slightly with an increasing number of parameters. Thus, this tradeoff must be considered individually for each problem. In general though, the authors anticipate that an integral formulation will almost always outperform a quadrature reformulation, especially for small problems or problems with highly nonlinear integrands.

7.5. OIL SHALE PYROLYSIS

The final problem examined is a fixed final time formulation of the Oil Shale Pyrolysis problem originally posed in Luus (1990). The problem is stated below.

$$\begin{aligned}
 \min_{u(t)} & -x_2(t_f) \\
 \dot{x}_1 &= -(k_1x_1 + k_3x_1x_2 + k_4x_1x_2 + k_5x_1x_2) \\
 \dot{x}_2 &= k_1x_1 - k_2x_2 + k_3x_1x_2 \\
 k_i &= a_i \exp\left[\frac{-b_i/R}{698.15 + 50u}\right], \quad i = 1, \dots, 5 \\
 \mathbf{x}(t_0) &= (1, 0) \\
 \bar{\mathcal{X}} &= [0, 1]^2 \\
 t \in (0, 1] \quad & 0 \leq u(t) \leq 1 \\
 (\mathbf{p}^*, \mathbf{x}^*(t)) &= (\mathbf{p}^U, \mathbf{x}^U(t)), \quad \forall t \in (t_0, t_f],
 \end{aligned}$$

Table 2. Results for singular control problem in original formulation

No. time intervals	Objective function	Location	CPUs	Nodes	Heuristics
1	0.497	(4.07)	2.0	21	No
1	0.497	(4.07)	1.8	15	Yes
2	0.277	(5.57, -4.00)	26.5	89	No
2	0.277	(5.57, -4.00)	22.5	47	Yes
3	0.146	(8.05, -1.85, 6.09)	814.3	1225	No
3	0.146	(8.05, -1.85, 6.09)	540.3	489	Yes

Table 3. Results for singular control problem in quadrature variable reformulation

No. time intervals	Objective function	Location	CPUs	Nodes	Heuristics
1	0.497	(4.07)	5.2	33	No
1	0.497	(4.07)	3.4	15	Yes
2	0.277	(5.57, -4.00)	55.1	193	No
2	0.277	(5.57, -4.00)	28.8	49	Yes
3	0.146	(8.05, -1.85, 6.09)	1929.0	3931	No
3	0.146	(8.05, -1.85, 6.09)	816.3	789	Yes

Table 4. Results for oil shale pyrolysis problem

No. time intervals	objective function	location	CPUs	nodes	heuristics
1	-0.348	(0.231)	27.3	127	No
1	-0.348	(0.231)	26.2	115	Yes
2	-0.351	(0.431, 0.00)	1720.7	5807	No
2	-0.351	(0.431, 0.00)	1597.3	4933	Yes

where the variable \mathbf{p} in the reference trajectory derives from a piecewise constant control parameterization, and the values for a_i and b_i/R are defined in Floudas et al. (1999). This problem was solved within an absolute error of 10^{-3} for a piecewise constant control profile. The problem was solved with both one and two equally spaced time intervals. The results are found in Table 4.

From the relative change in the objective function between using a one stage and two stage constant profile, we can conclude that increasing the number of control stages beyond two is unnecessary. Furthermore, this problem clearly demonstrates the worst case exponential complexity of the branch-and-bound algorithm. This problem does not converge rapidly because the state bounds tighten very slowly with the branching in the parameter space. This in turn implies weaker state relaxations and hence slower overall convergence.

8. Conclusions

In this paper, we presented a branch-and-bound algorithm for globally solving optimization problems with an integral objective function subject to nonquasimonotone differential equations. The branch-and-bound algorithm converges to the global solution by creating a sequence of converging upper and lower bounds on the original problem. In general, a local solution to the optimization problem is utilized as the upper bound,

and we show that a convex relaxation for the problem is derived by integrating a convex relaxation of the integrand. Because the integrand is typically defined by a composition of both state variables and parameters, we illustrate a dynamic extension to the composition technique of McCormick (1976) that incorporates previously developed ideas concerning convex bounding of the solution of nonquasimonotone differential equations. Utilizing the derived relaxations, we prove the convergence of the branch-and-bound algorithm. The remainder of the paper was devoted to an implementation of the developed theory. In order to solve global dynamic optimization problems, a compiler was developed for applying the relaxation theory to generate automatically residual files for a numerical ODE solver. The numerical ODE solver was then combined with a local optimization routine and a branch-and-bound program to solve Problem 2.1. Several case study problems were solved utilizing the implementation.

The algorithm we have presented for solving Problem 2.1 has four distinct advantages over other deterministic algorithms in the literature. First, the problem scales well in the complexity of the lower bounding problem. Constructing convex relaxations only requires four additional state equations for each state and no additional equations for increasing numbers of parameters. Second, the problem exploits the integral structure of the objective function to generate tighter relaxations than reformulating the integral as an additional quadrature variable. Third, the theory we have developed extensively exploits linear relaxations whenever possible and thus replaces numerical integration with linear algebra computations. Fourth, the algorithm generates tight state bounds for nonquasimonotone ODEs subject to the availability of physical bounds. However, the efficient solution of Problem 2.1 for large scale problems still remains a daunting challenge. The compiler developed for solving the problems in this paper is merely a proof of concept prototype. A second generation compiler is currently being developed that extensively incorporates ideas from algorithmic differentiation to eliminate many redundant computations in the residual calculation. While the new compiler implementation will improve the numerical efficiency of the algorithm, many research areas still remain open for efficiently solving Problem 2.1. In particular, tighter convex relaxations, tighter state bounds, and reliable heuristics need to be developed before efficiently solving global dynamic optimization problems for large scale systems becomes a reality.

Acknowledgements

ABS would like to thank Dr. Benoît Chachuat for nearly daily discussions concerning the refinement of the implementation of the theory presented in this paper. In particular, Dr. Chachuat contributed to and lobbied for

the implementation exploiting the affine structure of the convex and concave relaxation for the purpose of decreasing the computational time for computing both upper and lower bounds for Problem 2.1. This material is based upon work supported by the National Science Foundation under Grant CTS0120441.

Appendix A: Derivation of state bounds for Section 7.2

From Section 7.2, the following nonquasimonotone differential equations are given defining the state variables (where the hats have been dropped for notational convenience):

$$\begin{aligned}\dot{x}_1 &= -k_1 x_1 + k_2 x_2 \\ \dot{x}_2 &= k_1 x_1 - (k_2 + k_3)x_2 + k_4 x_3 \\ \dot{x}_3 &= k_3 x_2 - k_4 x_3\end{aligned}$$

where the sets

$$\bar{\mathcal{X}} = [\bar{x}_1^L, \bar{x}_1^U] \times [\bar{x}_2^L, \bar{x}_2^U] \times [\bar{x}_3^L, \bar{x}_3^U]$$

and

$$K = [k_1^L, k_1^U] \times [k_2^L, k_2^U] \times [k_3^L, k_3^U] \times [k_4^L, k_4^U]$$

are utilized instead of their respective numeric values. From Theorem 4.1, the following equations define the state bounds for the above differential equations:

$$\begin{aligned}\dot{x}_1^L &= -\max\{k_1^L x_1^L, k_1^U x_1^L\} + \min\{k_2^L \max\{x_2^L, \bar{x}_2^L\}, k_2^L \\ &\quad \times \min\{x_2^U, \bar{x}_2^U\}, k_2^U \max\{x_2^L, \bar{x}_2^L\}, k_2^U \min\{x_2^U, \bar{x}_2^U\}\}\end{aligned}$$

$$\begin{aligned}\dot{x}_2^L &= \min\{k_1^L \max\{x_1^L, \bar{x}_1^L\}, k_1^L \min\{x_1^U, \bar{x}_1^U\}, k_1^U \max\{x_1^L, \bar{x}_1^L\}, k_1^U \\ &\quad \times \min\{x_1^U, \bar{x}_1^U\}\} - \max\{k_2^L x_2^L, k_2^U x_2^L\} - \max\{k_3^L x_2^L, k_3^U x_2^L\} \\ &\quad + \min\{k_4^L \max\{x_3^L, \bar{x}_3^L\}, k_4^L \min\{x_3^U, \bar{x}_3^U\}, k_4^U \max\{x_3^L, \bar{x}_3^L\}, k_4^U \\ &\quad \times \min\{x_3^U, \bar{x}_3^U\}\}\end{aligned}$$

$$\begin{aligned}\dot{x}_3^L &= -\max\{k_4^L x_3^L, k_4^U x_3^L\} + \min\{k_3^L \max\{x_2^L, \bar{x}_2^L\}, k_3^L \min\{x_2^U, \bar{x}_2^U\}, k_3^U \\ &\quad \times \max\{x_2^L, \bar{x}_2^L\}, k_3^U \min\{x_2^U, \bar{x}_2^U\}\}\end{aligned}$$

$$\begin{aligned}\dot{x}_1^U &= -\min\{k_1^L x_1^U, k_1^U x_1^U\} + \max\{k_2^L \max\{x_2^L, \bar{x}_2^L\}, k_2^L \min\{x_2^U, \bar{x}_2^U\}, k_2^U \\ &\quad \times \max\{x_2^L, \bar{x}_2^L\}, k_2^U \min\{x_2^U, \bar{x}_2^U\}\}\end{aligned}$$

$$\begin{aligned} \dot{x}_2^U = & \max\{k_1^L \max\{x_1^L, \bar{x}_1^L\}, k_1^L \min\{x_1^U, \bar{x}_1^U\}, k_1^U \max\{x_1^L, \bar{x}_1^L\}, k_1^U \\ & \times \min\{x_1^U, \bar{x}_1^U\}\} - \min\{k_2^L x_2^U, k_2^U x_2^U\} - \min\{k_3^L x_2^U, k_3^U x_2^U\} \\ & + \max\{k_4^L \max\{x_3^L, \bar{x}_3^L\}, k_4^L \min\{x_3^U, \bar{x}_3^U\}, k_4^U \max\{x_3^L, \bar{x}_3^L\}, \\ & \times k_4^U \min\{x_3^U, \bar{x}_3^U\}\} \end{aligned}$$

$$\begin{aligned} \dot{x}_3^U = & -\min\{k_4^L x_3^U, k_4^U x_3^U\} + \max\{k_3^L \max\{x_2^L, \bar{x}_2^L\}, k_3^L \min\{x_2^U, \bar{x}_2^U\}, k_3^U \\ & \times \max\{x_2^L, \bar{x}_2^L\}, k_3^U \min\{x_2^U, \bar{x}_2^U\}\}. \end{aligned}$$

References

1. Adams, M. and Guillemin, V. (1996), *Measure Theory and Probability*. Birkhäuser, Boston.
2. Adjiman, C., Dallwig, S., and Floudas, C. (1998a), A global optimization method, α BB, for general twice-differentiable constrained NLPs – II. Implementation and computational results, *Computers and Chemical Engineering* 22(9), 1159–1179.
3. Adjiman, C., Dallwig, S., Floudas, C., and Neumaier, A. (1998b), A global optimization method, α BB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers and Chemical Engineering* 22(9), 1137–1158.
4. Arthur E. Bryson, Jr. and Ho, Y.-C. (1975), *Applied Optimal Control*. Taylor & Francis, Briston, Pennsylvania.
5. Bruschi, R. and Schappelle, R. (1973), Solution of highly constrained optimal control problems using nonlinear programming, *AIAA Journal* 11(2), 135–136.
6. Esposito, W.R. and Floudas, C.A. (2000a), Deterministic global optimization in nonlinear optimal control problems. *Journal of Global Optimization* 17, 97–126.
7. Esposito, W.R. and Floudas, C.A. (2000b), Global optimization for the parameter estimation of differential-algebraic systems, *Industrial and Engineering Chemistry Research* 39, 1291–1310.
8. Feehery, W., Tolsma, J., and Barton, P. (1997), Efficient sensitivity analysis of large-scale differential-algebraic systems, *Applied Numerical Mathematics* 25(1), 41–54.
9. Floudas, C.A., Pardalos, P.M., Adjiman, C.S., Esposito, W.R., Gumus, Z.H., Harding, S.T., Klepeis, J.L., Meyer, C.A., and Schweiger, C.A. (1999), *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, Dordrecht.
10. Gill, P.E., Murray, W., Saunders, M.A., and Wright, M.H. (1998), *User's Guide for NPSOL 5.0: A Fortran Package for Nonlinear Programming*. Technical report, Stanford University.
11. Harrison, G. (1977), Dynamic models with uncertain parameters, In: Avula, X. (ed.), *Proceedings of the First International Conference on Mathematical Modeling*, vol. 1, pp. 295–304.
12. Hindmarsh, A.C. and Serban, R. (2002), *User Documentation for CVODES, An ODE Solver with Sensitivity Analysis Capabilities*. Technical report, Lawrence Livermore National Laboratory.
13. Horst, R. and Tuy, H. (1993), *Global Optimization*. Springer-Verlag, Berlin.
14. Luus, R. (1990), Optimal control by dynamic programming using systematic reduction in grid size, *International Journal of Control* 5, 995–1013.
15. Maranas, C. and Floudas, C. (1994), Global minimum potential energy conformations of small molecules, *Journal of Global Optimization* 4, 135–170.

16. McCormick, G.P. (1976). Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems, *Mathematical Programming* 10, 147–175.
17. Neuman, C. and Sen, A. (1973), A suboptimal control algorithm for constrained problems using cubic splines, *Automatica* 9, 601–613.
18. Papamichail, I. and Adjiman, C.S. (2002), A rigorous global optimization algorithm for problems with ordinary differential equations, *Journal of Global Optimization* 24, 1–33.
19. Pontriagin, L. (1962), *The Mathematical Theory of Optimal Processes*. Interscience Publishers, New York.
20. Ryoo, H. and Sahinidis, N. (1995), Global optimization of nonconvex NLPs and MINLPs with application in process design, *Computers and Chemical Engineering* 19(5), 551–566.
21. Singer, A.B. (2004a). *Global Dynamic Optimization*, Ph.D. thesis, Massachusetts Institute of Technology.
22. Singer, A.B. (2004b). *LibBandB Version 3.2 Manual*, Technical report, Massachusetts Institute of Technology.
23. Singer, A.B. and Barton, P.I. (2004), Global Solution of linear dynamic embedded optimization problems, *Journal of Optimization Theory and Applications* 121(3), 613–646.
24. Teo, K., Goh, G., and Wong, K. (1991), *A Unified Computational Approach to Optimal Control Problems*. Pitman Monographs and Surveys in Pure and Applied Mathematics. John Wiley & Sons, New York.
25. Tjoa, I. and Biegler, L. (1991), Simultaneous solution and optimization strategies for parameter-estimation of differential-algebraic equation systems, *Industrial and Engineering Chemistry Research* 30(2), 376–385.
26. Troutman, J.L. (1996). *Variational Calculus and Optimal Control: Optimization with Elementary Convexity*, 2nd edn., Springer-Verlag, New York.
27. Tsang, T., Himmelblau, D., and Edgar, T. (1975), Optimal control via collocation and nonlinear programming, *International Journal of Control* 21, 763–768.
28. Walter, W. (1970), *Differential and Integral Inequalities*, Springer-Verlag, Berlin.