

Deterministic Global Optimization of Nonlinear Dynamic Systems

Youdong Lin and Mark A. Stadtherr*

Department of Chemical and Biomolecular Engineering
University of Notre Dame, Notre Dame, IN 46556, USA

(August 1, 2006)
revised, December 1, 2006

*Author to whom all correspondence should be addressed. E-mail: markst@nd.edu

Abstract

A new approach is described for the deterministic global optimization of dynamic systems, including optimal control problems. The method is based on interval analysis and Taylor models and employs a type of sequential approach. A key feature of the method is the use of a new validated solver for parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued parameters. This is combined with a new technique for domain reduction based on the use of Taylor models in an efficient constraint propagation scheme. The result is that an ϵ -global optimum can be found with both mathematical and computational certainty. Computational studies on benchmark problems are presented showing that this new approach provides significant improvements in computational efficiency, well over an order of magnitude in most cases, relative to other recently described methods.

Keywords: Global optimization; Dynamic modeling; Interval analysis; Validated computing; Optimal control

Introduction

The dynamic behavior of many physical systems of practical interest can be modeled using systems of ordinary differential equations (ODEs). Optimization problems involving these dynamic models arise when optimal performance measures are sought for such systems. There are many applications of dynamic optimization, including parameter estimation from time series data, determination of optimal operating profiles for batch and semi-batch processes, optimal start-up, shut-down, and switching of continuous system, etc.

To address this problem, one class of methods is based on discretization techniques to reduce what is essentially an infinite-dimensional optimization problem to a finite-dimensional problem. Two different discretization strategies are available: (a) the complete discretization or simultaneous approach,^{1,2} in which both state variables and control parameters are discretized, and (b) the control parameterization or sequential approach,^{3,4} in which only the control parameters are discretized. In this paper, only the control parameterization approach is considered. Since these problems are often nonconvex and thus may exhibit multiple local solutions, the classical techniques based on solving the necessary conditions for a local minimum may fail to determine the global optimum. This is true even for a rather simple temperature control problem with a batch reactor.⁵ Therefore, there is a need to develop *global* optimization algorithms which can rigorously *guarantee* optimal performance.

The deterministic global optimization of dynamic systems has been a topic of significant recent interest. Esposito and Floudas^{6,7} used the α BB approach^{8,9} for addressing this problem. In this method convex underestimating functions are used in connection with a branch-and-bound framework. A theoretical guarantee of attaining an ϵ -global solution is offered as long as rigorous underestimators are used, and this requires that sufficiently large values of α be used. However, the determination of proper values of α depends on the Hessian of the function being underestimated,

and, when the sequential approach is used, this matrix is not available in explicit functional form. Thus, Esposito and Floudas^{6,7} did not use rigorous values of α in their implementation of the sequential approach, and so did not obtain a theoretical guarantee of global optimality. This issue is discussed in more detail by Papamichail and Adjiman.¹⁰ Recently, alternative approaches have been given by Chachuat and Latifi¹¹ and by Papamichail and Adjiman^{10,12} that provide a theoretical guarantee of ϵ -global optimality. However, this is achieved at a high computational cost. Singer and Barton¹³ have recently described a branch-and-bound approach for determining an ϵ -global optimum with significantly less computational effort. In this method, convex underestimators and concave overestimators are used to construct two bounding initial value problems (IVPs), which are then solved to obtain lower and upper bounds on the trajectories of the state variables.¹⁴ However, as implemented,¹⁵ the bounding IVPs are solved using standard numerical methods that do not provide guaranteed error estimates. Thus, strictly speaking, this approach cannot be regarded as providing computationally guaranteed results.

We present here a new approach for the deterministic global optimization of dynamic systems. This method is based on interval analysis and Taylor models and employs a type of sequential approach. Instead of the usual branch-and-bound approach, we incorporate a new domain reduction technique, and thus use a type of branch-and-reduce strategy.¹⁶ A key feature of the method is the use of a new validated solver¹⁷ for parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued parameters. The result is that an ϵ -global optimum can be found with both mathematical and computational certainty. The computational efficiency of this approach will be demonstrated through application to benchmark problems, including optimal control problems. In the context of optimal control, a global minimization algorithm based on different validated ODE solvers has recently been presented by Rauh et al.¹⁸

The remainder of this paper is organized as follows. In the next section, we present the mathematical formulation of the problem to be solved. This is followed by a section that provides background on interval analysis and Taylor models, a section in which we review the new validated method¹⁷ for parametric ODEs, and a section in which we outline the algorithm for deterministic global optimization of dynamic systems. Finally, we present the results of some numerical experiments that demonstrate the effectiveness of the approach presented.

Problem Statement

In this section we give the mathematical formulation of the nonlinear dynamic optimization problem to be solved. Assume the system is described by the nonlinear ODE model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$. Here \mathbf{x} is the vector of state variables (length n) and $\boldsymbol{\theta}$ is a vector of adjustable parameters (length p), which may be a parameterization of a control profile $\theta(t)$. The model is given as an autonomous system; a non-autonomous system can easily be converted into autonomous form by treating the independent variable (t) as an additional state variable with derivative equal to 1. The objective function ϕ is expressed in terms of the adjustable parameters and the values of the states at discrete points t_μ , $\mu = 0, 1, \dots, r$. That is, $\phi = \phi[\mathbf{x}_\mu(\boldsymbol{\theta}), \boldsymbol{\theta}; \mu = 0, 1, \dots, r]$, where $\mathbf{x}_\mu(\boldsymbol{\theta}) = \mathbf{x}(t_\mu, \boldsymbol{\theta})$. If an integral appears in the objective function, it can be eliminated by introducing an appropriate quadrature variable.

The optimization problem is then stated as

$$\begin{aligned}
& \min_{\boldsymbol{\theta}, \mathbf{x}_\mu} \phi[\mathbf{x}_\mu(\boldsymbol{\theta}), \boldsymbol{\theta}; \mu = 0, 1, \dots, r] & (1) \\
& \text{s.t.} & \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \\
& & \mathbf{x}_0 = \mathbf{x}_0(\boldsymbol{\theta}) \\
& & t \in [t_0, t_r] \\
& & \boldsymbol{\theta} \in \Theta.
\end{aligned}$$

Here Θ is an interval vector that provides upper and lower parameter bounds. We assume that \mathbf{f} is $(k - 1)$ -times continuously differentiable with respect to the state variables \mathbf{x} , and $(q + 1)$ -times continuously differentiable with respect to the parameters $\boldsymbol{\theta}$. We also assume that ϕ is $(q + 1)$ -times continuously differentiable with respect to the parameters $\boldsymbol{\theta}$. Here k is the order of the truncation error in the interval Taylor series (ITS) method to be used in the integration procedure, and q is the order of the Taylor model to be used to represent parameter dependence. When a typical sequential approach is used, an ODE solver is applied to the constraints with a given set of parameter values, as determined by the optimization routine. This effectively eliminates $\mathbf{x}_\mu, \mu = 0, 1, \dots, r$, and leaves a bound-constrained minimization in the adjustable parameters $\boldsymbol{\theta}$ only.

A new method is described below for the deterministic global solution of Problem (1). This method can also be easily extended to solve optimization problems with state path constraints and more general equality or inequality constraints involving the parameters. This can be done by adapting the constraint propagation procedure to handle the additional constraints.

Background

Interval analysis

A real interval X is defined as the set of real numbers lying between (and including) given upper and lower bounds; that is,

$$X = [\underline{X}, \overline{X}] = \{x \in \mathbb{R} \mid \underline{X} \leq x \leq \overline{X}\}. \quad (2)$$

Here an underline is used to indicate the lower bound of an interval and an overline is used to indicate the upper bound. A real interval vector $\mathbf{X} = (X_1, X_2, \dots, X_n)^T$ has n real interval components and can be interpreted geometrically as an n -dimensional rectangle or box. Note that in this context uppercase quantities are intervals, and lowercase quantities or uppercase quantities with underline or overline are real numbers.

Basic arithmetic operations with intervals are defined by

$$X \text{ op } Y = \{x \text{ op } y \mid x \in X, y \in Y\}, \quad (3)$$

where $\text{op} \in \{+, -, \times, \div\}$. Interval versions of the elementary functions can be similarly defined. It should be emphasized that, when machine computations with interval arithmetic operations are done, as in the procedures outlined below, the endpoints of an interval are computed with a directed (outward) rounding. That is, the lower endpoint is rounded down to the next machine-representable number and the upper endpoint is rounded up to the next machine-representable number. In this way, through the use of interval, as opposed to floating-point arithmetic, any potential rounding error problems are avoided. Several good introductions to interval analysis, as well as interval arithmetic and other aspects of computing with intervals, including their use in global optimization, are available.^{19–22} Implementations of interval arithmetic and elementary functions are also readily available, and recent compilers from Sun Microsystems directly support interval arithmetic and an interval data type.

For an arbitrary function $f(\mathbf{x})$, the interval extension, $F(\mathbf{X})$ encloses all possible values of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$. That is, $F(\mathbf{X}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$ encloses the range of $f(\mathbf{x})$ over \mathbf{X} . It is often computed by substituting the given interval \mathbf{X} into the function $f(\mathbf{x})$ and then evaluating the function using interval arithmetic. This so-called “natural” interval extension is sometimes wider than the actual range of function values, though it always includes the actual range. For example, the natural interval extension of $f(x) = x/(x-1)$ over the interval $X = [2, 3]$ is $F([2, 3]) = [2, 3]/([2, 3]-1) = [2, 3]/[1, 2] = [1, 3]$, while the true function range over this interval is $[1.5, 2]$. This overestimation of the function range is due to the “dependency” problem, which may arise when a variable occurs more than once in a function expression. While a variable may take on any value within its interval, it must take on the *same* value each time it occurs in an expression. However, this type of dependency is not recognized when the natural interval extension is computed. In effect, when the natural interval extension is used, the range computed for the function is the range that would occur if each instance of a particular variable was allowed to take on a different value in its interval range. For the case in which $f(\mathbf{x})$ is a single-use expression, that is, an expression in which each variable occurs only once, natural interval arithmetic will always yield the true function range. For example, rearrangement of the function expression used above gives $f(x) = x/(x-1) = 1 + 1/(x-1)$, and now $F([2, 3]) = 1 + 1/([2, 3]-1) = 1 + 1/[1, 2] = 1 + [0.5, 1] = [1.5, 2]$, the true range.

In some situations, dependency issues can be avoided through the use of the dependent subtraction operation (also known as the cancellation operation). Assume that there is an interval S that depends additively on the interval A . The dependent subtraction operation is defined by $S \ominus A = [\underline{S} - \underline{A}, \overline{S} - \overline{A}]$. For example, let $A = [1, 2]$, $B = [2, 3]$, $C = [3, 4]$ and $S = A + B + C = [6, 9]$. Say that only S is stored and that later it is desired to compute $A + B$ by subtracting C from S . Using the standard subtraction operation yields $S - C = [6, 9] - [3, 4] = [2, 6]$, which overestimates

the true $A + B$. Using the dependent subtraction operation, which is allowable since S depends additively on C , yields $S \ominus C = [6, 9] \ominus [3, 4] = [3, 5]$, which is the true $A + B$. For more general situations, there are a variety of other approaches that can be used to try to tighten interval extensions,^{19–22} including the use of Taylor models, as described in the next subsection.

Taylor models

Makino and Berz²³ have described a remainder differential algebra (RDA) approach for bounding function ranges and control of the dependency problem of interval arithmetic.²⁴ In this method, a function is represented using a model consisting of a Taylor polynomial and an interval remainder bound. Such a model is called a Taylor model.

One way of forming a Taylor model of a function is by using a truncated Taylor series. Consider a function $f : \mathbf{x} \in \mathbf{X} \subset \mathbb{R}^m \rightarrow \mathbb{R}$ that is $(q + 1)$ times partially differentiable on \mathbf{X} and let $\mathbf{x}_0 \in \mathbf{X}$. The Taylor theorem states that for each $\mathbf{x} \in \mathbf{X}$, there exists a $\zeta \in \mathbb{R}$ with $0 < \zeta < 1$ such that

$$f(\mathbf{x}) = \sum_{i=0}^q \frac{1}{i!} [(\mathbf{x} - \mathbf{x}_0) \cdot \nabla]^i f(\mathbf{x}_0) + \frac{1}{(q+1)!} [(\mathbf{x} - \mathbf{x}_0) \cdot \nabla]^{q+1} f[\mathbf{x}_0 + (\mathbf{x} - \mathbf{x}_0)\zeta], \quad (4)$$

where the partial differential operator $[\mathbf{g} \cdot \nabla]^k$ is

$$[\mathbf{g} \cdot \nabla]^k = \sum_{\substack{j_1 + \dots + j_m = k \\ 0 \leq j_1, \dots, j_m \leq k}} \frac{k!}{j_1! \dots j_m!} g_1^{j_1} \dots g_m^{j_m} \frac{\partial^k}{\partial x_1^{j_1} \dots \partial x_m^{j_m}}. \quad (5)$$

The last (remainder) term in Eq. (4) can be quantitatively bounded over $0 < \zeta < 1$ and $\mathbf{x} \in \mathbf{X}$ using interval arithmetic or other methods to obtain an interval remainder bound R_f . The summation in Eq. (4) is a q -th order polynomial (truncated Taylor series) in $(\mathbf{x} - \mathbf{x}_0)$ which we denote by $p_f(\mathbf{x} - \mathbf{x}_0)$. A q -th order Taylor model $T_f = p_f + R_f$ for $f(\mathbf{x})$ then consists of the polynomial p_f and the interval remainder bound R_f and is denoted by $T_f = (p_f, R_f)$. Note that $f \in T_f$ for $\mathbf{x} \in \mathbf{X}$ and thus T_f encloses the range of f over \mathbf{X} .

Taylor models of functions can also be formed by performing Taylor model operations. Arithmetic operations with Taylor models can be done using the remainder differential algebra (RDA) described by Makino and Berz.^{23–25} Let T_f and T_g be the Taylor models of the functions $f(\mathbf{x})$ and $g(\mathbf{x})$, respectively, over the interval $\mathbf{x} \in \mathbf{X}$. For $f \pm g$,

$$f \pm g \in T_f \pm T_g = (p_f, R_f) \pm (p_g, R_g) = (p_f \pm p_g, R_f \pm R_g). \quad (6)$$

Thus a Taylor model of $f \pm g$ is given by

$$T_{f \pm g} = (p_{f \pm g}, R_{f \pm g}) = (p_f \pm p_g, R_f \pm R_g). \quad (7)$$

For the product $f \times g$,

$$f \times g \in (p_f, R_f) \times (p_g, R_g) \subseteq p_f \times p_g + p_f \times R_g + p_g \times R_f + R_f \times R_g. \quad (8)$$

Note that $p_f \times p_g$ is a polynomial of order $2q$. Since a q -th order polynomial is sought for the Taylor model of $f \times g$, this term is split $p_f \times p_g = p_{f \times g} + p_e$. Here the polynomial $p_{f \times g}$ contains all terms of order q or less, and p_e contains the higher order terms. A q -th order Taylor model for the product $f \times g$ can then be given by $T_{f \times g} = (p_{f \times g}, R_{f \times g})$, with

$$R_{f \times g} = B(p_e) + B(p_f) \times R_g + B(p_g) \times R_f + R_f \times R_g. \quad (9)$$

Here $B(p) = P(\mathbf{X} - \mathbf{x}_0)$ denotes an interval bound on the polynomial $p(\mathbf{x} - \mathbf{x}_0)$ over $\mathbf{x} \in \mathbf{X}$. Similarly, an interval bound on an overall Taylor model $T = (p, R)$ will be denoted by $B(T)$, and is computed by obtaining $B(p)$ and adding it to the remainder bound R ; that is, $B(T) = B(p) + R$. The method we use to obtain the polynomial bounds is described below. In storing and operating on a Taylor model, only the coefficients of the polynomial part p are used, and these are point valued. However, when these coefficients are computed in floating point arithmetic, numerical errors may occur and they must be bounded. To do this in our current implementation of Taylor model arithmetic, we have used the ‘‘tallying variable’’ approach, as described by Makino and

Berz.²⁵ This approach has been analyzed in detail by Revol et al.²⁶ This results in an error bound on the floating point calculation of the coefficients in p being added to the interval remainder bound R .

Taylor models for the reciprocal operation, as well as the intrinsic functions (exponential, logarithm, sine, etc.) can also be obtained.^{23,25,27} Using these, together with the basic arithmetic operations defined above, it is possible to start with simple functions such as the constant function $f(x) = k$, for which $T_f = (k, [0, 0])$, and the identity function $f(x_i) = x_i$, for which $T_f = (x_{i0} + (x_i - x_{i0}), [0, 0])$, and then to compute Taylor models for very complicated functions. Altogether, it is possible to compute a Taylor model for any function that can be represented in a computer environment by simple operator overloading through RDA operations. It has been shown that, compared to other rigorous bounding methods, the Taylor model often yields sharper bounds for modest to complicated functional dependencies.^{23,24,28} A discussion of the uses and limitations of Taylor models has been given by Neumaier.²⁸

The range bounding of the interval polynomials $B(p) = P(\mathbf{X} - \mathbf{x}_0)$ is an important issue, which directly affects the performance of Taylor model methods. Unfortunately, exact range bounding of an interval polynomial is NP hard, and direct evaluation using interval arithmetic is very inefficient, often yielding only loose bounds. Thus, various bounding schemes^{28,29} have been used, mostly focused on exact bounding of the dominant parts of P , i.e., the first- and second-order terms. However, exact bounding of a general interval quadratic is also computationally expensive (in the worst case, exponential in the number of variables m). Thus, we have adopted here a very simple compromise approach, in which only the first-order and the diagonal second-order terms are considered for exact bounding, and other terms are evaluated directly. That is,

$$B(p) = \sum_{i=1}^m \left[a_i (X_i - x_{i0})^2 + b_i (X_i - x_{i0}) \right] + Q, \quad (10)$$

where Q is the interval bound of all other terms, and is obtained by direct evaluation with interval

arithmetic. In Eq. (10), since X_i occurs twice, there exists a dependency problem. However, we can rearrange Eq. (10) such that each X_i occurs only once; that is,

$$B(p) = \sum_{i=1}^m \left[a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right] + Q. \quad (11)$$

In this way, the dependence problem in bounding the interval polynomial is alleviated so that a sharper bound can be obtained. Since we prefer not to divide by a very small number, Eq. (11) will be used only if $|a_i| \geq \omega$, where ω is a very small positive number. If $|a_i| < \omega$, direct evaluation with Eq. (10) will be used instead.

Validating Solver for Parametric ODEs

When a traditional sequential approach is applied to the optimization of nonlinear dynamic systems, the objective function ϕ is evaluated, for a given value of θ , by applying an ODE solver to the constraints to eliminate the state variables \mathbf{x} . In the global optimization algorithm described here, we will use a sequential approach based on interval analysis. This approach requires the evaluation of bounds on ϕ , given some parameter interval Θ . Thus, we need an ODE solver that can compute bounds on $\mathbf{x}_\mu, \mu = 0, 1, \dots, r$, for the case in which the parameters are interval valued. Interval methods (also called validated methods or verified methods) for ODEs,³⁰ provide a natural approach for computing the desired enclosure of the state variables at $t_\mu, \mu = 0, 1, \dots, r$. An excellent review of interval methods for IVPs has been given by Nedialkov et al.³¹ Much work has been done for the case in which the initial values are given by intervals, and there are several available software packages that deal with this case. However, relatively little work has been done on the case in which parameters are given by intervals. In our method for deterministic global optimization of dynamic systems, we will use a new validated solver for parametric ODEs,¹⁷ called VSPODE (Validating Solver for Parametric ODEs), which is used to produce guaranteed bounds on

the solutions of dynamic systems with interval-valued initial states and parameters. In this section, we review the key ideas behind the new method used in VSPODE, and outline the procedures used. Additional details are given by Lin and Stadtherr.¹⁷

Traditional interval methods usually consist of two processes applied at each integration step.³¹ In the first process, existence and uniqueness of the solution are proved using the Picard-Lindelöf operator and the Banach fixed point theorem,³² and a rough enclosure of the solution is computed. In the second process, a tighter enclosure of the solution is computed. In general, both processes are realized by applying interval Taylor series (ITS) expansions with respect to time, and using automatic differentiation to obtain the Taylor coefficients. A major difficulty in interval methods is the overestimation of bounds caused by the dependency problem of interval arithmetic and by the wrapping effect.³⁰ The accumulation of overestimations at successive time steps may ultimately lead to an explosion of enclosure sizes, causing the integration procedure to abort. Several schemes for reducing the overestimation of bounds have been proposed for the case of interval-valued initial states. For example, Lohner’s AWA package employs a QR-factorization method which features efficient coordinate transformations to tackle the wrapping effect.³³ Nedialkov’s VNODE package employs QR together with an interval Hermite-Obreschkoff method,^{34,35} which can be viewed as a type of generalized Taylor method, and improves on AWA. Janssen et al.³⁶ have introduced a constraint satisfaction approach to these problems, which enhances traditional interval methods with a pruning step based on a global relaxation of the ODEs. Another approach for addressing the dependency problem and the wrapping effect has been described by Berz and Makino³⁷ and implemented in the beam dynamics package COSY INFINITY. This scheme is based on expressing the dependence on initial values and time using a Taylor model. Neher et al.³⁸ have recently described this Taylor model approach in some detail and compared it to traditional interval methods. Taylor models are also used in VSPODE, though they are determined and used in a different way,

and a new type of Taylor model, involving a parallelepiped remainder bound, is introduced.¹⁷

Consider the parametric ODE system occurring in the constraints of the optimization problem:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}), \quad \mathbf{x}_0 \in \mathbf{X}_0, \quad \boldsymbol{\theta} \in \Theta, \quad (12)$$

where $t \in [t_0, t_r]$ for some $t_r > t_0$. The interval vectors $\mathbf{X}_0 = \mathbf{X}_0(\Theta)$ and Θ represent enclosures of initial values and parameters, respectively. It is desired to determine a validated enclosure of all possible solutions to this initial value problem. We denote by $\mathbf{x}(t; t_j, \mathbf{X}_j, \Theta)$ the set of solutions $\mathbf{x}(t; t_j, \mathbf{X}_j, \Theta) = \{\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \mid \mathbf{x}_j \in \mathbf{X}_j, \boldsymbol{\theta} \in \Theta\}$, where $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta})$ denotes a solution of $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ for the initial condition $\mathbf{x} = \mathbf{x}_j$ at t_j . We will outline a method for determining enclosures \mathbf{X}_j of the state variables at each time step $j = 1, \dots, r$, such that $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta) \subseteq \mathbf{X}_j$.

Assume that at t_j we have an enclosure \mathbf{X}_j of $\mathbf{x}(t_j; t_0, \mathbf{X}_0, \Theta)$, and that we want to carry out an integration step to compute the next enclosure \mathbf{X}_{j+1} . Then, in the first phase of the method, the goal is to find a step size $h_j = t_{j+1} - t_j > 0$ and an a priori enclosure (coarse enclosure) $\widetilde{\mathbf{X}}_j$ of the solution such that a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \in \widetilde{\mathbf{X}}_j$ is guaranteed to exist for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\boldsymbol{\theta} \in \Theta$. We apply the traditional interval method, with high order enclosure, to the parametric ODEs by using an interval Taylor series (ITS) with respect to time. That is, we determine h_j and \mathbf{X}_j such that for $\mathbf{X}_j \subseteq \widetilde{\mathbf{X}}_j^0$,

$$\widetilde{\mathbf{X}}_j = \sum_{i=0}^{k-1} [0, h_j]^i \mathbf{F}^{[i]}(\mathbf{X}_j, \Theta) + [0, h_j]^k \mathbf{F}^{[k]}(\widetilde{\mathbf{X}}_j^0, \Theta) \subseteq \widetilde{\mathbf{X}}_j^0. \quad (13)$$

Here $\widetilde{\mathbf{X}}_j^0$ is an initial estimate of $\widetilde{\mathbf{X}}_j$, k denotes the order of the Taylor expansion, and the coefficients $\mathbf{F}^{[i]}$ are interval extensions of the Taylor coefficients $\mathbf{f}^{[i]}$ of $\mathbf{x}(t)$ with respect to time, which can be

obtained recursively in terms of $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$ by

$$\begin{aligned} \mathbf{f}^{[0]} &= \mathbf{x} \\ \mathbf{f}^{[1]} &= \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) \\ \mathbf{f}^{[i]} &= \frac{1}{i} \left(\frac{\partial \mathbf{f}^{[i-1]}}{\partial \mathbf{x}} \mathbf{f} \right) (\mathbf{x}, \boldsymbol{\theta}), \quad i \geq 2. \end{aligned} \tag{14}$$

Satisfaction of Eq. (13) demonstrates that there exists a unique solution $\mathbf{x}(t; t_j, \mathbf{x}_j, \boldsymbol{\theta}) \in \widetilde{\mathbf{X}}_j$ for all $t \in [t_j, t_{j+1}]$, all $\mathbf{x}_j \in \mathbf{X}_j$, and all $\boldsymbol{\theta} \in \boldsymbol{\Theta}$.³⁹ $\widetilde{\mathbf{X}}_j^0$ is initialized and h_j is iteratively reduced, if needed to satisfy Eq. (13), using the method described by Nedialkov et al.³⁵

In the second phase of the method, we compute a tighter enclosure $\mathbf{X}_{j+1} \subseteq \widetilde{\mathbf{X}}_j$, such that $\mathbf{x}(t_{j+1}; t_0, \mathbf{X}_0, \boldsymbol{\Theta}) \subseteq \mathbf{X}_{j+1}$. This will be done by using an ITS approach to compute a Taylor model $\mathbf{T}_{\mathbf{x}_{j+1}}$ of \mathbf{x}_{j+1} in terms of the parameter vector $\boldsymbol{\theta}$, and then obtaining the enclosure $\mathbf{X}_{j+1} = B(\mathbf{T}_{\mathbf{x}_{j+1}})$ by bounding $\mathbf{T}_{\mathbf{x}_{j+1}}$ over $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. For the Taylor model computations, we begin by representing the parameters by the Taylor model $\mathbf{T}_{\boldsymbol{\theta}}$, with components

$$T_{\theta_i} = (m(\Theta_i) + (\theta_i - m(\Theta_i)), [0, 0]), \quad i = 1, \dots, p, \tag{15}$$

where $m(\Theta_i)$ indicates the midpoint of Θ_i . To determine enclosures of the interval Taylor series coefficients $\mathbf{f}^{[i]}(\mathbf{x}_j, \boldsymbol{\theta})$ a novel approach combining RDA operations with the mean value theorem is used to obtain the Taylor models $\mathbf{T}_{\mathbf{f}^{[i]}}$. Now using an interval Taylor series for \mathbf{x}_{j+1} with coefficients given by $\mathbf{T}_{\mathbf{f}^{[i]}}$, one can obtain a result for $\mathbf{T}_{\mathbf{x}_{j+1}}$ in terms of the parameters. In order to address the wrapping effect,³⁰ results are propagated from one time step to the next using a new type of Taylor model, in which the remainder bound is not an interval, but a parallelepiped. That is, the remainder bound is a set of the form $\mathcal{P} = \{Av \mid v \in V\}$, where $A \in \mathbb{R}^{n \times n}$ is a real and regular matrix. If A is orthogonal, as from a QR-factorization, then \mathcal{P} can be interpreted as a rotated n -dimensional rectangle. Complete details of the computation of $\mathbf{T}_{\mathbf{x}_{j+1}}$ are given by Lin and Stadtherr.¹⁷

The approach outlined above, as implemented in VSPODE, has been tested by Lin and Stadtherr,¹⁷ who compared its performance with results obtained using the popular VNODE package³⁵ (in using VNODE, interval-valued parameters are treated as additional state variables with interval-valued initial states). For the test problems used, VSPODE provided tighter enclosures on the state variables than VNODE, and required significantly less computation time.

Deterministic Global Optimization Method

In this section, we present a new method for the deterministic global optimization of dynamic systems. This is a generalization of the approach used by Lin and Stadtherr⁴⁰ for the special case of parameter estimation, in which the objective is a sum of squares function. As noted above, when a sequential approach is used, the state variables are effectively eliminated using the ODE constraints, in this case by employing VSPODE, leaving a bound-constrained minimization of $\phi(\boldsymbol{\theta})$ with respect to the adjustable parameters (decision variables) $\boldsymbol{\theta}$. The new approach can be thought of as a type of branch-and-bound method, with a constraint propagation procedure used for domain reduction. Therefore, it can also be viewed as a branch-and-reduce algorithm.¹⁶ The basic idea is that only those parts of the decision variable space Θ that satisfy the constraint $c(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta}) - \hat{\phi} \leq 0$, where $\hat{\phi}$ is a known upper bound on the global minimum, needs to be retained. We now describe a constraint propagation procedure, based on the use of Taylor models, that exploits this constraint information for domain reduction.

Constraint propagation on Taylor models

Partial information expressed by a constraint can be used to eliminate incompatible values from the domain of its variables. This domain reduction can then be propagated to all constraints on that variable, where it may be used to further reduce the domains of other variables. This process

is known as constraint propagation. In this subsection, we show how to apply such a constraint propagation procedure using Taylor models.

Let T_c be the Taylor model of the function $c(\mathbf{x})$ over the interval $\mathbf{x} \in \mathbf{X}$, and say the constraint $c(\mathbf{x}) \leq 0$ needs to be satisfied. In the constraint propagation procedure (CPP) described here, $B(T_c)$ is determined and then there are three possible outcomes: 1. If $\underline{B(T_c)} > 0$, then no $\mathbf{x} \in \mathbf{X}$ will ever satisfy the constraint; thus, the CPP can be stopped and \mathbf{X} discarded. 2. If $\overline{B(T_c)} \leq 0$, then every $\mathbf{x} \in \mathbf{X}$ will always satisfy the constraint; thus \mathbf{X} cannot be reduced and the CPP can be stopped. 3. If neither of previous two cases occur, then part of the interval \mathbf{X} may be eliminated; thus the CPP continues as described below, using an approach based on the range bounding strategy for Taylor models described above.

For some component i of \mathbf{x} , let a_i and b_i be the polynomial coefficients of the terms $(x_i - x_{i0})^2$ and $(x_i - x_{i0})$ of T_c , respectively. Note that $x_{i0} \in X_i$ and is usually the midpoint $x_{i0} = m(X_i)$; the value of x_{i0} will not change during the CPP. For $|a_i| \geq \omega$, the bounds on T_c can be expressed using Eq. (11) as

$$B(T_c) = a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} + S_i, \quad (16)$$

where

$$S_i = \sum_{\substack{j=1 \\ j \neq i}}^m \left[a_j \left(X_j - x_{j0} + \frac{b_j}{2a_j} \right)^2 - \frac{b_j^2}{4a_j} \right] + Q. \quad (17)$$

We can reduce the computational effort to obtain S_i by recognizing that this quantity is just $B(T_c)$ less the i -th term in the summation, and $B(T_c)$ was already computed earlier in the CPP. Thus, for each i , S_i can be determined by dependent subtraction (see above) using

$$S_i = B(T_c) \ominus \left[a_i \left(X_i - x_{i0} + \frac{b_i}{2a_i} \right)^2 - \frac{b_i^2}{4a_i} \right]. \quad (18)$$

Now define the intervals $U_i = X_i - x_{i0} + \frac{b_i}{2a_i}$ and $V_i = \frac{b_i^2}{4a_i} - S_i$, so that $B(T_c) = a_i U_i^2 - V_i$. The goal is to identify and retain only the part of X_i that contains values of x_i for which it is possible

to satisfy $c(\mathbf{x}) \leq 0$. In other words, the part of X_i that is going to be eliminated is guaranteed not to satisfy the constraint $c(\mathbf{x}) \leq 0$. Since $B(T_c) = a_i U_i^2 - V_i$ bounds the range of $c(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$, the part of X_i in which it is possible to satisfy $c(\mathbf{x}) \leq 0$ can be bounded by finding X_i such that all elements of $a_i U_i^2$ are less than or equal to at least one element of V_i . That is, we require that

$$\overline{a_i U_i^2} \leq \overline{V_i}. \quad (19)$$

Then, the set U_i that satisfies Eq. (19) can be determined to be

$$U_i = \begin{cases} \emptyset & \text{if } a_i > 0 \text{ and } \overline{V_i} < 0 \\ \left[-\sqrt{\frac{\overline{V_i}}{a_i}}, \sqrt{\frac{\overline{V_i}}{a_i}}\right] & \text{if } a_i > 0 \text{ and } \overline{V_i} \geq 0 \\ [-\infty, \infty] & \text{if } a_i < 0 \text{ and } \overline{V_i} \geq 0 \\ \left[-\infty, -\sqrt{\frac{\overline{V_i}}{a_i}}\right] \cup \left[\sqrt{\frac{\overline{V_i}}{a_i}}, \infty\right] & \text{if } a_i < 0 \text{ and } \overline{V_i} < 0. \end{cases} \quad (20)$$

The part of X_i to be retained is then $X_i = X_i \cap \left(U_i + x_{i0} - \frac{b_i}{2a_i}\right)$.

If $|a_i| < \omega$, then Eq. (11) should not be used (to avoid division by a very small number), but Eq. (10) can be used instead. Following a procedure similar to that used above, we now have $B(T_c) = b_i U_i - V_i$ with $U_i = X_i - x_{i0}$ and $V_i = -(B(T_c) \ominus b_i(X_i - x_{i0}))$. Note that all quadratic terms are now included in V_i . To identify bounds on the part of X_i in which is it possible to satisfy the constraint, we can now use the condition

$$\overline{b_i U_i} \leq \overline{V_i}. \quad (21)$$

Then, the set U_i that satisfies Eq. (21) can be determined to be

$$U_i = \begin{cases} \left[-\infty, \frac{\overline{V_i}}{b_i}\right] & \text{if } b_i > 0 \\ \left[\frac{\overline{V_i}}{b_i}, \infty\right] & \text{if } b_i < 0, \end{cases} \quad (22)$$

where it is assumed that $|b_i| \geq \omega$, to avoid dividing by a very small number. The part of X_i to be

retained is then $X_i = X_i \cap (U_i + x_{i0})$. If both $|a_i|$ and $|b_i|$ are less than ω , then no CPP will be applied on X_i .

The overall CPP is implemented by beginning with $i = 1$ and proceeding component by component. If, for any i , the result $X_i = \emptyset$ is obtained, then no $\mathbf{x} \in \mathbf{X}$ can satisfy the constraint; thus, \mathbf{X} can be discarded and the CPP stopped. Otherwise the CPP proceeds until all components of \mathbf{X} have been updated. Note that, in principle, each time an improved (smaller) X_i is found, it could be used in computing S_i for subsequent components of \mathbf{X} . However, this requires recomputing the bound $B(T_c)$, which, for the function $c(\mathbf{x})$ that is of interest here, is expensive. Thus, the CPP for each component is done using the bounds $B(T_c)$ computed from the original \mathbf{X} . If, after each component is processed, \mathbf{X} has been sufficiently reduced (by more than $\omega_1 = 10\%$ by volume), then a new bound $B(T_c)$ is obtained, now over the smaller \mathbf{X} , and a new CPP is started. Otherwise, the CPP terminates.

Global optimization algorithm

As with any type of procedure incorporating branch-and-bound, an important issue is how to initialize $\hat{\phi}$, the upper bound on the global minimum. There are many ways in which this can be done, and clearly, it is desirable to find a $\hat{\phi}$ that is as small as possible (i.e., the tightest possible upper bound). To initialize $\hat{\phi}$, we run p^2 local minimizations (p is the number of adjustable parameters) using a local optimization routine from randomly chosen starting points, and then choose the smallest value of ϕ found to be the initial $\hat{\phi}$. For this purpose, we use the bound-constrained quasi-Newton method L-BFGS-B⁴¹ as the local optimization routine, and DDASSL⁴² as the integration routine. Additional initialization steps are to set either a relative convergence tolerance ϵ^{rel} or an absolute convergence tolerance ϵ^{abs} , and initialize a work list \mathcal{L} . The work list (stack) \mathcal{L} will contain a sequence of subintervals (boxes) that need to be tested and initially

$\mathcal{L} = \{\Theta\}$, the entire parameter (decision variable) space.

The core steps in the iterative process involve the testing of boxes in the work list. This is an objective range test combined with domain reduction done using the CPP described above. Beginning with $k = 0$, at the k -th iteration a box is removed from the front of \mathcal{L} and is designated as the current subinterval $\Theta^{(k)}$. The Taylor model T_{ϕ_k} of the objective function ϕ over $\Theta^{(k)}$ is computed. To do this, Taylor models of \mathbf{x}_μ , the state variables at times $t_\mu, \mu = 1, \dots, r$, in terms of θ are determined using VSPODE, as described above. Note that T_{ϕ_k} then consists of a q -th order polynomial in the decision variables θ , plus a remainder bound. The part of $\Theta^{(k)}$ that can contain the global minimum must satisfy the constraint $c(\theta) = \phi(\theta) - \hat{\phi} \leq 0$. Thus the constraint propagation procedure (CPP) described above is now applied using this constraint. Recall that there are three possible outcomes in the CPP:

1. Testing for the first possible outcome, $\underline{B}(T_c) > 0$, amounts to checking if the lower bound of T_{ϕ_k} , $\underline{B}(T_{\phi_k})$, is greater than $\hat{\phi}$. If so, then $\Theta^{(k)}$ can be discarded because it cannot contain the global minimum and need not be further tested.

2. Testing for the second possible outcome, $\overline{B}(T_c) \leq 0$, amounts to checking if the upper bound of T_{ϕ_k} , $\overline{B}(T_{\phi_k})$, is less than $\hat{\phi}$. If so, then all points in $\Theta^{(k)}$ satisfy the constraint and the CPP can be stopped since no reduction in $\Theta^{(k)}$ can be achieved. This also indicates, with certainty, that there is a point in $\Theta^{(k)}$ that can be used to update $\hat{\phi}$. Thus, if $\overline{B}(T_{\phi_k}) < \hat{\phi}$, a local optimization routine, starting at some point in $\Theta^{(k)}$, is used to find a local minimum, which then provides an updated (smaller) $\hat{\phi}$, that is, a better upper bound on the global minimum. In our implementation, the midpoint of $\Theta^{(k)}$ is used as the starting point for the local optimization. A new CPP is then started on $\Theta^{(k)}$ using the updated value of $\hat{\phi}$.

3. If neither of the previous two outcomes occurs, then the full CPP described above is applied to reduce $\Theta^{(k)}$. Note that if $\Theta^{(k)}$ is sufficiently reduced (by more than $\omega_1 = 10\%$ by volume) in

comparison to its volume at the beginning of CPP, then new bounds $B(T_{\phi_k})$ are obtained, now over the smaller $\Theta^{(k)}$, and a new CPP is started.

After the CPP terminates, a convergence test is performed. If $(\hat{\phi} - \underline{B(T_{\phi_k})})/|\hat{\phi}| \leq \epsilon^{\text{rel}}$, or $(\hat{\phi} - \underline{B(T_{\phi_k})}) \leq \epsilon^{\text{abs}}$, then $\Theta^{(k)}$ need not be further tested and can be discarded. Otherwise, we will check to what extent $\Theta^{(k)}$ has been reduced compared to its volume at the beginning of the objective range test. If the subinterval has been reduced by more than $\omega_2 = 70\%$ by volume, it will be added to the front of the sequence \mathcal{L} of boxes to be tested. Otherwise, it will be bisected, and the resulting two subintervals added to the front of \mathcal{L} . Various strategies can be used to select the component to be bisected. For the problems solved here, the component with the largest relative width was selected for bisection. The relative width of a parameter component $\Theta_i^{(k)}$ is defined as $(\overline{\Theta_i^{(k)}} - \underline{\Theta_i^{(k)}}) / \max\{|\overline{\Theta_i^{(k)}}|, |\underline{\Theta_i^{(k)}}|, 1\}$. The volume reduction targets ω_1 and ω_2 can be adjusted as needed to tune the algorithm; the default values given above were used in the computational studies described below. At the end of this testing process, k is incremented, a box is removed from the front of \mathcal{L} , and the testing process is begun again. At termination, \mathcal{L} will become empty, and $\hat{\phi}$ is the ϵ -global minimum.

The method described above is an ϵ -global algorithm. It is also possible to incorporate interval-Newton steps in the method, and to thus make it an exact algorithm. This requires the application of VSPODE on the first- and second-order sensitivity equations. An exact algorithm using interval-Newton steps has been implemented by Lin and Stadtherr⁴⁰ for the special case of parameter estimation problems, but has not yet been fully implemented for the more general case described here.

Computational Studies

In this section, three example problems are presented to illustrate the theoretical and computational aspects of the proposed approach. All example problems were solved on an Intel Pentium 4 3.2GHz machine running Red Hat Linux. The VSPODE package,¹⁷ with a $k = 17$ order interval Taylor series, $q = 3$ order Taylor model, and QR approach for wrapping, was used to integrate the dynamic systems in each problem. Using a smaller ITS order k will result in the need for smaller step sizes in the integration and so will tend to increase computation time. Using a larger Taylor model order q will result in somewhat tighter bounds on the states, though at the expense of additional complexity in the Taylor model computations. The algorithm was implemented in C++.

Illustrative example

This problem has been used by several previous authors as an illustrative example. It involves the optimization of a simple dynamic system, with one decision variable. The problem is formulated as:

$$\begin{aligned} \min_{\theta} \quad & \phi = -x^2(t_f) & (23) \\ \text{s.t.} \quad & \dot{x} = -x^2 + \theta \\ & x_0 = 9 \\ & t \in [t_0, t_f] = [0, 1] \\ & \theta \in [-5, 5]. \end{aligned}$$

It has been shown that Problem (23) has two local minima, one at each bound of the parameter domain.^{11,12}

We will illustrate here the optimization procedure described above by solving this problem to

an absolute tolerance of $\epsilon^{\text{abs}} = 10^{-3}$. First, the upper bound $\widehat{\phi}$ on the global minimum is initialized using local optimization. This results in finding the local solution $\widehat{\theta} = -5$ with objective value $\widehat{\phi} = -8.23262154$. The work list is initialized to $\mathcal{L} = \{-5, 5\}$.

The main iteration is now begun with $k = 0$ (in this section k is an iteration counter and does not refer to the order of the interval Taylor series used), and $\Theta^{(0)} = [-5, 5]$ is removed from the work list. VSPODE is now applied and a Taylor model T_{ϕ_0} of the objective function over $\Theta^{(0)}$ is obtained, with the result

$$T_{\phi_0} = (-1.4347 - 0.6394(\theta - 0) - 0.0865(\theta - 0)^2 + 0.0167(\theta - 0)^3, [-9.212744, 9.212744]).$$

The bounds $B(T_{\phi_0}) = [-18.098954, 11.053372]$ are now computed and the CPP is performed. The CPP results in no reduction of the current interval; thus it is bisected and the resulting two subintervals $[-5, 0]$, and $[0, 5]$ are added to the work sequence, giving $\mathcal{L} = \{-5, 0\}, [0, 5\}$.

For the next iteration, $k = 1$ and $\Theta^{(1)} = [-5, 0]$ is removed from the front of \mathcal{L} . VSPODE is applied and a Taylor model T_{ϕ_1} of the objective function over $\Theta^{(1)}$ is obtained, resulting in

$$T_{\phi_1} = (-0.2969 + 0.3949(\theta + 2.5) - 0.4569(\theta + 2.5)^2 + 0.1223(\theta + 2.5)^3, [-2.205915, 2.205915]).$$

The bounds are computed as $B(T_{\phi_1}) = [-8.256285, 3.905142]$ and the CPP is applied. Applying the CPP reduces the current subinterval to $\Theta^{(1)} = [-5, -4.9911]$ and gives $\underline{B(T_{\phi_1})} = -8.256285$. This does not satisfy the convergence test, but the volume of $\Theta^{(1)}$ has been reduced by more than 70%. Thus, the reduced subinterval is added to the front of the work array, giving $\mathcal{L} = \{-5, 4.9911\}, [0, 5\}$.

Next, with $k = 2$ and $\Theta^{(2)} = [-5, -4.9911]$, the Taylor model of the objective computed using

VSPODE is

$$T_{\phi_2} = (-8.18836 + 9.9814(\theta + 4.99555) - 6.0120(\theta + 4.99555)^2 \\ + 2.7786(\theta + 4.99555)^3, [-0.000001, 0.000001]),$$

and $B(T_{\phi_2}) = [-8.232622, -8.144335]$. After the CPP, the subinterval has been reduced to $[-5, -4.9999]$ and $\underline{B(T_{\phi_2})} = -8.232622$. Since $\underline{B(T_{\phi_2})}$ satisfies the convergence condition, this subinterval needs no further testing and can be discarded. Now $\mathcal{L} = \{[0, 5]\}$.

Next, with $k = 3$ and $\Theta^{(3)} = \{[0, 5]\}$, the Taylor model of the objective function is

$$T_{\phi_3} = (-2.8176 - 0.8901(\theta - 2.5) - 0.0223(\theta - 2.5)^2 \\ + 0.0035(\theta - 2.5)^3, [-0.066276, 0.066276]),$$

and $B(T_{\phi_3}) = [-5.302934, -0.610588]$. Now in the CPP it is seen that $\underline{B(T_{\phi_3})} > \hat{\phi}$, so this subinterval can be discarded. Now $\mathcal{L} = \emptyset$, indicating termination. Thus the global minimum is $\phi^* = \hat{\phi} = -8.23262154$ at $\theta^* = \hat{\theta} = -5$. The total CPU time required was 0.05 seconds.

Singular control problem

This example is a nonlinear singular optimal control problem originally formulated by Luus⁴³ and also considered by Esposito and Floudas,⁶ Chachuat and Latifi¹¹ and Singer and Barton.¹³ This problem is known to have multiple local solutions. The problem to be solved is:

$$\begin{aligned} \min_{\theta(t)} \quad \phi &= \int_{t_0}^{t_f} [x_1^2 + x_2^2 + 0.0005(x_2 + 16t - 8 - 0.1x_3\theta^2)^2] dt & (24) \\ \text{s.t.} \quad \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_3\theta + 16t - 8 \\ \dot{x}_3 &= \theta \\ \mathbf{x}_0 &= (0, -1, -\sqrt{5})^T \\ t &\in [t_0, t_f] = [0, 1] \\ \theta &\in [-4, 10]. \end{aligned}$$

In order to express the ODE constraints in the autonomous form of Problem (1), we introduce an additional state variable x_4 representing t , and an additional equation, $\dot{x}_4 = 1$. We also introduce a quadrature variable x_5 . The reformulated singular control problem is then given by:

$$\begin{aligned}
\min_{\theta(t)} \quad & \phi = x_5(t_f) & (25) \\
s.t. \quad & \dot{x}_1 = x_2 \\
& \dot{x}_2 = -x_3\theta + 16x_4 - 8 \\
& \dot{x}_3 = \theta \\
& \dot{x}_4 = 1 \\
& \dot{x}_5 = x_1^2 + x_2^2 + 0.0005(x_2 + 16x_4 - 8 - 0.1x_3\theta^2)^2 \\
& \mathbf{x}_0 = (0, -1, -\sqrt{5}, 0, 0)^T \\
& t \in [t_0, t_f] = [0, 1] \\
& \theta \in [-4, 10].
\end{aligned}$$

The control $\theta(t)$ was parameterized as a piecewise constant profile with a specified number of equal time intervals. Five problems are considered, corresponding to one, two, three, four and five time intervals in the parameterization. For example, for the two-interval case, there are two decision variables, θ_1 and θ_2 , corresponding to the constant values of $\theta(t)$ over the first and second halves of the overall time interval of interest. Each problem was solved to an absolute tolerance of $\epsilon^{\text{abs}} = 10^{-3}$. The results are presented in Table 1. This shows, for each problem, the globally optimal objective value ϕ^* and the corresponding optimal controls $\boldsymbol{\theta}^*$, as well as the CPU time (in seconds) and number of iterations required.

Comparisons with computation times reported for other methods can give only a very rough idea of the relative efficiency of the methods, due to differences in implementation and in the machine used for the computation. Chachuat and Latifi¹¹ solved the two-interval problem to ϵ -

global optimality using four different strategies, with the most efficient requiring 502 CPU seconds, using an unspecified machine and a “prototype” implementation. Singer and Barton¹³ solved the one-, two- and three-interval cases with $\epsilon^{\text{abs}} = 10^{-3}$ using two different problem formulations (with and without a quadrature variable) and two different implementations (with and without branch-and-bound heuristics). Best results in terms of efficiency were achieved with heuristics and without a quadrature variable, with CPU times of 1.8, 22.5 and 540.3 seconds (1.667 GHz AMD Athlon XP2000+) on the one-, two- and three-interval problems, respectively. This compares to CPU times of 0.02, 0.32 and 10.88 seconds (3.2 GHz Intel Pentium 4) for the method given here. Even accounting for the roughly factor of two difference in the speeds of the machines used, the method described here appears to be well over an order of magnitude faster. The four- and five-interval problems were solved here in 369 and 8580.6 CPU seconds, respectively, and apparently have not been solved previously using a method rigorously guaranteed to find an ϵ -global minimum. It should be noted that our solution to the three-interval problem, as given in Table 1, differs from the result reported by Singer and Barton,¹³ which is known to be a misprint.⁴⁴

Oil shale pyrolysis problem

This example is a fixed final time formulation of the oil shale pyrolysis problem originally formulated by Luus⁴³ and also considered by Esposito and Floudas⁶ and Singer and Barton.¹³ The

problem formulation is:

$$\begin{aligned}
\min_{\theta(t)} \quad & \phi = -x_2(t_f) & (26) \\
s.t. \quad & \dot{x}_1 = -k_1x_1 - (k_3 + k_4 + k_5)x_1x_2 \\
& \dot{x}_2 = k_1x_1 - k_2x_2 + k_3x_1x_2 \\
& k_i = a_i \exp\left(\frac{-b_i/R}{\theta}\right), i = 1, \dots, 5 \\
& \mathbf{x}_0 = (1, 0)^T \\
& t \in [t_0, t_f] = [0, 10] \\
& \theta \in [698.15, 748.15].
\end{aligned}$$

The values for a_i and b_i/R are defined by Floudas et al.,⁴⁵ and shown in Table 2. Singer and Barton¹³ indicate $t_f = 1$ in their statement of the problem, but give results for the case $t_f = 10$, as specified above.

In Problem (26), the reciprocal operation on the control variable is required to calculate the k_i , which imposes a significant overhead on the related Taylor model computations. Thus, for the control variable we use the transformation

$$\bar{\theta} = \frac{698.15}{\theta}. \tag{27}$$

The transformed problem then becomes:

$$\begin{aligned}
\min_{\bar{\theta}(t)} \quad & \phi = -x_2(t_f) & (28) \\
s.t. \quad & \dot{x}_1 = -k_1x_1 - (k_3 + k_4 + k_5)x_1x_2 \\
& \dot{x}_2 = k_1x_1 - k_2x_2 + k_3x_1x_2 \\
& k_i = a_i \exp(-\bar{\theta}b_i/R), i = 1, \dots, 5 \\
& \mathbf{x}_0 = (1, 0)^T \\
& t \in [t_0, t_f] = [0, 10] \\
& \bar{\theta} \in [698.15/748.15, 1].
\end{aligned}$$

As in the previous example, the control $\theta(t)$ was parameterized as a piecewise constant profile with a specified number of equal time intervals. Four problems are considered, corresponding to one, two, three and four time intervals in the parameterization. Each problem was solved to an absolute tolerance of $\epsilon^{\text{abs}} = 10^{-3}$. The results are presented in Table 3.

Singer and Barton¹³ solved the one- and two-interval cases with $\epsilon^{\text{abs}} = 10^{-3}$ using two different implementations (with and without branch-and-bound heuristics). Best results in terms of efficiency were achieved using the heuristics, with CPU times of 26.2 and 1597.3 seconds (1.667 GHz AMD Athlon XP2000+) on the one- and two-interval problems, respectively. This compares to CPU times of 3.2 and 26.8 seconds (3.2 GHz Intel Pentium 4) for the method given here. As in the previous problem, even after accounting for the roughly factor of two difference in the speeds of the machines used, the method described here appears to be significantly more efficient, by well over a order of magnitude in the two-interval case. The three- and four-interval problems were solved here in 251.6 and 2443.5 CPU seconds, respectively, and apparently have not been solved previously using a rigorously guaranteed method. The worst-case exponential complexity seen in these results, as well as those in the previous example, reflects the fact that global optimization for

nonlinear problems is in general an NP-hard problem.

Concluding Remarks

We have presented here a new approach for the deterministic global optimization of dynamic systems, including optimal control problems. This method is based on interval analysis and Taylor models and employs a type of sequential approach. Instead of the usual branch-and-bound approach, we incorporate a new domain reduction technique, and thus use a type of branch-and-reduce strategy. A key feature of the method is the use of a new validated solver¹⁷ for parametric ODEs, which is used to produce guaranteed bounds on the solutions of dynamic systems with interval-valued parameters. The result is that an ϵ -global optimum can be found with both mathematical and computational certainty. Computational studies on benchmark problems have been done showing that this new approach provides significant improvements in computational efficiency, well over an order of magnitude in most cases, relative to other recently described methods.

Acknowledgments

This work was supported in part by the State of Indiana 21st Century Research and Technology Fund under Grant #909010455, and by the Department of Energy under Grant DE-FG02-05CH11294.

References

1. Neuman C, Sen A. A suboptimal control algorithm for constraint problems using cubic splines. *Automatica*. 1973;9:601–613.
2. Tsang TH, Himmelblau DM, Edgar TF. Optimal control via collocation and nonlinear programming. *Int J Control*. 1975;21:763–768.
3. Brusch R, Schappelle R. Solution of highly constrained optimal control problems using nonlinear programming. *AIAA J*. 1973;11:135–136.
4. Teo K, Goh G, Wong K. *A Unified Computational Approach to Optimal Control Problems*, volume 55 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. New York, NY: Wiley, 1991.
5. Luus R, Cormack DE. Multiplicity of solutions resulting from the use of variational methods in optimal control problems. *Can J Chem Eng*. 1972;50:309–311.
6. Esposito WR, Floudas CA. Deterministic global optimization in nonlinear optimal control problems. *J Global Optim*. 2000;17:97–126.
7. Esposito WR, Floudas CA. Global optimization for the parameter estimation of differential-algebraic systems. *Ind Eng Chem Res*. 2000;39:1291–1310.
8. Adjiman CS, Androulakis IP, Floudas CA, Neumaier A. A global optimization method, α BB, for general twice-differentiable NLPs–I. Theoretical advances. *Comput Chem Eng*. 1998;22:1137–1158.
9. Adjiman CS, Dallwig S, Floudas CA, Neumaier A. A global optimization method, α BB, for general twice-differentiable NLPs–II. Implementation and computational results. *Comput Chem Eng*. 1998;22:1159–1179.

10. Papamichail I, Adjiman CS. A rigorous global optimization algorithm for problems with ordinary differential equations. *J Global Optim.* 2002;24:1–33.
11. Chachuat B, Latifi MA. A new approach in deterministic global optimisation of problems with ordinary differential equations. In: Floudas CA, Pardalos PM, eds., *Frontiers in Global Optimization*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 2004; 83–108.
12. Papamichail I, Adjiman CS. Global optimization of dynamic systems. *Comput Chem Eng.* 2004;28:403–415.
13. Singer AB, Barton PI. Global optimization with nonlinear ordinary differential equations. *J Global Optim.* 2006;34:159–190.
14. Singer AB, Barton PI. Bounding the solutions of parameter dependent nonlinear ordinary differential equations. *SIAM J Sci Comput.* 2006;27:2167–2182.
15. Singer AB. *Global Dynamic Optimization*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 2004.
16. Sahinidis, NV, Bliet C, Jermann C, Neumaier A. Global optimization and constraint satisfaction: The branch-and-reduce approach. *Lect Notes Comput Sci.* 2003;2861:1–16.
17. Lin Y, Stadtherr MA. Validated solutions of initial value problems for parametric ODEs. *Appl Numer Math.* 2006; in press.
18. Rauh A, Minisini J, Hofer EP. Interval techniques for design of optimal and robust control strategies. Presented at: *12th GAMM-IMACS International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics (SCAN 2006)*, Duisburg, Germany, September 2006.

19. Hansen E, Walster GW. *Global Optimization Using Interval Analysis*. New York, NY: Marcel Dekker, 2004.
20. Jaulin L, Kieffer M, Didrit O, Walter É. *Applied Interval Analysis*. London, UK: Springer-Verlag, 2001.
21. Kearfott RB. *Rigorous Global Search: Continuous Problems*. Dordrecht, The Netherlands: Kluwer Academic Publishers, 1996.
22. Neumaier A. *Interval Methods for Systems of Equations*. Cambridge, UK: Cambridge University Press, 1990.
23. Makino K, Berz M. Remainder differential algebras and their applications. In: Berz M, Bishof C, Corliss G, Griewank A, eds., *Computational Differentiation: Techniques, Applications, and Tools*. Philadelphia, PA: SIAM, 1996; 63–74.
24. Makino K, Berz M. Efficient control of the dependency problem based on Taylor model methods. *Reliab Comput*. 1999;5:3–12.
25. Makino K, Berz M. Taylor models and other validated functional inclusion methods. *Int J Pure Appl Math*. 2003;4:379–456.
26. Revol N, Makino K, Berz M. Taylor models and floating-point arithmetic: Proof that arithmetic operations are validated in COSY. *J Logic Algebr Progr*. 2005;64:135–154.
27. Makino K. *Rigorous Analysis of Nonlinear Motion in Particle Accelerators*. Ph.D. thesis, Michigan State University, East Lansing, Michigan, USA, 1998.
28. Neumaier A. Taylor forms - Use and limits. *Reliab Comput*. 2003;9:43–79.
29. Makino K, Berz M. Taylor model range bounding schemes. Presented at: *Third International Workshop on Taylor Methods*. Miami Beach, FL, December 2004.

30. Moore RE. *Interval Analysis*. Englewood Cliffs, NJ: Prentice-Hall, 1966.
31. Nedialkov NS, Jackson KR, Corliss GF. Validated solutions of initial value problems for ordinary differential equations. *Appl Math Comput*. 1999;105:21–68.
32. Eijgenraam P. The solution of initial value problems using interval arithmetic. Mathematical Centre Tracts No. 144, Stichting Mathematisch Centrum, Amsterdam, The Netherlands, 1981.
33. Lohner RJ. Enclosing the solutions of ordinary initial and boundary value problems. In: Kaucher E, Kulisch U, Ullrich C, eds., *Computer Arithmetic: Scientific Computation and Programming Languages*. Stuttgart, Germany: Teubner, 1987; 255–286.
34. Nedialkov NS, Jackson KR. Some recent advances in validated methods for IVPs for ODEs. *Appl Numer Math*. 2002;42:269–284.
35. Nedialkov NS, Jackson KR, Pryce JD. An effective high-order interval method for validating existence and uniqueness of the solution of an IVP for an ODE. *Reliab Comput*. 2001;7:449–465.
36. Janssen M, Hentenryck PV, Deville Y. A constraint satisfaction approach for enclosing solutions to parametric ordinary differential equations. *SIAM J Numer Anal*. 2002;40:1896–1939.
37. Berz M, Makino K. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliab Comput*. 1998;4:361–369.
38. Neher M, Jackson KR, Nedialkov NS. On Taylor model based integration of ODEs. Technical Report, Department of Computer Science, University of Toronto, Toronto, Canada, 2005.
39. Corliss GF, Rihm R. Validating an a priori enclosure using high-order Taylor series. In: Alefeld G, Frommer A, eds., *Scientific Computing and Validated Numerics: Proceedings of the International Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics (SCAN'95)*. Berlin, Germany: Akademie Verlag, 1996; 228–238.

40. Lin Y, Stadtherr MA. Deterministic global optimization for parameter estimation of dynamic systems. *Ind Eng Chem Res.* 2006;45:8438–8448.
41. Byrd RH, Lu P, Nocedal J, Zhu C. A limited memory algorithm for bound constrained optimization. *SIAM J Sci Comput.* 1995;16:1190–1208.
42. Brenan KE, Campbell SL, Petzold LR. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations.* Philadelphia, PA: SIAM, 1996.
43. Luus R. Optimal control by dynamic programming using systematic reduction in grid size. *Int J Control.* 1990;5:995–1013.
44. Singer AB. Personal communication, 2006.
45. Floudas CA, Pardalos PM, Adjiman CS, Esposito WR, Gumus ZH, Harding ST, Klepeis JL, Meyer CA, Schweiger CA. *Handbook of Test Problems in Local and Global Optimization.* Dordrecht, The Netherlands: Kluwer Academic Publishers, 1999.

Table 1: Results for the singular control problem.

| # of time intervals | ϕ^* | θ^* | CPU (s) | Iterations |
|---------------------|----------|--------------------------------------|---------|------------|
| 1 | 0.4965 | (4.071) | 0.02 | 9 |
| 2 | 0.2771 | (5.575, -4.000) | 0.32 | 71 |
| 3 | 0.1475 | (8.001, -1.944, 6.042) | 10.88 | 1414 |
| 4 | 0.1237 | (9.789, -1.200, 1.257, 6.256) | 369.0 | 31073 |
| 5 | 0.1236 | (10.00, 1.494, -0.814, 3.354, 6.151) | 8580.6 | 493912 |

Table 2: Data for the oil shale pyrolysis problem.

| i | $\ln a_i$ | b_i/R |
|-----|-----------|---------|
| 1 | 8.86 | 10215.4 |
| 2 | 24.25 | 18820.5 |
| 3 | 23.67 | 17008.9 |
| 4 | 18.75 | 14190.8 |
| 5 | 20.70 | 15599.8 |

Table 3: Results for the oil shale pyrolysis problem.

| # of time intervals | ϕ^* | $\bar{\theta}^*$ | CPU (s) | Iterations |
|---------------------|----------|-------------------------------|---------|------------|
| 1 | -0.3479 | (0.984) | 3.2 | 21 |
| 2 | -0.3510 | (0.970, 1.000) | 26.8 | 178 |
| 3 | -0.3517 | (1.000, 0.963, 1.000) | 251.6 | 1531 |
| 4 | -0.3523 | (1.000, 0.9545, 1.000, 1.000) | 2443.5 | 12874 |