Verified Simulation of ODEs and DAEs in ValEncIA-IVP*

Andreas Rauh
Chair of Mechatronics, University of Rostock, D-18059
Rostock, Germany
Andreas.Rauh@uni-rostock.de

Ekaterina Auer
Faculty of Engineering, INKO, University of Duisburg-Essen, D-47048 Duisburg, Germany
Auer@inf.uni-due.de

Abstract

ValEncIA-IVP is a verified solver for initial value problems for sets of ordinary differential equations which determines guaranteed enclosures of all reachable states. In this paper, we present its new features which allow for a wider application domain. They also improve the performance of ValEncIA-IVP. Especially for the simulation of asymptotically stable systems, we present a new exponential enclosure technique which describes contracting state enclosures. Besides, computation of differential sensitivities as well as verified solution approaches for sets of differential-algebraic equations are introduced.

Keywords: ordinary differential equations, differential-algebraic equations, initial value problems, inverse control problems, ValEncIA-IVP
AMS subject classifications: 65L05, 65L09, 34A55, 34H05, 49J15

1 Introduction

ValEncIA-IVP relies on a simple iteration scheme which can be derived using Banach's fixed-point theorem. We describe state enclosures of initial value problems (IVPs) for sets of ordinary differential equations (ODEs) using interval boxes. Therefore, like any other developers of interval software, we need techniques to handle typical sources of overestimation: the wrapping effect and multiple dependencies of the iteration formula on common interval variables [7]. To prevent diameters of the interval enclosures from growing for asymptotically stable systems, a new exponential enclosure technique is derived in this paper. This approach allows us to describe solution sets which are contracting over time. This technique is combined with further methods to detect and eliminate regions in the state space resulting from overestimation.

*Submitted: January 19, 2009; Revised: February 9, 2010; Accepted: March 1, 2010.

These methods are based on consistency tests using backward integration of subintervals and physically motivated dynamical constraints. In addition, we extend the use of a basic preconditioning strategy to nonlinear problems. This algorithm is based on the transformation of linear systems into real Jordan normal form (Sections 2–3).

ValEncIA-IVP can exploit specific system properties to automatically readjust state enclosures if guaranteed a-priori bounds for them are known (e.g. if state variables are known to be non-negative). Additionally, a verified test for cooperativity is available to tighten the set of solutions (Section 5).

For ODEs, routines for verified sensitivity analysis are available (Section 4). In ValEncIA-IVP, the sensitivities are defined as partial derivatives of the trajectories of all system states with respect to (uncertain) parameters. The corresponding sensitivity equations are given by sets of ODEs which are derived automatically using algorithmic differentiation provided by FADBAD++.

In addition to ODEs, ValEncIA-IVP can be applied to sets of differential-algebraic equations (DAEs) (Section 6). An important application area of verified DAE solvers is reachability analysis without symbolic elimination of algebraic constraints. Further, so-called inverse control problems can be handled. In this case, time-dependent algebraic equations in sets of DAEs serve as constraints which specify the desired output signals of a dynamical system [4]. The unknown quantities are the corresponding inputs and the enclosures of the trajectories of the state variables. For the solution of IVPs for DAEs, verified solvers for nonlinear algebraic equations were integrated into the core of ValEncIA-IVP. For simulation and control synthesis, guaranteed enclosures of consistent initial conditions are computed automatically in a first stage.

2 Verified Simulation of Sets of ODEs

ValEncIA-IVP is a verified solver for the computation of guaranteed enclosures of the solution to IVPs for ODEs

$$\dot{x}\left(t\right)=f\left(x\left(t\right),t\right) \tag{1}$$

with the initial conditions

$$\left[x^{0}\right]:=\left[\underline{x}\left(0\right)\ ;\ \overline{x}\left(0\right)\right]\quad . \tag{2}$$

To apply the algorithms implemented in ValEncIA-IVP, we consider state equations with $f : D \mapsto \mathbb{R}^{n_x}$, $D \subset \mathbb{R}^{n_x} \times \mathbb{R}^1$ open, $f \in C^1(D, \mathbb{R}^{n_x})$. The state vector $x\left(t\right)$ may also contain time-varying system parameters $p\left(t\right)$ with a-priori bounds $\left[p\left(t\right)\right] := \left[\underline{p}\left(t\right)\ ;\ \overline{p}\left(t\right)\right]$ and $\dot{p}\left(t\right) = \Delta p\left(t\right)$, $\Delta p\left(t\right) \in \left[\Delta p\left(t\right)\right] := \left[\underline{\Delta p}\left(t\right)\ ;\ \overline{\Delta p}\left(t\right)\right]$ for their variation rates.

The goal of ValEncIA-IVP is to compute guaranteed interval enclosures of all states which can be reached in a given finite time horizon under consideration of all uncertainties in the initial states and system parameters. As already mentioned, a-priori bounds for state variables or time-varying system parameters can be taken into account directly in ValEncIA-IVP. For that purpose, state enclosures computed with the help of the following Algorithms 1 and 2 are intersected with the a-priori bounds at the end of each iteration step. A proof of these two algorithms and their detailed derivation can be found in [2, 8]. Both algorithms are implemented in C++ using Profil/BIAS [5] for interval arithmetic and FADBAD++ [3] for algorithmic differentiation.

The basis of ValEncIA-IVP is the assumption that time-varying state enclosures can be expressed by

$$[x_{encl}(t)] := x_{app}(t) + [R(t)] \tag{3}$$

with a non-verified approximate solution $x_{app}(t)$ and guaranteed error bounds $[R(t)]$ for all $t \in [0\ ;\ t_f]$. The error bounds $[R(t)]$ are computed by the following two-stage procedure:

Algorithm 1.

1. Compute an interval enclosure of all possible time derivatives $\left[\dot{R}(t)\right]$ of the error term by

$$\begin{aligned}\left[\dot{R}^{(\kappa+1)}(t)\right] &= -\dot{x}_{app}(t) + f\left(\left[x^{(\kappa)}_{encl}(t)\right], t\right)\\ &= -\dot{x}_{app}(t) + f\left(x_{app}(t) + \left[R^{(\kappa)}(t)\right], t\right)\\ &=: r\left(\left[R^{(\kappa)}(t)\right], t\right) \quad .\end{aligned} \tag{4}$$

This iteration converges to a verified enclosure of $\left[\dot{R}(t)\right]$ if $\left[\dot{R}^{(\kappa+1)}(t)\right] \subseteq \left[\dot{R}^{(\kappa)}(t)\right]$ holds. The iteration (4) is continued until $\left[\dot{R}^{(\kappa+1)}(t)\right] \approx \left[\dot{R}^{(\kappa)}(t)\right]$.

2. Integrate $\left[\dot{R}^{(\kappa+1)}(t)\right]$, $0 \leq t \leq T$, $T \in [0\ ;\ t_f]$, with respect to time in a verified way according to

$$\begin{aligned}\left[R^{(\kappa+1)}(t)\right] &\subseteq \left[R^{(\kappa+1)}(0)\right] + \int_0^t \left[\dot{R}^{(\kappa+1)}(\tau)\right] d\tau\\ &= \left[R^{(\kappa+1)}(0)\right] + \int_0^t r\left(\left[R^{(\kappa)}(\tau)\right], \tau\right) d\tau \quad ,\end{aligned} \tag{5}$$

where the last integral is replaced by its guaranteed bound

$$\left[R^{(\kappa+1)}(t)\right] \subseteq \left[R^{(\kappa+1)}(0)\right] + t \cdot r\left(\left[R^{(\kappa)}([0\ ;\ t])\right], [0\ ;\ t]\right) \quad . \tag{6}$$

These updated error bounds are required for the evaluation of the formula (4) in the next iteration step. Uncertainties of the initial conditions are accounted for by choosing $[R(0)]$ such that $[x_0] \subseteq x_{app}(0) + [R(0)]$ is fulfilled.

To prevent overestimation for asymptotically stable systems, the following exponential enclosure approach is introduced.

Algorithm 2.

1. Compute guaranteed state enclosure $[x_{encl}]$ using Algorithm 1.

2. Check the condition $0 \notin [x_{encl,i}([0\ ;\ T])]$ for all $i = 1, \ldots, n_x$. If this property if fulfilled, the exponential enclosure approach is applicable.

3. Compute an interval enclosure of all reachable states at a point of time t described by the exponential term

$$[x_{encl}(t)] := e^{[\Lambda]\cdot t} \cdot [x_{encl}(0)] \tag{7}$$

with the diagonal matrix $[\Lambda] = \mathrm{diag}[\lambda_i]$, $i = 1, \dots, n_x$. This enclosure can be obtained if the coefficients $[\lambda_i]$ are determined using the iteration formula

$$\left[\lambda_i^{(\kappa+1)}\right] := \frac{f_i\left(e^{\left(\left[\Lambda^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl}\,(0)]\,,[0\ ;\ T]\right)}{e^{\left(\left[\lambda_i^{(\kappa)}\right]\cdot[0\ ;\ T]\right)}\cdot[x_{encl,i}\,(0)]} \quad \text{with} \quad T \in [0\ ;\ t_f] \ . \tag{8}$$

3 Application: Catalytic Reactor

Consider the catalytic reaction scheme

$$A_1 \quad \underset{k_2}{\overset{k_1}{\rightleftarrows}} \quad A_2 \quad \xrightarrow{k_3} \quad A_3 \tag{9}$$

with three substances A_1, A_2, A_3 and the corresponding normalized concentrations x_1, x_2, x_3, $x := \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$. The reaction is described by the ODEs

$$\dot{x}(t) = \begin{bmatrix} -u(t)\cdot k_1 & u(t)\cdot k_2 \\ u(t)\cdot k_1 & -k_3 - u(t)\cdot(k_2 - k_3) \end{bmatrix}\cdot x(t) \quad \text{with} \quad x\,(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{10}$$

and the catalyst concentration $u\,(t)$ controlling the reaction between A_1 and A_2 as well as the concentration $1 - u\,(t)$ for the reaction between A_2 and A_3.

Since the total concentration is constant in this example, the algebraic equation $x_3\,(t) = 1 - x_1\,(t) - x_2\,(t)$ holds. The nominal system parameters (corresponding to the reaction rates) are given by $k_1 = 1$, $k_2 = 10$, and $k_3 = 1$. The goal of a verified simulation is to compute guaranteed state enclosures for $x_1\,(t)$ and $x_2\,(t)$ for $t \geq 0$.

If the standard iteration formula of ValEncIA-IVP (Algorithm 1) is applied without any preconditioning of the state equations, the state enclosures are diverging rapidly. We show two possibilities to prevent this behavior.

First, dynamical constraints can be introduced in such a way as to minimize the dependency on interval variables. For this example, the new state vector

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ k_3 + u\,(0)\cdot(k_2 - k_3) & u\,(0)\cdot k_2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} =: \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{11}$$

is introduced for which the corresponding state-space representation

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix}\begin{bmatrix} -u(t)\cdot k_1 & u(t)\cdot k_2 \\ u(t)\cdot k_1 & -k_3 - u(t)\cdot(k_2 - k_3) \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{12}$$

holds with the initial states

$$z\,(0) = \begin{bmatrix} 1 \\ k_3 + u\,(0)\cdot(k_2 - k_3) \end{bmatrix} \quad \text{for} \quad x\,(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \ . \tag{13}$$

Subintervals $[\tilde{x}\,(t)]$ of the enclosure $[x\,(t)] := \begin{bmatrix} [x_1\,(t)] & [x_2\,(t)] \end{bmatrix}^T$ with

$$[z_i\,(t)] \cap w_i^T \cdot [\tilde{x}\,(t)] = \emptyset\ , \quad i \in \{1, 2\}\ , \tag{14}$$

are inconsistent and can be eliminated. In (14), the enclosure $[z\,(t)]$ is obtained by verified integration of (12). Since only the algebraic expression (11) needs to be evaluated for the subintervals of $[x\,(t)]$, this consistency test is performed before the consistency

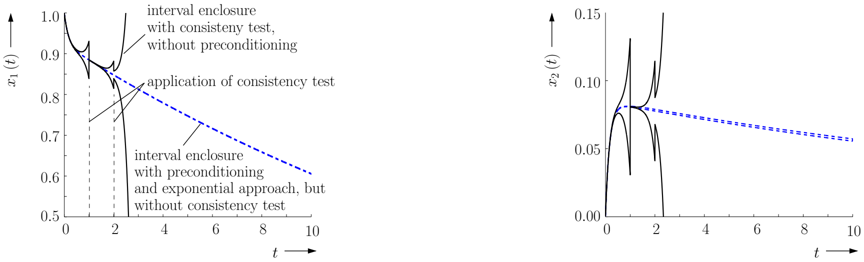


Figure 1: Comparison of different simulation methods for the catalytic reactor.

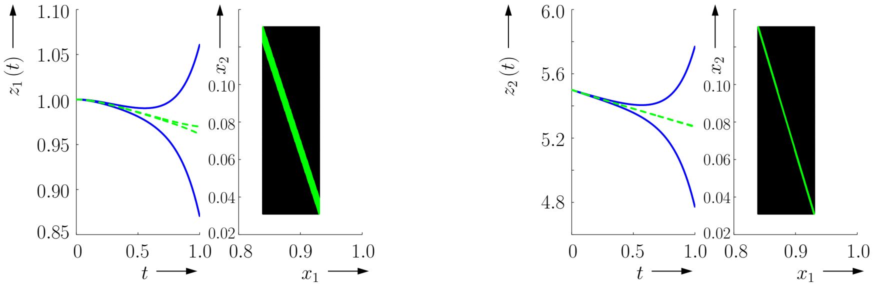


Figure 2: Dynamic constraints $z_1(t)$ and $z_2(t)$ and their mapping into the $(x_1; x_2)$-plane for $t = 1.0$. Solid lines: evaluation of $z_i(t)$ using equation (11); dashed lines: evaluation of $z_i(t)$ using equation (12).

test [7] based on backward integration. In Fig. 1, a significant reduction of overestimation can be seen at the points $t = 1.0$ and $t = 2.0$ at which the consistency was tested according to (14). The simulation was performed for $u(t) = 0.5 = \text{const}$. The regions which can be identified as inconsistent are marked in black in Fig. 2.

The new exponential enclosure technique according to Algorithm 2 leads to significantly tighter bounds in this example. To improve its efficiency, linear state equations are transformed into real Jordan normal form. This leads to a decoupling of the state equations and reduces the wrapping effect. For nonlinear systems, the transformation is performed by the matrix of the eigenvectors of the Jacobian of the state equations which are evaluated for the interval midpoints of all uncertain variables. However, this procedure does not lead to a complete decoupling of the state equations. Therefore, a combination of this technique with the previously mentioned consistency test is inevitable for non-negligible uncertainties as well as nonlinear or not asymptotically stable systems.

4 Verified Sensitivity Analysis

The sensitivity of the solution $x(t)$ of the set of ODEs $\dot{x}(t) = f(x(t), p)$ with respect to a time-invariant parameter vector p is defined by the partial derivative of $x(t) \in \mathbb{R}^{n_x}$

with respect to all p_i, $p \in \mathbb{R}^{n_p}$. The introduction of new state vectors

$$s_i(t) := \frac{\partial x(t)}{\partial p_i} \in \mathbb{R}^{n_x} \tag{15}$$

for all $i = 1, \ldots, n_p$ leads to the additional ODEs

$$\dot{s}_i(t) = \frac{\partial f(x(t), p)}{\partial x} \cdot s_i(t) + \frac{\partial f(x(t), p)}{\partial p_i} \quad \text{with} \quad s_i(0) = \frac{\partial x(0, p)}{\partial p_i} \quad . \tag{16}$$

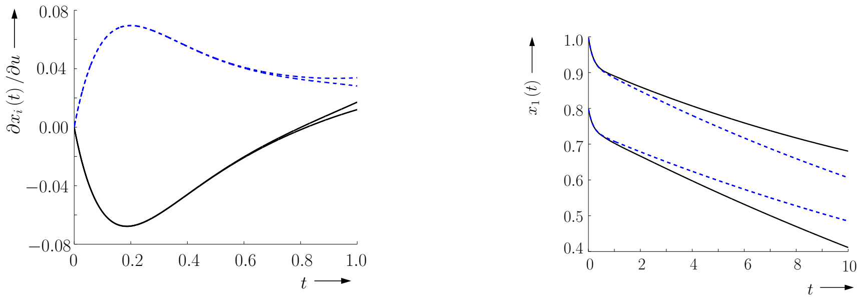
The differential equations for $s_i(t)$ are evaluated along the verified enclosures of the trajectories of the system states $x(t)$. In ValEncIA-IVP, all partial derivatives — also those required to determine the sensitivities $s_i(t)$ — are computed by algorithmic differentiation provided by FADBAD++. Therefore, there is no need to derive the sensitivity equations (16) manually.

Important special cases are $s_i(0) = 0$ occurring when the initial conditions do not depend on the uncertain parameters and $s_i(0) = e_i$, $\frac{\partial f(x(t),p)}{\partial p_i} = 0$ when sensitivities with respect to initial conditions $x(0)$ are determined (e_i is the i-th unit vector, $i = 1, \ldots, n_x$). Fig. 3(a) shows the verified sensitivities for the catalytic reactor with respect to a constant catalyst concentration $u(t) = 0.5$. It can be seen that an increase of u leads to a larger concentration $x_2(t)$ for all $t \in [0\ ;\ 1]$ (dashed lines) and, at the beginning, to a smaller concentration $x_1(t)$ (solid lines). Towards the end of this interval, the absolute values of the sensitivities get significantly smaller.

Additionally, the sensitivity analysis provides a possibility for refinement of the state enclosures $[x_{encl}(t)]$ using the mean-value theorem

$$x(t) \in [x_m(t)] + \left.\frac{\partial x(t)}{\partial p}\right|_{p \in [p]} \cdot ([p] - p_m) \quad , \tag{17}$$

where $[x_m(t)]$ is the interval enclosure for the nominal IVP with $p_m = \mathrm{mid}([p])$.



(a) Sensitivity analysis.

(b) Cooperativity test.

Figure 3: Sensitivity analysis for the catalytic reactor w.r.t. the control input u and verified cooperativity test for reduction of overestimation.

5 Cooperativity Test of Dynamical Systems

So far, we discussed techniques for reduction of overestimation which rely on modifications in the evaluation of the iteration formulas (4) and (8) of ValEncIA-IVP or which make use of consistency tests relying on dynamical constraints.

Additionally, we can exploit cooperativity for this purpose. This system property is relevant for many mathematical models in bioprocess and chemical engineering. Due to the modular structure of ValEncIA-IVP, the routines for verified sensitivity analysis can be reused in an automatic cooperativity test. Consider two vectors $x^{<1>}(t_0)$ and $x^{<2>}(t_0)$ of initial conditions of a dynamical system with

$$x_i^{<1>}(t_0) \geq x_i^{<2>}(t_0) \quad \text{for all} \quad i = 1, \ldots, n_x \ . \tag{18}$$

A system is cooperative if the property

$$x_i^{<1>}(t) \geq x_i^{<2>}(t) \quad \text{for all} \quad i = 1, \ldots, n_x \tag{19}$$

holds, where the superscript indices denote the solutions for the two different initial conditions. In this case, the simulation of an uncertain dynamical system can be restricted to separate verified simulations for all vertices of $[x(t_0)]$, see [1]. A guaranteed enclosure of each component $x_i(t)$ is then given by the smallest and largest resulting state values for each point of time.

Cooperativity can be checked using verified evaluation of the sign conditions

$$\frac{\partial f_i}{\partial x_j} \geq 0 \quad \text{for all} \quad i, j = 1, \ldots, n_x \quad \text{with} \quad i \neq j \ . \tag{20}$$

The sign conditions (20) along with the non-negativity of the state variables are *sufficient* for the cooperativity of a dynamical system. The non-negativity of the states is guaranteed for non-negative initial conditions if the differential inequalities

$$\dot{x}_i = f_i(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_{n_x}) \geq 0 \tag{21}$$

hold for all $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n_x} \geq 0$. A generalization of the above-mentioned sufficient condition for cooperativity is given by

$$(-1)^{\epsilon_i + \epsilon_j} \cdot \frac{\partial f_i}{\partial x_j} \geq 0 \quad \text{for all} \quad i, j = 1, \ldots, n_x, \quad i \neq j, \text{ and } \ \epsilon_i, \epsilon_j \in \{0; 1\} \ . \tag{22}$$

Consider the catalytic reactor (10) with the uncertain initial conditions $x(0) \in \begin{bmatrix}[0.8\ ;\ 1.0] & [0\ ;\ 0]\end{bmatrix}^T$. This system is cooperative because the conditions (20)–(22) hold. In Fig. 3(b), the verified enclosures for the state variable x_1 are compared for simulations using the exponential enclosure technique (solid lines, computed with Algorithm 2) for the complete initial state interval and for a simulation which makes use of the cooperativity property (dashed lines). In the latter case, the maximum deviation between the exact range and the width of the interval enclosure of $x_1(t)$ determined with the cooperativity test is smaller than $7.32 \cdot 10^{-4}$ for all $t \in [0\ ;\ 10]$. This accuracy is computed as the sum of the interval diameters of the verified simulations leading to the lower and upper interval bounds of $x_1(t)$ in the cooperativity test.

6 Application of ValEncIA-IVP to DAEs

ValEncIA-IVP has been extended recently to compute guaranteed state enclosures for DAEs. In the following, we consider semi-explicit DAEs

$$\dot{x}(t) = f(x(t), y(t), t) \qquad \text{with} \ \ f : D \mapsto \mathbb{R}^{n_x} \tag{23}$$

$$0 = g(x(t), y(t), t) \qquad \text{with} \ \ g : D \mapsto \mathbb{R}^{n_y}, D \subset \mathbb{R}^{n_x} \times \mathbb{R}^{x_y} \times \mathbb{R}^1 \tag{24}$$

with the consistent initial conditions $x(0)$ and $y(0)$. These DAEs may further depend on uncertain parameters p. To simplify the notation, the dependency on p is not explicitly denoted. However, all presented criteria are also applicable to systems with $p \in [\underline{p} \; ; \; \overline{p}]$.

Using verified DAE solvers, we can compute guaranteed state enclosures after determining consistent initial conditions for x and y. A further possible application is the computation of open-loop control strategies. They are determined in such a way that the system's output signal matches a predefined time response (inverse control problem).

6.1 Solving DAE Systems using Interval Arithmetic

The approach for solving DAEs using ValEncIA-IVP is based on the description of state enclosures for $x_i(t)$ and $y_j(t)$ by

$$\begin{aligned} [x_i(t)] &:= x_{app,i}(t_k) + (t - t_k) \cdot \dot{x}_{app,i}(t_k) + [R_{x,i}(t_k)] + (t - t_k) \cdot [\dot{R}_{x,i}(t)] \\ [y_j(t)] &:= y_{app,j}(t_k) + (t - t_k) \cdot \dot{y}_{app,j}(t_k) + [R_{y,j}(t)] \end{aligned} \tag{25}$$

with $i = 1, \ldots, n_x$, $j = 1, \ldots, n_y$, and $t \in [t_k \; ; \; t_{k+1}]$, $t_0 \leq t \leq t_f$. In (25), t_k and t_{k+1} are two subsequent points of time between which guaranteed state enclosures are determined. For $t = t_0$, the conditions $[x(t_0)] = x_{app}(t_0) + [R_x(t_0)]$ and $[y(t_0)] = y_{app}(t_0) + [R_y(t_0)]$ have to be fulfilled with non-verified approximate solutions $x_{app}(t)$ and $y_{app}(t)$ computed using, for example, DAETS [6][1]. Note that in contrast to the proposed extension of ValEncIA-IVP, DAETS provides floating point approximations to the solutions of DAEs. Therefore, it cannot be used to take into account interval uncertainties in parameters as demonstrated, for example, in the following subsection for the computation of consistent initial conditions with the help of ValEncIA-IVP.

Algorithm 3.

1. Determine hidden constraints that have to be fulfilled for both verified enclosures of initial conditions and the time responses $x(t)$ and $y(t)$. For that purpose, we consider algebraic equations $g_i(x)$ which do not depend explicitly on y. Differentiation with respect to time leads to

$$\frac{d^j g_i(x)}{dt^j} = \left(\frac{\partial L_f^{j-1} g_i(x)}{\partial x} \right)^T \cdot f(x, y) = L_f^j g_i(x) = 0 \; , \; L_f^0 g_i(x) = g_i(x) \; . \tag{26}$$

The Lie derivatives $L_f^j g_i(x)$ are computed by FADBAD++ up to the smallest order $j > 0$ for which $L_f^j g_i(x)$ depends on at least one component of y (i.e., up to the differentiation index of DAEs)[2].

2. Compute consistent initial conditions for the equations (23) and (24) using the Krawczyk iteration as an interval Newton technique so that the constraints (24) and (26) are fulfilled.

[1] For further information on DAETS see `http://www.cas.mcmaster.ca/~nedialk/daets/`.

[2] The procedure for computation of Lie derivatives in (26) is based on [9]. However, the author of [9] did not use this approach in a framework for verified solution of DAEs.

Table 1: Candidates for consistent initial conditions.

	$[x_1]$	$[x_2]$	$[x_3]$
candidate 1	$[\ 0.7716\ ;\ \ 0.8016]$	$[-0.6362\ ;\ -0.5979]$	$[\ 0.7535\ ;\ \ 0.8162]$
candidate 2	$[-1.0010\ ;\ -0.9995]$	$[-0.0312\ ;\ \ 0.0291]$	$[-0.0315\ ;\ \ 0.0294]$
candidate 3	$[-0.8016\ ;\ -0.7708]$	$[-0.6361\ ;\ -0.5979]$	$[-0.8162\ ;\ -0.7535]$
candidate 4	$[\ 0.9995\ ;\ \ 1.0009]$	$[-0.0312\ ;\ \ 0.0291]$	$[-0.0294\ ;\ \ 0.0315]$

3. Substitute the state enclosures (25) for the vectors x and y in (23) and (24) and solve the resulting equations for $[\dot{R}_x(t)]$ and $[R_y(t)]$ with the help of the Krawczyk iteration. Consider the hidden constraints (26) to restrict the set of feasible solutions.

To tighten the state enclosures computed by Algorithm 3, we can extend it in the following way. Consider intervals $[R_y(t)]$ determined in Algorithm 3 as parameters of the ODEs (23). Then, use the techniques presented in Algorithm 1 and 2 for ODEs to improve the enclosures for the differential state variables x. In principle, techniques for verified sensitivity analysis can also be implemented for sets of DAEs using algorithmic differentiation.

6.2 Consistent Initial Conditions: A Simple Pendulum

To demonstrate the use of the hidden constraints (26), the pendulum example from [6] which is rewritten as a set of first order differential equations is considered in the normalized form

$$\begin{aligned} &\dot{x}_1 = x_3 \qquad && \dot{x}_3 = -x_1 y \qquad && g(x) = x_1^2 + x_2^2 - 1 = 0 \\ &\dot{x}_2 = x_4 && \dot{x}_4 = -x_2 y + 1\ . && \end{aligned} \tag{27}$$

As described in Algorithm 3, ValEncIA-IVP can be used to verify given sets of initial conditions and to determine enclosures of consistent initial values within a certain domain. For that purpose, it can be specified which components of the initial state vectors are allowed to be modified by ValEncIA-IVP and which values or intervals with non-vanishing diameters have to stay fixed. This routine is interfaced with SmartMOBILE [2] for modeling and simulation of multibody dynamics. In Tab. 1, all candidates for consistent initial state enclosures for the simple pendulum were computed using the Krawczyk iteration for $x_1 \in [-5.0\ ;\ 5.0]$, $x_2 \in [-5.0\ ;\ 5.0]$, and $x_3 \in [-5.0\ ;\ 5.0]$ as well as the *fixed* intervals $x_4 \in [0.99\ ;\ 1.01]$ and $y \in [0.99\ ;\ 1.01]$.

With the help of the Krawczyk iteration, it is possible to show that consistent initial conditions for the system (27) are guaranteed to exist in each of the four interval boxes. This verification is only possible since the initial search domain is subdivided automatically by ValEncIA-IVP. For the purpose of verification, the interval Newton method is used to compute consistent initial conditions within each of these subintervals under consideration of the hidden constraints (26) of the orders $j \in \{0, 1, 2\}$. Note that non-verified solvers such as DAETS can only provide approximations for one of the solutions in Tab. 1 and cannot take into account the fixed bounds for $[x_4]$ and $[y]$.

6.3 Control Sequences for a Simple Electrical Network

Consider the DAE system

$$\begin{aligned} i_C &= C \cdot \dot{u}_C & u_{in} &= u_R + u_L + u_C & R \cdot i_R &= u_R \\ u_L &= L \cdot \dot{i}_L & i_R &= i_L = i_C & & \end{aligned} \tag{28}$$

which describes the dynamics of a simple electrical network. It is a series connection of a resistor R, a capacitor C, and an inductor L. The voltages and currents are denoted by u_i and i_i, $i \in \{R, C, L\}$.

The desired output voltage $u_{out}(t) = u_C(t)$ is specified by an additional time-dependent, algebraic constraint

$$u_{out}(t) = u_C(t) = 1 + \frac{2}{3}\sqrt{3}\exp\left(-\frac{1}{2}t\right)\sin\left(\frac{1}{2}\sqrt{3}t\right) \quad . \tag{29}$$

To solve this control problem using a verified DAE solver, we introduce the state vectors

$$x(t) = \begin{bmatrix} u_C & i_L \end{bmatrix}^T \qquad \text{and} \qquad y(t) = \begin{bmatrix} i_C & u_L & u_R & i_R & u_{out} \end{bmatrix}^T \quad . \tag{30}$$

The consistent initial conditions which are considered in this example are

$$x(0) = \begin{bmatrix} 1 & 1 \end{bmatrix}^T \ , \quad y(0) = \begin{bmatrix} 1 & -1 & 1 & 1 & 1 \end{bmatrix}^T \ , \quad \text{and} \quad u_{in}(0) = 1 \ . \tag{31}$$

In Fig. 4, the results for computation of the admissible range of a continuous control sequence u_{in} with the a-priori bounds $\dot{u}_{in} \in [-50; 50]$ are shown. The enclosure of $u_C(t)$ contains the desired output for all points of time. Additionally, Fig. 4 shows that the a-priori bounds for $u_{in}(t)$, resulting from the initial bounds for $\dot{u}_{in}(t)$, are narrowed to a range that is consistent with $u_{out}(t)$.

If inconsistent initial conditions are specified or if the range for $\dot{u}_{in}$ is too small, ValEncIA-IVP adjusts both intervals automatically. In this case, interval components specified to be fixed are not modified. It is furthermore reported if a control problem cannot be solved within the specified output tolerances.

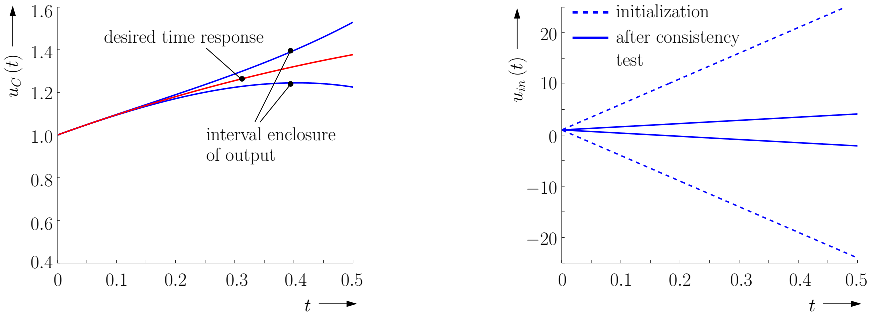


Figure 4: Controller design for an electrical network using verified DAE solvers.

This problem is usually solved analytically in control engineering. For example, for nonlinear exactly input-to-state linearizable sets of ODEs, the control input $u(t)$ (as one component of $y(t)$ in (23),(24)) is expressed in terms of the state variables after exact feedback linearization. However, numerical design approaches based on interval analysis are more flexible since uncertainties and robustness requirements can be taken into account directly. For that purpose, system models described by ODEs and DAEs are extended by time-dependent algebraic constraints to specify the desired output. The corresponding solution provides both the control sequence and an enclosure of all reachable states.

7 Conclusions and Outlook on Future Research

In this paper, we presented several algorithmic improvements of the verified ODE solver ValEncIA-IVP for reduction of overestimation and for enhancement of performance. Moreover, we extended its applicability to IVPs for DAEs and verified control synthesis.

In future work, the DAE solver will be applied to further nonlinear system models. Additionally, it will be used for state, parameter, and disturbance estimation in open-loop and closed-loop control systems. For both ODEs and DAEs, this task can be formulated as a system of DAEs if at least one component of the state vector is accessible for measurements. Finally, we will address the problem of controller design for nonlinear differentially flat systems which cannot be linearized exactly using static feedback control. An important extension of this task will be trajectory planning and control for non-flat system models.

References

[1] D. Angeli and E. D. Sontag, “Monotone Control Systems”, *IEEE Transactions on Automatic Control*, vol. 48, no. 10, pp. 1684–1698, 2003.

[2] E. Auer, A. Rauh, E. P. Hofer, and W. Luther, “Validated modeling of mechanical systems with SmartMOBILE: improvement of performance by ValEncIA-IVP”, In: *Proc. of Dagstuhl Seminar 06021: Reliable Implementation of Real Number Algorithms: Theory and Practice, Dagstuhl, Germany*, Springer Lecture Notes in Computer Science, vol. 5045, pp. 1–27, 2008.

[3] C. Bendsten and O. Stauning, FADBAD++ 2.1, 2007, `http://www.fadbad.com`.

[4] P. P. Czechowski, L. Giovannini, and A. W. Ordys, “Testing algorithms for inverse simulation” *Proc. of the 2006 IEEE Intl. Conference on Control Applications*, Munich, Germany, 2006.

[5] Ch. Keil, Profil/BIAS 2.0.6, 2008, `www.ti3.tu-harburg.de/keil/profil/`

[6] N. S. Nedialkov, “Interval tools for ODEs and DAEs”, In: *CD-Proc. of the 12th GAMM-IMACS Intl. Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany*, IEEE Computer Society, 2007.

[7] A. Rauh, E. Auer, and E. P. Hofer, “ValEncIA-IVP: A comparison with other initial value problem solvers”, In: *CD-Proc. of the 12th GAMM-IMACS Intl. Symposium on Scientific Computing, Computer Arithmetic, and Validated Numerics SCAN 2006, Duisburg, Germany*, IEEE Computer Society, 2007.

[8] A. Rauh, M. Brill, and C. Günther, "A novel interval arithmetic approach for solving differential-algebraic equations with ValEncIA-IVP", *Intl. Journal of Applied Mathematics and Computer Science AMCS*, vol. 19, no. 3, pp. 381–397, 2009.

[9] K. Röbenack, "On the efficient computation of higher order maps $ad_f^k g(x)$ using Taylor arithmetic and the Campbell-Baker-Hausdorff Formula", In: A. Zinober and D. Owens, editors, *Nonlinear and Adaptive Control*, vol. 281 of *LNCIS*, pp. 327–336. Springer, 2002.