Automatic Verification of Stability and Safety for Delay Differential Equations

Liang Zou[1], Martin Fränzle[2⋆], Naijun Zhan[1⋆⋆], and Peter Nazier Mosaad[2⋆⋆⋆]

[1] State Key Laboratory of Computer Science, Institute of Software, CAS, China
[2] Dpt. of Computing Science, C. v. Ossietzky Universität, Oldenburg, Germany

Abstract. Delay differential equations (DDEs) arise naturally as models of, e.g., networked control systems, where the communication delay in the feedback loop cannot always be ignored. Such delays can prompt oscillations and may cause deterioration of control performance, invalidating both stability and safety properties. Nevertheless, state-exploratory automatic verification methods have until now concentrated on ordinary differential equations (and their piecewise extensions to hybrid state) only, failing to address the effects of delays on system dynamics. We overcome this problem by iterating bounded degree interval-based Taylor overapproximations of the time-wise segments of the solution to a DDE, thereby identifying and automatically analyzing the operator that yields the parameters of the Taylor overapproximation for the next temporal segment from the current one. By using constraint solving for analyzing the properties of this operator, we obtain a procedure able to provide stability and safety certificates for a simple class of DDEs.

1 Introduction

"Despite [...] very satisfactory state of affairs as far as [ordinary] differential equations are concerned, we are nevertheless forced to turn to the study of more complex equations. [...] the rate of change of physical systems depends not only on their present state, but also on their past history." [2, p. iii]

Ever since we first managed to make a children's swing oscillate with ourselves sitting happily on top, all of us are perfectly aware of the impact of feedback delays on the performance of control loops. The same is true for more serious applications in automatic control, where digital implementation of the controller, though adding flexibility, comes at the price of introducing increasingly relevant delays into the feedback loop between controller and plant. The sources of

⋆ Work of this author has been supported by the Deutsche Forschungsgemeinschaft as part of the Transregional Collaborative Research Center SFB/TR 14 AVACS.

⋆⋆ Work of this author has been supported by "973 Program" under grant No. 2014CB340701, by NSFC under grants 91118007 and 91418204, and by the CAS/SAFEA International Partnership Program for Creative Research Teams.

⋆⋆⋆ Work of this author has been supported by the Deutsche Forschungsgemeinschaft as part of the Research Training Group DFG GRK 1765 SCARE.

such delays are manifold: conversions between analog and digital signal domains, complex digital signal-processing chains enhancing, filtering, and fusing sensory signals before they enter control, sensor networks harvesting multiple sensor sources before feeding them to control, or network delays in networked control applications physically removing the controller(s) from the control path. In each such application, describing the feedback dynamics of the controlled system by conjoining the ordinary differential equations (ODEs) describing the plant dynamics with the ODEs describing control may be misleading, as the delays introduced into the feedback loop may induce significantly deviating dynamics; cf. Fig. 1 for a simple example. Delays may prompt oscillations in otherwise convergently stable feedback loops or vice versa, they can destabilize otherwise stable orbits [40], can stretch dwell times, may induce residual error that never settles, or can cause transient overshoot into unsafe operational regimes (e.g. to negative values in Fig. 1), to name just a few of the various possible effects fundamentally altering system dynamics. Unmodeled delays in a control loop thus have the potential to invalidate any stability or safety certificate obtained on the delay-free model, as delays may significantly deteriorate control performance.

Given the omnipresence of such delays in modern control schemes, the apparent lack of tools permitting their safe automatic analysis surprises. While delay differential equations (DDEs) describing system dynamics as a function

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = f(\boldsymbol{x}(t), \boldsymbol{x}(t-\delta_1), \ldots, \boldsymbol{x}(t-\delta_n)), \text{ with } \delta_n > \ldots > \delta_1 > 0, \quad (1)$$

of past system states have long been suggested as an adequate means of modeling delayed feedback systems [2], their tool support still seems to be confined to numerical simulation based on integration from discontinuity to discontinuity, e.g. by Matlab's `dde23` algorithm. Such numerical simulation, despite being extremely useful in system analysis, nevertheless fails to provide reliable certificates of system properties, as it is numerically approximate only — in fact, error control even is inferior to ODE simulation codes as dynamic step-size control is much harder to attain for DDEs due to the non-local effects of step-size changes. Counterparts to the plethora of techniques for safely enclosing set-based initial value problems of ODEs, be it safe interval enclosures [27, 37, 25], Taylor models [3, 28], or flow-pipe approximations based on polyhedra [6], zonotopes [13], ellipsoids [19], or support functions [22], are thus urgently needed for DDEs. As in the ODE case, such techniques would safely (and preferably tightly) overapproximate the set of states reachable at any given time point from the set of initial values. The reason for their current lack is that DDEs are in some respect much more complex objects than ODEs: DDEs belong to the class of systems with functional state, i.e., the future (and past) is not determined by a single temporal snapshot of the state variables, yet by a segment of a trajectory. This renders the systems infinite-dimensional; in fact, as can be seen from Eq. (1), transformed copies of the initial segment of duration δ_n will generally be found in higher-order derivatives of $\boldsymbol{x}(t)$ even after arbitrarily long time.

A safe enclosure method for DDEs therefore has to manipulate computational enclosures of sets of trajectory segments $\boldsymbol{x} : [a, b] \to \mathbb{R}^n$ rather than computational enclosures of sets of states $\boldsymbol{x} \in \mathbb{R}^n$, like interval boxes, zonotopes, ellipsoids, or support functions. A reasonable data structure could be interval-based Taylor forms, being able to enclose a set of functions by a parametric Taylor series with parameters in interval form. To avoid dimension explosion incurred by the ever-growing degree of the Taylor series along the time axis, following the idea of Taylor models [3, 28], we employ Taylor series of fixed degree and move higher-degree terms into the parametric uncertainty. We use this data structure to iterate bounded degree Taylor overapproximations of the time-wise segments of the solution to a DDE, thereby identifying and automatically analyzing the operator that yields the parameters of the Taylor overapproximation for the next temporal segment from the current one. By using constraint solving for analyzing the properties of this operator, we obtain an automatic procedure providing stability and safety certificates for a simple class of DDEs of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t) = f(\boldsymbol{x}(t-\delta)) \tag{2}$$

with linear or polynomial $f : \mathbb{R}^n \to \mathbb{R}^n$. While this form is very restrictive, in particular excluding immediate feedback between the state vector $\boldsymbol{x}(t)$ and its dynamics $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t)$ in the model of the physical plant, it serves well as an illustrative example for exposing the method, and can easily be generalized by combination with the well-developed techniques for flow-pipe approximations of ODEs.

2 Related Work

Driven by the demand for safety cases (in a broad sense) for safety-critical control systems, we have over the past decades seen a rapidly growing interest in automatic verification procedures for system models involving continuous quantities and dynamics described by, a.o., differential equations. Verification problems of primary interest are thereby invariance properties concerning the dynamically reachable states and stability properties describing the long-term behavior.

Invariance properties are a prototypical safety property. A natural approach to their automatic verification is state-space exploration aiming at computing the reachable state space. Unfortunately, only very few families of restrictive linear dynamic systems feature a decidable state reachability problem [20, 16]. A more generally applicable option is to compute overapproximations of the state sets reachable under time-bounded continuous dynamics, and then to embed them, e.g., into depth-bounded automatic verification by bounded model checking, or into unbounded verification by theorem proving. Among the many abstraction techniques proposed for over-approximating reachable sets of continuous dynamics given as ordinary differential equations are use of interval arithmetic [33, 27, 37, 25], Taylor models [3, 28], flow-pipe approximations based on polyhedra [6], zonotopes [13], ellipsoids [19], or support functions [22], and abstraction based on discovering invariants [36, 32, 31, 23]. There are several bounded model checkers available for continuous and hybrid systems, like iSAT-ODE [8], Flow* [5], and dReach [18], to name just a few. Theorem provers for ODE dynamics and hybrid systems are also available, e.g., KeYmaera [30] or HHL Prover [41].

Safety verification is complemented by automatic procedures for providing certificates of stability. Most such methods are based on the automatic construction of Lyapunov functions [4] or piecewise Lyapunov functions [29]. Again, such procedures can only be complete for restricted, mostly linear cases, though incomplete extensions to rather general classes exist, e.g. [24].

Delay differential equations (DDEs) [2] model continuous processes with delayed feedback, be it natural dynamic systems [1] or technical applications in automatic control, which increasingly feature feedback delay due to, a.o., communication networks. As the delay substantially alters system behavior, verification of properties of DDE is an independent area of research. Albeit there is extensive literature on the theory of DDEs, obviously also addressing the question of how to manually verify stability, fully automatic proof procedures for such models are currently lacking and thus provide an open area of research. To this end, it should be noted that DDE model a richer class of delay phenomena than sample-and-hold devices or sampled controllers, even if the latter come equipped with delayed output delivery. Such devices can well be modeled by hybrid automata, providing an infinite-state yet finite-dimensional Markovian model, and consequently can be analyzed by the corresponding verification tools. The functional state of DDE, in contrast, is infinite-dimensional.

3 Overview of Our Approach

A reasonably small delay does not affect the solution of a linear ordinary differential equation (ODE) much, such that analyzing the ODE derived from the DDE by ignoring the delays may be indicative of the overall behavior. Unfortunately, it is unclear how much delay can be ignored in general, as this depends on the property under investigation. The following example demonstrates the difference between a DDE and the related ODE obtained by neglecting delays.

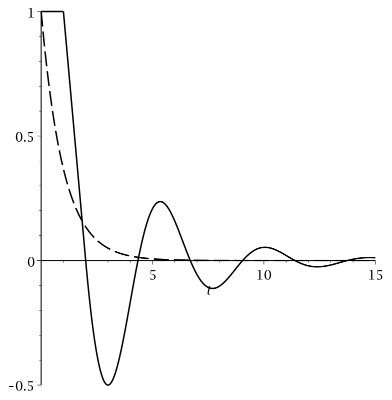

Fig. 1. Solutions to the ODE $\dot{x} = -x$ (dashed graph) and the related DDE $\dot{x}(t) = -x(t-1)$ (solid line), both on similar initial conditions $x(0) = 1$ and $x([0, 1]) \equiv 1$, respectively.

In Fig. 1, the dashed and solid lines represent the solution of the ODE $\dot{x} = -x$ without delay and of the related DDE $\dot{x}(t) = -x(t-1)$ with 1 second delay, respectively. Both are given as initial value problems, where for the ODE we assume an initial value $x(0) = 1$, which we generalize for the DDE to $x([0, 1]) \equiv 1$. Figure 1 demonstrates that the delay tremendously prolongs dwell times, as well as invalidates some safety properties: the dashed line (representing the ODE behavior) always stays above the horizontal axis whereas, in contrast, the solid line (representing the DDE solution) visits the negative range repeatedly. Even though the difference between the solutions of the ODE and the DDE becomes smaller when the delay turns smaller, it is in general hard to say how small a delay may ensure conservation of some safety property valid of the ODE. Hence, it is necessary to have native methods for analyzing the behaviour of DDE.

3.1 Computing Enclosures by Taylor Models

In the following, we will as a running example show how to analyze the DDE

$$\dot{x}(t) = -x(t-1) \tag{3}$$

with the initial condition $x([0,1]) \equiv 1$.

The solution to the DDE (3) can be computed segment-wise by integration, computing segments of duration 1 each. As we know the initial segment, we can set $f_0(t) = 1$ for $t \in [0,1]$, and can assume that the segment number $n \in \mathbb{N}$ satisfies $n \geq 1$ in what follows. Clearly, the solution of Eq. (3) over the time interval $(n, n+1]$ can be represented by using its solution over the previous 1 second interval (i.e., the solution on $(n-1, n]$) as follows:

$$x(n+t) = x(n) + \int_{n-1}^{n-1+t} -x(s)ds, \text{ for } t \in (0,1]. \tag{4}$$

We simplify Eq. (4) by renaming $x(n+t_1)$ to $f_n(t_1)$. Thus, $f_n(t) : (0,1] \to \mathbb{R}$ is the solution of Eq. (3) on interval $(n, n+1]$, but the domain of the solution is shifted to interval $(0,1]$ to obtain a normalized presentation, i.e.,

$$f_n(t) = f_{n-1}(1) + \int_0^t -f_{n-1}(s)ds, \qquad t \in (0,1] \tag{5}$$

From Eq. (5) it follows that the degree of the solution f_n over the n-th interval will be $n-1$, e.g., 3599 after one hour. Therefore, even if the DDE easily is solvable by polynomials, its representation rapidly gets too complex to be algorithmically analyzable due to excessive degrees and number of monomials. For instance, it is hard to calculate the reachable set of the DDE in Eq. (3).

In order to address this issue, we will propose a method based on bounded-degree interval Taylor models to over-approximate the solution by polynomials with fixed degree. For instance, suppose we are trying to over-approximate the solution by polynomials of degree 2. We can then predefine a template of the form $f_n(t) = a_{n0} + a_{n1}t + a_{n2}t^2$ on interval $[n, n+1]$, where a_{n0}, a_{n1}, and a_{n2} are interval parameters able to incorporate the approximation error necessarily incurred by bounding the degree of the polynomial. Thus, the solution on the next interval can be safely over-approximated using such a Taylor model.

To compute the Taylor model, we first need to obtain the first and second derivative $f_{n+1}^{(1)}(t)$ and $f_{n+1}^{(2)}(t)$ of solution segment $n+1$ based on the preceding segment. The first derivative $f_{n+1}^{(1)}(t)$ is computed directly from Eq. (3) as

$$f_{n+1}^{(1)}(t) = -f_n(t) = -a_{n0} - a_{n1}t - a_{n2}t^2 .$$

The second derivative $f^{(2)}_{n+1}(t)$ is computed based on $f^{(1)}_{n+1}(t)$ by

$$f^{(2)}_{n+1}(t) = \frac{d\,(f^{(1)}_{n+1}(t))}{dt} = -a_{n1} - 2a_{n2}t\,.$$

Note that while polynomial derivative rules do in general not lift to interval Taylor series, as the interval parameters permit to cover functions locally exhibiting larger derivatives, the generation process of the interval Taylor series for the first derivative avoids this fallacy here.

Using a Lagrange remainder with fresh variable $\xi_n \in [0,1]$, we hence obtain

$$\begin{aligned} f_{n+1}(t) &= f_n(1) + \frac{f^{(1)}_{n+1}(0)}{1!}t + \frac{f^{(2)}_{n+1}(\xi_n)}{2!}t^2 \\ &= (a_{n0} + a_{n1} + a_{n2}) - a_{n0}t - \frac{a_{n1} + 2a_{n2}\xi_n}{2}t^2. \end{aligned}$$

In order to proceed towards analysis of the asymptotic behavior of the system, we in a second step derive the operator expressing the relation between Taylor coefficients in the current and the next step. By replacing $f_{n+1}(t)$ with its parametric form $a_{n+1_0} + a_{n+1_1}t + a_{n+1_2}t^2$ in the above equation, one therefore derives the operator

$$\begin{bmatrix} a_{n+1_0} \\ a_{n+1_1} \\ a_{n+1_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\xi_n \end{bmatrix} \begin{bmatrix} a_{n0} \\ a_{n1} \\ a_{n2} \end{bmatrix} \tag{6}$$

mapping the coefficients of the Taylor form at step f_n to the coefficients of the Taylor form of f_{n+1}. Hence, the coefficients change every second according to the above linear operator, which can be made time-invariant (yet interval-valued) by replacing ξ_n with its interval $[0,1]$.

Having obtained such a linear and time-invariant discrete system, we can in a third step determine whether this discrete dynamic system is *asymptotically* or *robustly stable* using the method proposed in [7]. If this holds, the sequence of coefficients finally converges to an equilibrium point, which in turn implies that the DDE in Eq. (3) is also *asymptotically* or *robustly stable*.

If we are interested in safety verification rather than stability, the above operator can be iterated within *bounded model checking* (BMC), using any BMC tool built on top of an arithmetic SMT solver being able to address polynomial arithmetic, e.g. iSAT [11]. For a given safety property like $S(x) \hat{=} -1 \leq x \leq 1$, the requirement in the n-th segment translates to $\forall t \in [0,1] : S(f_n(t))$, where f_n is the Taylor form stemming from the n-th iteration of the above linear operator. Hence, the safety property $S(x)$ for system (3) becomes safety property $\forall n \in \mathbb{N}, t \in [0,1] : S(f_n(t))$ in system (6). Discharging this proof obligation in BMC requires polynomial constraint solving due to the Taylor forms involved.

We can also conduct *unbounded safety verification* by means of pursuing BM-C for sufficiently many steps k_s in case our DDE is stabilizing. The corresponding upper bound k_s on the number of steps can be computed via the following procedure (please refer to [7] for details): The *asymptotic* or *robust stability* of the linear time-invariant discrete dynamic system in Eq. (6) is guaranteed by solving a linear matrix inequality given by Theorem 1, which also gives a Lyapunov function $V(a_{n0}, a_{n1}, a_{n2}, \xi_n)$ (denoted by $V(\boldsymbol{A}(n), \xi_n)$ in the following, where $\boldsymbol{A}(n)$ represents a_{n0},a_{n1}, and a_{n2}). Using the Lyapunov function, we first compute by iSAT3 the largest c such that $V(\boldsymbol{A}(n), \xi_n) \leq c \wedge \neg\mathcal{S}(f_n(t))$ is unsatisfiable. Then we calculate the minimum reduction d_m on the condition $V(\boldsymbol{A}(n), \xi_n) \geq c$, i.e.

$$d_m = \min\{V(\boldsymbol{A}(n), \xi_n) - V(\boldsymbol{A}(n+1), \xi_{n+1}) \mid V(\boldsymbol{A}(n), \xi_n) \geq c\},$$

where the constraint can be eliminated by Lagrange multipliers and d_m can be calculated by Matlab function `fmincon`. The existence of such c implies that $V(\boldsymbol{A}(n), \xi_n) \leq c \to \mathcal{S}(f_n(t))$ holds, which implies that after $k_s = \frac{V(\boldsymbol{A}(0),\xi_0)-c}{d_m}$ steps we can be sure to reside inside the safety region $S(f_n(t))$. As $V(\boldsymbol{A}(0), \xi_0)$ is linear in ξ_0 (as explained in the next section), it follows that it is monotonic or antimonic in ξ_0 and thus $\max\left(\frac{V(\boldsymbol{A}(0),0)-c}{d_m}, \frac{V(\boldsymbol{A}(0),1)-c}{d_m}\right)$ provides an upper bound for k_s. Hence, all that remains to be done is to pursue BMC for k_s steps, as safety violations can only arise transiently during those first k_s steps.

In fact, there is no need to blindly unwind and compute the BMC problems up to depth k_s. Instead, it suffices to do so until the Lyapunov function decreases to below c —which is guaranteed after at most k_s steps, but maybe faster— and then stop. Hence, we may save a lot of computations by checking for the goal $\neg\mathcal{S}(f_n(t)) \vee V(\boldsymbol{A}, \xi) \leq c$ at each step in our BMC process. If the condition holds, the bounded model checking procedure terminates immediately. Then of course we need to disambiguate cases by determining which disjunct in $\neg\mathcal{S}(f_n(t)) \vee V(\boldsymbol{A}, \xi) \leq c$ is satisfied. If the first alternative $\neg\mathcal{S}(f_n(t))$ is satisfied, then a counter-example to the safety property is found, otherwise the safety property has been certified by the BMC in at most k_s steps.

In this example, no linear (i.e., Taylor order 1) enclosure for the DDE in Eq. (3) suffices to prove the safety property $-1 \leq x \leq 1$, but the enclosure computed for degree 2 guarantees it.

4 Formal Analysis of Polynomial DDEs

In this section, we will generalize the basic idea to a general technique for polynomial DDE of shape (2). The DDE under consideration thus are of the form

$$\dot{\boldsymbol{x}}(t+\delta) = \boldsymbol{g}(\boldsymbol{x}(t)),\ \forall t \in [0, \delta] : \boldsymbol{x}(t) = \boldsymbol{p}_0(t), \tag{7}$$

where $\boldsymbol{x}$ is a state vector in $\mathbb{R}^m$, $\boldsymbol{p}_0(t)$ is a vector of polynomials in $\mathbb{R}^m[\boldsymbol{x}]$ representing the initial condition as a trajectory of the DDE in the initial δ time units, and $\boldsymbol{g}$ is a vector of polynomials in $\mathbb{R}^m[\boldsymbol{x}]$.

In order to compute an enclosure for the trajectory defined by DDE (7), we predefine a template interval Taylor form of fixed degree k as

$$\boldsymbol{f}_n(t) = \boldsymbol{a}_{n_0} + \boldsymbol{a}_{n_1} t + \cdots + \boldsymbol{a}_{n_k} t^k, \tag{8}$$

where $\boldsymbol{a}_{n_0}, \ldots, \boldsymbol{a}_{n_k}$ are interval-vector parameters. As before, $\boldsymbol{f}_n$ is used to enclose the trajectory for time interval $[n\delta, (n+1)\delta]$. In what follows, we set $\boldsymbol{f}_0(t) = \boldsymbol{p}_0(t)$ and will compute the successive $\boldsymbol{f}_n$ recursively from it. For notational convenience, we denote $[\boldsymbol{a}_{n_0}, \ldots, \boldsymbol{a}_{n_k}]$ by a matrix $\boldsymbol{A}(n)$ in $\mathbb{R}^{m\times(k+1)}$.

4.1 Constraints on Interval Parameters

As explained in Section 3, the trajectory induced by the DDE in Eq. (7) can be represented by a piecewise function, with the duration of each piece being the feedback delay δ. In order to compute an enclosure for the whole trajectory of the DDE, we may calculate the relation between $\boldsymbol{A}(n)$ and $\boldsymbol{A}(n+1)$. In contrast to the linear case of the previous section, we now need to exploit different orders of Lie derivatives $\boldsymbol{f}^{(1)}_{n+1}, \boldsymbol{f}^{(2)}_{n+1}, \ldots, \boldsymbol{f}^{(k)}_{n+1}$, which can be computed as follows:

$$\boldsymbol{f}^{(1)}_{n+1}(t) = \boldsymbol{g}(\boldsymbol{f}_n(t)), \boldsymbol{f}^{(2)}_{n+1}(t) = \frac{d\,\boldsymbol{f}^{(1)}_{n+1}(t)}{dt}, \ldots, \boldsymbol{f}^{(k)}_{n+1}(t) = \frac{d\,\boldsymbol{f}^{(k-1)}_{n+1}(t)}{dt}, \tag{9}$$

i.e., the first-order Lie derivative is obtained directly from Eq. (7) and the $(i+1)$-st order Lie derivative is computed from the i-th order Lie derivative by symbolic differentiation. The Taylor expansion of $\boldsymbol{f}_{n+1}(t)$ is derived from this as

$$\boldsymbol{f}_{n+1}(t) = \boldsymbol{f}_n(\delta) + \frac{\boldsymbol{f}^{(1)}_{n+1}(0)}{1!}t + \cdots + \frac{\boldsymbol{f}^{(k-1)}_{n+1}(0)}{(k-1)!}t^{k-1} + \frac{\boldsymbol{f}^{(k)}_{n+1}(\boldsymbol{\xi}_n)}{k!}t^k\,, \tag{10}$$

where $\boldsymbol{\xi}_n$ is a vector ranging over $[0,\delta]^m$.

From Eq. (10), by comparing the coefficients of the monomials with the same degree at the two sides, a relation between $\boldsymbol{A}_n$ and $\boldsymbol{A}_{(n+1)}$ is obtained. It can be represented as a vector of polynomial equations possibly involving $\boldsymbol{\xi}_n$, say

$$\boldsymbol{A}(n+1) = \boldsymbol{R}(\boldsymbol{A}(n), \boldsymbol{\xi}_n) \tag{11}$$

where $\boldsymbol{R}$ is a vector of polynomial functions of overall type $\mathbb{R}^{m(k+2)} \to \mathbb{R}^{m(k+1)}$.

After substituting $\boldsymbol{\xi}$ with interval $[0,\delta]$, Eq. (11) again forms a time-invariant discrete dynamic system. The stability of this system can again be determined by existing approaches, as can the bounded and unbounded model-checking problems of the original system (7). We will elaborate on the approach subsequently.

4.2 Stability of the Time-Invariant Discrete Dynamic System

In this section, we discuss how to determine the stability of the resulting time-invariant discrete dynamic system in Eq. (11), which implies stabilization of the original system (7) to a stable orbit $\boldsymbol{f}_{\to\infty}$ which cycles through every δ time units. We distinguish a *linear* and a more general *polynomial case* concerning the right-hand side $\boldsymbol{g}$ of the DDE (as well as the initial condition).

Linear $\boldsymbol{g}$: In case $\boldsymbol{g}$ in (7) is a linear function, $\boldsymbol{f}_{n+1}^{(1)}(t), \ldots, \boldsymbol{f}_{n+1}^{(k)}(t)$ are all linear in the entries of $\boldsymbol{A}(n)$ according to Eq. (9). Using Eq. (10), the equation (11) can hence be reformulated as

$$\boldsymbol{A}(n+1) = T(\boldsymbol{\xi}_n)\boldsymbol{A}(n)\,, \tag{12}$$

with $T(\boldsymbol{\xi}_n)$ an $m \times m$-matrix whose entries are linear in the components of $\boldsymbol{\xi}_n$.

The stability analysis for a linear time-invariant discrete dynamic system of form (12) can be pursued using the following theorem from [7]:

Theorem 1 (Stability Analysis [7]). *A system of the form*

$$\boldsymbol{x}(n+1) = T(\boldsymbol{\xi}_n)\boldsymbol{x}(n)\,, \quad T(\boldsymbol{\xi}_n) = \sum_{i=1}^{N} \boldsymbol{\lambda}_{ni} T_i\,, \quad \boldsymbol{\lambda}_{ni} \geq 0, \quad \sum_{i=1}^{N} \boldsymbol{\lambda}_{ni} = 1$$

is asymptotically/robustly stable *if and only if there exist symmetric positive definite matrices S_i, S_j and matrices G_i with appropriate dimensions such that*

$$\begin{bmatrix} G_i + G_i^T - S_i & G_i^T\, T_i^T \\ T_i\, G_i & S_j \end{bmatrix} > \mathbf{0}$$

for all $i = 1, ..., N$ and $j = 1, ..., N$. Moreover, the corresponding Lyapunov function is $V(\boldsymbol{x}(n), \boldsymbol{\xi}_n) = \boldsymbol{x}(n)^T (\sum_{i=1}^{N} \boldsymbol{\lambda}_{ni} S_i^{-1}) \boldsymbol{x}(n)$.

In order to exploit Theorem 1, we have to reformulate $T(\boldsymbol{\xi}_n)$ in Eq. (12) to

$$T(\boldsymbol{\xi}_n) = \sum_{i=1}^{N} \boldsymbol{\lambda}_{ni} T_i, \qquad \text{where } \boldsymbol{\lambda}_{ni} \geq 0, \quad \sum_{i=1}^{N} \boldsymbol{\lambda}_{ni} = 1. \tag{13}$$

From Equations (8) and (9), we recover that the degree of $\boldsymbol{f}_n^{(i)}(t)$ is $k+1-i$, for $i = 1, \cdots, k$. Furthermore, according to Eq. (12), each entry t_{ij} of $T(\boldsymbol{\xi}_n)$ is linear in the components of $\boldsymbol{\xi}_n$, written as $t_{ij}(\boldsymbol{\xi}_n)$, for $i = 1, \ldots, m$ and $j = 1, \ldots, m$. For each $t_{ij}(\boldsymbol{\xi}_n)$, we have

$$t_{ij}(\boldsymbol{\xi}_n) = (1 - \frac{\boldsymbol{\xi}_{n1}}{\delta}) t_{ij}(\boldsymbol{\xi}_n)[0/\boldsymbol{\xi}_{n1}] + \frac{\boldsymbol{\xi}_{n1}}{\delta} t_{ij}[\delta/\boldsymbol{\xi}_{n1}], \tag{14}$$

where $e[b/a]$ stands for substituting b for a in e. Hence,

$$T(\boldsymbol{\xi}_n) = (1 - \frac{\boldsymbol{\xi}_{n1}}{\delta}) T(\boldsymbol{\xi}_n)[0/\boldsymbol{\xi}_{n1}] + \frac{\boldsymbol{\xi}_{n1}}{\delta} T(\boldsymbol{\xi}_n)[\delta/\boldsymbol{\xi}_{n1}] \tag{15}$$

as $t_{ij}(\boldsymbol{\xi}_n)$ is linear in $\boldsymbol{\xi}_{n1}$. Obviously, $0 \leq 1 - \frac{\boldsymbol{\xi}_{n1}}{\delta} \leq 1$ and $0 \leq \frac{\boldsymbol{\xi}_{n1}}{\delta} \leq 1$, as $\boldsymbol{\xi}_{n1} \in [0, 1]$. By repeating the above procedure m times, we obtain

$$T(\boldsymbol{\xi}_n) = \sum_{i=1}^{2^m} \lambda_i(\boldsymbol{\xi}_n) T_i, \quad \lambda_i(\boldsymbol{\xi}_n) \geq 0, \quad \sum_{i=1}^{2^m} \lambda_i(\boldsymbol{\xi}_n) = 1, \tag{16}$$

where T_i is a matrix, by substituting for each component of $\boldsymbol{\xi}_n$ either of the extremal values 0 or δ in $T(\boldsymbol{\xi}_n)$. This is sound due to the linearity in $\boldsymbol{\xi}_n$. Equation (16) constitutes a form where the stability of the time-invariant linear discrete dynamic system of Eq. (12) can be determined by the method of Theorem 1. Note that stabilization of the sequence of Taylor forms implies global stabilization of the underlying linear DDE (12), as the Taylor forms converge towards 0.

Polynomial $\boldsymbol{g}$: When $\boldsymbol{g}$ is *nonlinear*, the relation between $\boldsymbol{A}(n+1)$ and $\boldsymbol{A}(n)$ expressed in Eq. (11) becomes *nonlinear*. Thanks to existing methods on computing parametric Lyapunov functions, such as [24, 34], we can apply such techniques to analyze the stability of a time-invariant *polynomial* discrete dynamic system of Eq. (11), as it arises for polynomial $\boldsymbol{g}$. In this paper, we build on the idea from [34] and adapt it to the discrete-time setting of Eq. (11).

A parametric polynomial in $\boldsymbol{y}$ of degree k is of the form $\sum_{(\sum \boldsymbol{\alpha})\leq k} b_{\boldsymbol{\alpha}} \boldsymbol{y}^{\boldsymbol{\alpha}}$, where $\boldsymbol{y} = (y_1, \cdots, y_m), \boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_m), \boldsymbol{y}^{\boldsymbol{\alpha}} = y_1^{\alpha_1} \cdots y_m^{\alpha_m}, \sum \boldsymbol{\alpha} = \sum_{i=1}^{m} \alpha_i$. We will subsequently denote such a polynomial by $p(\boldsymbol{y}, \boldsymbol{b})$, where $\boldsymbol{b}$ stands the vector of the coefficients.

Definition 1. *Given a dynamic system as in Eq.* (11) *and state sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{BA}$ with $\mathcal{BA} \subset \mathcal{A}$, a parametric polynomial $p((\boldsymbol{A}, \boldsymbol{\xi}), \boldsymbol{b})$ is called a relaxed Lyapunov function with respect to $\mathcal{A}$ and $\mathcal{BA}$ iff*

$$\begin{aligned} &\exists \boldsymbol{b} \in \mathcal{B}. \forall \boldsymbol{A}(n) \in \mathcal{A}. \forall \boldsymbol{\xi}_n, \boldsymbol{\xi}_{n+1} \in [0, \delta]^m. \\ &\quad \boldsymbol{A}(n) \notin \mathcal{BA} \implies p((\boldsymbol{A}(n+1), \boldsymbol{\xi}_{n+1}), \boldsymbol{b}) - p((\boldsymbol{A}(n), \boldsymbol{\xi}_n), \boldsymbol{b}) < 0 \,, \end{aligned} \tag{17}$$

where $\mathcal{A}$ and $\mathcal{B}$ are domain constraints on $\boldsymbol{A}(n)$ and $\boldsymbol{b}$ respectively, $\mathcal{BA}$ is a basin of attraction.

In Definition 1, for any $\boldsymbol{b}_0$ which satisfies (17), the *relaxed Lyapunov function* $p((\boldsymbol{A}, \boldsymbol{\xi}), \boldsymbol{b}_0)$ behaves like Lyapunov function in $\mathcal{A} \setminus \mathcal{BA}$ for the time-invariant discrete polynomial dynamic system in Eq. (11), abbreviated as $V(\boldsymbol{A}, \boldsymbol{\xi})$. Computing such $\boldsymbol{b}$ satisfying (17) can be achieved with interval arithmetic according to the method given in [34] as follows:

Step 1: Replace vector variables $\boldsymbol{A}(n)$, $\boldsymbol{\xi}_n$, and $\boldsymbol{\xi}_{n+1}$ in (17) respectively with the corresponding intervals ($\boldsymbol{A}(n)$ is bounded by $\mathcal{A}$), and simplify the formula with interval arithmetic.

Step 2: Solve the resulting constraints on $\boldsymbol{b}$, which are a set of linear interval inequalities (LIIs), by Rohn's approach [35] (will be elaborated below).

Step 3: If the LIIs do not have a solution, bisect the intervals for vector variables $\boldsymbol{A}(n)$, $\boldsymbol{\xi}_n$ and $\boldsymbol{\xi}_{n+1}$, and repeat the steps 1 and 2 until a solution of $\boldsymbol{b}$ is found or the length of intervals is smaller than a prescribed threshold ϵ.

LIIs can be solved almost exactly using Rohn's approach [35] as follows: first, replace each variable a by an expression $a_1 - a_2$, where $a_1 \geq 0$ and $a_2 \geq 0$; then, replace $Ia \rhd 0$ by $Ia_1 - Ia_2 \rhd 0$, which is equivalent to $I^+a_1 - I^-a_2 \rhd 0$, where $I^+ = \max I, I^- = \min I$ and $\rhd \in \{<, \leq\}$. One thus derives a system of linear inequalities that can be solved by linear programming. The only problem is that $(Ia_1 - Ia_2)$ is not equivalent to (Ia) in general. However, if $\mathcal{B}$ is symmetric with respect to the origin, i.e. $b_i \in [-\mathcal{B}_i, \mathcal{B}_i]$ for each b_i of $\boldsymbol{b}$, and I^+ and I^- have a same sign, then the two formulas are equivalent by interval arithmetic. In an actual implementation, it is easy to guarantee the above condition.

If the method succeeds, it proves that the parameters of the Taylor form will eventually converge from anywhere in region $\mathcal{A}$ into region $\mathcal{BA}$. This implies that DDE (7) will converge into a corresponding region of its state space defined by the range over $t = [0, \delta]$ of the Taylor polynomials with parameters in $\mathcal{BA}$, whenever the DDE is started on initial conditions defined by the Taylor polynomials with parameters in $\mathcal{A}$. It thus constitutes a proof of local stability of the DDE.

4.3 Guaranteeing Safety

Now, we show how to compute the upper bound k_s on steps potentially leaving the safety region, as needed for unbounded verification of a given invariant $\mathcal{S}(\boldsymbol{x})$ by means of bounded model checking (BMC). As computation of k_s for linear $\boldsymbol{g}$ has already been elaborated in the end of section 3, the following discusses the computation of such k_s for polynomial $\boldsymbol{g}$ based on the above method.

When replacing the right-hand side constant 0 in (17) with a positive constant d_m, the above algorithm for computing relaxed Lyapunov functions will find a relaxed Lyapunov function that decreases by at least d_m for each step outside $\mathcal{BA}$. This can be used for unbounded safety verification, as it provides a computable bound for convergence into $\mathcal{BA}$, where for simplicity we here assume that $\mathcal{BA}$ is such that the range over $t = [0, \delta]$ of the Taylor polynomials with parameters in $\mathcal{BA}$ is a subset of our safety region $\mathcal{S}(\boldsymbol{x})$, i.e. the conjectured invariant.

Given such a minimum reduction d_m outside $\mathcal{BA}$, and thus outside the safety region $\mathcal{S}(\boldsymbol{x})$, we use iSAT to compute the largest c such that $V(\boldsymbol{A}(n), \boldsymbol{\xi}_n) \leq c \wedge \neg\mathcal{S}(\boldsymbol{f}_n(t)) \wedge t \in [0, \delta]$ is not satisfiable, where $S(\boldsymbol{x})$ is the invariant to be verified. Clearly, the existence of such c implies that $V(\boldsymbol{A}(n), \boldsymbol{\xi}_n) \leq c \to \mathcal{S}(\boldsymbol{f}_n)$ holds. Hence, after $k_s = \frac{V(\boldsymbol{x}(0), \boldsymbol{\xi}(0)) - c}{d_m}$ steps $S(\boldsymbol{f}_n)$ will necessarily hold, and safety violations can only occur transiently during the first k_s steps. Hence, using bounded model-checking for k_s steps yields an unbounded safety certificate in case no violation is detected before that step bound. Note that BMC here again requires polynomial SMT solving due to the Taylor forms. Again there is no need to always unwind the BMC problem to depth k_s, as checking the disjunctive goal $\neg\mathcal{S}(\boldsymbol{f}_n(t)) \vee V(\boldsymbol{A}(n), \boldsymbol{\xi}_n) \leq c$ and disambiguating the outcome probably permits early termination as in Sect. 3.

5 Implementation

The algorithms exposed in the previous section have been implemented in Matlab and C++, thereby taking advantage of the iSAT3 tool through its API. Given a DDE and the parameters relevant to the analysis, Matlab's symbolic computation is first employed for computing the Lie derivatives and thus identifying the discrete time-invariant operator connecting segments of the Taylor approximation. For the linear case, stability analysis is then conducted by the Matlab LMI solver, where actually synthesizing the pertinent Lyapunov function is done by Matlab matrix functions, and the minimum descent d_m outside the safety region is calculated by Matlab function `fmincon`. Polynomial stability analysis is based on Matlab and interval arithmetic packages `b4m` and `Profil`. Computation of a barrier value characterizing the safety region in terms of Lyapunov ranges is done by calling iSAT3. The same applies for bounded model checking.

6 Examples

In this section, we will introduce several examples to demonstrate how the approach works in practice. All these examples have been processed fully automatically by our prototype implementation.[3]

Example 1. Consider the linear DDE $\dot{x}(t) = -x(t-1)$ from Eq. (3)) with initial condition $x([0,1]) \equiv 1$ and check its stability as well as the safety property $\Box(-1 \leq x \leq 1)$.

Using a Taylor model with degree 1, we calculate the operator relating the parameters of successive Taylor forms to

$$\boldsymbol{A}(n+1) = \begin{bmatrix} 1 & 1 \\ -1 & -\xi_n \end{bmatrix} \boldsymbol{A}(n)\,.$$

This operator cannot be shown stable by the method of Theorem 1.

The operator automatically obtained for degree 2 has already been presented in Eq. (6). For this operator, stability verification by the method of Theorem 1 succeeds, as does (unbounded) safety verification for the property $\Box(-1 \leq x \leq 1)$ by bounded model checking.

Example 2. Consider the three-dimensional linear DDE

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -1 & \frac{1}{2} & 0 \\ \frac{1}{2} & -1 & \frac{1}{4} \\ 0 & \frac{1}{4} & -1 \end{bmatrix} \boldsymbol{x}(t - \frac{1}{100}) \tag{18}$$

with initial condition $\boldsymbol{x}([0,1]) \equiv [-\frac{125}{11}, -\frac{360}{11}, -\frac{90}{11}]$. This system, which has been inspired by [15, p. 585ff], models heat dissipation in a typical home with an insulated ground floor, topped by an attic without significant insulation and supported by a basement surrounded by earth. Up to a coordinate shift introduced in order to move the equilibrium point to $(0,0,0)$, its three variables x_1 to x_3 model the temperatures in the basement, the ground floor, and the attic, respectively. The standard model usually encountered in introductory textbooks on modeling with differential equations takes the dissipation equations to be

[3] The prototype implementation of the verification tool as well as the examples are available for download from `https://github.com/liangdzou/isat-dde`

ODEs; it is, however, reasonable to assume that the heat transfer through the walls between the three compartments actually takes time (one could also add delays for convective heat transport within the rooms). A DDE model thus seems in place here. While the actual transport delays would be state dependent, any reasonably sized constant delay will already make the model better. Not yet being able to deal with state-dependent delays, we have set the delay to $\frac{1}{100}$h for the sake of demonstration.

For the resulting system, we have automatically checked stability as well as safety with respect to the invariance property $\Box(x_2 \leq \frac{25}{11})$, where x_2 denotes the (shifted) temperature in the ground floor.

Using Taylor models of degree 1, we compute the operator relating successive parameters of the Taylor forms to

$$\boldsymbol{A}(n+1) = \begin{bmatrix} 1 & \frac{1}{100} & 0 & 0 & 0 & 0 \\ -1 & -\xi_1 & \frac{1}{2} & \frac{\xi_1}{2} & 0 & 0 \\ 0 & 0 & 1 & \frac{1}{100} & 0 & 0 \\ \frac{1}{2} & \frac{\xi_2}{2} & -1 & -\xi_2 & \frac{1}{4} & \frac{\xi_2}{4} \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{100} \\ 0 & 0 & \frac{1}{4} & \frac{\xi_3}{4} & -1 & -\xi_3 \end{bmatrix} \boldsymbol{A}(n)\,.$$

This operator has been shown stable by the method from Theorem 1 and the unbounded safety property has been verified by BMC.

Example 3. This example is an adaption of Gustafson's model of nutrient flow in an aquarium [15, p. 589f]. It deals with using a radioactive tracer for the food chain consisting of two aquatic plankton varieties drifting with the currents. The variables in this three-dimensional system reflect the isotope concentrations in the water, a phytoplankton species, and a zooplankton species, respectively. The original model was an ODE model; a concise model would presumably have to use PDE (partial differential equations) to model spacial variations and the necessary drifts of species in the predator-prey part of the food chain; our DDE model here is a compromise between these two extremes. Therefore consider the three-dimensional linear DDE

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} -3 & 6 & 5 \\ 2 & -12 & 0 \\ 1 & 6 & -5 \end{bmatrix} \boldsymbol{x}(t - \frac{1}{100}) \tag{19}$$

with initial condition $\boldsymbol{x}([0,1]) \equiv [10,0,0]$ and the conjectured invariant $\Box(x_1 - x_2 - x_3 \geq 5)$.

Beware that the eigenvalues of the matrix in this example are 0, $-10-\sqrt{6}$, $-10+\sqrt{6}$, which implies that not even the corresponding ODE is asymptotically stable. Hence, it comes as no surprise that we are not able to find an asymptotically stable enclosure to the DDE. Using Taylor models of degree 1, we calculate the operator relating successive parameter vectors to

$$\boldsymbol{A}(n+1) = \begin{bmatrix} 1 & \frac{1}{100} & 0 & 0 & 0 & 0 \\ -3 & -3\xi_1 & 6 & 6\xi_1 & 5 & 5\xi_1 \\ 0 & 0 & 1 & \frac{1}{100} & 0 & 0 \\ 2 & 2\xi_2 & -12 & -12\xi_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \frac{1}{100} \\ 1 & \xi_3 & 6 & 6\xi_3 & -5 & -5\xi_3 \end{bmatrix} \boldsymbol{A}(n)\,,$$

which cannot be proven stable by the method of Theorem 1. By translating above system into the form required for Theorem 1, we get a set of T_i. After calculation, we find out that the spectral radius of all the T_i is no more than 1, which at least shows that $\boldsymbol{A}(n)$ does not grow too fast. Using bounded model-checking on the Taylor approximation, we have been able to show the overapproximation unsafe in step 12 (corresponding to $t \in [0.12, 0.13]$s), while Simulink simulation confirms this for $t = 0.1452$s. Further exploiting iSAT3 for BMC on the Taylor-based overapproximation, we actually found the safe set $\{\boldsymbol{x} \mid x_1 - x_2 - x_e \geq 5\}$ itself (not its complement) being unreachable in step 18, i.e. for $t = 0.18$. This constitutes a rigorous automatic proof that the system actually is unsafe.

Example 4. Consider the polynomial DDE $\dot{x}(t) = -x(t-1)^3$ with initial condition $x([0,1]) \equiv c$, $c \in [3,6]$ arbitrary, and safety condition $\Box(-3000 < x < 3000)$.

This system is unsafe and Taylor approximations of arbitrary degree will thus eventually reach the complement of the safe set $S = \{x \mid -3000 < x < 3000\}$. Using a Taylor approximation of degree 5, we are able to show by BMC that the safe set S surely is left in the beginning of step 3, i.e., at $t = 3$, thus obtaining a rigorous automatic proof that the system actually is unsafe.

Execution times for each evaluation step in each example above are stated in Table 1, where the individual steps are calculating the Lyapunov function, the barrier c characterizing the safe set wrt. Lyapunov values, the minimum per-step reduction d_m of the Lyapunov function outside the safe set, and verifying the safety property, respectively. All benchmarks were performed on a 1.80GHz Intel Core-i5 processor with 4GB RAM running 64-bit Ubuntu 14.04.

7 Conclusions and Future Work

In this paper, we have exposed an automatic method for the stability and safety verification of a simple class of delay differential equations (DDEs). The method

	CLF(s)	CBV(s)	CSR(s)	VSP(s)
Ex. 1	1.9869	10.514	20.012	0.0302
Ex. 2	12.732	52.892	78.258	22.121
Ex. 3	55.053	skipped	skipped	0.0003
Ex. 4	timeout	skipped	skipped	0.0003

CLF	=	computing Lyapunov function
CBV	=	computing barrier value
CSR	=	computing per-step reduction
VSP	=	verifying/falsifying safety prop.

Table 1. Analysis times for the sample problems.

is based on using interval Taylor forms for safely enclosing segments of the solutions of DDEs with point- or set-valued initial conditions. It thus complements the established methods for enclosing reachable state sets of ordinary differential equations (ODEs), lifting their power to DDEs. It consequently covers the situations actually encountered in many modern control applications, where the feedback dynamics entails delays due to communication networks etc. and thus can reasonably be described by DDEs. Relaxing these DDEs to ODEs in verification may yield misleading results due to the impact of delays on system dynamics. To provide verification and reliable certificates for system properties, e.g., stability and safety properties, we have thus established a safe enclosure method for DDEs. Interval-based Taylor forms are used as a suitable data structure, facilitating to enclose a set of trajectories by parametric Taylor series with parameters in interval form. This data structure is used to iterate bounded degree interval-based Taylor overapproximations of the time-wise segments of the solution to a DDE. Given a DDE, we thereby identify the operator that computes the parameters of the Taylor overapproximation for the next temporal segment from the current one, and we employ constraint solving for automatically analyzing its properties. Based on such analysis by numeric constraint solving as implemented in the iSAT tool [11], we were able to obtain an automatic procedure able to provide stability certificates for a simple class of DDE.

For this introductory exposition of the method, we assumed that the system dynamics is represented as a DDE with a single, constant delay, i.e., is of the restricted form given by Eq. (2). Several dynamical systems can be modeled by DDE with a single constant delay as in biology [14, 26], optics [17], economics [39, 38], ecology [10]. In control applications, one may however want to combine delayed feedback, as imposed by communication networks, with immediate state feedback as suggested by ODE models of the plant dynamics derived from, e.g., Newtonian models. Such cases can be addressed by a layered combination of Taylor-model computation for ODEs, e.g. [28], with the ideas exposed herein. The pertinent algorithms are currently under development and will be exposed in future work. Beyond that, we want to extend the method to still more general kinds of DDEs, like DDEs with multiple different discrete delays (cf. Eq. (1)), DDEs with randomly distributed delay, or DDEs with time-dependent or more generally state-dependent delay [21]. Likewise, this work can (and will) be extended to facilitate the automatic verification and analysis for hybrid systems featuring delays, extending Egger's method for integrating safe ODE enclosures into a SAT modulo theory (SMT) solver [8, 9] from ODE enclosures to DDE enclosures. In this case, one will need to extend the enclosure methods for DDEs to a constraint propagator mutually narrowing intervals of pre- and post-states and integrate that propagator into the iSAT SMT solver as in [12].

References

1. Balakumar Balachandran, Tamás Kalmár-Nagy, and David E Gilsinn. *Delay Differential Equations.* Springer, 2009.

2. Richard Bellman and Kenneth L. Cooke. Differential-difference equations. Technical Report R-374-PR, The RAND Corporation, Santa Monica, California, January 1963.
3. Martin Berz and Kyoko Makino. Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369, 1998.
4. Stephen Boyd, Laurent El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), 1994.
5. Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *CAV'13*, pages 258–263, 2013.
6. Alongkrit Chutinan and Bruce H. Krogh. Computing polyhedral approximations to flow pipes for dynamic systems. In *Proc. of the 37th International Conference on Decision and Control (CDC'98)*, 1998.
7. Jamal Daafouz and Jacques Bernussou. Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & Control Letters*, 43(5):355 – 359, 2001.
8. Andreas Eggers, Martin Fränzle, and Christian Herde. SAT modulo ODE: A direct SAT approach to hybrid systems. In *Automated Technology for Verification and Analysis*, pages 171–185. Springer, 2008.
9. Andreas Eggers, Nacim Ramdani, Nedialko S Nedialkov, and Martin Fränzle. Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods. *Software & Systems Modeling*, pages 1–28, 2012.
10. Joaquim Fort and Vicenç Méndez. Time-delayed theory of the neolithic transition in europe. *Physical review letters*, 82(4):867, 1999.
11. Martin Fränzle, Christian Herde, Stefan Ratschan, Tobias Schubert, and Tino Teige. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation – Special Issue on SAT/CP Integration*, 1:209–236, 2007.
12. Martin Fränzle, Tino Teige, and Andreas Eggers. Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *Journal of Logic and Algebraic Programming*, 79:436–466, 2010.
13. Antoine Girard. Reachability of uncertain linear systems using zonotopes. In *HSCC'05*, pages 291–305, 2005.
14. Leon Glass and Michael C Mackey. *From clocks to chaos: the rhythms of life.* Princeton University Press, 1988.
15. Grant B. Gustafson. Systems of differential equations. In *Manuscript for Course Eng Math 2250-1 Spring 2014*, chapter 11. Dpt. of Mathematics, University of Utah, 2014.
16. Thomas A. Henzinger, Peter W. Kopke, Anuj Puri, and Pravin Varaiya. What's decidable about hybrid automata? *Journal of Computer and System Sciences*, 57(1):94–124, 1998.
17. Kensuke Ikeda and Kenji Matsumoto. High-dimensional chaotic behavior in systems with time-delayed feedback. *Physica D*, 29(1-2):223–235, 1987.
18. Soonho Kong, Sicun Gao, Wei Chen, and Edmund Clarke. dReach: Delta-reachability analysis for hybrid systems. In *TACAS'15*, 2015.
19. Alexander B. Kurzhanski and Pravin Varaiya. Ellipsoidal techniques for hybrid dynamics: The reachability problem. In *New Directions and Applications in Control Theory*, volume 321 of *Lecture Notes in Control and Information Sciences*, pages 193–205. Springer-Verlag, 2005.

20. Gerardo Lafferriere, George J. Pappas, and Sergio Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231–253, 2001.
21. Muthusamy Lakshmanan and Dharmapuri Vijayan Senthilkumar. *Dynamics of nonlinear time-delay systems*. Springer Science & Business Media, 2011.
22. Colas Le Guernic and Antoine Girard. Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262, 2010.
23. Jiang Liu, Naijun Zhan, and Hengjun Zhao. Computing semi-algebraic invariants for polynomial dynamical systems. In *EMSOFT'11*, pages 97–106, New York, NY, USA, 2011. ACM.
24. Jiang Liu, Naijun Zhan, and Hengjun Zhao. Automatically discovering relaxed Lyapunov functions for polynomial dynamical systems. *Mathematics in Computer Science*, 6(4):395–408, 2012.
25. Rudolf Lohner. *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben*. PhD thesis, Fakultät für Mathematik der Universität Karlsruhe, Karlsruhe, 1988.
26. Michael C Mackey, Leon Glass, et al. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.
27. Ramon E. Moore. Automatic local coordinate transformation to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations. In L. B. Ball, editor, *Error in Digital Computation*, volume II, pages 103–140. Wiley, New York, 1965.
28. Marcus Neher, K. R. Jackson, and N. S. Nedialkov. On Taylor model based integration of ODEs. *SIAM Journal on Numerical Analysis*, 45(1):236–262, 2007.
29. Jens Oehlerking. *Decomposition of Stability Proofs for Hybrid Systems*. Doctoral dissertation, Carl von Ossietzky Universität Oldenburg, Department of Computing Science, 2011.
30. André Platzer. Differential-algebraic dynamic logic for differential-algebraic programs. *J. Log. and Comput.*, 20(1):309–352, February 2010.
31. André Platzer and Edmund M. Clarke. Computing differential invariants of hybrid systems as fixedpoints. In Aarti Gupta and Sharad Malik, editors, *CAV'08*, volume 5123 of *LNCS*, pages 176–189. Springer Berlin Heidelberg, 2008.
32. Stephen Prajna, Ali Jadbabaie, and George J. Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428, 2007.
33. Stefan Ratschan and Zhikun She. Safety verification of hybrid systems by constraint propagation based abstraction refinement. In *HSCC 2005*, volume 3414 of *Lecture Notes in Computer Science*, pages 573–589, 2005.
34. Stefan Ratschan and Zhikun She. Providing a basin of attraction to a target region by computation of Lyapunov-like functions. In *Computational Cybernetics, 2006. ICCC 2006. IEEE International Conference on*, pages 1–5, 2006.
35. Jiri Rohn and Jana Kreslová. Linear interval inequalities. *Linear and Multilinear Algebra*, 38(1-2):79–82, 1994.
36. Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Constructing invariants for hybrid systems. In Rajeev Alur and George J. Pappas, editors, *HSCC'04*, volume 2993 of *LNCS*, pages 539–554. Springer Berlin Heidelberg, 2004.
37. Ole Stauning. *Automatic Validation of Numerical Solutions*. PhD thesis, Technical University of Denmark, Lyngby, 1997.
38. Marek Szydłowski and Adam Krawiec. The stability problem in the kaldor–kalecki business cycle model. *Chaos, Solitons & Fractals*, 25(2):299–305, 2005.

39. Marek Szydłowski, Adam Krawiec, and Janusz Tobołą. Nonlinear oscillations in business cycle model with time lags. *Chaos, Solitons & Fractals*, 12(3):505–517, 2001.
40. Xianhua Tang and Xinfu Zou. Global attractivity in a predator-prey system with pure delays. *Proc. Edinburgh Math. Soc.*, 51:495–508, 2008.
41. Liang Zou, Jidong Lv, Shuling Wang, Naijun Zhan, Tao Tang, Lei Yuan, and Yu Liu. Verifying Chinese train control system under a combined scenario by theorem proving. In *VSTTE'13*, LNCS 8164, pages 262–280, 2013.