Carl von Ossietzky Universität Oldenburg

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

Handling Delay Differential Equations in Automatic Verification

Dissertation zur Erlangung des Grades
eines Doktors der Ingenieurwissenschaften

vorgelegt von

M.Sc. Peter Nazier Mosaad

Gutachter:
Prof. Dr. Martin Fränzle
PD Dr. Markus Neher (Karlsruher Institut für Technologie (KIT))

Tag der mündlichen Prüfung: 21. Oktober 2019

I would like to dedicate this dissertation to my loving wife, Martina, my adorable daughter, Jolicia, and my beloved parents.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

M.Sc. Peter Nazier Mosaad

June 2020

Acknowledgements

This dissertation is the culmination of my journey of PhD which was just like climbing a high peak step by step accompanied with hardship and frustration, but also with encouragement, support, and trust. At this moment of accomplishment I am greatly indebted to my research guide, Prof. Dr. Martin Fränzle, who accepted me as his PhD student and offered me his mentorship, fatherly love and care. Without his guidance and involvement, this research work would not have been possible. With his guidance, I successfully overcame many research challenges and learned a lot. His passion and absolute devotion to scientific research has always inspired me to do more. For all these, I sincerely thank him from bottom of my heart and will be truly indebted to him throughout my life time.

This dissertation is supported by the German Research Foundation through the Research Training Group DFG-GRK 1765: "System Correctness under Adverse Conditions" (SCARE, scare.uni-oldenburg.de). It is with immense gratitude and profound thanks that I acknowledge the support of the DFG for giving me the opportunity to make contributions to the state of the art. My sincere thanks to Prof. Dr. Ernst-Rüdiger Olderog for his valuable guidance and support throughout SCARE research project work. I greatly appreciate and acknowledge the support received from the supervisors of SCARE research project. I am extremely thankful to all my colleagues from SCARE, the Hybrid Systems research group, OFFIS, and the university of Oldenburg for their support, encouragement, and love when it was most required. Big thanks must definitely go to Dr. Bai Xue for his cooperation and support throughout the research tenure. My sincere thanks to Prof. Dr. Markus Neher for accepting to co-referee my dissertation. Special thanks are extended to Prof. Dr. Oliver Kramer and Dr. Jörg Bremer for serving on my thesis committee, and for the friendly atmosphere they brought to my thesis defense.

I acknowledge the people who mean a lot to me, my parents, for showing faith in me and giving me the chance to study and find my way. Also, I express my thanks to my brothers Osama, Remon, Ramy, and sisters in law Amaal, Mariam, Ania for their love and moral support. My heartfelt thanks go to my father in law, mother in law for their love and valuable prayers. Many thanks to my sister in law, Youstina, and my brother in law, Mina, for their moral support and their valuable prayers. Also, it is my fortune to gratefully acknowledge the support of all my friends for being beside me during the happy and hard moments to push me and motivate me.

I owe thanks to a very special person, my wife Martina, for her continued and unfailing love during my pursuit of PhD degree that made the completion of this dissertation possible. I appreciate my little girl, Jolicia, for abiding my ignorance and the patience she showed during writing my dissertation and the preparation for disputation. I consider myself the luckiest in the world to have both of you in my life.

At this moment of accomplishment, my PhD journey almost flashed before my eyes, I can look back and see God's unconditional love, His mercy, and His hands in all things I accomplished. First and above all, I thank the Lord Almighty to whom I owe my very existence for providing me this opportunity and granting me the capability to proceed successfully. I am grateful for His provision of joys, challenges, and grace for growth that have been bestowed upon me during my PhD journey, and indeed, throughout my life: "The God of heaven will make us prosper, and we His servants will arise and build." (English Standard Version, Nehemiah 2:20).

Abstract

Ordinary differential equations (ODEs) are traditionally used for modeling the continuous behavior within continuous- or hybrid-state feedback control systems. In practice, delay is introduced into the feedback loop if components are spatially or logically distributed. Such delays may significantly alter the system dynamics and unmodeled delays in a control loop consequently have the potential to invalidate any stability and safety certificate obtained on the delay-free model. An appropriate generalization of ODE able to model the delay within the framework of differential equations is delay differential equations (DDEs), as studied by Bellman and Cooke in their seminal work. Beyond distributed control, DDEs play an important role in the modeling of many processes with time delays, both natural and manmade processes, in biology, physics, economics, engineering, etc. This induces an interest especially in the area of modeling embedded control and formal methods for its verification.

In this thesis, we focus on automatic safety analysis and verification for continuous systems featuring delays, extending the techniques of safely enclosing set-based initial value problem of ODEs to DDEs. First, as a result of collaborative work, we expose interval-based Taylor over-approximation method to enclose the solution of a simple class of DDE for stability and safety verification. Then, we explore different means of computing safe over- and under-approximations of reachable sets for DDEs by lifting the set-boundary reachability analysis based method of ODEs to a class of DDEs. Furthermore, for the sake of extending the safety properties by involving a number of critical properties such as timing requirements and bounded response rather than just invariance properties, we propose an approach —extending interval-based Taylor over-approximation method for a class of DDEs— to verify arbitrary time-bounded metric interval temporal logic (MITL) formulae, including nesting of modalities.

Zusammenfassung

Zur Modellierung des kontinuierlichen Verhaltens von Regelkreisen mit kontinuierlichen oder hybriden Zuständen werden traditionell gewöhnliche Differentialgleichungen (ODEs) verwendet. Sind die Komponenten des Systems räumlich oder logisch verteilt, entstehen in der Praxis Verzögerungen in der Rückkopplungsschleife. Solche Verzögerungen können die Systemdynamik erheblich verändern. Folglich können Stabilitäts- und Sicherheitszertifikate ungültig werden, falls Verzögerungen im untersuchten Modell nicht berücksichtigt werden. Retardierte Differentialgleichungen (DDEs) sind, wie schon Bellman und Cooke zeigten, eine Verallgemeinerung von gewöhnliche Differentialgleichungen (ODEs), die es erlaubt, Verzögerungen zu modellieren. DDEs spielen über das Gebiet der verteilten Regelung hinaus eine wichtige Rolle bei der Modellierung vieler künstlicher oder natürlicher Prozesse, beispielsweise in Biologie, Physik, Wirtschaft oder Ingenieurswesen. Daraus erwächst ein besonderes Interesse auf dem Gebiet der Modellierung und der formalen Methoden zur Verifikation eingebetter Systeme.

Diese Arbeit fokussiert auf die automatische Sicherheitsanalyse und Verifikation kontinuierlicher Systeme mit Verzögerungen. Dabei werden die Techniken der sicheren Überapproximation von mengenbasierten Anfangswertproblemen von ODEs auf DDEs erweitert. Zunächst wird als das Ergebnis kollaborativer Arbeit die intervallbasierte Taylor-Überapproximation zur Annäherung von Lösungen einfacher DDE-Klassen zum Nachweis von Stabilitäts- und Sicherheitseigenschaften vorgestellt. Anschließend werden verschiedene Methoden der sicheren Über- und Unterapproximation von Erreichbarkeitsmengen für DDEs untersucht, indem die mengenrandbasierte Erreichbarkeitsanalyse für ODEs für eine Klasse von DDEs adaptiert wird. Um erweiterte sicherheitskritische Eigenschaften wie Anforderungen an Reaktionszeiten einbeziehen zu können, wird ein Ansatz vorgeschlagen, der die intervallbasierte Taylor-Überapproximation erweitert

sodass beliebige zeitbeschränkte Formeln der metrischen Intervall-Temporallogik (MITL) einschließlich geschachtelter Modalitäten verifiziert werden können.

(I thank Paul Kröger for the German translation)

Table of contents

List of figures

List of tables

Chapter 1

Introduction

"Indecision and delays are the parents of failure."

[attributed to George Canning[1], 1770–1827]

The rapidly increasing deployment of cyber-physical systems into diverse safety-critical application domains ranging from, a.o., transportation systems over chemical processes to health-care renders safety analysis and verification for these systems societally important. Cyber-physical systems are complex systems exhibiting both discrete and continuous behaviors, and are networked and/or distributed with possibly humans in the loop. They have applications in a wide-range of systems spanning communication, infrastructure, energy, health-care, manufacturing, military, robotics and transportation. Consequently, they are believed to be the systems of the future with an immense impact on the engineering systems technology at least comparable to the impact of the internet on the information systems. A suitable model for such cyber-physical systems is a hybrid system that comprises continuous and discrete aspects of a system. Hybrid systems are mathematical models that allow us to model, specify and verify several types of cyber-physical systems, including physical systems of the environment, logic-dynamic controllers, and even internet congestion. Ordinary differential equations (ODEs) are traditionally used to model the continuous behavior within continuous- or hybrid-state systems. Significant research has been invested to achieve automatic verification for ODEs (and their piecewise extensions to hybrid state).

[1] George Canning was a British politician in the early 19th century.

An ODE model formulation of a system, however, ignores the presence of any delay [92].

> *"Time delays are natural components of the dynamic processes of biology, ecology, physiology, economics, epidemiology and mechanics"[59] and "to ignore them is to ignore reality"[83].*

Delayed coupling between state variables of dynamic systems occurs in many domains. For instance, in population dynamics, where birth rate follows changes in population size with a delay related to reproductive age, spreading of infectious diseases, where delay is induced by the incubation period, exhaust gas control in internal combustion engines, where relevant sensors —like the λ probe— are located downstream the exhaust system such that gas transport induces a delay between the controlled combustion processes and sensing their effect, or networked control systems through the communication networks if the components are spatially or logically distributed, to name just a few. Obviously, most examples feature feedback dynamics and the presence of feedback delays reduces controllability due to the impossibility of immediate reaction and enhances likelihood of transient overshoot or even oscillation in the feedback system. In practice, the introduction of delays into a feedback system may reduce stabilization rates of or even destabilize an otherwise stable system, it may provoke overshoot and drive the system to otherwise unreachable states, it is likely to stretch dwell times, and it may induce residual error that never cancels. As this implies that safety or stability certificates obtained on idealized, delay-free models of systems prone to delayed coupling may be erratic, automated methods for system verification ought to address models of system dynamics reflecting delays, rendering verification tools only addressing ODEs and their derived models —like hybrid automata— vastly insufficient. An appropriate generalization of ODE able to model delays within the framework of differential equations is provided by delay differential equations (DDEs), as studied by Bellman and Cooke in their seminal work [13].

> *"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."* [13, p. iii]

DDEs play an important role in the modeling of natural or artificial processes with time delays in biology, physics, economics, engineering, etc. As a consequence, attention has gone to developing tools permitting their mechanical analysis. Generalizing techniques developed for ODE to DDE, however, is not as straightforward as it may seem at first glance. The reason is that DDEs are in some respect much more complex objects than ODEs: the future evolution of a DDE is no longer governed by the current state instant only, but depends on a chunk of its past trajectory such that introducing a delay immediately renders a system with finite-dimensional state into an infinite-dimensional dynamical system. In other words, DDEs belong to the class of systems with functional state, i.e., the time derivatives at the current time depend on the solution and possibly its derivatives at previous times as well. While DDEs describing system dynamics as a function

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = f(\vec{x}(t),\vec{x}(t-\tau_1),\ldots,\vec{x}(t-\tau_n)), \text{ with } \tau_n > \ldots > \tau_1 > 0, \tag{1.1}$$

of past system states have long been suggested as an adequate means of modeling delayed feedback systems [13], their tool support is still not at the level of ODE. Although the tool support of DDEs has benefited a great deal from the advances made in ODE during the past several years, the state-of-the-art for handling DDE, especially in formal verification, is an active area of research.

For instance, some methods are developed for solving DDEs numerically such as the 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, well-established and still being subject of active research, techniques for safely enclosing set-based initial value problems of ODEs, be it safe interval enclosures [102, 141, 91], Taylor models [17, 110], or flow-pipe approximations based on polyhedra [29], zonotopes [55], ellipsoids [85], or support functions [88], are thus urgently needed for DDEs. As in the ODE case, such techniques would safely (and preferably tightly) over-approximate the set of states reachable at any given time point from the set of initial values. On the other hand, techniques for computing under-approximations of ODEs (e.g., [77, 152, 81, 62, 28, 157]) are also needed to

be lifted to DDEs. Such techniques are incorporated into a variety of applications in engineering domains, e.g., for detecting falsification of safety properties by finding counterexamples.

In this thesis we will expose an interval-based Taylor over-approximation method, as collaboratively proposed in [162], to enclose the solution of a simple class of DDE for stability and safety verification. Then, we will explore different means of computing safe over- and under-approximations of reachable sets for DDEs by lifting the reachability analysis method based on set-boundary of ODEs, discussed in [155, 157], to a class of DDEs. Furthermore, for the sake of extending the safety properties by involving a number of critical properties such as timing requirements and bounded response rather than just invariance properties, we will extend interval-based Taylor over-approximation method for a class of DDEs such that it can verify arbitrary time-bounded metric interval temporal logic (MITL) formulae [4].

1.1 Delay Differential Equations

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. Differential equations can be divided into two types: ordinary differential equations (ODEs) having a finite dimensional state vector and partial differential equation (PDEs), i.e., infinite dimensional. Over the past decades, ODEs are widely used in the research field to model the continuous behavior of many processes especially in automatic verification purposes. Thus, a plethora of techniques are developed for ODEs to be algorithmically analyzable by safely enclosing set-based initial value problems of ODEs, e.g., [91, 110, 29, 55, 85, 88]. Due to increasing expectations on the accuracy of predicting, the engineers need their models to behave like the real processes by considering, e.g., the delay (also known as aftereffect phenomena) in the framework of differential equation. Many processes include delay (or aftereffect phenomena) in their inner dynamics [13, 39]. Besides, the delays are inevitably introduced in the feedback control loop by including the time taken for a signal to travel to the controlled object [83]. Finally, besides actual delays, time lags are frequently used to simplify very high dimensional models without delays by

lower dimensional models with delays [112], as extensively used in process control industry (see [79]). This way, formulation as a functional differential equation (FDE) (or differential equation with deviating arguments), which includes all delay differential equations (DDEs), enables both the current and all previous values of a function and/or its derivatives to be considered when determining the future behavior of a system. Based on this, it often leads to an improved model of a process since 'an increase in the complexity of the mathematical models can lead to a better quantitative consistency with real data', but at cost [92, 8]. Note that the modeler's decision about the choice of model formulation is influenced by the size of the delay relative to the underlying time-scales [7]. Those systems, which their model based on a FDE are more appropriate than the models based on an ODE, can usually be referred to as "problems with memory" [92].

DDEs (time-delay systems or system with aftereffect or dead-time) are a type of differential equations which belong to the class of system with functional state, i.e., PDEs which are infinite dimensional as opposed to ODEs. DDEs have become so popular as infinite-dimensional models in the very complex area of PDEs and the interest for DDEs keeps on growing in the research field especially in control engineering [83], where the control of delay systems presents many stimulating challenges for further research. In this vein, one may think that the simplest approach would be in replacing such infinite dimensional systems by some finite-dimensional approximations ignoring the effects which are adequately represented by DDEs. Unfortunately, it is not a general alternative: in the best case (e.g., constant and known delays), it leads to the same degree of complexity in the control design, and in worst cases (e.g., time-varying delays), it is potentially disastrous in terms of safety, stability and oscillations. On the other hand, several studies have shown that voluntary introduction of delays can also benefit the control [134]. One may also think that a reasonably small delay does not affect the solution of a linear 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 simple example, taken from [162], demonstrates the difference between DDE and their related ODE obtained by neglecting delays. Furthermore, in Section 1.1.1, we show that small delay may have significant effect.

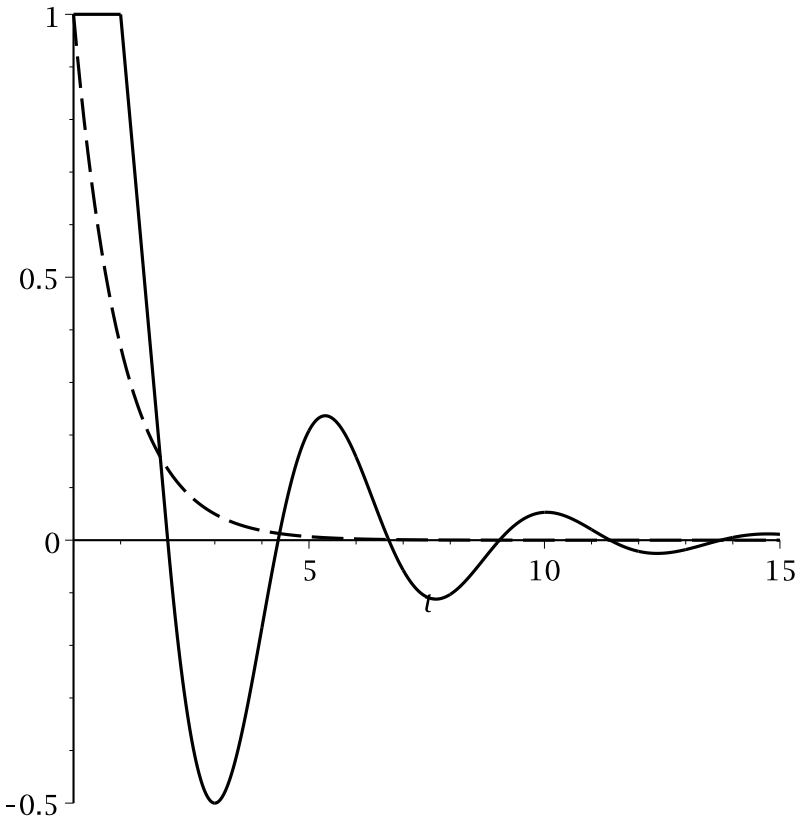

Fig. 1.1 Solutions to the ODE $\dot{x} = -x$ (dashed graph) and the related DDE $\dot{x}(t) = -x(t-1)$ (solid line), both on similar initial conditions $x(0) = 1$ and $x([0,1]) \equiv 1$, respectively.

In Fig. 1.1, 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 $x([0,1]) \equiv 1$. Fig. 1.1 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 obvious that the difference between the ODE and the DDE is substantial and necessitates analyzing the behavior of the DDE.

1.1.1 Small Delays Have Significant Effect

As mentioned, due to societal safety requirements, engineers are required to improve the dynamic performance of their models by considering, e.g., the delay in the framework of differential equation. The presence of delayed dynamics may invalidate any stability and safety certificate obtained on the related delay-free model, as delays may significantly alter the overall shape of the system dynamics. This situation is illustrated through the following simple example taken from [83], where arbitrarily small delays have significant effect on state dynamics.

Example 1.1.1 (taken from [83]). *The solution of the ODE*

$$\dot{x}(t) + 2\dot{x}(t) = -x(t) \tag{1.2}$$

is asymptotically stable, converging to the equilibrium point $x = 0$ *from any initial state. However, the solution of its corresponding DDE*

$$\dot{x}(t) + 2\dot{x}(t-\tau) = -x(t) \tag{1.3}$$

is unstable for any positive delay τ.

Therefore, taking time-delay terms into account to either verify or falsify properties of systems by performing safe automatic analysis is not just desirable, but ought to be imperative for systems that are more accurately modeled by DDEs, especially in safety-critical applications. Also, the delay can act as a stabiliser or a destabiliser of ODEs models [92, 59, 12, 19]. The following simple example, taken from [92, 10], illustrates this situation.

Example 1.1.2 (taken from [92, 10]). *Consider the equation*

$$\dot{x}(t) = \lambda x(t) + \mu x(t-\tau), \quad \tau \geq 0. \tag{1.4}$$

The zero solution of Eq.(1.4) is asymptotically stable if $\lambda + |\mu| \leq 0$. In case of $\mu = 0$, an ODE is obtained whose zero solution is asymptotically stable if $\lambda < 0$. On the other hand, positive values of λ, with corresponding negative values of μ, give rise to asymptotic stability of Eq.(1.4). Thus, the delay term τ can stabilise an unstable ODE. Alternatively, if the delay term $\tau = 0$, again leading to an ODE which

is asymptotically stable if and only if $\lambda + \mu < 0$. However, if the delay term $\tau > 0$, $\lambda + \mu < 0$ is insufficient to guarantee stability, and thus the introduction of a delay term τ can destabilise a stable solution.

Given the omnipresence of the effects of delays in, a.o., modern control schemes, one might thus expect tools permitting automatic safety analysis of such delayed systems to abound. Unfortunately that is not the case; their validated tool support still seems to be in its infancy and thus provides an open area of research. The reason for their current lack is that DDEs are in some respect much more complex objects than ODEs. Fortunately, there are many similarities between the theory of ODEs and that of DDEs, and thus analytical methods for ODEs have been extended to DDEs when possible [92]. On the other hand, their differences have necessitated new approaches. Similarly, for DDEs to be algorithmically analyzable in automatic verification purposes, developed techniques for ODEs are required to be lifted to DDEs.

1.1.2 Important Differences Between ODEs and DDEs

"The future depends on what we do in the present."

[attributed to Mahatma Gandhi[2], 1869–1948]

It has been well said by Mr. Gandhi, however, it is not completely true when it comes to delay differential equations (DDEs) because we cannot describe the future based on just the current situation but we need to always look back a bit into history, i.e., the history function is needed. In other words, the effect of any changes to a system of DDEs is not instantaneous and the past history is taken into account. 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.(1.1), transformed copies of the initial segment of duration τ_n will generally be found in higher-order derivatives of $x(t)$ even after arbitrarily long time. Such differences, for example, lead to a significant challenge in order to generalize

[2]Mohandas Karamchand Gandhi was an Indian activist who was the leader of the Indian independence movement against British rule.

the developed techniques for ODEs to handle DDEs. Table 1.1, taken from [92], summarizes the important differences between ODEs and DDEs, e.g., the need for an initial (history) function and the infinite dimensionality of a DDE.

ODE model	DDE model
assumes that the effect of any changes to the system is instantaneous (a principle of causality).	*assumes* that the effect of any changes to the system is not instantaneous, i.e., past history is taken into account.
generates a system that is finite dimensional.	*generates* a system that is infinite dimensional.
needs an initial value (to determine a unique solution).	*needs* an initial (history) function (to determine a unique solution).
	enables a more accurate reflection of the system being modeled, however, safe automatic analysis of the solution is much less developed.

Table 1.1 Important Differences between ODEs and DDEs [92].

Unfortunately, the presence of an initial function, yet not an initial value as in ODE case, has several unwelcome consequences, e.g., unlike ordinary equations, there is no longer injectivity[3] between the set of initial data and the set of solutions; the solutions corresponding to different initial function data can intersect [92, 8, 19]. This fact, for example, leads to a significant challenge in order to generalize verification techniques of ODE to DDE (this point will be discussed in details in Chapter 3). For more consequences, the interested reader is referred to [92] and the references therein.

Note that the dynamical structure exhibited by DDEs is richer than that of ODEs. Also, DDE model is 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 finite-dimensional 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] injective function or one-to-one function preserves distinctness: every element of the function's codomain is the image of at most one element of its domain.

1.2 Related Work

For automatic (safety) verification, especially in case of autonomous systems [49], formal methods have been developed as a rigorous solution to address the incomplete verification of design provided by simulation, where confidence obtained via simulation is based primarily on intuition and experience [37]. Formal methods, however, increase the confidence by including techniques rooted in logic, mathematics, and computer science. Roughly, a specification is constructed and a formal reasoning is used to show whether the model under analysis satisfies such specification. In general, there are two broad classes of methods to formally specify and verify dynamical systems. One class of methods, which is out of the scope of this thesis, is based on logical proof, where such methods represent the behavior of the systems in some type of logic, and then deductive reasoning is used to construct a proof [121, 65, 66, 97, 11, 74, 23, 34]. For example, automated theorem proving is a set of deductive techniques that are purely automatic, e.g., [133, 73, 35, 41], contrary to interactive theorem proving [100] with a manual complete proof that is constructed by hand (potentially with help from a proof assistant) using a base set of axioms and rules, e.g., [118, 116, 60, 16]. Also, an essential difference between interactive and automated theorem provers is that automated theorem provers are confined to a much smaller fragment of logical theories. This consequently affects the types of conjectures that can be analyzed [37]. The other class of methods is state-space exploration, where such methods are mainly based on representing the behavior as a transition system and then verifying temporal properties over the resulting state machine. Model checking [31] is the popular example in this class of methods where an exhaustive search of the state space is undertaken. It is a preferred method, though memory constraints pose a significant problem for model checking [32], because it is fully automatic. Notice that model checking can also be combined with automated theorem proving and benefit from the power of this approach to verification. In this thesis, we focus on such a class of methods.

Formally, the safety verification problem can often be reduced to a problem of deciding whether the system of interest, starting from legal initial states, may in its evolution touch a specified set of unsafe states. This way, a natural approach to the automatic verification is state-space exploration aiming at computing the reachable state space. In this context, we will recap some state-of-the-art verification techniques used for continuous/hybrid dynamics given as ODEs that are urgently needed to be lifted

to DDEs. Then, we will recap some state-of-the-art verification techniques for DDEs, where the main ideas and the relations to our work are introduced.

1.2.1 Automatic Verification Techniques for ODEs

As outlined above, ODEs are traditionally used to model a vast variety of dynamical systems. The safety verification problems of primary interest are invariance properties concerning the dynamically reachable states. Invariance properties are a prototypical safety property and computing the reachable state space is the natural approach to their automatic verification. Therefore, reachability analysis plays a fundamental role in addressing safety verification challenges. While there is no closed form for the solution of ODE in many cases, one may resort to numerical approximations, e.g., given by one-step methods like the Euler method or the more general family of Runge-Kutta methods, which approximate the solution in several discrete steps in time. Numerical simulation, however, introduces imprecision stemming from, e.g., rounding in floating-point arithmetic or truncation errors in the integration of differential equations [42]. Thus, there are no guarantees on the quality of the approximation, as they are numerically approximate only with a finite number of simulation runs[4], in the sense that they fail to provide reliable certificate of system properties, especially in safety-critical applications. This way, methods have been developed that are supposed to compute enclosures of the solution. In a sense, they do not compute the solution with approximate values, but with sets enclosing the solution [76].

Unfortunately, only very few families of restrictive linear dynamic systems feature a decidable state reachability problem [86, 69]. A more generally applicable option is to compute over-approximations 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. Computing over-approximations of the reachable sets of continuous dynamics (discrete as well) is a fundamental for formally verifying given safety properties that is used to counteract imprecise approximations made, e.g., in traditional numerical simulation tools. Much progress has been made towards reachable set over-approximations for linear

[4] Since it is impossible to cover all possible initial conditions and all possible inputs with a finite number of simulation runs, no matter how close samples are chosen, there is probability that a value in between (known as corner case) causes undesired behavior.

as well as nonlinear continuous/hybrid dynamics featuring ODEs. Among the many abstraction techniques proposed for over-approximating reachable sets of continuous dynamics given as ODEs is use of interval arithmetic [102]. Interval arithmetic suffers from the wrapping effect, i.e., large over-approximations when enclosing rotated boxes in a box, and the dependency problem, i.e., it cannot track dependencies between variables. A family of algorithms overcomes the wrapping effect with QR-decomposition is well-studied based on Taylor series expansions and computing with interval arithmetic (surveyed, e.g., by Nedialkov in [107]), implemented in tools like `AWA` [91], `ADIODES` [141], `VNODE` [109], and `VNODE-LP` [108]. A different abstraction technique based on Taylor models is used to relieve the wrapping effect and the dependency problem as studied by Berz and Makino [17, 110]. Also, flow-pipe approximations based on polyhedra [29], zonotopes [55], ellipsoids [85], or support functions [88] are used for over-approximating the reachable sets of the continuous dynamics given as ODEs. Other than the above methods, abstraction based on discovering invariants technique is also used, e.g., [137, 129, 123, 89], to prove safety of continuous and hybrid systems featuring ODEs. Unlike the computation of flow-pipe over-approximation, invariant computation technique derives a system of constraints such that all states of the system satisfy them. Then, if the derived constraints are inconsistent with the specification of the unsafe set, it means that the system is safe. Furthermore, verifying delayless dynamical systems, in particular ODE, using numerical simulation has been well-studied, e.g., [38, 40, 57, 105], where the concepts based on sensitivity information provided by discrepancy functions or simulation functions, respectively, have been presented to bloat the traces obtained from simulations to "trajectory tubes" over-approximating time-bounded reach sets. While the first settings resorted to user-supplied sensitivity information, Fan and Mitra in [48] proposed an algorithm for automatically computing piecewise exponential discrepancy functions, i.e., the key ingredient is an on-the-fly discrepancy computation.

While over-approximations serve as states that "may" be reachable, under-approximations represent states that "must" be reachable. The case of computing under-approximations of the reachable sets for linear as well as nonlinear continuous dynamics featuring ODEs, though less attention has been given to such case, is also considered in the literature, cf. e.g., [62, 81, 152, 28, 157]. Computing under-approximations of the reachable sets are incorporated into a variety of applications in engineering domains. They can be used for, e.g., designing robust artificial pancreas

[157, 154, 132], computing under-approximations of backward reachable sets helps to find a set of feasible states such that every trajectory originating from it will definitely enter a specified region (e.g., normal blood glucose ranges) at a specified time instant, helps to judge the quality of related over-approximations by comparing the states that "may" be reachable with the states that "must" be reachable, proving attractive properties by checking if all the trajectories originating from them will stay in them forever and eventually enter some specified desired sets [131], detecting falsification of safety properties by finding counterexamples [119]. Furthermore, computing under- and over-approximations of reachable sets may provide an indication of the precision of an estimate of the exact reachability region [157, 70]. Also, it can help us prove desired reach-while-avoid properties that are common in many control systems: the system must reach a specified target set of states, while avoiding a set of unsafe states [28, 155].

For automatic verification and analysis of continuous dynamics featuring ODEs, there are several mature bounded model checkers available for continuous and hybrid systems, like `iSAT-ODE` [43], `Flow*` [27], `dReach` [80], to new just a few. In case of unbounded automatic verification, theorem provers for ODE dynamics and hybrid systems are also available, e.g., `KeYmaera` [122] or `HHL Prover` [163].

Confining safety properties to a set of unsafe states (invariance properties) considerably restricts the ability of designers to adequately express the desired safe behavior of the system that may involve a number of critical properties such as timing requirements and bounded response. Metric temporal logic (MTL), introduced by Koymans [82], is popular formalism for expressing such properties as a real-time extension of linear temporal logic (LTL) [96] to specify real-time properties. Then, Alur *et al.* in [4] introduced metric interval temporal logic (MITL) to address the undecidability problem of MTL by relaxing the punctuality of the temporal operators. The bounded-time verification or falsification of such properties has been studied for continuous/hybrid systems given as ODEs in [46, 47, 136, 120, 95]. Besides, safety verification is complemented by automatic procedures for providing certificates of stability. The great majority of such methods are based on Lyapunov functions [21] or piecewise Lyapunov functions [113]. Interestingly, such procedures can only be complete for restricted, mostly linear cases, though incomplete extensions to rather general classes exist, e.g., [90].

1.2.2 Automatic Verification Techniques for DDEs

Delay differential equations (DDEs), as reviewed in [160, 61], were initially introduced in the 18th century by Laplace and Condorcet. Some preliminary work concerning stability of systems described by equations of this type was carried out in 1942 by Pontryagin. Later, important works have been written by Bellman and Cooke in their seminal work in 1963, Smith in 1957, Pinney in 1958, Halanay in 1966, El'sgol'ts and Norkin in 1971, Myshkis in 1972, Hale in 1977, Yanushevski in 1978, and Marshal in 1979. For a more detailed review, the interested reader is referred to [160, 61], and the references therein.

To our knowledge, there are some methods that have been hypothesized to solve DDE. Among these methods are the method of steps, the classical Laplace Transform method, and a Least Square method. Such methods are of interest for the reason that they are widely used in studies involving the solution of DDEs and have the potential to be automated using tools provided by computer algebra, like `Maple` [68]. First, the method of steps is an elementary method that can be used to analytically solve very simple linear DDEs. This method, however, suffers from being too tedious and thus it is usually discarded for many cases (only in few cases the tedium might be removed using computer algebra). Second, the classical method of Laplace Transforms are confined to solve simple linear problems with constant delays. This method usually leads to a non-harmonic Fourier series solution for linear problems with constant delays [68, 160]. Third, the Least Square method, which involves the Lambert W function and a numerical Least Square method, is used also to solve some simple linear DDEs. This method is studied by Corless as well as by Asl and Ulsoy, who developed a solution method that incorporates the Lambert W function into an approximation of the solution of a linear DDE with constant delays [159, 5, 160]. The aforementioned methods, unfortunately, fail to algorithmically analyze a large group of linear DDEs as well as nonlinear DDEs. For instance, it is intractable to calculate the reachable set of a large group of linear DDEs as well as nonlinear DDEs using the aforementioned methods. Also, the numerical simulation, which numerically approximates only the solution of DDE, e.g., Matlab's `dde23` and its successor `ddesd` [139], though being useful in system analysis, fails to provide reliable certificates of system properties.

Albeit there is a reasonable amount of literature on the theory and computational practice of DDEs, see for example [13, 61, 139] for pointers to introductions, surveys and tutorials, also addressing the question of how to manually verify stability of some linear DDEs (e.g., [71, 161]), even some nonlinear DDEs (e.g., [15]), or using numerical approaches (e.g., [98, 22]), fully automatic validated proof procedures for linear as well as nonlinear DDEs that can be used in automatic verification purposes such as safety verification are currently lacking and thus providing an open area of research. This induces an interest in fully automatic validated proof procedures for linear and nonlinear DDEs that can be used in automatic verification purposes, e.g., calculating over- and under-approximations of the reachable set of the DDE.

Unfortunately, few researchers have addressed the problem of automated analysis and (safety) verification of time-delay systems (i.e., DDEs). In the recent years, however, there has been growing interest in such problems driven by the demand for safety cases (in a broad sense) for high-dynamic-performance systems (i.e., modeled by DDEs). Prajna and Jadbabaie in [128] were among of the first to consider the safety verification of time-delay systems. In the introduction of their seminal paper [128], they have argued that most of the available analysis results in the field of time-delay systems are focused on stability, robustness, or input-output properties and not on safety or reachability (see, e.g., [63, 111]). They have extended the barrier certificate methodology for ordinary differential equations (ODEs) to the polynomial time-delay differential equations setting, in which the safety verification problem is formulated as a problem of solving sum-of-square programs. In [126], Pola *et al.* proposed an approach abstracting incrementally input-to-state stable (δ-ISS) nonlinear control systems with constant and known delays to finite-state symbolic models, and establish approximate bisimilarity between them. In [125], they extended the work in [126] to incrementally-input-delay-to-state stable (δ-IDSS) nonlinear control systems with time-varying and unknown delays, and proved that the original δ-IDSS nonlinear control systems and the corresponding symbolic models are alternating approximately bisimilar. The work in [72] presents a technique for simulation-based time-bounded invariant verification of nonlinear networked dynamical systems with delayed interconnections by computing bounds on the sensitivity of trajectories (or solutions) to changes in initial states and inputs of the system. A similar simulation method integrating error analysis of the numeric solving and the sensitivity-related state bloating algorithms was proposed in [24], as a result of collaborative work with Chen *et al.*, to obtain safe enclosures of

time-bounded reach sets for systems modeled by DDEs. These approaches, however, are not well suited to unbounded safety verification problems. In [162], as a result of collaborative work with Zou, Fränzle *et al.*, we have exposed interval-based Taylor over-approximation method to enclose the solution of a class of DDEs (i.e., with single constant delay) for stability and safety verification. This way, unbounded safety verification problem is presented by means of pursuing bounded model checking (BMC) for sufficiently many steps k_{depth} in case the DDE is stabilizing [162].

For the case of computing the under-approximations of reachable sets for DDEs, unfortunately, a recent review of the literature found that the state-of-the-art of techniques for DDEs is not at the level of ODEs. The work in [149] has considered finding outer bounds of forwards reachable sets and inner bounds of backwards reachable sets for linear systems with interval time-varying delays and unknown-but-bounded disturbances. Subsequently, computing the smallest box to bound all reachable sets of a class of nonlinear time-delay systems with bounded disturbances was considered in [106]. In [156], with Xue, Fränzle *et al.*, we have explored different means of computing safe over- and under-approximations of reachable sets for a class of DDEs. In this paper we have lifted the reachability analysis method based on set-boundary of ODEs, discussed in [155, 157] by Xue *et al.*, to a class of DDEs. In this context we have employed sensitivity analysis for bounding time-lags of DDEs to ensure the homeomorphism property. The practical implication is a rigorous method for selecting appropriate components (e.g., sensors) guaranteeing sufficiently low latency in the feedback loop with the ability of computing over- and under-approximation of the reachable sets for such class of DDEs. Our technique, in that paper, is believed to attract considerable interest as it can leverage many more techniques for ODE on DDE, like stability and bifurcation analysis. Reachability analysis thus constitutes just an example.

As reported above, reducing safety verification problems to only reachability problem (i.e., invariance properties) may restrict the ability to adequately express the desired safe behavior of the system. The formal specification of a wider range of safety properties, to the best of our knowledge, at the time of writing this dissertation, seems to be unsupported for DDEs. In [103] and its extended revised version [104] we tried to extend the safety properties by involving a number of critical properties such as timing requirements and bounded response rather than just invariance properties. With this

in mind, within the framework of interval-based Taylor over-approximation method introduced in [162], we have exploited metric interval temporal logic (MITL) with a continuous-time semantics evaluating signals over metric spaces, which is more well-suited to the case of DDEs rather than, e.g., dynamic logic (DL) [65, 66] as introduced by Platzer in [121] for continuous/hybrid dynamics given as ODEs. For instance, dynamic logics are inherently situation-based logics, yet DDE are not situation-based (rather history-based) in their dynamics. This is irrelevant in the combination DDE model vs. MITL specification, where the non-situational model generates a trajectory that can be interpreted by a situation-based logics. Yet w/o an independent model, as in DL, we cannot reasonably encode DDE dynamics. The problem is that we cannot describe the future based on just the current situation, but need to always look back a bit into history. This way, exploiting MITL with continuous-time semantics as requirements specification language within the framework of interval-based Taylor over-approximation method, we could solve time-bounded verification problems of temporal logic properties for a class of DDEs. For the unbounded verification problems that are aimed to be facilitated by interval-based Taylor over-approximation method, it is still an active area of research.

1.3 Contributions

This dissertation is written within the research project SCARE— System Correctness under Adverse Conditions —(DFG GRK 1765) which is funded by the DFG[5]. It is the culmination of almost three years of research with peer-reviewed publications to produce the research work in this dissertation. In fact, this dissertation is mainly based on the research work that has been peer-reviewed and published before the thesis is written. This fact is a well-accepted state of affairs in computer science and those who grade this dissertation are well aware of it.

In this thesis the author, together with other researchers who contributed mostly their thoughts in discussions, sometimes their guidance, and the writing of the papers, has studied the problem of handling delay differential equations (DDEs) in automatic verification purposes, which is in its infancy and thus provides an open area of research as reported above. The contributions of the author in this PhD work are as follows.

[5]Deutsche Forschungsgemeinschaft

- The author of this thesis was involved in the research work [162] together with Zou, Fränzle *et al.* to develop a safe enclosure method for a class of DDEs by using 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 [17, 110], we have employed interval Taylor series of fixed degree and moved higher-degree terms into the parametric uncertainty. By iterating bounded degree interval-based Taylor over-approximations of the time-wise segments of the solution to a DDE, we could identify and automatically analyze the operator that yields the parameters of the Taylor over-approximation for the next temporal segment from the current one. This way, by using constraint solving for analyzing the properties of this operator, we have obtained a procedure able to provide stability and safety certificates for a simple class of DDEs (i.e., with single constant delay).

  My contribution in this work: discussion on the main approach, discussion on the related work, performing some experiments, and co-writing the paper.

- The author *et al.* in [156] have developed a different means of computing safe over-approximations as well as under-approximations for a higher class in complexity of DDEs than the class of DDEs discussed in [162], where the right-hand side of characterizing the differential equation is a combination of ordinary differential equation (ODE) and DDE with single constant delay. We have lifted the set-boundary reachability analysis based method of ODEs, discussed in [155, 157] by Xue *et al.*, to a class of DDEs featuring a local homeomorphism property. This topological property facilitates construction of over- and under-approximations of their full reachable sets by performing reachability analysis on the boundaries of their initial sets, thereby permitting an efficient lifting of reach-set computation methods for ODEs to DDEs. Membership in this class of DDEs is determined by conducting sensitivity analysis of the solution mapping with respect to the initial states to impose a bound constraint on the time-lag term. We then generalize boundary-based reachability analysis to such DDEs.

  My contribution in this work: defining the research problem, central contribution to develop the main approach, and writing the paper.

- The author *et al.* in [103] and its extended revised version [104] has extended the safety properties within the framework of interval-based Taylor over-approximation method introduced in [162] by involving a number of critical properties such as timing requirements and bounded response rather than just invariance properties. We have exploited metric interval temporal logic (MITL) with a continuous-time semantics as requirements specification language, aiming at automatic safety verification of a simple class of DDEs (i.e., with single constant delay) against requirements expressed in a linear-time temporal logic. We have employed the over-approximation method based on interval Taylor series to enclose the solution of the DDE and thereby reduce the continuous-time verification problem for MITL formulae to a discrete-time problem over sequences of Taylor coefficients. We have encoded sufficient conditions for satisfaction of the MITL formulae as SMT formulae over polynomial arithmetic and used the iSAT3[6] SMT solver in its bounded model-checking mode for discharging the resulting proof obligations, thus proving satisfaction of time-bounded MITL specifications by the trajectories induced by a DDE. In contrast to our preliminary work in [162], we could verify arbitrary time-bounded MITL formulae, including nesting of modalities, rather than just invariance properties.

  My contribution in this work: defining the research problem, the fundamental contribution to develop the main approach, the experiments, writing the paper, the improvements on the main approach, and writing the journal extension.

1.4 Structure of the thesis

The next chapter, Chapter 2, revisits some preliminaries that we will use in the rest of this research work. This thesis is logically divided into three chapters according to the contributions mentioned above. Chapter 3 gives an introduction of using Taylor model methods in analysis and verification of (nonlinear) continuous systems and then we will review interval-based Taylor over-approximation method for a class of DDEs as studied in [162]. In Chapter 4, we will discuss in detail different means of computing safe over-approximations for a class of DDEs as well as under-approximations through lifting the reachability analysis method based on set-boundary of ODEs as discussed

[6]The iSAT3 implementation of the iSAT algorithm [52] is available at `http://projects.informatik.uni-freiburg.de/projects/isat3/`

in [156]. In Chapter 5, we will extend the safety properties within the framework of interval-based Taylor over-approximation method introduced in Chapter 3 to verify a class of DDEs against arbitrary time-bounded metric interval temporal logic (MITL) formulae as presented in [103, 104]. Finally, conclusions, limitations of the research, and some suggestions about promising points for further research will be drawn in Chapter 6.

Chapter 2

Preliminaries

"It is evident that one cannot say anything demonstrable about the problem before having resolved these preliminary questions, and yet we hardly possess the necessary information to solve some of them[1]*."*

[Georges Cuvier[2], 1769–1832]

In this chapter we gather together some general definitions and some preliminaries that we will use throughout this dissertation. We shall not go into details but cite references in which the interested reader can find further details. The first section, Section 2.1, is devoted to introducing delay differential equations (DDEs). The second section, Section 2.2, is devoted to the reachability problem where we give a brief overview of reach set computation and generally state the definitions of over- and under-approximation of the reachable set. Section 2.3 looks at expressing specifications on the desired temporal evolution to the system under investigation. We introduce, in Section 2.3, a brief overview of temporal logics, especially linear temporal logic (LTL) that will be useful for presenting our method in Chapter 5.

[1] As stated in 1796 before the National Institute of Sciences and Arts in Paris, concerning fossil elephants.

[2] Baron Georges Léopold Chrétien Frédéric Dagobert Cuvier (August 23, 1769 – May 13, 1832) was a French naturalist and zoologist.

2.1 Introduction to Delay Differential Equations

Delay differential equations (DDEs) differ from ordinary differential equations in that they belong to the class of systems with functional state, i.e., the derivative at any time depends on the solution (and in the case of neutral equations on the derivative) at prior times. The DDE with constant delays has the form

$$\dot{x}(t) = f(t, x(t), x(t-\tau_1), \ldots, x(t-\tau_n)), \tag{2.1}$$

where the time delays (lags) $\tau_1, \ldots, \tau_n$ are positive constants. More generally, in population dynamics and epidemic problems the delay time has also been represented as a function of the state variable itself, i.e., DDE with state-dependent delay $\tau_i = \tau_i(t, x(t))$ [87].

Systems of DDEs have central importance in many areas of science and particularly in the biological sciences (e.g., population dynamics and epidemiology) [147]. In [9], the reader is referred to some references for several application areas. Furthermore, DDEs arise naturally as models of, e.g., networked control systems, where the communication delay in the feedback loop cannot always be ignored [162]. Here, we revisit the basic property of DDEs which is the initial history function as discussed in [147, 139], and the references therein. This property highlights the main obstacle to lifting the power of established verification methods for ODEs to much more complex objects as DDEs, i.e., the main aim of this dissertation as stated in the Introduction (Chapter 1).

2.1.1 Initial History Function

Unlike ODE, a system of DDEs assumes that the effect of any changes to the system is not instantaneous, i.e., past history is taken into account [92]. Hence, an initial value problem (IVP) requires additional information than an analogous problem for a system of ODEs. For an ordinary differential system, a unique solution is determined by an initial point in Euclidean space at an initial time t_0 [50]. Since the derivative in (2.1) depends on the solution at the prior times $t - \tau_i$, it is necessary to provide an initial history function that conveys the value of the solution before the initial time t_0 [147]. The future evolution of a DDE is no longer governed by the current

state instant only, but depends on a chunk of its past trajectory. This generates a system that is infinite-dimensional. Therefore, a system of DDEs cannot be modeled by finite-dimensional hybrid automata, and consequently cannot be analyzed by the corresponding verification tools.

In most instances, the presence of an initial function may cause a jump derivative discontinuity at the initial time. In other words, the DDE and the initial history are incompatible: for some derivative order, usually the first, the left and right derivatives are not equal [147]. In the area concerned with numerical analysis, numerical methods for both ODEs and DDEs are intended for problems with solutions that have several continuous derivatives [139]. Thus, discontinuities in low-order derivatives need special attention. Due to the nature of DDEs, though such discontinuities are not unusual for ODEs, this is a much more serious matter. One reason why discontinuities are much more serious for DDEs is that they are almost always present for DDEs: generally there is a discontinuity in the first derivative of the solution at the initial point. There can also be discontinuities at times both before and after the initial point. In some problems, there might be histories with discontinuities in low-order derivatives. Another reason is that such derivative discontinuities are propagated in time. The detection and location of derivative discontinuities is a central issue in the design of robust solvers for solving DDEs numerically [153]. Some solvers are available to numerically find approximate solutions to DDEs, e.g., `dde23`, `ddesd` [139]. Several of these solvers use explicit Runge-Kutta methods with a suitable interpolation method, e.g., Hermite interpolation, to integrate systems of DDEs. A discussion of numerical methods for DDEs falls outside the scope of this dissertation; the interested reader is referred to [139], and the references therein.

The numerical simulation is extremely useful in system analysis in order to study the behavior of systems whose mathematical models are too complex to provide analytical solutions, as in most nonlinear systems. Several studies, for example, have used simulation methods to detect unsafe behaviors of many systems [25, 56]. On the other hand, numerical simulation methods fail to provide reliable certificates of system properties, as it is numerically approximate only. Another way to reliably verify, e.g., a safety property, on a system is formal verification, which is our concern in this dissertation. In formal verification, the system is usually defined by a mathematical model, e.g., DDE model, and we try to prove that no behavior of the model violates the given safety property. For a prototypical safety property that is defined by a set

of unsafe states, we compute all reachable states of the model, if no unsafe state in included then the system is safe. This natural approach to the automatic formal verification is also called reachability analysis [25, 36].

2.2 Reachability Problem

Reachability is a crucial problem that appears in several different areas: discrete and continuous systems, time critical systems, hybrid systems, Petri nets, probabilistic systems, open systems modeled as games, to name just a few [36]. A reachability problem consists in checking whether a given set of target states, e.g., unsafe states, can be reached starting from a fixed set of initial states. In this context, reachability analysis plays a fundamental role in addressing safety verification challenges. While one side of the problem is that only very few families of restrictive linear dynamic systems feature a decidable state reachability problem [86, 69]. The other side is that many continuous systems, modeled by ODEs and/or DDEs, do not have closed form for their solutions. This way, one may resort to numerical approximations, e.g., Runge-Kutta methods, for computing the solutions with approximate values. For formal verification, however, we need to compute enclosures of the solutions. A more generally applicable option is computing an over-approximation for the reachable state set. If the over-approximation does not contain any unsafe state, then neither does the exact reachable set, and thus the system is safe. In case of negative verdict, the safety then is unknown due to excessive over-approximation, and we may need to refine the over-approximation [25]. While over-approximations serve as states that "may" be reachable, under-approximations represent states that "must" be reachable. Therefore, computing under-approximations may help to detect falsification of safety properties by finding counterexamples [119].

2.2.1 Reach Set Computation

For a given safety analysis problem, we specify the problem by a tuple $\mathcal{S} = (\mathcal{H}, \mathcal{I}_0, \mathcal{T})$, where $\mathcal{H}$ is a system model, $\mathcal{I}_0$ is the initial set, and $\mathcal{T}$ is the unsafe set or target [101]. Roughly, reachability analysis seeks to determine whether trajectories of $\mathcal{H}$ can reach $\mathcal{T}$ from $\mathcal{I}_0$. Reachability can be determined by simulating individual trajectories

of $\mathcal{H}$ and observing whether these trajectories can reach $\mathcal{T}$. In fact, such simulation is a typical method by which safety is disproved [101]. For proof of safety, however, all possible trajectories need to be investigated to increase the confidence in the correctness of the system. This is a challenging task in continuous and hybrid systems where the number of states is infinite [101]. Reachability computation is thus an exhaustive exploration of the state space in order to compute at each time step all states reachable by all possible inputs. It is evident that this set-based simulation is more costly than the simulation of individual trajectories of a given system [94]. However, for the safety problem, it provides more confidence in the correctness of the system.

Mathematically, let us assume that the trajectory of the system model $\mathcal{H}$ is defined to be

$$\boldsymbol{\phi}(t;\boldsymbol{x}_0) = \boldsymbol{x}(t),$$

where $\boldsymbol{x}(t)$ is the solution of $\mathcal{H}$ that satisfies the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ at time instant $t = 0$. We define the reachable set Ω of a given initial set $\mathcal{I}_0$ for any time $t \geq 0$ as follows.

Definition 2.2.1. *The reachable set $\Omega(t;\mathcal{I}_0)$ at time $t \geq 0$ is a set of states visited by trajectories originating from $\mathcal{I}_0$ at time $t = 0$ after time duration t, i.e.*

$$\Omega(t;\mathcal{I}_0) = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{\phi}(t;\boldsymbol{x}_0), \boldsymbol{x}_0 \in \mathcal{I}_0\}.$$

Reachability analysis relies on a reach-set computation algorithm, which is closely related to the mathematical model of the given system [148]. The term reachability algorithm or reach-set computation algorithm is usually reserved for techniques that determine the set of states traversed by all trajectories originating from a given set [101]. There are direct and indirect reachability algorithms as discussed in, e.g., [101]. For direct reachability algorithms, there are two main classes. First, Lagrangian approaches which represent the set or tube with information that moves with the flow of the underlying dynamics. These approaches are typically described in terms of forward reachability that starts with states in the initial set I_0 and follows trajectories forward in time. Another class of direct reachability algorithms is Eulerian approaches. These approaches work with a discretization that is not moving with dynamics, although it may be refined during computation. They are typically described in terms of backward reachability that starts with states in the target set T and follows trajectories backwards in time. More details on the comparison between forward and backward reachability

as tools for safety analysis, the interested reader is referred to [101] and the citations within. Also, there are at least two other classes of indirect reachability algorithms for continuous and/or hybrid systems, e.g., automated Lyapunov type methods, the interested reader is referred to [101, 127] and the references therein.

The representation of the computed reachable sets has a crucial effect on the efficiency of the whole procedure, e.g., the computational complexity [143]. Consequently, it has a deciding impact on the applicability of the reachability techniques. The representation of the reachable sets is also strongly related to the mathematical model of the given system. For instance, convex geometric objects such as hyper-rectangles, e.g., [26], polyhedra obtained from convex hull computations, e.g., [29], Zonotopes, e.g., [55], and ellipsoids, e.g., [85], can be successfully used as flow-pipe over-approximations in the reachability analysis for the hybrid automata with all dynamics defined by linear expressions [25]. On the other hand, convex representations are not suitable for flow-pipe overapproximations for the hybrid automata with non-linear dynamics. In this case, some proposed representations such as orthogonal polyhedron, e.g., [20], interval sets, e.g., [130], and Taylor models, e.g., [17], are more suitable. Generally speaking, each of these representations has strengths and weaknesses. More details on this topic can be found in, e.g., [143] and the references therein.

2.2.2 Over- and Under-Approximations

Over-approximation is defined as an approximation that is higher than the true value. Contrary to this, under-approximation is defined as an approximation that is lower than the true value. As already known about the limitations of computing exact reachable sets especially for nonlinear systems, over- and under-approximations for the reachable sets are usually computed for certain applications. For example, computing an over-approximation for the reachable set is usually used for verification purposes. If the over-approximation does not contain any of the defined unsafe states, then the system is reliably safe. On the other hand, for example, computing an under-approximation for the reachable set is used for detecting falsification of safety properties by finding counterexamples. In a sense, when the under-approximation contain any of the defined unsafe states, the system is thus unsafe based on concrete counterexample. Furthermore, as mentioned in the Introduction, computing under- and over-approximations of reachable sets may provide an indication of the precision of an

estimate of the exact reachability region [157, 70]. Also, it can help us prove desired reach-while-avoid properties that are common in many control systems: the system must reach a specified target set of states, while avoiding a set of unsafe states [28, 155].

For the computed reachable set $\Omega(t;\mathcal{I}_0)$ from Definition 2.2.1, we compute the corresponding over- and under-approximation as follows.

Definition 2.2.2. *An over-approximation of the reachable set $\Omega(t;\mathcal{I}_0)$ is a set $O(t;\mathcal{I}_0)$, where $\Omega(t;\mathcal{I}_0) \subseteq O(t;\mathcal{I}_0)$. In contrast, an under-approximation $U(t;\mathcal{I}_0)$ of the reachable set is a nonempty subset of the reachable set $\Omega(t;\mathcal{I}_0)$.*

Notice that from Definition 2.2.2, the over-approximation $O(t;\mathcal{I}_0)$ is an enclosure s.t.

$$\forall \boldsymbol{x}_0 \in \mathcal{I}_0 : \boldsymbol{\phi}(t;\boldsymbol{x}_0) \in O(t;\mathcal{I}_0)$$

holds, for any time $t \geq 0$. On the other hand, the under-approximation $U(t;\mathcal{I}_0)$ is a nonempty set s.t.

$$\forall \boldsymbol{x}(t) \in U(t;\mathcal{I}_0) : \exists \boldsymbol{x}_0 \in \mathcal{I}_0 : \boldsymbol{x}(t) = \boldsymbol{\phi}(t;\boldsymbol{x}_0).$$

As stated in the Introduction, confining safety properties to a set of unsafe states (invariance properties) restricts the ability of designers to formally specify the desired behavior of the system under investigation. In the next section, we present linear temporal logic (LTL) that has more expressive power to specify the desired behavior of the system in formal verification. This introduction to LTL will be useful while presenting our method in Chapter 5.

2.3 Temporal Logic

Temporal logic has found an important application in formal verification, where it is used to specify the desired properties over time [54]. Temporal logics include many types with different expressive powers, e.g., linear temporal logic (LTL), computation tree logic (CTL), to name just a few. For a detailed review on temporal logics see [54] and the references therein. Here, we give a brief overview of linear temporal logic (LTL) serving as a preliminary material that is useful throughout our research work in Chapter 5. In Chapter 5, we will exploit metric interval temporal logic (MITL) which

is a real-time extension of LTL as requirements specification language. More details on MITL will be given in Chapter 5.

2.3.1 Linear Temporal Logic

Linear temporal logic or linear-time temporal logic (LTL) is the most popular and widely used temporal logic in computer science [54]. It was first proposed for the formal verification of computer programs by the seminal work of Amir Pnueli in 1977 [124]. In this context, Pnueli has proposed the application of temporal logics to the specification and verification of reactive and concurrent programs and systems. For instance, in order to ensure the correct behavior of a reactive program, it is necessary to formally specify and verify the acceptable infinite executions of the program. Also, for a concurrent program, where two or more processors are working in parallel, it is necessary to formally specify and verify their interaction and synchronization [54, 96]. The logic LTL can express some properties like safety and liveness properties. We define the syntax and the semantics of LTL as introduced in the literature, e.g., [96, 6].

Syntax

LTL is built up from a finite set of atomic propositions *AP*, the Boolean connectors like conjunction $\wedge$, the negation $\neg$, two basic temporal modalities $\bigcirc$, i.e., the *next* operator, and $\mathcal{U}$, i.e., the *until* operator. The set of LTL formulae over *AP* is inductively defined as in [6].

Definition 2.3.1. ***(Syntax of LTL).*** *The LTL formulae over the set AP of atomic propositions are formed according to the following grammar:*

$$\varphi ::= \top \mid \rho \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \bigcirc\varphi_1 \mid \varphi_1 \,\mathcal{U}\, \varphi_2$$

where $\rho \in AP$, and $\top$ is the Boolean constant true.

The *until* operator is a binary infix operator and requires two LTL formulae as argument, e.g., φ_1 and φ_1. For example, a formula $\varphi_1 \,\mathcal{U}\, \varphi_2$ holds at the current moment, if φ_1 holds at all moments until φ_2 holds at the current or a future moment. The *next* operator is a unary prefix operator and requires a single LTL as argument,

e.g., φ_1. For example, a formula $\bigcirc\varphi_1$ holds at the current moment, if φ_1 holds at the next state (or step) [6].

The full power of propositional logic can be obtained by using the Boolean connectives $\wedge$ and $\neg$. The constant *false* can be derived by $\bot \equiv \neg\top$. Also, for example, other Boolean connectives such as *disjunction* $\vee$, implication $\Rightarrow$, and equivalence $\Leftrightarrow$ can be derived as follows:

$$\begin{aligned}\varphi_1 \vee \varphi_2 &\equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2),\\ \varphi_1 \Rightarrow \varphi_2 &\equiv \neg\varphi_1 \vee \varphi_2,\\ \varphi_1 \Leftrightarrow \varphi_2 &\equiv (\varphi_1 \Rightarrow \varphi_2) \wedge (\varphi_2 \Rightarrow \varphi_1)\end{aligned}$$

From the *until* operator, we can also define the temporal modalities $\Diamond$, i.e., *eventually* operator, and $\Box$, i.e., *always* operator, as follows:

$$\begin{aligned}\Diamond\varphi_1 &\equiv \top\ \mathcal{U}\ \varphi,\\ \Box\varphi &\equiv \neg\Diamond\ \neg\varphi.\end{aligned}$$

This way, $\Diamond\varphi_1$ ensures that φ_1 will be true eventually in the future. $\Box\varphi_1$ holds from now on forever (*always*). This is equivalent to the case that *eventually* $\neg\varphi_1$ does not hold. By combining the temporal modalities, e.g., $\Diamond$ and $\Box$, new temporal modalities can be obtained, e.g., $\Box\Diamond\varphi_1$ that is infinitely often φ_1, and $\Diamond\Box\varphi_1$ that is eventually forever φ_1.

The additional temporal operator *release* $\mathcal{R}$ can also be defined in terms of $\mathcal{U}$ as follows:

$$\varphi_1\ \mathcal{R}\ \varphi_2 \equiv \neg((\neg\varphi_1)\ \mathcal{U}\ (\neg\varphi_2)).$$

It ensures that φ_2 remains true until and including the point where φ_1 first becomes true. If φ_1 never becomes true, then φ_2 must remain true forever.

Semantics

LTL formulae represent properties of paths (or traces). This means that an LTL formula is checked on a path or trace, and thus the path can either fulfill the LTL formula or not. To precisely formulate this, the semantics of LTL formula φ is defined as a language $Words(\varphi)$ that contains all infinite words over the alphabet 2^{AP} that satisfy φ. In other words, an LTL formula φ can be satisfied by an infinite sequence of truth evaluations of variables in the set of atomic propositions AP. Let $W = w_0, w_1, w_2, \ldots$ be such an infinite word, $W(i) = w_i$, $W^i = w_i, w_{i+1}, w_{i+2}, \ldots$, and $\rho \in AP$. The satisfaction relation $\models$ between a word W and an LTL formula is formally defined as follows:

$$
\begin{aligned}
&W \models \rho && \text{iff } \rho \in W(0),\\
&W \models \neg\varphi_1 && \text{iff } W \not\models \varphi_1,\\
&W \models \varphi_1 \wedge \varphi_2 && \text{iff } W \models \varphi_1 \text{ and } W \models \varphi_2,\\
&W \models \bigcirc\varphi_1 && \text{iff } W^1 \models \varphi_1,\\
&W \models \varphi_1 \ \mathcal{U} \ \varphi_2 && \text{iff } \exists i \geq 0 \text{ such that } W^i \models \varphi_2 \text{ and } W^k \models \varphi_1, \text{ for all } 0 \leq k < i.
\end{aligned}
$$

This semantics is extended to an interpretation over paths and states of a transition system TS [6]. We say the transition system TS satisfies an LTL formula φ if all initial paths of TS, i.e., the paths starting from the intial state(s), satisfy φ. For more details on this topic, the interested reader is referred to [6].

Chapter 3

Taylor Model for Continuous Systems

> *"Part of the charm in solving a differential equation is in the feeling that we are getting something for nothing. So little information appears to go into the solution that there is a sense of surprise over the extensive results that are derived."*
>
> [George Robert Stibitz, [142]]

In this chapter, we briefly revisit the method of approximation using Taylor polynomials from [17]. After a brief introduction about Taylor approximations, more details on interval-based Taylor over-approximation method will be presented to enclose the solution of a simple class of delay differential equations (DDEs) [162] that we have applied in order to perform safety and stability verification.

3.1 Taylor approximations

The first version of Taylor's theorem was stated by the mathematician *Brook Taylor* in 1712 [117]. Taylor's theorem provides an approximation of a k-times differentiable function around a given point by a k-th order Taylor polynomial. An explicit expression of the error was provided later by *Joseph-Louis Lagrange*. However, an earlier version of the result was already mentioned in 1671 by *James Gregory* [78].

Given a univariate function f that is κ times differentiable over the domain $(a,b) \subseteq \mathbb{R}$. Taylor approximation of the order k, where $k \leq \kappa$ of function f at $x = c$ for some $c \in (a,b)$ is

$$p_k(x) = f(c) + f^{(1)}(c)(x-c) + \frac{1}{2!}f^{(2)}(c)(x-c)^2 + \cdots + \frac{1}{k!}f^{(k)}(c)(x-c)^k \tag{3.1}$$

such that $f^{(i)}$ denotes the i-th order derivative of f at $x = c$.

The approximation error of $p_k(x)$ for any $x \in (a,b)$ is expressed by the Lagrange remainder term. If f is also $(k+1)$ times differentiable, the Lagrange remainder term is expressed as follows:

$$r_k(x) = f(x) - p_k(x) = \frac{1}{(k+1)!}f^{(k+1)}(\xi(x))(x-c)^{k+1} \tag{3.2}$$

Berz and Makino have originally developed Taylor models [17, 110] to provide over-approximate representations for continuous functions. Taylor models combine Taylor polynomials and intervals that obtain over-approximations [25].

Definition 3.1.1. *(Taylor model). A Taylor model is denoted by a pair (p,I) such that p is a polynomial over a set of variables $\vec{x}$ ranging in an interval domain, and I is the interval remainder.*

3.2 Interval based Taylor Over-approximation for a Class of DDEs

An increasing number of studies have been found for using Taylor models, among several representation techniques, to enclose the solution of ordinary differential equations (ODEs) which are traditionally used to model a vast variety of dynamical systems. Driven by the demand for safety cases (in a broad sense) for safety-critical control systems and the engineers need their models to behave like the real processes by considering, a.o., the delay, in the collaborative work with Zou and Fränzle *et al.*,

we have considered Taylor models to over-approximate the solution of a simple class of delay differential equations (DDEs) [162].

A safe enclosure method using Taylor series with coefficients in interval form was presented in [162][1]. To avoid dimension explosion incurred by the ever-growing degree of the Taylor series along the time axis, the method depends on fixing the degree for the Taylor series and moving higher-degree terms into the parametric uncertainty permitted by the interval form of the Taylor coefficients. By using this data structure to iterate bounded degree Taylor over-approximations of the time-wise segments of the solution to a DDE, the approach identifies the operator that yields the parameters of the Taylor over-approximation for the next temporal segment from the current one. Employing constraint solving to analyze the properties of this operator, an automatic procedure is obtained to provide stability and safety verification for a simple class of DDEs of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = f(\vec{x}(t-\tau)) \tag{3.3}$$

with linear or polynomial vector field $f : \mathbb{R}^N \to \mathbb{R}^N$, where the derivative at t is a function of the trajectory at $t-\tau$, i.e., the signal value determines the future evolution with delay of τ. In order to compute an enclosure for the trajectory defined by an initial value problem of the DDE (3.3), a template interval Taylor form of fixed degree k is defined as

$$f_n(t) = a_{n_0} + a_{n_1}t + \cdots + a_{n_k}t^k, \tag{3.4}$$

where f_n encloses the trajectory for time interval $[n\delta, (n+1)\tau]$, the constant τ is the feedback delay from Eq. (3.3), and $a_{n_0}, \ldots, a_{n_k}$ are interval-vector parameters. The trajectory induced by DDE (3.3) can be represented by a piece-wise function, with the duration of each piece being the feedback delay τ. To compute the enclosure for the whole solution of the DDE, we need to calculate the relation between the interval Taylor coefficients in successive time steps as pre-post-constraints on these interval parameters. For notational convenience, we denote the interval parameters $[a_{n_0}, \ldots, a_{n_k}]$ by a matrix $A(n)$ in $\mathbb{R}^{N\times(k+1)}$. The relation between $A(n)$ and $A(n+1)$

[1]The corresponding prototype implementation of the interval Taylor over-approximation method for DDEs as well as some examples are available for download from `https://github.com/liangdzou/isat-dde`.

can be computed, exploiting different orders of Lie derivatives $f_{n+1}^{(1)}, f_{n+1}^{(2)}, \ldots, f_{n+1}^{(k)}$, as follows:

$$f_{n+1}^{(1)}(t) = g(f_n(t)),\ f_{n+1}^{(2)}(t) = \frac{d\, f_{n+1}^{(1)}(t)}{dt}, \ldots, f_{n+1}^{(k)}(t) = \frac{d\, f_{n+1}^{(k-1)}(t)}{dt}, \tag{3.5}$$

i.e., the first order is obtained directly from the given DDE (3.3) and the $(i+1)$-st order is computed from the i-th order by symbolic differentiation. Then, the Taylor expansion of $f_{n+1}(t)$ with fixed degree k is derived as follows:

$$f_{n+1}(t) = f_n(\delta) + \frac{f_{n+1}^{(1)}(0)}{1!}t + \cdots + \frac{f_{n+1}^{(k-1)}(0)}{(k-1)!}t^{k-1} + \frac{f_{n+1}^{(k)}(\xi_n)}{k!}t^k, \tag{3.6}$$

where ξ_n is a vector ranging over $[0,\tau]^N$.

From Eq. (3.6), by comparing the coefficients of monomials with the same degree at the two sides and by replacing ξ_n by the interval vector $[0,\tau]^N$, we can obtain a time-invariant operator which represents the relation between $A(n)$ and $A(n+1)$. The details of this construction can be retrieved from the example underneath. Hence, we safely enclose the trajectory induced by the DDE (3.3) by a discrete-time model providing a timed state sequence on a state space $\mathcal{S} \subseteq \mathbb{R}^{N\times(k+1)}$.

By constructing a time-invariant discrete dynamic system, the stability of this system can be determined by existing approaches. In case of a linear time-invariant discrete dynamic system, the method proposed in [33] is used to determine whether this discrete dynamic system is asymptotically or robustly stable. For a more general polynomial case concerning the right-hand side of Eq. (3.3), as well as the initial condition, the stability of a time-invariant polynomial discrete dynamic system can be analyzed thanks to existing methods on computing parametric Lyapunov functions, such as [90, 131]. More details on stability concern for DDEs in the form of Eq. (3.3) are given in [162], and references therein.

Turning our attention now to safety verification rather than stability, the main objective of this dissertation, the constructed time-invariant discrete dynamic system 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

[52]. An unbounded safety verification is also possible by means of pursuing BMC for sufficiently many steps k_{depth} in case our DDE is stabilizing. More details on unbounded safety verification are found in [162] and references therein. On the other hand, we mainly focus on time-bounded safety verification in this dissertation.

3.2.1 Time-Wise Discretization of DDEs into Timed State Sequences

> *"Most of the fundamental ideas of science are essentially simple, and may, as a rule, be expressed in a language comprehensible to everyone."*
>
> [Albert Einstein, [45]]

Motivated by a lesson learned from the long-dead, great man, especially when it comes to explaining complex mathematical problems, in this section, interval-based Taylor over-approximation method is demonstrated in a very simple example providing the discrete-time model that encloses the solution of a DDE like Eq. (3.3). The running example is the linear DDE with a single constant delay as discussed in [162] and introduced in our motivation, Chapter 1, as follows:

$$\dot{x}(t) = -x(t-1) \tag{3.7}$$

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

The method provided in [162] aims at over-approximating the solution of DDE (3.7) by iterating bounded degree interval-based Taylor over-approximations of the time-wise segments of the solution to the DDE. That way, we identify the operator that yields the parameters of the Taylor over-approximation for the next temporal segment from the current one. For instance, suppose we are trying to over-approximate the solution of DDE (3.7) by polynomials of degree 2. Then we can predefine a template Taylor form

$$f_n(t) = a_{n_0} + a_{n_1} t + a_{n_2} t^2$$

on interval $[n, n+1]$, where a_{n_0}, a_{n_1}, and a_{n_2} are interval parameters able to incorporate the approximation error eventually necessarily incurred by bounding the degree of the

polynomial to (in this example) 2. Here, $f_n(t)$ corresponds to the solution x of DDE (3.7) at time $n+t$, i.e., $f_n(t)$ over-approximates $x(n+t)$ in the sense of $x(n+t) \in f_n(t)$.

In order to compute the Taylor model, 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 (where both segments are of duration 1 each) have to be calculated. The first derivative $f_{n+1}^{(1)}(t)$ is computed directly from Eq. (3.7) as

$$f_{n+1}^{(1)}(t) = -f_n(t) = -a_{n_0} - a_{n_1} t - a_{n_2} t^2 .$$

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

$$f_{n+1}^{(2)}(t) = \frac{d\,(f_{n+1}^{(1)}(t))}{dt} = -a_{n_1} - 2a_{n_2} t .$$

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

$$\begin{aligned} f_{n+1}(t) &= f_n(1) + \frac{f_{n+1}^{(1)}(0)}{1!} t + \frac{f_{n+1}^{(2)}(\xi_n)}{2!} t^2 \\ &= (a_{n_0} + a_{n_1} + a_{n_2}) - a_{n_0} t - \frac{a_{n_1} + 2a_{n_2}\xi_n}{2} t^2 . \end{aligned}$$

Then, the operator expressing the relation between Taylor coefficients in the current and the next step can be derived by replacing both $f_n(t)$ and $f_{n+1}(t)$ with their parametric forms $a_{n_0} + a_{n_1} t + a_{n_2} t^2$ and $a_{n+1_0} + a_{n+1_1} t + a_{n+1_2} t^2$ in the above equation and pursuing coefficient matching. As a result, one obtains the operator

$$\begin{bmatrix} a_{n+1_0} \\ a_{n+1_1} \\ a_{n+1_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\xi_n \end{bmatrix} \begin{bmatrix} a_{n_0} \\ a_{n_1} \\ a_{n_2} \end{bmatrix} \tag{3.8}$$

mapping the coefficients of the Taylor form at step f_n to the coefficients of the Taylor form of f_{n+1}. The coefficients change at every τ time units (every second in the given example) according to the above operator, which therefore defines a discrete-time

dynamical system corresponding to the DDE. The discrete-time operator can be rendered time-invariant, yet interval-valued by substituting the uncertain time varying parameter ξ_n with its interval $[0,\tau]$. Hence, we can safely enclose the solution of DDE (3.7) by a sequence of parametric Taylor series with parameters in interval form. In the case of system (3.7), as well as for any other linear DDE, the operator generating this sequence is a set-valued linear operator definable by an effectively computable interval matrix.

For a time-bounded safety verification, a given safety property like $S(x) \hat{=} -1 \leq x \leq 1$, for example, 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 (3.7) becomes safety property $\forall n \in \mathbb{N}, t \in [0,1] : S(f_n(t))$ in system (3.8). Discharging this proof obligation in BMC requires polynomial constraint solving due to the Taylor forms involved. We use bounded model checking (BMC) mode in `iSAT3` [138] as given in details later in Chapter 5. In what follows, we start warming up to prepare the readers for the method that will be discussed in Chapter 5.

3.2.2 Solving Time-Bounded Verification Problems by `iSAT3`

We opted `iSAT3` solver [138] in bounded model checking (BMC) mode for solving the time-bounded verification problem, taking advantage of the solver for efficiently solving (un)bounded verification problems that involve polynomial (and, if needed, transcendental) arithmetic. Thus, it is optimal in our case, owing to the Taylor forms involved. The `iSAT3` solver is a stable version implementation of the iSAT algorithm, introduced by Fränzle *et al.* in [52].

In [162], the presented method facilitate solving time-(un)bounded verification problems against invariance safety properties. In this context, the safety properties can be expressed using bounded *always* temporal operator, i.e., $\Box_{k_{depth}}$, where BMC of the Taylor over-approximation transition system aims at finding a run of bounded length k_{depth} that starts in an initial state of the system, complies with the system's transition relation, i.e., the constructed operator (3.8), and ends in a state in which a certain (un)desired property holds. The bounded model checking engine then constructs a formula which is satisfiable if and only if a trace with a given property exists.

For example, assume we want to verify the constructed Taylor over-approximation model of the above running example against a given safety property like $\Box_{k_{depth}}(x \leq 1.2)$, where $k_{depth} = 10$. We encode the constructed discrete transition system on Taylor model, i.e., the operator (3.8), according to the input format for the `iSAT3` BMC: declaring all variables, describing the initial state(s) of the system, describing the transition relation in symbolic form, i.e., the operator (3.8), and finally characterizing the state(s) whose reachability is to be checked that is expressed by the given property $\Box_{k_{depth}}(x \leq 1.2)$. The input format of the `iSAT3` BMC for such running example is illustrated in Listing 3.1.

```
1  DECL
2  -- the range of each variable has to be bounded
3    float [-1000, 1000] a0, a1, a2, x;
4    float [0,1] t, xi;
5
6  -- define counter for the bounded verification problem
7    int [0,9] counter;
8
9  INIT
10 -- initial value of x over [0,1]
11   x = 1;
12
13 -- initialize Taylor coefficients
14   a0 = 1;
15   a1 = 0;
16   a2 = 0;
17
18 -- initialize the counter observing the time interval
19 -- covered by the bounded always
20   counter = 0;
21
22 TRANS
23 -- relation between Taylor coefficients current and next step
24   a0' = a0 + a1 + a2;
25   a1' = -a0;
26   a2' = -0.5*a1 - xi*a2;
27
```

```
28 -- x(t) is given by a Taylor form of degree 2
29   x' = a0' + a1'*t + a2'*(t^2);
30 -- note the implicit existential quantification of t
31
32 -- increment the counter by 1 after each time frame
33   counter' = counter + 1;
34
35 TARGET
36 -- state to be reached in bounded time
37   x > 1.2 and
38   counter <= 9;
```

Listing 3.1 The input format of iSAT3 BMC for the transition system of Equation 3.7.

The solver unwinds the transition relation k_{depth} times, conjoins the resulting formula with the formulae describing the initial state(s) and the target state(s), and then solves the obtained formula. More details on this will be given in Chapter 5 that is built based on this chapter. There, we will consider wider range of temporal properties rather than just invariance properties as considered in [162].

3.3 Discussion

"Nothing is particularly hard if you divide it into small jobs."

[Henry Ford[2], 1863–1947]

The interval-based Taylor over-approximation method is based on using interval Taylor forms for safely enclosing segments of the solution of delay differential equations (DDEs) with point- or set-valued initial conditions. Interestingly, this method complements the established methods for enclosing reachable state sets of ordinary differential equations (ODEs), lifting their power to DDEs. These early successes in a new area, as considering DDEs in automatic formal verification, may give hope to cover 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.

[2]United States manufacturer of automobiles who pioneered mass production.

For this introductory research in the verification area of DDEs, it is assumed that the system dynamics is represented as a DDE with a single, constant delay, i.e., the restricted form given by Eq. (3.3). This study has gone some way towards the verification of several dynamical systems that can be modeled by DDE with a single constant as in biology [58, 93], optics [75], economics [145, 144], ecology [51]. 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. Presumably such cases can be addressed by a layered combination of Taylor-model computation for ODEs, e.g. [110], with the ideas exposed herein. Note that we have considered in [156] a higher class in complexity of DDEs, where the right-hand side of characterizing the differential equation is a combination of ODE and DDE with single constant delay. This method will be introduced in Chapter 4. Also, it should be noted that we have considered the class of systems that involves a combination of ODE and DDE with multiple constant delays as a collaborative work with Chen *et al.* in [24] using validated simulation-based verification technique; however, this work is not considered in this dissertation.

Although the interval-based Taylor over-approximation method, presented in this chapter, is interesting, it may fail due to excessive over-approximation, which would be induced by selecting an insufficient bound on the degree of the Taylor forms. A more practical solution for this problem might be by selecting a higher degree of the Taylor forms. However, with a negative verdict, it is unclear whether failure of the verification attempt is an artifact of excessive over-approximation or an inherent property of the system under investigation. Further work needs to be performed to develop a clear improvement on our method for disambiguating these two cases [162].

Chapter 4

Over- and Under-Approximations for a Class of DDEs

> *"Until now, physical theories have been regarded as merely models with approximately describe the reality of nature. As the models improve, so the fit between theory and reality gets closer. Some physicists are now claiming that supergravity is the reality, that the model and the real world are in mathematically perfect accord."*
>
> [P.C.W. Davies[1]]

In this chapter, we discuss in detail lifting another method, which was introduced by Xue *et al.* in [155, 157] for ordinary differential equations (ODEs), to compute safe over-approximations as well as under-approximations for a higher class in complexity of delay differential equations (DDEs), compared to the class of DDE discussed in previous chapter, where the right-hand side of characterizing the differential equation in this class is a combination of ODE and DDE with single constant delay. The original paper we have published on which this chapter is based, is [156]. Note that we have considered also the class of systems that involves a combination of ODE and DDE with multiple constant delays as a collaborative work with Chen *et al.* in [24] using validated simulation-based verification techniques; however, this work is not considered in this dissertation, as already mentioned.

[1] Superforce (1984, 1985), 149.

While we have discussed in the previous chapter an over-approximation method for a simple class of DDEs, our contribution in this chapter not only adapting a developed method for a more complex class of DDEs, but computing both over- and under-approximations for such a class of DDEs. The motivation of this work is not only the inspiring work by Xue *et al.* in [155, 157] for ODEs, but computing under-approximations of the reachable sets are incorporated into a variety of application in engineering domains and accordingly the need to compute under-approximations for DDEs as we have discussed in the introduction of this dissertation.

The method discussed in [155, 157] is reachability analysis method based on set-boundary of ODEs. The main idea of this method relies on the fact that the solution mapping of ODE is a homeomorphism and thus preserves set boundaries. This way, one can retrieve safe over- and under-approximations for ODEs from enclosures of the dynamic images of the boundaries of the initial set. This raises many questions regarding whether the set-boundary based method could be used for DDEs taking into consideration the fact that the solution mappings of DDEs need not be homeomorphisms. Therefore we will devote ourselves, in this chapter, to answer the raised questions and lift the reachability analysis method based on set-boundary of ODEs to DDEs.

4.1 Preliminaries

In this section, we formally define the dynamical systems of interest, i.e., a combination of ODE and DDE with single constant delay, and recall the basic notion of reachability used throughout this chapter. The following conventions will be used in the remainder of this chapter: the space of continuously differentiable functions on $\mathcal{X}$ is denoted by $\mathcal{C}^1(\mathcal{X})$; for a set Δ, the decorations Δ°, Δ^c and $\partial\Delta$ represent its interior, complement, and boundary respectively; vectors in the $\mathbb{R}^n$ as well as of functions are denoted by boldface letters. The set of $n \times n$ matrices over the field $\mathbb{R}$ of real numbers is denoted by $\mathbb{R}^{n\times n}$.

We consider systems, in this chapter, that can be modeled by delay differential equations (DDEs) of the form

$$\dot{\boldsymbol{x}} = \begin{cases} \boldsymbol{g}(\boldsymbol{x}), & \text{if } t \in [0,\tau), \boldsymbol{x}(0) \in \mathcal{I}_0 \\ \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau), & \text{if } t \in [\tau, K\tau], \end{cases} \tag{4.1}$$

where $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))' \in \mathcal{X}$, $\boldsymbol{x}_\tau = (x_1(t-\tau), x_2(t-\tau), \ldots, x_n(t-\tau))' \in \mathcal{X}$, $\mathcal{X} \subseteq \mathbb{R}^n$, $K \geq 2$ is a positive integer, $\boldsymbol{g} : \mathcal{X} \mapsto \mathbb{R}^n$ describes the process which the initial function is determined by the initial value $\boldsymbol{x}(0) \in \mathcal{I}_0$, and $\mathcal{I}_0 \subset \mathbb{R}^n$ is a simply connected[2] compact[3] set and $\boldsymbol{f} : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}^n$ is globally Lipschitz continuous over the variables $\boldsymbol{x}(t)$ and $\boldsymbol{x}(t-\tau)$. Also, we require that $\boldsymbol{g}(\boldsymbol{x}) \in \mathcal{C}^1(\mathcal{X})$ and $\boldsymbol{g} : \mathcal{X} \mapsto \mathbb{R}^n$ satisfies the Lipschitz continuity condition with respect to the variables $\boldsymbol{x}(t)$, guaranteeing that $\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})$ with initial value $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathcal{I}_0$ has a unique solution on $[0,\tau]$. Therefore, Eq.(4.1) describes a deterministic process on $[0, K\tau]$. Besides, we assume that max norms $\|\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}\|_{\max}$, $\|\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\|_{\max}$ and $\|\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}}\|_{\max}$ of the matrices $\|\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}\|$, $\|\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\|$ and $\|\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}}\|$ are uniformly bounded for any combination of $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \mathcal{X}$, i.e.,

$$\|\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}\|_{\max} \leq M', \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{x}}\|_{\max} \leq M, \|\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})}{\partial \boldsymbol{y}}\|_{\max} \leq N, \tag{4.2}$$

where M', M and N are positive real numbers.

Given System (4.1) with an initial set $\mathcal{I}_0$, and a finite time duration t, where $0 \leq t \leq K\tau$ and $K \geq 2$ is a positive integer, the set of allowable initial functions selected by $\boldsymbol{g}(\boldsymbol{x})$ is just a set of solutions of the ODE

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})$$

initialised in $\mathcal{I}_0$ with respect to the time interval $[0,\tau]$. The trajectory of System (4.1) is defined to be

$$\boldsymbol{\phi}(t;\boldsymbol{x}_0) = \boldsymbol{x}(t),$$

where $\boldsymbol{x}(t)$ is the solution of System (4.1) that satisfies the initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$ at time instant $t = 0$. In addition, we recall some definitions from Chapter 2 to define the reachable set of a given initial set $\mathcal{I}_0$ for any time $t \geq 0$ and its corresponding over- and under-approximations as follows.

[2] A connected set is a set which cannot be partitioned into two nonempty subsets which are open in the relative topology induced on the set.

[3] The set is called compact if it is closed and bounded.

Definition 4.1.1. *The reachable set* $\Omega(t;\mathcal{I}_0)$ *at time* $t \geq 0$ *is a set of states visited by trajectories originating from* $\mathcal{I}_0$ *at time* $t = 0$ *after time duration* t*, i.e.*

$$\Omega(t;\mathcal{I}_0) = \{\boldsymbol{x} : \boldsymbol{x} = \boldsymbol{\phi}(t;\boldsymbol{x}_0), \boldsymbol{x}_0 \in \mathcal{I}_0\}.$$

As intuitively known about the limitations of computing exact reachable sets especially for nonlinear systems, the approximate solutions are usually computed. Hence, it is convenient to consider their over- and under-approximations for certain applications. Its corresponding over- and under-approximations are defined below.

Definition 4.1.2. *An over-approximation of the reachable set* $\Omega(t;\mathcal{I}_0)$ *is a set* $O(t;\mathcal{I}_0)$*, where* $\Omega(t;\mathcal{I}_0) \subseteq O(t;\mathcal{I}_0)$*. In contrast, an under-approximation* $U(t;\mathcal{I}_0)$ *of the reachable set is a nonempty subset of the reachable set* $\Omega(t;\mathcal{I}_0)$*.*

Notice that from Definition 4.1.2, the over-approximation $O(t;\mathcal{I}_0)$ is an enclosure s.t.

$$\forall \boldsymbol{x}_0 \in \mathcal{I}_0 : \boldsymbol{\phi}(t;\boldsymbol{x}_0) \in O(t;\mathcal{I}_0)$$

holds, where $0 \leq t \leq K\tau$. On the other hand, the under-approximation $U(t;\mathcal{I}_0)$ is a nonempty set s.t.

$$\forall \boldsymbol{x}(t) \in U(t;\mathcal{I}_0) : \exists \boldsymbol{x}_0 \in \mathcal{I}_0 : \boldsymbol{x}(t) = \boldsymbol{\phi}(t;\boldsymbol{x}_0).$$

As we mentioned above, aiming at computing safe over- as well as under-approximations for DDEs in the form of Eq.(4.1), we wish to extend the reachability analysis method based on set-boundary of ODEs, discussed in [157], to DDEs. The method in [157] relies on the fact that the solution mapping of ODE is a homeomorphism and thus preserves set boundaries, permitting to retrieve safe over- and under-approximations from enclosures of the dynamic images of the boundaries of the initial set. Actually it is a real challenge to lift such a method to DDEs.

Homeomorphism is a continuous function between topological spaces that has a continuous inverse function. In other words, homeomorphism is a one-to-one correspondence between two topological spaces such that the two mutually-inverse mappings defined by this correspondence are continuous [67]. In fact, the solution mappings of DDEs in the form of Eq.(4.1), however, need not be homeomorphisms. That is, the inverse of the solution mapping of Eq.(4.1) may have numerous branches,

not a unique inverse as ODE, that need to be taken into account. This is a serious barrier to extend the reachability method based on set-boundary of ODEs to DDEs. Hence, we devote ourselves to exposing a class of systems of the form (4.1) with solution mappings having that desirable property. Then, we compute safe over- and under-approximations for such a class of DDEs. We study, in this chapter, the following problems:

Problem 1. Which class of systems characterized by Eq. (4.1) has solution mappings forming a homeomorphism?

Problem 2. How can we efficiently compute safe over- and under-approximations of the reachable set for the systems described in **Problem 1** if the initial set $\mathcal{I}_0$ is a simply connected compact set?

4.1.1 Nonlinear Control Systems

Nonlinear control systems obviously arise in natural as well as artificial dynamic processes including biology, physics, economics, engineering, etc. In this vein, they cover a wider class of systems as a real-world systems. They are characterized by the presence of nonlinear elements in the right-hand side of the characterizing differential equation. Such non-linearities may stem from both the system under control (i.e., the plant) and the controller itself.

Ordinary differential equations (ODEs) are traditionally used to model the continuous behavior of such systems. In general, the nonlinear control systems that are modeled by ODEs with a control input are of the following form

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \tag{4.3}$$

where $\boldsymbol{x}(0) \in \mathcal{X}_0 \subseteq \mathbb{R}^n$, $\boldsymbol{u}(t) \in \boldsymbol{U} \subseteq \mathbb{R}^m$, and $\mathcal{X}_0$, $\boldsymbol{U}$ are both compact sets. Equation (4.3) is required to be (globally) Lipschitz-continuous and the input trajectory $\boldsymbol{u}(\cdot) : \mathbb{R}^+ \mapsto \boldsymbol{U}$ is required to be piecewise continuous so that a solution is guaranteed to exist globally in the sense for all $t \geq 0$. For convenience, we denote the space of piecewise continuous functions from $\mathbb{R}^+$ to $\boldsymbol{U}$ as $\mathcal{P}$.

Let us denote the solution to System (4.3) for a given initial state and an input trajectory by $\chi(t;\boldsymbol{x}_0,\boldsymbol{u}(\cdot))$, where $t \geq 0$, $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathcal{X}_0$ and $\boldsymbol{u}(\cdot) \in \boldsymbol{U}$ is the input trajectory within the time interval $[0,t]$. The reachable set at time $t = r$ can be defined for a set of initial states $\mathcal{X}_0$ and a set of input values $\boldsymbol{U}$ as

$$\mathcal{R}(r) = \{\chi(r;\boldsymbol{x}_0,\boldsymbol{u}) \in \mathbb{R}^n | \boldsymbol{x}_0 \in \mathcal{X}_0, \boldsymbol{u} \in \mathcal{P}\}.$$

Althoff's approaches [3, 1] are among the many methods for computation of over-approximations of the reachable set $\mathcal{R}(r)$. Such methods can also be applied to over-approximating the reachable set for cases involving DDEs of the form (4.1) by regarding the delay term $\boldsymbol{x}_\tau$ as the time-varying uncertainty $\boldsymbol{u}$ (cf. [72] for such an algorithm).

4.2 Reachable Sets Computation

This section mainly focuses on solving **Problem 1** and **Problem 2** as presented in Section 4.1. Firstly, we address **Problem 1** by conducting sensitivity analysis on the solution mappings $\boldsymbol{\phi}(t;\cdot)$ with respect to the initial states for DDEs of the form of Eq. (4.1). This facilitates imposition of a bound constraint on the time-lag term such that the homeomorphism property is guaranteed. Then, addressing **Problem 2**, we generalize the reachability method based on set-boundary of ODEs, discussed in [155, 157], to the computation of safe approximations of reach sets for systems of the form (4.1). This way, we can construct over- and under-approximations of their reachable sets.

4.2.1 Sensitivity Analysis Theory

For a system governed by the ODE

$$\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x}),$$

where $t \in [0,\tau]$, its flow mapping $\boldsymbol{\phi}(t;\boldsymbol{x}_0)$ as a function of $\boldsymbol{x}_0$ is differentiable with respect to the initial state $\boldsymbol{x}_0$, if $\boldsymbol{g} \in \mathcal{C}^1(\mathcal{X})$ and $\boldsymbol{g}$ is Lipschitz continuous. The

sensitivity of solutions at time $t \in [0, \tau]$ to initial conditions is defined by

$$s_{\boldsymbol{x}_0}(t) = \frac{\partial \boldsymbol{\phi}(t;\boldsymbol{x}_0)}{\partial \boldsymbol{x}_0}, \tag{4.4}$$

where $s_{\boldsymbol{x}_0}(t)$ is a square matrix of order n. The $(i,j)_{th}$ element of $s_{\boldsymbol{x}_0}$ basically represents the influence of variations in the i_{th} coordinate $x_{0,i}$ of $\boldsymbol{x}_0$ on the j_{th} coordinate $x_j(t)$ of $\boldsymbol{\phi}(t;\boldsymbol{x}_0)$. To compute the sensitivity matrix, we first apply the chain rule to get the derivative of $s_{\boldsymbol{x}_0}$ with respect to time [38], as follows:

$$\frac{d}{dt}\frac{\partial \boldsymbol{\phi}(t;\boldsymbol{x}_0)}{\partial \boldsymbol{x}_0} = D_{\boldsymbol{g}}(\boldsymbol{\phi}(t;\boldsymbol{x}_0))\frac{\partial \boldsymbol{\phi}(t;\boldsymbol{x}_0)}{\partial \boldsymbol{x}_0},$$

which yields the ODE

$$\dot{s}_{\boldsymbol{x}_0} = D_{\boldsymbol{g}} s_{\boldsymbol{x}_0}$$

describing evolution of sensitivity over time, where $D_{\boldsymbol{g}}$ is the Jacobian matrix of vector field $\boldsymbol{g}$ along the trajectory $\boldsymbol{\phi}(t;\boldsymbol{x}_0)$. This equation is a linear time-varying ODE and the relevant initial value $s_{\boldsymbol{x}_0}(0)$ is the identity matrix $\boldsymbol{I} \in \mathbb{R}^{n\times n}$.

Remark 4.2.1. *From the definition of the sensitivity matrix $s_{\boldsymbol{x}_0}(t)$, we observe that $s_{\boldsymbol{x}_0}(t)$ is also the Jacobian matrix of the mapping $\boldsymbol{\phi}(t;\cdot) : \mathcal{I}_0 \mapsto \Omega(t;\mathcal{I}_0)$, where $t \in [0,\tau]$.*

Assume that the solution mapping $\boldsymbol{\phi}(t;\boldsymbol{x}_0)$ of System (4.1) for time ranging over $t \in [(k-1)\tau, k\tau]$ and the state variable $\boldsymbol{x}_0 \in \mathcal{I}_0$, could be equivalently reformulated as a continuously differentiable function of the state variable $\boldsymbol{x}((k-1)\tau)$ in $\Omega((k-1)\tau;\mathcal{I}_0)$ and the time variable $t \in [(k-1)\tau, k\tau]$, i.e.,

$$\boldsymbol{\phi}(t;\boldsymbol{x}_0) = \boldsymbol{\psi}_{k-1}(t;\boldsymbol{x}((k-1)\tau),(k-1)\tau),$$

where $k \in \{1,\ldots,K-1\}$, and $\boldsymbol{x}((k-1)\tau) = \boldsymbol{\phi}((k-1)\tau;\boldsymbol{x}_0)$. Also assume the determinant of the Jacobian matrix of the mapping $\boldsymbol{\psi}_{k-1}(t;\boldsymbol{x}((k-1)\tau),(k-1)\tau)$ with respect to any state $\boldsymbol{x}((k-1)\tau) \in \Omega((k-1)\tau;\mathcal{I}_0)$ is not zero for any $t \in [(k-1)\tau, k\tau]$. Then, we deduce what follows.

Lemma 4.2.1. *Given the above assumptions, the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t) = \frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{x}(k\tau)}$, $t \in [k\tau,(k+1)\tau]$, for System (4.1) satisfies the following linear time-varying ODE:*

$$\dot{s}_{\boldsymbol{x}(k\tau)} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}} s_{\boldsymbol{x}(k\tau)} + \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau}\frac{\partial \boldsymbol{x}_\tau}{\partial \boldsymbol{x}(k\tau)}, \tag{4.5}$$

where $\dot{s}_{\boldsymbol{x}(k\tau)} = \frac{ds_{\boldsymbol{x}(k\tau)}}{dt}$, *and* $s_{\boldsymbol{x}(k\tau)}(k\tau) = \boldsymbol{I} \in \mathbb{R}^{n\times n}$.

Proof. Since the determinant of the Jacobian matrix of the mapping $\boldsymbol{x}(t) = \boldsymbol{\psi}_{k-1}(t;\boldsymbol{x}((k-1)\tau,(k-1)\tau)$ with respect to any state $\boldsymbol{x}((k-1)\tau) \in \Omega((k-1)\tau;\mathcal{I}_0)$ is not zero for $t \in [(k-1)\tau, k\tau]$, then for any fixed $t \in [(k-1)\tau, k\tau]$, the mapping

$$\boldsymbol{x}(t) = \boldsymbol{\psi}_{k-1}(t;\ \cdot\ ,(k-1)\tau) : \Omega((k-1)\tau;\mathcal{I}_0) \longmapsto \Omega(t;\mathcal{I}_0)$$

is a bijection and its inverse mapping from $\Omega(t;\mathcal{I}_0)$ to $\Omega((k-1)\tau;\mathcal{I}_0)$ is continuously differentiable. Thus, the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t)$ for $t \in [k\tau,(k+1)\tau]$ satisfies the sensitivity equation:

$$\dot{s}_{\boldsymbol{x}(k\tau)} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}} s_{\boldsymbol{x}(k\tau)} + \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau} \frac{\partial \boldsymbol{x}_\tau}{\partial \boldsymbol{x}(k\tau)},$$

with $s_{\boldsymbol{x}(k\tau)}(k\tau) = \boldsymbol{I} \in \mathbb{R}^{n\times n}$.

□

From the definition of the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t) = \frac{\partial \boldsymbol{x}(t)}{\partial \boldsymbol{x}(k\tau)}$ together with the fact that its determinant is not equal to zero, the solution mapping $\boldsymbol{\phi}(t;\ \cdot) : \mathcal{I}_0 \mapsto \Omega(t;\mathcal{I}_0)$ for $t \in [k\tau,(k+1)\tau]$ could be formulated equivalently as a continuously differentiable function of the state variable $\boldsymbol{x}(k\tau) \in \Omega(k\tau;\mathcal{I}_0)$ for any fixed $t \in [k\tau,(k+1)\tau]$, and this mapping from $\Omega(k\tau;\mathcal{I}_0)$ to $\Omega(t;\mathcal{I}_0)$ for $t \in [k\tau,(k+1)\tau]$ is a continuously differentiable homeomorphism between two topological spaces $\Omega(k\tau;\mathcal{I}_0)$ and $\Omega(t;\mathcal{I}_0)$. This assertion is formalized in Corollary 4.2.1.

Corollary 4.2.1. *If the determinant of the sensitivity matrix* $s_{\boldsymbol{x}(k\tau)}(t)$ *with respect to any state* $\boldsymbol{x}(k\tau) \in \Omega(k\tau;\mathcal{I}_0)$ *at time* $k\tau$ *is not zero for any* $t \in [k\tau,(k+1)\tau]$, *then* $\boldsymbol{\phi}(t;\boldsymbol{x}_0)$ *for* $\boldsymbol{x}_0 \in \mathcal{I}_0$ *and* $t \in [k\tau,(k+1)\tau]$ *could be equivalently reformulated as a continuously differentiable function of the state variable* $\boldsymbol{x}(k\tau) \in \Omega(k\tau;\mathcal{I}_0)$ *and the time variable* $t \in [k\tau,(k+1)\tau]$, *and the state* $\boldsymbol{x}(t) = \boldsymbol{\phi}(t;\boldsymbol{x}_0)$ *is uniquely determined by the state* $\boldsymbol{x}(k\tau)$ *for any fixed* $t \in [k\tau,(k+1)\tau]$, *where* $\boldsymbol{x}(k\tau) = \boldsymbol{\phi}(k\tau;\boldsymbol{x}_0)$.

4.2.2 Generating a Constraint Bounding the Time-Lag Term

According to what we discussed above, here, we will infer a class of DDEs of the form (4.1), where the determinant of the corresponding sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t)$ with respect to the state variable $\boldsymbol{x}(k\tau) \in \Omega(k\tau;\mathcal{I}_0)$ at time $k\tau$ is not zero for $t \in [k\tau,(k+1)\tau]$, and $k = 0,\ldots,K-1$. Such a class of equations is derived by appropriately confining the time-lag term of the DDE (4.1), i.e., τ. In what follows, first, we review the classical result about diagonally dominant matrices from Varah [151].

If a matrix $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, i.e.,

$$\Delta_i(A) = |A_{ii}| - \sum_{j\neq i} |A_{ij}| > 0, \text{ with } 1 \leq i \leq n,$$

where A_{ij} is the entry in the i_{th} row and j_{th} column of the matrix A, then the inverse of the matrix A satisfies the bound

$$\|A^{-1}\|_\infty \leq max_{1\leq i\leq n} \frac{1}{\Delta_i(A)}.$$

Note that, by convention, $\|\cdot\|_\infty$ is the maximum absolute row sum of a matrix. Based on this classical result, we derive a constraint on the time-lag term τ in System (4.1) rendering the sensitivity matrix mentioned in Lemma 4.2.1 strictly diagonally dominant. We begin with the time interval $[0,\tau]$.

Lemma 4.2.2. *There exist $R > 1$ and $\varepsilon > 1$ s.t. if*

$$\tau \leq \min\left\{\frac{\varepsilon-1}{\varepsilon n^2 M' R}, \frac{\ln R}{2\sqrt{n} n M'}\right\},$$

the matrix $s_{\boldsymbol{x}_0}(t)$ in Eq. (4.4) is diagonally dominant and satisfies $\|s_{\boldsymbol{x}_0}(t)\|_{\max} \leq R$ and $\max_{1\leq i\leq n} \frac{1}{\Delta_i(s_{\boldsymbol{x}_0}(t))} \leq \varepsilon$ for $t \in [0,\tau]$ and $\boldsymbol{x}_0 \in \mathcal{I}_0$, where M' is presented in (4.2).

Proof. Since the sensitivity matrix $s_{\boldsymbol{x}(0)}(t)$ for $t \in [0,\tau]$ with respect to the state $\boldsymbol{x}(0)$ satisfies the sensitivity equation

$$\dot{s}_{\boldsymbol{x}(0)} = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} s_{\boldsymbol{x}(0)}, \text{ with } s_{\boldsymbol{x}(0)}(0) = \boldsymbol{I}. \tag{4.6}$$

In the following, we employ the comparison principle for ODEs to derive a bound on the solution to Eq. (4.6).

Let

$$M_d = \max_{0\le t\le \tau} 2\sqrt{n}n\|\boldsymbol{A}(t)\|_{\max},$$

where $\boldsymbol{A}(t) = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}$. It is obvious that $M_d \le 2\sqrt{n}nM'$.

We take the j_{th} column of the sensitivity matrix $s_{\boldsymbol{x}(0)}(t)$ and the matrix $\boldsymbol{b}(t)$ as a vector $\boldsymbol{y}(t)$ and $\boldsymbol{b}_j(t)$, where $j \in \{1,\ldots,n\}$. Let $u(t) = \|\boldsymbol{y}(t)\|_2^2 = \langle \boldsymbol{y}(t), \boldsymbol{y}(t)\rangle$ with $u(0) = 1$, where $\|\boldsymbol{y}(t)\|_2$ is the 2-norm for $\boldsymbol{y}$ and $\langle\cdot,\cdot\rangle$ is an inner product in $\mathbb{R}^n$.

Based on Cauchy-Schwarz inequality and the fact that $2\|\boldsymbol{y}\|_2 \le \|\boldsymbol{y}\|_2^2 + 1$ as well as $\|\boldsymbol{A}(t)\boldsymbol{y}\|_2 \le \|\boldsymbol{A}(t)\|_F\|\boldsymbol{y}\|_2 \le \sqrt{n}\|\boldsymbol{A}(t)\|_2\|\boldsymbol{y}\|_2$, where $\|\boldsymbol{A}(t)\|_F$ is the Frobenius norm of the matrix $\boldsymbol{A}(t)$, we obtain

$$\begin{aligned} \dot{u} &= 2\langle \boldsymbol{y}, \dot{\boldsymbol{y}}\rangle \le 2\|\boldsymbol{y}\|_2\|\dot{\boldsymbol{y}}\|_2 = 2\|\boldsymbol{y}\|_2\|\boldsymbol{A}(t)\boldsymbol{y}\|_2 \le 2\sqrt{n}\|\boldsymbol{y}\|_2^2\|\boldsymbol{A}(t)\|_2 \\ &\le 2\sqrt{n}\|\boldsymbol{A}(t)\|_2\|\boldsymbol{y}\|_2^2 \le M_d\|\boldsymbol{y}\|_2^2 = M_d u. \end{aligned} \tag{4.7}$$

Applying Gronwall's inequality [14] to Eq. (4.11), we deduce that

$$u(t) \le u_0 e^{M_d t} = u_0 e^{M_d t} \le R_d$$

for $0 \le t \le \tau$, where $u_0 = u(0) = 1$, and

$$R_d = e^{M_d \tau}.$$

Therefore, $\|\boldsymbol{y}(t)\|_2^2 \le R_d$ for $0 \le t \le \tau$. By solving the inequality $R_d \le R^2$, we conclude that $\|s_{\boldsymbol{x}(0)}(t)\|_{\max} \le R$ for $t \in [0,\tau]$ holds if

$$\tau \le \frac{\ln R}{2\sqrt{n}nM'} \quad .$$

For the sensitivity matrix $s_{\boldsymbol{x}(0)}(t)$ with t ranging in the interval $[0,\tau]$, the diagonal element in the i-th row of the matrix $s_{\boldsymbol{x}(0}(t)$ is equal to

$$1+\left[\frac{\partial g_i(\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial x_{0,i}}\right]_{t=\xi_i} t,$$

the element in the i_{th} row and j_{th} column is equal to

$$\left[\frac{\partial g_i(\boldsymbol{x})}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial x_{0,j}}\right]_{t=\xi_j} t,$$

where $j\in\{1,\ldots,n\}\setminus\{i\}$ and ξ_l, for $l=1,\ldots,n$, is some value in $(0,\tau)$.

Thus $\Delta_i(s_{\boldsymbol{x}(0)}(t))$ is larger than

$$1-\tau\sum_{j=1}^{n}\left|\frac{\partial g_i(\boldsymbol{x}}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial x_{0,j}}\right|_{t=\xi_j},$$

which in turn is larger than $1-n^2M'R\tau$.

By solving the inequality $\frac{1}{1-n^2M'R\tau}\leq\varepsilon$, we obtain that $\tau\leq\frac{\varepsilon-1}{\varepsilon n^2M'R}$. Therefore, if

$$\tau\leq\min\left\{\frac{\varepsilon-1}{\varepsilon n^2M'R},\frac{\ln R}{2\sqrt{n}nM'}\right\},$$

then $\|s_{\boldsymbol{x}(0)}(t)\|_{\max}\leq R$ and $\max_{1\leq i\leq n}\frac{1}{\Delta_i(s_{\boldsymbol{x}(0)}(t))}\leq\varepsilon$ hold, and $s_{\boldsymbol{x}(0)}(t)$ is also diagonally dominant for $t\in[0,\tau]$ since $\tau\leq\frac{\varepsilon-1}{\varepsilon n^2M'R}$, $1-n^2M'R\tau>0$ holds.

□

Assume that the sensitivity matrix $s_{\boldsymbol{x}((k-1)\tau)}(t)$ is strictly diagonally dominant s.t.

$$\|s_{\boldsymbol{x}((k-1)\tau)}(t)\|_{max}\leq R, \tag{4.8}$$

$$max_{1\leq i\leq n}\frac{1}{\Delta_i(s_{\boldsymbol{x}((k-1)\tau)}(t))}\leq\varepsilon, \tag{4.9}$$

for any $t\in[(k-1)\tau,k\tau]$, where $k\in\{1,\ldots,K-1\}$, $\varepsilon>1$, and $R>1$. Then, we construct the bound constraint on the time-lag term τ as follows.

Lemma 4.2.3. *Based on Eq. (4.8) and (4.9), if the time-lag term is*

$$\tau \leq \min\left\{\frac{\varepsilon-1}{\varepsilon(n^2MR+n^2NR\varepsilon)}, \frac{\ln\frac{R^2+1}{2}}{\sqrt{n}(2nM+n^2NR\varepsilon)}\right\},$$

where M and N are presented in Constraint (4.2), then $s_{\boldsymbol{x}(k\tau)}(t)$ for $t\in[k\tau,(k+1)\tau]$ is strictly diagonally dominant with the property of

$$\|s_{\boldsymbol{x}(k\tau)}(t)\|_{\max} \leq R, \text{ and}$$

$$\max_{1\leq i\leq n}\frac{1}{\Delta_i(s_{\boldsymbol{x}(k\tau)}(t))} \leq \varepsilon.$$

Proof. Since the sensitivity matrix $s_{\boldsymbol{x}((k-1)\tau)}(t)$ is strictly diagonally dominant and Eq. (4.9) holds, the inequality

$$\|s^{-1}_{\boldsymbol{x}((k-1)\tau)}(t)\|_\infty \leq \varepsilon,$$

also holds, where $t\in[(k-1)\tau,k\tau]$ and $k\in\{1,\ldots,K-1\}$. Accordingly, this implies that $\|s^{-1}_{\boldsymbol{x}((k-1)\tau)}(t)\|_{\max}\leq\varepsilon$. This way, according to Lemma 4.2.1, the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t)$ for $t\in[k\tau,(k+1)\tau]$ with respect to the state $\boldsymbol{x}(k\tau)$ satisfies the sensitivity equation

$$\dot{s}_{\boldsymbol{x}(k\tau)} = \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}}s_{\boldsymbol{x}(k\tau)} + \frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau}\frac{\partial \boldsymbol{x}_\tau}{\partial \boldsymbol{x}(k\tau)}, \text{ with } s_{\boldsymbol{x}(k\tau)}(k\tau)=\boldsymbol{I}. \tag{4.10}$$

In the following, we employ the comparison principle for ODEs to derive a bound on the solution to Eq. (4.10).

Let

$$M_d = \max_{k\tau\leq t\leq(k+1)\tau}\sqrt{n}(2n\|\boldsymbol{A}(t)\|_{\max}+\|\boldsymbol{b}(t)\|_{\max}),$$

$$N_d = \max_{k\tau\leq t\leq(k+1)\tau}\sqrt{n}\|\boldsymbol{b}(t)\|_{\max},$$

where $\boldsymbol{A}(t)=\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}}$ and $\boldsymbol{b}(t)=\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau}\frac{\partial \boldsymbol{x}_\tau}{\partial \boldsymbol{x}((k-1)\tau)}\frac{\partial \boldsymbol{x}((k-1)\tau)}{\partial \boldsymbol{x}(k\tau)}$. It is obvious that $M_d\leq\sqrt{n}(2nM+n^2NR\varepsilon)$ and $N_d\leq\sqrt{n}n^2NR\varepsilon$.

We take the j_{th} column of the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t)$ and the matrix $\boldsymbol{b}(t)$ as a vector $\boldsymbol{y}(t)$ and $\boldsymbol{b}_j(t)$, where $j\in\{1,\ldots,n\}$. Let $u(t)=\|\boldsymbol{y}(t)\|_2^2=\langle\boldsymbol{y}(t),\boldsymbol{y}(t)\rangle$ with $u(k\tau)=1$, where $\|\boldsymbol{y}(t)\|_2$ is the 2-norm for $\boldsymbol{y}$ and $\langle\cdot,\cdot\rangle$ is an inner product in $\mathbb{R}^n$.

Based on Cauchy-Schwarz inequality and the fact that $2\|\boldsymbol{y}\|_2 \leq \|\boldsymbol{y}\|_2^2 + 1$ as well as $\|\boldsymbol{A}(t)\boldsymbol{y}\|_2 \leq \|\boldsymbol{A}(t)\|_F \|\boldsymbol{y}\|_2 \leq \sqrt{n}\|\boldsymbol{A}(t)\|_2 \|\boldsymbol{y}\|_2$, where $\|\boldsymbol{A}(t)\|_F$ is the Frobenius norm of the matrix $\boldsymbol{A}(t)$, we obtain

$$\begin{aligned}\dot{u} &= 2\langle \boldsymbol{y}, \dot{\boldsymbol{y}} \rangle \leq 2\|\boldsymbol{y}\|_2 \|\dot{\boldsymbol{y}}\|_2 = 2\|\boldsymbol{y}\|_2 \|\boldsymbol{A}(t)\boldsymbol{y} + \boldsymbol{b}_j(t)\|_2 \leq 2\sqrt{n}\|\boldsymbol{y}\|_2^2 \|\boldsymbol{A}(t)\|_2 + 2\|\boldsymbol{y}\|_2 \|\boldsymbol{b}_j(t)\|_2 \\ &\leq 2\sqrt{n}\|\boldsymbol{A}(t)\|_2 \|\boldsymbol{y}\|_2^2 + \|\boldsymbol{b}_j(t)\|_2 (\|\boldsymbol{y}\|_2^2 + 1) \leq M_d \|\boldsymbol{y}\|_2^2 + N_d = M_d u + N_d. \end{aligned} \tag{4.11}$$

Applying Gronwall's inequality [14] to Eq. (4.11), we deduce that

$$u(t) \leq u_0 e^{M_d(t-k\tau)} + \int_{k\tau}^{t} N_d e^{M_d(t-s)} ds = u_0 e^{M_d(t-k\tau)} + \frac{N_d}{M_d} e^{M_d(t-k\tau)} - \frac{N_d}{M_d} \leq R_d$$

for $k\tau \leq t \leq (k+1)\tau$, where $u_0 = u(k\tau) = 1$, and

$$R_d = \left(1 + \frac{N_d}{M_d}\right) e^{M_d \tau} - \frac{N_d}{M_d}.$$

Therefore, $\|\boldsymbol{y}(t)\|_2^2 \leq R_d$ for $k\tau \leq t \leq (k+1)\tau$. By solving the inequality $R_d \leq R^2$, we conclude that $\|s_{\boldsymbol{x}(k\tau)}(t)\|_{\max} \leq R$ for $t \in [k\tau, (k+1)\tau]$ holds if

$$\tau \leq \frac{\ln \frac{R^2+1}{2}}{\sqrt{n}(2nM + n^2 NR\varepsilon)} \ ,$$

where the right side of this inequality could be gained when $M_d = N_d$.

For the sensitivity matrix $s_{\boldsymbol{x}(k\tau)}(t)$ with t ranging in the interval $[k\tau, (k+1)\tau]$, the diagonal element in the i-th row of the matrix $s_{\boldsymbol{x}(k\tau)}(t)$ is equal to

$$1 + \left[\frac{\partial f_i(\boldsymbol{x}, \boldsymbol{x}_\tau)}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial x_{k\tau,i}} + \frac{\partial f_i(\boldsymbol{x}, \boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau} \frac{\partial \boldsymbol{x}_\tau}{\partial x_{k\tau,i}}\right]_{t=\xi_i} (t - k\tau),$$

the element in the i_{th} row and j_{th} column is equal to

$$\left[\frac{\partial f_i(\boldsymbol{x}, \boldsymbol{x}_\tau)}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{x}}{\partial x_{k\tau,j}} + \frac{\partial f_k(\boldsymbol{x}, \boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau} \frac{\partial \boldsymbol{x}_\tau}{\partial x_{k\tau,j}}\right]_{t=\xi_j} (t - k\tau),$$

where $j \in \{1, \ldots, n\} \setminus \{i\}$ and ξ_l, for $l = 1, \ldots, n$, is some value in $(k\tau, (k+1)\tau)$.

Thus $\Delta_i(s_{\boldsymbol{x}(k\tau)}(t))$ is larger than

$$1-\tau\sum_{j=1}^{n}\left|\frac{\partial f_i(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}}\frac{\partial \boldsymbol{x}}{\partial x_{k\tau,j}}+\frac{\partial f_i(\boldsymbol{x},\boldsymbol{x}_\tau)}{\partial \boldsymbol{x}_\tau}\frac{\partial \boldsymbol{x}_\tau}{\partial x_{k\tau,j}}\right|_{t=\xi_j},$$

which in turn is larger than $1-(n^2MR+n^2NR\varepsilon)\tau$.

By solving the inequality $\frac{1}{1-(n^2MR+n^2NR\varepsilon)\tau}\leq\varepsilon$, we obtain that $\tau\leq\frac{\varepsilon-1}{\varepsilon(n^2MR+n^2NR\varepsilon)}$. Therefore, if

$$\tau\leq\min\left\{\frac{\varepsilon-1}{\varepsilon(n^2MR+n^2NR\varepsilon)},\frac{\ln\frac{R^2+1}{2}}{\sqrt{n}(2nM+n^2NR\varepsilon)}\right\},$$

then $\|s_{\boldsymbol{x}(k\tau)}(t)\|_{\max}\leq R$ and $\max_{1\leq i\leq n}\frac{1}{\Delta_i(s_{\boldsymbol{x}(k\tau)}(t))}\leq\varepsilon$ hold, and $s_{\boldsymbol{x}(k\tau)}(t)$ is also diagonally dominant for $t\in[k\tau,(k+1)\tau]$ since $\tau\leq\frac{\varepsilon-1}{\varepsilon(n^2MR+n^2NR\varepsilon)}$, $1-(n^2MR+n^2NR\varepsilon)\tau>0$ holds.

□

Combining Lemma 4.2.2 and Lemma 4.2.3, we deduce the following theorem.

Theorem 4.2.1. *If the time-lag term of DDE (4.1) is*

$$\tau\leq\min\left\{\frac{\varepsilon-1}{\varepsilon n^2M'R},\frac{\ln R}{2\sqrt{n}nM'},\frac{\varepsilon-1}{\varepsilon(n^2MR+n^2NR\varepsilon)},\frac{\ln\frac{R^2+1}{2}}{\sqrt{n}(2nM+n^2NR\varepsilon)}\right\},$$

then the solution mapping $\boldsymbol{\phi}(t;\cdot):\mathcal{I}_0\mapsto\Omega(t;\mathcal{I}_0)$ to System (4.1) is a homeomorphism between two topological spaces $\mathcal{I}_0$ and $\Omega(t;\mathcal{I}_0)$ for any $t\in[0,K\tau]$.

When the time-lag τ satisfies the condition presented in Theorem 4.2.1, the homeomorphism property in Theorem 4.2.1 implies that the solution mapping $\boldsymbol{\phi}(t;\cdot):\mathcal{I}_0\mapsto\Omega(t;\mathcal{I}_0)$ to System (4.1), where $t\in[0,K\tau]$, maps the boundary and interior points of the initial set $\mathcal{I}_0$ onto the boundary and interior points of the set $\Omega(t;\mathcal{I}_0)$ respectively. Therefore, the full reachable set induced by the initial set of System (4.1) could be retrieved by computing the reachable set just of the initial set's boundary. We illustrate Theorem 4.2.1 through the following example involving a delay τ that could be caused by sensor circuitry. Determining a bound on that delay could thus help facilitate the choice of appropriate sensors such that the delay τ incurred satisfies the conditions of Theorem 4.2.1.

Example 4.2.1. *Consider a modified model of an electromechanical oscillation of a synchronous machine,*

$$\dot{\boldsymbol{x}} = \begin{cases} \boldsymbol{g}(\boldsymbol{x}), & \text{if } t \in [0,\tau), \boldsymbol{x}(0) \in \mathcal{I}_0 \\ \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau), & \text{if } t \in [\tau, K\tau], \end{cases} \tag{4.12}$$

with $\boldsymbol{x} = (\delta, w)'$, $\boldsymbol{x}_\tau = (\delta_\tau, w_\tau)$, $\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}))' = (0,0)'$, $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau) = (f_1(\boldsymbol{x},\boldsymbol{x}_\tau), f_2(\boldsymbol{x},\boldsymbol{x}_\tau))' = (w, 0.2 - 0.7 sin\delta_\tau - 0.05 w_\tau)'$, *and* $\mathcal{I}_0 = [-0.5, 0.5] \times [2.5, 3.5]$, $K = 60$ *and* $\mathcal{X} = [-100, 100] \times [-100, 100]$. *Through simple calculations, we obtain that* $M' = 0, M = 1$, $N = 0.7$, $R = 2.5$ *and* $\varepsilon = 2.5$, *thus any* $\tau \leq 0.0218$ *satisfies the condition in Theorem 4.2.1. In our experiments, we set* $\tau = 0.02$.

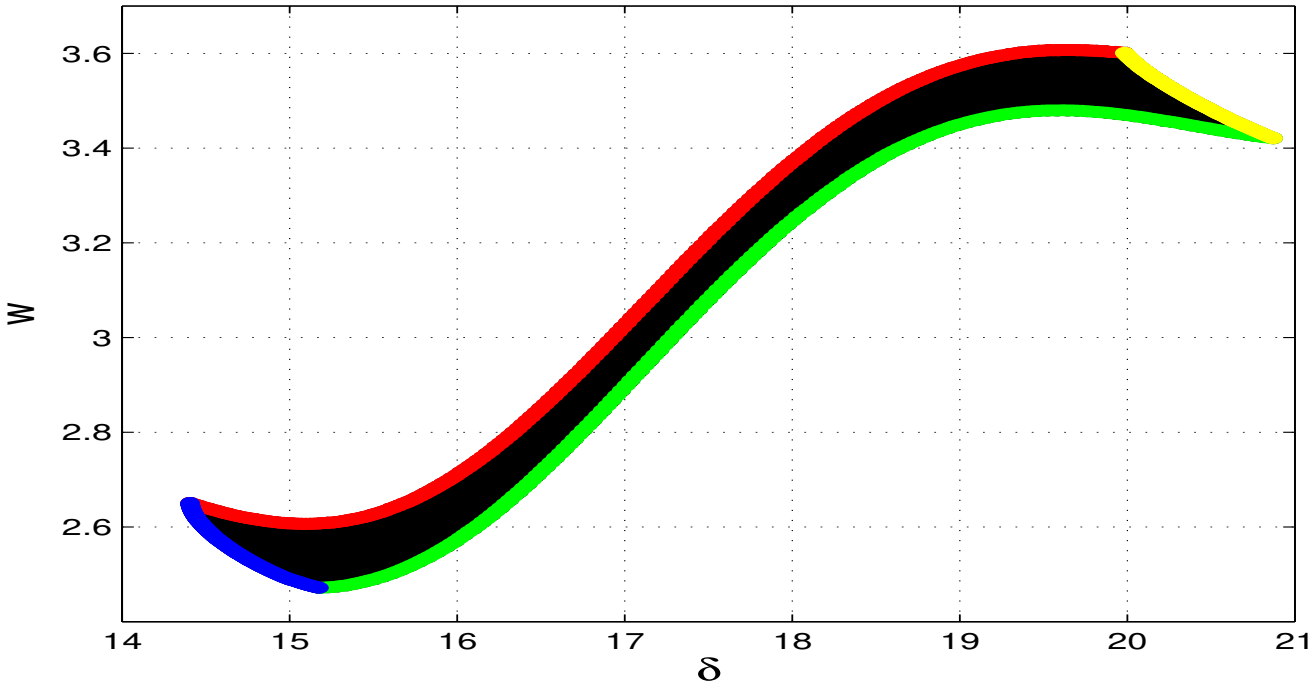

Fig. 4.1 An illustration of the reachable set for Example 4.2.1 at time $t = 6.0$ using simulation methods, (red, green, blue and yellow points – the approximate sampling states reachable from the boundary subsets $[-0.5,-0.5] \times [2.5,3.5]$, $[0.5,0.5] \times [2.5,3.5]$, $[-0.5,0.5] \times [2.5,2.5]$ and $[-0.5,0.5] \times [3.5,3.5]$ respectively; black points – the approximate sampling states reachable from the entire initial set).

From the result illustrated in Fig. 4.1, we conclude that the corresponding solution mapping $\boldsymbol{\phi}(6;\cdot) : \mathcal{I}_0 \mapsto \Omega(6;\mathcal{I}_0)$ maps the boundary and interior points of the initial

set $\mathcal{I}_0$ onto the boundary and interior points of the set $\Omega(6;\mathcal{I}_0)$ respectively, as the homeomorphism property suggests.

Based on this insight, in the section to follow, we present a reachability algorithm for our method. This algorithm is resting on techniques for computing reachable sets for nonlinear control systems in order to construct over- and under-approximations of the reachable set for System (4.1) with a time-lag τ that satisfies the conditions of Theorem 4.2.1.

4.2.3 Constructing Reachable Sets

In this section, the reachability analysis method based on set-boundary of ODEs, i.e., introduced in [155, 157] for nonlinear control systems, is extended to reachability computations of System (4.1) with a time-lag τ satisfying the conditions of Theorem 4.2.1. The reduction is based on considering the delayed state variable $\boldsymbol{x}_\tau$ in System (4.1) as a control input $\boldsymbol{u}(t)$, and the confinement to set boundaries adds precision as it significantly reduces the volume of the tube containing all such input trajectories $\boldsymbol{x}_\tau$. Note that, in our algorithm we obviously restrict the initial set $\mathcal{I}_0$ to a specific family of computer-representable sets in the $\mathbb{R}^n$ such as polytopes [18].

Assume that the initial set's boundary can be represented as union of m subsets from the respective family, i.e.,

$$\partial\mathcal{I}_0 = \cup_{i=1}^{m} I_{0,i}.$$

For $t \in [0,\tau]$, the system is governed by ODE $\dot{\boldsymbol{x}} = \boldsymbol{g}(\boldsymbol{x})$. Therefore, we can apply any existing reachability analysis technique for ODEs that is able to deal with reachability computations with initial sets of forms, such as polytopes, to the computation of an enclosure $\mathcal{B}_{0,t}$ of the reachable set for the initial set's boundary $\partial\mathcal{I}_0$ at time $t \in [0,\tau]$, where $\mathcal{B}_{0,t} = \cup_{i=1}^{m} B_{0,i}(t)$ and $B_{0,i}(t)$ is an over-approximation of the reachable set at time $t \in [0,\tau]$ starting from the set $I_{0,i}$, for $i = 1,\ldots,m$. The corresponding over- and under-approximations of the reachable set at time t could be constructed by including (excluding, resp.) the set $\mathcal{B}_{0,t}$ from the set obtained from convex combinations of points in $B_{0,i}(t)$, according to [157].

Based on these computations for the initial trajectory segment up to time τ, for $t \in [k\tau,(k+1)\tau]$, $k = 1,\ldots,K-1$, the following steps are used to compute its corresponding over- and under-approximations of the reachable set respectively.

1. First, we compute an enclosure $B_{k,i}(t)$, for $t \in [k\tau,(k+1)\tau]$, of the reachable set $\Omega(t;I_{0,i})$ for System (4.1) with the initial set $B_{k-1,i}(k\tau)$ and $\boldsymbol{x}_\tau \in B_{k-1,i}(t-\tau)$. This enclosure can be computed by employing reachability analysis methods for nonlinear control systems of the form (4.3) with a time-varying input

$$\boldsymbol{u}(t) = \boldsymbol{x}_\tau \in B_{k-1,i}(t-\tau).$$

   Therefore, $\mathcal{B}_{k,t} = \cup_{i=1}^{m} B_{k,i}(t)$ is an enclosure of the reachable set for the initial set's boundary $\partial\mathcal{I}_0$ at time $t \in [k\tau,(k+1)\tau]$.

2. Then, we construct a simply connected compact polytope $O_{k,t}$ such that it covers $\mathcal{B}_{k,t}$. The set $O_{k,t}$ is an over-approximation of the reachable set $\Omega(t;\mathcal{I}_0)$ at time $t \in [k\tau,(k+1)\tau]$ according to Lemma 1 in [157].

3. Finally, we construct a simply connected polytope $U_{k,t}$ that satisfies two conditions: 1) the enclosure of the reachable set from the boundary of the initial set, i.e., $\mathcal{B}_{k,t}$, is obtained to be a subset of the enclosure of its complement, and 2) it intersects the interior of the reachable set $\Omega(t;\mathcal{I}_0)$. Then, according to Lemma 2 in [157], $U_{k,t}$ is an under-approximation of the reachable set $\Omega(t;\mathcal{I}_0)$ at time $t \in [k\tau,(k+1)\tau]$.

4.3 Examples

In this section, we test our method on two examples of a two-dimensional system and a seven-dimensional system. Our implementation is based on Althoff's *continuous reachability analyzer (CORA)* [2], which is a MATLAB toolbox for prototype design of algorithms for reachability analysis. All computations are carried out on an i5-3337U 1.8GHz CPU with 4GB running Ubuntu Linux 13.10.

Example 4.3.1. *Consider a modified Lotka-Volterra two-variables system with a delay τ, given by*

$$\dot{\boldsymbol{x}} = \begin{cases} \boldsymbol{g}(\boldsymbol{x}), & \text{if } t \in [0,\tau), \boldsymbol{x}(0) \in \mathcal{I}_0 \\ \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau), \text{if } t \in [\tau, K\tau] \end{cases} \tag{4.13}$$

with $\boldsymbol{x} = (x,y)'$, $\boldsymbol{x}_\tau = (x_\tau, y_\tau)'$, $\boldsymbol{g}(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}))' = (y, -0.2x + y - 0.2x^2y)'$, $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau) = (f_1(\boldsymbol{x},\boldsymbol{x}_\tau), f_2(\boldsymbol{x},\boldsymbol{x}_\tau))' = (y, -0.2x_\tau + y - 0.2x^2y)'$, $\mathcal{I}_0 = [0.9,1.1] \times [0.9,1.1]$ *with* $\partial\mathcal{I}_0 = \cup_{i=1}^{4} I_{0,i}$ *and* $\mathcal{X} = [0.5,3.5] \times [0.2,1.5]$, *where* $I_{0,1} = [0.9,0.9] \times [0.9,1.1]$, $I_{0,2} = [1.1,1.1] \times [0.9,1.1]$, $I_{0,3} = [0.9,1.1] \times [0.9,0.9]$ *and* $I_{0,4} = [0.9,1.1] \times [1.1,1.1]$.

In this example, the valuations $M' = 2.3, M = 2.10, N = 0.2$, $R = 2$ and $\varepsilon = 2$ fulfill the condition in Lemma 4.2.3. Through simple calculations, $\tau = 0.01$ satisfies the requirement in Theorem 4.2.1. Also, K is assigned to 100, i.e. the entire time interval is $[0,1.0]$. The over- and under-approximation of the reachable set illustrated in Fig. 4.2 and 4.3 are represented by polytopes. The computation time for computing over- and under-approximations is 111.56 seconds.

Example 4.3.2. *Consider a seven-dimensional system with a delay τ*[4]*,*

$$\dot{\boldsymbol{x}} = \begin{cases} \boldsymbol{g}(\boldsymbol{x}), & \text{if } t \in [0,\tau), \boldsymbol{x}(0) \in \mathcal{I}_0 \\ \boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau), \text{if } t \in [\tau, K\tau] \end{cases} \tag{4.14}$$

with $\boldsymbol{x} = (x_1,\ldots,x_7)'$, $\boldsymbol{x}_\tau = (x_{1,\tau},\ldots,x_{7,\tau})'$, $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{0}$, $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{x}_\tau) = (1.4x_3 - 0.9x_{1,\tau}, 2.5x_5 - 1.5x_2, 0.6x_7 - 0.8x_3x_2, 2.0 - 1.3x_4x_3, 0.7x_1 - 1.0x_4x_5, 0.3x_1 - 3.1x_6, 1.8x_6 - 1.5x_7x_2)'$, $\mathcal{I}_0 = [1.1,1.3] \times [0.95,1.15] \times [1.4,1.6] \times [2.3,2.5] \times [0.9,1.1] \times [0.0,0.2] \times [0.35,0.55]$ *and* $\mathcal{X} = [0.5,1.5] \times [0.5,1.5] \times [1.0,2.0] \times [2.0,3.0] \times [0.5,1.5,] \times [0.0,0.5] \times [0.0,1.0]$.

The valuations $M' = 0, M = 3.9, N = 0.9$, $R = 2$ and $\varepsilon = 9$ fulfill the condition in Lemma 4.2.3. Thus, $\tau \leq 0.01$ satisfies the requirement in Theorem 4.2.1. Also, τ and K are assigned to 0.01 and 30 respectively, i.e., the entire time interval is $[0,0.03]$.

The computed over-approximation at time instant 0.03 is $O(0.03;\mathcal{I}_0) = [1.121,1.336] \times [0.971,1.178] \times [1.368,1.575] \times [2.221,2.430] \times [0.859,1.057] \times [0.009,0.194] \times [0.332,0.538]$. The computed under-approximation at time instant 0.003 is $U(0.03;\mathcal{I}_0) = [1.141,1.317] \times [0.991,1.159] \times [1.387,1.555] \times [2.241,2.411] \times [0.878,1.037] \times [0.028,0.175] \times [0.351,0.519]$.

[4]The delay-free system could be found in the Package CORA.

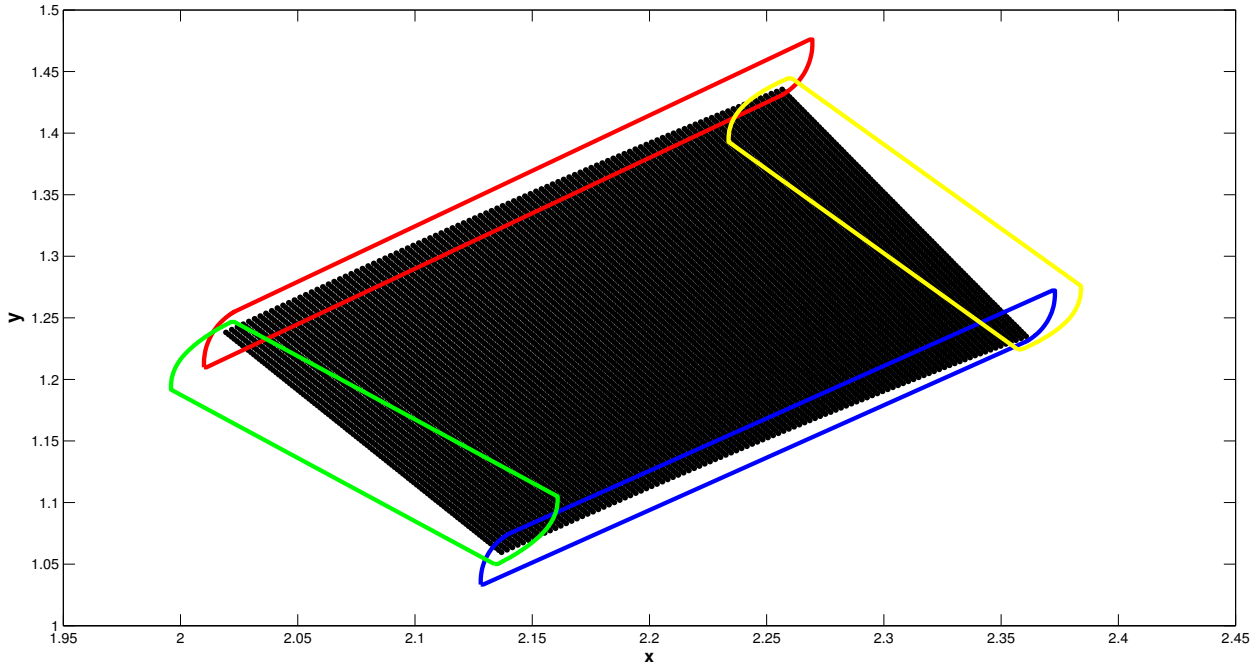

Fig. 4.2 An illustration of the reachable set of the initial set's boundary for Example 4.3.1 at time $t = 1.0$, (red curve – $\partial O(1.0; I_{0,1})$; blue curve – $\partial O(1.0; I_{0,2})$; green curve – $\partial O(1.0; I_{0,3})$; yellow curve– $\partial O(1.0; I_{0,4})$; black points – the approximate sampling states reachable from the initial set $\mathcal{I}_0$ after time duration of 1.0, which are computed using simulation methods).

The computation time for both is 900.23 seconds. The projections for over-and under-approximations at time instants $t = 0.01, 0.02, 0.03$ on the $x_1 - x_2$ space are illustrated in Fig. 4.4.

4.4 Discussion

Fig. 4.2 presents the approximation of the reachable set's boundary obtained by applying numerical simulation methods along with the set-boundary based method to Example 4.3.1. Our findings would seem to show that the set-boundary based method is able to produce a valid over-approximation of the reachable set's boundary when the delay-lag term τ satisfies the conditions in Theorem 4.2.1. Furthermore, it is shown in Fig. 4.3 that the set-boundary based method, as discussed in Subsection 4.2.3, is

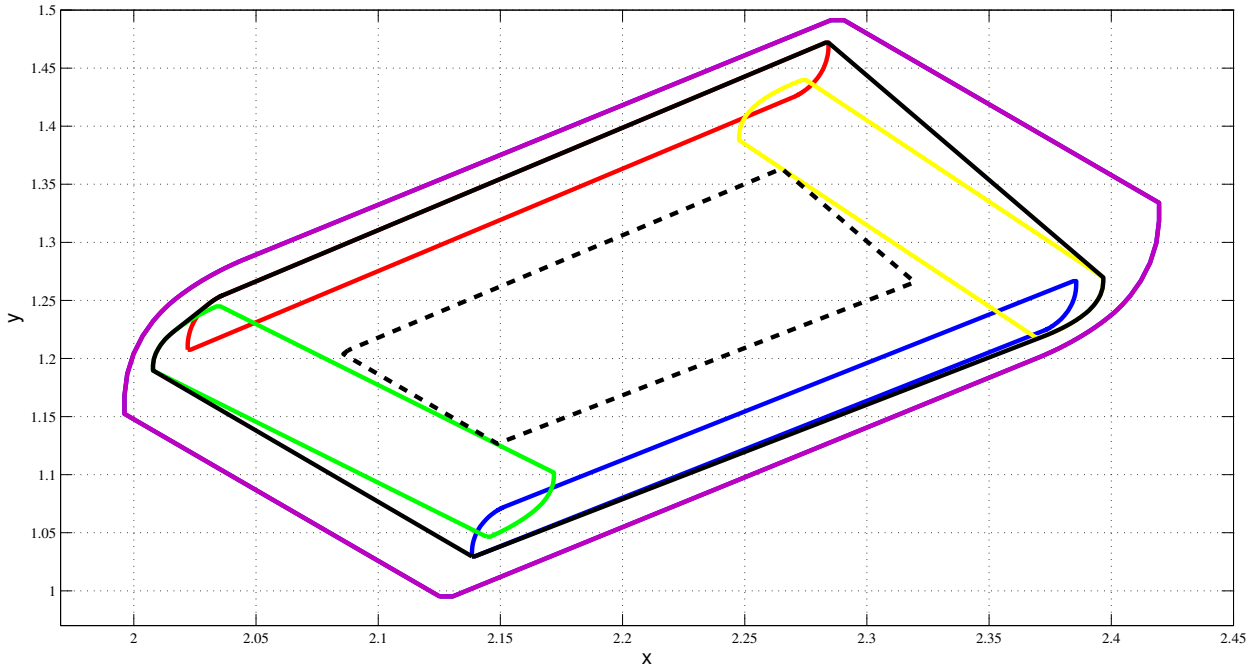

Fig. 4.3 An illustration of the reachable set of the initial set's boundary for Example 4.3.1 at time $t = 1.0$, (red curve – $\partial O(1.0; I_{0,1})$; blue curve – $\partial O(1.0; I_{0,2})$; green curve – $\partial O(1.0; I_{0,3})$; yellow curve – $\partial O(1.0; I_{0,4})$; black curve – boundary $\partial O(1.0; \mathcal{I}_0)$ of the over-approximation obtained by our boundary method; black dash curve – boundary $\partial U(1.0; \mathcal{I}_0)$ of the under-approximation obtained by our boundary method; purple curve – boundary $\partial O(1.0; \mathcal{I}_0)$ of less tight over-approximation obtained by extrapolating the entire initial set rather than its boundaries).

able to output validated over- and under-approximations of the reachable sets. Also, from our findings in Fig. 4.3, we argue that the set-boundary based method induces a smaller wrapping effect in performing reachability analysis compared with extrapolating the entire initial set for the reason that the boundaries of the initial set have definitely much smaller volume than the entire initial set.

For Example 4.3.2, the approximations of the interval form, as illustrated in Fig. 4.4, are computed for the sake of reducing computational burden. Note that the bound imposed for maintaining homeomorphism property applies to the time-lag in the DDE only and is not a bound on the temporal horizon coverable by reach-set computation, which can be arbitrarily larger if only the time-lag suits the condition. The relatively

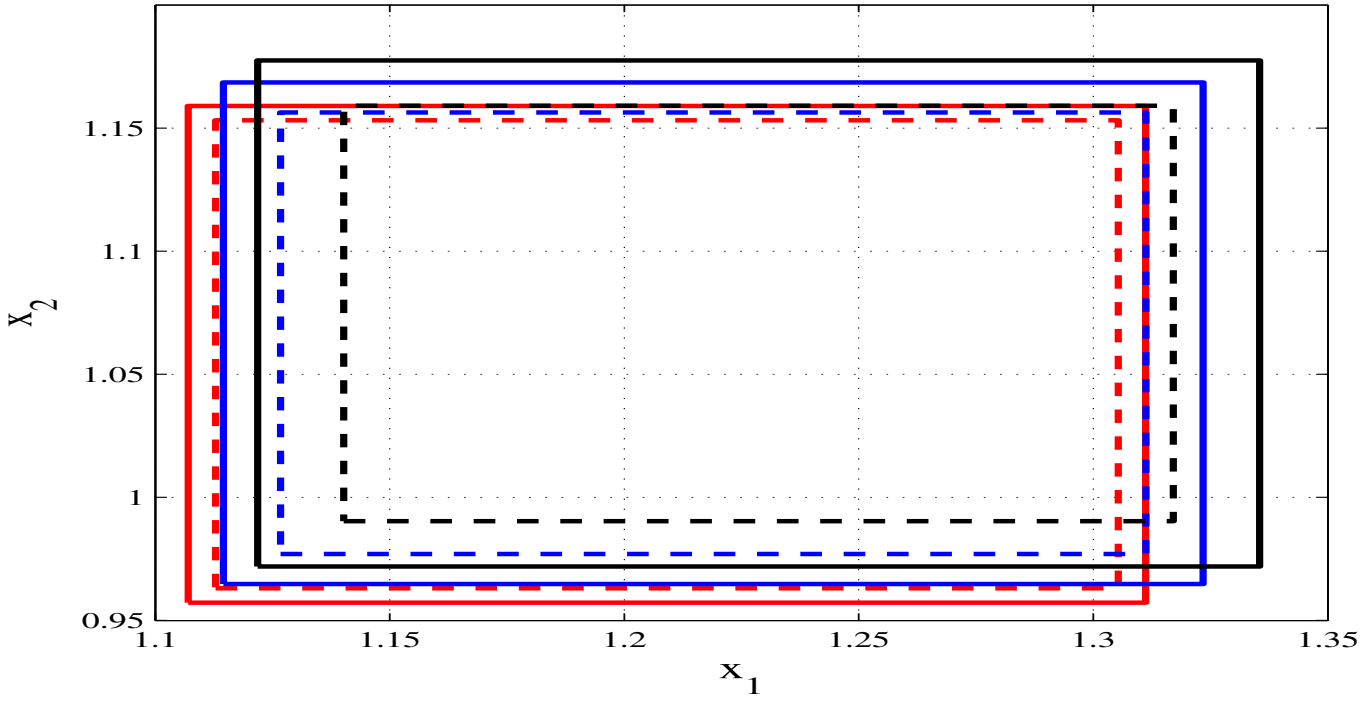

Fig. 4.4 An illustration of the reachable set on the $x_1 - x_2$ space for Example 4.3.2 at times $t = 0.01, 0.02, 0.03$, (red, blue and black solid lines – the boundaries of over-approximations on the $x_1 - x_2$ space at time instants $t = 0.01, 0.02, 0.03$ respectively; red, blue and black dashed lines – the boundaries of under-approximations on the $x_1 - x_2$ space at time instants $t = 0.01, 0.02, 0.03$ respectively).

small horizons in these examples are due to the wrapping effect in the underlying reachability techniques, not the method itself, as discussed below.

There is evidence to support the hypothesis that the positive aspect induced by using polytopes for representations is that they enable the analysis of some properties such as safety and reliability by reasoning in the theory of linear arithmetic. On the other side, they might not be the best representations of the reachable sets for nonlinear systems for the reason that the reachable sets of nonlinear systems modeled by ODEs and DDEs may be far from being convex as demonstrated in Fig. 4.1, thereby generating poor results when employing polytopes to characterize the reachable sets. In order to address the issue of conservativeness induced by polytopes, we will try to employ representations of more complex shapes such as semi-algebraic sets in the construction of the reachable sets at the expense of computational efficiency.

Another undesirable feature that might be in our implementation, is due to the excessive use of previous state information to compute the set of current reachable states from the boundaries of the initial set. In a sense, while computing the set of reachable states at time $t \in [k\tau,(k+1)\tau]$, the entire reachable set of the past states within the time interval $[(k-1)\tau,k\tau]$ is used for the computations rather than the set of reachable states at just time instant $t-\tau$. Therefore, a large amount of spurious states might be introduced that are not actually reachable at previous time from the boundaries of the initial set. Consequently, it is not surprising that this causes the wrapping effect to increase. Due to constructing over- and under-approximations by including (excluding, resp.) the obtained boundary enclosure from certain convex combination of points, a pessimistic over-approximation of the reachable sets from the boundaries of the initial set may reduce the tightness of computed results accordingly. In order to circumvent this issue, we will extend Taylor-model based reachability analysis for ODEs to the proposed class of DDEs in the future work. Since Taylor models are functions being explicitly dependent on time and state variables, this dependence enables the use of an over-approximation associated with the reachable sets of the boundaries of the initial set at previous time $t-\tau$ rather than within the time interval $[(k-1)\tau,k\tau]$ to over-approximate the set of states reachable from the boundaries of the initial set at current time $t \in [k\tau,(k+1)\tau]$, thereby resulting in a significant reduction in the wrapping effect.

Finally, we should point out that our method is suitable for systems modeled by DDEs of the form (4.1) with solutions having homeomorphism property. But, it is restricted to a class of DDEs with time-lag term τ satisfying the conditions in Theorem 4.2.1. As a future work, we will expand such class of systems by loosing bound constraints on τ. Also, in order to measure the conservativeness on such bounds, we plan to deduce constraints on τ such that the solution to the associated system does not equip with homeomorphism property. Besides, if such homeomorphism property fails, one feasible solution to compute its over- and under-approximations of reachable sets is first to reformulate the associated DDE as an ODE via the method of steps in [146] and then apply the reachability method based on set-boundary in [155, 157] to the obtained ODE. However, the formulated ODE suffers an increase of space dimension over reachability time of interest. We will investigate more about this in future work.

Chapter 5

Temporal Logic Verification for a Class of DDEs

> *"In mathematics, logic is static. It deals with connections among entities that exist in the same time frame. When one designs a dynamic computer system that has to react to ever changing conditions, [...] one cannot design the system based on a static view. It is necessary to characterize and describe dynamic behaviors that connect entities, events, and reactions at different time points. Temporal Logic deals therefore with a dynamic view of the world that evolves over time."*
>
> [Amir Pnueli[1]]

Until now, the presentation of our work in Chapter 3 and Chapter 4 seems to be confined to reducing safety verification problem to reachability problem (i.e., invariance properties) that may restrict the ability of adequately express the desired safe behavior of the system. This makes some controversy surrounding if we able to use the presented methods to consider a wider class of safety properties. In the original papers on which this chapter is based, [103] and its extended revised version [104], we have extended the safety properties by involving a number of critical properties such as timing requirements and bounded response rather than just invariance properties employing the framework of interval-based Taylor over-approximation method introduced in [162] and reviewed in Chapter 3.

[1] To quote Amir Pnueli from his talk after receiving the Israel Prize, https://amturing.acm.org/award_winners/pnueli_4725172.cfm.

5.1 Problem Formulation

In this section, we formulate the verification problem of a simple class of DDEs in the form of Eq. (3.3)

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = f(\vec{x}(t-\tau))$$

against a class of safety requirements specified using an appropriate linear-time temporal logic. As we deal with continuous state and time, we adopt metric interval temporal logic (MITL) [4] for the purpose of requirements specification language. In this section, we review its syntax and its continuous-time, signal-based semantics.

Let $\mathbb{R}$ be the set of the real numbers. Our time domain is the set of nonnegative real numbers $\mathbb{R}_{\geq 0}$. Also, the trajectory of the DDE of Eq. (3.3) on an initial condition $x([0,\tau]) \equiv c \in \mathbb{R}$ is a function $x(t)$ such that $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ satisfies the initial condition and $\forall t \geq \tau : \frac{\mathrm{d}}{\mathrm{d}t}\vec{x}(t) = f(\vec{x}(t-\tau))$, where the positive integer N denotes the dimension of the state space. In order to specify the temporal properties of interest, we exploit MITL with continuous semantics, as meaningful when the states evolve in metric spaces like in Eq. (3.3). We say that $\mathcal{P}(\mathcal{C})$ denotes the powerset of a set $\mathcal{C}$ and assume that AP is a set of atomic propositions. Then, the predicate mapping $\mathcal{M} : AP \to \mathcal{P}(\mathbb{R}^N)$ is a set valued function that assigns to each atomic proposition $\rho \in AP$ a set of states $\mathcal{M}(\rho) \subseteq \mathbb{R}^N$. In this paper, we take the set of atomic propositions AP to be bound constraints $e \sim c$ on state expressions, where e is an expression formed over the state variables, like $x_1 x_2 - 2\sin x_3$, and being compared via a relation $\sim \in \{<, \leq, >, \geq\}$ to a constant $c \in \mathbb{Q}$. Such atomic propositions come equipped with their natural semantics.

5.1.1 Metric Interval Temporal Logic

Metric interval temporal logic (MITL) [4] is a linear-time temporal logic designed for capturing properties of signals evolving over quantitative and thus metric rather than qualitative time, an assumption met by continuous-state systems as in Eq. (3.3). It is a real-time extension of linear temporal logic (LTL), i.e., discussed in Chapter 2, where the modalities of LTL are constrained with quantitative timing bounds. Metric temporal logic (MTL) was first introduced by Koymans [82] to specify real-time properties. In

order to address the undecidability problem of MTL, Alur et al. in [4] relaxed the punctuality of the temporal operators s.t. they cannot constrain to singleton intervals. We employ MITL to formally characterize the desired behavior of DDEs. Along the following lines, we review and suitably adapt the syntax and the continuous-time and continuous-space semantics of MITL as presented in [4, 115].

Definition 5.1.1. ***(Syntax of MITL).*** *An MITL formula φ is built from a set of atomic propositions AP using Boolean connectives and timed-constrained versions of the until operator. It is inductively defined according to the grammar*

$$\varphi ::= \top \mid \rho \mid \neg\varphi_1 \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{\mathcal{I}} \, \varphi_2$$

where $\rho \in AP$, $\top$ is the Boolean constant true and $\mathcal{I} \subseteq \mathbb{Q}_{\geq 0}$ is a nonsingular interval imposing timing bounds on the temporal operators, where $\mathbb{Q}_{\geq 0}$ is the set of non-negative rational numbers.

We can derive the constant *false* by $\bot \equiv \neg\top$. Also, we can define additional, time-constrained version of, temporal operators such as *release* $\mathcal{R}_{\mathcal{I}}$, *eventually* $\Diamond_{\mathcal{I}}$, and *always* $\Box_{\mathcal{I}}$ as follows:

$$\begin{aligned} \varphi_1 \ \mathcal{R}_{\mathcal{I}} \ \varphi_2 &\equiv \neg((\neg\varphi_1) \ \mathcal{U}_{\mathcal{I}} \ (\neg\varphi_2)), \\ \Diamond_{\mathcal{I}} \ \varphi &\equiv \top \ \mathcal{U}_{\mathcal{I}} \ \varphi, \text{ and} \\ \Box_{\mathcal{I}} \ \varphi &\equiv \bot \ \mathcal{R}_{\mathcal{I}} \ \varphi \equiv \neg\Diamond_{\mathcal{I}} \ \neg\varphi. \end{aligned}$$

Notice that the *release* operator is a temporal modality that is dual to the *until* operator. A formula $\varphi_1 \ \mathcal{R}_{\mathcal{I}} \ \varphi_2$ holds if φ_2 always holds, a requirement that is released as soon as φ_1 becomes valid with respect to the time bounds $\mathcal{I}$.

Also, note that MITL has no *next* operator as the time domain is dense. For $\mathcal{I} = [0, \infty]$, we can remove the subscript $\mathcal{I}$ from the temporal operators, obtaining the traditional modalities of LTL. Another notice, we would like to point out that the decidability problem of MITL in the continuous semantics for both model checking and satisfiability problems is out of the scope of this dissertation. For details about the decidability problem, we refer the reader to [4, 114]. Furthermore, it is an open issue whether the model property of DDE with respect to MITL formulae is decidable.

Negation Normal Form

We consider MITL formulae in *negation normal form (NNF)*, which can be achieved by pushing all negations inside into the atoms [135]. If we admit the *release* modality and disjunction in our syntax, then every formula φ has a semantically equivalent negation normal form $nnf(\varphi)$. Such an NNF can be obtained by applying *De Morgan's laws* as well as the dualities between *until* and *release* in order to push negations inwards, and thereafter eliminating double negations. This is done by exploiting the following equivalences as rewrite rules from left to right:

$$\begin{aligned}
\neg\neg\varphi_1 &\equiv \varphi_1,\\
\neg(\varphi_1 \wedge \varphi_2) &\equiv \neg\varphi_1 \vee \neg\varphi_2,\\
\neg(\varphi_1 \vee \varphi_2) &\equiv \neg\varphi_1 \wedge \neg\varphi_2,\\
\neg(\varphi_1 \ \mathcal{U}_{\mathcal{I}} \ \varphi_2) &\equiv \neg\varphi_1 \ \mathcal{R}_{\mathcal{I}} \ \neg\varphi_2,\\
\neg(\varphi_1 \ \mathcal{R}_{\mathcal{I}} \ \varphi_2) &\equiv \neg\varphi_1 \ \mathcal{U}_{\mathcal{I}} \ \neg\varphi_2.
\end{aligned}$$

These rewrite rules can also be lifted to the derived operators as follows:

$$\begin{aligned}
\neg\Diamond_{\mathcal{I}} \ \varphi &\equiv \Box_{\mathcal{I}} \ \neg\varphi,\\
\neg\Box_{\mathcal{I}} \ \varphi &\equiv \Diamond_{\mathcal{I}} \ \neg\varphi.
\end{aligned}$$

Continuous-Time, Continuous-State Semantics of MITL

The continuous semantics of MITL formulae is used to express specifications on the desired temporal evolution to the solutions of DDEs in the form of Eq. (3.3). This semantics is based on real-valued signals $x : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ over time. We say that expression e over the state variables x satisfies atomic formula $e \sim c$ at time $t \geq 0$, denoted $e,t \models e \sim c$, iff $e(t) \sim c$ holds. Based on this, semantics of arbitrary MITL formulae is defined inductively, with the semantics of Boolean connectives $\neg$ and $\wedge$ as well as the constant $\top$ being standard. The semantics of the time-constrained *until* operator is defined as follows: $e,t \models \varphi_1 \ \mathcal{U}_{\mathcal{I}} \ \varphi_2$ iff for some $t' \in \mathcal{I}$, $e,t+t' \models \varphi_2$ holds and furthermore $e,t \models \varphi_1$ for all $t \in (t,t+t')$.

By convention, we say that the DDE of Eq. (3.3) with an initial value $x([0,\tau]) \equiv c$ satisfies an MITL formula φ if the expression $e(t)$ over its solution trajectory satisfies φ in the sense of $e,0 \models \varphi$. In what follows, we employ the interval-based Taylor over-approximation method, introduced in [162] and revisited in Chapter 3, to enclose the solution of such a DDE. As this method factually generates a discrete sequence of Taylor coefficients rather than a continuous trajectory, we are thus able to reduce a correctness problem over continuous time into a corresponding problem of a time-invariant operator over discrete time. Therefore, it is however necessary to recover the continuous semantics on the actual solution of the DDE from the timed state sequence semantics on the Taylor coefficients.

5.1.2 Bounded Model Checking Mode in `iSAT3`

In order to encode the Taylor model corresponding to a DDE as discussed in Chapter 3, we use bounded model checking (BMC) mode in `iSAT3` [138]. The `iSAT3` solver is a satisfiability checker for Boolean combinations of arithmetic constraints over real- and integer-valued variables as well as a bounded model-checker for transition systems over the same fragment of arithmetic. It is a stable version implementation of the iSAT algorithm [52]. The solver can efficiently solve bounded verification problems that involve polynomial (and, if needed, transcendental) arithmetic. Hence, it is a good option to solve our proposed problem due to the Taylor forms involved. Also, it allows us to verify/falsify a variety of MITL formulae built on atomic predicates defined over simple bounds, linear, and nonlinear constraints [84]. Bounded model checking (BMC) of a transition system aims at finding a run of bounded length k_{depth} which

- starts in an initial state of the system,
- complies with the system's transition relation, and
- ends in a state in which a certain (un)desired property holds.

The bounded model checking engine then constructs a formula which is satisfiable if and only if a trace with above properties exists.[2] In case of satisfiability, any

[2]It should be noted that this semantic property does not imply that the solver engine subsequently checking that formula for satisfiability can exactly determine its satisfiability. In the case of iSAT, a sound, yet incomplete unsatisfiability check is implemented, as necessitated by the undecidable fragment of arithmetic addressed.

satisfying valuation of this formula corresponds to such a trace. For encoding the discrete transition system on Taylor model in BMC mode, `iSAT3` has an input file format consisting of four sections:

- `DECL`: This part contains declaration of all variables (i.e., variables of the dynamic system, Taylor coefficients of the Taylor over-approximation solution, the duration of each segment $t \in [0, \tau]$, the uncertain time-varying parameter $\xi \in [0, \tau]$).
- `INIT`: This part is a formula describing the initial state(s) of the system to be investigated.
- `TRANS`: This formula describes the transition relation in symbolic form; in our case the evolution of the time-discrete Taylor model. We encode a template interval Taylor form of fixed degree k, i.e., $f_n(t)$, and the relation between interval Taylor coefficients in the current and the next step. Variables may occur in primed (e.g., a') or unprimed (e.g., a) form. A primed variable represents the value of that variable in the successor step, i.e., after the transition has taken place.
- `TARGET`: This formula characterises the state(s) whose reachability is to be checked; in our case it represents satisfaction of the given MITL formula.

The solver unwinds the transition relation k_{depth} times, conjoins the resulting formula with the formulae describing the initial state(s) and the target state(s), and then solves the obtained formula. For our transition relation in terms of Taylor coefficients, the solver recursively for each time frame $[0, k_{depth}\tau]$ constructs the following formula:

$$init\ (\vec{\mathbf{a}}^0) \wedge \bigwedge_{i=0}^{k_{depth}-1} trans\ (\vec{\mathbf{a}}^i, \vec{\mathbf{a}}^{i+1}) \wedge target\ (\vec{\mathbf{a}}^{k_{depth}}),$$

where $\vec{\mathbf{a}}$ is the interval-vector of the Taylor coefficients of the fixed-degree Taylor polynomial $f_n(t)$.

Returning to Chapter 3 in order to recall our running example in Sect. 3.2.1, we explain the `iSAT3` encoding through the constructed operator (3.8)

$$\begin{bmatrix} a_{n+1_0} \\ a_{n+1_1} \\ a_{n+1_2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -\frac{1}{2} & -\xi_n \end{bmatrix} \begin{bmatrix} a_{n_0} \\ a_{n_1} \\ a_{n_2} \end{bmatrix}$$

that defines a safe enclosure to the solution of DDE (3.7) by a sequence of parametric Taylor series with parameters in interval form using iSAT3. The iSAT3 encoding for this example is as shown in Listing (5.1).

In the *DECL* part, we declare all variables; the variables of the dynamic system, i.e., x, the Taylor coefficients of degree 2, i.e., a_0, a_1, and a_2, the duration of each segment $t \in [0,1]$, and the uncertain time varying parameter $\xi \in [0,1]$. Notice that the range of each variable has to be bounded in iSAT3. We initialize the system variable x and the Taylor coefficients in *INIT* part according to the given initial condition(s) in our example. Then, in the *TRANS* part, we state the interval Taylor form of degree 2, i.e., $f_n(t)$ corresponds to the solution x of DDE (3.7), as shown in line 22, and the relation between Taylor coefficients in the current (unprimed variables) and the next segment (primed variables) according to the generated operator (3.8), where the segments are of duration 1 each.

```
1  DECL
2  -- the range of each variable has to be bounded
3    float [-1000, 1000] a0, a1, a2, x;
4    float [0,1] t, xi;
5
6  INIT
7  -- initial value of solution
8    x = 1;
9
10 -- initialize Taylor coefficients
11   a0 = x;
12   a1 = 0;
13   a2 = 0;
14
15 TRANS
16 -- relation betw. Taylor coefficients in current and next step
```

```
17    a0' = a0 + a1 + a2;
18    a1' = -a0;
19    a2' = -0.5*a1 - xi*a2;
20
21  -- x(t) is given by a Taylor form of degree 2
22    x' = a0' + a1'*t + a2'*(t^2);
23
24  TARGET
25  -- state to be reached, e.g.
26    x = -0.25;
```

Listing 5.1 The encoding in iSAT3 of the running example in Sect. 3.2.1.

5.1.3 Proving Continuous-Time Properties on the Time Discretization

Operator (3.8) straightaway defines a safe temporal discretization of the DDE system in Eq. (3.3), i.e., an operator generating a classical timed state sequence in the sense of [4, 46]. We can, however, not simply apply the discrete-time interpretation of MITL to this timed state sequence, as it ranges over a different state space —namely the Taylor coefficients— than the requirements specification in terms of the original state variables. Therefore, we have to translate forth and back between the different state spaces and time models. In detail, the iterated execution of operator (3.8), starting from an initial vector $a_{0_0},\ldots,a_{0_k}$ of Taylor coefficients encoding the initial solution segment $x([0,\tau])$, generates a timed state sequence over (interval) Taylor coefficients, with time stamps $t_i = i\tau$, rather than a continuous signal over the state variables $x_1,\ldots,x_N$. Reflecting this encoding, we need a translation step generating conditions over the timed sequence of Taylor coefficients from which we are able to recover the original continuous-time, continuous-state signal-based semantics on the actual solution x of the DDE, as defined in Sect. 5.1.1.

As has already been observed in [162], such a mapping is straightforward when invariance properties are to be dealt with, for which a sufficient —yet, in the light of over-approximation of the solution, obviously not necessary— condition can be obtained as follows. For an invariance requirement $\Box x \in \mathit{Safe}$, where *Safe* is a set of safe states,

the requirement in the n-th segment is translated to the stronger condition

$$\forall t \in [0,\tau]\,\forall \xi \in [0,\tau]\,\forall a_0 \in A_{n,0},\ldots,a_k \in A_{n,k} : f_n(t) \in \mathit{Safe}, \tag{5.1}$$

where f_n is the underlying Taylor form and $A_{n,0}$ to $A_{n,k}$ are the intervals Taylor coefficients stemming from the n-th iteration of the operator (3.8). As this Taylor form provides an over-approximation of the solution x over time frame $[n\tau,(n+1)\tau]$, the condition (5.1) implies $\forall t \in [n\tau,(n+1)\tau] : x(t) \in \mathit{Safe}$. Consequently, the continuous-time safety property $\Box x \in \mathit{Safe}$ for system (3.7) is translated into a sufficient condition according to Eq. (5.1) for $t,\xi \in [0,1]$ over the sequence of Taylor coefficients of Taylor polynomial of degree 2. As its violation is an existential statement both instantiations of Taylor coefficients within given intervals and existentially quantified time points t and ξ, a solver for satisfiability modulo theory over the existential theory of polynomial arithmetic can be used to solve the safety verification problem. It requires polynomial constraint solving due to the Taylor forms, i.e., polynomial expressions involved in the statement $f_n(t) \in \mathit{Safe}$.

Different proof schemes can be implemented using such a solver: using k-induction [140] or interpolation-based unbounded proof schemes [99], absence of any time point in the sequence of valuations generated by operator (3.8) satisfying $\exists n \in \mathbb{N}, \exists t \in [0,1], \exists \xi \in [0,1], \exists a_0 \in A_{n,0},\ldots,a_k \in A_{n,k} : f_n(t) \notin \mathit{Safe}$ can be shown, thereby rigorously showing safety of the DDE system under investigation. Bounded model checking of the same system could, on the other hand, generate counterexamples to safety, which may however be spurious due to the over-approximation involved in the Taylor enclosure.

5.2 Solving Continuous-Time MITL Formulae by Reduction to Time-Discrete Taylor Approximations

We extend the above idea of generating sufficient conditions for MITL specifications on DDEs in terms of the sequences of enclosing (interval) Taylor coefficients. The aim is to cover a large fragment of MITL, expanding well beyond the invariance properties addressed in [162]. As explained in the previous section, we have obtained a generator for a timed state sequence —the operator (3.8)— representing the solution of the DDE,

yet ranging over a different state space, namely the Taylor coefficients. Hence, the continuous interpretation of the MITL formulae over DDE solutions has to be translated into a semantically appropriate discrete interpretation on a timed state sequence with time stamps $t_i = i\tau$. This translation needs to restore, in the sense of providing sufficient conditions for the solution being a counterexample (i.e., a witness of violation of the property), the continuous semantics of the MITL formulae over the discrete model of the timed state sequence. We do so by first transforming the MITL formula into negation normal form, then generating a sufficient condition by adding the appropriate conditions to the Taylor model that meet the semantics of the property for searching for (possibly spurious) counterexamples with the help of an efficient SMT solver.

5.2.1 Atomic Proposition

According to the MITL syntax of Sect. 5.1, atomic propositions are of the form $e \sim c$, where e is an expression over the state variables, c is a constant and $\sim$ an inequational relational operator, i.e., one of $<, \leq, >, \geq$. Using bounded model-checking based on SMT solving, we attempt to find a counterexample of the MITL formula, or, in other words, look for a witness for the negation of the MITL formula. As we transform that negated formula into NNF, atomic propositions occur in positive context only. Then, sufficient conditions for truth of such propositions throughout a time frame $[i\tau, (i+1)\tau]$ can, as already observed in Eq. (5.1)), obviously be expressed as follows:

$$\forall t \in [0,\tau]\, \forall \xi \in [0,\tau]\, \forall a_0 \in A_{i,0}, \ldots, a_k \in A_{i,k} : \bigwedge_{i=1}^{n} x_i = f_i(t) \wedge e \sim c. \tag{5.2}$$

As mentioned in Sect. 5.1.3, when using SMT solving for finding violations of condition (5.2), we use the negation of the universally quantified condition Eq. (5.2). As this is an existential formula, it is amenable to standard SMT solving.

5.2.2 Boolean Connectives

For solving complex-structured formulae, we use a Tseitin-like definitional translation [150], where we introduce a fresh Boolean helper variable $\langle \psi \rangle_i$ for each subformula ψ

and each index i of a time frame $[i\tau,(i+1)\tau]$. The intuition is that $\langle\psi\rangle_i$ being true implies that ψ holds for each time point $t \in [i\tau,(i+1)\tau]$. Note that this is a one-sided implication, as we cannot decide properties exactly.

Note that we have in the previous section already obtained appropriate definitions for the case that ψ is an atomic formula, such that we can define

$$\neg\langle e \sim c\rangle_i \Rightarrow \left(\begin{array}{c} \exists t \in [0,\tau]\, \exists \xi \in [0,\tau]\, \exists a_0 \in A_{i,0}, \ldots, a_k \in A_{i,k} : \\ \bigwedge_{i=1}^{n} x_i = f_i(t) \wedge e \not\sim c \end{array} \right) \tag{5.3}$$

as sufficient condition for validity of an atomic formula $e \sim c$, where e is an expression over the state variables and $\not\sim$ is the converse of the relation $\sim$.

Given a compound formula of the form $\psi_1 = \varphi_1 \wedge \varphi_2$ or $\psi_2 = \varphi_1 \vee \varphi_2$, the encoding for the compound formula is obtained by conjoining to the "axiomatisations" of φ_1 and φ_2 the following definitional translations:

$$\langle \varphi_1 \wedge \varphi_2\rangle_i \Leftrightarrow \langle \varphi_1\rangle_i \wedge \langle \varphi_2\rangle_i$$
$$\langle \varphi_1 \vee \varphi_2\rangle_i \Leftrightarrow \langle \varphi_1\rangle_i \vee \langle \varphi_2\rangle_i$$

Note that a single-sided implication "$\Leftarrow$" from right to left would actually suffice, as we target sufficient conditions only.

5.2.3 Unary Temporal Operators

Assume we have an MITL formula $\psi_1 = \Diamond_{\mathcal{I}}\ \varphi$ or $\psi_2 = \Box_{\mathcal{I}}\ \varphi$ featuring a time-constrained *eventually* or *always* temporal operator as its outermost operator. Let the lower and upper bound of $\mathcal{I}$ for simplicity be integer multiples $l\tau$ and $u\tau$ of τ. For each time frame, the value of a given MITL formula is encoded with the help of new Boolean variables for the truth values of its subformulae in particular time instants. The

encoding of ψ_1 and ψ_2 can be recursively understood as follows:

$$\begin{aligned}
&\langle \Diamond_{[l\tau,u\tau]} \rangle_i \Leftrightarrow \langle \Diamond_{[0,(u-l)\tau]} \varphi \rangle_{i+l}, && \text{if } 1 \leq l < u \\
&\langle \Diamond_{[0,u\tau]} \varphi \rangle_i \Leftrightarrow \langle \varphi \rangle_i \vee \langle \Diamond_{[0,(u-1)\tau]} \varphi \rangle_{i+1}, && \text{if } 1 < u \\
&\langle \Diamond_{[0,\tau]} \varphi \rangle_i \Leftrightarrow \langle \varphi \rangle_i \\
&\langle \Box_{[l\tau,u\tau]} \rangle_i \Leftrightarrow \langle \Box_{[0,(u-l)\tau]} \varphi \rangle_{i+l}, && \text{if } 1 \leq l < u \\
&\langle \Box_{[0,u\tau]} \varphi \rangle_i \Leftrightarrow \langle \varphi \rangle_i \wedge \langle \Box_{[0,(u-1)\tau]} \varphi \rangle_{i+1}, && \text{if } 1 < u \\
&\langle \Box_{[0,\tau]} \varphi \rangle_i \Leftrightarrow \langle \varphi \rangle_i
\end{aligned}$$

Single-sided implications "$\Leftarrow$" from right to left would again suffice for a sound definitional translation.

In the case of the eventually modality, the condition for detecting satisfaction of φ is somewhat stronger than necessary, actually requiring it to hold throughout a full time frame rather than just once inside.

5.2.4 Binary Temporal Operators

Assume we have a subformula of shape $\psi_1 = \varphi_1 \ \mathcal{U}_{\mathcal{I}} \ \varphi_2$ or $\psi_2 = \varphi_1 \ \mathcal{R}_{\mathcal{I}} \ \varphi_2$ featuring a time-constrained *until* or *release* operator as its outermost connective. For simplicity, we assume that the lower bound of $\mathcal{I}$ is 0. Such a form can always be achieved by prepending the modality with a unary temporal operator. Then the encoding of a sufficient condition for validity of ψ_1 or ψ_2, resp., over time frame $[i\tau,(i+1)\tau]$ is as follows:

$$\begin{aligned}
&\langle \varphi_1 \ \mathcal{U}_{[0,u\tau]} \ \varphi_2 \rangle_i \Leftrightarrow \langle \varphi_2 \rangle_i \vee (\langle \varphi_1 \rangle_i \wedge \langle \varphi_1 \ \mathcal{U}_{[0,(u-1)\tau]} \ \varphi_2 \rangle_{i+1}), && \text{if } 1 < u \\
&\langle \varphi_1 \ \mathcal{U}_{[0,\tau]} \ \varphi_2 \rangle_i \Leftrightarrow \langle \varphi_2 \rangle_i \\
&\langle \varphi_1 \ \mathcal{R}_{[0,u\tau]} \ \varphi_2 \rangle_i \Leftrightarrow \langle \varphi_2 \rangle_i \wedge (\langle \varphi_1 \rangle_i \vee \langle \varphi_1 \ \mathcal{R}_{[0,(u-1)\tau]} \ \varphi_2 \rangle_{i+1}), && \text{if } 1 < u \\
&\langle \varphi_1 \ \mathcal{R}_{[0,\tau]} \ \varphi_2 \rangle_i \Leftrightarrow \langle \varphi_2 \rangle_i
\end{aligned}$$

As in the case of the eventually modality, the condition for detecting φ_2 in the case of until and of φ_1, resp., in the case of release again is somewhat stronger than necessary, requiring it to hold throughout the respective time frame instead of just once inside.

5.2.5 Correctness

Let ψ be an MITL formula and $[\psi]_0$ be the definitional translation of ψ obtained by recursively unfolding and conjoining the above definitions of $\langle\psi\rangle_0$ and all $\langle\varphi\rangle_j$ occurring therein. Let $\frac{d\vec{x}}{dt}(t+\tau) = f(\vec{x}(t))$ be a DDE with initial value $\vec{x}([0,\tau]) \equiv \vec{i}$, and let A be the interval matrix obtained from it due to Eq. (3.8). Let k be the highest index j of any Tseitin variable $\langle\varphi\rangle_j$ occurring in $[\psi]_0$.

Lemma 5.2.1. *If $\vec{a}_0 \hat{=} \vec{i} \wedge \bigwedge_{i=0}^{k} \vec{a}_{i+1} = A\vec{a}_i \wedge [\psi]_0 \wedge \neg\langle\psi\rangle_0$ is unsatisfiable then $\vec{x}$ satisfies ψ, where $\vec{a}_0 \hat{=} \vec{i}$ denotes the appropriate initialisation of the Taylor coefficients and $\vec{x}$ is the exact solution of the DDE.*

Proof. The sequence $\vec{a}_0 \hat{=} \vec{i} \wedge \bigwedge_{i=0}^{k} \vec{a}_{i+1} = A\vec{a}_i$ of interval Taylor forms generates an over-approximation of x. The construction of $[\psi]_0$ is such that $\vec{a}_0 \hat{=} \vec{i} \wedge \bigwedge_{i=0}^{k} \vec{a}_{i+1} = A\vec{a}_i \wedge [\psi]_0 \models \langle\psi\rangle_0$ if all trajectories y enclosed by the sequence of interval Taylor forms, and thus also x itself, satisfies ψ. Satisfiability of $\vec{a}_0 \hat{=} \vec{i} \wedge \bigwedge_{i=0}^{k} \vec{a}_{i+1} = A\vec{a}_i \wedge [\psi]_0 \wedge \neg\langle\psi\rangle_0$ consequently is a necessary condition for violation of ψ by x. □

Note that $\vec{a}_0 \hat{=} \vec{i} \wedge \bigwedge_{i=0}^{k} \vec{a}_{i+1} = A\vec{a}_i \wedge [\psi]_0 \wedge \neg\langle\psi\rangle_0$ is a purely existential statement and thus amenable to standard SAT-modulo-theory solving by removing the explicit existential quantifiers in each instant of Eq. (5.3) by introducing fresh variables.

5.2.6 Verification Examples

In this section, we use the `iSAT3` SMT solver to discharge the above proof obligations. In order to be able to present the encodings in a compact form suitable for manual inspection and for publication in print, we slightly deviate from a strict implementation

of the above scheme, and instead employ the bounded model checking (BMC) mode of iSAT and symbolic counter variables as abbreviation mechanisms whenever appropriate in the search for witnesses as counterexamples of the MITL formulae. The results are, however, the same and the logics behind the encodings is equivalent to the point it can be in a BMC encoding. In particular, the method for using existential arithmetic constraints as sufficient conditions to determine the truth values of propositional (sub)formulae based on the over-approximation model of the DDE and thus recover the continuous semantics of the MITL formula on the actual solution x of the DDE from the timed state sequence semantics is exactly as in Eq. (5.3).

We demonstrate the approach based on illustrative examples of DDEs in the form of Eq. (3.3). In our examples, we first consider the DDE (3.7) presented in Sect. 3.2.1 with different conjectured MITL formulae to be verified. Thereafter, we apply our method to an adaptation of Gustafson's model of nutrient flow in an aquarium (three dimensional example) [64, p. 589f].

Example 5.2.1. *We consider the linear DDE* $\dot{x}(t) = -x(t-1)$ *with initial condition* $x([0,1]) \equiv 1$ *and the conjectured safety property* $\Box_{[0,10]}\ (x \leq 1.2)$.

The bounded degree interval-based sequence of Taylor forms can be generated by the operator Eq. (3.8)). Adopting degree 2 Taylor forms, we can encode this generator in the `iSAT3` input language as shown in lines 24–26 of Listing 5.2. The encoding is a discrete-time dynamic system over the variables x representing (snapshots of) the DDE solution, Taylor coefficients of the Taylor over-approximation solution, i.e., $a_0, a_1,$, and a_2, a time point in each segment $t \in [0,1]$, and the uncertain time varying parameter $\xi \in [0,1]$. Also, we declare a *counter* to observe the timing bound on the temporal operator.

In order to solve the given MITL formula $\Box_{[0,10]}\ (x \leq 1.2)$ in `iSAT3` in the sense of trying to construct a counterexample, we

1. in accordance with Eq. (5.3) search for a time frame within which x, being defined as the image of the Taylor polynomial for some $t \in [0,1]$, $\xi \in [0,1]$ in line 29 of the listing, exceeds 1.2, as encoded by condition $x > 1.2$ in the target (line 37), and

2. enforce the count of the time frame to be at most 9 (target, line 38), as time frame n ranges from time n to $n+1$.

For verifying the property at hand, it obviously suffices to check this formula up to unwinding depth 9.[3] Such bounds on unwinding depths can in `iSAT3` be set with the `-start-depth` and `-max-depth` command line options.

In our example, the solver outputs that the system is *safe* for unwinding depth 10, i.e., no state satisfying the target property could be reached within the relevant depth. This constitutes a rigorous proof that the system actually satisfies the MITL formula.

```
1  DECL
2  -- the range of each variable has to be bounded
3    float [-1000, 1000] a0, a1, a2, x;
4    float [0,1] t, xi;
5
6  -- define counter for the bounded verification problem
7    int [0,9] counter;
8
9  INIT
10 -- initial value of x over [0,1]
11   x = 1;
12
13 -- initialize Taylor coefficients
14   a0 = 1;
15   a1 = 0;
16   a2 = 0;
17
18 -- initialize the counter observing the time interval
19 -- covered by the bounded always
20   counter = 0;
21
22 TRANS
23 -- relation between Taylor coefficients current and next step
24   a0' = a0 + a1 + a2;
```

[3] `iSAT` counts unwindings starting from 0 such that an unwinding of depth 9 yields a trace comprising 10 time instants.

```
25   a1' = -a0;
26   a2' = -0.5*a1 - xi*a2;
27
28 -- x(t) is given by a Taylor form of degree 2
29   x' = a0' + a1'*t + a2'*(t^2);
30 -- note the implicit existential quantification of t
31
32 -- increment the counter by 1 after each time frame
33   counter' = counter + 1;
34
35 TARGET
36 -- state to be reached in bounded time
37   x > 1.2 and
38   counter <= 9;
```

Listing 5.2 The encoding of Example 5.2.1 in iSAT3.

Example 5.2.2. *Consider the same DDE equation as Example* (5.2.1) *with the same initial condition, but for solving the conjectured safety property of (bounded) until operator* $(x \leq 1.2)\ \mathcal{U}_{[0,10]}\ (x \leq 1.0)$.

This time, we employ four Boolean helper variables, of which iSAT's BMC mode will instantiate a fresh copy in each step:

1. Boolean state variable b records a sufficient condition for $x \leq 1.2$ being true throughout the current time frame in the sense that b is true only if $x(t) \leq 1.2$ holds for each time instant t in the current time frame (cf. lines 28 and 50 in Listing 5.3);

2. Boolean state variable c records a sufficient condition for $x \leq 1.0$ being true throughout the current time frame (cf. lines 31 and 52);

3. the Boolean state variable u records a sufficient condition for the temporal porperty $(x \leq 1.2)\ \mathcal{U}_{[0,10-n]}\ (x \leq 1.0)$ being true in the current step, with n being the number of the current step (cf. line 54);

4. Boolean state variable done is a helper variable necessitated by the confined expressiveness of the BMC mode, which permits reference to current and next

states only. It records whether the termination condition $x \leq 1.0$ has already been true in the past (lines 34 and 55).

```
1  DECL
2  -- the range of each variable has to be bounded
3    float [-1000, 1000] a0, a1, a2, x1, x2;
4    float [0,1] t1, t2, xi;
5  -- each of the atomic subformulae needs its own
6  -- fresh copy of the state variable x and the time
7  -- instant t due to the quantifier elimination
8
9  -- define counter for bounded verification problem
10   int [0,9] counter;
11
12 -- define Boolean helper variables
13   boole b, c, u, done;
14 -- b records sufficient condition for x <= 1.2
15 -- c records sufficient condition for x <= 1.0
16 -- u records sufficient condition for until
17 -- done records whether c has been true in the past
18
19 INIT
20   x1  = 1;
21   x2  = 1;
22 -- initialize Taylor coefficients
23   a0  = 1;
24   a1  = 0;
25   a2  = 0;
26
27 --initialize b, the sufficient condition of everywhere x <= 1.2
28   (not b)  -> (x1 > 1.2);
29
30 --initialize c, the sufficient condition of everywhere x <= 1.0
31   (not c)  -> (x2 > 1.0);
32
33 -- initialize done, the variable memoizing x <= 1.0
34   done  <-> c;
35
```

```
36 -- counter observes the time interval
37   counter = 9;
38
39 TRANS
40 -- description of the transition system of DDE model
41   a0' = a0 + a1 + a2;
42   a1' = -a0;
43   a2' = -0.5*a1 - xi *a2;
44
45 -- tracing the bounded until
46 -- find witness points
47   x1' = a0' + a1'*t1 + a2'*(t1^2);
48   x2' = a0' + a1'*t2 + a2'*(t2^2);
49 -- b is sufficient condition for x <= 1.2 throughout time frame
50   (not b') -> (x1' > 1.2);
51 -- c is sufficient condition for x <= 1.0 throughout time frame
52   (not c') -> (x2' > 1.0);
53 -- recurrence rules for until
54   u  <-> done or (b and u');
55   done' <-> (c' and counter > 0) or done;  -- remembers c
56   (counter > 0) -> (counter' = counter-1);
57   (counter = 0) -> (counter' = 0);
58
59 TARGET
60 -- for constructing a counterexample, the until formula ought
61 -- to be violated in the initial time instant
62   (not u) and (counter = 9);
```

Listing 5.3 The encoding of Example 5.2.2 in iSAT3.

Checking above example for an appropriate unwinding depth of at least 9, iSAT will report unsatisfiable, which approves absence of a counterexample and thus proves the property to be satisfied.

Example 5.2.3. *This example (taken from [162]) is an adaptation of Gustafson's model of nutrient flow in an aquarium [64, 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

$$\dot{\vec{x}}(t) = \begin{bmatrix} -3 & 6 & 5 \\ 2 & -12 & 0 \\ 1 & 6 & -5 \end{bmatrix} \vec{x}(t - \frac{1}{100}) \tag{5.4}$$

with initial condition $\vec{x}([0,1]) \equiv [10,0,0]$ *and a conjectured MITL formula specifying the distance between the isotope concentrations of two aquatic plankton varieties always stays below* 10 *in a bounded time* $[0,50]$*, i.e.,* $\Box_{[0,50]} \mid x_2 - x_3 \mid \leq 10$.

Using Taylor models of degree 1, we calculate the operator relating successive parameter vectors to be

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

In this example, the solver outputs that the system is *safe*, which means that any state satisfying the target property is *unreachable* within depth 50 w.r.t. the over-approximation model of the DDE. This constitutes a rigorous proof that the system actually satisfies the given property.

```
1  DECL
2  -- the range of each variable has to be bounded
3    float [-1000, 1000] a01, a11, a02, a12, a03, a13, x1, x2, x3;
4    float [0,1/100] t, xi1, xi2, xi3;
5
6  -- define counter for the bounded verification problem
7    int [0,49] counter;
8
9  INIT
10 -- initial values for the three components of the state
```

```
11   x1  = 10;
12   x2  = 0;
13   x3  = 0;
14 -- initialize Taylor coefficients
15   a01 = 10;
16   a11 = 0;
17   a02 = 0;
18   a12 = 0;
19   a03 = 0;
20   a13 = 0;
21 -- initialize the counter
22   counter = 0;
23
24 TRANS
25 --description of the transition system of DDE model
26   x1' = a01' + a11'*t;
27   x2' = a02' + a12'*t;
28   x3' = a03' + a13'*t;
29   a01' = a01 + (1/100)*a11;
30   a11' = ((-3)*a01) - ((3*xi1)*a11) + (6*a02)
31                     + ((6*xi1)*a12) + (5*a03) + ((5*xi1)*a13);
32   a02' = a02 + (1/100)*a12;
33   a12' = (2*a01) + ((2*xi2)*a11) - (12*a02) - ((12*xi2)*a12);
34   a03' = a03 + (1/100)*a13;
35   a13' = a01 + (xi3*a11) + (6*a02) + ((6*xi3)*a12)
36                     - (5*a03) - ((5*xi3)*a13);
37
38 -- increment the counter by 1 for each time frame
39   counter' = counter + 1;
40
41 TARGET
42 -- state to be reached in bounded time
43   abs(x2-x3) > 10 and counter <= 49;
```

Listing 5.4 The encoding of Example 5.2.3 in iSAT3.

5.3 Discussion

Our research has elaborated a necessarily incomplete method to verify/falsify temporal specifications of time-delay systems modeled by a simple class of delay differential equations (DDEs) with a single constant delay. Although the above verification procedure for temporal specifications is interesting, it could suffer from providing false negatives and corresponding counter-examples, due to excessive over-approximation of the DDE's solution. As this problem would be induced by selecting an insufficient bound on the degree of the Taylor forms, one could simply select a higher degree. Therefore it should, however, be clear that the negative verdict actually is spurious due to excessive over-approximation. Automatic methods to check whether the reported counterexample is spurious or not remain to be developed. The situation is thus still semi-automatic due to adjusting manually the fixed degree of the Taylor forms. We hypothesize that one possible solution to make it fully automatic could be by using *counter-example guided abstraction refinement* (CEGAR) [30] for enhancing the over-approximation model. Another possible promising solution would be by conducting sensitivity analysis, and hence refining the over-approximating Taylor model. That latter solution aims at eliminating the wrapping effect due to the dependency issues in interval arithmetic. A preliminary study of that latter solution will be manifested as a future research direction.

Chapter 6

Conclusion, Limitations, and Future Research

"Problems worthy of attack prove their worth by hitting back[1]*."*

[Piet Hein]

This chapter attempts to put the dissertation into broader context and to subject it to criticism. It is more than just a summary of the chapters or data the author has presented in the main chapters of the dissertation. Along with providing a synthesis of the key findings and arguments projected by our research to answer the research questions of this dissertation, this chapter makes a stand regarding the dissertation statement. In other words, this chapter is able to stand on its own and provide a justification and defense of the dissertation. Hence, in an attempt to have a clear structure of this chapter, we first in Section 6.1 conclude our research, where we draw the attention of the reader to the dissertation statement upon which our research was conducted. And we provide evidence and synthesis of arguments, presented in the body of our dissertation, to show how these converge to answer our research questions. Then, in Section 6.2, we identify the various limitations which were encountered during our research. This could also be combined with Section 6.3, future research, demonstrating how future research could build on from this research, recognizing and responding to the limitations.

[1] 'Problems', p. 2 in GROOKS, by Piet Hein.

6.1 Conclusion

Delay differential equations (DDEs) are yet among the most commonly investigated types of differential equations for the reason that they play an important role in the modeling of natural or artificial processes with time delays in biology, physics, economics, engineering, etc. In engineering, our area of focus, the community has raised some concerns about increasing expectations of the accuracy in describing dynamic performances, i.e., the engineers expect their models to behave like the real processes by considering, e.g., the delay, in the framework of differential equations, consequently raising concerns about DDEs especially for safety-critical control systems driven by the demand for safety cases in a broad sense. DDEs are reasonably used towards enhancing the understanding of the situations encountered in many modern control applications, where the feedback dynamics entails delays due to communication networks etc. Relaxing these DDEs to ordinary differential equations (ODEs) in automatic verification, in many cases, may yield misleading results owing to the impact of delays on system dynamics. Thus, handling DDEs in automatic verification is crucial.

Unlike the plethora of automatic verification techniques developed for ODEs, automatic verification techniques available for DDEs appear to be not well supported, where their tool support still seems to mostly be confined to numerical simulation based on integration from discontinuity to discontinuity, e.g. by Matlab's `dde23` algorithm. Although such numerical simulation is extremely useful in system analysis, it fails to provide reliable certificates of system properties, as it is numerically approximate only. For this reason, lifting the power of established verification methods for ODEs to much more complex objects than ODEs, i.e., DDEs[2], is necessary to provide reliable certificates for DDE-system properties, e.g., safety properties. This dissertation has given an account of the progress in this direction by answering many research questions raised during the author's research journey.

In Chapter 3 which is based on [162], we have succeeded in using interval-based Taylor forms as a reasonable data structure to enclose the solution of a DDE, where the delay introduced in the framework of the differential equation is a single constant delay τ. To sum up, this study has gone some way towards providing a safe enclosure

[2]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.

method for a simple class of DDEs and obtaining a procedure able to provide stability and safety certificates. The findings of this study suggest that 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 over-approximations 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 over-approximation for the next temporal segment from the current one, and we employ constraint solving for automatically analyzing its properties. Our findings suggest that iSAT tool [52] could be taken advantage of in order to obtain an automatic procedure able to provide stability and safety certificates for a simple class of DDEs.

These early successes, in our view, represent an excellent initial step towards the main aim of this dissertation: handling DDEs in automatic verification, concerning time-bounded safety verification. We believe that our results in [162] have important implications for providing reliable certificates of the system dynamics represented as a DDE with a single constant delay τ, i.e., of the restricted form given by DDE (3.3), against time-(un)bounded invariance (stability and) safety properties. For pursuing our research goal on a wider scale, we have planned further research in two directions. The first direction is to consider more complex classes of DDE than DDE with a single constant delay. In this direction, Chen *et al.* collaborated in [24] to investigate the class of systems that involves a combination of ODE and DDE with multiple constant delays by using validated simulation-based verification techniques, however, this work is not considered in this dissertation. More details on this study can be found in [24]. Interestingly, we have found a promising method to consider more complex classes of DDE and to provide over- and under-approximations of the reachable set for a DDE. It is reachability analysis method based on set-boundary of ODEs that is introduced by Xue *et al.* in [155, 157]. In Chapter 4 which is based on [156], we have adapted this method to handle a more complex class of DDE than the class discussed in [162]. The second direction is to verify the system dynamics modeled by DDE against wider ranges of temporal properties rather than just invariance properties. We have carried out this work in [103] and its extended revised version [104], as presented in Chapter 5.

For our study in the first direction, in order to lift the power of the reachability analysis method based on set-boundary of ODEs to DDEs, many concerns have

been raised. In summary, the reachability analysis method based on set-boundary of ODEs that is in [155, 157] relies on the fact that the solution mapping of ODE is a homeomorphism and thus preserves set boundaries. Based on this, one can retrieve safe over- and under-approximations for ODEs from enclosures of the dynamic images of the boundaries of the initial set. Contrary to Xue's reasoning in [155, 157], however, the solution mappings for DDEs need not be homeomorphisms[3]. In [156], and presented in Chapter 4, we have managed to expose a class of DDEs exhibiting homeomorphic dependency on initial conditions. This class is a higher class in complexity compared to the class of DDE discussed in our very initial work in this field, where the right-hand side of the differential equation in this class is a combination of ODE and DDE with single constant delay τ, facilitating the coverage of many situations encountered in modern control applications. Membership in this class is determined by conducting sensitivity analysis of the solution mapping with respect to the initial states, therefrom deriving an upper bound on the time-lag term τ of the DDE that ensures homeomorphic dependency. One of the primary benefits of the existence of a corresponding homeomorphism is that state extrapolation can be pursued from the boundaries of the initial set only, rather than the full initial set, as the homeomorphism preserves boundaries and interiors of sets. As (appropriate enclosures of) the boundaries of the initial set have much smaller volume, such an approach tremendously reduces the wrapping effect incurred when using set-based state extrapolation on ODE with inputs as a means for enclosing solutions to the DDE. Furthermore, it allows us to construct an over- and under-approximations of the full reachable set by including (excluding, resp.) the obtained boundary enclosure from certain convex combinations of points in that boundary enclosure. Taken together, our findings in this study highlight significant results on computing both under- and over-approximations of the reachable sets for DDEs that may have important implications for providing an indication of the precision of an estimate of the exact reachability region. We believe that these findings add a significant contribution to a growing body of literature on studying the automatic verification of system dynamics modeled by DDE.

For our study in the second direction, as stated in the motivation of our research in Chapter 1, confining safety properties to a set of unsafe states or invariance properties restricts the ability of designers to adequately express the desired safe behavior of the

[3]The inverse of the solution mapping of DDE may have numerous branches, not a unique inverse as for ODE.

system that may involve a number of critical properties such as timing requirements and bounded response. The aim of our research in [103] and its extended revised version [104], as presented in Chapter 5, was thus to cover the automatic formal verification of DDEs against arbitrary temporal properties, rather than just against invariance properties as considered in our initial research. In conclusion, we have elaborated in [103, 104] a method to verify/falsify temporal specifications of time-delay systems modeled by a simple class of DDEs with a single constant delay τ. Taking advantage of a fixed degree interval-based Taylor over-approximation method discussed in our very initial work, our research has suggested that metric interval temporal logic (MITL) [4] could be exploited as requirements specification language in order to express time-bounded properties with continuous-time semantics on the solutions of the DDEs.

In this study, we have built our method around a fixed degree interval-based Taylor over-approximation technique [162] in order to provide a safe enclosure method for DDEs, thereby obtaining timed state sequences spanned by the piecewise valid Taylor coefficients. In this way, the continuous semantics of the MITL formulae is reduced to a time-discrete problem on timed state sequences in terms of Taylor coefficients. Then, we have devised sufficient conditions on these timed state sequences recovering the continuous-time interpretation of MITL on the actual solutions of the DDEs. To achieve this, we have first built sufficient conditions for validation of the atomic predicates over time frames of the Taylor over-approximation model of the DDE. We have then extended this approach to arbitrary bounded MITL formulae in negation normal form. Exploiting this as a tableaux or using a related encoding as a bounded model checking (BMC) problem, we could employ an appropriate arithmetic SMT solver addressing (a.o.) polynomial arithmetic as a tool able to automatically provide certificates of temporal properties for DDEs. In our case, we have used the `iSAT3` solver[4], which is the third implementation of the iSAT algorithm [52]. In very first experiments on simple DDEs, the `iSAT3` solver proved able to solve the temporal properties expressed in MITL formulae, thereby safely determining satisfaction of the formulae in an over-approximation setting. We were able to verify formulae of temporal logic also involving Boolean connectives and temporal modalities, like the *(bounded) until* operator. The soundness of the method is guaranteed due to the over-approximation employed in

[4] `iSAT3` is a satisfiability checker for Boolean combinations of arithmetic constraints over real- and integer-valued variables. The iSAT3 implementation of the iSAT algorithm [52] is available at `http://projects.informatik.uni-freiburg.de/projects/isat3/`

the DDE enclosure by Taylor forms and the sufficient conditions of determining the truth values of the atomic propositions over the time frames.

Finally, as a summary of this conclusion, our research in this dissertation has suggested several courses of action in order to address the problem of handling a complex class of differential equations than ODEs, i.e., DDEs, in time-bounded automatic formal verification. We have lifted the power of the established methods for enclosing reachable state sets of ODEs to specific classes of DDEs that may cover many possible situations encountered in many applications. Furthermore we have expanded the expressiveness of the properties by verifying DDE-systems against arbitrary time-bounded MITL formulae, including nesting of modalities, rather than just invariance properties. On the other hand, as normally known about scientific research, especially in engineering field, our study has encountered a number of limitations, which need to be considered.

6.2 Limitations

"As we advance in life we learn the limits of our abilities."

[James Anthony Froude[5]]

That's right, Mr. Froude! Also it would seem right when it comes to scientific research in general. It is well known and generally accepted that all research work unavoidably faces some limitations, raising some questions that need further research to be answered. This way, this section could also be combined with the next section on recommendations for future research. In fact, knowledge and discussion of limitations are essential for genuine scientific progress: they are useful for understanding research findings, placing the current work in context, and ascribing a credibility level to it. In this section, we focus on the limitations of our research. In other words, we raise some open research questions.

Our work in [162], and presented in Chapter 3, was clearly limited to a simple class of DDE, since we have assumed that the system dynamics is represented as a

[5]Inaugural Address Delivered to the University of St. Andrews, March 19, 1869 (1869), 3.

DDE with a single constant delay τ. Despite this we believe our work in [162] could be a springboard for handling DDEs in automatic formal verification by providing reliable certificates for invariance properties, e.g., stability and safety properties. Also, the restricted form of DDE considered in [162] is able to cover the models of several dynamical systems as in biology [58, 93], optics [75], economics [144, 145], ecology [51], to name just a few. On the other hand, 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. The cutting-edge solution for such cases could be through a layered combination of Taylor-model computation for ODEs, e.g. [110], with the ideas exposed in [162]. We recommend that further research should be done in this area. On a wider level, research is also needed to address more general kinds of DDE, like DDE with multiple different discrete delays, DDE with randomly distributed delay, or DDE with time-dependent or more generally state-dependent delay [87].

In [156], as presented in Chapter 4, we have considered a more complex class of DDE, where the system dynamics is represented as a combination of ODE and DDE with single constant delay τ. The most important limitation in this work is due to the fact that we have imposed an upper bound on the time-lag term τ of the DDE. Although this could be interesting in engineering process, where this upper bound on the time-lag can be considered as an automatically derived design space constraint, asking the development engineers for selection of appropriate components (sensors, processors, actuators, communication networks) guaranteeing sufficiently low latency in the feedback loop, the bound on the time-lag might suffer from conservativeness especially for high dimensional systems. This is based on the fact that the derived upper bound on the time-lag term τ depends explicitly on the dimension of the system n, i.e., the derived upper bound on τ is decreased with increasing n and tends to zero as n tends to infinity. One possible solution for this problem is to derive a new bound on τ independent of the system dimension n as suggested in [158]. Although the derived bound on the time-lag term τ in [156] was not optimal, we nevertheless believe that our innovative work in [156] has important implications for research into DDEs as it represents an important reference in order to leverage techniques for ODE on DDE. We are also confident that our results in [156] have improved knowledge about DDEs and can lead to further research built on from our research, e.g., [158].

On the one hand, in [103] and its extended revised version [104] we have addressed the confinement of our method in [162] to considering only invariance properties by extending the verification of DDE-systems against arbitrary temporal properties expressed by MITL formulae; on the other hand, the limitation to a specific class of DDE, i.e., DDE with single constant delay τ, is inherited. Our work presented in Chapter 5, which is based on [103, 104], has also a limitation to solving time-bounded verification problems of temporal logic properties for a class of DDEs. For the unbounded verification problems that are aimed to be facilitated by interval-based Taylor over-approximation method, it is still an active area of research. Apart from that, it is not surprising that our work presented in Chapter 5 is interesting preliminary attempt to verify system dynamics modeled by complex class of differential equations, i.e., DDEs, against arbitrary temporal properties expressed by MITL formulae that make the designers to adequately express the desired safe behavior of the system. At the same time we believe that our research will serve as a basis for future studies on the verification of DDEs. In other words, this work has successfully raised some good research questions that are intended to be answered in future research.

6.3 Future Research

"In three words I can sum up everything I've learned about life: It goes on.[6]*"*

[Robert Frost, American poet (1874-1963)]

And the research shall go on as well. In this context, we suggest in this section some recommendations for future research. In fact, the limitations of our research discussed above in Section 6.2, point towards topics to be addressed in the future, recognizing and responding to the limitations. Thus, this section could also be integrated with the section above, the limitations, in order to demonstrate how future research could build on from our research. We also have mentioned some future research ideas throughout different sections of this dissertation. Besides, we propose in this section two main research directions to broaden and enrich handling DDEs in automatic formal

[6]The acclaimed American poet Robert Frost was asked as an octogenarian what he had learned about life, and he succinctly replied: It goes on. 1954 September 5, The Cincinnati Enquirer, Section: This Week Magazine, Robert Frost's Secret by Ray Josephs, Quote Page 2, Column 1, Cincinnati, Ohio.

verification, responding to the motivation of our research. We propose that further research should be undertaken in the following areas:

- **Extension to general types of DDEs.** Future studies should aim at extending the discussed methods in this dissertation to more general types of DDEs. In fact, on a wider level, lifting the power of established verification methods for ODEs to much more complex objects than ODEs, i.e., DDEs with their several types, will be a challenge for future research for years. Some examples on the types of DDEs that may be a vital issue for future research are the following [87]:

  – DDE with discrete delays: DDEs with multiple delays are represented as

  $$\dot{x}(t) = f(t, x(t), x(t-\tau_i)),$$

  where the quantities $\tau_i > 0, i = 1, 2, \ldots,$ are time-lags or discrete delays and f is a vector valued smooth continuous function. This type of multiple delays inevitably occurs, for example, in networks of coupled dynamical systems with different architectures in analogy with the consideration of weighted networks, where different weights are considered at the couplings to account for the different degree of interactions between the various dynamical units in the network [87].

  – DDE with distributed delay: DDEs with distributed or continuous delay can be represented in general as [87]

  $$\dot{x}(t) = f(t, x(t), \int_0^\infty \mu(\tau) x(t-\tau) d\tau).$$

  Such models that are based on distributed delays have important implications in many areas such as biology, ecology, neurology, viscoelasticity, and economics [87].

  – DDE with state-dependent delay: DDEs with state-dependent delay can be represented in general as [87]

  $$\dot{x}(t) = f(t, x(t), x(t-\tau(t, x(t)))).$$

  State-dependent delay appears in many processes, where the delay depends on the present state and also on a delayed one, e.g., the delay for turning

processes in the milling operations [87]. For instance, the introduced time delay for turning processes in the milling operations is not only determined by the rotation of the workpiece but is also affected by the current and the delayed position of the tool. Consequently, this results in a DDE with state-dependent delay.

We believe that our current research will serve as a basis for future studies on handling several types of DDEs in automatic verification by lifting the power of ODEs' verification methods to DDEs, e.g., as already done in [158] by extending our method in [156] to address perturbed DDEs, where the dynamics of the class of DDE considered in [156] is subject to perturbations.

- **Extension to hybrid systems.** Further work needs to be carried out to extend the discussed methods herein in order to facilitate the automatic verification and analysis for hybrid systems that feature delays. Hybrid systems are a class of dynamical systems which exhibit both continuous and discrete behaviors [25]. They are yet among the most commonly investigated systems that usually appear in safety-critical situations, thus it is significant to model the continuous behavior within hybrid-state systems featuring delays by DDEs, rather than just ODEs that ignore the presence of any delay. A good way to go is to extend Egger's method for integrating safe ODE enclosures into a SAT modulo theory (SMT) solver [43, 44] 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 to integrate that propagator into the iSAT SMT solver as in [53]. We believe that our findings in this dissertation might be transferable hybrid systems that feature DDE models for continuous behaviors.

"There is no real ending. It's just the place where you stop the story."

[Frank Herbert[7], 1920–1986]

[7] An American science fiction writer.

References

[1] Althoff, M. (2013). Reachability analysis of nonlinear systems using conservative polynomialization and non-convex sets. In Belta, C. and Ivancic, F., editors, *Proceedings of the 16th international conference on Hybrid systems: computation and control, HSCC 2013, April 8-11, 2013, Philadelphia, PA, USA*, pages 173–182. ACM.

[2] Althoff, M. (2016). *CORA 2016 Manual*. http://www6.in.tum.de/Main/SoftwareCORA.

[3] Althoff, M., Stursberg, O., and Buss, M. (2008). Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In *Proceedings of the 47th IEEE Conference on Decision and Control, CDC 2008, December 9-11, 2008, Cancún, México*, pages 4042–4048. IEEE.

[4] Alur, R., Feder, T., and Henzinger, T. A. (1996). The benefits of relaxing punctuality. *J. ACM*, 43(1):116–146.

[5] Asl, F. M. and Ulsoy, A. G. (2003). Analysis of a system of linear delay differential equations. *Transactions-American Society of Mechanical Engineers Journal of Dynamic Systems Measurement and Control*, 125(2):215–223.

[6] Baier, C. and Katoen, J. (2008). *Principles of model checking*. MIT Press.

[7] Baker, C. T. (2000). Retarded differential equations. *Journal of Computational and Applied Mathematics*, 125(1):309–335.

[8] Baker, C. T., Bocharov, G. A., and Rihan, F. A. (1999). *A report on the use of delay differential equations in numerical modelling in the biosciences*. Manchester Centre for Computational Mathematics Manchester, UK.

[9] Baker, C. T., Paul, C. A., and Willé, D. R. (1995). A bibliography on the numerical solution of delay differential equations. Technical report, Manchester Centre for Computational Mathematics Manchester, UK.

[10] Baker, C. T., Thomas, R., Tian, H., Willé, D., Bocharov, G., Filiz, A., Ford, N., Paul, C., Rihan, F., and Tang, A. (1998). Numerical modelling by retarded functional differential equations. Technical report, Manchester Centre for Computational Mathematics Manchester, UK.

[11] Beckert, B., Hähnle, R., and Schmitt, P. H., editors (2007). *Verification of Object-Oriented Software. The KeY Approach - Foreword by K. Rustan M. Leino*, volume 4334 of *Lecture Notes in Computer Science*. Springer Verlag.

[12] Bellen, A. and Zennaro, M. (2013). *Numerical methods for delay differential equations*. Oxford University Press.

[13] Bellman, R. and Cooke, K. L. (1963). Differential-difference equations. Technical Report R-374-PR, The RAND Corporation, Santa Monica, California.

[14] Bellman, R. et al. (1943). The stability of solutions of linear differential equations. *Duke math. J*, 10(4):643–647.

[15] Berezansky, L. and Braverman, E. (2006). On stability of some linear and nonlinear delay differential equations. *Journal of Mathematical Analysis and Applications*, 314(2):391–411.

[16] Bertot, Y. and Castéran, P. (2004). *Interactive Theorem Proving and Program Development - Coq'Art: The Calculus of Inductive Constructions*. Texts in Theoretical Computer Science. An EATCS Series. Springer Verlag.

[17] Berz, M. and Makino, K. (1998). Verified integration of ODEs and flows using differential algebraic methods on high-order Taylor models. *Reliable Computing*, 4(4):361–369.

[18] Bisztriczky, T., McMullen, P., Schneider, R., and Weiss, A. I. (2012). *Polytopes: abstract, convex and computational*, volume 440. Springer Science & Business Media.

[19] Bocharov, G. A. and Rihan, F. A. (2000). Numerical modelling in biosciences using delay differential equations. *Journal of Computational and Applied Mathematics*, 125(1):183–199.

[20] Bournez, O., Maler, O., and Pnueli, A. (1999). Orthogonal polyhedra: Representation and computation. In Vaandrager, F. W. and van Schuppen, J. H., editors, *Hybrid Systems: Computation and Control, Second International Workshop, HSCC'99, Berg en*

Dal, The Netherlands, March 29-31, 1999, Proceedings, volume 1569 of *Lecture Notes in Computer Science*, pages 46–60. Springer Verlag.

[21] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, volume 15 of *Studies in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM).

[22] Breda, D., Maset, S., and Vermiglio, R. (2014). *Stability of Linear Delay Differential Equations: A Numerical Approach with MATLAB*. Springer Verlag.

[23] Chaochen, Z., Ravn, A. P., and Hansen, M. R. (1992). An extended duration calculus for hybrid real-time systems. In Grossman, R. L., Nerode, A., Ravn, A. P., and Rischel, H., editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 36–59. Springer Verlag.

[24] Chen, M., Fränzle, M., Li, Y., Mosaad, P. N., and Zhan, N. (2016). Validated simulation-based verification of delayed differential dynamics. In Fitzgerald, J. S., Heitmeyer, C. L., Gnesi, S., and Philippou, A., editors, *FM 2016: Formal Methods - 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings*, volume 9995 of *Lecture Notes in Computer Science*, pages 137–154. Springer Verlag.

[25] Chen, X. (2015). *Reachability Analysis of Non-Linear Hybrid Systems Using Taylor Models*. PhD thesis, RWTH Aachen University.

[26] Chen, X. and Ábrahám, E. (2011). Choice of directions for the approximation of reachable sets for hybrid systems. In Moreno-Díaz, R., Pichler, F., and Quesada-Arencibia, A., editors, *Computer Aided Systems Theory - EUROCAST 2011 - 13th International Conference, Las Palmas de Gran Canaria, Spain, February 6-11, 2011, Revised Selected Papers, Part I*, volume 6927 of *Lecture Notes in Computer Science*, pages 535–542. Springer Verlag.

[27] Chen, X., Ábrahám, E., and Sankaranarayanan, S. (2013). Flow*: An analyzer for non-linear hybrid systems. In Sharygina, N. and Veith, H., editors, *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, volume 8044 of *Lecture Notes in Computer Science*, pages 258–263. Springer Verlag.

[28] Chen, X., Sankaranarayanan, S., and Ábrahám, E. (2014). Under-approximate flowpipes for non-linear continuous systems. In *Formal Methods in Computer-Aided*

Design, FMCAD 2014, Lausanne, Switzerland, October 21-24, 2014, pages 59–66. IEEE.

[29] Chutinan, A. and Krogh, B. H. (1998). Computing polyhedral approximations to flow pipes for dynamic systems. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 2, pages 2089–2094. IEEE.

[30] Clarke, E. M., Grumberg, O., Jha, S., Lu, Y., and Veith, H. (2000). Counterexample-guided abstraction refinement. In Emerson, E. A. and Sistla, A. P., editors, *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, volume 1855 of *Lecture Notes in Computer Science*, pages 154–169. Springer Verlag.

[31] Clarke, E. M., Grumberg, O., and Peled, D. A. (2001). *Model checking*. MIT Press.

[32] Clarke, E. M., McMillan, K. L., Campos, S. V. A., and Hartonas-Garmhausen, V. (1996). Symbolic model checking. In Alur, R. and Henzinger, T. A., editors, *Computer Aided Verification, 8th International Conference, CAV '96, New Brunswick, NJ, USA, July 31 - August 3, 1996, Proceedings*, volume 1102 of *Lecture Notes in Computer Science*, pages 419–427. Springer Verlag.

[33] Daafouz, J. and Bernussou, J. (2001). Parameter dependent Lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & control letters*, 43(5):355–359.

[34] Davoren, J. M. (1997). On hybrid systems and the modal μ-calculus. In Antsaklis, P. J., Kohn, W., Lemmon, M. D., Nerode, A., and Sastry, S., editors, *Hybrid Systems V*, volume 1567 of *Lecture Notes in Computer Science*, pages 38–69. Springer Verlag.

[35] de Moura, L. M. and Bjørner, N. (2008). Z3: an efficient SMT solver. In Ramakrishnan, C. R. and Rehof, J., editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer Verlag.

[36] Delzanno, G. and Potapov, I., editors (2011). *Reachability Problems - 5th International Workshop, RP 2011, Genoa, Italy, September 28-30, 2011. Proceedings*, volume 6945 of *Lecture Notes in Computer Science*. Springer Verlag.

[37] Denman, W. (2017). Automated verification of continuous and hybrid dynamical systems. Technical report, University of Cambridge Computer Laboratory.

[38] Donzé, A. and Maler, O. (2007). Systematic simulation using sensitivity analysis. In Bemporad, A., Bicchi, A., and Buttazzo, G. C., editors, *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, volume 4416 of *Lecture Notes in Computer Science*, pages 174–189. Springer Verlag.

[39] Driver, R. D. (2012). *Ordinary and delay differential equations*, volume 20. Springer Science & Business Media.

[40] Duggirala, P. S., Mitra, S., and Viswanathan, M. (2013). Verification of annotated models from executions. In *Proceedings of the International Conference on Embedded Software, EMSOFT 2013, Montreal, QC, Canada, September 29 - Oct. 4, 2013*, pages 26:1–26:10. IEEE.

[41] Dutertre, B. and De Moura, L. (2006). The YICES SMT solver. *Tool paper at http://yices. csl. sri. com/tool-paper. pdf*, 2(2):1–2.

[42] Eggers, A. (2014). Direct handling of ordinary differential equations in constraint-solving-based analysis of hybrid systems. Doctoral Dissertation, Universität Oldenburg.

[43] Eggers, A., Fränzle, M., and Herde, C. (2008). SAT modulo ODE: A direct SAT approach to hybrid systems. In Cha, S. D., Choi, J., Kim, M., Lee, I., and Viswanathan, M., editors, *Automated Technology for Verification and Analysis, 6th International Symposium, ATVA 2008, Seoul, Korea, October 20-23, 2008. Proceedings*, volume 5311 of *Lecture Notes in Computer Science*, pages 171–185. Springer Verlag.

[44] Eggers, A., Ramdani, N., Nedialkov, N. S., and Fränzle, M. (2015). Improving the SAT modulo ODE approach to hybrid systems analysis by combining different enclosure methods. *Software and System Modeling*, 14(1):121–148.

[45] Einstein, A. and Infeld, L. (1961). *The Evolution of Physics, Etc.[Edited by Leopold Infeld.]*. Simon & Schuster.

[46] Fainekos, G. E., Girard, A., and Pappas, G. J. (2006). Temporal logic verification using simulation. In Asarin, E. and Bouyer, P., editors, *Formal Modeling and Analysis of Timed Systems, 4th International Conference, FORMATS 2006, Paris, France,*

September 25-27, 2006, Proceedings, volume 4202 of *Lecture Notes in Computer Science*, pages 171–186. Springer Verlag.

[47] Fainekos, G. E. and Pappas, G. J. (2006). Robustness of temporal logic specifications for finite state sequences in metric spaces. Technical report, Technical Report MS-CIS-06-05, Dept. of CIS, Univ. of Pennsylvania.

[48] Fan, C. and Mitra, S. (2015). Bounded verification with on-the-fly discrepancy computation. In Finkbeiner, B., Pu, G., and Zhang, L., editors, *Automated Technology for Verification and Analysis - 13th International Symposium, ATVA 2015, Shanghai, China, October 12-15, 2015, Proceedings*, volume 9364 of *Lecture Notes in Computer Science*, pages 446–463. Springer Verlag.

[49] Fisher, M., Dennis, L. A., and Webster, M. P. (2013). Verifying autonomous systems. *Commun. ACM*, 56(9):84–93.

[50] Forde, J. E. (2005). *Delay differential equation models in mathematical biology*. University of Michigan.

[51] Fort, J. and Méndez, V. (1999). Time-delayed theory of the neolithic transition in Europe. *Physical review letters*, 82(4):867.

[52] Fränzle, M., Herde, C., Teige, T., Ratschan, S., and Schubert, T. (2007). Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *Journal on Satisfiability, Boolean Modeling and Computation JSAT*, 1(3-4):209–236.

[53] Fränzle, M., Teige, T., and Eggers, A. (2010). Engineering constraint solvers for automatic analysis of probabilistic hybrid automata. *J. Log. Algebr. Program.*, 79(7):436–466.

[54] Galton, A. (2008). Temporal logic. *Stanford Encyclopedia of Philosophy*.

[55] Girard, A. (2005). Reachability of uncertain linear systems using zonotopes. In Morari, M. and Thiele, L., editors, *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, volume 3414 of *Lecture Notes in Computer Science*, pages 291–305. Springer Verlag.

[56] Girard, A. and Pappas, G. J. (2006). Verification using simulation. In Hespanha, J. P. and Tiwari, A., editors, *Hybrid Systems: Computation and Control, 9th International Workshop, HSCC 2006, Santa Barbara, CA, USA, March 29-31, 2006, Proceedings*, volume 3927 of *Lecture Notes in Computer Science*, pages 272–286. Springer Verlag.

[57] Girard, A. and Pappas, G. J. (2011). Approximate bisimulation: A bridge between computer science and control theory. *Eur. J. Control*, 17(5-6):568–578.

[58] Glass, L. and Mackey, M. C. (1988). *From clocks to chaos: the rhythms of life*. Princeton University Press.

[59] Gopalsamy, K. (2013). *Stability and oscillations in delay differential equations of population dynamics*, volume 74. Springer Science & Business Media.

[60] Gordon, M. J. (1988). Hol: A proof generating system for higher-order logic. *VLSI specification, Verification and Synthesis*, 35:73–128.

[61] Gorechi, H., Fuksa, S., Grabowski, P., and Korytowski, A. (1989). *Analysis and synthesis of time delay systems*. John Wiley & Sons, PWN-Polish Scientific Publishers Warszawa.

[62] Goubault, E., Mullier, O., Putot, S., and Kieffer, M. (2014). Inner approximated reachability analysis. In Fränzle, M. and Lygeros, J., editors, *17th International Conference on Hybrid Systems: Computation and Control (part of CPS Week), HSCC'14, Berlin, Germany, April 15-17, 2014*, pages 163–172. ACM.

[63] Gu, K., Chen, J., and Kharitonov, V. L. (2003). *Stability of time-delay systems*. Springer Science & Business Media.

[64] Gustafson, G. B. (2014). Systems of differential equations. In *Manuscript for Course Eng Math 2250-1 Spring 2014*, chapter 11. Dpt. of Mathematics, University of Utah.

[65] Harel, D. (1979). *First-Order Dynamic Logic*, volume 68 of *Lecture Notes in Computer Science*. Springer Verlag.

[66] Harel, D., Kozen, D., and Tiuryn, J. (2001). Dynamic logic. *SIGACT News*, 32(1):66–69.

[67] Hazewinkel, M. (2013). *Encyclopaedia of mathematics: A-integral—coordinates*, volume 1. Springer Verlag.

[68] Heffernan, J. M. and Corless, R. M. (2006). Solving some delay differential equations with computer algebra. *Mathematical Scientist*, 31(1):21–34.

[69] Henzinger, T. A., Kopke, P. W., Puri, A., and Varaiya, P. (1998). What's decidable about hybrid automata? *J. Comput. Syst. Sci.*, 57(1):94–124.

[70] Herrero, P., Calm, R., Vehí, J., Armengol, J., Georgiou, P., Oliver, N., and Tomazou, C. (2012). Robust fault detection system for insulin pump therapy using continuous glucose monitoring. *Journal of diabetes science and technology*, 6(5):1131–1141.

[71] Huang, C. and Chang, Q. (2001). Linear stability of general linear methods for systems of neutral delay differential equations. *Appl. Math. Lett.*, 14(8):1017–1021.

[72] Huang, Z., Fan, C., and Mitra, S. (2017). Bounded invariant verification for time-delayed nonlinear networked dynamical systems. *Nonlinear Analysis: Hybrid Systems*, 23:211–229.

[73] Hurd, J. (2003). First-order proof tactics in higher-order logic theorem provers. *Design and Application of Strategies/Tactics in Higher Order Logics, number NASA/CP-2003-212448 in NASA Technical Reports*, pages 56–68.

[74] Hutter, D., Langenstein, B., Sengler, C., Siekmann, J. H., Stephan, W., and Wolpers, A. (1996). Deduction in the verification support environment (VSE). In Gaudel, M. and Woodcock, J., editors, *FME '96: Industrial Benefit and Advances in Formal Methods, Third International Symposium of Formal Methods Europe, Co-Sponsored by IFIP WG 14.3, Oxford, UK, March 18-22, 1996, Proceedings*, volume 1051 of *Lecture Notes in Computer Science*, pages 268–286. Springer Verlag.

[75] Ikeda, K. and Matsumoto, K. (1987). High-dimensional chaotic behavior in systems with time-delayed feedback. *Physica D: Nonlinear Phenomena*, 29(1-2):223–235.

[76] Immler, F. (2014). Formally verified computation of enclosures of solutions of ordinary differential equations. In Badger, J. M. and Rozier, K. Y., editors, *NASA Formal Methods - 6th International Symposium, NFM 2014, Houston, TX, USA, April 29 - May 1, 2014. Proceedings*, volume 8430 of *Lecture Notes in Computer Science*, pages 113–127. Springer Verlag.

[77] Kaynama, S., Maidens, J. N., Oishi, M., Mitchell, I. M., and Dumont, G. A. (2012). Computing the viability kernel using maximal reachable sets. In Dang, T. and Mitchell, I. M., editors, *Hybrid Systems: Computation and Control (part of CPS Week 2012), HSCC'12, Beijing, China, April 17-19, 2012*, pages 55–64. ACM.

[78] Kline, M. (1990). *Mathematical Thought From Ancient to Modern Times: Volume 3*, volume 3. Oxford University Press, OUP USA.

[79] Kolmanovskii, V. and Myshkis, A. (2013). *Introduction to the theory and applications of functional differential equations*, volume 463. Springer Science & Business Media.

[80] Kong, S., Gao, S., Chen, W., and Clarke, E. (2015). dreach: δ-reachability analysis for hybrid systems. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, pages 200–205. Springer Verlag.

[81] Korda, M., Henrion, D., and Jones, C. N. (2013). Inner approximations of the region of attraction for polynomial dynamical systems. *IFAC Proceedings Volumes*, 46(23):534–539.

[82] Koymans, R. (1990). Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299.

[83] Kuang, Y. (1993). *Delay differential equations: with applications in population dynamics*, volume 191. Academic Press.

[84] Kupferschmid, S. and Becker, B. (2011). Craig interpolation in the presence of non-linear constraints. In Fahrenberg, U. and Tripakis, S., editors, *Formal Modeling and Analysis of Timed Systems - 9th International Conference, FORMATS 2011, Aalborg, Denmark, September 21-23, 2011. Proceedings*, volume 6919 of *Lecture Notes in Computer Science*, pages 240–255. Springer Verlag.

[85] Kurzhanski, A. B. and Varaiya, P. (2000). Ellipsoidal techniques for reachability analysis. In Lynch, N. A. and Krogh, B. H., editors, *Hybrid Systems: Computation and Control, Third International Workshop, HSCC 2000, Pittsburgh, PA, USA, March 23-25, 2000, Proceedings*, volume 1790 of *Lecture Notes in Computer Science*, pages 202–214. Springer Verlag.

[86] Lafferriere, G., Pappas, G. J., and Yovine, S. (2001). Symbolic reachability computation for families of linear vector fields. *J. Symb. Comput.*, 32(3):231–253.

[87] Lakshmanan, M. and Senthilkumar, D. V. (2011). *Dynamics of nonlinear time-delay systems*. Springer Science & Business Media.

[88] Le Guernic, C. and Girard, A. (2010). Reachability analysis of linear systems using support functions. *Nonlinear Analysis: Hybrid Systems*, 4(2):250–262.

[89] Liu, J., Zhan, N., and Zhao, H. (2011). Computing semi-algebraic invariants for polynomial dynamical systems. In *EMSOFT'11*, pages 97–106, New York, NY, USA. ACM.

[90] Liu, J., Zhan, N., and Zhao, H. (2012). Automatically discovering relaxed Lyapunov functions for polynomial dynamical systems. *Mathematics in Computer Science*, 6(4):395–408.

[91] Lohner, R. (1988). *Einschließung der Lösung gewöhnlicher Anfangs- und Randwertaufgaben und Anwendungen*. PhD thesis, Fakultät für Mathematik der Universität Karlsruhe, Karlsruhe.

[92] Lumb, P. M. (2004). *Delay differential equations: Detection of small solutions*. PhD thesis, University of Liverpool (Chester College of Higher Education).

[93] Mackey, M. C., Glass, L., et al. (1977). Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289.

[94] Maler, O. (2008). Computing reachable sets: An introduction. *Tech. rep. French National Center of Scientific Research*.

[95] Maler, O. and Nickovic, D. (2004). Monitoring temporal properties of continuous signals. In Lakhnech, Y. and Yovine, S., editors, *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, Joint International Conferences on Formal Modelling and Analysis of Timed Systems, FORMATS 2004 and Formal Techniques in Real-Time and Fault-Tolerant Systems, FTRTFT 2004, Grenoble, France, September 22-24, 2004, Proceedings*, volume 3253 of *Lecture Notes in Computer Science*, pages 152–166. Springer Verlag.

[96] Manna, Z. and Pnueli, A. (1992). *The temporal logic of reactive and concurrent systems - specification*. Springer Verlag.

[97] Manna, Z. and Sipma, H. (1998). Deductive verification of hybrid systems using step. In Henzinger, T. A. and Sastry, S., editors, *Hybrid Systems: Computation and Control, First International Workshop, HSCC'98, Berkeley, California, USA, April 13-15, 1998, Proceedings*, volume 1386 of *Lecture Notes in Computer Science*, pages 305–318. Springer Verlag.

[98] Maset, S. (2000). Stability of Runge-Kutta methods for linear delay differential equations. *Numerische Mathematik*, 87(2):355–371.

[99] McMillan, K. L. (2003). Interpolation and sat-based model checking. In Jr., W. A. H. and Somenzi, F., editors, *Computer Aided Verification, 15th International Conference, CAV 2003, Boulder, CO, USA, July 8-12, 2003, Proceedings*, volume 2725 of *Lecture Notes in Computer Science*, pages 1–13. Springer Verlag.

[100] Melham, T. F. (1993). *Higher Order Logic and Hardware Verification*, volume 31 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

[101] Mitchell, I. M. (2007). Comparing forward and backward reachability as tools for safety analysis. In Bemporad, A., Bicchi, A., and Buttazzo, G. C., editors, *Hybrid Systems: Computation and Control, 10th International Workshop, HSCC 2007, Pisa, Italy, April 3-5, 2007, Proceedings*, volume 4416 of *Lecture Notes in Computer Science*, pages 428–443. Springer Verlag.

[102] Moore, R. E. (1965). Automatic local coordinate transformation to reduce the growth of error bounds in interval computation of solutions of ordinary differential equations. In Ball, L. B., editor, *Error in Digital Computation*, volume II, pages 103–140. Wiley, New York.

[103] Mosaad, P. N., Fränzle, M., and Xue, B. (2016). Temporal logic verification for delay differential equations. In Sampaio, A. and Wang, F., editors, *Theoretical Aspects of Computing - ICTAC 2016 - 13th International Colloquium, Taipei, Taiwan, ROC, October 24-31, 2016, Proceedings*, volume 9965 of *Lecture Notes in Computer Science*, pages 405–421. Springer Verlag.

[104] Mosaad, P. N., Fränzle, M., and Xue, B. (2017). Model checking delay differential equations against metric interval temporal logic. *Scientific Annals of Computer Science*, 27(1):77–109.

[105] Nahhal, T. and Dang, T. (2007). Test coverage for continuous and hybrid systems. In Damm, W. and Hermanns, H., editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 449–462. Springer Verlag.

[106] Nam, P. T., Pathirana, P. N., and Trinh, H. (2015). Reachable set bounding for nonlinear perturbed time-delay systems: The smallest bound. *Appl. Math. Lett.*, 43:68–71.

[107] Nedialkov, N. S. (2006). Interval tools for odes and daes. In *Scientific Computing, Computer Arithmetic and Validated Numerics, 2006. SCAN 2006. 12th GAMM-IMACS International Symposium on*, pages 4–4. IEEE.

[108] Nedialkov, N. S. (2011). Implementing a rigorous ode solver through literate programming. In *Modeling, Design, and Simulation of Systems with Uncertainties*, pages 3–19. Springer Verlag.

[109] Nedialkov, N. S. and Jackson, K. R. (2002). The design and implementation of a validated object-oriented solver for IVPs for ODEs. Technical report, McMaster University.

[110] Neher, M., Jackson, K. R., and Nedialkov, N. S. (2007). On Taylor model based integration of ODEs. *SIAM J. Numerical Analysis*, 45(1):236–262.

[111] Niculescu, S.-I. (2001). *Delay effects on stability: a robust control approach*, volume 269. Springer Science & Business Media.

[112] Niculescu, S.-I. and Gu, K. (2012). *Advances in time-delay systems*, volume 38. Springer Science & Business Media.

[113] Oehlerking, J. (2011). *Decomposition of Stability Proofs for Hybrid Systems*. Doctoral dissertation, Carl von Ossietzky Universität Oldenburg, Department of Computing Science.

[114] Ouaknine, J. and Worrell, J. (2005). On the decidability of metric temporal logic. In *20th IEEE Symposium on Logic in Computer Science (LICS 2005), 26-29 June 2005, Chicago, IL, USA, Proceedings*, pages 188–197. IEEE Computer Society.

[115] Ouaknine, J. and Worrell, J. (2008). Some recent results in metric temporal logic. In Cassez, F. and Jard, C., editors, *Formal Modeling and Analysis of Timed Systems, 6th International Conference, FORMATS 2008, Saint Malo, France, September 15-17, 2008. Proceedings*, volume 5215 of *Lecture Notes in Computer Science*, pages 1–13. Springer Verlag.

[116] Owre, S., Rushby, J. M., and Shankar, N. (1992). PVS: A prototype verification system. In Kapur, D., editor, *Automated Deduction - CADE-11, 11th International Conference on Automated Deduction, Saratoga Springs, NY, USA, June 15-18, 1992, Proceedings*, volume 607 of *Lecture Notes in Computer Science*, pages 748–752. Springer Verlag.

[117] O'Connor, J. J. and Robertson, E. F. (2001). The mactutor history of mathematics archive. *World Wide Web page< http://www-history. mcs. st-and. ac. uk/>(accessed April 22, 2004)*.

[118] Paulson, L. C. (1994). *Isabelle - A Generic Theorem Prover (with a contribution by T. Nipkow)*, volume 828 of *Lecture Notes in Computer Science*. Springer Verlag.

[119] Plaku, E., Kavraki, L. E., and Vardi, M. Y. (2007). Hybrid systems: From verification to falsification. In Damm, W. and Hermanns, H., editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 463–476. Springer Verlag.

[120] Plaku, E., Kavraki, L. E., and Vardi, M. Y. (2009). Falsification of LTL safety properties in hybrid systems. In Kowalewski, S. and Philippou, A., editors, *Tools and Algorithms for the Construction and Analysis of Systems, 15th International Conference, TACAS 2009, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2009, York, UK, March 22-29, 2009. Proceedings*, volume 5505 of *Lecture Notes in Computer Science*, pages 368–382. Springer Verlag.

[121] Platzer, A. (2008). Differential dynamic logics - automated theorem proving for hybrid systems. Doctoral Dissertation, Carl von Ossietzky University of Oldenburg.

[122] Platzer, A. (2010). Differential-algebraic dynamic logic for differential-algebraic programs. *J. Log. and Comput.*, 20(1):309–352.

[123] Platzer, A. and Clarke, E. M. (2008). Computing differential invariants of hybrid systems as fixedpoints. In Gupta, A. and Malik, S., editors, *CAV'08*, volume 5123 of *LNCS*, pages 176–189. Springer Verlag.

[124] Pnueli, A. (1977). The temporal logic of programs. In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*, pages 46–57. IEEE Computer Society.

[125] Pola, G., Pepe, P., and Benedetto, M. D. D. (2015). Symbolic models for time-varying time-delay systems via alternating approximate bisimulation. *International Journal of Robust and Nonlinear Control*, 25:2328–2347.

[126] Pola, G., Pepe, P., Benedetto, M. D. D., and Tabuada, P. (2010). Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Systems & Control Letters*, 59(6):365–373.

[127] Prajna, S. and Jadbabaie, A. (2004). Safety verification of hybrid systems using barrier certificates. In Alur, R. and Pappas, G. J., editors, *Hybrid Systems: Computation and Control, 7th International Workshop, HSCC 2004, Philadelphia, PA, USA, March 25-27, 2004, Proceedings*, volume 2993 of *Lecture Notes in Computer Science*, pages 477–492. Springer Verlag.

[128] Prajna, S. and Jadbabaie, A. (2005). Methods for safety verification of time-delay systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4348–4353. IEEE.

[129] Prajna, S., Jadbabaie, A., and Pappas, G. J. (2007). A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8):1415–1428.

[130] Ramdani, N. and Nedialkov, N. S. (2009). Computing reachable sets for uncertain nonlinear hybrid systems using interval constraint propagation techniques. In Giua, A., Mahulea, C., Silva, M., and Zaytoon, J., editors, *3rd IFAC Conference on Analysis and Design of Hybrid Systems, ADHS 2009, Zaragoza, Spain, September 16-18, 2009*, volume 42 of *IFAC Proceedings Volumes*, pages 156–161. Elsevier.

[131] Ratschan, S. and She, Z. (2010). Providing a basin of attraction to a target region of polynomial systems by computation of Lyapunov-like functions. *SIAM J. Control and Optimization*, 48(7):4377–4394.

[132] Revert, A., Calm, R., Vehí, J., and Bondia, J. (2011). Calculation of the best basal-bolus combination for postprandial glucose control in insulin pump therapy. *IEEE Trans. Biomed. Engineering*, 58(2):274–281.

[133] Riazanov, A. and Voronkov, A. (2002). The design and implementation of VAMPIRE. *AI Commun.*, 15(2-3):91–110.

[134] Richard, J.-P. (2003). Time-delay systems: an overview of some recent advances and open problems. *automatica*, 39(10):1667–1694.

[135] Robinson, A. J. and Voronkov, A. (2001). *Handbook of automated reasoning*, volume 1. Elsevier.

[136] Sankaranarayanan, S. and Fainekos, G. E. (2012). Falsification of temporal properties of hybrid systems using the cross-entropy method. In Dang, T. and Mitchell, I. M., editors, *Hybrid Systems: Computation and Control (part of CPS Week 2012), HSCC'12, Beijing, China, April 17-19, 2012*, pages 125–134. ACM.

[137] Sankaranarayanan, S., Sipma, H. B., and Manna, Z. (2004). Constructing invariants for hybrid systems. In Alur, R. and Pappas, G. J., editors, *HSCC'04*, volume 2993 of *LNCS*, pages 539–554. Springer Verlag.

[138] Scheibler, K. (2016). *iSAT3 Manual*. Available at https://projects.avacs.org/attachments/download/671/isat3_manual-0.03-20161213.pdf.

[139] Shampine, L. F. and Thompson, S. (2001). Solving DDEs in Matlab. *Applied Numerical Mathematics*, 37(4):441–458.

[140] Sheeran, M., Singh, S., and Stålmarck, G. (2000). Checking safety properties using induction and a SAT-solver. In Jr., W. A. H. and Johnson, S. D., editors, *Formal Methods in Computer-Aided Design, Third International Conference, FMCAD 2000, Austin, Texas, USA, November 1-3, 2000, Proceedings*, volume 1954 of *Lecture Notes in Computer Science*, pages 108–125. Springer Verlag.

[141] Stauning, O. (1997). *Automatic Validation of Numerical Solutions*. PhD thesis, Technical University of Denmark, Lyngby.

[142] Stibitz, G. R. and Larrivee, J. A. (1957). *Mathematics and computers*. McGraw-Hill.

[143] Stursberg, O. and Krogh, B. H. (2003). Efficient representation and computation of reachable sets for hybrid systems. In Maler, O. and Pnueli, A., editors, *Hybrid Systems: Computation and Control, 6th International Workshop, HSCC 2003 Prague, Czech Republic, April 3-5, 2003, Proceedings*, volume 2623 of *Lecture Notes in Computer Science*, pages 482–497. Springer Verlag.

[144] Szydłowski, M. and Krawiec, A. (2005). The stability problem in the kaldor–kalecki business cycle model. *Chaos, Solitons & Fractals*, 25(2):299–305.

[145] Szydłowski, M., Krawiec, A., and Toboła, J. (2001). Nonlinear oscillations in business cycle model with time lags. *Chaos, Solitons & Fractals*, 12(3):505–517.

[146] Taylor, S. R. (2004). *Probabilistic properties of delay differential equations*. PhD thesis, University of Waterloo.

[147] Thompson, S. (2007). Delay-differential equations. *Scholarpedia*, 2(3):2367.

[148] Torrisi, F. D. (2003). *Modeling and reach-set computation for analysis and optimal control of discrete hybrid automata*. PhD thesis, ETH Zurich.

[149] Trinh, H., Nam, P. T., Pathirana, P. N., and Le, H. P. (2015). On backwards and forwards reachable sets bounding for perturbed time-delay systems. *Applied Mathematics and Computation*, 269:664–673.

[150] Tseitin, G. S. (1983). On the complexity of derivation in propositional calculus. In *Automation of reasoning*, pages 466–483. Springer Verlag.

[151] Varah, J. M. (1975). A lower bound for the smallest singular value of a matrix. *Linear Algebra and its Applications*, 11(1):3–5.

[152] Wang, T., Lall, S., and West, M. (2013). Polynomial level-set method for polynomial system reachable set estimation. *IEEE Trans. Automat. Contr.*, 58(10):2508–2521.

[153] Willé, D. R. and Baker, C. T. (1992). The tracking of derivative discontinuities in systems of delay-differential equations. *Applied Numerical Mathematics*, 9(3-5):209–222.

[154] Xue, B., Easwaran, A., and Cho, N. (2015). Poster: Towards robust artificial pancreas based on reachability analysis techniques. Workshop on Medical Cyber-Physical Systems (MCPS). Hosted at Cyber-Physical Systems Week 2015 in Seattle.

[155] Xue, B., Easwaran, A., Cho, N., and Fränzle, M. (2017a). Reach-avoid verification for nonlinear systems based on boundary analysis. *IEEE Transactions on Automatic Control*, 62(7):3518–3523.

[156] Xue, B., Mosaad, P. N., Fränzle, M., Chen, M., Li, Y., and Zhan, N. (2017b). Safe over- and under-approximation of reachable sets for delay differential equations. In Abate, A. and Geeraerts, G., editors, *Formal Modeling and Analysis of Timed Systems - 15th International Conference, FORMATS 2017, Berlin, Germany, September 5-7, 2017, Proceedings*, volume 10419 of *Lecture Notes in Computer Science*, pages 281–299. Springer Verlag.

[157] Xue, B., She, Z., and Easwaran, A. (2016). Under-approximating backward reachable sets by polytopes. In Chaudhuri, S. and Farzan, A., editors, *Computer Aided Verification - 28th International Conference, CAV 2016, Toronto, ON, Canada, July 17-23, 2016, Proceedings, Part I*, volume 9779 of *Lecture Notes in Computer Science*, pages 457–476. Springer Verlag.

[158] Xue, B., Wang, Q., Feng, S., and Zhan, N. (2018). Over-and under-approximating reachable sets for perturbed delay differential equations. *arXiv preprint arXiv:1812.11718*.

[159] Yi, S. and Ulsoy, A. G. (2006). Solution of a system of linear delay differential equations using the matrix lambert function. In *American Control Conference, 2006*, pages 6–pp. IEEE.

[160] Yi, S., Ulsoy, A. G., and Nelson, P. W. (2006). Solution of systems of linear delay differential equations via laplace transformation. In *Decision and Control, 2006 45th IEEE Conference on*, pages 2535–2540. IEEE.

[161] Zhang, Y. and Sun, J. (2005). Stability of impulsive linear differential equations with time delay. *IEEE Trans. on Circuits and Systems*, 52-II(10):701–705.

[162] Zou, L., Fränzle, M., Zhan, N., and Mosaad, P. N. (2015). Automatic verification of stability and safety for delay differential equations. In Kroening, D. and Pasareanu, C. S., editors, *Computer Aided Verification - 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, volume 9207 of *Lecture Notes in Computer Science*, pages 338–355. Springer Verlag.

[163] Zou, L., Lv, J., Wang, S., Zhan, N., Tang, T., Yuan, L., and Liu, Y. (2013). Verifying chinese train control system under a combined scenario by theorem proving. In Cohen, E. and Rybalchenko, A., editors, *Verified Software: Theories, Tools, Experiments - 5th International Conference, VSTTE 2013, Menlo Park, CA, USA, May 17-19, 2013, Revised Selected Papers*, volume 8164 of *Lecture Notes in Computer Science*, pages 262–280. Springer Verlag.