Reachability Analysis of Hybrid Systems via Predicate and Relational Abstraction
(述語と関係を用いた抽象化による ハイブリッドシステムの到達可能性解析)

by

Kengo Kido
木戸 肩吾

A Doctor Thesis
博士論文

Submitted to
the Graduate School of the University of Tokyo
on December 8, 2017
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Information Science and Technology
in Computer Science

Thesis Supervisor: Naoki Kobayashi　小林 直樹
Professor of Computer Science

ABSTRACT

Cars, robots, drones, medical equipment and other embedded systems are ubiquitous these days. In such systems, the embedded computer (controller) and the controlled physical object (plant) interact with each other via sensors and actuators. The combined system of the controller software and the physical plant is called a *cyber-physical system (CPS)*.

Since the plant in a CPS behaves in the physical environment, a malfunction of the system has physical effects. Therefore safety assurance of CPSs is an important topic. There are mainly two approaches for safety assurance of CPSs: one is *simulation/testing*, and the other is *formal methods*. Currently, the dominant approach for safety assurance of CPSs in industry is simulation and testing, in which the behavior of the system is checked by giving some concrete inputs to the system. However, testing may miss a malfunction because for complex systems we cannot check all the possible inputs. Thus taking advantage of formal methods, which are mathematically valid techniques to analyze the behavior of systems, is recognized as an important direction of safety assurance.

The main topic of this thesis is formal methods for CPSs. More concretely, among a lot of techniques in formal methods, we focus on *reachability analysis* in this thesis. Compared to software that works in desktop computers, CPSs have special characteristics. Among those characteristics of CPSs, we focus on its *real-time property* and *hybrid dynamics*.

First, we consider the real-time property. One of the common ways a CPS works is that the sensing occurs periodically, and the controller determines the next input to the plant based on the sensed data. In an ideal system, the input to the plant is changed periodically, at the same moment as the data is sensed. However, in practice, computation and data transfer cause delays from the sensing of the data to the change of the input. The usage of *networked control*—digital control of physical systems via computer networks—makes analyzing the effect of delays more important. In networked control, plants and controllers are separated physically. This physical distance leads to inevitable communication delays. What is worse, the use of *cloud control* makes both physical and logical distances between system components even bigger and unpredictable. In such settings, precise estimation of communication delays is often hard, and the delays have effects on the behavior of the system that cannot be ignored.

Then, we consider the hybrid dynamics of CPSs. A CPS consists of a plant and a controller, and it exhibits the hybrid behavior of continuous behavior in the physical environment and discrete behavior of the controller software. In that sense, a CPS is called a *hybrid system*.

Taking the above two characteristics into consideration, we consider a hybrid system with bounded time-delays as a model of a CPS. In this thesis, we introduce a methodology to calculate an overapproximation of its reachable set in two steps: an overapproximation of the errors due to time-delays in the first step, and an overapproximation of the reachable set of the delay-free model in the second. This separation of concern enables us to replace the methodology used in each step with another existing methodology. In particular, for the second step where we do reachability analysis of delay-free hybrid systems, there is a lot of existing work and we can choose a suitable technique depending on the system under consideration.

The first step is to approximate a CPS with time-delays with the ideal model without time-delays. Under the assumption of incremental stability, our proposed methodology gives an upper bound of the error between the system with bounded delays and the system without delays. In order to do this, we construct a transition

system whose state is a triple consisting of a memory state, time and a mode. We define a premetric on these extended states, and give an upper bound of the premetric by constructing an approximate bisimulation relation on the transition system. As the first example, we show that our method can successfully analyze the effect of the delays of the boost DC-DC converter that is used in hybrid and electric vehicles. The dynamics of this example is characterized by linear ODEs, but we also show that our method can be applied to nonlinear ODEs using an example of nonlinear water tank.

In the above methodology to approximate a delayed system with its delay-free model, we computed an upper bound of the error of the two states at the same time instant. We extend this methodology by changing the definition of the premetric so that it can give a more precise upper bound of the Skorokhod distance between the trajectories of the two systems. The Skorokhod distance allows some mismatches in time. In spite of the timing mismatches, the resulting overapproximation of the Skorokhod distance can be used, for example, to reduce the reachability analysis of the delayed system to that of its delay-free model.

Then, we compute an overapproximation of the reachable set in the second step. For this purpose, we extend Cousot and Cousot's *abstract interpretation* framework to hybrid systems, using Robinson's *nonstandard analysis (NSA)*. The approach of using NSA for verification of hybrid systems has been introduced by Suenaga and Hasuo. They model hybrid systems as programs with an infinitesimal constant, by regarding continuous behavior as infinitely many infinitesimal discrete jumps. This approach enables us to extend usual formal methods for discrete systems to hybrid systems almost as they are, thanks to the *transfer principle* in NSA. As a result, our extended abstract interpretation framework enables us a sound approximation of the reachable sets of hybrid systems. Using the domain of convex polyhedra, we can analyze linear water tank example with fixed time-delays, and nonlinear water tank without delays.

As mentioned above, the proposed methodologies for both of the two steps are applicable to nonlinear dynamics. Thus, by combining the two analysis, we successfully analyze the nonlinear water tank with bounded delays, and verify that the water level stays within a certain region. We also illustrate an advantage of the two-step analysis—the methodologies proposed in this thesis for each step can be replaced with another existing methodology—using the example of boost DC-DC converter. For this example, we can even synthesize a controller using the existing work by Girard instead of the extension of abstract interpretation.

論文要旨

自動車，ロボット，ドローン，医療機器といったシステムにはコンピュータが組み込まれ，それによって制御されて動作する．これらに組み込まれたコンピュータ（コントローラ）はセンサとアクチュエータを介して物理的な物体（プラント）と相互に作用する．コントローラとプラントとを合わせたシステム全体を物理情報システム（cyber-physical system, CPS）と呼ぶ．CPS のプラントは物理的な振る舞いをし，その誤動作は物理的な影響を持つため，CPS の安全保証は重要な課題である．CPS の安全保証へのアプローチには，大きく分けてテスト，形式手法の二つがある．現在のところ産業界における CPS の安全保証に向けた主要な取り組みはテストである．しかしながらテストはシステムに入力を与えてその動作を確認するものであり，未確認の入力が残るため，誤動作を見逃す可能性がある．そのため，数学的に正しい解析を行う形式手法の活用が安全保証に向けた重要な方向性であると考えられている．

本論文では CPS に対する形式手法，その中でも特に CPS の形式手法の中で重要な役割を果たす過大近似を扱う．CPS には従来のデスクトップコンピュータで動作するソフトウェアでは考えられてこなかった特有の性質が多数ある．私の提案する手法は，そのうちリアルタイム性，ハイブリッド性の二つに着目し，CPS の振る舞いの到達可能性を二段階に分けて過大近似するものである．まずリアルタイム性を考える．CPS ではセンサでプラントの状態をセンシングし，そのセンシングしたデータを用いてコントローラがプラントをリアルタイムにアクチュエートする．理想的なシステムではセンシングと同時にアクチュエーションがおきるが，実際にはデータの転送やコントローラにおける計算によりセンシングからアクチュエーションまでに遅延が生じる．特にコントローラがプラントとネットワークを通じて繋がっているようなネットワークコントロール，さらにはコントローラがクラウド上にあるクラウドコントロールではデータ転送による遅延は避けられず，その影響の考慮が必要になる．二つ目にハイブリッド性を考える．これは，微分方程式で表されるような物理環境下における連続的な振る舞い（例えばアクセル開度に応じた自動車の速度の連続的変化）と，プログラムとして表されるコントローラの離散的な振る舞いの双方を含んでいるという性質である．この意味で CPS はハイブリッドシステムと呼ばれる．

上記二つの性質を考慮に入れ，本論文ではセンシングからアクチュエーションまでに遅延を含むハイブリッドシステムを CPS のモデルとして考える．そしてその到達可能領域の過大近似を，遅延による影響の過大近似と遅延を含まないシステムの到達可能領域の過大近似の二段階に分けて計算する手法を提案する．解析を二段階に分けたことにより，各段階での解析に用いるために本論文で提案する手法の代わりに，ほかの既存の解析手法を使用することもできる．特に遅延を含まないハイブリッドシステムの到達可能性解析を行う第二段階目については多数の既存研究が存在するため，解析したいシステムに応じて適切な手法を用いることができる．

まず一段階目として，遅延を含むシステムを遅延の含まれない理想的なシステムで

近似し，二つのシステム間の振る舞いの差を過大近似する．本論文の主要な結果として，増分安定性を保証するリヤプノフ関数が与えられたとき，システムのセンシングからアクチュエーションまでの遅延の上限を用いて遅延の有無による誤差の上限を与えた．この上限をもとめるため，まずそれぞれのシステムからメモリ状態，時刻，モードの三つ組からなる状態遷移系を作る．この状態間に前距離を適切に定め，近似双模倣関係を導出することで差の上限が計算される．例として，実際に HV や EV の自動車で用いられる DC/DC 昇圧回路を遅延のない理想的なシステムで近似した．この DC/DC 昇圧回路の例では微分方程式は線形であるが，この手法は非線形なシステムに対しても適用可能である．実際に非線形な water tank の例を用いて，非線形なシステムにも適用できることも示した．

ここで述べた手法では，二つのシステムの同時刻の状態を比較した差の上限を求めたが，結果のさらなる拡張として，前距離の定義を変更することにより二つのシステムの振る舞いの Skorokhod 距離の上限をより精密に求めることのできる手法も提案する．Skorokhod 距離では，時間軸をずらした状態を比較することが可能になる．この拡張によって，例えば到達可能性解析等を行う際には，はじめに定義した前距離によるものと比較して，より正確な近似を求めることが可能となる．

次に二段階目として，遅延を含まないハイブリッドシステムの到達可能領域を過大近似する．本論文では Cousot・Cousot による抽象解釈を，Robinson の超準解析を用いることでハイブリッドシステムに適用できるよう拡張し，到達可能領域を過大近似する手法を提案する．超準解析をハイブリッドシステムの検証に用いるアプローチは末永・蓮尾によって提案されている．そのアプローチでは，連続的な振る舞いを無限に繰り返される無限小の離散的な変化とみなすことにより，ハイブリッドシステムが無限小の定数を含むプログラムとしてモデルされる．さらに，超準解析の移転原理によって，通常の形式手法がハイブリッドシステムにほぼそのままの形で拡張される．結果として，本論文で提案する拡張された抽象解釈により，ハイブリッドシステムの到達可能性を大きく見積もることが可能となる．ここでは凸多角形の抽象領域を用いて，線形な water tank で一定の遅延を含むものと，非線形な water tank で遅延を含まないものが解析できることを示した．

以上の二段階それぞれに関して私の提案する手法はともに非線形のダイナミクスに対しても適用可能である．実際に例としてスイッチングに遅延を含む非線形な water tank の二つの段階の過大近似を順に計算し，結果を組み合わせることで，システムの状態が危険な領域に到達しないことを数学的に保証することができた．また，解析を二段階に分けたことによって各段階を既存手法で置き換えることができるという利点も，線形な DC/DC 昇圧回路の例を用いて例示する．具体的にこの例では，解析の二段階目として抽象解釈の拡張の代わりに Girard による手法を用いることで，到達可能性解析のみならず，より難しい制御器生成まで行うことが可能であることを示した．

Acknowledgements

My Ph.D. research described in this thesis cannot be accomplished without the support from many people. First of all, I would like to express my deepest gratitude to my two supervisors, Dr. Ichiro Hasuo and Professor Naoki Kobayashi. Dr. Hasuo had been my supervisor since I was writing my Bachelor thesis until the second year of the Ph.D. program. He has also taken care of my daily supervision after I moved to Kobayashi laboratory. He always gave me pertinent comments every time I presented my progress. It encourages me to do further research and finally leads to this thesis. He was very supportive of me on my Ph.D. research. When I had something to discuss, he always took time even when he was very busy. I am proud of doing my Ph.D. research under his supervision. In the third year of the Ph.D. program, Professor Naoki Kobayashi was my supervisor. Even though my daily supervision was consigned to Dr. Hasuo, he gave me a chance to have a meeting once a month and gave a lot of comments and suggestions on the direction of my Ph.D. studies.

In the third year, I was a research assistant at ERATO Metamathematics for Systems Design Project (ERATO MMSD) and I am thankful to the members of the project, especially to Dr. Fuyuki Ishikawa, the group leader of G3, for his useful comments at weekly G3 seminars. I also would like to thank my co-supervisor at ERATO MMSD, Dr. Sean Sedwards, for having weekly meetings and giving me a lot of advice.

Throughout my Ph.D. studies, I belonged to the Graduate Program for Social ICT Global Creative Leaders (GCL) and I would like to thank the professors for supporting us. I have learned a lot that could not be learned if I were not in the program, such as how to facilitate workshops. There are many professors and students from other graduate schools in GCL, and even people from industry or government offices are involved. It was a great opportunity to look at my own research objectively.

During my Ph.D. studies (including my master's,) I was fortunately given opportunities to visit Dr. Kohei Suenaga at Kyoto University, Dr. Swarat Chaudhuri at Rice University, Dr. Xavier Rival at École Normale Supérieure and Professor Krzysztof Czarnecki at University of Waterloo, as GCL internships. I would like to thank each of them for hosting me. Discussion with them and other members of each group was very inspiring. Regarding my visit to the University of Waterloo, I also would like to thank Mr. Hisashi Miyashita at Maplesoft and his family. They kindly invited me to their home very often and gave me great advice on my future career.

I would like to thank Professor Masami Hagiya, Professor Reiji Suda, Professor Yoshihide Yoshimoto, Professor Shinpei Kato and Professor Shoji Yuen—my Ph.D. thesis examiners—for their careful reading. This thesis has been improved very much thanks to their constructive suggestions and comments. I would also

like to express my gratitude to the admin support at Hasuo lab, ERATO MMSD, Kobayashi lab, Department of Computer Science, and GCL.

It is impossible to name here all of the friends, colleagues and other people, but I am really appreciate for them to spend some time with me together. Especially I am very much grateful to my family and my girlfriend Maho Tsugawa, for all their encouragement and support.

Finally, I would like to thank JSPS Research Fellowship for their financial support throughout my Ph.D. studies. Their support was necessary for me to focus on my Ph.D. research.

Contents

List of Figures

List of Codes

Chapter 1

Introduction

Cars, robots, drones, medical equipment and other embedded systems are ubiquitous these days. In such systems, the embedded computer (controller) and the controlled physical object (plant) interact with each other via sensors and actuators. The combined system of the controller software and the physical plant is called a *cyber-physical system (CPS)*. Compared to software that works in desktop computers, a CPS has some special characteristics such as constraints on the energy efficiency or uncertainty of the data obtained from the sensors, which pose new challenges (a survey can be found in [88], for example).

Among those characteristics of a CPS, one of the most important aspects is that it exhibits hybrid behavior of continuous behavior of the plant in the physical environment and discrete behavior of the controller software. In that sense, a CPS is called a *hybrid system*. The continuous behavior of the plant is often modeled by ordinary differential equations (ODEs). The discrete behavior of the controller is defined by a program. Two communities have been working together to analyze the behavior of such systems. One is control theory that has originally worked on continuous physical dynamics. The other is formal methods community in computer science that has dealt with discrete dynamics of software. For researchers in formal methods, the main challenge of extending their work to hybrid systems is to incorporate continuous flow dynamics. Most of existing work such as those based on hybrid automata [6] uses ODEs explicitly in the syntax to model continuous behavior.

Another important aspect of a CPS is its real-time property. One of the common ways a CPS works is that the sensor obtains the data of the plant periodically, and the controller determines the next input to the plant based on the sensed data. In an ideal system, the input to the plant is changed periodically, at exactly the same moment as the data is sensed. However, in practice, computation and data transfer cause delays from the sensing of the data to the change of the input. The usage of *networked control*—digital control of physical systems via computer networks—makes analysis of the effect of delays more important. In networked control, plants and controllers are separated physically. This physical distance leads to inevitable communication delays. What is worse, *cloud control* makes both physical and logical distances between the plant and the controller even bigger and unpredictable. In such settings, precise estimation of communication delays is often hard and the delays have effects on the behavior of the system that cannot be ignored.

Taking the above two characteristics into consideration, the overall system

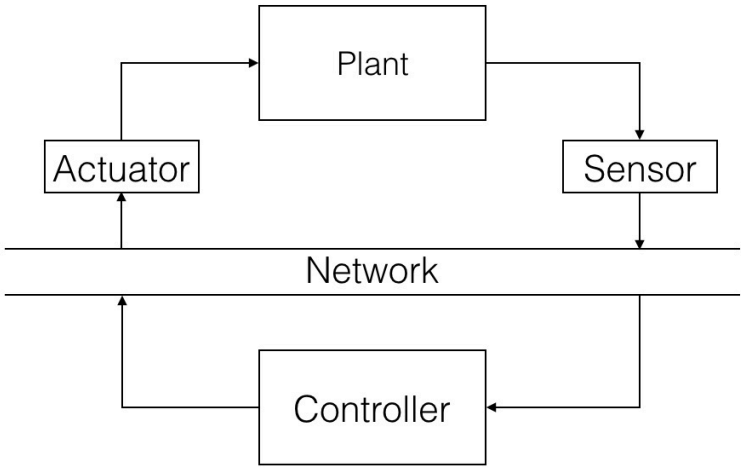

Figure 1.1: Architecture of our target system (omitting D/A and A/D converters).

we are interested in looks like Fig. 1.1. Since the behavior of a CPS includes physical behavior realized by its actuators, a malfunction of the system has physical effects. For example, a malfunction of the controller software embedded in a car could lead to an accident resulting in injury or death. Thus safety assurance of CPSs is obviously an important topic, and many researchers and also people in industry are working on this challenge. There are mainly two approaches for safety assurance of CPSs. One is *simulation/testing*, and the other is *formal methods*. In testing, the behavior of the system is checked by giving some concrete inputs to the system. In formal methods, we analyze the behavior of the system using mathematically valid techniques. Currently, the dominant approach for safety assurance of CPSs in industry is simulation and testing. One of the main reasons is the limited applicability and scalability of formal methods. However, testing may miss a malfunction because for complex systems we cannot check all the possible inputs. Thus taking advantage of formal methods is recognized as an important direction of safety assurance.

The topic of this thesis is formal methods for CPSs. More concretely, among a lot of techniques in formal methods, we focus on *reachability analysis* in this thesis. Reachability analysis computes an overapproximation of the set of reachable states, and plays the central role in verification of safety specifications. We propose a reachability analysis methodology for CPSs with hybrid dynamics and bounded time-delays. A distinctive feature of our methodology is that we separate the reachability analysis of a delayed CPS into the following two steps: 1) approximation of a delayed CPS with its delay-free model; and 2) reachability analysis of the delay-free model. The detailed comparison with existing work for reachability analysis of hybrid dynamics with delays based on symbolic abstraction will be in §7.4.

The first step of our proposed methodology is to approximate a CPS with time-delays with the ideal one without time-delays. It can be applied to possibly nonlinear dynamics under the assumption of incremental stability. Given a Lyapunov function that certifies incremental stability, our proposed methodology gives an upper bound of the error between the system with bounded delays and

the system without delays. In order to do this, we construct transition systems whose state is a triple consisting of a memory state, time and a mode. We define a premetric on these extended states and give an upper bound of the premetric by constructing an approximate bisimulation relation between the transition system for delayed system and the one for its delay-free model. As the first example, we show that our method can successfully analyze the effect of the delay of the boost DC-DC converter that is used in hybrid and electric vehicles. The dynamics of this example is characterized by linear ODEs, but we also show that our method can be applied to nonlinear ODEs using nonlinear water tank example.

In the above methodology to approximate a system with delays with the one without delays, we computed an upper bound of the pointwise metric that gives the error of the two states at the same time instant. However, this pointwise metric sometimes returns large distances even if two systems are close in terms of, for example, reachability. We extend this methodology by changing the definition of the premetric so that it can give a more precise upper bound of the Skorokhod distance between the trajectories of the two systems. The Skorokhod metric allows mismatches in the timeline. The resulting overapproximation of the Skorokhod distance can be used not only to the reachability analysis but also for verification of other temporal specifications.

Then, we compute an overapproximation of the reachable set of the delay-free model in the second step. For this purpose, we extend Cousot and Cousot's *abstract interpretation* framework to hybrid systems, using Robinson's *nonstandard analysis (NSA)*. The approach of using NSA for verification of hybrid systems has been introduced by Suenaga and Hasuo. They model hybrid systems as programs with an infinitesimal constant, by regarding continuous behavior as infinitely many infinitesimal discrete jumps. This modeling does not rely on ODEs explicitly, and it enables us to extend usual formal methods for discrete systems to hybrid systems almost as they are, thanks to the *transfer principle* in NSA. As a result, our extended abstract interpretation framework enables us a sound approximation of the reachable sets of hybrid systems. Using the domain of convex polyhedra, we can analyze linear water tank example with fixed time-delays, and nonlinear water tank without delays.

As mentioned above, the proposed methodologies for both of the two steps are applicable to nonlinear dynamics. Thus, by combining the two analysis, we successfully analyze the nonlinear water tank with bounded delays and verify that the water level stays within a certain region.

Since we separated the analysis into the two steps, the methodology used in each step can be substituted by another methodology that is suitable for the system we are interested in. In particular, for the second step where we do reachability analysis of delay-free hybrid systems, there is a lot of existing work. For the example of boost DC-DC converter, we can even synthesize a controller using the existing work by Girard instead of the extension of abstract interpretation.

In the subsequent sections, we discuss the background and our contribution in each topic of the thesis.

1.1 Approximate Bisimulation for Switching Delay

In this section, we discuss the background and our contribution regarding the first topic of this thesis—calculation of an error bound between a CPS with time-delays and the ideal system without time-delays.

1.1.1 Background

Delays occur in various steps in the control of a CPS. One major source of delays is the computation time. In Fig. 1.1, even if we assume that the sensing occurs exactly periodically, the input to the actuator can be changed only after the computation is finished. Another major kind of delays is the ones caused by data transfer. This kind of delay is not small enough to ignore since the use of the networked control is getting common. In networked control, the sensed data from the plant are transferred via network. Then some computation is executed in the controller. Finally, the input for the actuator is sent back from the controller via network again (see Fig. 1.1.)

We introduce a methodology to bound the error of the behavior of a CPS caused by these potential time-delays. Given the hybrid nature of CPSs, a natural idea is to use the notion of *approximate bisimulation*—an achievement of the hybrid systems community that combines ideas from control theory and computer science. We give an approximate bisimulation relation between an actual system (with potential time-delays) and an idealized system without delays. The latter system is simpler and one can use it for the purpose of analysis; then the result of the analysis is also true for the actual system, up-to certain errors that are bounded by the approximate bisimulation. The idealized system without delays can also be used for control design; then the resulting controller is guaranteed to work well with the actual system up-to certain errors.

Approximate bisimulation is a topic that gathers attention from many researchers in control theory and computer science. The notion of approximate bisimulation was first introduced in [41] as a quantitative relaxation of bisimulation, a well-established equivalence notion between discrete transition systems [74]. The theory of approximate bisimulation has been rapidly developed since then; one of the notable results is its connection to *incremental stability* [43, 77]. These theoretical results have found a number of successful applications, too. A prototypical application is the construction of discretized and symbolic models, and control synthesis therein via discrete synthesis techniques [40].

In this thesis, we shall focus on switched systems. A switched system is a common model of a CPS, which can be seen as a more abstract model than a hybrid automaton [6]. In switched systems, a plant has finitely many operation modes, and the possible mode changes will be given as a set of switching signals that are sent from a controller. Usually, the definition of the set of switching signals ignores the detailed restrictions such as guard or invariant conditions in hybrid automata. This simple model enables us to express various real-world networked control systems, as is argued e.g. in [43].

1.1.2 Contribution

Our technical developments are based on the previous work [43] in which approximate bisimulation is used for the purpose of discretization of nonlinear switched systems. Our system model Σ_{τ,δ_0} is a (potentially nonlinear) switched system where switching signals are nearly periodic with a fixed period τ; the system exhibits potential switching delays within a fixed maximum delay δ_0.

More concretely, our technical contributions are as follows. We show how a switched system Σ_{τ,δ_0} with nearly periodic switching signals can be turned into a transition system $T(\Sigma_{\tau,\delta_0})$. Between the resulting transition system $T(\Sigma_{\tau,\delta_0})$ and the one $T(\Sigma_\tau)$ derived from the ideal delay-free system Σ_τ, we establish an approximate bisimulation that witnesses an upper bound of the error between the behaviors of Σ_{τ,δ_0} and Σ_τ. The approximate bisimulation is derived from the same incremental stability assumption as in [43] (namely δ-$GUAS$). More specifically, we introduce a construction that turns Lyapunov-type witnesses for δ-GUAS into an approximate bisimulation. While we use the same incremental stability assumption as in [43], we also identify some additional technical constraints (such as Assumption 3.4.1) that are unique to the current problem setting. Using the proposed method, we successfully analyze the effect of the time-delays in the examples of boost DC-DC converter and nonlinear water tank.

1.2 Skorokhod Distance Caused by Switching Delays

The proposed methodology discussed in the previous section is extended so that timing mismatches are allowed. More concretely, the extended methodology gives an upper bound of the *Skorokhod metric*, to obtain a tighter error bound that can be used for reachability analysis.

1.2.1 Background

In the methodology we discussed in the previous section, the states in the transition system are equipped with a premetric that is defined so that it would approximate the pointwise distance between trajectories. The pointwise distance is a very simple distance that compares two states at the same time instant (see Fig. 1.2).

However, the pointwise metric sometimes returns large distances even if two systems are close in terms of, for example, reachability. See the example in Fig. 1.3. The pointwise distance gives the length of the black arrows at switchings. Once we obtain this distance (say ε), the reachability of the blue behavior is overapproximated by an expansion of the reachability of the red one by ε. However, the actual reachability of the two systems in red and blue is the same.

The source of this unnecessarily large distance is that the pointwise metric does not allow any mismatches of the timing. Quantifying the closeness between trajectories that allows timing mismatches has recently been studied in the field of conformance testing. A beginning of the study in this direction is the conformance degree based on $(T, J, (\tau, \varepsilon))$-closeness introduced in [2]. In this definition of closeness, the parameter τ is the closeness in time and ε is the closeness in space. Then in [34], the conformance between two trajectories was defined using the Skorokhod metric [4], which is related to Fréchet distance as discussed

in [68, 69].

These metrics are defined so that it captures the closeness both in time and space. One of the advantages of these metrics is the transference of temporal properties, which means that if two trajectories are close with respect to the metric, their validity of temporal logic specifications is also "close". The conformance degree based on $(T, J, (\tau, \varepsilon))$-closeness can transfer the properties in a variant of metric temporal logic (MTL). The Skorokhod metric can transfer timed linear temporal logic (TLTL) or Freeze linear temporal logic (FLTL).

1.2.2 Contribution

Our target system is exactly the same as the one discussed in the previous section—incrementally stable switched systems with delays and without delays. We extend the previous methodology by changing the definition of the premetric so that it can give a more precise upper bound of the Skorokhod distance between the trajectories of the two systems (see Fig. 1.2). Then, we construct an approximate bisimulation relation that bounds the new premetric. The examples showcase that the reachability analysis becomes more precise using this extended methodology. This extended methodology also has an advantage that it is applicable to the cases where the maximum delay δ_0 is larger than the switching period τ, which the previous methodology could not deal with. Although we only focus on reachability in this thesis, the resulting upper bound is also useful for verification of FLTL specifications.

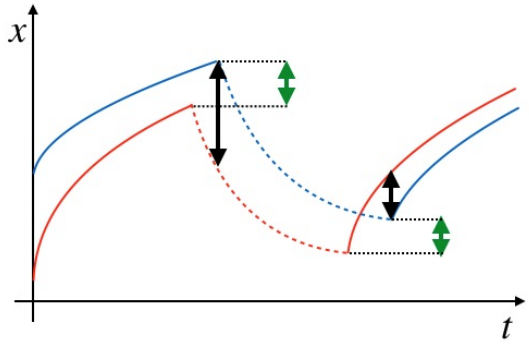

Figure 1.2: For each switching, the premetric for the pointwise distance is shown by a black arrow, and the premetric for the Skorokhod distance is a green arrow. The behavior of the periodic system is in red, and the delayed system is in blue. Solid and broken curves indicate two different modes.

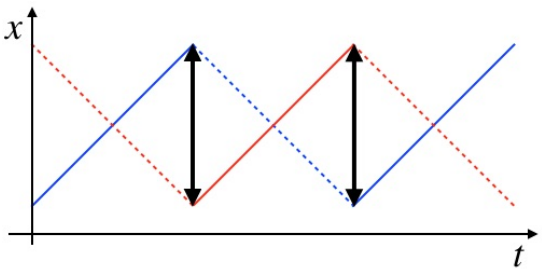

Figure 1.3: The behavior of two systems are presented in red and blue. Solid and broken lines indicate two different modes.

1.3 Extension of Abstract Interpretation with Infinitesimals

In this section, we discuss a methodology to overapproximate the reachable set of hybrid systems (without delays). The proposed framework is an extension of well-known abstract interpretation with NSA.

1.3.1 Background

Hybrid systems exhibit both discrete *jump* and continuous *flow* dynamics. Researchers in control theory and formal methods have been working together to analyze the behavior of such systems. For researchers in formal methods who originally work on discrete behavior of software, the main challenge is how to deal with continuous behavior of the physical plant. Most existing work models continuous behavior using ODEs explicitly An example is hybrid automata [6] (see the discussion of related work in Chapter 7 for more details.)

In [91], instead, an alternative approach of *nonstandard static analysis*—combining *static analysis* and NSA—has been proposed. Its basic idea is to introduce a constant `dt` for an *infinitesimal* (i.e. infinitely small) value, and *turn flow into infinitely many jumps*. With the constant `dt`, the continuous operation of integration can be represented by a while-loop. To this program with an infinitesimal constant, existing discrete techniques such as Hoare-style program logics [55] can be applied almost as they are. For a rigorous mathematical development, they employ NSA beautifully formalized by Robinson [84].

Concretely, in [91] they took the common combination of a WHILE-language and a Hoare logic (e.g. in the textbook [95]); and added a constant `dt` to obtain a modeling and verification framework for hybrid systems. Its components are called $\text{WHILE}^{\texttt{dt}}$ and $\text{HOARE}^{\texttt{dt}}$. The soundness of $\text{HOARE}^{\texttt{dt}}$ is proved against denotational semantics defined in the language of NSA. Then in [50], they presented a prototype automatic theorem prover for $\text{HOARE}^{\texttt{dt}}$. In [92], they applied the same idea to stream processing systems, realizing a verification framework for *signal processing* as in Simulink.

These technical developments are based on the idea of so-called *sectionwise execution*. Although the methodology we propose in this thesis does not rely explicitly on it, it is still useful to give some intuition of $\text{WHILE}^{\texttt{dt}}$ modeling in nonstandard static analysis. See the following example.

Example 1.3.1. Let c_{elapse} be the program in Code 1.1. The value of `dt` is infinitesimal; therefore the `while` loop will not terminate within finitely many steps. Nevertheless, it is somehow intuitive to expect that after an "execution" of this program, the value of t should be infinitesimally close to 1 and larger than it.

```
1 t := 0
2  while t <= 1 do
3    t := t+dt
```

Code 1.1: c_{elapse}

```
1 t := 0
2  while t <= 1 do
3    t := t+1/(i+1)
```

Code 1.2: $c_{\mathsf{elapse}}|_i$

One possible way of thinking is to imagine *sectionwise execution*. Note that an infinitesimal number can be approximated progressively by the sequence

$$(\,1,\ \tfrac{1}{2},\ \tfrac{1}{3},\ \ldots,\ \tfrac{1}{i+1},\ \ldots\,)\ .$$

Based on this approximation of the infinitesimal constant $\mathtt{dt}$, we consider the *i-th section* of the program c_{elapse}, denoted by $c_{\mathsf{elapse}}|_i$ and shown in Code 1.2, for each natural number i. Concretely, $c_{\mathsf{elapse}}|_i$ is obtained by replacing the infinitesimal $\mathtt{dt}$ in c_{elapse} with $\frac{1}{i+1}$. Informally $c_{\mathsf{elapse}}|_i$ is the "i-th approximation" of the original c_{elapse}.

Note that each section $c_{\mathsf{elapse}}|_i$ is just a usual imperative program without infinitesimal constants. It terminates within finite steps and yields $1+\frac{1}{i+1}$ as the value of t. Now we collect the outcomes of sectionwise executions and obtain a sequence

$$(\,1+1,\ 1+\tfrac{1}{2},\ 1+\tfrac{1}{3},\ \ldots,\ 1+\tfrac{1}{i+1},\ \ldots\,)\ , \tag{1.1}$$

which is thought of as a progressive approximation of the actual outcome of the original program c_{elapse}. Indeed, in the language of NSA, the sequence (1.1) represents a *hyperreal number* r that is infinitesimally close to 1.

We note that While$^{\mathtt{dt}}$ is a modeling language and While$^{\mathtt{dt}}$ programs are *not* intended to be executed: the program c_{elapse} does not terminate. However, it is an advantage of *static* approaches to verification and analysis, that programs need not be executed to prove their correctness. Instead, well-defined mathematical semantics suffices. This is what we do here as well as in [50, 91, 92], with the denotational semantics of While$^{\mathtt{dt}}$ exemplified in Example 1.3.1.

1.3.2 Contribution

In the previous work [50,91,92] of nonstandard static analysis, reachability analysis (or *invariant discovery* in other words) has been a big obstacle in scalability of the proposed verification techniques—as is usual in deductive verification. We tackle this problem in nonstandard static analysis. Technically, we extend *abstract interpretation* [28] with infinitesimals. The abstract interpretation methodology is known for its ample applicability (it is employed in model checking as well as in many deductive verification frameworks) and scalability (the static analyzer Astrée [30] has been successfully used e.g. for Airbus's flight control system).

We establish the theory of *nonstandard abstract interpretation* where (standard) abstract domains are "$*$-transformed," in a rigorous NSA sense, to the abstract domains for hyperreals. It includes their soundness in overapproximating the semantics of While$^{\mathtt{dt}}$ programs (hence reachability of hybrid systems modeled in them). We also introduce the notion of *uniform* widening operators. Using uniform widening operators, the inductive approximation is guaranteed to terminate within finitely many steps even after extension to the nonstandard setting. We show that many known widening operators, if not all, are indeed uniform. Although we focus on the domain of convex polyhedra in this thesis, it is also possible to extend other abstract domains like ellipsoids [35] in the same way.

These theoretical results form a basis for our prototype implementation,[1] that successfully analyzes: *linear water tank*, a common example of piecewise-linear

[1]The prototype is available online: http://group-mmm.org/~kkido/

hybrid dynamics; and also *nonlinear water tank*. The prototype deals with the constant `dt` as a truly infinitesimal number using computer algebra system.

1.4 Two-Step Analysis of Switched Systems with Delays

In this section, we discuss the overall two-step reachability analysis workflow we propose for switched systems with delays. In Chapter 6, we illustrate the two-step reachability analysis using the example of nonlinear water tank. It is analyzed by combining the results of Chapter 4 and Chapter 5. Also for the example of the boost DC-DC converter, a safety controller is synthesized by combining the result of Chapter 4 and the results in [40, 43].

1.4.1 Background

Our goal is the reachability analysis of a hybrid system with delays. We assume that the controller of the hybrid system is state-dependent. In Chapter 3 and Chapter 4, we construct approximate bisimulation relations that can be used to reduce the reachability of the delayed system to that of the delay-free model. However, in these results, we do not consider the existence of a controller and just assume that the same mode is always enabled for both the delayed system and its delay-free model. This is not the case if the system with delays and its delay-free model are controlled by the same state-dependent controller. Following the results in [40], we need to consider different controllers for the delayed system and the delay-free model, so that the same mode is always enabled.

Here we discuss the results in [40], which aim at controller synthesis based on approximate bisimulation. Assume that we have a complicated transition system and its ε-approximately bisimilar simple transition system. The goal is to synthesize a safety controller that keeps the trajectory of the complicated transition system within a designated safety region. Its workflow for this purpose is as follows: first, we define a safety region for the simpler transition system, by contracting the safety region for the complicated model by ε; then, for that shrunk safety region, a safety controller for the simpler model is synthesized; finally, using the approximate bisimulation relation, we can construct a safety controller for the original complicated model, from the safety controller obtained for the simpler model with respect to the contracted safety region.

This methodology has been applied to the example of delay-free boost DC-DC converter in [40]. The approximately bisimilar simpler model they consider is the symbolic model with discretized state space obtained in [43]. For this model, its safety controller can be constructed using supervisory control in [81].

1.4.2 Contribution

Our goal is two-step reachability analysis of switched systems with delays. A distinctive feature of our proposed framework from existing work for reachability analysis of nonlinear control systems with delays is that we abstract away the effect of delays in the first step.

In the first step of the two-step analysis, using the methodology introduced in Chapter 4, we compute an upper bound of the Skorokhod distance between the trajectory of the switched system with delays and the one without delays, as

Thm. 4.2.4 and Thm. 4.2.6 ensures. We can also use the methodology introduced in Chapter 3 in the same way, but we use the one in Chapter 4 for better precision. Using the resulting upper bound, the reachability of the switched system with delays reduces to the reachability of the delay-free model.

Then, in the second step, we need to analyze the reachability of the switched system without delays. Since we have separated the first step and the second step, we can choose any existing reachability analysis technique for hybrid systems that is suitable for the target system. One possibility is the methodology introduced in Chapter 5. In §6.2, we use the nonlinear water tank as an example to show the applicability of our method to nonlinear dynamics. In §6.3, we use the boost DC-DC converter as an example. For this example, we do not use the methodology in Chapter 5. Instead, since the dynamics of this dynamics is linear, we can even synthesize a safety controller that keeps the trajectory within a safe region, using the state-space discretization method introduced in [40].

The soundness of the overall two-step analysis for state-dependent controller is guaranteed because we combine two sound steps using the shrunk safe region as we explained in the previous background section.

1.5 Thesis Organization

In Chapter 2, we will recall necessary definitions and results that will be used later in Chapter 3–Chapter 5. In Chapter 3, a methodology for calculation of an error bound between a switched system with time-delays and the ideal model without time-delays is introduced. In Chapter 4, the methodology introduced in the previous chapter is extended to enable more accurate analysis by finding an upper bound of the Skorokhod distance between trajectories. In Chapter 5, a methodology to overapproximate the reachability of a hybrid system is introduced. In Chapter 6, by combining 1) the error analysis between a switched system with delays and its delay-free model, and 2) the reachability analysis of the delay-free model, two examples of switched systems with delays are analyzed. In Chapter 7, we discuss related work. In Chapter 8, we conclude this thesis, including discussion on the future direction of research.

Chapter 2

Preliminaries

In this chapter, we review a minimal set of definitions and results that are necessary for our main results in Chapter 3–Chapter 5. In §2.1, we recall switched systems and its incremental stability. It also introduces Lyapunov functions to witness the incremental stability. We also review the notion of approximate bisimulation on transition systems. The contents of this section will be used in Chapter 3 and Chapter 4. Then in §2.2, we review the basic definitions and results in NSA and abstract interpretation. It also includes the basic definitions and results of domain theory transferred by NSA. For abstract interpretation, we also recall a specific abstract domain of convex polyhedra that will be used in Chapter 5. The contents in §2.2 will only be used in Chapter 5, and the readers can skip this section until it is needed.

Notations The set of real numbers, nonnegative real numbers and natural numbers are denoted by $\mathbb{R}$, $\mathbb{R}^+$ and $\mathbb{N}$ respectively. The set of boolean values is denoted by $\mathbb{B} = \{\mathrm{tt}, \mathrm{ff}\}$ We let $\|\_\|$ denote the usual Euclidean norm on $\mathbb{R}^n$. Given a set X, $\mathcal{P}(X)$ denotes the powerset of X. The basic notions in real analysis such as smoothness, Lipschitz continuity and so on can be found in [90], for example. In this thesis, for a set X, a function $d : X \times X \to \mathbb{R}^+ \cup \{\infty\}$ is called a *premetric* on X.

2.1 Preliminaries for Approximate Bisimulation for Switching Delays

2.1.1 Switched Systems

We first review switched systems. It is one of the common modeling methodologies for hybrid systems used in control theory, and more abstract model than a hybrid automaton. In the following definition, a function is said to be *non-Zeno* if it has only finitely many discontinuities, or has infinitely many discontinuities at $t_1 < t_2 < \cdots < t_k < \cdots$, and satisfies $lim_{k\to\infty} t_k = \infty$.

Definition 2.1.1 (switched system)**.** A *switched system* is a quadruple $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ that consists of:

- A *state space* $\mathbb{R}^n$;
- A finite set $P = \{1, 2, \ldots, m\}$ of *modes*;

- A set of *switching signals* $\mathbf{P} \subseteq \mathcal{S}(\mathbb{R}^+, P)$, where $\mathcal{S}(\mathbb{R}^+, P)$ is the set of functions from $\mathbb{R}^+$ to P that satisfy the following conditions: 1) piecewise constant, 2) continuous from the right, and 3) non-Zeno;
- The set of vector fields $F = \{f_1, f_2, \ldots, f_m\}$ indexed by $p \in P$, where each f_p is a locally Lipschitz continuous function from $\mathbb{R}^n$ to $\mathbb{R}^n$.

Given a switched system model, its trajectory is defined as follows.

Definition 2.1.2 (trajectory)**.** A continuous and piecewise $\mathcal{C}^1$ function $\mathbf{x} : \mathbb{R}^+ \to \mathbb{R}^n$ is called a *trajectory* of the switched system Σ if there exists a switching signal $\mathbf{p} \in \mathbf{P}$ such that

$$\dot{\mathbf{x}}(t) = f_{\mathbf{p}(t)}(\mathbf{x}(t))$$

holds at each time $t \in \mathbb{R}^+$ when the switching signal $\mathbf{p}$ is continuous.

We let $\mathbf{x}(t, x, \mathbf{p})$ denote the point reached at time $t \in \mathbb{R}^+$, starting from the state $x \in \mathbb{R}^n$ (at $t = 0$), under the switching signal $\mathbf{p} \in \mathbf{P}$. In the special case where the switching signal is constant (i.e. $\mathbf{p}(s) = p$ for all $s \in \mathbb{R}^+$), the point reached at time $t \in \mathbb{R}^+$ starting from $x \in \mathbb{R}^n$ is denoted by $\mathbf{x}(t, x, p)$. The *continuous subsystem* of Σ with the constant switching signal $\mathbf{p}(s) = p$ for all $s \in \mathbb{R}^+$ is denoted by Σ_p. If P is a singleton $P = \{1\}$, the system $\Sigma = \Sigma_1$ is a continuous system without switching.

2.1.2 Incremental Stability

After the pioneering work [77], a number of frameworks rely on the assumption of *incremental stability* for the construction of approximate bisimulations. Our results in Chapter 3 and Chapter 4 are also based on this assumption. Intuitively, a dynamical system is incrementally stable if, under any choice of an initial state, the resulting trajectory asymptotically converges to one reference trajectory. In this section, we review the notion of incremental stability. We also recall the result that it is witnessed by the existence of a variant of Lyapunov function.

In the subsequent definitions we will be using the following classes of functions. A continuous function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ is a *class $\mathcal{K}$ function* if it is strictly increasing and $\gamma(0) = 0$. A $\mathcal{K}$ function is a *$\mathcal{K}_\infty$ function* if $\gamma(x) \to \infty$ when $x \to \infty$. A continuous function $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$ is a *class $\mathcal{KL}$ function* if 1) the function defined by $x \mapsto \beta(x, t)$ is a $\mathcal{K}_\infty$ function for any fixed t; and 2) for any fixed x, the function defined by $t \mapsto \beta(x, t)$ is strictly decreasing, and $\beta(x, t) \to 0$ when $t \to \infty$.

Definition 2.1.3 (δ-GAS system [8])**.** Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system. For each mode $p \in P$, the continuous subsystem Σ_p is *incrementally globally asymptotically stable (δ-GAS)* if there exists a $\mathcal{KL}$ function β such that

$$\|\mathbf{x}(t, x, p) - \mathbf{x}(t, y, p)\| \leq \beta(\|x - y\|, t)$$

for all $x, y \in \mathbb{R}^n$ and $t \in \mathbb{R}^+$.

The notion of δ-GAS is a well-known one among various notions of incremental stability. Directly establishing that a given system is δ-GAS is often hard. A usual technique in the field is to let a Lyapunov-type function play the role of witness for δ-GAS [8].

Definition 2.1.4. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system, and for each mode $p \in P$, Σ_p be the continuous subsystem of Σ. A smooth function $V : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is a *δ-GAS Lyapunov function for Σ_p* if there exist $\mathcal{K}_\infty$ functions $\underline{\alpha}$, $\overline{\alpha}$ and $\kappa > 0$ such that the following hold for all $x, y \in \mathbb{R}^n$.

$$\underline{\alpha}(\|x - y\|) \leq V(x, y) \leq \overline{\alpha}(\|x - y\|) \tag{2.1}$$

$$\frac{\partial V}{\partial x}(x, y) f_p(x) + \frac{\partial V}{\partial y}(x, y) f_p(y) \leq -\kappa V(x, y) \tag{2.2}$$

Note that the left-hand side of (2.2) is much like the Lie derivative of V along the vector field f_p.

Theorem 2.1.5 (Rem. 2.4 in [8])**.** *Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system. For each mode $p \in P$, the continuous subsystem Σ_p is δ-GAS if and only if it has a δ-GAS Lyapunov function.* □

The notions so far are for continuous systems without switching. Their extensions to switched systems are introduced in [43].

Definition 2.1.6. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system. Σ is said to be *incrementally globally uniformly asymptotically stable (δ-GUAS)* if there exists a $\mathcal{KL}$ function β such that the following holds for all $x, y \in \mathbb{R}^n$, $t \in \mathbb{R}^+$ and $\mathbf{p} \in \mathbf{P}$.

$$\|\mathbf{x}(t, x, \mathbf{p}) - \mathbf{x}(t, y, \mathbf{p})\| \leq \beta(\|x - y\|, t)$$

This incremental stability (δ-GUAS) has been used in [43] as the main assumption to establish an approximate bisimulation relation for symbolic abstraction of nonlinear switched systems. We will also use δ-GUAS as the main assumption to establish an approximate bisimulation relation for delay abstraction of nonlinear switched systems in Chapter 3 and Chapter 4.

As was the case with δ-GAS, δ-GUAS can be witnessed by the existence of a variant of Lyapunov function. A sufficient condition for a switched system to be δ-GUAS is the existence of a common δ-GAS Lyapunov function.

Definition 2.1.7. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system. A smooth function $V \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ is called a *common δ-GAS Lyapunov function for Σ* if there exist $\mathcal{K}_\infty$ functions $\underline{\alpha}$, $\overline{\alpha}$ and $\kappa > 0$ that make the following hold for all $x, y \in \mathbb{R}^n$.

$$\underline{\alpha}(\|x - y\|) \leq V(x, y) \leq \overline{\alpha}(\|x - y\|) \tag{2.3}$$

$$\frac{\partial V}{\partial x}(x, y) f_p(x) + \frac{\partial V}{\partial y}(x, y) f_p(y) \leq -\kappa V(x, y) \quad \text{for all } p \in P \tag{2.4}$$

Theorem 2.1.8 (Thm. 2 in [43])**.** *Let Σ be a switched system. If there exists a common δ-GAS Lyapunov function V of Σ, then Σ is δ-GUAS.* □

Another sufficient condition is the existence of *multiple δ-GAS Lyapunov functions*, under an additional assumption on the set of switching signals [43]. The use of multiple Lyapunov functions for hybrid and switched systems was first advocated in [20]. We let $\mathcal{S}_{\tau_d}(\mathbb{R}^+, P) \subseteq \mathcal{S}(\mathbb{R}^+, P)$ denote the set of switching signals with a *dwell time* $\tau_d > 0$, which means that the intervals between switching times are always longer than τ_d.

We introduce the following notations. Given a switched system $\Sigma = (\mathbb{R}^n, P = \{1, 2, \cdots, p, \ldots, m\}, \mathbf{P}, F)$, recall that Σ_p denotes the switching-free subsystem where the mode is fixed to p (see §2.1). Assume that, for each $p \in \{1, 2, \ldots, m\}$, we have a δ-GAS Lyapunov function V_p for the subsystem Σ_p. Then there exist a constant $\kappa_p \in \mathbb{R}^+$ and two $\mathcal{K}_\infty$ functions $\underline{\alpha}_p$ and $\overline{\alpha}_p$ as in Def. 2.1.3. Let us now define

$$\begin{array}{l} \underline{\alpha} := \min(\underline{\alpha}_1, \ldots, \underline{\alpha}_m) \ , \quad \overline{\alpha} := \max(\overline{\alpha}_1, \ldots, \overline{\alpha}_m) \ , \\ \kappa := \min(\kappa_1, \ldots, \kappa_m) \ , \end{array} \tag{2.5}$$

where $\min(f_1, \ldots, f_k)$ and $\max(f_1, \ldots, f_k)$ are defined by $\min(f_1, \ldots, f_k)(x) = \min(f_1(x), \ldots, f_k(x))$ and $\max(f_1, \ldots, f_k)(x) = \max(f_1(x), \ldots, f_k(x))$, respectively.

Theorem 2.1.9 (Thm. 3 in [43])**.** *Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system. Assume that $P = \{1, 2, \ldots, m\}$, and that its set $\mathbf{P}$ of switching signals satisfies $\mathbf{P} \subseteq \mathcal{S}_{\tau_d}(\mathbb{R}^+, P)$. Assume further that, for each $p \in P$, there exists a δ-GAS Lyapunov function V_p for the subsystem Σ_p. We also assume that there exists $\mu \in \mathbb{R}^+$ such that*

$$V_p(x, y) \leq \mu V_{p'}(x, y) \quad \textit{for all } x, y \in \mathbb{R}^n \textit{ and } p, p' \in P.$$

If the dwell time τ_d satisfies $\tau_d > \frac{\log \mu}{\kappa}$, then Σ_{τ_d} is δ-GUAS. □

Remark 2.1.10. Incremental stability notion such as δ-GAS and δ-GUAS requires that any pair of trajectories converges to each other. It is known that for linear systems, δ-GAS is equivalent to usual global asymptotic stability that requires the convergence to the equilibrium (see e.g. [97]). However, for nonlinear systems, it is stronger than usual asymptotic stability with respect to the equilibrium. In the results in this thesis, we will assume that Lyapunov functions that witness δ-GUAS are given. Therefore, the applicability of our results depends on how strong this incremental stability assumption is. Proving that a nonlinear system is incrementally stable is a challenging task and it is studied in control theory (e.g. [58]). Combining our results with such work is imminent future work.

2.1.3 Approximate Bisimulation

Our main results in Chapter 3 and Chapter 4 to obtain an error bound between a switched system with delays and its delay-free model are given by establishing an approximate bisimulation relation. In this section, we review the notion of approximate bisimulation [41, 42], a (co)inductive construct that guarantees henceforth proximity of behaviors of two states. It is a relation between the states of two transition systems.

Definition 2.1.11 (transition system)**.** A *transition system* is a sextuple $T = (Q, L, \longrightarrow, O, H, I)$, where

- Q is a set of states;
- L is a set of labels;

- $\longrightarrow \subseteq Q \times L \times Q$ is a transition relation;
- O is a set of outputs;
- $H : Q \to O$ is an output function; and
- $I \subseteq Q$ is a set of initial states.

We let $q \xrightarrow{l} q'$ denote the fact that $(q, l, q') \in \longrightarrow$. A transition system T is said to be *premetric* if the set O of outputs is equipped with a premetric d.

An infinite sequence $\big((q_0, l_0), (q_1, l_1), \cdots, (q_i, l_i), \cdots\big)$ in $(Q \times L)^\omega$ is a *state trajectory* of the transition system T if $q_0 \in I$ and $q_i \xrightarrow{l_i} q_{i+1}$ for all $i \in \mathbb{N}$. An *output trajectory* $\big((H(q_0), l_0), (H(q_1), l_1), \cdots, (H(q_i), l_i), \cdots\big)$ in $(O \times L)^\omega$ is associated with a state trajectory $\big((q_0, l_0), (q_1, l_1), \cdots, (q_i, l_i), \cdots\big)$. We also consider state trajectories and output trajectories of finite length $N \in \mathbb{N}$, defined similarly.

Note that in [41,42], they defined approximate bisimulation on *metric* transition systems, but we have weakened this condition to premetric, to fit our setting used in Chapter 3 and Chapter 4.

Then, approximate bisimulation relation is defined between states.

Definition 2.1.12. Let $T_i = (Q_i, L, \underset{i}{\longrightarrow}, O, H_i, I_i)$ $(i = 1, 2)$ be two premetric transition systems with premetric d; note that T_1 and T_2 share the same sets of actions L and outputs O. Let $\varepsilon \in \mathbb{R}^+$ be a nonnegative real number; we call it a *precision*. A relation $R \subseteq Q_1 \times Q_2$ is called an *ε-approximate bisimulation relation* between T_1 and T_2 if the following three conditions hold for all $(q_1, q_2) \in R$.

- $d(H_1(q_1), H_2(q_2)) \leq \varepsilon$;
- For all l and q_1' satisfying $q_1 \xrightarrow[1]{l} q_1'$, there exists q_2' such that $q_2 \xrightarrow[2]{l} q_2'$ and $(q_1', q_2') \in R$ hold; and
- For all l and q_2' satisfying $q_2 \xrightarrow[2]{l} q_2'$, there exists q_1' such that $q_1 \xrightarrow[1]{l} q_1'$ and $(q_1', q_2') \in R$ hold.

The transition systems T_1 and T_2 are *approximately bisimilar with precision ε* if there exists an ε-approximate bisimulation relation R that satisfies the following conditions:

- For all $q_1 \in I_1$, there exists $q_2 \in I_2$ such that $(q_1, q_2) \in R$;
- For all $q_2 \in I_2$, there exists $q_1 \in I_1$ such that $(q_1, q_2) \in R$.

We let $T_1 \sim_\varepsilon T_2$ denote the fact that T_1 and T_2 are approximately bisimilar with precision ε.

An approximate bisimulation relation gives an upper bound of the *language metric* between two transition systems. The details will be found in [42]. The construction of the transition system from a given switched system, and how approximate bisimulation relation is established under the assumption of incremental stability will be discussed later in Chapter 3 and Chapter 4.

2.2 Preliminaries for Abstract Interpretation with Infinitesimals

The definitions and results introduced in this section will be used in Chapter 5 and are not necessary in Chapter 3 and Chapter 4. Readers can skip this section until reaching Chapter 5.

First, some basic notions in NSA are explained in §2.2.1. Then, an extension of domain theory using NSA is presented in §2.2.2. They are based on [92]. Next, the general theory of abstract interpretation and the specific domain of convex polyhedra is introduced in §2.2.3. Finally in §2.2.4, we showcase how standard abstract interpretation on convex polyhedra computes an overapproximation of the reachable set using an example of discretized water tank.

2.2.1 Nonstandard Analysis

In this section we present necessary definitions and results in NSA [84]. Further details of NSA are found e.g. in [44, 57].

The following notions will play important roles:

- *Hyperreals* that extend reals by infinitesimals, infinites, etc.;
- The *transfer principle*, a celebrated result in NSA that states that reals and hyperreals share "the same properties";
- The first-order language $\mathscr{L}_X$ that specifies formulas in which syntax, precisely, are preserved by the transfer principle; and finally
- The semantical construct of *superstructure* for interpreting $\mathscr{L}_X$-formulas.

Among these notions, the transfer principle is particularly important; in order to formulate it in a mathematically rigorous manner, the two last items (the language $\mathscr{L}_X$ on the syntactic side, and superstructures on the semantical side) are used. The first-order language $\mathscr{L}_X$ is essentially that of set theory and has two predicates $=$ and $\in$. The *superstructure* $V(X)$ is then a semantical "universe" for such formulas, constructed from the base set X: concretely $V(X)$ is the union of X, $\mathcal{P}(X)$, $\mathcal{P}(X \cup \mathcal{P}(X))$, and so on. Finally, when we take $X = \mathbb{R}$ then the set ${}^*X = {}^*\mathbb{R}$ is that of *hyperreals*; and the transfer principle claims that A holds for reals if and only if ${}^*A$—a formula essentially the same as A—holds for hyperreals. Its precise statement is:

Lemma 2.2.1 (the transfer principle)**.** *For any closed formula A in $\mathscr{L}_X$, the following are equivalent.*

- *The formula A is valid in the superstructure $V(X)$.*
- *The* *-transform *${}^*A$ of A—this is a formula in the language $\mathscr{L}_{{}^*X}$—is valid in the superstructure $V({}^*X)$.* □

The transfer principle guarantees that we can employ the same abstract interpretation framework, for reals and hyperreals. Concretely, various constructions and meta results (such as soundness and termination) in abstract interpretation will be expressed as $\mathscr{L}_{\mathbb{R}}$-formulas, and since they are valid in $V(\mathbb{R})$, they are valid in the "nonstandard universe" $V({}^*\mathbb{R})$ too, by the transfer principle.

Hyperreals We fix an *index set* $I = \mathbb{N}$ throughout this thesis. A family $\mathcal{F} \subseteq \mathcal{P}(I)$ is a *filter* on I if it is closed under supersets and finite intersections, i.e. 1) if $X \subseteq Y$ and $X \in \mathcal{F}$, then $Y \in \mathcal{F}$; and 2) if $X, Y \in \mathcal{F}$, then $X \cap Y \in \mathcal{F}$. A *proper filter* is a nonempty filter that does not contain the empty set $\emptyset$. A proper filter $\mathcal{F}$ is called an *ultrafilter* if it is maximal, i.e. there is no filter $\mathcal{F}'$ such that $\mathcal{F} \subsetneq \mathcal{F}' \subsetneq \mathcal{P}(I)$. Note that 1) for any $S \subseteq I$, either $S \in \mathcal{F}$ or $I \setminus S \in \mathcal{F}$ (but not both); and 2) if $S \subseteq I$ is *cofinite* (i.e. $I \setminus S$ is finite), then $S \in \mathcal{F}$. We fix an ultrafilter $\mathcal{F}$ throughout this thesis.

Definition 2.2.2 (hyperreal $r \in {}^*\mathbb{R}$)**.** We define the set ${}^*\mathbb{R}$ of *hyperreal numbers* (or *hyperreals*) by ${}^*\mathbb{R} := \mathbb{R}^I/\!\sim_{\mathcal{F}}$. It is therefore the set of infinite sequences on $\mathbb{R}$ modulo the following equivalence $\sim_{\mathcal{F}}$: we have $(a_0, a_1, \dots) \sim_{\mathcal{F}} (a'_0, a'_1, \dots)$ if

$$\{i \in I \mid a_i = a'_i\} \in \mathcal{F} \ , \quad \text{for which we say "} d_i = d'_i \text{ for almost every } i\text{."} \tag{2.6}$$

A *hypernatural* $n \in {}^*\mathbb{N}$ is defined similarly.

Remark 2.2.3 (choice of the index set I)**.** In NSA, it is common to consider an index set I that is larger than $\mathbb{N}$. The advantage of taking such I is beyond the scope of this thesis; see [57, Chap. II] for more details. In this thesis, we will keep using the set $\mathbb{N}$ of natural numbers as the index set I for concreteness.

It follows that: two sequences $(a_i)_i$ and $(a'_i)_i$ that coincide except for finitely many indices i represent the same hyperreal. The predicates besides $=$ (such as $<$) are defined in the same way. A notable consequence is the existence of infinite numbers in the set of hyperreals and hypernaturals: $\omega := [(1, 2, 3, \dots)]$ is a positive infinite since it is larger than any positive real $r = [(r, r, \dots)]$ ($i > r$ for almost every $i \in \mathbb{N}$). In addition, the set of hyperreals includes infinitesimal numbers: a hyperreal $\omega^{-1} := [\,(1, \frac{1}{2}, \frac{1}{3}, \dots)\,]$ is positive ($0 < \omega^{-1}$) but is smaller than any (standard) positive real r.

Superstructure A *superstructure* is a "universe," constructed step by step from a certain base set X (whose typical examples are $\mathbb{R}$ and ${}^*\mathbb{R}$). We assume $\mathbb{N} \subseteq X$.

Definition 2.2.4 (superstructure)**.** A *superstructure* $V(X)$ over X is defined by $V(X) := \bigcup_{n \in \mathbb{N}} V_n(X)$, where $V_0(X) := X$ and $V_{n+1}(X) := V_n(X) \cup \mathcal{P}(V_n(X))$.

The superstructure $V(X)$ might seem to be a closure of X only under powersets, but it accommodates many set-forming operations. For example, ordered pairs (a, b) and tuples $(a_1, \dots, a_m)$ are defined in $V(X)$ as is usually done in set theory, e.g. $(a, b) := \{\{a\}, \{a, b\}\}$.The function space $a \to b$ is thought of as a collection of special binary relations (i.e. $a \to b \subseteq \mathcal{P}(a \times b)$), hence is in $V(X)$.

The First-Order Language $\mathscr{L}_X$ We use the following first-order language $\mathscr{L}_X$, defined for each choice of the base set X like $\mathbb{R}$ and ${}^*\mathbb{R}$.

Definition 2.2.5 (the language $\mathscr{L}_X$)**.** *Terms* in $\mathscr{L}_X$ consist of: variables $x, y, x_1, x_2, \dots$; and a constant a for each entity $a \in V(X)$.

Formulas in $\mathscr{L}_X$ are constructed as follows:

- The predicate symbols are $=$ and $\in$; both are binary. The *atomic formulas* are of the form $s = t$ or $s \in t$ (where s and t are terms).
- We allow Boolean combinations of formulas. We use the symbols $\wedge, \vee, \neg$ and $\Rightarrow$.
- Given a formula A, a variable x and a term s, the expressions $\forall x \in s.\, A$ and $\exists x \in s.\, A$ are formulas.

Note that quantifiers always come with a bound s. The language $\mathscr{L}_X$ depends on the choice of X (it determines the set of constants). We shall also use the following syntax sugars in $\mathscr{L}_X$, as is common in set theory and NSA.

(s,t)	pair	$(s_1,\dots,s_m)$	tuple	$s \times t$	direct product
$s \subseteq t$	inclusion, short for $\forall x \in s.\, x \in t$				
$s(t)$	function application; short for x such that $(t,x) \in s$				
$s \circ t$	function composition, $(s \circ t)(x) = s(t(x))$				
$s \leq t$	inequality in $\mathbb{N}$; short for $(s,t) \in \leq$ where $\leq \subseteq \mathbb{N}^2$				

Definition 2.2.6 (semantics of $\mathscr{L}_X$)**.** We interpret $\mathscr{L}_X$ in the superstructure $V(X)$ in the obvious way. Let A be a closed formula; we say A is *valid* if A is true in $V(X)$.

The $*$-Transform and the Transfer Principle As we mentioned, the transfer principle says that a closed formula A in the language $\mathscr{L}_X$ is valid in $V(X)$ if and only if ${}^*A$ in $\mathscr{L}_{{}^*X}$ is valid in $V({}^*X)$. We shall describe how we syntactically transform A in $\mathscr{L}_X$ into ${}^*A$ in $\mathscr{L}_{{}^*X}$.

For that purpose, in particular in translating constants in $\mathscr{L}_X$ (for entities in $V(X)$) to $\mathscr{L}_{{}^*X}$, we will need the *semantical* translation

$$ {}^*(\_) \;:\; V(X) \longrightarrow V({}^*X) \;, \qquad a \longmapsto {}^*a \tag{2.7} $$

that is called the **-transform*. It is a map from the universe $V(X)$ of standard entities to $V({}^*X)$ of nonstandard entities. This mapping factorizes into the following three steps.

$$ \begin{array}{ccc} V(X) & \xrightarrow{\quad {}^*(\_) \quad} & V({}^*X) \\ {\scriptstyle \overline{(\_)}}\downarrow & & \uparrow{\scriptstyle M} \\ \bigcup_{n\in\mathbb{N}} \big(V_n(X) \setminus V_{n-1}(X)\big)^I & \xrightarrow[\;[\_]\;]{} & \prod^0_{\mathcal{F}} V(X) \end{array} \tag{2.8} $$

The first factor $\overline{(\_)}$ maps $a \in V(X)$ to the constant function $\overline{a}$ such that $\overline{a}(i) = a$ for each $i \in I$; recall that we have chosen $I = \mathbb{N}$ (Rem. 2.2.3). The second $[\_]$ takes a quotient modulo the ultrafilter $\mathcal{F}$; finally the third factor M is the so-called *Mostowski collapse*. The details of this construction are beyond our scope; they are found in [57, §II.4].

For an intuition let us exhibit these maps in the simple setting with $a \in X$. The first factor $\overline{(\_)}$ corresponds to forming constant streams: $a \mapsto \overline{a} = (a, a, \dots)$. The second $[\_]$ is quotienting modulo $\sim_{\mathcal{F}}$ of (2.6). The third map M does nothing—it is a book-keeping function that is only needed in the extended setting of superstructures.

The above map ${}^*(\_)\colon V(X) \to V({}^*X)$ becomes a *monomorphism*, a notion in NSA. Most notably it will satisfy the *transfer principle* (Lem. 2.2.8).

Definition 2.2.7 (*-transform of formulas)**.** Let A be a formula in $\mathscr{L}_X$. The **-transform* of A, denoted by ${}^*A$, is a formula in $\mathscr{L}_{{}^*X}$ obtained by replacing each constant a occurring in A with the constant ${}^*a$ that designates the element ${}^*a \in V({}^*X)$.

Lemma 2.2.8 (the transfer principle)**.** *For any closed formula A in $\mathscr{L}_X$, A is valid (in $V(X)$) if and only if ${}^*A$ is valid (in $V({}^*X)$).* □

The transfer principle is a powerful result and we rely on it in §2.2.2 and Chapter 5. Here are the first examples of its use; they are proved by transferring a suitable formula A.

Lemma 2.2.9. *1. For $a \in V(X) \setminus X$ we obtain an injective map*

$$ {}^*(\_) \ : \ a \longrightarrow {}^*a \ , \quad (b \in a) \longmapsto ({}^*b \in {}^*a) \tag{2.9} $$

as a restriction of ${}^*(\_)$ in (2.7).

2. If a is a finite set, the map (2.9) is an isomorphism $a \xrightarrow{\cong} {}^*a$.

3. Let $a \to b$ be the set of functions from a to b. We have ${}^*(a \to b) \subseteq {}^*a \to {}^*b$.

4. ${}^*(a_1 \times \cdots \times a_m) = {}^*a_1 \times \cdots \times {}^*a_m$; and ${}^*(a_1 \cup \cdots \cup a_m) = {}^*a_1 \cup \cdots \cup {}^*a_m$.

5. For a binary relation $r \subseteq a \times a$, we have ${}^*r \subseteq {}^*a \times {}^*a$. Moreover, r is an order if and only if ${}^*r$ is an order. □

Internal Sets In §2.2.2, especially Rem. 2.2.19, we will see that being *internal* is crucial for transfer. The difference between internal and *external* entities is important in NSA, but we present only the necessary definition and lemma here. For more detailed discussion, see [57, §II.6].

Definition 2.2.10 (internal entity)**.** An element $b \in V({}^*X)$ is *internal* with respect to ${}^*(\_) : V(X) \to V({}^*X)$ if there is $a \in V(X)$ such that $b \in {}^*a$. It is *external* if it is not internal.

Lemma 2.2.11. *A function $f : {}^*a \to {}^*b$ is internal if and only if $f \in {}^*(a \to b)$.* □

2.2.2 Domain Theory, Transferred

The collecting semantics of While$^{\mathsf{dt}}$ is introduced by solving recursive equations on ${}^*\mathcal{P}(\mathbb{R}^n)$. Here we present necessary theoretical foundations for that. We identify the set ${}^*\mathcal{P}(\mathbb{R}^n)$ as a hyperdomain and *-transferring domain theory.

The current section is an adaptation of what appeared in the appendix of [92]; and the definitions and results are similar to those in [15, §2.2]. In [15], a hyperdomain is called an *internal domain*, and a *-continuous function is called an *internal continuous function*.

Definition 2.2.12. In what follows we employ the theory of NSA presented in §2.2.1. As the base set of a superstructure $V(X)$ (Def. 2.2.4), we take $X = \mathbb{R} \cup \mathbb{B} \cup \mathbf{Var}$.

Definition 2.2.13 (hyperdomain)**.** Let $(D, \sqsubseteq)$ be a cpo. The pair $({}^*D, {}^*\sqsubseteq)$ of its *-transforms is called a *hyperdomain*.

Example 2.2.14. The set $\mathcal{P}(\mathbf{Var} \to \mathbb{R})$ is a complete lattice with respect to the inclusion order $\subseteq$, therefore is a cpo. Its $*$-transform $\big({}^*\big(\mathcal{P}(\mathbf{Var} \to \mathbb{R})\big), {}^*\subseteq\big)$ constitutes a hyperdomain.

We note that the set ${}^*\big(\mathcal{P}(\mathbf{Var} \to \mathbb{R})\big)$ coincides with the set of internal subsets of the space $\{f \colon {}^*\mathbf{Var} \to {}^*\mathbb{R} \mid f \text{ is an internal function}\}$. Moreover, under the assumption that **Var** is a finite set (e.g. the set of variables occurring in a program c), we can see that the last set $\{f \colon {}^*\mathbf{Var} \to {}^*\mathbb{R} \mid f \text{ is an internal function}\}$ coincides with the function space $\mathbf{Var} \to {}^*\mathbb{R}$. For this we use Lem. 2.2.9.4.

Note that for a hyperdomain $({}^*D, {}^*\sqsubseteq)$, ${}^*\sqsubseteq$ is an order in ${}^*D$ (Lem. 2.2.9.5). We will establish a fixed point theorem on a hyperdomain. First, we encode the definitions of cpo and continuous functions as $\mathscr{L}_X$-formulas to be used in the proofs.

Definition 2.2.15. We define the following $\mathscr{L}_X$-formulas:

$$
\begin{aligned}
&\mathsf{BinRel}_{a,r} :\equiv\ r \subseteq a \times a\\
&\mathsf{Refl}_{a,r} :\equiv\ \forall x \in a.\, (x,x) \in r\\
&\mathsf{Trans}_{a,r} :\equiv\ \forall x,y,z \in a.\, \big((x,y) \in r \wedge (y,z) \in r \Rightarrow (x,z) \in r \big)\\
&\mathsf{AntiSym}_{a,r} :\equiv\ \forall x,y \in a.\, \big((x,y) \in r \wedge (y,x) \in r \Rightarrow x = y \big)\\
&\mathsf{Preord}_{a,r} :\equiv\ \mathsf{BinRel}_{a,r} \wedge \mathsf{Refl}_{a,r} \wedge \mathsf{Trans}_{a,r}\\
&\mathsf{Poset}_{a,r} :\equiv\ \mathsf{Preord}_{a,r} \wedge \mathsf{AntiSym}_{a,r}\\
&\mathsf{HasBot}_{a,r} :\equiv\ \exists x \in a. \forall y \in a.\, (x,y) \in r\\
&\mathsf{AscCn}_{a,r}(s) :\equiv\ \forall x, x' \in \mathbb{N}.\, (x \le x' \Rightarrow (s(x), s(x')) \in r)\\
&\mathsf{UpBd}_{a,r}(b,s) :\equiv\ \forall x \in \mathbb{N}.\, ((s(x), b) \in r)\\
&\mathsf{Sup}_{a,r}(p,s) :\equiv\ \mathsf{UpBd}_{a,r}(p,s) \wedge \forall b \in a.\, (\mathsf{UpBd}_{a,r}(b,s) \Rightarrow (p,b) \in r)\\
&\mathsf{CPO}_{a,r} :\equiv\ \mathsf{Poset}_{a,r} \wedge \mathsf{HasBot}_{a,r}\\
&\qquad\qquad \wedge \forall s \in (\mathbb{N} \to a).\, \big(\mathsf{AscCn}_{a,r}(s) \Rightarrow \exists p \in a.\, \mathsf{Sup}_{a,r}(p,s) \big)\\
&\mathsf{Conti}_{a_1,r_1,a_2,r_2}(f) :\equiv\ \forall s \in (\mathbb{N} \to a_1). \forall p \in a_1.\\
&\qquad\qquad \Big(\big(\mathsf{AscCn}_{a_1,r_1}(s) \wedge \mathsf{Sup}_{a_1,r_1}(p,s) \big) \Rightarrow \mathsf{Sup}_{a_2,r_2}(f(p), f \circ s) \Big)\ .
\end{aligned}
$$

Definition 2.2.16 (*-continuous function)**.** Let $({}^*D_1, {}^*\sqsubseteq_1)$ and $({}^*D_2, {}^*\sqsubseteq_2)$ be hyperdomains. A function $f : {}^*D_1 \to {}^*D_2$ is *$*$-continuous* if it is internal and satisfies the *-transform of the formula $\mathsf{Conti}_{D_1,\sqsubseteq_1,D_2,\sqsubseteq_2}$. That is to be precise: ${}^*(\mathsf{Conti}_{D_1,\sqsubseteq_1,D_2,\sqsubseteq_2})(f)$ is valid. The set of *-continuous functions from ${}^*D_1$ to ${}^*D_2$ is denoted by ${}^*D_1 \to_{{}^*\mathrm{ct}} {}^*D_2$.

Note that in the condition ${}^*(\mathsf{Conti}_{D_1,\sqsubseteq_1,D_2,\sqsubseteq_2})(f)$, the range of a chain s is the set of internal functions ${}^*(\mathbb{N} \to D_1)$.

Lemma 2.2.17. $({}^*D_1 \to_{{}^*\mathrm{ct}} {}^*D_2) = {}^*(D_1 \to_{\mathrm{ct}} D_2)$. *Here* $\to_{\mathrm{ct}}$ *denotes the set of continuous functions.*

Proof. Assume $f \in {}^*(D_1 \to_{\mathrm{ct}} D_2)$. The following closed formula is valid in $V(X)$:

$$\forall f' \in (D_1 \to D_2).\, \big(f' \in (D_1 \to_{\mathrm{ct}} D_2) \Leftrightarrow \mathsf{Conti}(f') \big)\ ,$$

where Conti is short for $\mathsf{Conti}_{D_1,\sqsubseteq_1,D_2,\sqsubseteq_2}$. By transfer we have

$$\forall f' \in {}^*(D_1 \to D_2).\, \big(f' \in {}^*(D_1 \to_{\mathrm{ct}} D_2) \Leftrightarrow {}^*\mathsf{Conti}(f')\big) \tag{2.10}$$

valid in $V({}^*X)$. Thus f satisfies ${}^*\mathsf{Conti}(f')$. Obviously f is internal; therefore $f \in ({}^*D_1 \to_{{}^*\mathrm{ct}} {}^*D_2)$.

Conversely, assume $f \in ({}^*D_1 \to_{{}^*\mathrm{ct}} {}^*D_2)$. By the definition of *-continuity, f is internal, hence by Lem. 2.2.11 we have $f \in {}^*(D_1 \to D_2)$. Moreover, using the definition of *-continuity and (2.10), we have $f \in {}^*(D_1 \to_{\mathrm{ct}} D_2)$. □

The following lemma enables us to define a collecting semantics of $\textsc{While}^{\mathsf{dt}}$ programs as a least fixed point later in §5.1.

Lemma 2.2.18. *Let $({}^*D, {}^*\sqsubseteq)$ be a hyperdomain. Then a *-continuous function $f : {}^*D \to {}^*D$ has a least fixed point. Moreover, the function ${}^*\mathrm{lfp}$ that maps f to its least fixed point $({}^*\mathrm{lfp})(f)$ is *-continuous.*

Proof. By the usual construction in a cpo, we obtain the map

$$\mathrm{lfp} \;:\; (D \to_{\mathrm{ct}} D) \to_{\mathrm{ct}} D \;\;,\quad f \mapsto \textstyle\bigsqcup_{n\in\mathbb{N}} f^n(\bot) \;\;.$$

Continuity of lfp is easy and standard. As its *-transform we obtain a function ${}^*\mathrm{lfp} : ({}^*D \to_{{}^*\mathrm{ct}} {}^*D) \to_{{}^*\mathrm{ct}} {}^*D$, where we used Lem. 2.2.17 and Lem. 2.2.9. The fact that ${}^*\mathrm{lfp}$ returns least fixed points is shown by the transfer of the following $\mathscr{L}_X$-formula.

$$\forall f \in (D \to_{\mathrm{ct}} D).\, \big(\; f(\mathrm{lfp}(f)) = \mathrm{lfp}(f) \;\wedge \forall x \in D.\, (f(x) = x \;\Rightarrow \mathrm{lfp}(f) \sqsubseteq x)\;\big)$$

□

Remark 2.2.19. In the proofs of the previous lemmas, it is necessary that $f : {}^*D \to {}^*D$ is an internal function. The transfer principle Lem. 2.2.8 can be applied only to a closed formula in $\mathscr{L}_X$; and $\mathscr{L}_X$ only allows bounded quantifiers ($\forall x \in s$ with some bound s). When f is internal, we have such a bound $f \in {}^*(D \to D)$, but there are no such bounds for external f.

2.2.3 Theory of Abstract Interpretation

Abstract interpretation [31] is a well-established technique in static analysis. We make a brief review of its basic theory. The goal of abstract interpretation is to overapproximate a *concrete semantics* defined on an *concrete domain* by an *abstract semantics* on an *abstract domain*. We also review the specific abstract domain of convex polyhedra in this section.

Concretization-Based Abstract Interpretation Framework First we recall the concretization-based framework described in [29]. Note that we are not using the formalization in [31], which is based on Galois connection.

We assume that the concrete semantics is defined as a least fixed point on the concrete domain. The following proposition guarantees the overapproximation of the least fixed point in the concrete domain by a prefixed point in the abstract domain. In the proposition, the order $\sqsubseteq$ on the domain L is extended to the order on $L \to L$ pointwisely. And the *least fixed point relative to* $\bot$, denoted by $\mathrm{lfp}_{\bot} F$, is the least among the fixed points of F above $\bot$; by the cpo structure of L and the continuity of F, it is given by $\bigsqcup_{n\in\mathbb{N}} F^n(\bot)$.

Proposition 2.2.20. *Let $(L, \sqsubseteq)$ be a cpo; $F : L \to L$ be a continuous function; and $\bot \in L$ be such that $\bot \sqsubseteq F(\bot)$. Let $(\overline{L}, \overline{\sqsubseteq})$ be a preorder; $\gamma : \overline{L} \to L$ be a function (it is called* concretization*) such that $\overline{a} \mathrel{\overline{\sqsubseteq}} \overline{b} \Rightarrow \gamma(\overline{a}) \sqsubseteq \gamma(\overline{b})$ for all $\overline{a}, \overline{b} \in \overline{L}$; and $\overline{F} : \overline{L} \to \overline{L}$ be a monotone function such that $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$. Assume further that $\overline{x} \in \overline{L}$ is a prefixed point of $\overline{F}$ (i.e. $\overline{F}(\overline{x}) \mathrel{\overline{\sqsubseteq}} \overline{x}$) such that $\bot \sqsubseteq \gamma(\overline{x})$.*

Then $\overline{x}$ overapproximates $\mathrm{lfp}_{\bot} F$, that is, $\mathrm{lfp}_{\bot} F \sqsubseteq \gamma(\overline{x})$. □

Later in §2.2.4 where we analyze the discretized linear water tank, the set $\mathcal{P}(\mathbb{R}^n)$ of subsets of memory states is used as a concrete domain L; and the *domain of convex polyhedra* is used as an abstract domain $\overline{L}$. The interpretations F and $\overline{F}$ on each domains are defined in a standard manner. Towards the goal of obtaining $\overline{x}$ in Prop. 2.2.20, (i.e. finding a prefixed point in the abstract domain), the following notion of *widening* is used (often together with *narrowing* that we will not be using). Note that in the following definition and proposition, the domain $(L, \sqsubseteq)$ is the abstract domain, corresponding to $(\overline{L}, \overline{\sqsubseteq})$ in Prop. 2.2.20.

Definition 2.2.21 (widening operator)**.** Let $(L, \sqsubseteq)$ be a preorder. A function $\nabla : L \times L \to L$ is said to be a *widening operator* if the following two conditions hold.

- (*Covering*) For any $x, y \in L$, $x \sqsubseteq x \nabla y$ and $y \sqsubseteq x \nabla y$.
- (*Termination*) For any ascending chain $\langle x_i \rangle \in L^{\mathbb{N}}$, the chain $\langle y_i \rangle \in L^{\mathbb{N}}$ defined by $y_0 = x_0$ and $y_{i+1} = y_i \nabla x_{i+1}$ for each $i \in \mathbb{N}$ is ultimately stationary.

A widening operator on a fixed abstract domain $\overline{L}$ is not at all unique. In this thesis, we will discuss three widening operators previously introduced for the domain of convex polyhedra $\mathbb{CP}_n$.

The use of widening is as in the following proposition: the covering condition in Def. 2.2.21 ensures that the outcome is a prefixed point; and the procedure terminates thanks to the termination condition in Def. 2.2.21.

Proposition 2.2.22 (convergence of iteration sequences)**.** *Let $(L, \sqsubseteq)$ be a preorder; $F : L \to L$ be a monotone function; $\bot \in L$ be such that $\bot \sqsubseteq F(\bot)$; $\nabla : L \times L \to L$ be a widening operator; and $\langle X_i \rangle_{i \in \mathbb{N}} \in L^{\mathbb{N}}$ be the infinite sequence defined by*

$$X_0 = \bot \ ; \qquad \text{and, for each } i \in \mathbb{N}, \quad X_{i+1} = \begin{cases} X_i & (\text{if } F(X_i) \sqsubseteq X_i) \\ X_i \nabla F(X_i) & (\text{otherwise}) \end{cases}$$

Then the sequence $\langle X_i \rangle_{i \in \mathbb{N}}$ is increasing and ultimately stationary; moreover its limit $\bigsqcup_{i \in \mathbb{N}} X_n$ is a prefixed point of F such that $\bot \sqsubseteq \bigsqcup_{i \in \mathbb{N}} X_n$. □

The Domain of Convex Polyhedra The domain of convex polyhedra, introduced in [31], is one of the most often used relational abstract domain.

Definition 2.2.23 (domain of convex polyhedra $\mathbb{CP}_n$)**.** An n-dimensional *convex polyhedron* is the intersection of finitely many (closed) affine half-spaces. We denote the set of convex polyhedra in $\mathbb{R}^n$ by $\mathbb{CP}_n$. Its preorder $\sqsubseteq$ is given by the inclusion order (actually it is a partial order). The concretization function $\gamma_{\mathbb{CP}_n} : \mathbb{CP}_n \to \mathcal{P}(\mathbb{R}^n)$ is defined in an obvious manner.

A convex polyhedron can be represented in two ways: as a *constraint system* and a *generator system*. A constraint system is a finite set of linear constraints. A constraint system C represents the convex polyhedron that consists of all the points that satisfies all linear constraints in C. A constraint system is sometimes implicitly required to be in so-called *minimal form*. Its details are in [31], and we skip them in this thesis. The function con maps a constraint system to the corresponding convex polyhedron. The linear constraints include both linear equations and linear inequalities. The function eq maps a constraint system C to the set of linear equations in C. The function repr maps a constraint system to the set of linear inequalities splitting each linear equation into two linear inequalities.

A generator system is a triple of three finite sets of vectors (L, R, P), where L is the set of *lines*, R is the set of *rays* and P is the set of *points*. The generator system $(L = \{\vec{l_1}, \cdots, \vec{l_l}\}, R = \{\vec{r_1}, \cdots, \vec{r_r}\}, P = \{\vec{p_1}, \cdots, \vec{p_p}\})$ represents the convex polyhedron

$$\left\{ \sum_{i=1}^{l} \lambda_i \vec{l_i} + \sum_{i=1}^{r} \rho_i \vec{r_i} + \sum_{i=1}^{p} \pi_i \vec{p_i} \middle| \lambda_i \in \mathbb{R}, \rho_i \in \mathbb{R}^+, \pi_i \in \mathbb{R}^+, \sum_{i=1}^{p} \pi_i = 1 \right\}.$$

A generator system is sometimes implicitly required to be in so-called *orthogonal form*, but we skip the details. The function gen maps a generator system to the corresponding convex polyhedron.

The function $\overline{F}$ is defined in a obvious manner and we skip it. (Implementing them is a little hard. See [31] for more details.)

We recall three widening operators on convex polyhedra: the standard widening operator introduced in [46], the widening operator ∇_M up to M introduced in [47, 49] and the precise widening operator introduced in [13].

Definition 2.2.24 (Standard widening)**.** Let $P_1 = \mathsf{con}(C_1), P_2 = \mathsf{con}(C_2) \in \mathbb{CP}_n$ be two convex polyhedra. *The standard widening operator* $\nabla_s : \mathbb{CP}_n \times \mathbb{CP}_n \to \mathbb{CP}_n$ on $\mathbb{CP}_n$ is defined by

$$P_1 \nabla_s P_2 := \begin{cases} P_2 & \text{if } P_1 = \emptyset \\ \mathsf{con}\left(\begin{array}{l} \{i \in \mathsf{repr}(C_1) \mid i \text{ is true everywhere in } P_2\} \\ \cup \left\{ j \in \mathsf{repr}(C_2) \middle| \begin{array}{l} \text{there exists } i \in \mathsf{repr}(C_1) \\ \text{s.t. } P_1 = \mathsf{con}(\mathsf{repr}(C_1)[j/i]) \end{array} \right\} \end{array} \right) & \text{otherwise.} \end{cases}$$

Intuitively, $P_1 \nabla_s P_2$ is represented by the set of linear constraints of P_1 that are satisfied by P_2. The second argument of $\cup$ in the second case is just for well-definedness.

A widening operator that is more precise than the standard widening operator is the following widening up to.

Definition 2.2.25 (Widening up to)**.** Let $P_1, P_2 \in \mathbb{CP}_n$ be two convex polyhedra, ∇_s be the standard widening operator on $\mathbb{CP}_n$ and M be a finite set of linear constraints. *The widening operator up to* M is defined by

$$P_1 \nabla_M P_2 := \mathcal{P}_1 \nabla_s P_2 \cap \mathsf{con}(\{m \in M \mid P_i \subseteq \mathsf{con}(\{m\}) \text{ for } i \in \{1, 2\}\}).$$

By applying the standard widening operator ∇_s, we just discard the linear constraints of P_1 that are violated by P_2. By applying ∇_M, the linear constraints

of P_1 that is violated by P_2 are discarded, but instead the linear constraints in a fixed set M that are satisfied by both P_1 and P_2 are added if any. The set M of linear constraints is usually fixed to the set of linear inequalities that occur in the boolean expressions in the given program (we regard an linear equation as the conjunction of two linear inequalities).

Another widening operator is the precise widening operator. It is defined using a kind of strict preorder defined as follows.

Definition 2.2.26 (∇-compatible limited growth ordering)**.** Let L be a poset. The strict version of a finitely computable preorder on L that satisfies the ascending chain condition is a *limited growth ordering (lgo)*. Let ∇ be a widening operator on L. An lgo $\curvearrowright$ is ∇*-compatible* if $x \sqsubset y \Rightarrow x \curvearrowright x \nabla y$ for all $x, y \in L$.

Definition 2.2.27 (Upper bound operators)**.** Let L be a poset. An operator $h : L \times L \to L$ is an *upper bound operator* if for any $x, y \in L, x \sqsubseteq h(x, y)$ and $y \sqsubseteq h(x, y)$.

The following proposition defines a more precise widening operator than (or a widening operator as precise as) given widening operator.

Proposition 2.2.28 (Improving widening operators)**.** *Let L be a poset, ∇ be a widening operator on L, $\curvearrowright$ be a ∇-compatible lgo on L and $h : L \times L \to L$ be an upper bound operator. The operator ∇' defined as follows is a widening operator on L that is at least as precise as ∇:*

$$x \nabla' y = \begin{cases} h(x, y) & \text{if } x \curvearrowright h(x, y) \sqsubseteq x \nabla y \\ x \nabla y & \text{otherwise.} \end{cases}$$

□

To apply Prop. 2.2.28 to the standard widening operator ∇_s that is defined in Def. 2.2.24, a ∇_s-compatible lgo and an upper bound operator are needed. First, we define a ∇-compatible lgo on the domain of convex polyhedra. As preparations, we introduce the three notions: $\bot$-lifting, multiset ordering [33] and the number of non-null coordinates.

Definition 2.2.29 ($\bot$-lifting)**.** The $\bot$-lifting of a preorder $\preceq$ on L is the preorder $\{(\bot, x) | x \in L\} \cup \preceq$.

Definition 2.2.30 (Multiset ordering)**.** Let M and N be finite multisets over $\mathbb{N}$. The multiset ordering $\sqsubseteq_{ms}$ is defined as follows:

$$M \sqsubseteq_{ms} N \overset{\text{def}}{\iff} \begin{cases} M = N, \text{ or there exists } i \in \mathbb{N} \text{ s.t.} \\ \begin{pmatrix} \#(i, M) > \#(i, N) \text{ and} \\ \text{for all } j \in \mathbb{N}, \left(j > i \Rightarrow \#(j, M) = \#(j, N)\right) \end{pmatrix}, \end{cases}$$

where $\#(k, S)$ is the number of occurrences of $k \in N$ in S. Note that $\sqsubseteq_{ms}$ is a partial order satisfying the ascending chain condition.

Definition 2.2.31 (Number of non-null coordinates)**.** Let V be a subset of $\mathbb{R}^n$. The multiset $\kappa(V)$ is the multiset of the number of non-null coordinates of each vectors in V.

Using these three notions, we define the following order on $\mathbb{CP}_n$.

Definition 2.2.32 ($\curvearrowright_N \subseteq \mathbb{CP}_n \times \mathbb{CP}_n$)**.** Let $P_1 = \mathsf{con}(C_1) = \mathsf{gen}((L_1, R_1, P_1)) \in \mathbb{CP}_n$ and $P_2 = \mathsf{con}(C_2) = \mathsf{gen}((L_2, R_2, P_2)) \in \mathbb{CP}_n$. Assume that P_1 and P_2 are both nonempty. The five preorders $\preceq_d, \preceq_l, \preceq_c, \preceq_p$ and $\preceq_r$ on $\mathbb{CP}_n$ are defined as the $\bot$-lifting of the following preorders:

$$
\begin{aligned}
P_1 \preceq_d P_2 &\overset{\text{def}}{\iff} \#\mathsf{eq}(C_1) \geq \mathsf{eq}(C_2);\\
P_1 \preceq_l P_2 &\overset{\text{def}}{\iff} \#L_1 \leq \#L_2;\\
P_1 \preceq_c P_2 &\overset{\text{def}}{\iff} \#C_1 \geq \#C_2;\\
P_1 \preceq_p P_2 &\overset{\text{def}}{\iff} \#P_1 \geq \#P_2;\\
P_1 \preceq_r P_2 &\overset{\text{def}}{\iff} \kappa(R_1) \sqsubseteq_{ms} \kappa(R_2).
\end{aligned}
$$

The strict version of the lexicographic product of the five preorders $\preceq_d, \preceq_l, \preceq_c, \preceq_p$ and $\preceq_r$ in this order is denoted by $\curvearrowright_N$ and this is a ∇_s-compatible lgo on $\mathbb{CP}_n$.

Four upper bound operators have been introduced in [13]: do not widen (given P_1 and P_2, and just output P_2), combining constraints h_c, evolving points h_p, evolving rays h_r. We do not go into their details because they are not necessary when transferring the meta-theorems with NSA later. The precise widening operator on the domain of convex polyhedra is defined as follows. In the following definition, we assume that $P_1 \sqsubseteq P_2$. This condition is usually satisfied because P_2 is often defined as the convex hull of P_1 and another convex polyhedron.

Definition 2.2.33 (Precise widening operator ∇_N)**.** Let P_1, $P_2 \in \mathbb{CP}_n$. Assume that $P_1 \sqsubseteq P_2$. *The precise widening operator* ∇_N is defined as follows:

$$
P_1 \nabla_N P_2 := \begin{cases}
P_2 & \text{if } P_1 \curvearrowright_N P_2;\\
h_c(P_1, P_2) & \text{if } P_1 \curvearrowright_N h_c(P_1, P_2) \subset P_1 \nabla_s P_2;\\
h_p(P_1, P_2) & \text{if } P_1 \curvearrowright_N h_p(P_1, P_2) \subset P_1 \nabla_s P_2;\\
h_r(P_1, P_2) & \text{if } P_1 \curvearrowright_N h_r(P_1, P_2) \subset P_1 \nabla_s P_2;\\
P_1 \nabla_s P_2 & \text{otherwise.}
\end{cases}
$$

2.2.4 Analysis of Discretized Linear Water Tank by (Standard) Abstract Interpretation

In this section, we illustrate how the abstract interpretation on the domain of convex polyhedra analyzes standard programs (without infinitesimals), using a discretized version of the linear water tank example. The concrete problem is as follows. See Fig.2.1. A water tank has a constant drain (2 cm per second). When the water level x gets lower than 5 cm the switch is turned on, which eventually makes the pump work but only after a time lag of two seconds. While the pump is working, the water level x rises by 1 cm per second. Once x reaches 10 cm, the switch is turned off, which will shut down the pump but again after a time lag of two seconds. Our goal is the *reachability analysis* of this hybrid dynamics, that is, to see the water level x remains in a certain "safe" range (for the hybrid model, we will see that the range is $1 \leq x \leq 12$ later in Chapter 5).

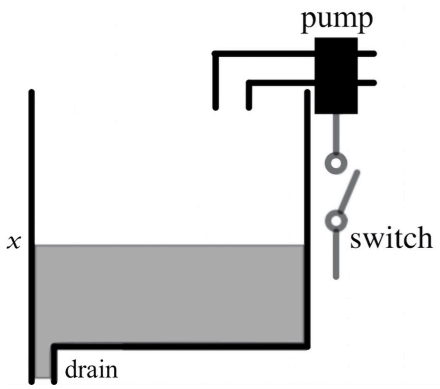

Figure 2.1: A water tank with a drain and a pump, adapted from [91].

```
1  /*Discretized Linear Water Tank*/
2  l := 0; x := 1; p := 1; s := 0; dtprime := 0.2;
3  while true do {
4    if p = 1 then x := x + dtprime
5    else x := x - 2 * dtprime;
6    if (x <= 5 && p = 0) then s := 1
7    else {
8      if (x >= 10 && p = 1) then s := 1
9      else s := 0
10   };
11   if s = 1 then l := l + dtprime
12   else skip;
13   if s = 1 && l >= 2 then {p := 1 - p; s := 0; l := 0}
14   else skip
15 }
```

Code 2.1: Discretized linear water tank

We will use the *discretized* model of the linear water tank in Code 2.1, where each iteration of its unique loop amounts to the lapse of $\texttt{dt}' = 0.2$ (the variable `dtprime` in the code) seconds. The model in Code 2.1 is in an imperative programming language with while loops, a typical subject of analyses by abstract interpretation.

More specifically: x is the water level, l is the counter for the time lag, p stands for the state of the pump ($p = 0$ if the pump is off, and $p = 1$ if on) and s is for "signals," meaning that $s = 1$ if the pump has not yet responded to a signal from the switch (such as, when the switch is on but the pump is not on yet).

The first step in the usual abstract interpretation workflow is to fix *concrete* and *abstract domains*. Here in §2.2.4 we will use the followings.

- **The concrete domain: $\left(\mathcal{P}(\mathbb{R}^2)\right)^4$.** We have two numerical variables l, x and two Boolean ones p, s in Code 2.1, therefore a canonical concrete domain would be $\mathcal{P}(\mathbb{B}^2 \times \mathbb{R}^2)$. We have the powerset operation $\mathcal{P}$ in it since we are now interested in the *reachable* set of memory states.

However, for a better fit with our abstract domain (namely convex polyhedra), we shall use the set $\big(\mathcal{P}(\mathbb{R}^2)\big)^4$ that is isomorphic to the above set $\mathcal{P}(\mathbb{B}^2 \times \mathbb{R}^2)$.

- **The abstract domain: $(\mathbb{CP}_2)^4$.** We use the domain of *convex polyhedra* [31], one of the most commonly-used abstract domains. Recall that a convex polyhedron is a subset of a Euclidean space characterized by a finite conjunction of linear inequalities. Specifically, we let $\mathbb{CP}_2$, the set of 2-dimensional convex polyhedra, approximate the set $\mathcal{P}(\mathbb{R}^2)$. Therefore, as an abstract domain for the program in Code 2.1, we take $(\mathbb{CP}_2)^4$ (that approximates $\big(\mathcal{P}(\mathbb{R}^2)\big)^4$).

The next step in the workflow is to overapproximate the set of memory states that are reachable by the program in Code 2.1—this is a subset of the concrete domain $\big(\mathcal{P}(\mathbb{R}^2)\big)^4$—using the abstract domain $(\mathbb{CP}_2)^4$. Since the desired set can be thought of as a least fixed point, this overapproximation procedure involves: 1) *abstract execution* of the program in $(\mathbb{CP}_2)^4$ (that is straightforward, see e.g. [31]); and 2) acceleration of least fixed-point computation in $(\mathbb{CP}_2)^4$ via suitable use of a *widening operator*. We will use here ∇_M, the widening up to M operator in Def. 2.2.25. One big reason for this choice is the *uniformity* of the operator (a notion we introduce later in §5.2.3), among others. The set M of linear constraints is a parameter for this widening operator; we fix it as usual, collecting the linear constraints that occur in the program in question. That is, $M = \{x \leq 5, x \geq 5, x \leq 10, x \geq 10, l \leq 2, l \geq 2\}$.

This overapproximation procedure is depicted in the *iteration sequence* in Fig. 2.2. Let us look at some of its details. The graph 0 represents the initial memory state (before the first iteration), where the pump is on and the water level x is precisely 1. After one iteration the water level will be incremented by $1 \times \mathtt{dt}' = 0.2$ cm; as usual in abstract interpretation, however, at this moment we invoke the widening operator ∇_M, and the next "abstract reachable set" is $x \in [1, 5]$ instead of $x \in [1, 1.2]$. Here the upper bound 5 comes from the constraint $x \leq 5$ that is in the parameter M of the widening operator ∇_M. This results in the graph 1 in Fig. 2.2.

In the iteration sequence (Fig. 2.2) the four polyhedra (in four different colors) gradually grow: in the graph 2 the water level x can be 10 cm so in the graph 3 appears a green polyhedron (meaning that a signal is sent from the switch to the pump); after the graphs 3 and 9 we *delay* widening, a heuristic commonly employed in abstract interpretation [26]. In the end, in the graph 12 we have a prefixed point (meaning that the polyhedra do not grow any further). There we can see, from the range of x spanned by the polyhedra, that the water level never reaches beyond $0.6 \leq x \leq 12.2$.

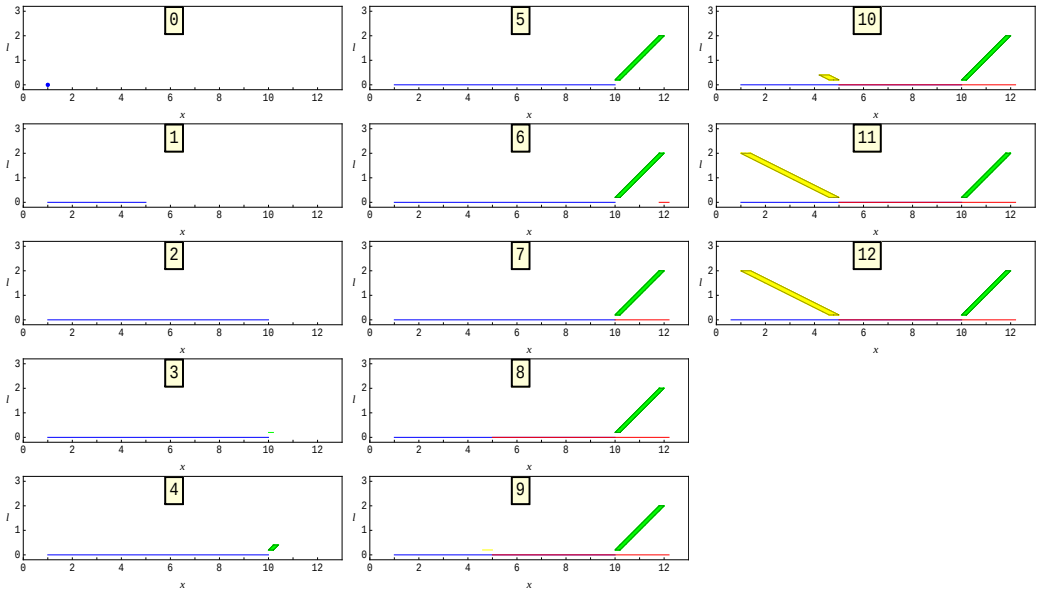

Figure 2.2: An iteration sequence for the linear water tank example.
To save space, here we depict an element of $(\mathbb{CP}_2)^4$—i.e. a quadruple of convex polyhedra—on the same plane $\mathbb{R}^2$. The four convex polyhedra come in different colors: those in blue, green, red and yellow correspond to the values $(p, s) = (1, 0), (1, 1), (0, 0)$ and $(0, 1)$ of the Boolean variables, respectively.

Chapter 3

Approximate Bisimulation for Switching Delays

In this chapter, we propose a methodology to overapproximate the error caused by the switching time delays using approximate bisimulation. The technical development is based on the previous work [43], where the notion of approximate bisimulation is used to obtain symbolic models of incrementally stable switched systems. We consider two switched systems: the ideal one in which switching can only occur at exactly every τ seconds; and the delayed version in which each switching time has a possible delay less than δ_0 seconds. From each of the two switched systems, we first construct a corresponding transition system. We put the same incremental stability assumption δ-GUAS as the one in [43], and assume that Lyapunov functions to witness δ-GUAS are given. Using the Lyapunov functions, we construct a variant of approximate bisimulation relation between the two transition systems.

3.1 Periodic Switched Systems with and without Delays

The models we are considering are given as switched systems introduced in Def. 2.1.1. We are interested in the error between ideal periodic switched system without delays and the one with switching delay. Note that when we say "periodic" in this thesis, we do not mean that the same behavior of the system is repeated with a certain period τ. What we mean is that the change of the mode can occur only at every τ. This is defined formally as follows.

Definition 3.1.1 (periodicity, switching delay)**.** Given a switching signal $\mathbf{p}$, those time instants $t \in \mathbb{R}^+$ where the switching signal $\mathbf{p}$ is discontinuous are called *switching times*. If a switching signal is continuous except at $k\tau$ (where $\tau > 0$ is a constant and $k \in \mathbb{N}$), it is called *τ-periodic*. A switched system $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ is called *τ-periodic* if all the switching signals in $\mathbf{P}$ are τ-periodic.

Let $0 \leq \delta_0 < \tau$. A switching signal $\mathbf{p}$ is said to be *τ-periodic with switching delays within δ_0* if it has at most one discontinuity in each interval $[k\tau, k\tau + \delta_0]$ (where $k \in \mathbb{N}$). A switched system $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ is called *τ-periodic with switching delays within δ_0* if all the switching signals in $\mathbf{P}$ are τ-periodic with switching delays within δ_0.

Given a τ-periodic switching signal, even though switching does not always occur at every $t = k\tau$, we denote the switching that occurs at $t = k\tau$ by *k-th*

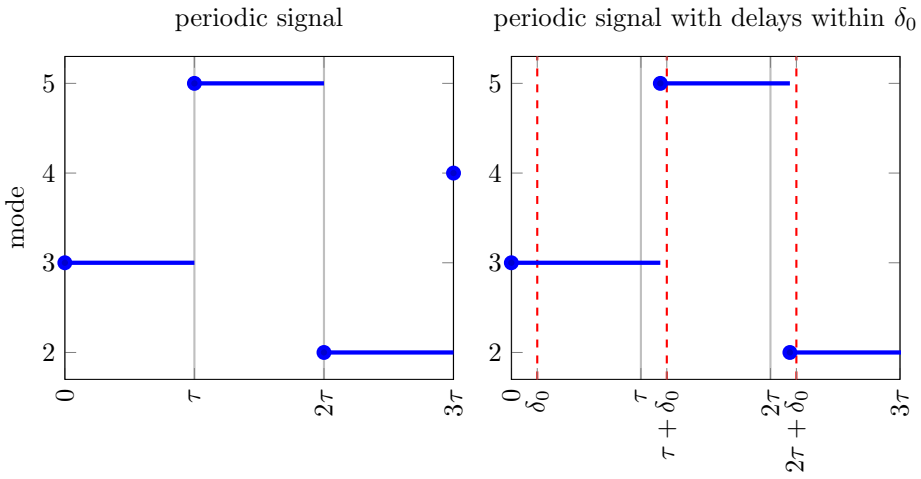

Figure 3.1: Periodic switching signals, with and without delays.

switching. Similarly, given a τ-periodic switching signal with switching delays, *k-th switching* means the switching that occurs at $t \in [k\tau, k\tau + \delta_0]$.

See Fig. 3.1 for illustration of periodic switching signals and those with delays.

In this thesis, we focus on periodic switched systems with switching delays, and their difference from those without switching delays. More specifically, we consider two switched systems

$$\begin{aligned} \Sigma_{\tau,\delta_0} &= (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F) \quad \tau\text{-periodic with delays} \leq \delta_0 \\ \Sigma_{\tau} &= (\mathbb{R}^n, P, \mathbf{P}_{\tau}, F) \quad \tau\text{-periodic} \end{aligned} \tag{3.1}$$

that have a common state space $\mathbb{R}^n$, a common set P of modes and a common set F of vector fields. For the former system Σ_{τ,δ_0}, the set $\mathbf{P}_{\tau,\delta_0}$ consists of all τ-periodic signals with delays within δ_0; for the latter system Σ_{τ} the set $\mathbf{P}_{\tau}$ consists of all τ-periodic switching signals.

3.2 Transition Systems Constructed from Switched Systems

For the two switched systems $\Sigma_{\tau,\delta_0} = (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F)$ and $\Sigma_{\tau} = (\mathbb{R}^n, P, \mathbf{P}_{\tau}, F)$ in (3.1), we shall construct associated transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_{\tau})$, respectively.

Definition 3.2.1 $(T(\Sigma_{\tau,\delta_0}), T(\Sigma_{\tau}))$**.** The transition system

$$T(\Sigma_{\tau,\delta_0}) = (Q_{\tau,\delta_0}, L, \xrightarrow[\tau,\delta_0]{}, O, H_{\tau,\delta_0}, I) \ ,$$

associated with the switched system Σ_{τ,δ_0} with delays in (3.1), is defined as follows:

- the set of states is $Q_{\tau,\delta_0} := \mathbb{R}^n \times \bigcup_{k\in\mathbb{N}}[k\tau, k\tau + \delta_0] \times P$;
- the set of labels L is the set of modes, i.e. $L := P$;

- the transition relation $\xrightarrow[\tau,\delta_0]{} \subseteq Q_{\tau,\delta_0} \times L \times Q_{\tau,\delta_0}$ is defined by $(x,t,p)\xrightarrow[\tau,\delta_0]{p''}(x',t',p')$ if $p = p''$, $x' = \mathbf{x}(t'-t,x,p)$ and there exists $k \in \mathbb{N}$ such that $t \in [k\tau, k\tau+\delta_0]$ and $t' \in [(k+1)\tau, (k+1)\tau+\delta_0]$;
- the set of outputs is $O := \mathbb{R}^n \times \mathbb{R}^+ \times P$;
- the output function $H_{\tau,\delta_0} \colon Q_{\tau,\delta_0} \to O$ is the canonical embedding function $\mathbb{R}^n \times \bigcup_{k\in\mathbb{N}}[k\tau, k\tau+\delta_0] \times P \to \mathbb{R}^n \times \mathbb{R}^+ \times P$; and
- the set of initial states is $I := \mathbb{R}^n \times \{0\} \times P$.

Intuitively, each state (x,t,p) of $T(\Sigma_{\tau,\delta_0})$ marks switching in the system Σ_{τ,δ_0}: $x \in \mathbb{R}^n$ is the (continuous) state at switching; t is time of switching; and p is the next mode. Note that, by the assumption on Σ_{τ,δ_0}, t necessarily belongs to the interval $[k\tau, k\tau+\delta_0]$ for some $k \in \mathbb{N}$.

Similarly, the transition system

$$T(\Sigma_\tau) = (Q_\tau, L, \xrightarrow[\tau]{}, O, H_\tau, I) \ ,$$

associated with the switched system Σ_τ without delays in (3.1), is defined as follows:

- the set of states is $Q_\tau := \mathbb{R}^n \times \{0, \tau, 2\tau, \dots\} \times P$;
- the set of labels L is the set of modes, i.e. $L := P$;
- the transition relation $\xrightarrow[\tau]{} \subseteq Q_\tau \times L \times Q_\tau$ is defined by $(x,t,p)\xrightarrow[\tau]{p''}(x',t',p')$ if $p = p''$, $t' = t+\tau$ and $x' = \mathbf{x}(\tau,x,p)$;
- the set of outputs is $O := \mathbb{R}^n \times \mathbb{R}^+ \times P$;
- the output function $H_\tau \colon Q_\tau \to O$ is the canonical embedding function; and
- the set of initial states is $I := \mathbb{R}^n \times \{0\} \times P$.

Note that, in both of $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$, the label p'' for a transition is uniquely determined by the mode component p of the transition's source (x,t,p). Therefore, mathematically speaking, we do not need transition labels.

In [43], the state space Q of the transition system is defined to be $\mathbb{R}^n$ and is the same as the state space of the original switched system. In comparison, our definition has two additional components, namely time t and the current mode p. It is notable that moving a mode p from transition labels to state labels allows us to analyze what happens during switching delays, that is, when the system keeps operating under the mode p while it is not supposed to do so.

Definition 3.2.2 (premetric on outputs)**.** On the set of outputs $O = \mathbb{R}^n \times \mathbb{R}^+ \times P$ that is common to the two transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$, we define the following premetric d:

$$d((x,t,p),(x',t',p')) := \begin{cases} \|x - \mathbf{x}(t-t',x',p)\| & \text{if } p = p', t' = k\tau \text{ and} \\ & \quad t \in [t', t'+\delta_0] \text{ for some } k \in \mathbb{N} \\ \infty & \text{otherwise.} \end{cases}$$

3.3 Relaxation of Approximate Bisimulation

In this thesis we are interested not only in a time-invariant precision ε, but also in an error bound that can grow in time. For this purpose we introduce the following relaxed notion of approximate bisimulation. It allows errors to potentially grow after each transition, in a manner regulated by some increasing function $g\colon \mathbb{R}^+ \to \mathbb{R}^+$.

Definition 3.3.1 (g-incrementing approximate bisimulation)**.** Let $T_i = (Q_i, L, \xrightarrow[i]{}, O, H_i, I_i)$ $(i = 1, 2)$ be two premetric transition systems with premetric d; they share the same sets of actions L and outputs O. Let $\varepsilon \in \mathbb{R}^+$ be a *precision*, and $g\colon \mathbb{R}^+ \to \mathbb{R}^+$ be an increasing function.

A family $\{R_\varepsilon\}_{\varepsilon \geq 0}$ of relations $R_\varepsilon \subseteq Q_1 \times Q_2$ indexed by $\varepsilon \geq 0$ is called an *g-incrementing approximate bisimulation* between T_1 and T_2 if the following conditions hold for all $\varepsilon \geq 0$ and for all $(q_1, q_2) \in R_\varepsilon$:

1. $d(H_1(q_1), H_2(q_2)) \leq \varepsilon$; and
2. There exists a function g such that
   (a) for all l and q_1' satisfying $q_1 \xrightarrow[1]{l} q_1'$, there exists q_2' such that $q_2 \xrightarrow[2]{l} q_2'$ and $(q_1', q_2') \in R_{g(\varepsilon)}$ hold; and
   (b) for all l and q_2' satisfying $q_2 \xrightarrow[2]{l} q_2'$, there exists q_1' such that $q_1 \xrightarrow[1]{l} q_1'$ and $(q_1', q_2') \in R_{g(\varepsilon)}$ hold.

3.4 Approximate Bisimulation for Switching Delays I: Common Lyapunov Functions

In §2.1.2, we reviewed two witness notions for the incremental stability notion of δ-GUAS: common and multiple δ-GAS Lyapunov functions. In [43], these two notions are successfully exploited to yield discrete-state abstraction of switched systems. It is our main contribution to leverage the same incremental stability assumptions and derive upper bounds for errors caused by switching delays. We focus on systems with periodic switching intervals, as already announced. The use of common δ-GAS Lyapunov functions is described in this section; the use of multiple δ-GAS Lyapunov functions is in the next section.

We will be using the following assumption.

Assumption 3.4.1 (bounded intermode derivative)**.** Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system, with $P = \{1, 2, \ldots, m\}$ and $F = \{f_1, f_2, \ldots, f_m\}$ being the set of vector fields associated with each mode. We say a function $V\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ has *bounded intermode derivatives* if there exists a real number $\nu \geq 0$ such that, for any $p, p' \in P$ that are distinct $(p \neq p')$, the inequality

$$\frac{\partial V}{\partial x}(x, y) f_p(x) + \frac{\partial V}{\partial y}(x, y) f_{p'}(y) \leq \nu \tag{3.2}$$

holds for each $x, y \in \mathbb{R}^n$.

Remark 3.4.2. Assumption 3.4.1 seems to be new: it is not assumed in the previous works on approximate bisimulation for switched systems, such as [43].

Imposing the assumption on δ-GAS Lyapunov functions, however, is not a severe restriction. In [43], they assume that there exists $\gamma \in \mathbb{R}^+$ such that, for all $x, y, z \in \mathbb{R}^n$,

$$|V(x,y) - V(x,z)| \leq \gamma(\|y - z\|) \tag{3.3}$$

(we do not need this assumption in the current work). It is claimed in [43] that (3.3) is readily guaranteed if the dynamics of the switched system is confined to a compact set $C \subseteq \mathbb{R}^n$, and if V is class $\mathcal{C}^1$ in the domain C. We can use the same compactness argument to ensure Assumption 3.4.1.

Definition 3.4.3 (the function V')**.** Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system, and let $V \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be a common δ-GAS Lyapunov function for Σ.

We define a function $V' : (\mathbb{R}^n \times \mathbb{R}^+ \times P) \times (\mathbb{R}^n \times \mathbb{R}^+ \times P) \to \mathbb{R}^+$ in the following manner:

$$V'\big((x,t,p),(x',t',p')\big) := \begin{cases} V\big(x, \mathbf{x}(t-t',x',p')\big) & \text{if } p = p' \text{ and } t \in [t', t'+\delta_0] \\ \infty & \text{otherwise.} \end{cases}$$

Recall that $\mathbf{x}(t-t', x', p')$ is the state reached from x' after time $t - t'$ following the vector field $f_{p'}$.

Here is our main technical lemma.

Lemma 3.4.4. *Let $\Sigma_\tau = (\mathbb{R}^n, P, \mathbf{P}_\tau, F)$ be a τ-periodic switched system, and $\Sigma_{\tau,\delta_0} = (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F)$ be a τ-periodic switched system with delays within δ_0. Assume that there exists a common δ-GAS Lyapunov function V for Σ_τ, and that V satisfies the additional assumption in Assumption 3.4.1. Then, for a suitable g, there exists a g-incrementing approximate bisimulation $\{R_\varepsilon\}_{\varepsilon \geq 0}$ between the transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$.*

Specifically, we define a function g by

$$g(\varepsilon) := \underline{\alpha}^{-1}\left(e^{-\kappa(\tau-\delta_0)}\underline{\alpha}(\varepsilon) + \nu\delta_0\right) \ ,$$

where $\underline{\alpha}$ and κ are from Def. 2.1.7 and ν is from Assumption 3.4.1. For each $\varepsilon \geq 0$, we define a relation $R_\varepsilon \subseteq (\mathbb{R}^n \times \mathbb{R}^+ \times P) \times (\mathbb{R}^n \times \mathbb{R}^+ \times P)$ by

$$(q,q') \in R_\varepsilon \overset{\text{def.}}{\iff} V'(q,q') \leq \underline{\alpha}(\varepsilon) \ . \tag{3.4}$$

Here V' is from Def. 3.4.3.

Proof. We will show that Condition 1 and Condition 2a in Def. 3.3.1 hold. Condition 2b can be proved in a similar way as Condition 2a, so we omit it.

For $q_{\tau,\delta_0} = (x_{\tau,\delta_0}, t_{\tau,\delta_0}, p_{\tau,\delta_0})$ and $q_\tau = (x_\tau, t_\tau, p_\tau)$, we assume that $(q_{\tau,\delta_0}, q_\tau) \in R_\varepsilon$ holds. From the construction of the transition system $T(\Sigma_\tau)$, we have

$$t_\tau = k\tau \text{ for some } k \in \mathbb{N}. \tag{3.5}$$

By the definition (3.4) of the relation R_ε and Def. 3.4.3, we have

$$p_{\tau,\delta_0} = p_\tau, \tag{3.6}$$
$$t_{\tau,\delta_0} \in [t_\tau, t_\tau + \delta_0], \text{ and} \tag{3.7}$$
$$V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) \leq \underline{\alpha}(\varepsilon). \tag{3.8}$$

By (2.3), we have

$$\underline{\alpha}\|x_{\tau,\delta_0} - \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)\| \leq V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)). \tag{3.9}$$

Then, by (3.8) and (3.9), we can say that $\underline{\alpha}\|x_{\tau,\delta_0} - \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)\| \leq \underline{\alpha}(\varepsilon)$. Thus, using the monotonicity of $\underline{\alpha}$, we have $\|x_{\tau,\delta_0} - \mathbf{x}(\delta, x_\tau, p_\tau)\| \leq \varepsilon$. By this inequality with the side conditions (3.5)–(3.7), we have $d(q_{\tau,\delta_0}, q_\tau) \leq \varepsilon$. This proves Condition 1.

Next, we assume that $q_{\tau,\delta_0} \xrightarrow[\tau,\,\delta_0]{p_\tau} q'_{\tau,\delta_0} = (\mathbf{x}(t'_{\tau,\delta_0} - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), t'_{\tau,\delta_0}, p'_{\tau,\delta_0})$ and $(q_{\tau,\delta_0}, q_\tau) \in R_\varepsilon$ for some ε. We note that in addition to (3.5)–(3.8), we have

$$t'_{\tau,\delta_0} \in [(k+1)\tau, (k+1)\tau + \delta_0], \tag{3.10}$$

for the same k as in (3.5). Our goal is to show that there exists $q'_\tau = (x'_\tau, t'_\tau, p'_\tau)$ such that $q_\tau \xrightarrow[\tau]{p_\tau} q'_\tau$ and $(q'_{\tau,\delta_0}, q'_\tau) \in R_{g(\varepsilon)}$. Specifically, we define q'_τ by $q'_\tau = (\mathbf{x}(\tau, x_\tau, p_\tau), t'_\tau, p'_\tau)$ where

$$\begin{aligned} t'_\tau &= (k+1)\tau, \text{ and} \\ p'_\tau &= p'_{\tau,\delta_0}. \end{aligned} \tag{3.11}$$

This definition of q'_τ guarantees $q_\tau \xrightarrow[\tau]{p_\tau} q'_\tau$.

Now we show $(q'_{\tau,\delta_0}, q'_\tau) \in R_{g(\varepsilon)}$ for this q'_τ. Note that (3.8) refers to the states of the two systems at the same time instant $t = t_{\tau,\delta_0}$. When time progresses for $t'_\tau - t_{\tau,\delta_0}$ with mode $p_{\tau,\delta_0} = p_\tau$ for both systems from $t = t_{\tau,\delta_0}$, we have

$$\begin{aligned} & V(\mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau)) \\ &\leq e^{-\kappa(t'_\tau - t_{\tau,\delta_0})} V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) \qquad \because (2.4) \\ &\leq e^{-\kappa(\tau - \delta_0)} V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) \qquad \because (3.7) \text{ and } (3.10). \end{aligned} \tag{3.12}$$

Note that the inequality (3.12) refers to the states of the two systems at $t = t'_\tau$. When time progresses for $t'_{\tau,\delta_0} - t'_\tau$ with mode $p_{\tau,\delta_0} (= p_\tau)$ for $T_{\tau,\delta_0}(\Sigma_{\tau,\delta_0})$ and with mode p'_τ for $T_\tau(\Sigma_\tau))$ from $t = t'_\tau$, we have

$$\begin{aligned} & V'(q'_{\tau,\delta_0}, q'_\tau) \\ &= V(\mathbf{x}(t'_{\tau,\delta_0} - t'_\tau, \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), p_\tau), \\ &\qquad \mathbf{x}(t'_{\tau,\delta_0} - t'_\tau, \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau), p'_\tau)) \\ &\leq V(\mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau)) + \nu(t'_{\tau,\delta_0} - t'_\tau) \\ & \qquad\qquad \because (3.2) \\ &\leq e^{-\kappa(\tau - \delta_0)} V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) + \nu(t'_{\tau,\delta_0} - t'_\tau) \qquad \because (3.12) \\ &\leq e^{-\kappa(\tau - \delta_0)} V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) + \nu\delta_0 \qquad \because (3.10) \text{ and } (3.11) \\ &\leq e^{-\kappa(\tau - \delta_0)} \underline{\alpha}(\varepsilon) + \nu\delta_0 \qquad \because (3.8) \\ &= \underline{\alpha}(g(\varepsilon)). \end{aligned} \tag{3.13}$$

Thus we have $(q'_{\tau,\delta_0}, q'_\tau) \in R_{g(\varepsilon)}$ and this proves Condition 2a in Def. 3.3.1. □

The last lemma about the step-wise growth of errors is used below to derive global error bounds. We compare the trajectories of Σ_{τ,δ_0} and Σ_τ from the same initial state x.

Theorem 3.4.5. *Assume the same assumptions as in Lem. 3.4.4. Let $\mathbf{p}_\tau$ be a τ-periodic switching signal, and $\mathbf{p}_{\tau,\delta_0}$ be the same signal but with delays within δ_0. That is, for each $s \in \mathbb{R}^+$,*

$$\mathbf{p}_{\tau,\delta_0}(s) = \begin{cases} \mathbf{p}_\tau(s) \text{ or } \mathbf{p}_\tau(s-\delta_0) & \text{if } s \in \bigcup_{k\in\mathbb{N},k\geq 1}[k\tau, k\tau+\delta_0) \\ \mathbf{p}_\tau(s) & \text{otherwise.} \end{cases}$$

(a) We have, for each $k \in \mathbb{N}$ and $t \in [k\tau,(k+1)\tau)$,

$$\left\|\mathbf{x}(t,x,\mathbf{p}_{\tau,\delta_0}) - \mathbf{x}(t,x,\mathbf{p}_\tau)\right\| \leq \underline{\alpha}^{-1}\left(\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}} + e^{-\kappa(\tau-\delta_0)k}\left(-\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}}\right)\right) .$$

(b) We have, for each $t \in \mathbb{R}^+$,

$$\left\|\mathbf{x}(t,x,\mathbf{p}_{\tau,\delta_0}) - \mathbf{x}(t,x,\mathbf{p}_\tau)\right\| \leq \underline{\alpha}^{-1}\left(\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}}\right) .$$

Note that the bound in Thm. 3.4.5 (a) can grow over time (i.e. over the number k of switching); the one in Thm. 3.4.5 (b) is a conservative bound that is time-invariant. We also note that, for any desired precision ε, there always exists a small enough delay bound δ_0 that achieves the precision ε (i.e. $\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}} \leq \varepsilon$).

Proof. Lem. 3.4.4 serves as a recurrence relation with respect to the number k of switching. By solving it with the initial condition of $d(q_{\tau,\delta_0,0}, q_{\tau,0}) = 0$, we obtain the result of

$$\begin{aligned} d(q_{\tau,\delta_0}, q_\tau) &\leq g^k(0) \\ &= \underline{\alpha}^{-1}\left(\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}} + e^{-\kappa(\tau-\delta_0)k}\left(-\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}}\right)\right), \end{aligned}$$

for all states q_{τ,δ_0} and q_τ that can be reached via a same sequence of actions of length k.

Since the definition of the premetric d is as Def. 3.2.2, this result only refers to the error between two systems at the switching time in $[k\tau,(k+1)\tau)$. It is straightforward, however, to have that this $\underline{\alpha}^{-1}\left(\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}} + e^{-\kappa(\tau-\delta_0)k}\left(-\frac{\nu\delta_0}{1-e^{-\kappa(\tau-\delta_0)}}\right)\right)$ is actually an upper bound of $\left\|\mathbf{x}(t,x,\mathbf{p}_{\tau,\delta_0}) - \mathbf{x}(t,x,\mathbf{p}_\tau)\right\|$, for all $t \in [k\tau,(k+1)\tau)$, from (2.4).

The error bound given in (a) is easily seen to be increasing with respect to k, and to have an obvious upper bound. This proves (b). □

Remark 3.4.6. It turns out that the upper bound $\overline{\alpha}$ of a δ-GAS Lyapunov function V (see (2.1)) is not used in the above results nor their proofs. In [43], the upper bound $\overline{\alpha}$ is used to define the state space discretization parameter η so that, for each initial state $q_1 \in I_1$, there will be an approximately bisimilar initial state in I_2 and vice versa. This is not necessary in our current setting where there is an obvious correspondence between the initial states.

That $\overline{\alpha}$ is unnecessary is also the case with the multiple Lyapunov function case in the next section.

3.5 Approximate Bisimulation for Switching Delays II: Multiple Lyapunov Functions

We follow [43] and investigate the use of another witness for δ-GUAS incremental stability—namely multiple δ-GAS Lyapunov functions, see §2.1.2—for bounding errors caused by switching delays.

The following is an analogue of Assumption 3.4.1.

Assumption 3.5.1. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system with $P = \{1, 2, \dots, m\}$. Let $V_1, \dots, V_m \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be smooth functions. We say the functions $V_1, \dots, V_m$ have *bounded intermode derivatives* if there exists a real number $\nu' \geq 0$ such that, for each $p, p' \in P$ that are distinct ($p \neq p'$), the inequality

$$\frac{\partial V_{p'}}{\partial x}(x, y) f_p(x) + \frac{\partial V_{p'}}{\partial y}(x, y) f_{p'}(y) \leq \nu' \tag{3.14}$$

holds for each $x, y \in \mathbb{R}^n$. (Note the occurrences of p and p'.)

Lemma 3.5.2. *Let $\Sigma_\tau = (\mathbb{R}^n, P, \mathbf{P}_\tau, F)$ be a τ-periodic switched system and $\Sigma_{\tau,\delta_0} = (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F)$ be a τ-periodic switched system with delays within δ_0. Assume that for each $p \in P$, there is a δ-GAS Lyapunov function V_p for the single-mode subsystem $\Sigma_{\tau,p}$. We additionally assume Assumption 3.5.1 for $V_1, \dots, V_m$, and that there exists $\mu \in \mathbb{R}^+$ such that*

$$V_p(x, y) \leq \mu V_{p'}(x, y) \text{ for all } x, y \in \mathbb{R}^n \text{ and } p, p' \in P \ . \tag{3.15}$$

The last assumption is the same as in Thm. 2.1.9.

Then, for a suitable g, there exists a g-incrementing approximate bisimulation $\{R_\varepsilon\}_{\varepsilon \geq 0}$ between the transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$.

Specifically, we define g by $g(\varepsilon) := \underline{\alpha}^{-1}\left(\mu e^{-\kappa(\tau-\delta_0)}\underline{\alpha}(\varepsilon) + \nu'\delta_0\right)$, where $\underline{\alpha}$ and κ are from (2.5) and ν' is from Assumption 3.5.1. For each $\varepsilon \geq 0$, we define a relation $R_\varepsilon \subseteq (\mathbb{R}^n \times \mathbb{R}^+ \times P) \times (\mathbb{R}^n \times \mathbb{R}^+ \times P)$ by $(q, q') \in R_\varepsilon \overset{\text{def.}}{\iff} V'(q, q') \leq \underline{\alpha}(\varepsilon)$. Here the function V' is defined as follows, adapting Def. 3.4.3 to the current multiple Lyapunov function setting.

$$V'\big((x,t,p),(x',t',p')\big) := \begin{cases} V_p\big(x, \mathbf{x}(t-t', x', p')\big) & \text{if } p = p' \text{ and } t \in [t', t'+\delta_0] \\ \infty & \text{otherwise.} \end{cases}$$

Proof. The proof of this lemma is almost the same as that of Lem. 3.4.4. The only difference is, after we derive

$$\begin{aligned} &V_p(\mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), \\ &\quad \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau)) \\ &\leq e^{-\kappa(\tau-\delta_0)} V_p(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)), \end{aligned}$$

which is the counterpart of the inequality (3.12), we derive

$$\begin{aligned} &V_{p'}(\mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), \\ &\quad \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau)) \\ &\leq \mu e^{-\kappa(\tau-\delta_0)} V_p(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) \end{aligned}$$

by using (3.15). Then, similarly to (3.13), we obtain the inequality

$$\begin{aligned} &V_{p'}(\mathbf{x}(t'_{\tau,\delta_0} - t'_\tau, \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), p_\tau), \\ &\quad \mathbf{x}(t'_{\tau,\delta_0} - t'_\tau, \mathbf{x}(t'_\tau - t_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau), p_\tau), p'_\tau)) \\ &\leq \mu e^{-\kappa(\tau-\delta_0)} V(x_{\tau,\delta_0}, \mathbf{x}(t_{\tau,\delta_0} - t_\tau, x_\tau, p_\tau)) + \nu'\delta_0. \end{aligned}$$

The rest of the proof is straightforward. □

The next result follows from Lem. 3.5.2. The proof is omitted since it is almost the same as Thm. 3.4.5. Note that in the following theorem, we need a *dwell-time assumption* $\tau - \delta_0 > \frac{\log \mu}{\kappa}$, which was not necessary for the common Lyapunov function case in Thm. 3.4.5, to derive the time-invariant upper bound. It is used to guarantee the convergence of the sequence $[g^k(0)]_{k\in\mathbb{N}}$

Theorem 3.5.3. *Assume the same assumptions as in Lem. 3.5.2, and let* $\mathbf{p}_\tau$ *and* $\mathbf{p}_{\tau,\delta_0}$ *be those periodic switching signals, without and with delays, as in Thm. 3.4.5.*

(a) We have, for each $k \in \mathbb{N}$ *and* $t \in [k\tau, (k+1)\tau)$,

$$\begin{aligned} &\left\|\mathbf{x}(t, x, \mathbf{p}_{\tau,\delta_0}) - \mathbf{x}(t, x, \mathbf{p}_\tau)\right\| \leq \\ &\quad \underline{\alpha}^{-1}\left(\frac{\nu'\delta_0}{1-\mu e^{-\kappa(\tau-\delta_0)}} + \mu e^{-\kappa(\tau-\delta_0)k}\left(-\frac{\nu'\delta_0}{1-\mu e^{-\kappa(\tau-\delta_0)}}\right)\right) . \end{aligned}$$

(b) If $\tau - \delta_0 > \frac{\log \mu}{\kappa}$, *we have, for each* $t \in \mathbb{R}^+$,

$$\left\|\mathbf{x}(t, x, \mathbf{p}_{\tau,\delta_0}) - \mathbf{x}(t, x, \mathbf{p}_\tau)\right\| \leq \underline{\alpha}^{-1}\left(\frac{\nu'\delta_0}{1-\mu e^{-\kappa(\tau-\delta_0)}}\right) . \qquad \square$$

3.6 Examples

We demonstrate our framework using two examples. The first is the boost DC-DC converter from [16], a common example of switched systems that is also used in [43]. For this example we have a common δ-GAS Lyapunov function V, and therefore we appeal to the results in §3.4. The second example is a water tank with nonlinear dynamics. It has multiple δ-GAS Lyapunov functions, and we use the results in §3.5.

3.6.1 Boost DC-DC Converter

System Description The system we consider is the boost DC-DC converter in Fig. 3.2. It is taken from [16]; here we follow and extend its analysis in [43]. The circuit includes a capacitor with capacitance x_c and an inductor with inductance x_l. The capacitor has the equivalent series resistance r_c, and the inductor has the internal resistance r_l. The input voltage is v_s, and the resistance r_o is the output load resistance. The state $x(t) = \begin{bmatrix} i_l(t) \\ v_c(t) \end{bmatrix}$ of this system consists of the inductor current i_l and the capacitor voltage v_c.

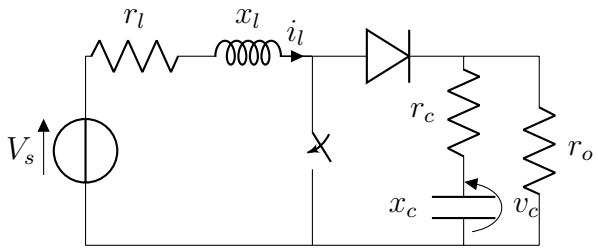

Figure 3.2: The boost DC-DC converter circuit.

The dynamics of this system has two modes $\{ON, OFF\}$[1], depending on whether the switch in the circuit is on or off. By elementary circuit theory, the dynamics in each mode is modeled by

$$\dot{x}(t) = A_p x(t) + b \quad \text{for } p \in \{ON, OFF\} \ , \text{ where}$$
$$A_{ON} = \begin{bmatrix} -\frac{r_l}{x_l} & 0 \\ 0 & -\frac{1}{x_c(r_o+r_c)} \end{bmatrix},$$
$$A_{OFF} = \begin{bmatrix} -\frac{r_l r_o + r_l r_c + r_o r_c}{x_l(r_o+r_c)} & -\frac{r_l r_o + r_l r_c + r_o r_c}{x_l(r_o+r_c)} \\ \frac{r_o}{x_c(r_o+r_c)} & -\frac{1}{x_c(r_o+r_c)} \end{bmatrix} \text{ and}$$
$$b = \begin{bmatrix} \frac{v_s}{x_l} \\ 0 \end{bmatrix} \ .$$

We use the parameter values from [16], that is, $x_c = 70$ p.u., $x_l = 3$ p.u., $r_c = 0.005$ p.u., $r_l = 0.05$ p.u., $r_o = 1$ p.u. and $v_s = 1$ p.u. The same parameter values are used in [43].

Analysis Following [43], we rescale the second variable of the system and redefine the state $x(t) = \begin{bmatrix} i_l(t) \\ 5v_c(t) \end{bmatrix}$ for better numerical conditioning. The ODEs are updated accordingly.

It is shown in [43] that the dynamics in each mode is δ-GAS. They share a common δ-GAS Lyapunov function

$$V(x,y) = \sqrt{(x-y)^T M (x-y)}, \text{ with } M = \begin{bmatrix} 1.0224 & 0.0084 \\ 0.0084 & 1.0031 \end{bmatrix}.$$

The common Lyapunov function V has $\underline{\alpha}(s) = s, \overline{\alpha}(s) = 1.0127s$ and $\kappa = 0.014$. This common Lyapunov function was discovered in [43] via SDP optimization; we use the same function as an ingredient for our approximate bisimulation.

Our ultimate goal is to synthesize a switching signal that keeps the dynamics in a safe region $\mathcal{S} := [1.3, 1.7] \times [5.7, 5.8]$. We shall follow the two-step workflow in Fig. 6.1.

Let us first use Thm. 3.4.5 and derive a bound ε_1 for errors caused by switching delays. We set the switching period $\tau = 0.5$ (this is the same as in [43]), and the maximum delay $\delta_0 = \frac{\tau}{1000}$. On top of the analysis in [43], we have to verify

[1] In the formalization of §2.1, the set P of modes is declared as $\{1, \cdots, m\}$. Here we instead use $P = \{ON, OFF\}$ for readability. The same applies to the water tank example in §3.6.2.

the condition we additionally impose (namely Assumption 3.4.1). Let us now assume that the dynamics stays in the safe region $\mathcal{S} = [1.3, 1.7] \times [5.7, 5.8]$—this assumption will be eventually discharged when we synthesize a safe controller. Then it is not hard to see that $\nu = 0.41$ satisfies the inequality (3.2). By Thm. 3.4.5, we obtain that the error between Σ_{τ,δ_0} (the boost DC-DC converter with delays) and Σ_τ (the one without delays) is bounded by $\varepsilon = 0.0294176$.

3.6.2 Nonlinear Water Tank

System Description The second example demonstrates our framework's applicability to nonlinear dynamics.

The water tank we consider is equipped with a drain and a valve. The system has two modes. When the switch is off, the drain is open and the valve is closed, causing the water level to decrease. When the switch is on, the drain is closed and the valve aperture is set according to the water level. We assume that dynamics of the water level x is modeled by:

$$\dot{x} = \begin{cases} f_{OFF}(x) := -a\sqrt{x} & \text{when the switch is off,} \\ f_{ON}(x) := b(c - x) & \text{when the switch is on.} \end{cases} \tag{3.16}$$

The behavior for the mode *OFF* is a well-known water level behavior, found e.g. in the MATLAB®/Simulink® example [93]. The water leaves at a rate that is proportional to $\sqrt{x}$. The behavior of the mode *ON* is a natural one when the valve aperture is governed by a float, as found in many toilet tanks.

Let us set the three parameters $a = \frac{1}{5}, b = \frac{1}{10}$ and $c = 11$. Our scenario is that we would like to control the switch so that the water level should stay in $[1, 10]$. We assume there are switching delays within $\delta_0 = 0.1$ seconds. We fix the switching period τ to be 10 seconds.

Analysis The dynamics of each mode has a δ-GAS Lyapunov function defined by

$$V_{OFF}(x, y) := |e^{\sqrt{x}} - e^{\sqrt{y}}| \quad \text{and} \quad V_{ON}(x, y) := |\sqrt{6}(x - y)| \ .$$

We obtain the following characteristics for these two δ-GAS Lyapunov functions in the safe region $[1, 10]$: $\underline{\alpha}_{OFF}(s) = s$, $\underline{\alpha}_{ON}(s) = \sqrt{6}s$, $\kappa_{OFF} = \kappa_{ON} = \frac{1}{10}$, $\mu = \frac{2\sqrt{6}}{3}$, and $\nu' = 2.94$.

These characteristics satisfy the dwell-time assumption $\tau - \delta_0 > \frac{\log \mu}{\kappa}$ in Thm. 3.5.3, and thus we obtain that the error between Σ_{τ,δ_0} and Σ_τ is bounded by $\varepsilon = 0.747678$.

Remark 3.6.1. The above Lyapunov functions $V_{OFF}(x, y)$ and $V_{ON}(x, y)$ for nonlinear water tank are not smooth at $x = y$, and it is not allowed in Def. 2.1.4. Intuitively, it is not a problem because we just need to consider the three directional derivatives at $x = y$. To deal with this nonsmoothness in a technically rigorous way, we can rely on Clarke's nonsmooth analysis, which nicely accommodates set-valued generalized derivatives of our Lyapunov functions when $x = y$. More concretely, our functions are *Clarke regular* and locally Lipschitz, and therefore we can apply standard results such as those in [11]. We do not get into its details in this thesis.

Chapter 4

Skorokhod Distance Caused by Switching Delays

In this chapter, we will extend the methodology we introduced in Chapter 3 by changing the definition of the distance. In Chapter 3, the distance in Def. 3.2.2 between two states of the transition systems is defined pointwisely, so that it will compare the states of the switched systems at the same time moment. Therefore, as stated in Thm. 3.4.5 and Thm. 3.5.3, the obtained bound ε overapproximates $\sup_{t\in\mathbb{R}^+}\left\|\mathbf{x}(t,x,\mathbf{p}_{\tau,\delta_0})-\mathbf{x}(t,x,\mathbf{p}_\tau)\right\|$, i.e. the error between the states of the switched systems at the same time instant. However, for a wide variety of applications, it is not necessary to compare the states at the same time instant. One simple example application is the reachability analysis. Assume that we obtain the error bound ε between Σ_τ and Σ_{τ,δ_0}. We can easily compute an overapproximation of the reachable set of the system Σ_{τ,δ_0} with delays from an overapproximation of the reachable set of the system Σ_τ without delays, just by enlarging it with the error bound ε. However, the necessary and sufficient condition to obtain a sound overapproximation of the reachable set of the system with delays is that ε is larger than $\sup_{t_\tau\in\mathbb{R}^+}\left(\inf_{t_{\tau,\delta_0}\in\mathbb{R}^+}\left\|\mathbf{x}(t_{\tau,\delta_0},x,\mathbf{p}_{\tau,\delta_0})-\mathbf{x}(t_\tau,x,\mathbf{p}_\tau)\right\|\right)$, which is smaller than $\sup_{t\in\mathbb{R}^+}\left\|\mathbf{x}(t,x,\mathbf{p}_{\tau,\delta_0})-\mathbf{x}(t,x,\mathbf{p}_\tau)\right\|$.

In [34], the conformance between two trajectory was defined using the Skorokhod metric. It has more general applications than the reachability analysis explained above. This distance is smaller than the pointwise distance in Chapter 3, but still can be used for sound analysis of a variant of the timed linear time logic (TLTL) or Freeze linear time logic (FLTL) specifications. We show that our extended methodology computes an upper bound of the Skorokhod metric.

4.1 Changing the Definition of the Premetric

The setting we consider in this section is almost the same as the one in Chapter 3. Let $\Sigma_{\tau,\delta_0}=(\mathbb{R}^n,P,\mathbf{P}_{\tau,\delta_0},F)$ be a τ-periodic switched system with switching delays within δ_0, and $\Sigma_\tau=(\mathbb{R}^n,P,\mathbf{P}_\tau,F)$ be a τ-periodic switched system (without delays). The only difference is that we do not impose that the maximum delay δ_0 is smaller than the switching period τ. This is because in the proof of Lem. 4.1.4, we do not need this condition, while in the proof of Lem. 3.4.4, it has been used implicitly.

Then, we construct the same transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$ as we did in Def. 3.2.1. We assume that the dynamics has a common δ-GAS Lyapunov

function. Since the extension to the multiple δ-GAS Lyapunov function case can be done in the same way as in Chapter 3, we skip its details and just state the main theorem at the end of § 4.2. The main technical difference in this section from Chapter 3 is the definition of the premetric defined in Def. 3.2.2.

Definition 4.1.1 (timing discrepant premetric)**.** On the set of outputs $O = \mathbb{R}^n \times \mathbb{R}^+ \times P$ that is common to the two transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$, we define the following premetric d':

$$d'((x,t,p),(x',t',p')) := \begin{cases} \|x - x'\| & \text{if } p = p', t' = k\tau \text{ and} \\ & \quad t \in [t', t' + \delta_0] \text{ for some } k \in \mathbb{N} \\ \infty & \text{otherwise.} \end{cases}$$

Note that in this definition of premetric, the Euclidean metric between two states is taken as it is without adjusting the time of the two states.

As preparation to prove our main lemma and theorem, we redefine the following function V' from δ-GAS Lyapunov function V. In this section we use the following V' instead of the one we defined before in Def. 3.4.3

Definition 4.1.2 (the function V')**.** Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system, and let $V \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be a common δ-GAS Lyapunov function for Σ.

We define a function $V' : (\mathbb{R}^n \times \mathbb{R}^+ \times P) \times (\mathbb{R}^n \times \mathbb{R}^+ \times P) \to \mathbb{R}^+$ in the following manner:

$$V'\big((x,t,p),(x',t',p')\big) := \begin{cases} V(x,x') & \text{if } p = p' \text{ and } t \in [t', t' + \delta_0] \\ \infty & \text{otherwise.} \end{cases}$$

Now we prove our technical main lemma, which is an analogue of Lem. 3.4.4. In Lem. 3.4.4, we put Assumption 3.4.1 but in this section, we use the following assumption instead.

Assumption 4.1.3 (bounded partial derivative)**.** Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system, with $P = \{1, 2, \ldots, m\}$ and $F = \{f_1, f_2, \ldots, f_m\}$ being the set of vector fields associated with each mode. We say a function $V \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ has *bounded partial derivatives* if there exists a real number $\nu'' \geq 0$ such that, for any $p \in P$, the inequality

$$\left| \frac{\partial V}{\partial x}(x,y) f_p(x) \right| \leq \nu''$$

holds for each $x, y \in \mathbb{R}^n$.

Our main technical lemma is as follows.

Lemma 4.1.4. *Let $\Sigma_\tau = (\mathbb{R}^n, P, \mathbf{P}_\tau, F)$ be a τ-periodic switched system, and $\Sigma_{\tau,\delta_0} = (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F)$ be a τ-periodic switched system with delays within δ_0. Assume that there exists a common δ-GAS Lyapunov function V for Σ_τ, and that V satisfies the additional assumption in Assumption 4.1.3. Then, for a suitable g, there exists a g-incrementing approximate bisimulation $\{R_\varepsilon\}_{\varepsilon \geq 0}$ between the*

transition systems $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$, where the premetric on the transition systems is d' in Def. 4.1.1.

Specifically, we define a function g by

$$g(\varepsilon) := \underline{\alpha}^{-1}\left(e^{-\kappa\tau}\underline{\alpha}(\varepsilon) + \nu''\delta_0\right) \ ,$$

where $\underline{\alpha}$ and κ are from Def. 2.1.7 and ν'' is from Assumption 4.1.3. For each $\varepsilon \geq 0$, we define a relation $R_\varepsilon \subseteq (\mathbb{R}^n \times \mathbb{R}^+ \times P) \times (\mathbb{R}^n \times \mathbb{R}^+ \times P)$ by

$$(q, q') \in R_\varepsilon \overset{\text{def.}}{\iff} V'(q, q') \leq \underline{\alpha}(\varepsilon) \ . \tag{4.1}$$

Here V' is from Def. 4.1.2.

Proof. To prove that $\{R_\varepsilon\}_{\varepsilon\geq 0}$ is a g-incrementing approximate bisimulation, we need to prove the conditions in Def. 3.3.1. We omit Condition 2b again, as the proof of Lem. 3.4.4. Condition 1 can be proved in the same way as in the proof of Lem. 3.4.4, so we also omit it and only prove Condition 2a.

We assume that $q_{\tau,\delta_0} \xrightarrow[\tau,\,\delta_0]{p_\tau} q'_{\tau,\delta_0} = (\mathbf{x}(t'_{\tau,\delta_0} - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), t'_{\tau,\delta_0}, p'_{\tau,\delta_0})$ and $(q_{\tau,\delta_0}, q_\tau) \in R_\varepsilon$ for some ε. Then, we define q'_τ by $q'_\tau := (\mathbf{x}(\tau, x_\tau, p_\tau), t'_\tau, p'_\tau)$ where

$$\begin{aligned} t'_\tau &= (k+1)\tau, \text{ and} \\ p'_\tau &= p'_{\tau,\delta_0}. \end{aligned}$$

This definition of q'_τ guarantees $q_\tau \xrightarrow[\tau]{p_\tau} q'_\tau$. Now we show $(q'_{\tau,\delta_0}, q'_\tau) \in R_{g(\varepsilon)}$ for this q'_τ in the following manner.

$$\begin{aligned} & V'(q'_{\tau,\delta_0}, q'_\tau) \\ &= V(\mathbf{x}(t'_{\tau,\delta_0} - t_{\tau,\delta_0}, x_{\tau,\delta_0}, p_\tau), \mathbf{x}(\tau, x_\tau, p_\tau)) \\ &\leq e^{-\kappa\tau} V(x_{\tau,\delta_0}, x_\tau) + \nu''|t'_{\tau,\delta_0} - t_{\tau,\delta_0} - \tau| \\ &\leq e^{-\kappa\tau} V'(q_{\tau,\delta_0}, q_\tau) + \nu''\delta_0 \\ &\leq e^{-\kappa\tau}\underline{\alpha}(\varepsilon) + \nu''\delta_0 \\ &= \underline{\alpha}(g(\varepsilon)). \end{aligned}$$

Thus we have $(q'_{\tau,\delta_0}, q'_\tau) \in R_{g(\varepsilon)}$ and this proves Condition 2a in Def. 3.3.1. □

4.2 Upper Bound of Skorokhod Metric

In Lem. 4.1.4, we proved that the new premetric between states is bounded by ε. There is timing discrepancy between states. Therefore we cannot bound the errors between the two systems in the same way as Thm. 3.4.5. Instead, what we obtain is actually an upper bound of the Skorokhod metric. The following definitions are taken from [34] and adapted to our setting. In the definitions, let I be $\mathbb{R}^+$ or its closed interval. It is used as the domain of time.

First we define retiming functions.

Definition 4.2.1 (retiming)**.** A function $\mathsf{r} : I \to I$ is a *retiming* if it is order-preserving, bijective and continuous. The set of all retiming functions is denoted by R. The identity retiming is denoted by $\mathcal{I} \in \mathsf{R}$.

Then, we define the Skorokhod metric using the sup norm $\|\_\|_\infty$ on the set of retimings R.

Definition 4.2.2 (Skorokhod metric)**.** Let r be a retiming, and $\pi, \pi' : I \to \mathbb{R}^n$ be two trajectories. Note that $\|\mathsf{r} - \mathcal{I}\|_\infty = \sup_{t\in I} |\mathsf{r}(t) - t|$, and that $\|\pi \circ \mathsf{r} - \pi'\|_\infty = \sup_{t\in I} \|\pi(\mathsf{r}(t)) - \pi'(t)\|$. Here, $\|\_\|$ on $\mathbb{R}^n$ is the usual Euclidean norm.

The *Skorokhod distance* between the trajectories π and π' is defined by

$$\mathcal{D}_{\mathcal{S}}(\pi, \pi') := \inf_{\mathsf{r}\in\mathsf{R}} \max\left(\|\mathsf{r} - \mathcal{I}\|_\infty, \|\pi \circ \mathsf{r} - \pi'\|_\infty\right).$$

The transference of temporal specifications enables one of the most important application of the Skorokhod metric—the application to conformance testing. In this thesis, we do not refer the full transference theorem for temporal specifications in [34], since we only use the Skorokhod metric for reachability analysis, not complicated temporal properties. For reachability, the following obvious proposition is enough.

Proposition 4.2.3. *Let Σ and Σ' be switched systems. Assume that for every trajectory π of Σ, there exists a trajectory π' of Σ' such that $\mathcal{D}_{\mathcal{S}}(\pi, \pi') \le \varepsilon$. Then, the reachable set of Σ is included in the ε-expansion $E_\varepsilon(S)$ of the reachable set S of Σ', where the ε-expansion $E_\varepsilon(S)$ is $\{x \in \mathbb{R}^n \mid$ there exists $y \in S$ such that $\|x - y\| \le \varepsilon\}$.* □

The following is our main theorem. It ensures that we can compute an over-approximation of the Skorokhod distance using the approximate bisimulation relation given in Lem. 4.1.4.

Theorem 4.2.4. *Assume the same assumptions as in Lem. 4.1.4. Let $\mathbf{p}_\tau$ be a τ-periodic switching signal, and $\mathbf{p}_{\tau,\delta_0}$ be the same signal but with delays within δ_0. That is, for each $s \in \mathbb{R}^+$,*

$$\mathbf{p}_{\tau,\delta_0}(s) = \begin{cases} \mathbf{p}_\tau(s) \text{ or } \mathbf{p}_\tau(s-\delta_0) & \text{if } s \in \bigcup_{k\in\mathbb{N},k\ge 1}[k\tau, k\tau+\delta_0) \\ \mathbf{p}_\tau(s) & \text{otherwise.} \end{cases}$$

Given a state $x \in \mathbb{R}^n$, we define two trajectories $\pi_{\tau,\delta_0,x}, \pi_{\tau,x} : \mathbb{R}^+ \to \mathbb{R}^n$ by

$$\pi_{\tau,\delta_0,x}(t) := \mathbf{x}(t, x, \mathbf{p}_{\tau,\delta_0}), \text{ and}$$
$$\pi_{\tau,x}(t) := \mathbf{x}(t, x, \mathbf{p}_\tau).$$

We also consider their restrictions to a closed interval I, i.e., $\pi_{\tau,\delta_0,x}|_I, \pi_{\tau,x}|_I : I \to \mathbb{R}^n$.

Then, we obtain an upper bound of the Skorokhod distance $\mathcal{D}_{\mathcal{S}}(\pi_{\tau,\delta_0,x}, \pi_{\tau,x})$ in the following way.

(a) We have, for each $k \in \mathbb{N}$ and $t \in [k\tau, (k+1)\tau)$,

$$\begin{aligned} &\mathcal{D}_{\mathcal{S}}(\pi_{\tau,\delta_0,x}|_{[0,t]}, \pi_{\tau,x}|_{[0,t]}) \\ &\le \max\left(\delta_0, \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{1-e^{-\kappa\tau}} + e^{-\kappa\tau k}\left(-\frac{\nu''\delta_0}{1-e^{-\kappa\tau}}\right)\right)\right). \end{aligned}$$

(b) We have, for each $t \in \mathbb{R}^+$,

$$\mathcal{D}_{\mathcal{S}}(\pi_{\tau,\delta_0,x}, \pi_{\tau,x}) \leq \max\left(\delta_0, \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{1-e^{-\kappa\tau}}\right)\right) .$$

Proof. Note that $\mathbf{p}_\tau$ is a τ-periodic switching signal, and $\mathbf{p}_{\tau,\delta_0}$ is the same signal but with delays within δ_0. For $k \in \mathbb{N}$, the k-th switching of $\mathbf{p}_\tau$ occurs at $t = k\tau$. The k-th switching time of $\mathbf{p}_{\tau,\delta_0}$ is denoted by $\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k)$. We define a retiming r as follows: for every $k \in \mathbb{N}$ and $t \in [0, \tau)$,

$$\mathsf{r}(k\tau + t) = \frac{(\tau - t)\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k) + t\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1)}{\tau}. \tag{4.2}$$

Intuitively, this retiming r adjusts each switching time of the periodic signal to that with delays, and the intervals between switchings are lengthened or shortened uniformly. It is easy to check that this r is order-preserving, bijective and continuous.

For this r, we have

$$\|\mathsf{r} - \mathcal{I}\|_\infty \leq \delta_0. \tag{4.3}$$

Then, our next goal is to show that for every $k \in \mathbb{N}$,

$$\begin{aligned}&\sup_{t\in[k\tau,(k+1)\tau]} \|\pi_{\tau,\delta_0,x}(\mathsf{r}(t)) - \pi_{\tau,x}(t)\| \\ \leq &\max\left(\underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{1-e^{-\kappa\tau}} + e^{-\kappa\tau k}\left(-\frac{\nu''\delta_0}{1-e^{-\kappa\tau}}\right)\right)\right).\end{aligned} \tag{4.4}$$

In the similar way as the proof of Thm. 3.4.5, using the result of Lem. 4.1.4 as a recurrence relation, it is not hard to see that

$$\|\pi_{\tau,\delta_0,x}(\mathsf{r}(k\tau)) - \pi_{\tau,x}(k\tau)\| \leq \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{1-e^{-\kappa\tau}} + e^{-\kappa\tau k}\left(-\frac{\nu''\delta_0}{1-e^{-\kappa\tau}}\right)\right). \tag{4.5}$$

Note that $\mathsf{r}(k\tau) = \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k)$.

We can see from (4.2) that in $t \in [k\tau, (k+1)\tau]$, the application of r quickens or slows down time progress uniformly by multiplying $\frac{\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1) - \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k)}{\tau}$. In other words, in $t \in [k\tau, (k+1)\tau]$, $\frac{d\mathsf{r}(t)}{dt} = \frac{\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1) - \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k)}{\tau}$.

This means that after the application of r, the trajectory $x = \pi_{\tau,\delta_0,x} \circ \mathsf{r}$ follows

$$\begin{aligned}\dot{x} = \dot{(\pi_{\tau,\delta_0,x} \circ \mathsf{r})} &= \frac{d\pi_{\tau,\delta_0,x}(\mathsf{r}(t))}{d\mathsf{r}(t)}\frac{d\mathsf{r}(t)}{dt} \\ &= \frac{\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1) - \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k)}{\tau} f_p(x),\end{aligned} \tag{4.6}$$

where p is the mode after k-th switching. By (2.4) and (4.6), we have

$$\begin{aligned}&\frac{\partial V}{\partial x}(x,y)\dot{(\pi_{\tau,\delta_0,x} \circ \mathsf{r})} + \frac{\partial V}{\partial y}(x,y)f_p(y) \\ \leq &-\kappa V(x,y) + \frac{\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1) - \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k) - \tau}{\tau}\frac{\partial V}{\partial x}(x,y)f_p(x).\end{aligned}$$

Using $\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k) \in [k\tau, k\tau+\delta_0]$ and $\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1) \in [(k+1)\tau, (k+1)\tau+\delta_0]$, we have

$$\begin{aligned}&\frac{\partial V}{\partial x}(x,y)(\dot{\pi}_{\tau,\delta_0,x} \circ \mathsf{r}) + \frac{\partial V}{\partial y}(x,y) f_p(y) \\ \leq& -\kappa V(x,y) + \frac{\delta_0}{\tau}\left|\frac{\partial V}{\partial x}(x,y) f_p(x)\right|.\end{aligned}$$

Using Assumption 4.1.3, we can say that

$$\frac{\partial V}{\partial x}(x,y)(\dot{\pi}_{\tau,\delta_0,x} \circ \mathsf{r}) + \frac{\partial V}{\partial y}(x,y) f_p(y) \leq -\kappa V(x,y) + \frac{\delta_0}{\tau}\nu''.$$

We can see that the RHS of this inequality is negative when $V(x,y) \geq \frac{\nu''\delta_0}{\kappa\tau}$.

By combining this result with (4.5), we obtain (4.4) as desired. Thus, by (4.3) and (4.4), we have

$$\begin{aligned}&\mathcal{D}_{\mathcal{S}}(\pi_{\tau,\delta_0,x}|_{[\mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k), \mathsf{st}_{\mathbf{p}_{\tau,\delta_0}}(k+1)]}, \pi_{\tau,x}|_{[k\tau,(k+1)\tau]}) \\ \leq& \max\left(\delta_0, \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''\delta_0}{1-e^{-\kappa\tau}} + e^{-\kappa\tau k}\left(-\frac{\nu''\delta_0}{1-e^{-\kappa\tau}}\right)\right)\right).\end{aligned}$$

Since this inequality holds for all $k \in \mathbb{N}$, the statement (a) of this theorem is proved.

The proof of (b) is straightforward. □

So far in this section we have assumed a common Lyapunov function. We can easily extend the setting to multiple Lyapunov functions in a similar way as §3.5, so we skip its details and present only the main theorem.

First, the following assumption is an analogue of Assumption 4.1.3.

Assumption 4.2.5. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a switched system with $P = \{1, 2, \ldots, m\}$. Let $V_1, \ldots, V_m \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^+$ be smooth functions. We assume that the functions $V_1, \ldots, V_m$ satisfy the following condition: there exists two real numbers $\nu''' \geq 0$ and $\nu'''' \geq 0$ such that, for each $p, p' \in P$ that are distinct, the inequalities

$$\begin{aligned}&\frac{\partial V_p}{\partial x}(x,y) f_p(x) \leq \nu''', \text{ and} \\ &\left|\frac{\partial V_p}{\partial y}(x,y) f_{p'}(y)\right| \leq \nu''''\end{aligned}$$

hold for each $x, y \in \mathbb{R}^n$.

Then, the main theorem for multiple Lyapunov function case is the following. The proof is omitted because it is almost the same as Thm. 4.2.4.

Theorem 4.2.6. *Let $\Sigma_\tau = (\mathbb{R}^n, P, \mathbf{P}_\tau, F)$ be a τ-periodic switched system and $\Sigma_{\tau,\delta_0} = (\mathbb{R}^n, P, \mathbf{P}_{\tau,\delta_0}, F)$ be a τ-periodic switched system with delays within δ_0. Assume that for each $p \in P$, there is a δ-GAS Lyapunov function V_p for the single-mode subsystem $\Sigma_{\tau,p}$. We additionally assume Assumption 4.2.5 for $V_1, \ldots, V_m$, and that there exists $\mu \in \mathbb{R}^+$ such that*

$$V_p(x,y) \leq \mu V_{p'}(x,y) \text{ for any } x, y \in \mathbb{R}^n \text{ and } p, p' \in P\ . \tag{4.7}$$

The last assumption is the same as in Thm. 2.1.9.

Let $\mathbf{p}_\tau$ be a τ-periodic switching signal, and $\mathbf{p}_{\tau,\delta_0}$ be the same signal but with delays within δ_0. Given a state $x \in \mathbb{R}^n$, we define two trajectories $\pi_{\tau,\delta_0,x}, \pi_{\tau,x} : \mathbb{R}^+ \to \mathbb{R}^n$ by

$$\pi_{\tau,\delta_0,x}(t) = \mathbf{x}(t, x, \mathbf{p}_{\tau,\delta_0}), \text{ and}$$
$$\pi_{\tau,x}(t) = \mathbf{x}(t, x, \mathbf{p}_\tau).$$

We also consider their restrictions to a closed interval I, i.e., $\pi_{\tau,\delta_0,x}|_I, \pi_{\tau,x}|_I : I \to \mathbb{R}^n$.

Then, we obtain an upper bound of the Skorokhod distance $\mathcal{D}_\mathcal{S}(\pi_{\tau,\delta_0,x}, \pi_{\tau,x})$ in the following way.

(a) We have, for each $k \in \mathbb{N}$ and $t \in [k\tau, (k+1)\tau)$,

$$\begin{aligned}&\mathcal{D}_\mathcal{S}(\pi_{\tau,\delta_0,x}|_{[0,t]}, \pi_{\tau,x}|_{[0,t]})\\ &\leq \max\left(\delta_0, \underline{\alpha}^{-1}\left(\frac{\nu'''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''''\delta_0}{1-\mu e^{-\kappa\tau}} + \mu e^{-\kappa\tau k}\left(-\frac{\nu''''\delta_0}{1-\mu e^{-\kappa\tau}}\right)\right)\right) .\end{aligned}$$

(b) If $\tau > \frac{\log\mu}{\kappa}$, we have, for each $t \in \mathbb{R}^+$,

$$\mathcal{D}_\mathcal{S}(\pi_{\tau,\delta_0,x}, \pi_{\tau,x}) \leq \max\left(\delta_0, \underline{\alpha}^{-1}\left(\frac{\nu'''\delta_0}{\kappa\tau}\right), \underline{\alpha}^{-1}\left(\frac{\nu''''\delta_0}{1-\mu e^{-\kappa\tau}}\right)\right) . \quad \square$$

4.3 Examples

In this section we use the same examples as in §3.6 to show how precise our extended framework makes the analysis compared to the one introduced in Chapter 3.

4.3.1 Boost DC-DC Converter

The first example is the boost DC-DC converter. The detailed system description is in §3.6.1. As in §3.6.1, we analyze the rescaled version of the system.

We will use the same common δ-GAS Lyapunov function $V(x,y) = \sqrt{(x-y)^T M(x-y)}$ as §3.6.1, where $M = \begin{bmatrix}1.0224 & 0.0084\\ 0.0084 & 1.0031\end{bmatrix}$. It has $\underline{\alpha}(s) = s, \overline{\alpha}(s) = 1.0127s$ and $\kappa = 0.014$. We set the switching period $\tau = 0.5$ and the maximum delay $\delta_0 = \frac{\tau}{1000}$. It is not hard to see that for a safe region $\mathcal{S} := [1.3, 1.7] \times [5.7, 5.8]$, $\nu'' = 0.33$ satisfies Assumption 4.1.3).

By Thm. 4.2.4, we obtain that the Skorokhod distance between Σ_{τ,δ_0} (the boost DC-DC converter with delays) and Σ_τ (the one without delays) is bounded by $\varepsilon = 0.023655$, which is 20 percent smaller than the error bound $\varepsilon = 0.0294176$.

4.3.2 Nonlinear Water Tank

The second example is the nonlinear water tank. The detailed system description is in §3.6.2.

We use the same δ-GAS Lyapunov functions as in §3.6.2, which are

$$V_{OFF}(x,y) := |e^{\sqrt{x}} - e^{\sqrt{y}}| \quad \text{and} \quad V_{ON}(x,y) := |\sqrt{6}(x-y)| .$$

We obtain the following characteristics for these two δ-GAS Lyapunov functions in the safe region $[1, 10]$: $\underline{\alpha}_{OFF}(s) = s$, $\underline{\alpha}_{ON}(s) = \sqrt{6}s$, $\kappa_{OFF} = \kappa_{ON} = \frac{1}{10}$, and $\mu = \frac{2\sqrt{6}}{3}$.

The new constants we need to find are ν''' and ν'''' in Assumption 4.2.5. It is not hard to see that $\nu''' = 2.45$ and $\nu''' = 1.55$ satisfy Assumption 4.2.5.

By Thm. 4.2.6, for $\tau = 10$ and $\delta_0 = 0.1$, we obtain that the Skorokhod distance between Σ_{τ,δ_0} and Σ_τ is bounded by $\varepsilon = 0.388234$, which is almost the half of the error bound $\varepsilon = 0.747678$ found in §3.6.

Chapter 5

Extension of Abstract Interpretation with Infinitesimals

In this chapter, we extend abstract interpretation for the purpose of reachability analysis of hybrid systems (without delays). It is well-known that the reachability problem of hybrid systems is undecidable (even for linear ones) [6]. Therefore, overapproximation has been playing an important role in many verification methodologies for hybrid systems. Compared to the overapproximation of the reachable set of discrete programs, that of hybrid systems usually needs some special care to cope with continuous dynamics defined by ODEs. We apply Cousot and Cousot's framework of abstract interpretation to hybrid systems, almost *as it is*, by regarding continuous dynamics as an infinite iteration of *infinitesimal* discrete jumps. This extension follows the line of work by Suenaga, Hasuo and Sekine [50, 91, 92], where deductive verification is extended for hybrid systems by 1) introducing a constant `dt` for an infinitesimal value; and 2) employing Robinson's *nonstandard analysis (NSA)* to define mathematically rigorous semantics. Our theoretical results include soundness and termination via *uniform* widening operators. Our prototype implementation successfully verifies some examples.

In §5.1, we start with an explanation of the modeling language While$^{\mathtt{dt}}$ introduced in [91]. It is an extension of a usual imperative language with a constant for an infinitesimal. In §5.2 we extend the theory of abstract interpretation with infinitesimals and build the theory of nonstandard abstract interpretation. Its theorems include soundness of approximation, and termination guaranteed by (the $*$-transform of) a *uniform* widening operator. In §5.3, we present how our nonstandard abstract interpretation framework works using the linear water tank example. In §5.4, we present our prototype implementation and the experimental results.

5.1 The Modeling Language While$^{\mathtt{dt}}$

While$^{\mathtt{dt}}$, a modeling language for hybrid systems based on NSA, was introduced in [91]. It is an augmentation of a usual imperative language (such as **IMP** in [95]) with a constant `dt` that expresses an infinitesimal number. In the following definition of While$^{\mathtt{dt}}$ syntax, we add a command "`if` $*$ `then` c_1 `else` c_2" for nondeterministic branching to the syntax introduced in [91]. It will be used in Code 5.2.

```
1  /*Linear Water Tank*/
2  l := 0; x := 1; p := 1; s := 0;
3  while true do {
4    if p = 1 then x := x + dt
5    else x := x - 2 * dt;
6    if (x <= 5 && p = 0) then s := 1
7    else {
8      if (x >= 10 && p = 1) then s := 1
9      else s := 0
10   };
11   if s = 1 then l := l + dt
12   else skip;
13   if s = 1 && l >= 2 then {p := 1 - p; s := 0; l := 0}
14   else skip
15 }
```

Code 5.1: Linear water tank in WHILE$^{\mathtt{dt}}$

Definition 5.1.1. Let **Var** be the set of variables. The syntax of WHILE$^{\mathtt{dt}}$ is as follows:

$$\begin{aligned}
\mathbf{AExp} \ni a ::= {} & x \mid r \mid a_1 \ \mathtt{aop} \ a_2 \mid \mathtt{dt} \\
& \text{where } x \in \mathbf{Var}, r \in \mathbb{R} \text{ and } \mathtt{aop} \in \{+, -, \cdot, \hat{}\} \\
\mathbf{BExp} \ni b ::= {} & \mathtt{true} \mid \mathtt{false} \mid b_1 \wedge b_2 \mid \neg b \mid a_1 < a_2 \\
\mathbf{Cmd} \ni c ::= {} & \mathtt{skip} \mid x := a \mid c_1; c_2 \mid \mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \\
& \mid \mathtt{if}\ *\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2 \mid \mathtt{while}\ b\ \mathtt{do}\ c.
\end{aligned}$$

An expression $a \in \mathbf{AExp}$ is an *arithmetic expression*, $b \in \mathbf{BExp}$ is a *Boolean expression* and $c \in \mathbf{Cmd}$ is a *command*.

The infinitesimal constant `dt` enables us to model not only discrete dynamics but also continuous dynamics without explicit ODEs. For example, the example of linear water tank [6] is modeled as a WHILE$^{\mathtt{dt}}$ program shown in Code 5.1. The analysis of this program will be presented later in §5.3.

The continuous dynamics modeled in this example is piecewise-linear. Even dynamics defined by nonlinear ODEs can be modeled in WHILE$^{\mathtt{dt}}$ in the same manner.

In the usual, standard abstract interpretation (without `dt`), a command c is assigned its *collecting semantics* $\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to \mathbb{R})$ (see e.g. [28]) as the concrete semantics. This is semantics by reachable sets of memory states, since a memory state is a function from the set of variables **Var** to $\mathbb{R}$. Presence of `dt` in the syntax of WHILE$^{\mathtt{dt}}$ calls for an infinitesimal number in the picture. The first thing to try would be to replace $\mathbb{R}$ with ${}^*\mathbb{R}$, and interpret WHILE$^{\mathtt{dt}}$ commands as functions of the type $\mathcal{P}(\mathbf{Var} \to {}^*\mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to {}^*\mathbb{R})$. This however is not suited for the purpose of interpreting recursion in presence of `dt`.[1] We rely instead on our theory of *hyperdomains* that is used in [92] and described

[1] If we interpret commands as functions $\mathcal{P}(\mathbf{Var} \to {}^*\mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to {}^*\mathbb{R})$, the interpretation $[\![\mathtt{while}\ x < 10\ \mathtt{do}\ x := x + \mathtt{dt}]\!]\{(x \mapsto 0)\}$ by a least fixed point will be $\{x \mapsto r \mid \exists n \in \mathbb{N}.\ r = n * \mathtt{dt}\}$, not $\{x \mapsto r \mid \exists n \in {}^*\mathbb{N}.\ r = n * \mathtt{dt} \wedge r \leq 10\}$ as we expect. The problem is that *internality*—an "well-behavedness" notion in NSA—is not preserved in such a modeling.

$$[\![x]\!]\boldsymbol{\sigma} := \boldsymbol{\sigma}(x) \text{ for each } x \in \mathbf{Var} \qquad [\![\mathtt{true}]\!]\boldsymbol{\sigma} := \mathrm{tt}$$
$$[\![r]\!]\boldsymbol{\sigma} := r \text{ for each } r \in \mathbb{R} \qquad [\![\mathtt{false}]\!]\boldsymbol{\sigma} := \mathrm{ff}$$
$$[\![a_1 \mathtt{\ aop\ } a_2]\!]\boldsymbol{\sigma} := [\![a_1]\!] \mathtt{\ aop\ } [\![a_2]\!] \qquad [\![b_1 \wedge b_2]\!]\boldsymbol{\sigma} := [\![b_1]\!] \wedge [\![b_2]\!]$$
$$[\![\mathtt{dt}]\!]\boldsymbol{\sigma} := [(1, \tfrac{1}{2}, \tfrac{1}{3}, \cdots)] \qquad [\![\neg b]\!]\boldsymbol{\sigma} := \neg([\![b]\!]\boldsymbol{\sigma})$$
$$[\![a_1 < a_2]\!]\boldsymbol{\sigma} := [\![a_1]\!] < [\![a_2]\!]$$

$$[\![\mathtt{skip}]\!]\mathbf{S} := \mathbf{S}$$
$$[\![x := a]\!]\mathbf{S} := \{\boldsymbol{\sigma}[[\![a]\!]\boldsymbol{\sigma}/x] \mid \boldsymbol{\sigma} \in \mathbf{S}\}$$
$$[\![c_1; c_2]\!]\mathbf{S} := [\![c_2]\!]([\![c_1]\!]\mathbf{S})$$
$$[\![\mathtt{if}\ b\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\mathbf{S} := \begin{array}{l}\{[\![c_1]\!]\boldsymbol{\sigma} \mid \boldsymbol{\sigma} \in \mathbf{S},\ [\![b]\!]\boldsymbol{\sigma} = \mathrm{tt}\} \\ \cup\, \{[\![c_2]\!]\boldsymbol{\sigma} \mid \boldsymbol{\sigma} \in \mathbf{S},\ [\![b]\!]\boldsymbol{\sigma} = \mathrm{ff}\}\end{array}$$
$$[\![\mathtt{if}\ *\ \mathtt{then}\ c_1\ \mathtt{else}\ c_2]\!]\mathbf{S} := \{[\![c_1]\!]\boldsymbol{\sigma} \mid \boldsymbol{\sigma} \in \mathbf{S}\} \cup \{[\![c_2]\!]\boldsymbol{\sigma} \mid \boldsymbol{\sigma} \in \mathbf{S}\}$$
$$[\![\mathtt{while}\ b\ \mathtt{do}\ c]\!] := {}^*\mathrm{lfp}\big({}^*\Phi\,([\![b]\!])\,([\![c]\!])\big)$$
where $\Phi : (\mathbf{St} \to \mathbb{B} \cup \{\bot\}) \to \big(\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to \mathbb{R})\big) \to \Big(\big(\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to \mathbb{R})\big) \to \big(\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to \mathbb{R})\big)\Big)$ is defined by
$$\Phi(f)(g) = \lambda\psi.\ \lambda S.\ S \cup \psi\{(g(\sigma)) \mid \sigma \in S,\ f(\sigma) = \mathrm{tt}\} \cup \{\sigma \mid \sigma \in S,\ f(\sigma) = \mathrm{ff}\}.$$

Table 5.1: WHILE$^{\mathtt{dt}}$ collecting semantics

in §2.2.2 ; see the interpretation of while loops in Table 5.1. This calls for the interpretation of commands to be of the type ${}^*\big(\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to \mathcal{P}(\mathbf{Var} \to \mathbb{R})\big)$, a subset of ${}^*\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to {}^*\mathcal{P}(\mathbf{Var} \to \mathbb{R})$. The last type will be used in the following definition.

Definition 5.1.2. *Collecting semantics* for WHILE$^{\mathtt{dt}}$, in Table 5.1, has the following types where $\mathbb{B}$ is $\{\mathrm{tt}, \mathrm{ff}\}$: $[\![a]\!] : {}^*(\mathbf{Var} \to \mathbb{R}) \to {}^*\mathbb{R}$ for $a \in \mathbf{AExp}$; $[\![b]\!] : {}^*(\mathbf{Var} \to \mathbb{R}) \to \mathbb{B}$ for $b \in \mathbf{BExp}$; and $[\![c]\!] : {}^*\mathcal{P}(\mathbf{Var} \to \mathbb{R}) \to {}^*\mathcal{P}(\mathbf{Var} \to \mathbb{R})$ for $c \in \mathbf{Cmd}$.

In [91], the semantics of a while loop is defined using the idea of sectionwise execution, instead of as a least fixed point. This is not suited for employing abstract interpretation—the latter is after all for computing least fixed points. The collecting semantics in Def. 5.1.2 (Table 5.1) does use least fixed points; it is based on the alternative WHILE$^{\mathtt{dt}}$ semantics introduced in [63]. In the definition of this alternative WHILE$^{\mathtt{dt}}$ semantics, Lem. 2.2.18 justifies the use of ${}^*\mathrm{lfp}$ as the least fixed point operator. The equivalence of the two semantics is also established in [63].

In what follows, we restrict the set of variables **Var** to be finite. This assumption—a realistic one when we focus on the program to be analyzed—makes our NSA framework much simpler. Therefore $\mathcal{P}(\mathbf{Var} \to \mathbb{R})$ and ${}^*\mathcal{P}(\mathbf{Var} \to \mathbb{R})$

are equal to $\mathcal{P}(\mathbb{R}^n)$ and ${}^*\mathcal{P}(\mathbb{R}^n)$ for some $n \in \mathbb{N}$ respectively; we prefer the latter notations hereafter. This enables us to work on the superstructure $\mathbb{U} = V(\mathbb{R})$, instead of $V(\mathbb{R} \cup \mathbb{B} \cup \mathbf{Var})$ used in §2.2.2.

5.2 Abstract Interpretation Augmented with Infinitesimals

In this section, as our main theoretical contribution, a metatheory of *nonstandard abstract interpretation* that justifies the workflow in §5.3 is described. (Standard) abstract interpretation infrastructure such as Prop. 2.2.20 and Prop. 2.2.22 is not applicable to WHILE$^{\mathtt{dt}}$ programs, since ${}^*\mathcal{P}(\mathbb{R}^n)$ is not a cpo. One can see that the ascending chain defined by $X_n := \{k * \mathtt{dt} \mid 0 \leq k \leq n\}$ does not have the supremum in ${}^*\mathcal{P}(\mathbb{R}^n)$ since $\{k * \mathtt{dt} \mid k \in \mathbb{N}\}$ is not internal (see § 2.2.1). Thus, we now extend the abstract interpretation framework for the analysis of WHILE$^{\mathtt{dt}}$ programs (and the hybrid systems modeled thereby). We introduce an *abstract hyperdomain* over ${}^*\mathbb{R}$ as the transfer of the (standard, over $\mathbb{R}$) domain of convex polyhedra. We then interpret WHILE$^{\mathtt{dt}}$ programs in it, and transfer the three widening operators mentioned in §2.2.3 to the nonstandard setting. We classify them into *uniform* ones—for which termination is guaranteed even in the nonstandard setting—and non-uniform ones. The main theorems are Thm. 5.2.4 and Thm. 5.2.10, for soundness (in place of Prop. 2.2.20) and termination (in place of Prop. 2.2.22) respectively.

5.2.1 The Domain of Convex Polyhedra over Hyperreals

We extend convex polyhedra to the current nonstandard setting.

Definition 5.2.1 (convex polyhedra over ${}^*\mathbb{R}$)**.** A *convex polyhedron* on $({}^*\mathbb{R})^n$ is an intersection of finite number of affine half-spaces on $({}^*\mathbb{R})^n$, that is, the set of points $\mathbf{x} \in ({}^*\mathbb{R})^n$ that satisfy a certain finite set of linear inequalities. The set of all convex polyhedra on $({}^*\mathbb{R})^n$ is denoted by $\mathbb{CP}_n^{{}^*\mathbb{R}}$.

Proposition 5.2.2. *The set $\mathbb{CP}_n^{{}^*\mathbb{R}}$ of all convex polyhedra over $({}^*\mathbb{R})^n$ is a (proper) subset of ${}^*\mathbb{CP}_n$, the $*$-transform of the (standard) domain of convex polyhedra over $\mathbb{R}^n$.*

Proof. The constraint system C for a (standard) convex polyhedron $P \in \mathbb{CP}_n$ can be expressed by a pair $(\mathbf{A}, \mathbf{b})$ of an $m \times n$-matrix $\mathbf{A}$ and an m-vector $\mathbf{b}$, where m is the number of linear inequalities in C. The same applies to a (nonstandard) convex polyhedron $P \in \mathbb{CP}_n^{{}^*\mathbb{R}}$. For each of $X \in \{\mathbb{R}, {}^*\mathbb{R}\}$, let us denote, by $\mathsf{Constr}_{X,m,n}$, the set of all convex polyhedra over X^n that can be expressed with m linear inequalities.

Then $\mathbb{CP}_n = \bigcup_{m \in \mathbb{N}} \mathsf{Constr}_{\mathbb{R},m,n}$ (with $\bigcup_{m \in \mathbb{N}}$ expressed using an existential quantifier $\exists m \in \mathbb{N}$) is a valid $\mathscr{L}_{\mathbb{R}}$-sentence by Def. 2.2.23. By the transfer principle (Lem. 2.2.8), we have a valid $\mathscr{L}_{{}^*\mathbb{R}}$-sentence ${}^*(\mathbb{CP}_n) = \bigcup_{m \in {}^*\mathbb{N}} \mathsf{Constr}_{{}^*\mathbb{R},m,n}$. It has as its subset the set $\mathbb{CP}_n^{{}^*\mathbb{R}} = \bigcup_{m \in \mathbb{N}} \mathsf{Constr}_{{}^*\mathbb{R},m,n}$ since $\mathbb{N} \subseteq {}^*\mathbb{N}$. This proves the claim. □

What lies in the difference between the two sets $\mathbb{CP}_n^{{}^*\mathbb{R}} \subsetneq {}^*\mathbb{CP}_n$ is, for example, a disk as a subset of $\mathbb{R}^2$ (hence of ${}^*\mathbb{R}^2$). In ${}^*\mathbb{CP}_2$ one can use a constraint system

whose number of linear constraints is a hypernatural number $m \in {}^*\mathbb{N}$; using e.g. $m = \omega = [(0, 1, 2, \dots)]$ allows us to approximate a disk with progressive precision.

In the following development of nonstandard abstract interpretation, we will use ${}^*\mathbb{CP}_n$ as an abstract domain since it allows transfer of properties of $\mathbb{CP}_n$. We note, however, that our overapproximation of the interpretation $[\![c]\!]$ of a loop-free WHILE$^{\mathsf{dt}}$ program c is always given in $\mathbb{CP}_n^{{}^*\mathbb{R}}$, i.e. with finitely many linear inequalities.

5.2.2 Theory of Nonstandard Abstract Interpretation

Our goal is to overapproximate the collecting semantics for WHILE$^{\mathsf{dt}}$ programs (Table 5.1) on convex polyhedra over ${}^*\mathbb{R}$. As we mentioned at the beginning of this section, however, abstract interpretation infrastructure cannot be applied since ${}^*\mathcal{P}(\mathbb{R}^n)$ is not a cpo. Fortunately it turns out that we can rely on the $*$-*transform* (§2.2.1) of the theory in §2.2.3, where it suffices to impose the cpo structure only on $\mathcal{P}(\mathbb{R})$ and the $*$-*continuity*—instead of the (standard) continuity—on the function $[\![c]\!]$. This theoretical framework of *nonstandard abstract interpretation*, which we shall describe here, is an extension of the transferred domain theory in § 2.2.2.

In the proofs of the results in this section, we will use the following notations in addition to those defined in Def. 2.2.15.

Definition 5.2.3. We define the following $\mathscr{L}_{\mathbb{R}}$-formulas:

$$
\begin{aligned}
&\mathsf{Concr}_{L_1,\sqsubseteq_1,L_2,\sqsubseteq_2,\gamma} :\equiv\ \forall \overline{x}, \overline{y} \in L_2.\ \overline{x} \sqsubseteq_2 \overline{y} \Rightarrow \gamma(\overline{x}) \sqsubseteq_1 \gamma(\overline{y})\\
&\mathsf{Monotone}_{L_1,\sqsubseteq_1,L_2,\sqsubseteq_2}(f) :\equiv\ \forall x, y \in L_1.\ x \sqsubseteq_1 y \Rightarrow f(x) \sqsubseteq_2 f(y)\\
&\mathsf{Basis}_{L,\sqsubseteq}(\bot, f) :\equiv\ \bot \sqsubseteq f(\bot)\\
&\mathsf{Cover}_{L,\sqsubseteq,\nabla} :\equiv\ \forall x, y \in L.\ (x \sqsubseteq x \nabla y) \wedge y \sqsubseteq x \nabla y)\\
&\mathsf{Term}_{L,\sqsubseteq,\nabla} :\equiv\ \forall x \in \mathbb{N} \to L.\ \mathsf{AscCn}(x) \Rightarrow\\
&\quad \Big(\forall y \in \mathbb{N} \to L.\ \Big(\big(y(0) = x(0) \wedge \forall n \in \mathbb{N}.\ y(n+1) = y(n) \nabla x(n+1)\big)\\
&\qquad\qquad \Rightarrow \exists k \in \mathbb{N}.\ y(k) = y(k+1)\Big)\Big)\\
&\mathsf{Widen}_{L,\sqsubseteq,\nabla} :\equiv\ \mathsf{Cover}_{L,\sqsubseteq,\nabla} \wedge \mathsf{Term}_{L,\sqsubseteq,\nabla}\\
&\mathsf{WidenSeq}_{L,\sqsubseteq,\nabla}(X, \bot, F) :\equiv\\
&\quad X(0) = \bot\ \wedge\ \forall n \in \mathbb{N}.\ X(n+1) = X(n) \nabla F(X(n)).
\end{aligned}
$$

Then, the following theorem ensures the conservative approximation.

Theorem 5.2.4. *Let $(L, \sqsubseteq)$ be a cpo; $F : {}^*L \to {}^*L$ be a *-continuous function; and $\bot \in {}^*L$ be such that $\bot\ {}^*\!\sqsubseteq\ F(\bot)$. Let $(\overline{L}, \overline{\sqsubseteq})$ be a preorder; $\gamma : \overline{L} \to L$ be a function such that $\overline{a} \mathrel{\overline{\sqsubseteq}} \overline{b} \Rightarrow \gamma(\overline{a}) \sqsubseteq \gamma(\overline{b})$ for all $\overline{a}, \overline{b} \in \overline{L}$; and $\overline{F} : {}^*\overline{L} \to {}^*\overline{L}$ be a *-continuous function that is monotone with respect to ${}^*\overline{\sqsubseteq}$ and satisfies $F \circ {}^*\gamma\ {}^*\!\sqsubseteq\ {}^*\gamma \circ \overline{F}$. Note that $({}^*\overline{L}, {}^*\overline{\sqsubseteq})$ is also a preorder. Assume further that $\overline{x} \in {}^*\overline{L}$ is a prefixed point of $\overline{F}$ (i.e. $\overline{F}(\overline{x})\ {}^*\overline{\sqsubseteq}\ \overline{x}$) such that $\bot\ {}^*\!\sqsubseteq\ {}^*\gamma(\overline{x})$.*

Then $\overline{x}$ overapproximates $\mathrm{lfp}_{\bot} F$, that is, $\mathrm{lfp}_{\bot} F\ {}^*\!\sqsubseteq\ {}^*\gamma(\overline{x})$.

Proof. Let $L, \overline{L} \in \mathbb{U}$ be sets, $\sqsubseteq \in \mathcal{P}(L \times L)$ and $\overline{\sqsubseteq} \in \mathcal{P}(\overline{L} \times \overline{L})$ be binary relations on L and $\overline{L}$ respectively, $\alpha : L \to \overline{L}$ and $\gamma : \overline{L} \to L$ be functions. Then, the following $\mathscr{L}_{\mathbb{R}}$-sentence is valid (because it is equivalent to Prop. 2.2.20):

$$\begin{aligned}
&\forall F \in L \to L.\ \forall \overline{F} \in \overline{L} \to \overline{L}.\ \forall \bot \in L.\ \forall \overline{x} \in \overline{L}.\\
&\Big(\mathsf{Cpo}_{L,\sqsubseteq} \wedge \mathsf{Preord}_{\overline{L},\overline{\sqsubseteq}} \wedge \mathsf{Conti}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge \mathsf{Monotone}_{\overline{L},\overline{\sqsubseteq},\overline{L},\overline{\sqsubseteq}}(\overline{F}) \wedge \mathsf{Concr}_{L,\sqsubseteq,\overline{L},\overline{\sqsubseteq},\gamma}\\
&\wedge\ F \circ \gamma \sqsubseteq \gamma \circ \overline{F} \wedge \bot \sqsubseteq F(\bot) \wedge \bot \sqsubseteq \gamma(\overline{x}) \wedge \overline{F}(\overline{x}) \mathrel{\overline{\sqsubseteq}} \overline{x}\\
&\Rightarrow \mathrm{lfp}_{\bot} F \sqsubseteq \gamma(\overline{x})\Big).
\end{aligned}$$

By applying Lem. 2.2.8 to this $\mathscr{L}_{\mathbb{R}}$-sentence, we have the following valid $\mathscr{L}_{{}^*\mathbb{R}}$-sentence:

$$\begin{aligned}
&\forall F \in {}^*(L \to L).\ \forall \overline{F} \in {}^*(\overline{L} \to \overline{L}).\ \forall \bot \in {}^*L.\ \forall \overline{x} \in {}^*\overline{L}.\\
&\Big({}^*\mathsf{Cpo}_{L,\sqsubseteq} \wedge {}^*\mathsf{Preord}_{\overline{L},\overline{\sqsubseteq}} \wedge {}^*\mathsf{Conti}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge {}^*\mathsf{Monotone}_{\overline{L},\overline{\sqsubseteq},\overline{L},\overline{\sqsubseteq}}(\overline{F}) \wedge {}^*\mathsf{Concr}_{L,\sqsubseteq,\overline{L},\overline{\sqsubseteq},\gamma}\\
&\wedge\ F \circ {}^*\gamma\ {}^*\sqsubseteq\ {}^*\gamma \circ \overline{F} \wedge \bot\ {}^*\sqsubseteq\ F(\bot) \wedge \bot\ {}^*\sqsubseteq\ {}^*\gamma(\overline{x}) \wedge \overline{F}(\overline{x})\ {}^*\overline{\sqsubseteq}\ \overline{x}\\
&\Rightarrow {}^*\mathrm{lfp}_{\bot} F\ {}^*\sqsubseteq\ {}^*\gamma(\overline{x})\Big).
\end{aligned}$$

This yields the statement of this theorem. For example, we can confirm that ${}^*\mathsf{Concr}_{L,\sqsubseteq,\overline{L},\overline{\sqsubseteq},\gamma}$ holds from the following hypothesis in the theorem statement: $\overline{a} \mathrel{\overline{\sqsubseteq}} \overline{b} \Rightarrow \gamma(\overline{a}) \sqsubseteq \gamma(\overline{b})$ for all $\overline{a}, \overline{b} \in \overline{L}$. □

Our goal is overapproximation of the semantics of iteration of a loop-free While$^{\mathsf{dt}}$ program c, relying on Thm. 5.2.4. Towards the goal, the next step is to find a suitable $\overline{F} : {}^*\overline{L} \to {}^*\overline{L}$ that "stepwise approximates" $F = [\![c]\!]$, the collecting semantics of c. The next result implies that the $*$-transformation of $[\![\_]\!]_{\mathbb{CP}}$ (defined in a usual manner in standard abstract interpretation, as mentioned in §2.2.3) can be used in such $\overline{F}$.

Proposition 5.2.5. *Let $(L, \sqsubseteq), (\overline{L}, \overline{\sqsubseteq}), \gamma : \overline{L} \to L$ satisfy the hypotheses in Thm. 5.2.4. Assume that a continuous function $F : L \to L$ is stepwise approximated by a monotone function $\overline{F} : \overline{L} \to \overline{L}$, that is, $F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$. Then the *-continuous function ${}^*F : {}^*L \to {}^*L$ is overapproximated by the monotone and internal function ${}^*\overline{F} : {}^*\overline{L} \to {}^*\overline{L}$, i.e. ${}^*F \circ {}^*\gamma\ {}^*\sqsubseteq\ {}^*\gamma \circ {}^*\overline{F}$.*

Proof. It can be proved by applying the transfer principle to $\overline{F} \in \overline{L} \to \overline{L} \wedge \mathsf{Monotone}_{\overline{L},\overline{\sqsubseteq},\overline{L},\overline{\sqsubseteq}}(\overline{F}) \wedge F \circ \gamma \sqsubseteq \gamma \circ \overline{F}$. □

We summarize what we observed so far on nonstandard abstract interpretation by instantiating the abstract domain to ${}^*\mathbb{CP}_n$. In the following $[\![c]\!]$ is from Def. 5.1.2.

Corollary 5.2.6 (soundness of nonstandard abstract interpretation on ${}^*\mathbb{CP}_n$)**.** *Let c be a loop-free* While$^{\mathsf{dt}}$ *command; and let $\bot \in {}^*(\mathcal{P}(\mathbb{R}^n))$ and $\overline{x} \in {}^*\mathbb{CP}_n$ be such that $({}^*[\![c]\!]_{\mathbb{CP}})(\overline{x})\ {}^*\sqsubseteq\ \overline{x}$ and $\bot\ {}^*\sqsubseteq\ {}^*\gamma_{\mathbb{CP}_n}(\overline{x})$. Then we have $\mathrm{lfp}_{\bot}[\![c]\!]\ {}^*\sqsubseteq\ {}^*\gamma_{\mathbb{CP}_n}(\overline{x})$.* □

5.2.3 Hyperwidening and Uniform Widening Operators

Towards our goal of using Thm. 5.2.4, the last remaining step is to find a prefixed point $\overline{x}$, i.e. $\overline{F}(\overline{x})\ {}^*\overline{\sqsubseteq}\ \overline{x}$. This is where widening operators are standardly used; see §2.2.3.

We can try $*$-transforming a (standard) notion—a strategy that we have used repeatedly in the current section. This yields the following result, that has a problem that is discussed shortly.

Theorem 5.2.7. *Let $(L, \sqsubseteq)$ be a preorder and $\nabla : L \times L \to L$ be a widening operator on L. Let $F : {}^*L \to {}^*L$ be a monotone and internal function; and $\bot \in {}^*L$ be such that $\bot\ {}^*\sqsubseteq\ F(\bot)$. The iteration* hyper*-sequence $\langle X_i \rangle_{i \in {}^*\mathbb{N}}$—indexed by hypernaturals $i \in {}^*\mathbb{N}$—that is defined by*

$$X_0 = \bot, \quad X_{i+1} = \begin{cases} X_i & (\textit{if } F(X_i)\ {}^*\sqsubseteq\ X_i) \\ X_i\,{}^*\nabla\, F(X_i) & (\textit{otherwise}) \end{cases} \text{ for all } i \in {}^*\mathbb{N}$$

reaches its limit within some hypernatural number of steps and the limit $\bigsqcup_{i \in \mathbb{N}} X_i$ is a prefixed point of F such that $\bot\ {}^*\sqsubseteq\ \bigsqcup_{i \in \mathbb{N}} X_i$.

Proof. Let $L \in \mathbb{U}$ be a set, $\sqsubseteq \in \mathcal{P}(L \times L)$ be a binary relation on L and $\nabla : L \times L \to L$ be a function. Then, the following $\mathscr{L}_{\mathbb{R}}$-sentence is valid (because it is equivalent to Prop. 2.2.22):

$$\begin{aligned} &\forall F \in L \to L.\ \forall \bot \in L.\ \forall X \in \mathbb{N} \to L. \\ &\mathsf{Preord}_{L,\sqsubseteq} \wedge \mathsf{Monotone}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge \mathsf{Basis}_{L,\sqsubseteq}(\bot, F) \wedge \mathsf{Widen}_{L,\sqsubseteq,\nabla} \\ &\wedge \mathsf{WidenSeq}_{L,\sqsubseteq,\nabla}(X, \bot, F) \\ &\Rightarrow \exists i \in \mathbb{N}.\ \forall j \in \mathbb{N}.\ i \leq j \Rightarrow X(i) = X(j) \\ &\wedge \forall k \in \mathbb{N}.\ \Big(\big(\forall l \in \mathbb{N}.\ k \leq l \Rightarrow X(k) = X(l) \big) \Rightarrow F\big(X(k)\big) \sqsubseteq X(k) \Big). \end{aligned}$$

By applying Lem. 2.2.8 to this $\mathscr{L}_{\mathbb{R}}$-sentence, we have the following valid $\mathscr{L}_{{}^*\mathbb{R}}$-sentence:

$$\begin{aligned} &\forall F \in {}^*(L \to L).\ \forall \bot \in {}^*L.\ \forall X \in {}^*(\mathbb{N} \to L). \\ &{}^*\mathsf{Preord}_{L,\sqsubseteq} \wedge {}^*\mathsf{Monotone}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge {}^*\mathsf{Basis}_{L,\sqsubseteq}(\bot, F) \wedge {}^*\mathsf{Widen}_{L,\sqsubseteq,\nabla} \\ &\wedge {}^*\mathsf{WidenSeq}_{L,\sqsubseteq,\nabla}(X, \bot, F) \\ &\Rightarrow \exists i \in {}^*\mathbb{N}.\ \forall j \in {}^*\mathbb{N}.\ i \leq j \Rightarrow X(i) = X(j) \\ &\wedge \forall k \in {}^*\mathbb{N}.\ \Big(\big(\forall l \in {}^*\mathbb{N}.\ k \leq l \Rightarrow X(k) = X(l) \big) \Rightarrow F\big(X(k)\big)\ {}^*\sqsubseteq\ X(k) \Big) \end{aligned}$$

This yields the statement of this theorem. Note that the well-definedness of the iteration hyper-sequence (by induction on $i \in {}^*\mathbb{N}$) is implicit in the above transfer arguments. □

The problem of Thm. 5.2.7 is that the *finite-step convergence* of iteration sequences for the original widening operator (described in Prop. 2.2.22) is now transferred to *hyperfinite-step convergence*. This is not desired. All the entities from NSA that we have used so far are constructs in denotational semantics—whose only role is to ensure soundness of verification methodologies[2] and on

[2]Recall that WHILE$^{\mathsf{dt}}$ is a *modeling* language and we do not execute them.

which we never actually operate—and therefore their infinite/infinitesimal nature has been not a problem. In contrast, computation of the iteration hypersequence $\langle X_i \rangle_{i \in {}^*\mathbb{N}}$ is what we actually compute to overapproximate program semantics; and therefore its termination guarantee within $i \in {}^*\mathbb{N}$ steps (Thm. 5.2.7) is of no use.

As a remedy we introduce a new notion of *uniformity* of the (standard) widening operators. It strengthens the original termination condition (Def. 2.2.21) by imposing a uniform bound i for stability of arbitrary chains $\langle x_i \rangle \in L^{\mathbb{N}}$. Logically the change means replacing $\forall\exists$ by $\exists\forall$.

Definition 5.2.8 (uniform widening)**.** Let $(L, \sqsubseteq)$ be a preorder. A function $\nabla : L \times L \to L$ is said to be a *uniform widening operator* if the following two conditions hold.

- (Covering) For any $x, y \in L$, $x \sqsubseteq x \nabla y$ and $y \sqsubseteq x \nabla y$.
- (Uniform termination) Let $x_0 \in L$. There exists a *uniform bound* $i \in \mathbb{N}$ such that: for any ascending chain $\langle x_k \rangle \in L^{\mathbb{N}}$ starting from x_0, there exists $j \leq i$ at which the chain $\langle y_k \rangle \in L^{\mathbb{N}}$, defined by $y_0 = x_0$ and $y_{k+1} = y_k \nabla x_{k+1}$ for all $k \in \mathbb{N}$, stabilizes (i.e. $y_j = y_{j+1}$).

It is straightforward that uniform termination implies termination.

We investigate uniformity of some of the commonly-known widening operators on convex polyhedra.

Theorem 5.2.9. *Among the three widening operators in §2.2.3, ∇_S (Def. 2.2.24) and ∇_M (Def. 2.2.25) are uniform, but ∇_N (Def. 2.2.33) is not.*

Proof. First, the uniformity of ∇_S is proved as follows. Let $\langle X_i \rangle_i$ be a iteration sequence defined by $\nabla_{\mathbb{CP}_n}$, a basis $X_0 = \mathsf{con}(C_0)$ and a monotone function F. Let $\langle C_i \rangle_i$ be the sequence of constraint systems that corresponds to $\langle X_i \rangle_i$. By definition of $\nabla_{\mathbb{CP}_n}$ and the construction of $\langle X_i \rangle_i$, regardless of the function F, $C_{i+1} \subseteq C_i$ for all $i \in \mathbb{N}$. Thus for any basis $X_0 = \mathsf{con}(C_0)$ and monotone function F, we can reach a prefixed point by iterating the widening operator at most $\#(C_0)$ times and this means the widening operator $\nabla_{\mathbb{CP}_n}$ is uniform.

Then, we can prove that ∇_M is also uniform in the following way. The constraints in M may be added in the iteration sequence, but by the definition of the standard widening ∇_S, a constraint in M will never appear once it is violated. Therefore the number of steps for an iteration sequence to converge is at most $\#(M)$ larger than the case of standard widening.

Finally, ∇_N is proved to be nonuniform as follows. Assume that $P_1 = \mathsf{con}\{0 \leq x \leq 1, 0 \leq y \leq 1, z = 0\} \in \mathbb{CP}_3$, $P_2 \in \mathbb{CP}_3$ includes P_1 and the linear equation "$z = 0$" is not included in C_2. Then $P_1 \curvearrowright_N P_2$ holds because $\#eq(C_1) > \#eq(C_2)$. The maximum number of steps for an iteration sequence starting from P_2 to converge is $\#C_2$. This is not limited uniformly because you can define P_2 such that $\#C_2$ is as large as you like. □

The following theorem is a "practical" improvement of Thm. 5.2.7; its proof relies on instantiating the uniform bound i in a suitable $\mathscr{L}_{\mathbb{R}}$-formula with a Skolem constant, before transfer.

Theorem 5.2.10. *Let $(L, \sqsubseteq)$ be a preorder and $\nabla \in L \times L \to L$ be a uniform widening operator on L. Let $F : {}^*L \to {}^*L$ be a monotone and internal function; and $\bot \in L$ be such that ${}^*\bot \mathrel{{}^*\sqsubseteq} F({}^*\bot)$. The iteration sequence $\langle X_i \rangle_{i \in \mathbb{N}}$ defined by*

$$X_0 = {}^*\bot, \quad X_{i+1} = \begin{cases} X_i & \text{if } F(X_i) \mathrel{{}^*\sqsubseteq} X_i \\ X_i \mathrel{{}^*\nabla} F(X_i) & \text{otherwise} \end{cases} \quad \text{for all } i \in \mathbb{N}$$

reaches its limit within some finite number of steps; and the limit $\bigsqcup_{i \in \mathbb{N}} X_i$ is a prefixed point of F such that ${}^*\bot \mathrel{{}^*\sqsubseteq} \bigsqcup_{i \in \mathbb{N}} X_i$.

Proof. We can characterize uniform widening operators as an $\mathscr{L}_{\mathbb{R}}$-sentence as follows (covering condition has been already expressed as an $\mathscr{L}_{\mathbb{R}}$-formula in Def. 2.2.15):

$$\begin{aligned} &\mathsf{UnifTerm}_{L,\sqsubseteq,\nabla} :\equiv \forall x_0 \in L.\, \exists i \in \mathbb{N}.\, \forall x \in \mathbb{N} \to L.\, (\mathsf{AscCn}(x) \wedge x(0) = x_0) \Rightarrow \\ &\Big(\forall y \in \mathbb{N} \to L.\, \Big(\big(y(0) = x(0) \wedge \forall n \in \mathbb{N}.\, y(n+1) = y(n) \nabla x(n+1)\big) \\ &\Rightarrow \exists j \in \mathbb{N}.\, \big(j \leq i \wedge y(j) = y(j+1)\big)\Big)\Big) \\ &\mathsf{UnifWiden}_{L,\sqsubseteq,\nabla} :\equiv \mathsf{Cover}_{L,\sqsubseteq,\nabla} \wedge \mathsf{UnifTerm}_{L,\sqsubseteq,\nabla} \end{aligned}$$

Let $L \in \mathbb{U}$ be a set, $\sqsubseteq \in \mathcal{P}(L \times L)$ be a binary relation on L and $\nabla : L \times L \to L$ be a function. Then, we can see directly that the following $\mathscr{L}_{\mathbb{R}}$-sentence is valid:

$$\forall \bot \in L.\, \exists i \in \mathbb{N}.\, \Psi(\bot)(i), \tag{5.1}$$

where

$$\begin{aligned} &\Psi(\bot)(i) = \\ &\quad \forall F \in L \to L.\, \forall X \in \mathbb{N} \to L. \\ &\quad \mathsf{Preord}_{L,\sqsubseteq} \wedge \mathsf{Monotone}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge \mathsf{Basis}_{L,\sqsubseteq}(\bot, F) \wedge \mathsf{UnifWiden}_{L,\sqsubseteq,\nabla} \\ &\quad \wedge \mathsf{WidenSeq}_{L,\sqsubseteq,\nabla}(X, \bot, F) \\ &\quad \Rightarrow \forall j \in \mathbb{N}.\, i \leq j \Rightarrow X(i) = X(j) \\ &\quad \wedge \forall k \in \mathbb{N}.\, \Big(\big(\forall l \in \mathbb{N}.\, k \leq l \Rightarrow X(k) = X(l)\big) \Rightarrow F\big(X(k)\big) \sqsubseteq X(k)\Big). \end{aligned}$$

Assume that $\bot \in L$ is given. Then, by the $\mathscr{L}_{\mathbb{R}}$-sentence (5.1), there exists $i \in \mathbb{N}$ such that $\Psi(\bot)(i)$ holds. Therefore, by transferring $\Psi(\bot)(i)$, ${}^*\Psi(\bot)(i)$ holds for such $i \in \mathbb{N}$. Note that ${}^*\Psi(\bot)(i)$ is the following $\mathscr{L}_{{}^*\mathbb{R}}$-sentence ($\bot$ and i are dealt with as constants in the following $\mathscr{L}_{{}^*\mathbb{R}}$-sentence because $\bot$ and i are defined outside the $\mathscr{L}_{\mathbb{R}}$-sentence):

$$\begin{aligned} &\forall F \in {}^*(L \to L).\, \forall X \in {}^*(\mathbb{N} \to L). \\ &{}^*\mathsf{Preord}_{L,\sqsubseteq} \wedge {}^*\mathsf{Monotone}_{L,\sqsubseteq,L,\sqsubseteq}(F) \wedge {}^*\mathsf{Basis}_{L,\sqsubseteq}({}^*\bot, F) \wedge {}^*\mathsf{UnifWiden}_{L,\sqsubseteq,\nabla} \\ &\wedge {}^*\mathsf{WidenSeq}_{L,\sqsubseteq,\nabla}(X, {}^*\bot, F) \\ &\Rightarrow \forall j \in {}^*\mathbb{N}.\, i \leq j \Rightarrow X(i) = X(j) \\ &\wedge \forall k \in {}^*\mathbb{N}.\, \Big(\big(\forall l \in {}^*\mathbb{N}.\, k \leq l \Rightarrow X(k) = X(l)\big) \Rightarrow F\big(X(k)\big) \mathrel{{}^*\sqsubseteq} X(k)\Big). \end{aligned}$$

This yields Thm. 5.2.10. □

Note that uniformity of ∇ is a *sufficient condition* for the termination of nonstandard iteration sequences (by ${}^*\nabla$); Thm. 5.2.10 does not prohibit other useful widening operators in the nonstandard setting. Furthermore, there can be a useful (nonstandard) widening operator other than a hyperwidening operator ${}^*\nabla$ that arise via a standard widening operator ∇.

It is a direct consequence of Thm. 5.2.10 and Thm. 5.2.9 that the analysis of WHILE$^{\mathtt{dt}}$ programs on ${}^*\mathbb{CP}_n$ is terminating with ∇_S or ∇_M.

5.3 Analysis of Linear Water Tank Example

In this section we illustrate how our framework described in §5.2 works. We use the well-known example of the linear water tank introduced in [6]. The details of the system can be also found in §2.2.4.

Recall that in §2.2.4, we reviewed the usual abstract interpretation workflow without extension with infinitesimals. We emphasize that our extended framework works just in the same manner: without any explicit ODEs or any additional theoretical infrastructure for ODEs; but only adding a constant `dt`. In the "standard" scenario in §2.2.4, we approximated the dynamics of the water level by discretizing the continuous notion of time ($\mathtt{dt}' = 0.2$). While this made the usual abstract interpretation workflow go around, there is a price to pay—the analysis result is not *precise*. Specifically, the reachable region thus over-approximated is $0.6 \leq x \leq 12.2$, while the real reachable region is $1 \leq x \leq 12$. There are also examples in which discretization even leads to *unsound* analysis results.

In our extended framework, the same (hybrid) dynamics of the linear water tank is modeled by a program in Code 5.1. Here we used the infinitesimal constant `dt` in WHILE$^{\mathtt{dt}}$, instead of $\mathtt{dt}' = 0.2$ in Code 2.1. For the analysis of this WHILE$^{\mathtt{dt}}$ program in Code 5.1, we can follow exactly the same path as in §2.2.4. The collecting semantics of the WHILE$^{\mathtt{dt}}$ program in Code 5.1 follows Def. 5.1.2. As in §2.2.4, we separate the state space using the values of the Boolean variables p and s. Therefore, the concrete domain is $\left({}^*\big(\mathcal{P}(\mathbb{R}^2)\big)\right)^4$. The corresponding abstract domain of convex polyhedra over hyperreals is $({}^*\mathbb{CP}_2)^4$.

On this abstract domain, we can iterate the loop in Code 5.1. After each iteration of the loop, we apply ${}^*(\nabla_M)$, the *-transform of the uniform widening operator ∇_M. It accelerates the convergence of the iteration sequence, and Thm. 5.2.9 and Thm. 5.2.10 ensure that it reaches a prefixed point in finitely many steps. The iteration sequence is much like in Fig. 2.2 obtained in §2.2.4, but here the two parallelograms depicted in green and yellow have infinitely small width of `dt` and `2dt`, respectively. This leads to the analysis result $1 - 2\mathtt{dt} \leq x \leq 12 + \mathtt{dt}$. The soundness of this result is guaranteed by Thm. 5.2.4. Since `dt` is an infinitesimal number, the last result is practically as good as $1 \leq x \leq 12$. In the next section, we will introduce a prototype implementation that automates this analysis.

5.4 Implementation and Experiments

5.4.1 Implementation

We implemented a prototype tool for analysis of WHILE$^{\mathtt{dt}}$ programs. The tool currently supports: ${}^*\mathbb{CP}_n$ as an abstract domain; and ${}^*\nabla_M$, *-transformation of ∇_M in Def. 2.2.25 as a widening operator. Its input is a WHILE$^{\mathtt{dt}}$ program. It outputs a convex polyhedron that overapproximates the set of reachable memory states for each modes (or the values of discrete variables). Our tool consists principally of the following two components: 1) an OCaml frontend for parsing, forming an iteration sequence and making the set M for ${}^*\nabla_M$; and 2) a Mathematica backend for executing operations on convex polyhedra. The two components are interconnected via MathLink.

There are some libraries such as Parma Polyhedra Library [12] that are commonly used to execute operations on convex polyhedra. They cannot be used in our implementation because we have to handle the infinitesimal constant `dt` as an truly infinitesimal value. Instead we implemented Chernikova's algorithm [22–24, 65] symbolically, using *computer algebra system (CAS)* on Mathematica based on Prop. 5.4.1.

In the implementation, we rely on the following proposition to check the validity of formulas including the infinitesimal constant `dt`.

Proposition 5.4.1. *Let A be an $\mathscr{L}_{\mathbb{R}}$-formula with a unique free variable x; to emphasize it we write $A(x)$ for A. Then the validity of the formula*

$$\exists r \in \mathbb{R}.\, (0 < r \wedge \forall x \in \mathbb{R}.\, (0 < x < r \Rightarrow A\,(x)))$$

(in $V(\mathbb{R})$) implies the validity of ${}^*A(\mathtt{dt})$ in $V({}^*\mathbb{R})$.

Proof. Assume that

$$0 < r \wedge \forall x \in \mathbb{R}.\, (0 < x < r \Rightarrow A\,(x))$$

is valid for some $r \in \mathbb{R}$. By transfer,

$$0 < r \wedge \forall x \in {}^*\mathbb{R}.\, (0 < x < r \Rightarrow {}^*A\,(x))$$

is also valid for that r. This implies ${}^*A(\mathtt{dt})$ since $0 < \mathtt{dt} < r$ for any positive $r \in \mathbb{R}$. □

Prop. 5.4.1 ensures that the transformation from ${}^*A(\mathtt{dt})$ to

$$\exists r \in \mathbb{R}.\, (0 < r \wedge \forall x \in \mathbb{R}.\, (0 < x < r \Rightarrow A\,(x)))$$

does not violate the soundness of the analysis. When we have to evaluate a formula including `dt`, we instead resolve $\exists r \in \mathbb{R}.\, (0 < r \wedge \forall x \in \mathbb{R}.\, (0 < x < r \Rightarrow A\,(x)))$ using CAS (e.g. quantifier elimination).

Remark 5.4.2. When we model nonlinear ODEs in WHILE$^{\mathtt{dt}}$, we have nonlinear expressions with an infinitesimal constant `dt` in the RHS of assignments. In classical abstract interpretation on the domain of convex polyhedra, when we encounter a nonlinear assignment to a variable, we discard all the information

about the variable (see [31]). This leads to an unacceptable imprecision of the analysis. In the current prototype implementation in which we rely on Mathematica backend, the constraints that must be satisfied after a nonlinear assignment are computed symbolically, and the linear constraints among them give the abstract state after the assignment. To deal with such nonlinear expressions in numerical implementation (which is an important direction of future work), we are considering combining our methodology with linearization techniques such as [3].

5.4.2 Experiments

We analyzed two WHILE$^{\mathtt{dt}}$ programs—the linear water tank (Code 5.1) and the nonlinear water tank (Code 5.2)—with our prototype. The experiments were on Apple MacBook Air with 2.2 GHz Intel Core i7 CPU and 8 GB memory. The execution times are the average of 10 runs.

Linear Water Tank The first example we analyzed using our prototype imprementation is the linear water tank in Code 5.1. This is a piecewise-linear dynamics and a typical example used in hybrid automata literature. Our tool automates the analysis presented in §5.3; the execution time was 20.815 sec.

Nonlinear Water Tank We consider the same dynamics as the one used in §3.6.2 and §4.3.2. Its WHILE$^{\mathtt{dt}}$ model is presented in Code 5.2. In the previous example of the linear water tank, we assumed the fixed time lag of 2 seconds and it was taken into account in the WHILE$^{\mathtt{dt}}$ model. In this example, the WHILE$^{\mathtt{dt}}$ program does not include time lags. The reachability analysis of this nonlinear water tank example with bounded delays can be done by combining the analysis in this section with the results in §3.6 or §4.3. It will be presented later in §6.2.

As we discussed in Rem. 5.4.2, the nonlinear ODE is modeled as a nonlinear assignment (in line 6 and 9 in Code 5.2). Our approach of nonstandard abstract interpretation successfully analyzes this example without explicit piecewise-linear approximation. We believe this result witnesses a potential of our approach. We skip how it analyzes this example since the procedure is the same as the linear water tank case. Our tool executes in 5.242 sec. and outputs an approximation from which we obtain an invariant $1 - \frac{\mathtt{dt}}{5} \leq x \leq 10 + \frac{\mathtt{dt}}{10}$.

```
1  /*Nonlinear Water Tank*/
2  x := 1;
3  while true do {
4    if x < 1 then x := x + ((11 - x) * dt / 10)
5    else {
6      if x > 10 then x := x - (((x ^ (1 / 2)) * dt) / 5)
7      else {
8        if * then x := x + ((11 - x) * dt / 10)
9        else x := x - (((x ^ (1 / 2)) * dt) / 5)
10     }
11   }
12 }
```

Code 5.2: Nonlinear Water Tank in WHILE^{dt}

Chapter 6

Two-Step Analysis of Switched Systems with Delays

In this chapter, we introduce a two-step reachability analysis workflow of a switched system with delays. It is a combination of an abstraction methodology of delays and the reachability analysis of delay-free hybrid systems. In §6.1, we explain some theoretical results to guarantee the soundness of the overall two-step analysis under state-dependent controllers. It is an adaptation of the results in [40]. Then in §6.2 and §6.3, we illustrate our proposed two-step analysis using examples.

In the first step of the two-step analysis, using the methodology introduced in Chapter 4, we compute an upper bound of the Skorokhod distance between the trajectory of the switched system with delays and the one without delays, as Thm. 4.2.4 and Thm. 4.2.6 ensures. We can also use the methodology introduced in Chapter 3 in the same way, but we use the one in Chapter 4 for better precision. Using the resulting upper bound, the reachability of the switched system with delays reduces to the reachability of the delay-free model.

Then, in the second step, we need to analyze the reachability of the switched system without delays. Because we have separated the first step and the second step, we can apply any existing reachability analysis technique for hybrid systems. One possibility is the methodology introduced in Chapter 5. In §6.2, we use the nonlinear water tank as an example to show the applicability of our method to nonlinear dynamics. In §6.3, we use the boost DC-DC converter as an example. We do not use the methodology in Chapter 5. Instead, since the dynamics of this example is linear, we can even synthesize a safety controller that keeps the trajectory within a safe region, using the state-space discretization method introduced in [40].

6.1 Theoretical Background for Reachability Analysis by Approximate Bisimulation under State-Dependent Controllers

In the subsequent examples, we will consider the reachability of the system controlled by a state-dependent controller. In Chapter 3 and Chapter 4, we did not consider the existence of a controller and just assumed that the same mode is always enabled for both systems. This is not the case if the system with delays and its delay-free model is controlled by the same state-dependent controller. In

this section, we describe necessary theoretical results for reachability analysis based on approximate bisimulations when a state-dependent controller is given. The following definitions and results are adaptations of the ones in [40], which are originally for safety controller synthesis, to our reachability analysis setting.

To analyze the reachability, we first fix a *controller* for the system. We focus on the controllers that only depend on the current state $x \in \mathbb{R}^n$ of a switched system.

Definition 6.1.1. Let $\Sigma = (\mathbb{R}^n, P, \mathbf{P}, F)$ be a τ-periodic switched system or a τ-periodic switched system with switching delays within δ_0. A *controller* for Σ is a function $\mathcal{S} : \mathbb{R}^n \to \mathbf{P}(P)$.

The dynamics of the transition system $T(\Sigma) = (Q, L, \longrightarrow, O, H, I)$ controlled by $\mathcal{S}$ from a set of initial states $X_0 \subseteq \mathbb{R}^n$ is described by another transition system $T(\Sigma)_{\mathcal{S},X_0} = (Q, L, \xrightarrow[\mathcal{S}]{}, O, H, I_{X_0})$, where $\xrightarrow[\mathcal{S}]{}$ is defined by $(x,t,p) \xrightarrow[\mathcal{S}]{p''} (x',t',p')$ if $p' \in \mathcal{S}(x)$ and $(x,t,p) \xrightarrow{p''} (x',t',p')$, and I_{X_0} is defined by $\{(x_0, 0, p) \in I \mid x_0 \in X_0 \text{ and } p \in P\}$.

Remark 6.1.2. We explain some technical differences between our setting and the one in [40]. In our setting, a controller is not a function from the set of states Q to the set of transition labels L on the transition systems we constructed in §3.2. In these transition systems, the next possible transition label is uniquely determined by the third element of the current state $q \in Q$.

Another difference is that for the transition systems constructed in that manner, any controller in $\mathbb{R}^n \to \mathcal{P}(P)$ is *well-defined*. In the definitions and theorems, we restrict ourselves to the transition systems obtained from switched systems in such a way that we described in §3.2, and assume the well-definedness of the controllers. For details of well-definedness, see [40].

Definition 6.1.3. Let $\mathcal{S}$ be a controller for a switched system Σ, and $X_0 \subseteq \mathbb{R}^n$ be a set of initial states. A subset $O_s \subseteq O$ is said to be a *controlled invariant* for $T(\Sigma)_{\mathcal{S},X_0}$ if for all initial state $q_0 \in I_{X_0}$ with $H(q_0) \in O_s$, and for each state trajectory$\big((q_0,l_0),(q_1,l_1),\cdots,(q_i,l_i),\cdots\big)$ of $T(\Sigma)_{\mathcal{S},X_0}$, $H(q_i) \in O_s$ for all $i \in \mathbb{N}$.

In the subsequent theorem and the corollary, we only consider $O_s \subseteq O = \mathbb{R}^n \times \mathbb{R}^+ \times P$ defined by $O_s = X \times \mathbb{R}^+ \times P$ for some $X \subseteq \mathbb{R}^n$, and regard O_s as the subset X of $\mathbb{R}^n$. We use the following notations.

- For $X \subseteq \mathbb{R}^n$, the *ε-expansion* $E_\varepsilon(X)$ is defined by $\{y \in \mathbb{R}^n \mid \text{there exists } x \in X \text{ such that } \|x - y\| \leq \varepsilon\}$.
- For $X \subseteq \mathbb{R}^n$, the *ε-contraction* $C_\varepsilon(X)$ is defined by $\{y \in X \mid \text{for all } x \in \mathbb{R}^n, \text{ if } \|x - y\| \leq \varepsilon \text{ then } x \in X\}$.
- For $X \subseteq \mathbb{R}^n$ and $\mathcal{S} : \mathbb{R}^n \to \mathcal{P}(P)$, $\mathcal{S}(X)$ is defined by $\bigcup_{x \in X} \mathcal{S}(x)$.
- For $R_\varepsilon \subseteq Q_{\tau,\delta_0} \times Q_\tau$ and $x \in \mathbb{R}^n$, $R_\varepsilon^{-1}(x)$ is defined by $\{y \in \mathbb{R}^n \mid \text{there exists } t, t' \in \mathbb{R}^+ \text{ and } p, p' \in P \text{ such that } (y,t,p) R_\varepsilon (x,t',p')\}$.
- For $R_\varepsilon \subseteq Q_{\tau,\delta_0} \times Q_\tau$ and $x \in \mathbb{R}^n$, $R_\varepsilon(x)$ is defined by $\{y \in \mathbb{R}^n \mid \text{there exists } t, t' \in \mathbb{R}^+ \text{ and } p, p' \in P \text{ such that } (x,t,p) R_\varepsilon (y,t',p')\}$.

The following theorem guarantees the soundness of the reachability analysis in §6.2. The proof is omitted because it can be proved easily in a similar way as [40, Thm. 1].

Theorem 6.1.4. *Let Σ_{τ,δ_0} be a τ-periodic switched system and Σ_τ be its delay-free model. Assume that there exists an ε-approximate bisimulation relation $R_\varepsilon \subseteq Q_{\tau,\delta_0} \times Q_\tau$ between the state spaces of $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$. Let $\mathcal{S}_{\tau,\delta_0}$ be a controller for Σ_{τ,δ_0}. Let $X_0 \subseteq \mathbb{R}^n$ be a set of initial states.*

Now we consider a controller $\mathcal{S}_\tau$ satisfying $\mathcal{S}_\tau(x) \supseteq \mathcal{S}_{\tau,\delta_0}(R_\varepsilon^{-1}(x))$ for all $x \in \mathbb{R}^n$. If O_s is a controlled invariant for $T(\Sigma_\tau)_{\mathcal{S}_\tau,X_0}$, then $E_\varepsilon(O_s)$ is a controlled invariant for $T(\Sigma_{\tau,\delta_0})_{\mathcal{S}_{\tau,\delta_0},X_0}$. □

For controller synthesis, we add the notion of *non-blockingness* for controllers.

Definition 6.1.5. Given a controller $\mathcal{S} \in \mathbb{R}^n \to \mathcal{P}(P)$, a state $x \in \mathbb{R}^n$ is said to be *non-blocking* if $\mathcal{S}(x) \neq \emptyset$.

Let Σ be a switched system (either periodic or periodic with delays), O_s be a subset of the set of outputs O, and $X_0 \subseteq \mathbb{R}^n$ be a set of initial states. A controller $\mathcal{S}$ for Σ is said to be a *safety controller* for specification O_s with initial states X_0 if for all non-blocking initial state $q_0 \in I_{X_0}$ with $H(q_0) \in O_s$, and for each state trajectory $\big((q_0, l_0), (q_1, l_1), \cdots, (q_{N-1}, l_{N-1}), q_N\big)$ of length N of $T(\Sigma)_{\mathcal{S},X_0}$, $H(q_i) \in O_s$ for all $i \in \{0, \cdots, N\}$ and q_N is non-blocking.

The following corollary is an adaptation of [40, Thm. 1]. It will be used in §6.3.

Corollary 6.1.6. *Let Σ_{τ,δ_0} be a τ-periodic switched system and Σ_τ be its delay-free model. Assume that there exists an ε-approximate bisimulation relation $R_\varepsilon \subseteq Q_{\tau,\delta_0} \times Q_\tau$ between the state spaces of $T(\Sigma_{\tau,\delta_0})$ and $T(\Sigma_\tau)$. Let $O_s \subseteq O$ be a given safety specification. Let $\mathcal{S}_\tau$ be a safety controller for Σ_τ, for specification $C_\varepsilon(O_s)$ with initial states $X_0 \subseteq \mathbb{R}^n$.*

Now we consider a controller $\mathcal{S}_{\tau,\delta_0}$ defined by $\mathcal{S}_{\tau,\delta_0}(x) = \mathcal{S}_\tau(R_\varepsilon(x))$ for all $x \in \mathbb{R}^n$. Then $\mathcal{S}_{\tau,\delta_0}$ is a safety controller for Σ_{τ,δ_0}, for specification O_s with initial states X_0. □

Soundness of the overall two-step reachability analysis is guaranteed by the combination of soundness of the abstraction of delays in the first step (e.g., Thm. 4.2.4 or Thm. 4.2.6), soundness of the reachability analysis in the second step (e.g., Thm. 5.2.4) and Thm. 6.1.4 (Cor. 6.1.6 for controller synthesis). In the following two sections, we will illustrate the two-step reachability analysis and safe controller synthesis using two examples.

6.2 Reachability Analysis of Nonlinear Water Tank

In this section, we compute an overapproximation of the reachability of the nonlinear water tank example with delays. The detailed description of the example can be found in §3.6.2.

To analyze the reachability of the delayed system, we first fix a controller. We assume the following controller $\mathcal{S}_{\tau,\delta_0}$ for the delayed system Σ_{τ,δ_0}:

$$\mathcal{S}_{\tau,\delta_0}(x) = \begin{cases} \{ON\} & \text{if } x < 2 \\ \{OFF\} & \text{if } x > 9 \\ \{ON, OFF\} & \text{otherwise} \end{cases}.$$

Note that we do not assume the periodic sensing for this example. To allow the similar analysis as §5.4.2, we assume that the sensing can occur at anytime. Hereafter, instead of assuming a period τ, we restrict ourselves to the switching signals whose minimum dwell time is τ, among the switching signals that are enabled by the controllers. It is easy to see that this restriction enables us to reuse exactly the same results in Chapter 3 and Chapter 4.

Recall that, in §4.3.2, we gave an approximate bisimulation relation R_ε with $\varepsilon = 0.388234$ between the two transition systems constructed from the nonlinear water tank Σ_{τ,δ_0} with delays and its delay-free model Σ_τ. To analyze the reachability of Σ_{τ,δ_0} controlled by the controller $\mathcal{S}_{\tau,\delta_0}$, we first define a controller $\mathcal{S}_\tau$ for Σ_τ by $\mathcal{S}_\tau(x) = \mathcal{S}_{\tau,\delta_0}(R_\varepsilon^{-1}(x))$. From the definition of R_ε, it is easy to see that the controller $\mathcal{S}_\tau$ works in the following way:

$$\mathcal{S}_\tau(x) = \begin{cases} \{ON\} & \text{if } x < 1.61177\ldots \\ \{OFF\} & \text{if } x > 9.38823\ldots \\ \{ON, OFF\} & \text{otherwise} \end{cases}.$$

By a similar analysis as §5.4.2, we can obtain that $[1.61177, 9.38824]$ is a controlled invariant for $T(\Sigma_\tau)_{\mathcal{S}_\tau}$. Finally, by Thm. 6.1.4, we can say that the delayed system is kept in the ε-expansion $[1.22354, 9.77647]$ by the controller $\mathcal{S}_{\tau,\delta_0}$.

6.3 Controller Synthesis of Boost DC-DC Converter

In this section, we use the boost DC-DC converter as an example. The description of the example is in §3.6.1. Since the dynamics of this example is linear, we can even synthesize a safety controller that keeps the trajectory within a safe region, based on the safety controller synthesis workflow using approximate bisimulation in [40]. For the reachability analysis in the previous example, we have reduced it to that of its delay-free model. The reachability of the delay-free model was analyzed by the methodology in Chapter 5. We follow a similar workflow for the safety controller synthesis here: first the controller synthesis problem of the original system is reduced to that of its delay-free discretized model via two approximate bisimulations; then the safety controller synthesis can be done for that symbolic model using an existing controller synthesis methodology.

The details of the workflow are shown in Fig. 6.1. Note that we use two different approximate bisimulations. Our proposed methodology in Chapter 4 is used to derive the first error bound ε_1 between (the transition system $T(\Sigma_{\tau,\delta_0})$ derived from) the actual system Σ_{τ,δ_0}, and (the transition system $T(\Sigma_\tau)$ derived from) the delay-free model Σ_τ. The latter system Σ_τ is a delay-free periodic switched system, to which we can apply the results of [43]. We thus construct a

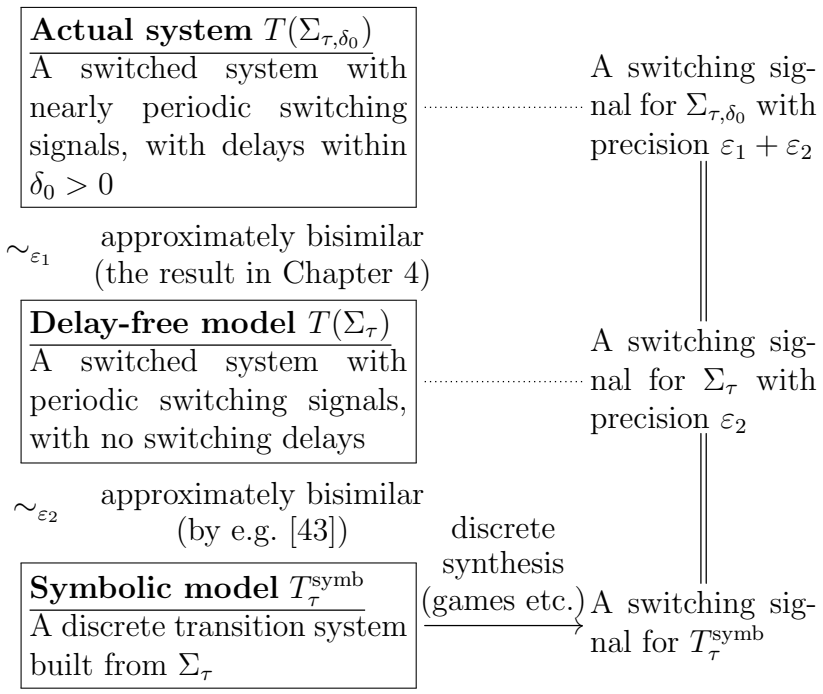

Figure 6.1: A control synthesis workflow for switched systems with delays. We separate two concerns: time-delays and discretization of state spaces. The same stability assumption on Σ_τ can be used for establishing both $\sim_{\varepsilon_1}$ and $\sim_{\varepsilon_2}$.

discretized symbolic model T_τ^{symb} and establish the second approximate bisimulation $\sim_{\varepsilon_2}$ in Fig. 6.1. The fact that our construction relies on same incremental stability assumptions as in [43] means that, for establishing both of the approximate bisimulations $\sim_{\varepsilon_1}$ and $\sim_{\varepsilon_2}$, we can reuse the same ingredient (namely a Lyapunov function for δ-GUAS). This can save a lot of efforts. Concretely, we obtained $\varepsilon_1 = 0.023655$ in §4.3.1. For ε_2, we can make ε_2 as small as you wish (see [43].)

We briefly sketch the concrete controller synthesis for this example, since it is just an adaptation of [40]. Our goal is to synthesize a safety controller for the system Σ_{τ,δ_0} with delays, for the safe set $[1.3, 1.7] \times [5.7, 5.8]$. First, we consider a shrunk safe region $[1.3+(\varepsilon_1+\varepsilon_2), 1.7-(\varepsilon_1+\varepsilon_2)] \times [5.7+(\varepsilon_1+\varepsilon_2), 5.8-(\varepsilon_1+\varepsilon_2)]$ for the symbolic model T_τ^{symb}. Then, we can apply to it various discrete techniques such as automata-theoretic synthesis [94], supervisory control of discrete event systems [81], algorithmic game theory [9], etc. Following [40, 43], we can employ an algorithm from supervisory control [81] and synthesize a set of safe switching signals that confine the dynamics of T_τ^{symb} to the shrunk safe region. This is the horizontal arrow at the bottom of Fig. 6.1. From the resulting controller, using Cor. 6.1.6, we can construct a safety controller for the original system with delays, for the safe set $[1.3, 1.7] \times [5.7, 5.8]$. For more detailed description of the resulting controllers, see [43, Fig. 3].

Chapter 7

Related Work

We will discuss related work on the methodologies we have proposed in Chapter 3–Chapter 5 so far. Then, we will discuss related work of the two-step reachability analysis as a whole.

7.1 Approximate Bisimulation for Switching Delays

In this section, we discuss related work on our methodology proposed in Chapter 3, where we used approximate bisimulation to find an error bound between a delayed system and its delay-free model. The notion of approximate bisimulation is first introduced in [41]. The classical bisimulation relation [71,74] is the equivalence relation between transition systems, that requires the external behavior of the two systems to be identical. Since this exact bisimulation is too restrictive for the systems that have continuous state space, the notion of approximate bisimulation does not impose the identical behavior and allows the possibility of error. Use of incremental stability as a source of approximate bisimulations is advocated in [77]. This useful technique has found its applications in a variety of system classes as well as in a variety of problems. A notable application is discretization of continuous state spaces to employ discrete verification/synthesis techniques. The original framework in [77] has been extended to switched systems [43], systems with disturbance [80], and so on. A comprehensive framework where discrete control synthesis is integrated is presented in [39]; the works discussed so far are nicely summarized in the overview paper [42].

In [59], they further generalize approximate bisimulation by allowing some mismatches on the transition labels. By regarding transition labels as time, it gives a framework that allows a kind of time-delays. However, the delays captured by this methodology can accumulate as the switching occurs repeatedly. This accumulation of delays does not suit our application scenario described in Chapter 1. In our framework, we put time in the states, rather than the transition labels, so that the accumulation of delays is prohibited.

The work in [78,79] is relevant to ours, which addresses the issue of time-delay. The work [79] deals with fixed time-delays and the one [78] considers unknown time-delays. The goal of these works, which is different from ours, is to construct a comprehensive symbolic (discretized) model that encompasses all possible delays and switching signals. In particular, possible delays are thought of as disturbances (i.e. demonic/adversarial nondeterminism) and consequently, they use *alternating* approximate bisimulations introduced in [80]. The main techni-

cal gadget in doing so is a spline-based finitary approximation of continuous-time signals. The work in [18,96] also studies symbolic abstraction of hybrid systems with delays based on alternating approximate bisimulations, assuming delays are discretized.

A recent line of works [61,62] tackles the challenge of time-delays too. They take *timing contracts* as specifications; and study a verification problem [61], and a scheduling problem under the single-processor multiple-task setting [62]. A crucial difference from the current work is that they assume linear dynamics, while we can deal with nonlinear dynamics (under the assumption of incremental stability).

7.2 Skorokhod Distance Caused by Switching Delays

In Chapter 4, we have extended the methodology proposed in Chapter 3 so that it allows some timing mismatches. Quantifying the closeness between trajectories that allows timing mismatches has recently been studied in the field of conformance testing. A beginning of the study in this direction is the conformance degree based on $(T, J, (\tau, \varepsilon))$-closeness introduced in [1,2]. In this definition of closeness, the parameter τ is the closeness in time and ε is the closeness in space.

In [2], the authors present two algorithms to compute the conformance degree between given two systems. The first algorithm is based on Rapidly-exploring Random Trees (RRTs) [25]. One major difference from ours is that it estimates the conformance degree by computing its underapproximation, not overapproximation. Another difference is that it assumes that a controller is given. Thus, we cannot apply this algorithm to the control synthesis scenario that we discussed in Chapter 6, for example. The second algorithm they introduced computes an overapproximation of the conformance degree as our proposed method. However, their algorithm is only applicable to linear switched systems. Ours can analyze incrementally stable nonlinear switched systems, by restricting ourselves to error analysis caused by switching delays.

Then in [34], the conformance between two trajectories was defined using the Skorokhod metric [4], which is related to Fréchet distance as discussed in [68,69]. An algorithm to compute the Skorokhod metric between given traces is introduced also in [34]. In each of [34,68,69], the Skorokhod metric is computed for some restricted type of traces (or the set of traces). A major difference from our proposed method is that they compute the Skorokhod metric between "given" traces (or trajectories). This is because their use case scenario is for conformance testing, where the traces are obtained by simulation. In our setting, the computation of the trace, which is hard for nonlinear ODEs, is not needed. Instead, we input the models and output an upper bound of the Skorokhod metric between the traces generated by the models.

In [32], a relaxation of the strict order-preservation condition for retiming to weak order-preservation is introduced. This relaxation is natural for hybrid settings such as ours, and adaptation of this relaxation may improve our result in Chapter 4.

7.3 Extension of Abstract Interpretation with Infinitesimals

In Chapter 5, we introduced an extension of abstract interpretation by Cousot & Cousot [28], with Robinson's nonstandard analysis (NSA) [84] for verification of hybrid systems. There has been a lot of research work for verification of hybrid systems, both from formal methods and control theory. One of the most successful work is the model of *timed automata* [7], an extension of usual finite automata with the notion of time. Technically, the extension is done by introducing real-valued *clocks*. For timed automata, the reachability problem is known to be decidable [7]. Abstract interpretation and symbolic model checking of real-time systems are also studied. For example, in [54], symbolic model checking of timed automata based on *zones* is introduced. In [47], delays in synchronous programs that express real-time systems are analyzed using abstract interpretation. There are some tools available for verification of timed automata, such as UPPAAL [64] and Kronos [19].

Since clock values increase at the same speed, we cannot model complicated continuous behavior in timed automata. A more general modeling methodology that allows us to model complicated dynamics is *hybrid automata* [5, 6]. A hybrid automaton is another extension of a finite automaton, in which a state corresponds to a continuous behavior expressed as ODEs, and a state transition corresponds to a discrete change of the control mode. In fact, the model of timed automata is a subclass of hybrid automata. The reachability of hybrid automata is undecidable even for linear hybrid automata.

There are quite a few hybrid system verification tools, including HyTech [53], PHAVer [37], SpaceEx [38], HySAT/iSAT [36], d/dt [10], CheckMate [89], Flow* [21] and KeYmaera [76]. KeYmaera is based on differential dynamic logic introduced in [75]; it is a kind of dynamic logic and verifies a hybrid system in a deductive manner. The other tools listed above are aiming at model checking or flowpipe construction of hybrid systems. There are also not a few invariant generation techniques for hybrid systems, e.g. in [66, 85–87]. All these rely on ODEs (or the explicit solutions of them) for expressing continuous dynamics, much like hybrid automata do.

Our nonstandard static analysis approach is completely different from those in the following point: we do not use ODEs at all, and model hybrid systems as an imperative program with an infinitesimal constant. Our usage of NSA to extend formal methods for hybrid systems is based on [50,91,92], and continuous flow is regarded as infinitely many infinitesimal jumps based on NSA. In [91], the framework WHILE$^{\mathtt{dt}}$, ASSN$^{\mathtt{dt}}$ and HOARE$^{\mathtt{dt}}$ is introduced. They do not define the semantics of while loop command in WHILE$^{\mathtt{dt}}$ as the least fixed point directly, but define it based on their original idea of "sectionwise execution." When transferring meta-theorems of abstract interpretation, it is more convenient if the semantics of WHILE$^{\mathtt{dt}}$ is given as the least fixed point. This was done in [63]. In [50], some invariant generation techniques are transferred and an automated HOARE$^{\mathtt{dt}}$ analyzer is introduced. In [92], a stream processing language is extended with infinitesimals. More recently, a denotational semantics of a functional programming language with infinitesimals is introduced in [73]. The nonstandard static analysis approach enables us to apply static methodologies

for discrete systems as they are.

The most relevant tool to our methodology is PHAVer, based on the work in [48, 49]. In [48, 49], they also use the domain of convex polyhedra to over-approximate the reachable sets of linear hybrid automata. We separate the discrete variables (p and s) and the continuous variables (x and l) in §5.3. This corresponds to make a hybrid automaton from the given WHILE$^{\mathtt{dt}}$ program, by separating the discrete and continuous behavior. The difference between our nonstandard abstract interpretation and [48, 49] that applies abstract interpretation on the domain of convex polyhedra to hybrid automata is the following. In our framework, continuous flow is interpreted as infinitely many infinitesimal jumps and the abstract interpretation techniques can be applied to a continuous behavior as they are, even if the behavior is not piecewise linear. In the approach of [48, 49], they need some special techniques such as linear phase-portrait [52], to reduce piecewise affine dynamics into piecewise linear one. In addition, it cannot analyze the dynamics beyond piecewise affine (such as the nonlinear water tank example we have used). Instead, we have to introduce the "uniformity" condition to guarantee the finite step convergence of iteration sequences.

7.4 Two-step Reachability Analysis of Switched Systems with Delays

In this section, we discuss related work on our two-step reachability methodology as a whole. Our framework is a combination of an abstraction of the effect of delays and a hybrid system reachability analysis that cannot deal with delays. The first step overapproximates the effect of delays to reduce the reachability analysis of a delayed hybrid system to that of the delay-free model. The proposed techniques for the reduction rely on the assumption of the incremental stability δ-GUAS. The precision loss in the first reduction is determined by how good Lyapunov function we can find.

To the best of our knowledge, the only existing work that can do reachability analysis or safe controller synthesis of nonlinear hybrid systems with delays is a combination of a symbolic abstraction methodology based on approximate bisimulation (e.g. [18, 78, 96] as we discussed in §7.1) with reachability analysis or controller synthesis of the obtained discrete systems.

In [18, 78, 96], approximate bisimulation-based frameworks for symbolic abstraction of the state space have been studied. The assumption of incremental stability is essentially the same as our framework. For example, in [78], its application to safe controller synthesis is explained as follows. The results in [78] yield a symbolic model as a two-player finite-state game $\mathcal{G}$ where angelic moves switching signals and demonic moves are time-delays. By solving the game $\mathcal{G}$ (e.g. by the algorithm in [60]) one obtains a control strategy.

The biggest difference of them from our two-step framework is that they do not abstract away the effect of delays. Instead, they make a symbolic model that takes all the possible delays into consideration. It seems that our two-step workflow has an advantage in complexity: by considering all the possible delays and switching signals, the obtained symbolic model tends to have a big number of transitions. It has to be noted, however, that the workflow following [18,78,96] applies to a greater variety of systems (than switched systems) and a resulting

control strategy can be more fine-grained (reacting to delays, while our controller always assumes the worst time-delays). One can make the proximity as small as desired, in a trade-off with the size of the symbolic model. In our results, the precision loss in abstracting away the delays is fixed from the given Lyapunov functions. Numerical comparison of the precision and scalability of the analysis using some benchmark examples is future work.

Chapter 8

Conclusions

In this thesis, we have tackled the reachability analysis of hybrid systems that include time-delays. For this purpose, we have proposed a two-step abstraction methodology. The first step of the analysis reduces the reachability of hybrid systems including delays to that of its delay-free model. The obtained delay-free model is a usual hybrid system without delays, and its reachability is analyzed in the second step. Note that the second step of the proposed two-step framework is the usual reachability analysis and there is a lot of existing work for it.

As a concrete methodology for the reduction in the first step, we have introduced a relational abstraction technique based on approximate bisimulation, to obtain an upper bound of the pointwise distance between the delayed system and its delay-free model in Chapter 3. We have also introduced an extension of the previous methodology to bound the Skorokhod distance instead of the pointwise distance in Chapter 4. Both of them can be applied to possibly nonlinear incrementally stable hybrid systems. The precision of the obtained error bounds depends on given Lyapunov functions. For the reachability analysis of delay-free hybrid systems in the second step, we have proposed a predicate abstraction methodology by extending abstract interpretation on the domain of convex polyhedra with an infinitesimal constant in Chapter 5. It can be applied to possibly nonlinear hybrid systems. Soundness of the analysis is proved by the transfer principle. Termination is also proved, but only for uniform widening operators. Soundness of the two-step analysis as a whole can be guaranteed by the fact that the obtained bound in the first step is an upper bound, and that the reachability analysis in the second step is sound.

We have successfully analyzed the reachability of the example of nonlinear water tank with delays, by combining the proposed methodologies in Chapter 4 and Chapter 5. We have also shown that a safety controller for the example of boost DC-DC converter with delays can be synthesized by combining the methodology in Chapter 4 and the one proposed in [40]. For both examples, we assumed that Lyapunov functions to ensure δ-GUAS are given.

Let us conclude this thesis with the summary and the future work of each of the concrete methodologies we have introduced in Chapter 3 to Chapter 5.

8.1 Conclusions and Future Work on Approximate Bisimulations for Switching Delays

In Chapter 3, based on the results in [43], we have introduced an approximate bisimulation framework and provided upper bounds for the pointwise errors that arise from switching delays in switched systems. The proposed methodology is applicable to possibly nonlinear dynamics under the assumption of δ-GUAS. Our focus on switched systems allows us to use the same incremental stability notion (δ-GUAS) as in [43] as an ingredient for an approximate bisimulation. This is an advantage in the two-step control synthesis workflow for switched systems with delays in §6.3.

In Chapter 4, we extended the methodology to bound the Skorokhod distance, instead of the pointwise metric. It gives us more precise error bounds for reachability analysis. It also enables us to deal with the cases where the delays might be longer than the switching period τ.

The first direction of future work is regarding the assumption of the incremental stability δ-GUAS. In the results, we assumed that a common δ-GAS function or multiple δ-GAS Lyapunov functions are given, as in [43] and other symbolic abstraction methodologies based on approximate bisimulation. The obtained upper bounds by our methodologies rely on these Lyapunov functions, and for better precision, we need to find Lyapunov functions that have better characteristics. Finding Lyapunov functions for δ-GUAS itself is a challenging task. For linear systems, it is well-known that a quadratic template often works well and SDP optimization can be used to find parameters for that. We need to investigate how to find a Lyapunov function for nonlinear systems, such as the work in [17]. Moreover, in [98], the authors introduced a methodology to establish an approximate simulation relation for constructing a symbolic model from a nonlinear control system without incremental stability assumption. Instead, they rely on the assumption of *incremental forward completeness*. It is also future work to check if the same assumption of incremental forward completeness can be used in our framework to deal with delays.

Secondly, since we have given an upper bound of the Skorokhod distance in Chapter 4, it is natural to apply the result to verification of temporal logic specifications other than reachability. For example, we are considering combining our result with the one in [82], for controller synthesis from STL specifications.

This direction of future work is related to our Global Design Workshop (GDWS), a requirement of the Graduate Program for Social ICT Global Creative Leaders (GCL). As GDWS, we organized a workshop that aims at examining the challenges in writing down formal temporal specifications extensively. Note that we did not aim at obtaining statistical data, and therefore the results are just informal suggestions from the observation of the participants. The main suggestions from the observation in the workshop include the effectiveness of support tools such as ViSpec [56] and the possibility of extension of the dual language approach [14]. Making a user-friendly environment for verification of hybrid systems with delays with respect to temporal specifications using these examinations is future work.

Extending the current results to a broader class of systems is another important direction of future work. In particular, we are interested in disturbances

and the consequent use of alternating approximate bisimulation [80]. Such an extension should be carefully devised so that the two-step workflow will remain valid.

8.2 Conclusions and Future Work on Extension of Abstract Interpretation with Infinitesimals

In Chapter 5, we have presented an extended abstract interpretation framework in which hybrid systems are *exactly* modeled as WHILE$^{\mathtt{dt}}$ programs that contain infinitesimal constants. In spite of the additional constant `dt` for an infinitesimal number, they can be soundly analyzed by our extended abstract interpretation. To ensure its termination, however, there is a gap from standard abstract interpretation—it is proved only for uniform widening operators. These meta-theorems (namely, soundness and termination) are established by using the logical infrastructure of NSA. We implemented a prototype analyzer that automates the nonstandard abstract interpretation; it currently supports the domain of convex polyhedra.

Regrettably, our current implementation is premature and does not compare—in precision or scalability—with the state-of-the-art tools for hybrid system reachability such as SpaceEx [38] and Flow* [21]. In fact, the two examples in §5.4.2 are the only ones that we have so far succeeded to analyze. For other complicated examples—especially nonlinear ones, to which our framework is applicable in principle—the analysis results are too imprecise to be useful. There are some possible directions of future work to enhance the precision and scalability.

The first direction is to make use of existing techniques in the domain of convex polyhedra in the standard abstract interpretation. For example, we could utilize narrowing operators (the use of narrowing operators in the domain of convex polyhedra is indicated in [51, §3.4]) to enhance the precision of the analysis. To guarantee the soundness and the termination of the analysis, we may need to impose new technical conditions as we restricted ourselves to uniform widening operators for termination.

The second possible direction is regarding hyperwidening operators and uniformity condition. Uniformity is a sufficient condition which ensures that a widening operator is also a widening operator when transferred. There might be widening operators in the nonstandard setting with infinitesimals that are not transferred uniform widening operators. In addition, even though the finite step convergence is not guaranteed, hyperwidening operators can be used as *extrapolation* operators in the nonstandard setting. We might make use of such operators.

Thirdly, we believe other abstract domains than convex polyhedra would be useful for scalability and precision. A lot of abstract domains are left unextended to our nonstandard setting: interval domain [27, 28], pentagons [67], octagons [72], ellipsoids [35], trace partitioning abstract domain [70, 83], etc. Moreover, these domains can be combined (see [45]). It would be also useful to come up with some new abstract domains that are tailored to nonlinear dynamics. In this regard, the uniformity of the widening operators for the abstract domains other than convex polyhedra must be checked.

Finally, the lack of scalability of our prototype analyzer is mainly due to our current way of eliminating `dt` (namely via Prop. 5.4.1): it relies on *quantifier elimination (QE)* that is highly expensive. We should improve our theory to make numerical methodology applicable.

References

[1] Houssam Abbas and Georgios E. Fainekos. Towards composition of conformant systems. *CoRR*, abs/1511.05273, 2015.

[2] Houssam Abbas, Hans D. Mittelmann, and Georgios E. Fainekos. Formal property verification in a conformance testing framework. In *Twelfth ACM/IEEE International Conference on Formal Methods and Models for Codesign, MEMOCODE 2014, Lausanne, Switzerland, October 19-21, 2014*, pages 155–164, 2014.

[3] Michael Perin Alexandre Marechal. Three linearization techniques for multivariate polynomials in static analysis using convex polyhedra. Technical Report TR-2014-7, Verimag Research Report.

[4] Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry and Applications*, 05(01n02):75–91, 1995.

[5] Rajeev Alur, Costas Courcoubetis, Nicolas Halbwachs, Thomas A. Henzinger, Pei-Hsin Ho, Xavier Nicollin, Alfredo Olivero, Joseph Sifakis, and Sergio Yovine. The algorithmic analysis of hybrid systems. *Theor. Comput. Sci.*, 138(1):3–34, 1995.

[6] Rajeev Alur, Costas Courcoubetis, Thomas A. Henzinger, and Pei-Hsin Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, pages 209–229, 1992.

[7] Rajeev Alur and David L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126:183–235, 1994.

[8] David Angeli. A Lyapunov approach to incremental stability properties. *IEEE Trans. Automat. Contr.*, 47(3):410–421, 2002.

[9] A. Arnold, A. Vincent, and I. Walukiewicz. Games for synthesis of controllers with partial observation. *Theor. Comput. Sci.*, 303(1):7–34, June 2003.

[10] Eugene Asarin, Thao Dang, and Oded Maler. The d/dt tool for verification of hybrid systems. In Ed Brinksma and Kim Guldstrand Larsen, editors, *Computer Aided Verification: 14th International Conference, CAV 2002 Copenhagen, Denmark, July 27–31, 2002 Proceedings*, pages 365–370, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[11] Bacciotti, Andrea and Ceragioli, Francesca. Stability and stabilization of discontinuous systems and nonsmooth Lyapunov functions. *ESAIM: COCV*, 4:361–376, 1999.

[12] R. Bagnara, P. M. Hill, and E. Zaffanella. The Parma Polyhedra Library: Toward a complete set of numerical abstractions for the analysis and verification of hardware and software systems. *Science of Computer Programming*, 72(1–2):3–21, 2008.

[13] Roberto Bagnara, Patricia M. Hill, Elisa Ricci, and Enea Zaffanella. Precise widening operators for convex polyhedra. *Sci. Comput. Program.*, 58(1-2):28–56, 2005.

[14] Luciano Baresi, Alessandro Orso, and Mauro Pezzè. Introducing formal specification methods in industrial practice. In *Proceedings of the 19th International Conference on Software Engineering*, ICSE '97, pages 56–66, 1997.

[15] Romain Beauxis and Samuel Mimram. A non-standard semantics for Kahn networks in continuous time. In *CSL*, pages 35–50, 2011.

[16] A. G. Beccuti, G. Papafotiou, and M. Morari. Optimal control of the boost dc-dc converter. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4457–4462, Dec 2005.

[17] Ruxandra Bobiti and Mircea Lazar. A sampling approach to finding Lyapunov functions for nonlinear discrete-time systems. In *Control Conference (ECC), 2016 European*, pages 561–566. IEEE, 2016.

[18] Alessandro Borri, Giordano Pola, and Maria D. Di Benedetto. A symbolic approach to the design of nonlinear networked control systems. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, HSCC '12, pages 255–264, 2012.

[19] Marius Bozga, Conrado Daws, Oded Maler, Alfredo Olivero, Stavros Tripakis, and Sergio Yovine. Kronos: A model-checking tool for real-time systems. In *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, pages 546–550, 1998.

[20] Michael S Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on automatic control*, 43(4):475–482, 1998.

[21] Xin Chen, Erika Ábrahám, and Sriram Sankaranarayanan. Flow*: An analyzer for non-linear hybrid systems. In *Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings*, pages 258–263, 2013.

[22] N.V. Chernikova. Algorithm for finding a general formula for the non-negative solutions of a system of linear equations. *USSR Computational Mathematics and Mathematical Physics*, 4(4):151–158, 1964.

[23] N.V. Chernikova. Algorithm for finding a general formula for the non-negative solutions of a system of linear inequalities. *USSR Computational Mathematics and Mathematical Physics*, 5(2):228–233, 1965.

[24] N.V. Chernikova. Algorithm for discovering the set of all the solutions of a linear programming problem. *USSR Computational Mathematics and Mathematical Physics*, 8(6):282–293, 1968.

[25] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, Pittsburgh, PA, June 2005.

[26] P. Cousot. Semantic foundations of program analysis. In S.S. Muchnick and N.D. Jones, editors, *Program Flow Analysis: Theory and Applications*, chapter 10, pages 303–342. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1981.

[27] Patrick Cousot and Radhia Cousot. Static determination of dynamic properties of programs. *Proceedings of the second International Symposium on Programming, Paris, France, April 13-15, 1976*, pages 106–130, 1976.

[28] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pages 238–252, 1977.

[29] Patrick Cousot and Radhia Cousot. Abstract interpretation frameworks. *J. Log. Comput.*, 2(4):511–547, 1992.

[30] Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, David Monniaux, and Xavier Rival. The ASTREÉ analyzer. In *Programming Languages and Systems, 14th European Symposium on Programming,ESOP 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, pages 21–30, 2005.

[31] Patrick Cousot and Nicolas Halbwachs. Automatic discovery of linear restraints among variables of a program. In *Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978*, pages 84–96, 1978.

[32] J. M. Davoren. Epsilon-tubes and generalized Skorokhod metrics for hybrid paths spaces. In Rupak Majumdar and Paulo Tabuada, editors, *Hybrid Systems: Computation and Control: 12th International Conference, HSCC 2009, San Francisco, CA, USA, April 13-15, 2009. Proceedings*, pages 135–149, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[33] Nachum Dershowitz and Zohar Manna. Proving termination with multiset orderings. In *Automata, Languages and Programming, 6th Colloquium, Graz, Austria, July 16-20, 1979, Proceedings*, pages 188–202, 1979.

[34] Jyotirmoy V. Deshmukh, Rupak Majumdar, and Vinayak S. Prabhu. Quantifying conformance using the Skorokhod metric. In Daniel Kroening and Corina S. Păsăreanu, editors, *Computer Aided Verification: 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part II*, pages 234–250, Cham, 2015. Springer International Publishing.

[35] Jérôme Feret. Static analysis of digital filters. In *Programming Languages and Systems, 13th European Symposium on Programming, ESOP 2004, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2004, Barcelona, Spain, March 29 - April 2, 2004, Proceedings*, pages 33–48, 2004.

[36] Martin Fränzle, Christian Herde, Tino Teige, Stefan Ratschan, and Tobias Schubert. Efficient solving of large non-linear arithmetic constraint systems with complex boolean structure. *JSAT*, 1(3-4):209–236, 2007.

[37] Goran Frehse. PHAVer: Algorithmic verification of hybrid systems past HyTech. In *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, pages 258–273, 2005.

[38] Goran Frehse, Colas Le Guernic, Alexandre Donzé, Scott Cotton, Rajarshi Ray, Olivier Lebeltel, Rodolfo Ripado, Antoine Girard, Thao Dang, and Oded Maler. SpaceEx: Scalable verification of hybrid systems. In *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, pages 379–395, 2011.

[39] Antoine Girard. Synthesis using approximately bisimilar abstractions: state-feedback controllers for safety specifications. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*, pages 111–120, 2010.

[40] Antoine Girard. Controller synthesis for safety and reachability via approximate bisimulation. *Automatica*, 48(5):947–953, 2012.

[41] Antoine Girard and George J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Trans. Automat. Contr.*, 52(5):782–798, 2007.

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

[43] Antoine Girard, Giordano Pola, and Paulo Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Trans. Automat. Contr.*, 55(1):116–126, 2010.

[44] R. Goldblatt. *Lectures on the Hyperreals: An Introduction to Nonstandard Analysis*. Graduate Texts in Mathematics. Springer New York, 1998.

[45] Sumit Gulwani and Ashish Tiwari. Combining abstract interpreters. In *Proceedings of the ACM SIGPLAN 2006 Conference on Programming Language Design and Implementation, Ottawa, Ontario, Canada, June 11-14, 2006*, pages 376–386, 2006.

[46] Nicolas Halbwachs. *Dtermination automatique de relations linaires vrifies par les variables d'un programme*. Thse de 3e cycle, Universit Scientifique et Mdicale de Grenoble, 1979.

[47] Nicolas Halbwachs. Delay analysis in synchronous programs. In *Computer Aided Verification, 5th International Conference, CAV '93, Elounda, Greece, June 28 - July 1, 1993, Proceedings*, pages 333–346, 1993.

[48] Nicolas Halbwachs, Yann-Eric Proy, and Pascal Raymond. Verification of linear hybrid systems by means of convex approximations. In *SAS*, pages 223–237, 1994.

[49] Nicolas Halbwachs, Yann-Erick Proy, and Patrick Roumanoff. Verification of real-time systems using linear relation analysis. *Formal Methods in System Design*, 11(2):157–185, 1997.

[50] Ichiro Hasuo and Kohei Suenaga. Exercises in nonstandard static analysis of hybrid systems. In *Computer Aided Verification - 24th International Conference, CAV 2012, Berkeley, CA, USA, July 7-13, 2012 Proceedings*, pages 462–478, 2012.

[51] Kim S. Henriksen, Gourinath Banda, and John P. Gallagher. Experiments with a convex polyhedral analysis tool for logic programs. *CoRR*, abs/0712.2737, 2007.

[52] Thomas A. Henzinger and Pei-Hsin Ho. Algorithmic analysis of nonlinear hybrid systems. In *Computer Aided Verification, 7th International Conference, Liège, Belgium, July, 3-5, 1995, Proceedings*, pages 225–238, 1995.

[53] Thomas A. Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. HYTECH: A model checker for hybrid systems. *STTT*, 1(1-2):110–122, 1997.

[54] Thomas A. Henzinger, Xavier Nicollin, Joseph Sifakis, and Sergio Yovine. Symbolic model checking for real-time systems. In *Proceedings of the Seventh Annual Symposium on Logic in Computer Science (LICS '92), Santa Cruz, California, USA, June 22-25, 1992*, pages 394–406, 1992.

[55] C. A. R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10):576–580, 1969.

[56] Bardh Hoxha, Nikolaos Mavridis, and Georgios Fainekos. Vispec: A graphical tool for elicitation of mtl requirements. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 3486–3492. IEEE, 2015.

[57] A.E. Hurd and P.A. Loeb. *An Introduction to Nonstandard Real Analysis*. Pure and Applied Mathematics. Elsevier Science, 1985.

[58] Jerome Jouffroy and Thor I Fossen. A tutorial on incremental stability analysis using contraction theory. *Modeling, Identification and control*, 31(3):93, 2010.

[59] A. Agung Julius, Alessandro D'Innocenzo, Maria Domenica Di Benedetto, and George J. Pappas. Approximate equivalence and synchronization of metric transition systems. *Systems & Control Letters*, 58(2):94–101, 2009.

[60] Marcin Jurdzinski. Small progress measures for solving parity games. In *STACS*, pages 290–301, 2000.

[61] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Verification and synthesis of timing contracts for embedded controllers. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control, HSCC 2016, Vienna, Austria, April 12-14, 2016*, pages 115–124, 2016.

[62] Mohammad Al Khatib, Antoine Girard, and Thao Dang. Scheduling of embedded controllers under timing contracts. In *Proceedings of the 20th International Conference on Hybrid Systems: Computation and Control, HSCC 2017, Pittsburgh, PA, USA, April 18-20, 2017*, pages 131–140, 2017.

[63] Kengo Kido. *An Alternative Denotational Semantics for an Imperative Language with Infinitesimals*. Bachelor's thesis, The University of Tokyo: Japan, 2013.

[64] Kim G. Larsen, Paul Pettersson, and Wang Yi. Uppaal in a nutshell. *International Journal on Software Tools for Technology Transfer*, 1(1):134–152, Dec 1997.

[65] H. Le Verge. A note on Chernikova's algorithm. Technical Report 635, IRISA, Rennes, France, February 1992.

[66] Edward A. Lee and Haiyang Zheng. Operational semantics of hybrid systems. In *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, pages 25–53, 2005.

[67] Francesco Logozzo and Manuel Fähndrich. Pentagons: a weakly relational abstract domain for the efficient validation of array accesses. In *Proceedings of the 2008 ACM Symposium on Applied Computing (SAC), Fortaleza, Ceara, Brazil, March 16-20, 2008*, pages 184–188, 2008.

[68] Rupak Majumdar and Vinayak S. Prabhu. Computing the Skorokhod distance between polygonal traces. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, HSCC '15, pages 199–208, New York, NY, USA, 2015. ACM.

[69] Rupak Majumdar and Vinayak S. Prabhu. Computing distances between reach flowpipes. In *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, HSCC '16, pages 267–276, New York, NY, USA, 2016. ACM.

[70] Laurent Mauborgne and Xavier Rival. Trace partitioning in abstract interpretation based static analyzers. In *Programming Languages and Systems, 14th European Symposium on Programming,ESOP 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, pages 5–20, 2005.

[71] R. Milner. *Communication and Concurrency*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[72] Antoine Miné. The octagon abstract domain. *Higher-Order and Symbolic Computation*, 19(1):31–100, 2006.

[73] Hirofumi Nakamura, Kensuke Kojima, Kohei Suenaga, and Atsushi Igarashi. A nonstandard functional programming language. In *Programming Languages and Systems - 15th Asian Symposium, APLAS 2017, Suzhou, China, November 27-29, 2017, Proceedings*, pages 514–533, 2017.

[74] D. M. R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings 5th GI Conference on Theoretical Computer Science*, volume 104 of *LNCS*, pages 15–32. Springer, Berlin, 1981.

[75] André Platzer. Differential dynamic logic for hybrid systems. *J. Autom. Reasoning*, 41(2):143–189, 2008.

[76] André Platzer and Jan-David Quesel. KeYmaera: A hybrid theorem prover for hybrid systems. In Alessandro Armando, Peter Baumgartner, and Gilles Dowek, editors, *IJCAR*, volume 5195 of *LNCS*, pages 171–178. Springer, 2008.

[77] Giordano Pola, Antoine Girard, and Paulo Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44(10):2508–2516, 2008.

[78] Giordano Pola, Pierdomenico Pepe, and Maria Domenica Di Benedetto. Alternating approximately bisimilar symbolic models for nonlinear control systems with unknown time-varying delays. In *Proceedings of the 49th IEEE Conference on Decision and Control, CDC 2010, December 15-17, 2010, Atlanta, Georgia, USA*, pages 7649–7654, 2010.

[79] Giordano Pola, Pierdomenico Pepe, Maria Domenica Di Benedetto, and Paulo Tabuada. Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Systems & Control Letters*, 59(6):365–373, 2010.

[80] Giordano Pola and Paulo Tabuada. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM J. Control and Optimization*, 48(2):719–733, 2009.

[81] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1):206–230, January 1987.

[82] Vasumathi Raman, Alexandre Donzé, Dorsa Sadigh, Richard M. Murray, and Sanjit A. Seshia. Reactive synthesis from signal temporal logic specifications. In *Proceedings of the 18th International Conference on Hybrid*

Systems: Computation and Control, HSCC '15, pages 239–248, New York, NY, USA, 2015. ACM.

[83] Xavier Rival and Laurent Mauborgne. The trace partitioning abstract domain. *ACM Trans. Program. Lang. Syst.*, 29(5), 2007.

[84] A. Robinson. *Non-standard Analysis*. Studies in logic and the foundations of mathematics. North-Holland Pub. Co., 1966.

[85] Enric Rodríguez-Carbonell and Ashish Tiwari. Generating polynomial invariants for hybrid systems. In *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, pages 590–605, 2005.

[86] Sriram Sankaranarayanan. Automatic invariant generation for hybrid systems using ideal fixed points. In *Proceedings of the 13th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2010, Stockholm, Sweden, April 12-15, 2010*, pages 221–230, 2010.

[87] Sriram Sankaranarayanan, Henny B. Sipma, and Zohar Manna. Constructing invariants for hybrid systems. *Formal Methods in System Design*, 32(1):25–55, 2008.

[88] Jianhua Shi, Jiafu Wan, Hehua Yan, and Hui Suo. A survey of cyber-physical systems. In *2011 International Conference on Wireless Communications & Signal Processing, WCSP 2011, Nanjing, China, November 9-11, 2011*, pages 1–6, 2011.

[89] B Izaias Silva, Keith Richeson, Bruce Krogh, and Alongkrit Chutinan. Modeling and verifying hybrid dynamic systems using CheckMate. In *ADPM 2000: 4th International Conference on Automation of Mixed Processes: Hybrid Dynamic Systems*, 2000.

[90] H.H. Sohrab. *Basic Real Analysis*. Basic Real Analysis. Birkhäuser Boston, 2003.

[91] Kohei Suenaga and Ichiro Hasuo. Programming with infinitesimals: A while-language for hybrid system modeling. In *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part II*, pages 392–403, 2011.

[92] Kohei Suenaga, Hiroyoshi Sekine, and Ichiro Hasuo. Hyperstream processing systems: nonstandard modeling of continuous-time signals. In *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, pages 417–430, 2013.

[93] The MathWorks, Inc. Passive control of water tank level (r2017b). `https://www.mathworks.com/help/control/examples/passive-control-of-water-tank-level.html`, 2017. [Online; accessed Oct. 2, 2017].

[94] Moshe Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Logics for Concurrency - Structure versus Automata (8th Banff Higher Order Workshop, August 27 - September 3, 1995, Proceedings)*, pages 238–266, 1995.

[95] Glynn Winskel. *The Formal Semantics of Programming Languages: An Introduction*. MIT Press, Cambridge, MA, USA, 1993.

[96] M. Zamani, M. Mazo Jr, M. Khaled, and A. Abate. Symbolic abstractions of networked control systems. *IEEE Transactions on Control of Network Systems*, PP(99):1–1, 2017.

[97] Majid Zamani and Rupak Majumdar. A lyapunov approach in incremental stability. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 302–307. IEEE, 2011.

[98] Majid Zamani, Giordano Pola, Manuel Mazo, and Paulo Tabuada. Symbolic models for nonlinear control systems without stability assumptions. *IEEE Transactions on Automatic Control*, 57(7):1804–1809, 2012.