City Research Online

City, University of London Institutional Repository

Citation: Ul Asad, H. (2016). Formal verification of analog and mixed signal circuits using deductive and bounded approaches. (Unpublished Doctoral thesis, City University London)

This is the accepted version of the paper.

This version of the publication may differ from the final published version.

Permanent repository link: https://openaccess.city.ac.uk/id/eprint/15185/

Link to published version:

Copyright: City Research Online aims to make research outputs of City, University of London available to a wider audience. Copyright and Moral Rights remain with the author(s) and/or copyright holders. URLs from City Research Online may be freely distributed and linked to.

Reuse: Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

City Research Online: http://openaccess.city.ac.uk/ publications@city.ac.uk

Formal Verification of Analog and Mixed Signal Circuits using Deductive and Bounded Approaches

Hafiz ul Asad

School of Mathematics, Computer Science and Engineering

City University London

A Thesis submitted in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

Jan 2016

To my mother, without her prayers this would have not been possible. To my wife and son, for their emotional support during the course of this Phd. To my late father, who would have been really proud of my Phd.

Acknowledgements

I would like to offer my deepest gratitude to my advisor Professor Kevin D. Jones for offering me the opportunity to conduct this research. This PhD was his brainchild, and I am indebted for his guidance, motivation, and help throughout the course of this PhD. I believe, his way of conducting my supervision has enabled me to become an independent researcher.

I would like to thank Dr. Peter Popov, for first agreeing to take over from Kevin as my supervisor, and then making sure that I finished my PhD well within time. I am also grateful to Dr. Frederic Surre for his support role as a second supervisor.

I owe a big thank you to the whole academic staff of the Centre for Software Reliability for their support and motivation. Specially, I would like to thank Dr. Kizito Salako for the useful technical discussions we have had during this period. I am also grateful to Professor Lorenzo Strigini for the support that he provided for some of my publications.

I am thankful to Mentor Graphics and in particular to Dr. Robert Hun for providing 50 % of the funding for this research.

Finally, this PhD work would have not been possible without the support of my mother Noor Jehan. I am indebted to her for allowing me to come to London and for being my strength emotionally. I am thankful to my wife Shafqat and son Kumail Asad for their love and emotional support that enabled me fulfilling my goal of completing this PhD.

Abstract

This thesis presents novel formal verification techniques to verify the important property of inevitability of states in analog and mixed signal (AMS) circuits. Two techniques to verify the inevitability of phase locking in a Charge Pump Phase Lock Loop (PLL) circuit are presented: mixed deductive-bounded and deductive-only verification approaches. The deductive-bounded approach uses Lyapunov-like certificates with bounded advection of sets to verify the inevitability of phase locking. The deductive-only technique uses a combination of Lyapunov and Escape certificates to verify the inevitability property. Both deductive-only and deductive-bounded verification approaches involve positivity/negativity checks of polynomials over semi-algebraic sets, which both belong to the NP-hard set of problems. The Sum of Squares (SOS) programming technique is used to transform the positivity tests of polynomials to the feasibility of semi-definite programs. The efficacy of the approach is demonstrated by verifying the inevitability of phase locking for a third and fourth order CP PLL. Similarly, the inevitability of oscillation in ring oscillators (ROs) is verified using a numeric-symbolic deductive approach. The global inevitability (of oscillation) property is specified as a conjunction of several sub-properties that are verified via different Lyapunov-like certificates in different subsets of the state space. The construction of these certificates is posed as the verification of First Order Formulas (FOFs) having Universal-Existential quantifiers. A tractable numeric-symbolic approach, based on SOS programming and Quantifier Elimination (QE), is used to verify these FOFs. The approach is applied to the verification of inevitability of oscillation in ROs with odd and even topologies.

Furthermore, frequency domain properties specification and verification for analog oscillators is presented. The behaviour of an oscillator in the frequency domain is specified, while it operates in close proximity to the desired limit cycle, employing finite Fourier series representation of a periodic signal. To be sufficiently robust enough against parameter variations, robustness of parameters is introduced in these specifications. These frequency domain properties are verified using a mixed time-frequency domain technique based on Satisfiability Modulo Ordinary Differential Equation (SMODE). The efficacy of the technique is demonstrated for the benchmark voltage controlled and tunnel diode oscillators.

Contents

List of Figures

List of Tables

Symbols and Abbreviations

β	Positive Integer
λ	Transconductance of a Transistor
$\mathbb{N}$	Set of all Natural Numbers
$\mathbb{R}$	Set of all Real Numbers
$\mathbb{R}_{\geq 0}$	Set of all Positive Real Numbers
$\mathbb{S}$	A Mathematical System
$\mathbb{Z}$	Set of all Integer Numbers
$\mathbb{Z}_{\geq 0}$	Set of all Positive Integer Numbers
$\mathbf{C}$	Cone Generated by real polynomials
$\mathbf{I}$	Ideal Generated by real polynomials
$\mathcal{AP}$	Almost Periodic
$\mathcal{C}$	Continuous flow sets
$\mathcal{D}$	Discrete Jump sets
$\mathcal{H}$	Hybrid Dynamical System
$\mathcal{M}$	Multiplicative Monoid
$\mathcal{P}_n$	Set of Positive Semidefinite Polynomials in n variables
$\mathcal{R}_n$	Set of Polynomials in n variables with real coefficients
$\mathcal{S}_n$	Set of SOS Polynomials in n variables

$\mathcal{T}$	Hybrid time
$\mathcal{Z}$	Function returning 0-sub-level-set
$\mathcal{Z}(.)$	Zero Sub-level Set
∇	$\frac{\partial}{\partial x}$
$\neg$	Boolean Negation
$\vee$	Boolean Disjunction
$\mid \mid$	Cardinality
$\mid\mid \mid\mid$	Euclidean Norm
$\wedge$	Boolean Conjunction
bd	boundary of a set
Cl	Closure
K_p	Conductance of a Transistor
L	Length of a Transistor
W	Width of a Transistor
AGI	Almost Global Inevitability
AI	Attractive Invariant
AMS	Analog and Mixed Signal
BDD	Binary Decision Diagram
BMC	Bounded Model Checking
CAD	Cylindrical Algebraic Decomposition
CDS	Continuous Dynamical System
CP	Charge Pump
CP PLL	Charge Pump Phase Lock Loop
CTL	Computational Tree Logic

DAE	Differential Algebraic Equation
Def	Definition
DNF	Disjunctive Normal Form
Eq	Equation
Fig	Figure
FOF	First Order Formula
HDS	Hybrid Dynamical System
KCL	Kirchhoff Current Law
KVL	Kirchhoff Voltage Law
LF	Low Pass Filter
LHPN	Labelled Hybrid Petri Net
LTL	Linear Temporal Logic
MILP	Mixed Integer Linear Programming
MOS	Metal Oxide Semiconductor
ODE	Ordinary Differential Equation
PFD	Phase Frequency Detector
PSD	positive-semidefinite
QE	Quantifier Elimination
RO	Ring Oscillator
SAT	Satisfiability
SMODE	Satisfiability Modulo Ordinary Differential Equation
SMT	Satisfiability Modulo Theory
SOC	System on a Chip
SOS	Sum of Squares

TDO	Tunnel Diode Oscillator
Th	Theorem
VCO	Voltage Controlled Oscillator

Chapter 1

Introduction

1.1 Background and Motivation

With the advent of system on a chip (SOC) technology more consumer electronic circuits are fabricated on chips of miniature sizes. Electronic devices making use of SOC technology are ubiquitous in our lives. They range from automobiles, air planes, computers, ATMs, medical devices, internet and mobile communication systems, banking systems, stock marketing, satellite communication, railway communication networks, security devices, smart grids, etc. Due to their enormous influence on human lives, these devices need to be designed such that they are free from all bugs and are robust enough against any process and parameter variations. SOCs being of a safety critical nature, a bug—that goes unnoticed in their design, may result in the loss of human lives. Furthermore, a huge amount of capital is involved in designing and fabricating these devices and a bug at a later stage could result in financial loss to the manufacturer. These factors necessitate the need for a rigorous verification methodology at the design stage of these devices so that potential design bugs are captured at an earlier stage of the design cycle.

SOC designs contain pure digital, pure analog, and analog and mixed signal (AMS) circuits. Conventionally, simulation and testing have been the tools validating the design of these circuits. Simulation is an approach where the mathematical model of a circuit is checked for a small number of test vectors. The design is considered to be accurate (inaccurate) if it works (does not work) for these limited test vectors, assuming it will work in all possible scenarios. Similarly, testing validates the design by checking the functionality of a physical prototype for limited test vectors. It was not until the mid-1990s that designers realized the deficiency of simulation based design

validation. There are several issues related to simulation/testing based hardware verification. Though it takes less time to capture well known design errors, it however does not cover the whole design space. It verifies circuits only for specific input stimuli, states and operating conditions (parameters, temperature etc), and therefore might not stimulate the hidden bugs at corners of the design space. Selection of these stimuli (similarly states, parameters) largely depends on the designer experience and expertise to manually select test benches. Clearly, lacking the coverage aspect of the design space, large number of simulations are carried out to cover most of the design space. Even then, the design can not be guaranteed to be free from bugs, and process variations at a later stage may cause a drastic change in circuit behaviour. It has been observed in industry that designs which were thought to be bug free, based on simulation results, turned out to have bugs which went unnoticed during simulation. The ring oscillator (RO) from Rambus [56], is a classic example where researchers found that for certain initial conditions it failed to oscillate. The pentium FDIV is another example of a bug in the floating point unit of the intel P5 processor that went unnoticed in the design phase [25]. This costed the company $500M. Therac-25 [26], a radiation therapy machine, caused several accidents of radiation overdose to patients because design bugs could not be captured by simulation. This emphasises the fact that electronic devices, being safety critical and involve huge capital, need to be verified at the design stage using rigorous verification techniques.

Complementary to simulation/testing is formal verification of a design. Formal methods overcome the deficiencies of simulation/testing by verifying a design through exhaustive checking of every possible scenario. Formal methods in hardware verification model a circuit conservatively at different levels of abstractions. The model of the circuit is then verified automatically for all possible inputs, states, and parameters. There are three well known formal techniques that have been used for hardware verification. These are equivalence checking, model checking, and theorem proving (Deductive Verification). Formal hardware verification has been very successful in validating digital hardware design, but their application in AMS circuit verification has been very limited. This is the reason that SPICE simulation has been the main verification tool for AMS circuits validation. As mentioned earlier, SPICE simulation can not check the complete design space since it takes a prohibitively long time to undertake this task. Several days and months are spent to validate small circuits like ROs and Phase Lock Loops (PLLs) using thousands of SPICE simulations. Even then, a design validated by SPICE simulation can not be guaranteed to be 100% free from bugs and can not be used in safety critical applications.

There are several bottlenecks in applying formal methods to AMS circuit verification. Being continuous in nature, it is very difficult to abstract the behaviour of AMS circuits using formal methods owing to infinite state space. Secondly, interaction between high dimensional continuous and discrete behaviours makes it difficult to formally express AMS circuits. The sensitivity of AMS circuits to physical phenomena, like temperature and process variations, can drastically modify the expected behaviour of these circuits. Furthermore, formal AMS circuit verification requires a formal property specification language to express properties of interest. Temporal logics have been successfully used for digital circuits, but their extension to AMS circuits is challenging due to the continuous time behaviour of these circuits. These difficulties mean that there has been very slow progress of formal methods in AMS circuit verification over the past two decades.

During the last decade, several works have been dedicated to the formal verification of AMS circuits. Mostly, these works modelled AMS circuits as set of ordinary differential equations (ODEs), hybrid automaton, piecewise linear switched system, difference equations, petri nets etc. Safety verification has been the well known problem taken up in these works. Time domain reachability analysis has been the main analysis tool in verifying the safety property. For most of these models, close form solutions of the ODEs do not exist and the reachability analysis is faced with the problem of decidability. Decidability has been the hardest barrier verifying the safety property using time domain reachability for these models. To overcome this hurdle, conservative approximate approaches have been used in verifying safety property of AMS circuits. Towards this goal, different set theoretic techniques have been used to approximate the solutions of ODEs. Well known set representations used are, polytopes, ellipsoids, zonotopes, and boxes. On the contrary, very little work has been dedicated to verify the inevitability (Liveness) of states (Phase locking, Oscillation) property in AMS circuits. Start up problems have been very common in AMS circuits such as ROs and PLLs. It has been observed that these circuit often fail to start after their fabrication on an IC. It is therefore very significant, from the designer point of view, to verify this property at an early stage of the design. Previously, global start up property has been verified for RO and CP PLLs. It has been verified using time domain reachability analysis. This time domain reachability technique is faced with several issues, which restricts the efficacy and scalability of the approach. The state space is partitioned and reachability of the desired state is verified for each discrete partition. To reduce the conservatism, hundreds of such partitions are used which makes the task of verifying the liveness property computationally very expensive. For CP PLL, modelled as a hybrid automaton, hun-

dreds of discrete jumps are required before the system reaches the locking state. This is the reason that time outs of the reachability tool have been reported in [105]. To get rid of these discrete transitions, [7] used a continuization technique to verify time to lock for a CP PLL. Even then, the author had to use hundreds of reach set computations for each partition of the state space. Similar reachability problems have also been reported in [66]. Furthermore, time domain reachability verifies a property for bounded time and does not say anything about what happens when time approaches infinity. The ODE equations are also explicitly solved, either exactly or approximately, and the sets are propagated along the time axis. On the contrary, deductive verification, based on certificates, verifies a property for infinite horizon with an additional advantage of avoiding the expensive discretization of the space. Moreover, ODEs are not solved explicitly and rather are abstracted by theorems characterizing the long term behaviour of their solutions.

In this dissertation, we have focussed on verifying the inevitability property for various AMS circuits. Specifically, we verify inevitability of phase locking in CP PLL and that of oscillations in ROs. A state of a system is said to be inevitable if it is invariant (once it is attained the system remains there forever) and eventually every possible system behaviour reaches this state. CP PLL are designed such that the output phase/frequency follows the phase/frequency of the reference input. We verify inevitability of phase locking in a CP PLL using an approach which is a combination of deductive and bounded verification. Being difficult to verify, we use the divide and rule strategy and split the inevitability in to the conjunction of two sub-properties. These properties are specified in two disjoint subsets of the state space. The CP PLL AMS circuit is a typical system consisting of discrete and continuous subsystems. For example, the control circuitry responsible for pumping the charge in and out of the CP PLL circuit operates in discrete steps, whereas the low pass filter, the Voltage Controlled Oscillator (VCO), and the frequency divider are all examples of continuous systems. We therefore model the CP PLL as a hybrid dynamical system, with continuous dynamics represented by ODEs, and discrete dynamics by algebraic jump equations. The first of the two sub-properties is specified such that there is a set where all system trajectories converge to the locking state. This set is called an attractive invariant set. The second sub-property is specified over the rest of the hybrid state space such that all trajectories eventually reach the attractive invariant set. In this thesis, we present two approaches verifying the inevitability property. These approaches differ in how the second sub-property is verified whereas the first property is verified similarly in both techniques. We use deductive verification to verify the attractive invariance of a set benefiting from

Lyapunov stability certificates for hybrid systems. We construct multiple Lyapunov certificates for each continuous subsystem of the CP PLL. The union of the sub-level sets characterized by the maximized level surfaces is the attractive invariant set. Convergence to this set from the set of states outside is verified following two different approaches. The first approach is a mixed deductive-bounded verification whereas the second is the deductive-only verification technique. The mixed deductive-bounded approach uses advection of sets for bounded time steps before the set of states reaches the attractive invariant set. For all those set of states which are still outside the attractive invariant set, we use the Escape certificate based deductive approach, and show that trajectories can not stay there forever and will eventually reach the attractive invariant set. The second approach is based on a pure deductive approach where we only use the Escape certificate argument for the trajectories to eventually escape the outer set and reach the attractive invariant set. Both deductive and bounded approaches involve testing of the positivity of multi-variate polynomials over semi-algebraic sets. This belongs to a set of NP-hard problems and therefore can not be solved using a sound and complete formal method. Therefore, we use the sound but incomplete Sum of Squares (SOS) programming approach to construct certificates/polynomials needed for their verification. The efficacy of our approach is demonstrated by verifying inevitability of a third and fourth order CP PLL.

Similarly, we verify inevitability of oscillation in ROs using a deductive verification methodology. We model ROs as continuous dynamical systems (CDS). Here too, we split the almost global inevitability property (a periodic limit cycle is inevitable from all but a negligible dead set of states) into the conjunction of several sub-properties. These sub-properties are defined such that they specify attractive invariance of set, Escape of trajectories from a set, eventuality of trajectories to reach a set, and global convergence to an equilibrium state. We use certificate based deductive verification for these properties. These certificates are structurally similar to the Lyapunov certificates discussed earlier. We show that a set, which is a tight over-approximation of the set enclosed by the periodic limit cycle, is attractive invariant in the sense that all trajectories outside eventually reach this set and no trajectory can ever escape this set. Within this attractive invariant set, we show that all trajectories escape a ball around the dead set and reach to within an arbitrarily small distance of the periodic limit cycle. We use certificates of Escape and Eventuality for this purpose. We demonstrate the applicability of our methodology by verifying inevitability property for odd and even stage ROs. Benefiting from the physical layout of the even stage RO, we divide its operation in differential and common modes. For the common mode of RO, we only verify that at

the steady state, all common mode voltages converge to the zero equilibrium state. We verify this using the Lyapunov certificate in an invariant differential state space. We formulate the construction of these certificates as First Order Formulas (FOFs) having polynomials, inequalities/equations and quantifiers (Universal/Existential). Though there are QE solvers that can verify these formulas, they however are very expensive for practical problems of more than two dimensions. We therefore resort to use of a numerical-symbolic strategy, and use SOS programming followed by the application of QE to construct feasible certificates in realistic computation time. SOS programming transforms the verification of these FOFs to the feasibility of a semi-definite program, which if feasible, returns a certificate within a limited numerical precision. To validate these certificates further, we use a symbolic QE tool and verify FOFs which have universal quantifiers only (the SOS program fixes the coefficients or in other words removes the existential quantifiers).

Besides time domain properties, AMS circuits designers are often interested in frequency domain properties of these circuits, for instance, robust oscillation frequency of oscillators, PLL lock up to the input frequency etc. Verifying AMS circuits in frequency domain is a difficult task, and consequently frequency domain approaches are limited to small signal AC analysis of approximate linearized models. Extension of these localized methods to non-linear models needs further research.

In this thesis, we present a novel property specification technique in frequency domain. We specify the behaviour of an oscillator using robust frequency domain specification such that it oscillates with the desired frequency and does not have undesired harmonics. Furthermore, with process and parameters variations, the oscillator circuit still complies the specification with a certain degree of robustness. Towards this goal, we use finite Fourier series approximation of the desired periodic signal. Taking care of the approximation error, we conservatively under and over-approximate the desired behaviour. To cater for parameter and process variations, we introduce a certain degree of robustness such that the desired signal satisfies the frequency domain specification with a certain degree of robustness. Instead of individual Fourier series coefficients, we use the periodogram specification, the energy content of each frequency component.

The verification of frequency domain properties is not straightforward. There are two options to perform this task. One is to carry out the verification in the frequency domain by having both the system and properties in the frequency domain, and performing the decision procedure in this domain. This is beyond the capabilities of the current state of the art solvers/approaches. Therefore, we are left with the choice of having a mixed time-frequency domain technique, where we have our properties speci-

M.T:=Model Type;T.Domain:=Time Domain;F.Domain:=Frequency Domain

M.T	T.Domain Property	F.Domain Property	AMS Circuit
CDS	**Inevitability**		ROs
HDS	**Inevitability**	**Periodogram inequalities**	CP PLLs,Oscillators

Figure 1.1: *Thesis Contribution*

fied in the frequency domain and we carry out the verification task in the time domain. Towards this goal, we use satisfiability modulo ODE (SMODE) technique for bounded model checking (BMC) of hybrid dynamical systems (HDS), and verify frequency domain properties by checking the distance of the timed traces of the oscillator model from the traces, generated from the frequency domain properties. If this distance is less than an arbitrary small positive number, we conclude satisfaction of the frequency domain property with a degree of robustness and vice versa.

In Summary, the objectives of this thesis are:

- To present novel deductive and deductive-bounded verification methodologies for the verification of the time domain inevitability property for various AMS circuits.
- To utilize techniques from control theory in certificate based verification techniques.
- To present various scalable and tractable relaxation methods, based on mathematical programming and symbolic analysis, for both deductive and bounded approaches.
- To present novel frequency domain property specification and verification for analog oscillators, using mixed time and frequency domain techniques.

1.2 Contributions

Contributions of this thesis are depicted in Fig. 1.1 and are listed below:

- We present a scalable deductive-bounded approach to verify inevitability of phase locking in CP PLL.
  - The Circuit is modelled as a HDS.
  - We formulate the inevitability property as a conjunction of two sub-properties, defined in two disjoint subsets. Their specification is such that there is an attractive invariant set where the first property is satisfied, whereas all outside system trajectories eventually reach this set.
  - While the first property is verified using a deductive-only approach, the second property is verified by either a mixed deductive-bounded or a deductive-only approach.
  - The attractive invariance of a set is verified using a Lyapunov certificate, by patching multiple Lyapunov functions, showing all trajectories in this set eventually converge to the phase-locking state. The size of the set is found from the union of the sets, which are the sub-level sets formed by the maximized level surfaces of the individual Lyapunov functions.
  - The verification of the second sub-property is concerned with showing that trajectories reach the attractive invariant set from the set outside it. We verify this property using two methods: mixed deductive-bounded and deductive-only. In the deductive-bounded approach, reachability of the attractive invariant set is verified by bounded advection of sets, followed by the application of Escape certificates to the sets where the advection is inconclusive. In the deductive-only approach, we only use the Escape certificate showing that trajectories can not stay in a set forever and in fact escape the set in bounded time.
  - The problem of certificate construction and advection of sets, being NP-hard, is solved through the numerical SOS programming technique. We verify inevitability of a third and fourth order CP PLL.
- We also verify inevitability of oscillations in ROs with odd and even topologies. Here we use a numeric-symbolic (SOS-QE) based deductive-only approach to verify inevitability.
  - We model an RO as CDS. Furthermore, due to its physical layout, we divide the operation of the even stage RO into differential and common modes. This reduces the dimension of the system as we have to deal with two systems of smaller cardinalities.

  - The inevitability of oscillation is divided into several sub-properties, namely, attractive invariance of a set, Escape of trajectories from a set, and Eventuality of trajectories to a target set. In addition, for the common mode of the even stage RO, we specify that all common mode trajectories converge to zero.
  - The verification of these properties is formulated as FOFs over polynomial inequalities/equation and universal and existential quantifiers.
  - These FOFs are solved using a numeric-symbolic approach of SOS-QE, consisting of the certificates construction using numerical SOS programming, followed by their symbolic validation using the QE tool.

- We present a novel frequency domain properties specification and verification technique for analog oscillators.
  - We use finite Fourier series representation of the periodic limit cycle. To cater for the approximation error due to finiteness of the series, we use the error of approximation by under-over approximating the desired periodic limit cycle. To specify this desired periodic limit cycle in the frequency domain, we use the periodogram specification for each frequency component. This ensures that we have a signal of desired fundamental and harmonic frequencies. To be robust enough against the process and parameter variations, we introduce the notion of "degree of robustness" in the periodogram specification. These specifications are basically non-linear polynomial inequalities in the Fourier series coefficients.
  - We verify frequency domain properties using a mixed time-frequency domain approach. Towards this goal, we model an analog oscillator as HDS and use SMODE technique verifying, that the distance of the hybrid arcs from the time domain trajectories, generated from the frequency domain specification, is less than an arbitrary small distance.

1.3 Publications

Parts of this thesis have been published in [97],[98], [99]. The frequency domain property specification and verification of Ch. 5 has been published in [99]. The mixed deductive-bounded and the deductive-only methodologies of Ch. 3 have been published in [98] and [97] respectively. The work on the inevitability verification of oscillation in ROs, i.e. Ch.4, has been submitted to the DATE 2016 conference.

1.4 Organization

This thesis is organized as follows:

- Chapter 2 discusses the related mathematical background necessary for the rest of the thesis. This includes, mathematical modelling, Lyapunov stability theory, polynomials in real closed fields, deductive and bounded verification techniques. It also presents a literature review of previous works that has been done to verify AMS circuits.

- Chapter 3 presents a novel mixed deductive-bounded approach for the inevitability verification of phase locking in CP PLL. It discusses modelling of the CP PLL as a HDS. It then presents the formalization of the inevitability property as a conjunction of two sub-properties. It gives various algorithms, using SOS programming, to verify these properties. It ends with the experimental results for a third and fourth order CP PLL followed by a brief discussion of these results.

- Chapter 4 illustrates a deductive-only methodology for the inevitability verification of oscillation in ROs. It presents modelling of these oscillators as CDSs. It discusses the formulation of the inevitability property as a conjunction of several sub-properties in different subsets of the state space. It then presents how to pose the verification of these properties as FOFs followed by discussion of the numeric-symbolic methodology for their verification. It illustrates an algorithm verifying all sub-properties using SOS programming and QE. The chapter concludes with the experimental results and a brief discussion of these results.

- Chapter 5 illustrates the frequency domain properties specification and verification for analog oscillators. It starts with the modelling of these oscillators as HDS‘s. This is followed by a detailed discussion of the frequency domain properties specification. It then gives a mixed time and frequency domain approach to verify these properties. It presents an algorithm, based on SMODE, to verify frequency domain properties in the time domain. Lastly, it presents experimental results of our methodology for voltage controlled and tunnel diode oscillators.

- Chapter 6 concludes the results of the research that has been undertaken in this thesis. It further gives future research directions that naturally stem from this thesis.

Chapter 2

Mathematical Background and Related Work

In this chapter, we discuss mathematical background of the modelling and verification techniques we use in this thesis. Also, we give a brief review of the work done in the area of AMS circuits verification.

2.1 Continuous and Hybrid Systems

An AMS circuit consists of sub-systems having continuous, discrete and a combination of these two types of signals. To model these circuits, we use techniques from continuous and hybrid dynamical systems.

Definition 2.1 (System)**.** *A mathematical system* $\mathbb{S}$ *is a tuple* ($\mathbf{X}$, $\mathbf{O}$, $\mathbf{U}$, $\mathbf{Y}$)*, where* $\mathbf{X}$ *is the set of variables,* $\mathbf{U}$ *is the set of inputs,* $\mathbf{O}$ *is the set of relations over* $\mathbf{X}$ *and* $\mathbf{U}$*, and* $\mathbf{Y}$ *is the set of outputs.*

This definition is broad, and based on how variables $\mathbf{X}$, inputs $\mathbf{U}$, and outputs $\mathbf{Y}$ are interpreted, a system can be classified as, i) continuous, ii) discrete, iii) hybrid (continuous+discrete). Note that in this thesis we do not consider digital systems, where variables, inputs and outputs are interpreted over the Boolean set. Before we formally define continuous and hybrid systems, we define a signal which lays the foundation for characterizing different systems.

Definition 2.2 (Signal)**.** *A real signal is a mapping* $r : D \rightarrow \mathbb{R}$*. Depending on the domain* D*, a signal can be termed as continuous, discrete and hybrid. If the domain*

$D = \mathbb{Z}$, then r is termed as a discrete signal, whereas, for $D = \mathbb{R}$, it is called a continuous signal. Furthermore, if $D = \mathbb{R} \cup \mathbb{Z}$, then r is called a hybrid signal.

AMS circuits belong to the family of systems that change behaviour over time, in response to the input as well as its current state in the state space. Concretely, we say that AMS circuits have the property of memory and there is a time dependence of their outputs on the internal states of the system.

Definition 2.3 (Dynamical System)**.** *A dynamical system* **DS** *is a tuple* $(\Delta, \mathbf{X}, \mathbf{W}, \mathbf{Y}, \Phi)$ *where* $\Delta \subset \mathbb{R}_{\geq 0}(\mathbb{Z}_{\geq 0})$ *is the time space,* $\mathbf{X}$ *is the set of state variables,* $\mathbf{W}$ *is the set of inputs,* $\mathbf{Y}$ *is the set of outputs, and* $\Phi : \Delta \times \mathcal{X} \times \mathcal{W} \to \mathcal{X}$ *is the transition map. Here,* $\mathcal{X}$ *is the set of valuations over which the variables* $\mathbf{X}$ *are interpreted, and* $\mathcal{W}$ *is the set of valuations over which the inputs* $\mathbf{W}$ *are interpreted.*

This definition emphasises the fact that a dynamical system has the property of having memory as oppose to the memoryless static systems, where the output is a function of inputs only. In this thesis, we only consider continuous and hybrid dynamical systems. We will also use a definition of dynamical systems where we replace the transition map Φ by its generator. In that case the time space Δ is considered implicit.

2.1.1 Continuous Dynamical Systems

In this thesis, besides other AMS circuits properties, we verify the inevitability property of ROs. We model these oscillators as continuous dynamical system (CDS) which we formally define below.

Definition 2.4 (Continuous Dynamical system (CDS))**.** *A continuous dynamical system* **CDS** *is a tuple* $(\mathbf{X}, \mathcal{X}_{initial}, \mathbf{W}, \mathbf{U}, \mathbf{Y}, f)$ *where* $\mathbf{X}$ *is a set of state variables interpreted over* $\mathbb{R}$, $\mathcal{X} = \mathbb{R}^{|\mathbf{X}|}$ *is the set of all possible valuations of the variables,* $\mathcal{X}_{initial} \subset \mathcal{X}$ *is the set of initial conditions,* $\mathbf{W}$ *is the set of inputs interpreted over* $\mathbb{R}$ *with* $\mathcal{W} = \mathbb{R}^{|\mathbf{W}|}$ *is the set of all possible valuations of the inputs,* $\mathbf{U}$ *is the set of parameters interpreted over* $\mathbb{R}$ *with* $U = \mathbb{R}^{|\mathbf{U}|}$ *is the set of all possible parameter valuations,* $\mathbf{Y}$ *is the set of outputs interpreted over* $\mathbb{R}$ *with* $Y = \mathbb{R}^{|\mathbf{Y}|}$ *is the set of all possible input valuations and*

$$f : \mathcal{X} \times \mathcal{W} \times \mathcal{U} \to \mathcal{X} \tag{2.1}$$

is the vector field characterizing the **CDS**.

By replacing $\mathcal{X}$ with $\mathbb{R}^n$, $\mathcal{W}$ with $\mathbb{R}^k$, and $\mathcal{U}$ with $\mathbb{R}^m$, where $n = |\mathbf{X}|$, $k = |\mathbf{W}|$, $m = |\mathbf{U}|$, $\mathbb{R}^n = \mathbb{R}^{|\mathbf{X}|}$, $\mathbb{R}^k = \mathbb{R}^{|\mathbf{W}|}$, $\mathbb{R}^m = \mathbb{R}^{|\mathbf{U}|}$, we also have, $f : \mathbb{R}^n \times \mathbb{R}^k \times \mathbb{R}^m \to \mathbb{R}^n$. Note that here we use f, the generator of Φ, and the timing space is implicit.

Assumption 2.1. *In this thesis, we assume that the vector field f is a polynomial function of $x \in \mathcal{X}$, called the polynomial vector field.*

Definition 2.5 (Polynomial Continuous Dynamical system)**.** *A continuous dynamical system* **CDS** *is called a polynomial continuous dynamical system* **PCDS**, *if the vector field $f = c_0 + c_1 x + ... + c_d x^d$, i.e., f is a polynomial function of $x \in \mathcal{X}$.*

Assumption 2.2. *We assume that f is Lipschitz in $x \in \mathcal{X}$.*

This assumption of f being Lipschitz ensures that the solution of Eq. 2.1 is unique and has continuity in initial conditions [107, Ch. 8, p. 163]. Let denote by $\Phi(x_0, t)$ the set of solutions of Eq. 2.1 for all $x_0 \in \mathcal{X}_{initial}$. The Lipschitz condition on f ensures that $\Phi(x_0, t)$ always exists, is unique and has a continuous dependence on the initial conditions x_0. Therefore, the semantics of **PCDS** is given by,

$$[\![\mathbf{CDS}]\!] := \Bigg\{ \Phi(x_0,t) : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \to \mathbb{R}^n \;\Big|\; \Phi_t(x_0) = \Phi(x_0,t),\ f = \frac{d}{dt}\Phi_t(x_0)\Big|_{t=0}, \\ \forall x_0 : x_0 \in \mathcal{X}_{init},\ \forall \Phi_t(x_0) : \Phi_t(x_0) \in \mathcal{X} \Bigg\}. \tag{2.2}$$

Hereinafter, we use $x(t) = \Phi(x, t)$, and using a slight abuse of notation, we use x representing the trajectory of the **CDS** until otherwise stated.

2.1.2 Hybrid Dynamical Systems

> Briefly, a hybrid dynamical system is an indexed collection of dynamical systems along with some map for "jumping" among them (switching dynamical system and/or resetting the state). This jumping occurs whenever the state satisfies certain conditions, given by its membership in a specified subset of the state space. Hence, the entire system can be thought of as a sequential patching together of dynamical systems with initial and final states, the jumps performing a reset to a (generally different) initial state of a (generally different) dynamical system whenever a final state is reached [16, page 27].

Hybrid models have different flavours that can be found in [68], [17], [2]. In this thesis, we use the hybrid system formalism described in [42]. We believe that this is

the most generalized formalism of hybrid systems and other formalisms can easily be derived from it, as has been demonstrated in [35].

A hybrid dynamical system (HDS) is a tuple $(\mathcal{C}, \mathcal{F}, \mathcal{D}, \mathcal{G})$. Here,

$$\{\mathcal{C} = \bigcup_{i \in I_C} C_i\} \subset \mathbb{R}^n, \text{ and } \{\mathcal{D} = \bigcup_{i \in I_D} D_i\} \subset \mathbb{R}^n \tag{2.3}$$

are the flow set and jump set for $i \in \mathbb{N},\ n \in \mathbb{N}$, respectively. I_C and I_D are finite disjoint index sets and it is possible that $C_i \cap D_i \neq \emptyset$. The flow and jump maps are, respectively,

$$\mathcal{F} = \bigcup_{i \in I_C} F_i, \text{ and } \mathcal{G} = \bigcup_{i \in I_D} G_i, \tag{2.4}$$

where each,

$$F_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^{m+n}, \text{ and, } G_i = \mathbb{R}^n \to \mathbb{R}^n \tag{2.5}$$

These two mappings characterize the continuous and discrete evolution of the system, whereas C_i and D_i describe subsets of $\mathbb{R}^n$ where such evolution may occur. We represent a hybrid system $\mathcal{H}$ as

$$\mathcal{H} = \begin{cases} \dot{x} = F_i(x, u) \in \mathcal{F} & x \in \mathcal{C},\ u \in \mathcal{U} \\ x^+ = G_i(x) \in \mathcal{G} & x \in \mathcal{D} \end{cases} \tag{2.6}$$

Here $u \in \mathcal{U}(\subset \mathbb{R}^m)$ is a vector of uncertain parameters. The ODE part of Eq. 2.6 represents the continuous dynamics of the hybrid system, whereas the algebraic equation characterises the discrete jumps exhibited by the HDS.

Definition 2.6. *The set $\mathcal{X}_\mathcal{H} = \mathcal{C} \cup \mathcal{D}$ is called the hybrid state space.*

The semantics of the hybrid system can be described from its solutions. The state of the hybrid system consists of alternate flows and jumps, through $\mathcal{C}$ and $\mathcal{D}$, according to F_i and G_i respectively. This hybrid phenomena can be described by the notion of hybrid time and arc.

Definition 2.7 (Hybrid Time Domain)**.** *A set $\mathcal{T} \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a hybrid time domain*

if

$$\mathcal{T} = \bigcup_{j=0}^{j-1} ([t_j, t_{j+1}], j) \tag{2.7}$$

where $0 = t_0 \leq t_1 \leq t_2 \leq, ...,$ *with the last interval possibly of the form* $[t_j, t_{j+1}] \times \{j\},\ [t_j, t_{j+1}) \times \{j\},$ *or* $[t_j, \infty) \times \{j\}$.

We describe the semantics of the hybrid system $\mathcal{H}$ by its solutions, called hybrid arcs.

Definition 2.8 (Hybrid Arc)**.** *A mapping* $\Phi_{\mathcal{H}} : \mathcal{T} \to \mathbb{R}^n$ *is a hybrid arc if,* $\mathcal{T}$ *is a hybrid time domain, and for each* $j \in I_C$*, the function* $t \mapsto x(t, j)$ *is locally absolutely continuous on the interval* $\mathcal{I}_j = \{t : (t, j) \in \mathcal{T}\}$.

We denote by dom $\Phi_{\mathcal{H}}$, the domain of hybrid arc which is the hybrid time domain. A hybrid arc is called complete if dom $\Phi_{\mathcal{H}}$ is unbounded, i.e. if length $(\mathcal{T}) = \infty$, and it is called compact if dom $\Phi_{\mathcal{H}}$ is a compact set. A hybrid arc $\Phi_{\mathcal{H}}$ is a solution to the HDS $\mathcal{H}$, if $\Phi_{\mathcal{H}}(0, 0) \in C_j \cup D_j$, and for each $j \in I_C$ such that $\mathcal{I}_j$ has a non-empty interior, we have the semantics of $\mathcal{H}$,

$$[\![\mathcal{H}]\!] := \Bigg\{ \Phi_{\mathcal{H}}(t, j) : \mathcal{T} \to \mathbb{R}^n \Big| \dot{\Phi_{\mathcal{H}}}(t, j) = F_j(\Phi_{\mathcal{H}}(t, j)),\ \forall t \in I_j,\ \forall j \in I_C,\ \Phi_{\mathcal{H}}(t, j) \in C_j,$$
$$\forall t \in [min\ I_j, sup\ I_j),\ \Phi_{\mathcal{H}}(t, j+1) = G(x(t, j)),\ \forall \Phi_{\mathcal{H}}(t, j) \in D_j,\ \forall j \in I_D \Bigg\} \tag{2.8}$$

From the semantics $[\![\mathcal{H}]\!]$ of $\mathcal{H}$ we note that the solution $\Phi_{\mathcal{H}}$ of $\mathcal{H}$, flows according to the ODE $\dot{\Phi_{\mathcal{H}}} = F(\Phi_{\mathcal{H}})$, when it is in the set C, and it follows the jump rule $\Phi^+_{\mathcal{H}} = G(\Phi_{\mathcal{H}})$ when it belongs to the set D. Similar to CDS, hereinafter, we use $x(t, j) = \Phi_{\mathcal{H}}(t, j)$, and using a slight abuse of notation, we use x representing the hybrid arc of the HDS $\mathcal{H}$ until otherwise stated. To be able to use polynomial verification tools, using exact/approximate methods, we use polynomial HDS modelling for AMS circuits in this thesis.

Assumption 2.3. *The flow maps* $F_i(x, u)$*, and jump maps* $G_i(x)$ *are polynomials. Furthermore, sets* C_i*, and* D_i *are represented by set of polynomial inequalities/equations, also called semi-algebraic sets.*

Definition 2.9 (Polynomial Hybrid systems)**.** *A hybrid system* $\mathcal{H}$ *is said to be a polynomial HDS, if* F*,* G *are polynomials, and* C*, and* D *are semi-algebraic sets.*

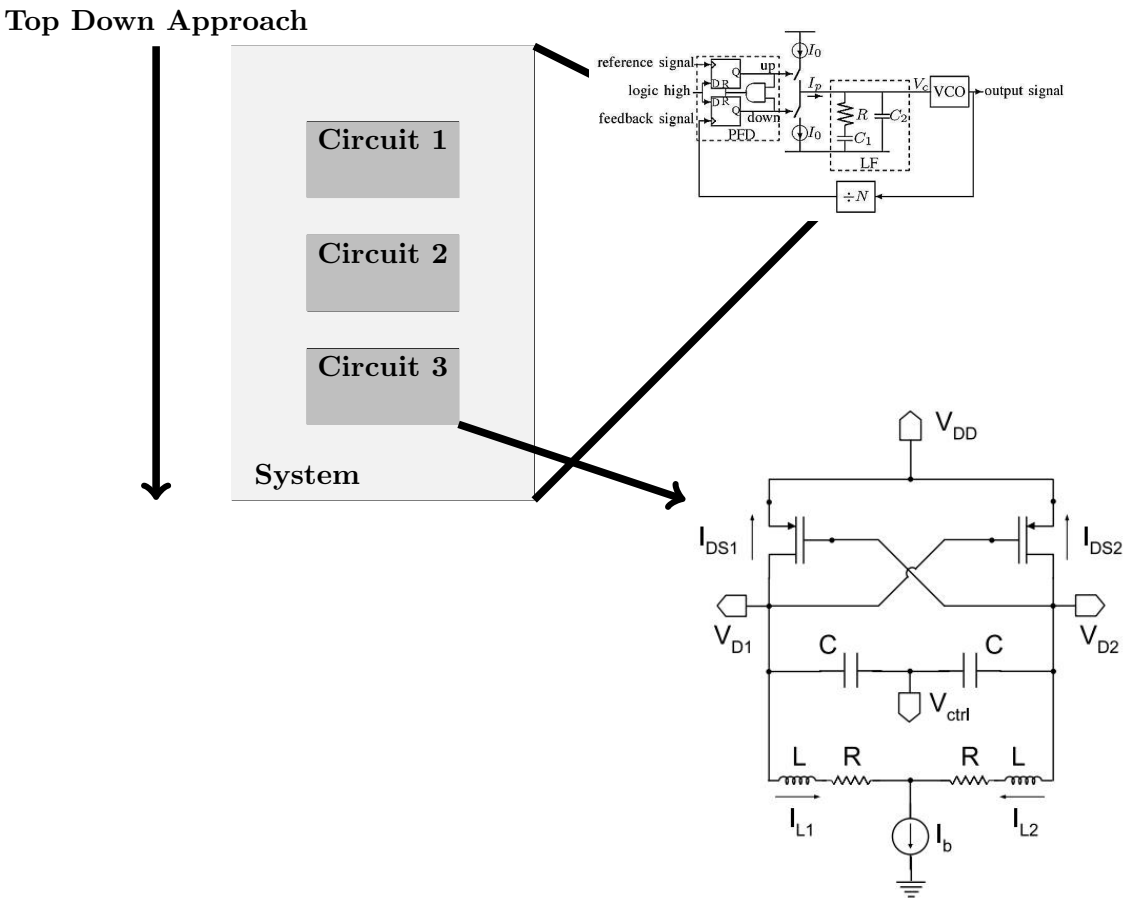


Figure 2.1: *Modelling Devices at Different Abstraction Levels*

2.2 AMS Device Models

In this thesis, we use Kirchhoff's circuit laws to find relations for currents (similarly voltages) flowing through (similarly across) different devices in AMS circuits [60]. This is followed by expressing these relations as systems of ODEs. These laws are respectively called Kirchhoff's current law (KCL) and Kirchhoff's voltage law (KVL). The KCL models the current passing through a node by taking the sum of all ingoing and outgoing currents differentiating them by +/- signs. Similarly, KVL describes the voltage across a circuit as the sum of voltage drops and rises in all loops of the circuit. These currents and voltages across different devices are non-linear functions of currents/voltages. To model these functions, we follow different strategies at different abstraction levels.

At system level we use behavioural modelling to model non-linear devices. On the other side, we use transistor level modelling at the circuit level. This is illustrated in

Fig. 2.1 ([103]). Here we consider an example of a CP PLL. At a system level, we model the devices, Voltage Controlled Oscillator (VCO) and charge pump (CP), using behavioural modelling. The non-linear behaviour of the CP is modelled as follows,

$$I_p = \begin{cases} \in [I_p^U\, I_P^U] & \text{UP=1, Down=0, } 0 \leq \phi_{VCO} < 2\pi \leq \phi_{ref} \\ \in [I_p^D I_P^D] & \text{UP=0, Down=1, } 0 \leq \phi_{ref} < 2\pi \leq \phi_{VCO} \\ \in [0^R\ 0^R] & \text{UP=0, Down=0, } 0 \leq \phi_{VCO}, \phi_{ref} < 2\pi \end{cases}$$

here I_p is the charge pump current, UP and Down are the controlled boolean signals generated by the Phase Frequency Detector (PFD) of CP PLL, ϕ_{VCO} and ϕ_{ref} are the phases of the VCO output and the reference signal [1]. Similarly, let f_{VCO}, and f_{ref}, represent the frequencies of the VCO output and the reference signal respectively. If K_p is the gain of the Low pass Filter (LF), then the behavioural model of the VCO is as follows,

$$f_{VCO} = K_p v_2/2\pi + f_O,\ \dot{\phi}_{VCO} = 2\pi f_{VCO}/N$$

where f_O is the free running frequency of the VCO. We use similar behavioural modelling to model ROs in Ch. 4.

At the circuit level, we use transistor level modelling to model the currents through the circuits. We illustrate this by the example VCO circuit shown in Fig. 2.1. The transistor is a non-linear system having complex behaviour which changes with voltage variations across its terminals. Depending on how accurate and complex analysis is performed, there are various flavours of non-linear models available for a metal oxide semiconductor (MOS) transistor. They are mainly physical models, empirical models and table models. Due to the simplicity and ease of analysis, we consider the physical model of a PMOS transistor. Physical models are obtained by mathematical equations describing the physical behaviour of the transistors. To increase the accuracy, more and more physical parameters are added to the model but at the cost of making it computationally expensive in analysis [96]. We use the Schichman-Hodges PMOS model and represent the current $I_{DS}(V_{GS}\ , V_{DS})$ through the transistor as a function of voltage

[1] A detailed description of the CP PLL modelling is given in Ch. 3

across Drain-to-Source and Gate-to-Source [72].

$$I_{DS} = \begin{cases} 0 & V_{GS} > V_{tp} \\ -K_p \frac{W}{L} \left((V_{GS} - V_{tp}) V_{DS} - \frac{1}{2} V_{DS}^2 \right) (1 - \lambda V_{DS}) & V_{GS} \leq V_{tp} \wedge V_{DG} > -V_{tp} \\ \frac{-K_p}{2} \frac{W}{L} (V_{GS} - V_{tp})^2 (1 - \lambda V_{DS}) & V_{GS} \leq V_{tp} \wedge V_{DG} \leq -V_{tp} \end{cases} \tag{2.9}$$

As can be seen, this model consists of three regions defined as Cut-off where $V_{GS} > V_{tp}$, Linear where $V_{GS} \leq V_{tp} \wedge V_{DG} > -V_{tp}$, and Saturation where $V_{GS} \leq V_{tp} \wedge V_{DG} \leq -V_{tp}$. Here W and L are the transistor width and length respectively, and λ and K_p are parameters representing transconductance and conductance respectively. This is a level 1 model, and higher order models can be derived by introducing more parameters considering other physical effects.

2.3 Lyapunov Stability of Continuous and Hybrid Dynamical Systems

Central to our deductive approach, this section discusses Lyapunov stability of polynomial continuous and hybrid dynamical systems. Throughout the thesis, we use Lyapunov-like certificates for the verification of properties like inevitability, eventuality, Escape from a set, and asymptotic stability. Here we consider only the Lyapunov stability, and other similar concepts will be discussed in later chapters.

2.3.1 Lyapunov Stability of Continuous Dynamical Systems

Though Lyapunov stability is a general concept and can be attributed to any set, in this thesis, we restrict its application to the stability of the equilibrium state which is defined below.

Definition 2.10 (Equilibrium State)**.** *A point $x_e \in \mathcal{X}$ is called an equilibrium state, if $f(x_e) = 0$.*

It is a convention that the equilibrium state $x_e = 0$. However, practical systems converge to states different from the zero equilibrium state. In that case, the equilibrium x_e can be shifted to zero by the introduction of a new variable $\bar{x} = x - x_e$. This results in

a new set of differential equations in $\bar{x}$ with equilibrium at $\bar{x} = 0$ which is similar to $x = x_e$. Therefore, without loss of generality, we assume that $x_e = 0$.

Definition 2.11 (Invariant Set)**.** *A set $\mathcal{X}_\mathbf{I} \subset \mathcal{X}$ is called invariant, iff, $\forall x : x \in \mathcal{X}_\mathbf{I}$, $x(t) \in \mathcal{X}_\mathbf{I}$ $\forall t : t \in \mathbb{R}_{\geq 0}$.*

Def. 2.11 implies that the equilibrium state x_e is an invariant set. Unlike the safety property, where the existence of an invariant set having an empty intersection with the unsafe state is enough, in this thesis we are concerned with attractive invariant sets.

Definition 2.12 (Attractive Invariant Set)**.** *A set $\mathcal{X}_\mathbf{AI} \subset \mathcal{X}$, such that $x_e \in \mathcal{X}_\mathbf{AI}$, is called an attractive invariant, if it is an invariant set (Def. 2.11), and $lim_{t\to\infty}\ x(t) = x_e$.*

The Def. 2.12 states that an attractive invariant set apart from being invariant, is attractive in the sense that every trajectory in the set eventually converges to the equilibrium state. For an equilibrium state $x_e = 0$, we now define the asymptotic stability.

Definition 2.13 (Asymptotic Stability [61])**.** *A polynomial continuous dynamical system* **PCDS** *is called stable if for every $\epsilon > 0$, there exists a $\delta > 0$ such that $\forall x(.) : x(.) \in \mathcal{X}$,*

$$\|x(0)\| < \delta \implies \|x(t)\| < \epsilon,\ \forall t : t \in \mathbb{R}_{\geq 0}.$$

It is called attractive if,

$$\|x(0)\| < \delta \implies lim_{t\to\infty}\ x(t) = 0.$$

A system that is both stable and attractive is called an asymptotically stable system.

This definition of asymptotic stability states that, if a system starts in the δ neighbourhood of the equilibrium, then it must stay in the ϵ neighbourhood forever. Now the question is how to determine this asymptotic stability of a CDS. Towards this goal, Lyapunov in 1892 introduced an abstract energy like function, called "Lyapunov function", that can be used to determine the stability of a CDS. We state the Lyapunov stability criterion in the following theorem called the Lyapunov stability theorem.

Theorem 2.1 (Lyapunov Stability [61])**.** *For the polynomial continuous dynamical system* **PCDS** *with an equilibrium $\{x_e = 0\} \in \mathcal{X}$, if there is a differentiable certificate*

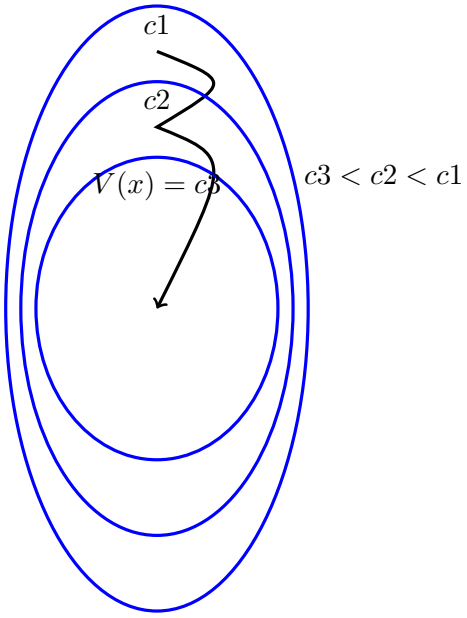


Figure 2.2: *Level Surfaces of Lyapunov Certificate*

$V : \mathcal{X} \to \mathbb{R}$ *such that,*

$$V(0) = 0 \text{ and } V(x) > 0 \ \forall x : x \in \mathcal{X} \setminus \{0\} \tag{2.10}$$

$$\frac{\partial V}{\partial x}(x) f(x) < 0 \ \forall x : x \in \mathcal{X} \setminus \{0\} \tag{2.11}$$

then the equilibrium x_e is asymptotically stable. This function $V(x)$ is called the Lyapunov certificate.

An interesting property of the Lyapunov certificate $V(x)$ is that the sub-level sets, $V(x) \leq c$, for all $c > 0$, are attractive invariant sets. This is illustrated intuitively in Fig. 2.2. Once a trajectory enters a level surface, it can not get out of the corresponding level set and eventually converges to the equilibrium state. It is this attractive invariance property of the sub-level sets, described by the Lyapunov surfaces, that we use in the verification of inevitability property of CP PLL in Ch. 3. Similarly, certificates in Ch. 3 and Ch. 4, are inspired in principle by the Lyapunov certificate.

2.3.2 Lyapunov Stability of Hybrid Dynamical Systems

Extension of the asymptotic stability concept to HDS needs to take in to account both continuous solutions and discrete jumps. For continuous flows due to F_i, through flow

sets C_i, for all i, asymptotic properties of HDS show what happens when time t approaches infinity. Similarly, for jumps G_i through D_i, asymptotic properties describe what happens when the number of discrete jumps j approaches infinity. Since asymptotic stability requires completeness of solutions of the HDS, we do not require that both "t" and "j" approach infinity. Instead, we require that the domain "t+j" should be unbounded [42]. Similar to the CDS, we assume here that the origin is the equilibrium state.

Definition 2.14 (Equilibrium State)**.** *A point $x(t,j) \in \mathcal{C} \cup \mathcal{D}$ is called an equilibrium, if $\exists t,\ \exists j,\ \exists u,\ F_j(x(t,j),u) = 0$.*

Definition 2.15 (Asymptotic stability of HDSs [42])**.** *The hybrid system $\mathcal{H}$ defined in Sec. 2.1.2, is called stable if for each $\epsilon > 0$ there exists $\delta > 0$ such that for each solution $\Phi_{\mathcal{H}}$ of $\mathcal{H}$ with,*

$$\|\Phi_{\mathcal{H}}(0,0)\| < \delta \implies \|\Phi_{\mathcal{H}}(t,j)\| < \epsilon,\ \forall (t,j) : (t,j) \in dom\ \Phi_{\mathcal{H}}.$$

It is called attractive, if there exists $\delta > 0$ such that for every complete solution $\Phi_{\mathcal{H}}$ such that

$$\|\Phi_{\mathcal{H}}(0,0)\| < \delta \implies lim_{(t+j)\to\infty}\ \Phi_{\mathcal{H}}(t,j) = 0\ \forall (t,j) : (t,j) \in dom\ \Phi_{\mathcal{H}}.$$

A hybrid system $\mathcal{H}$ is asymptotically stable if it is both stable and attractive.

Similar to CDS, we use a Lyapunov certificate to parametrize the stability and attractivity of a HDS. Extending the Lyapunov stability theorem to HDS, we need to consider discrete jumps apart from the continuous flows. This is the reason that classical Lyapunov theorems are not directly applicable to HDS owing to the continuity requirement of vector fields. However, two famous approaches have been used to parametrize the stability of a HDS. These are, common Lyapunov certificates and multiple Lyapunov certificates. In [65], the author introduced a common Lyapunov certificate for all discrete modes of the HDS considering identity jump maps.

Theorem 2.2. *For the hybrid system $\mathcal{H}$ having an equilibrium point $x_e = 0$, let $G_i(x) = x,\ \forall i : i \in I_D$. Let there exist a continuously differentiable Lyapunov certificate $V : \mathcal{C} \to \mathbb{R}$, such that,*

$$V(0) = 0 \text{ and } V(x) > 0,\ \forall x : x \in \mathcal{C} \setminus \{0\} \tag{2.12}$$

$$\frac{\partial V}{\partial x}(x)F_i(x,u) < 0,\ \forall i : i \in I_C\ \forall x : x \in \mathcal{C} \setminus \{0\},\ \forall F_i : F_i \in \mathcal{F},\ \forall u : u \in \mathcal{U}. \quad (2.13)$$

Then the equilibrium x_e is asymptotically stable.

This approach of using the common Lyapunov certificate is very restrictive and discards all knowledge about discrete jumps in the HDS. There are many stable HDSs that do not allow such common Lyapunov certificates [55]. Alternatively, an approach using multiple Lyapunov certificates, one for each continuous flow F_i, has been proposed by [16]. The author in [16] introduced the idea of having multiple Lyapunov certificates for each $i \in I_C$ such that every certificate, in addition to satisfying the conditions stated in Th. 2.2, must have a decreasing trend at each entry point of the interval I_i, during which $F_i(x,u)$ is the active continuous dynamics. In [74], the author introduced even stronger conditions for jump maps of a hybrid system that require the value of a Lyapunov certificate after the jump to be less than the value of another Lyapunov certificate before the jump. In this thesis we use this technique and restate it in the following theorem.

Theorem 2.3. *Let, $\mathcal{I}_0 \subseteq I_C$ be the set of indices that contain the equilibrium. For a hybrid system $\mathcal{H}$ having an equilibrium point $x_e = 0$, if there exist Lyapunov certificates V_i such that,*

$$V_i(0) = 0,\ \forall i : i \in \mathcal{I}_0, \quad (2.14)$$

$$V_i(x) > 0,\ \forall i : i \in I_C,\ \forall x : x \in \mathcal{C} \setminus x_e, \quad (2.15)$$

$$\frac{\partial V_i}{\partial x}(x)F_i(x,u) < 0,\ \forall i : i \in I_C,\ \forall x : x \in \mathcal{C} \setminus x_e,\ \forall F_i : F_i \in \mathcal{F},\ \forall u : u \in \mathcal{U} \quad (2.16)$$

$$V_j(G_i(x)) - V_{j'}(x) \leq 0,\ \forall j\ \forall j' : j, j' \in I_C,\ j \neq j',\ \forall i : i \in I_D, \forall x : x \in \mathcal{D} \setminus x_e, \forall G_i : G_i \in \mathcal{G}, \quad (2.17)$$

then x_e is asymptotically stable. Furthermore, the set $\mathcal{X}_{\mathbf{AI}} = \{\bigcup_i (V_i \leq c_{max})\} \subset \mathcal{X}_{\mathcal{H}}$ is an "attractive invariant" set.

This theorem, in addition to the conditions of positivity for each Lyapunov certificate and negativity for their respective derivatives, introduces an additional constraint for the jump maps. This constraint makes sure that after each jump, the corresponding

Lyapunov certificate must be less than in value than the Lyapunov certificate before taking the discrete jump.

2.4 Polynomials in Real Closed Fields

Several problems in the field of physics, mathematics and engineering can be formally expressed as a finite number of polynomial equalities and inequalities. In this thesis, we use deductive and bounded verification approaches which involve checking of polynomial positivity/negativity over a real closed field.

Definition 2.16 (Monomial)**.** *A monomial m_β is a mapping $m_\beta : \mathbb{R}^n \to \mathbb{R},\ \beta \in \mathbb{Z}_{>0}$, such that $m_\beta(x) = x^\beta := x_1^{\beta_1} x_2^{\beta_2} ... x_n^{\beta_n}$. The degree d of a monomial is defined as, $d(m_\beta) := \sum_{i=1}^{n} \beta_i$*

Definition 2.17 (Polynomial)**.** *A linear combination of monomials is called a polynomial, i.e. a polynomial p is,*

$$p := \sum_{\beta} c_\beta m_\beta,\ p(x) := \sum_{\beta} c_\beta m_\beta(x) = \sum_{\beta} c_\beta x^\beta,\ c_\beta \in \mathbb{R}\ \forall \beta : \beta \in \mathbb{Z}_{>0} \tag{2.18}$$

The degree d of the polynomial p is defined as, $d(p) := max_\beta\ d(m_\beta)$. A polynomial is called homogeneous if all monomials have the same degree d. Also, for homogeneous polynomials, $p(\lambda x) = \lambda^d p(x),\ \lambda \in \mathbb{R}$.

We denote the set of polynomials in n variables with real coefficients by $\mathcal{R}_n$. A subset of this set is the set of positive-semidefinite (PSD) polynomials in n variables denoted by $\mathcal{P}_n$ and defined as, $\mathcal{P}_n := \{q \in \mathcal{R}_n | q(x) \geq 0, \forall x \in \mathbb{R}^n\}$.

2.4.1 Sum of Squares Polynomial

A set of polynomials $\mathcal{S}_n \in \mathcal{R}_n$, is called sum of squares (SOS) polynomials, if

$$\mathcal{S}_n := \{s \in \mathcal{R}_n\ | s = \sum_{i}^{n} p_i^2, p_i \in \mathcal{R}_n,\ n \in \mathbb{Z}_{>0}\} \tag{2.19}$$

An interesting property of a SOS polynomial $s \in \mathcal{S}_n$, is that $s(x) \geq 0, \forall x \in \mathbb{R}^n$. This shows that $\mathcal{S}_n \subseteq \mathcal{P}_n$ [73]. As shown in the Hilbert seventeenth problem, there are subsets of polynomials for which the set of SOS polynomials $\mathcal{S}_n$ and PSD polynomials $\mathcal{P}_n$ are equal. These are homogeneous polynomials, in one variable ($n = 1$), quadratic ($d = 2$), and quartic in two variables ($n = 2, d = 4$). Generally, $\mathcal{S}_n \subset \mathcal{P}_n$. The Motzkin

polynomial given below is one such example which is PSD but is not a SOS polynomial [73].

$$M(x, y, z) = x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$$

2.4.2 SOS Programming

Our mixed deductive-bounded verification approach involves checking the positivity/negativity of polynomials in semi-algebraic sets. Essentially, given a multivariate polynomial $p(x)$, we are interested in verifying that,

$$p(x) \geq 0, \forall x \in \mathbb{R}^n \tag{2.20}$$

As stated in [73], there are decision procedures, such as the Tarski-Sedenberg [93], which provides exact solution of this positivity verification problem. However, the complexity of these decision procedures is NP-hard (when the degree is at least 4), and with large numbers of variables, the behaviour of these methods is unacceptable. Therefore, to avoid the computational complexity barrier of these exact methods, and provide a realistic scalable solution, a computable relaxed and sound approach has been proposed in [78], [73].

A sufficient condition for a multivariate polynomial $p(x)$ to be non-negative everywhere is that it can be decomposed as a sum of squares of polynomials. A polynomial $p(x)$ is a SOS, if there exist polynomials $p_1(x), ..., p_m(x)$ such that,

$$p(x) = \sum_{i=1}^{m} p_i^2(x), \;\; p_i(x) \in \mathcal{R}_n \tag{2.21}$$

Note that since SOS polynomials are always of even degree, we denote the set of SOS polynomials by $\mathcal{S}_{n,2d}, d \in \mathbb{Z}_{>0}$. In [19], the author presented a parametrization of the SOS polynomials called "Gram-matrix" shown below,

$$p(x) = Z^T(x)QZ(x), \;\; Z(x) = [1, x_1, x_2, ...x_n, x_1x_2, ..., x_n^d] \tag{2.22}$$

Here Q is a constant matrix and the vector Z is of length $\binom{n+d}{d}$. If Q is positive semidefinite, then $p(x)$ is non-negative. It can easily be shown that the set of matrices satisfying Eq. 2.22 is an affine subspace.

We can show that if $Q \succeq 0$, by the eigen value decomposition, then Eq. 2.22 holds

[73],

$$Q = T^T DT,\ D = diag\{d_i\},\ d_i \geq 0 \implies p(x) = \sum_i d_i (TZ)_i^2 \tag{2.23}$$

It can easily be shown that the number of squares in the representation is equal to the rank of the matrix Q. We borrow a theorem from [73] which shows that the decision procedure for a polynomial to be a SOS can be formulated as a semidefinite program.

Theorem 2.4. *The existence of a SOS decomposition of a polynomial in n variables of degree $2d$ can be decided by solving a semidefinite programming feasibility problem. If the polynomial is dense (no sparsity), the dimensions of the matrix inequality are equal to $\binom{n+d}{d} \times \binom{n+d}{d}$.*

From the above theorem, we notice that the size of the SDP problem is polynomial in both d or n if one or the other is fixed. It is exponential if both are variable. We avoid discussing semidefinite programming here and the reader can see Appendix A.1 for a detailed discussion. We give an example from [73], and show how this process of SOS decomposition works for a homogeneous polynomial.

Example 2.1. *Consider the quartic form in two variables,*

$$F(x,y) = 2x^4 + 2x^3y - x^2y^2 + 5y^4$$

$$= \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix}^T \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{12} & q_{22} & q_{23} \\ q_{13} & q_{23} & q_{33} \end{bmatrix} \begin{bmatrix} x^2 \\ y^2 \\ xy \end{bmatrix}$$

$$= q_{11}x^4 + q_{22}y^4 + (q_{33} + 2q_{12})x^2y^2 + 2q_{13}x^3y + 2q_{23}xy^3$$

Comparing coefficients of the monomials, the following linear equalities must hold,

$$q_{11} = 2, \quad q_{22} = 5, \quad q_{33} + 2q_{12} = -1, \quad 2q_{13} = 2, \quad 2q_{23} = 0. \tag{2.24}$$

Using Semidefinite programming, a positive semidefinite Q can be found that satisfies

the above linear equalities. One solution is,

$$Q = \begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix} = L^T L, \quad L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$$

therefore, the SOS decomposition of the polynomial $F(x, y)$ *is,*

$$F(x, y) = \frac{1}{2}(2x^2 - 3y^2 + xy)^2 + \frac{1}{2}(y^2 + 3xy)^2.$$

A SOS feasibility program, as defined in [79], is of the form,

Find

$p_i(x) \in \mathcal{R}_n$, for $i = 1, 2, ..., \hat{N}$

$p_i(x) \in \mathcal{S}_n$, for $i = 1, 2, ..., \hat{N}$

such that

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) = 0, \text{ for } j = 1, 2, ..., \hat{J},$$

$$a_{0,j}(x) + \sum_{i=1}^{N} p_i(x) a_{i,j}(x) \in \mathcal{S}_n \text{ for } j = (\hat{J} + 1), 2, ..., J.$$

Here $a_{i,j} \in \mathcal{R}_n$ are known polynomials. An optimization SOS program is similar with the addition of an objective function.

2.4.3 Positivstellensatz

In this thesis, we use a mathematical technique, called the S-procedure, to incorporate the domain constraints in SOS programming. Before illustrating this procedure, we describe a theorem from real algebraic geometry which is the foundation of this technique [73].

Definition 2.18. *For a set of polynomials* $\{g_1, ..., g_m\} \in \mathcal{R}_n$*, the Multiplicative Monoid is the set of all finite products of* g_m*'s including 1. We denote this by* $\mathcal{M}(g_1, ..g_m)$*.*

Definition 2.19. *For a set of polynomials* $\{h_1, ..., h_m\} \in \mathcal{R}_n$*, the cone generated by*

h_i's is,

$$\mathbf{C}(h_1,..h_m) = \{a_0 + \sum_i^N a_i b_i \mid N \in \mathbb{Z}_{>0}\ a_i \in \mathcal{S}_n,\ b_i \in \ \mathcal{M}(h_1,..h_m)\}. \tag{2.25}$$

Definition 2.20. *For a set of polynomials* $\{f_1,...,f_m\} \ \in \mathcal{R}_n$, *the Ideal generated by* f_k's *is,*

$$\mathbf{I}(f_1,..f_u) = \{\sum f_k p_k \mid p_k \in \mathcal{R}_n\}. \tag{2.26}$$

Theorem 2.5 (Positivstellensatz)**.** *For a set of sets of polynomials,* $\{g_1,...g_s\},\ \{h_1,..h_t\}$, $\{f_1,...f_u\} \in \mathcal{R}_n$, *the following statements are equivalent:*

- *The following set is empty,*

$$\left\{ \begin{array}{ll} g_j(x) \geq 0, & j = 1,...,s \\ h_k(x) \neq 0, & k = 1,...,t \\ f_l(x) = 0, & l = 1,...,u \end{array} \right\}$$

- *There exists* $g \in \mathbf{C},\ h \in \mathcal{M},\ f \in \mathbf{I}$, *such that* $g + h^2 + f = 0$.

Proof. See [73]. □

2.4.4 S-Procedure

In this thesis, we encounter problems that involve checking positivity/negativity of polynomials in a compact set. To incorporate these domain constraints, we use a mathematical technique, from [15] and generalized in [53], called the S-procedure.

Lemma 2.1. *Given the set of polynomials,* $\{p_i\}_{i=0}^m \in \mathcal{R}_n$, *if there exists a set of polynomials,* $\{a_i\}_i^m \in \mathcal{S}_n$ *such that,*

$$p_0 - \sum_{i=1}^m a_i p_i = q,\ q \in \mathcal{S}_n \tag{2.27}$$

then,

$$\bigcap_{i=1}^m \{x \in \mathbb{R}^n \mid p_i(x) \geq 0\} \subset \{x \in \mathbb{R}^n \mid p_0(x) \geq 0\} \tag{2.28}$$

Proof. We prove this lemma by the set emptiness of a set defined by polynomial inequalities using Positivstellensatz [53]. The condition in the Lemma is true if the set,

$$S = \{x \in \mathbb{R}^n \mid p_1(x) \geq 0, ..., p_m(x) \geq 0, -p_0(x) \geq 0, p_0(0) \neq 0\}$$

is empty. We define the cone l generated by $(-p_0, p_i)$ for $i \in (1, ...m)$, as $l = -qp_0 - \sum_{i=1}^{n} a_i p_0 p_i,\ q \in \mathcal{S}_n$, and the Multiplicative Monoid w of $p_0 \neq 0$ as $w = p_0(x)$. Set S is empty if $l + w^2 = 0$. Therefore,

$$l + w^2 = -qp_0 - \sum_{i=1}^{n} a_i p_0 p_i + p_0^2$$
$$= -(p_0 - \sum_{i=1}^{m} a_i p_i)p_0 - \sum_{i=1}^{n} a_i p_0 p_i + p_0^2 = 0$$

This proves emptiness of the set S. □

For a polynomial $q : \mathbb{R}^n \to \mathbb{R}$, differentiable scalar function, we define the 0-sub-level-set of q as $\mathcal{Z}(q) = \{x \in \mathbb{R}^n \mid q(x) \leq 0\}$. We present an important lemma to be used for polynomial level set operations such as intersection, union, and set inclusion [102].

Lemma 2.2. *For polynomials $p1, p2 \in \mathcal{P}_n$, if there exist SOS polynomials $s0,\ s1 \in \mathcal{S}_n$ such that*

$$s0 - s1p1 + p2 = 0\ \forall x \in \mathbb{R}^n \tag{2.29}$$

Then $\mathcal{Z}(p1) \subset \mathcal{Z}(p2)$

Proof. Follows directly from Lemma. 2.1. Also, see for example [102] and the references therein. □

2.5 Formal Verification of Continuous and Hybrid Systems

After their successful application for the verification of discrete systems in software and hardware design, researchers have been using formal methods for continuous and hybrid systems verification for the last decade. Since a CDS can be treated as a HDS with a single mode, here we discuss formal verification techniques for HDS. Bounded and deductive verification are the two famous techniques that have been used for HDS.

2.5.1 Bounded Verification

In this thesis, we use Bounded model checking and Bounded Advection of sets for HDS verification. Here, we give a brief discussion on these two bounded verification approaches.

2.5.1.1 Bounded Model Checking

In a model checking approach, the system (continuous, hybrid) is represented by a transition system. A transition system $\mathbf{T}$ has a finite number of states with transition rules from one state to another. The model checking algorithm exhaustively checks all possible finite states of the transition system $\mathbf{T}$, for a property which is specified in LTL (Linear Temporal Logic) or CTL (Computational Tree Logic). We avoid discussing these specification languages, and readers are referred to [76] and [22] for a detailed discussion on LTL and CTL respectively. A transition system $\mathbf{T}$ is said to satisfy a property P, if all possible runs Π of $\mathbf{T}$ are models of the property P [23].

$$\mathbf{T}_\pi \models P,\ \forall \pi : \pi \in \Pi. \tag{2.30}$$

If there is a state in the transition system $\mathbf{T}$ that does not satisfy a property, the model checking algorithm reports it as a counter example of the property P.

While model checking has been quite successfully used for temporal properties verification of finite state systems and time automata [10], it suffers from the problem of state space explosion for practical infinite state systems. An alternate solution of BMC has been proposed in [13] for infinite systems. In this approach, instead of searching for violation of the property in the entire state space of a transition system, a more modest approach is adopted, and the refutation of the property is checked for a bounded length of runs of the transition system. Specifically, the refutation of the property P is reduced to checking the satisfiability of the formula,

$$\mathbf{I}(x_0) \wedge \mathbf{T}(x_0, x_1) \wedge .....\mathbf{T}(x_{k-1}, x_k) \wedge \neg P(x_k). \tag{2.31}$$

Here $\mathbf{I}(x_0)$ is the predicate over the initial condition x_0, $\mathbf{T}(x_{k-1}, x_k)$ is the transition relation between the pre-state x_{k-1} and post-state x_k. The bound k is successively increased until we either get a counterexample or a pre-specified maximum bound of k is reached.

BMC of hybrid systems involves predicate encoding of the sequential behaviour of transition systems and the target formulas. This is followed by a decision procedure,

e.g. SAT, to find a satisfying instantiation of the target formula. Formally, for a HDS $\mathcal{H}$, and a formula ϕ(LTL or CTL), we ascertain that ϕ holds in $\mathcal{H}$ by looking for a counterexample on a k length trajectory. This can be reduced to the satisfiability of the following formula,

$$[\![\mathcal{H}, \neg\phi]\!]_k = \mathbf{I}(x^{(0)}) \wedge \bigwedge_{i=0}^{k} Inv(x^{(i)}) \wedge \bigwedge_{i=0}^{k} \mathbf{T}(x^{(i)}, x^{(i+1)}) \wedge \bigvee_{i=0}^{k} \neg\phi(x^{(i)}), \tag{2.32}$$

Where $\mathbf{I}(x^{(0)})$ is a predicate for initial conditions, $Inv(x^{(i)})$ is an invariant predicate at step i, and $\mathbf{T}(x^{(i)}, x^{(i+1)})$ is a transitional relation from step i to step $i+1$. If $[\![\mathcal{H}, \neg\phi]\!]_k$ is satisfiable, we find a counterexample within k steps. In case of unsatisfiability of $\neg\phi$, we check its satisfiability for an increasing number of steps until a given limit is reached. Andreas et al. in [33], presented an SMODE technique for the polynomial hybrid automata verification. Essentially, it is a technique based on the BMC of the polynomial hybrid automata, encoded as a large number of constraints; involving boolean, linear and non-linear semi-algebraic, and non-linear ODE constraints. The transition system $\mathbf{T}$ is in the form of non-linear ODEs, which is then conservatively solved to find the set of solutions in a particular unrolling k of the above FOF. We use this approach for the verification of frequency domain properties in Ch. 5.

2.5.1.2 Bounded Advection of Sets

A technique similar but dual to BMC, is the reach set computation as described in [69]. In this approach, starting from an initial set of states, the forward set of states is computed for a HDS. Intersection of reachable sets is tested with the target set and safety of the system is declared for an empty intersection set or otherwise. It is the dual of BMC in the sense that while BMC searches for the refutation of a formula, reach set computation checks that a property (safety) is verified in bounded steps. In this thesis, we use a technique, first introduced in [102] for CDSs, called advection of sets. We illustrate this technique for CDS and later in Ch. 3, we extend it to the verification of HDS.

For the continuous flow map $\psi : \mathbb{R}^n \times \mathbb{R}_{\geq 0} \to \mathbb{R}^n$ of the polynomial continuous dynamical system **PCDS**, a time t advection operator A_t is a map

$$A_t : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R}), \ t \in \mathbb{R}_{\geq 0} \tag{2.33}$$

such that,

$$U = A_t V \text{ for } \{U, V\} \in C(\mathbb{R}^n, \mathbb{R})$$

and

$$U(x) = V(\psi_{-t}(x)) \text{ for all } t \in \mathbb{R}_{\geq 0}.$$

Here $C : \mathbb{R}^n \to \mathbb{R}$ is a set of differential maps from euclidean space to the set of real numbers. The advection operator has an important property of linearity. For polynomial functions $U1, U2 \in C(\mathbb{R}^n, \mathbb{R})$, if

$$U2 = A_t U1$$

then

$$\mathcal{Z}(U2) = \psi_t(\mathcal{Z}(U1)) \tag{2.34}$$

This advection of level sets is shown in Fig. 2.3. For practical purposes, an approximation to the flow map ψ_h with time step h is used. For example, a first order Taylor approximation yields the following advection of a set,

$$U = B_h V, \text{ if } U(x) = V(x) - h\frac{\partial}{\partial x}V(x)f(x) \tag{2.35}$$

Here $B_h : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ is the first order Taylor approximation to A_h. Higher order advection maps result in more accurate results but at the cost of an increased degree of polynomials and computation time. Furthermore, the product $B_h V$ results in polynomials with a degree higher than V. Therefore, a conservative approximation for advected level sets is used. Introducing an approximation parameter $\mu \in \mathbb{R}_{\geq 0}$, we conservatively approximate the zero sub-level set of a polynomial q by backward advecting the polynomial p such that,

$$\mathcal{Z}(q) \subset \mathcal{Z}(B_{-h}p) \subset \mathcal{Z}(q - \mu) \tag{2.36}$$

To incorporate the truncation error due to Taylor approximation, let us introduce η such that $\|\nabla^2 p \frac{h^2}{2}\| \leq \eta$. This requires that

$$\mathcal{Z}(B_{-h}p + \eta) \subset \mathcal{Z}(A_{-h}p) \subset \mathcal{Z}(B_{-h}p - \eta) \tag{2.37}$$

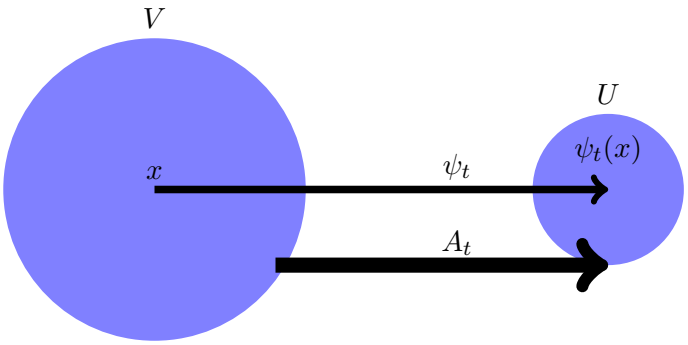


Figure 2.3: *Advection of Sets in Continuous Systems*

This implies that,

$$\mathcal{Z}(q) \subset \mathcal{Z}(B_{-h}p + \eta) \subset \mathcal{Z}(A_{-h}p) \subset \mathcal{Z}(B_{-h}p - \eta) \subset \mathcal{Z}(q - \mu) \tag{2.38}$$

We incorporate these set inclusions as SOS programs which will be discussed in Ch. 3. The reader is further advised to see [102] for an in depth discussion on the technical background of the advection of level sets.

2.5.2 Deductive Verification

In deductive verification, using a theorem prover, the correctness of a design is verified based on some pre-defined inference rules. Using inductive invariants, safety verification has been performed for discrete systems [8]. This involves identifying an invariant property ϕ which holds for the initial set, a safe set, and all transitions of the discrete system. Extending the concept of invariants to CDSs and HDSs, in [77], the author introduced barrier certificates for CDSs and HDSs. For a continuous dynamical system **CDS** with f as given in Eq. 2.1, initial set $\mathcal{X}_0 \subset \mathcal{X}$, and unsafe set $\mathcal{X}_u$, if there is a barrier certificate $\Upsilon : \mathcal{X} \to \mathbb{R}$, satisfying

$$\Upsilon(x) \leq 0, \ \forall x : x \in \mathcal{X}_0 \tag{2.39}$$

$$\Upsilon(x) > 0, \ \forall x : x \in \mathcal{X}_u \tag{2.40}$$

$$\frac{\partial \Upsilon}{\partial x}(x) f(x) \leq 0, \ \forall x : x \in \mathcal{X} \text{ such that } \Upsilon(x) = 0 \tag{2.41}$$

then **CDS** is safe and there is no system trajectory starting in $\mathcal{X}_0$ that will end up in $\mathcal{X}_u$. The same concept has been extended to HDS. The concept of a Barrier certificate has actually been borrowed from the Lyapunov certificate based stability verification of CDS and HDS, discussed in Sec. 2.3. Deductive verification of CDS and HDS has also been demonstrated in [91], and [90].

In certificate based deductive verification, the critical issue is how to find the certificate from an infinite set of feasible certificates of a given structure. Formally, the existence of these certificates can be formulated as a FOF having polynomial equations, inequalities, quantifiers $\{\forall, \exists\}$ and boolean operators $\{\wedge, \vee, \neg, \rightarrow, \text{etc}\}$. There are algorithms that can in principle generate quantifier free formulas from a universal-existential quantified FOF over the real numbers [89]. Let us denote these formulas by $\mathbf{F}_i(V_q, U_f),\ i = 1, ...k,$, where $V_q = (v_1, v_2, .., v_n) \in \mathbb{R}^n$ is a set of quantified variables, and $U = (u_1, u_2, .., u_m) \in \mathbb{R}^m$ is a set of unquantified parameters [51]. A quantified FOF is given as,

$$(Q_1 V_1 ....Q_r V_r)\psi(F_1 ....F_k) \tag{2.42}$$

where $\psi(F_1 ....F_k)$ is a quantifier free boolean formula such that $F_i = \mathbf{F}_i(V_q, U_f)\ \Delta\ 0$, for $i = 1, ...k,\ \Delta \in \{=, \geq, \leq, \neq\}$, V_i is a block of q_i quantified variables for $(i = 1, ..., r)$ from V_q such that $(q_1 + ... + q_r = n)$ and $V_l \cap V_s = \emptyset$ for all l and s $(l \neq s, l = 1, ...r, s = 1...r)$, and $Q_i \in \{\forall, \exists\}$. For example, the conditions of the Lyapunov certificate in Th. 2.1 can be formulated as a FOF as follows,

$$\begin{aligned}
\psi_0 &:= \exists p^{\mathcal{P}} : \psi_1 \\
\psi_1 &:= \forall x^{\mathcal{X}} : \psi_2 \\
\psi_2 &:= \Big((x = 0 \implies V(p, x) = 0) \wedge \\
&\quad (x \neq 0 \implies V(p, x) > 0) \wedge \\
&\quad (x \neq 0 \implies \frac{\partial V}{\partial x}(p, x).f(x) < 0 \Big)
\end{aligned}$$

A QE algorithm is used to replace such a quantified formula with another equivalent but unquantified formula. A decision procedure is used to come up with a true/false answer if there is no unquantified parameter in the formula. For a formula having both quantified variables and unquantified parameters, the QE reduces the formula to a quantifier free formula with only parameters. For example [89], in the following

quantified formula,

$$\exists x(ax + b = 0) \Longleftrightarrow a \neq 0 \vee (a = 0 \wedge b = 0),$$

variable x has been eliminated by the QE with the right hand side formula now a decision procedure to be either true or false depending on the value bounds of a and b.

Several techniques have been presented for QE, such as Cylindrical Algebraic Decomposition (CAD) [24] and Virtual Substitution [106]. However, while the virtual substitution method is better for simple problems of an academic nature, CAD has the doubly exponential worse-case complexity. Therefore, application of these QE techniques to real world problems has shown very little progress. To reduce the computational workload of these QE techniques, a combination of SOS-QE and SOS-HOL theorem proving has been used in [51],[85] and [47] respectively. In Chapter. 4, we use the same SOS-QE approach for the verification of "inevitability of oscillation" in ROs. The Gröbner basis has been used in [84] and [95] for the construction of invariants.

2.6 Formal Analog and Mixed Signal Circuits Verification

As mentioned in the last chapter, the conventional way of verifying AMS circuits is predominantly SPICE simulation. To improve its coverage, it was complemented by symbolic simulation to analyze parameter variation effects, but was still not enough to deal with non linear models. After successful use of formal methods in digital hardware verification, researchers have recently started applying the same for AMS circuit verification. A survey of different approaches can be found in [113]. Based on various methodologies used, we divide them into, Equivalence Checking, Model Checking, Runtime Verification, Deductive Methods.

2.6.1 Equivalence Checking

Equivalence checking is the art of checking whether two models of the same system, at different levels of abstraction, are equal to each other with reference to some criterion. In [12], Balivada et al. showed equivalence of a linear filter circuit and its specification. Represented as Laplace transformed transfer functions, the method discretized these using z-transform. Discrete domain binary decision diagrams (BDD) based equivalence was established between the two discrete models. In [49], the author computed value sets of an actual circuit and its specification transfer function for all possible parameter

variations and for a range of frequencies using interval arithmetic. The author showed by inclusion that value sets of actual circuit transfer function belong to the value sets of the specification transfer function. Hartong et al. in [48], used discretization of the state space and showed equivalence between vector fields of a behavioural model and a model that was obtained from the real circuit. The equivalence was established by showing that the modulus of the difference between the two vector field was bounded. In [82], the author showed equivalence of two VHDL-AMS models using rewriting rules and pattern matching to simplify models, and then using a SAT/BDD based approach to show their equivalence.

2.6.2 Model Checking

Model checking of AMS circuits exhaustively explores whether a model of AMS circuits satisfies a property. Models of the system could be discrete as well as continuous. In [63], Kurshan and McMillan presented the first approach to formally verify a digital circuit at a transistor level. They partitioned the state space in hypercubes as well as continuous input signals in to high and low logic with the assumption that they change values instantly. Time is similarly discretized in equal steps. They developed a transition relation between discrete states, and verified the model using the COPSON tool, against properties defined in ω-language. A similar discretization of state space based approach has been adopted in [48]. The difference here is that they used variable step based numerical integration, and adopted an automatic refinement of the discrete partitions so as to make them uniform. This process of partitions uniformity is based on the length and direction of the vector fields. Three types of transition relations between the partitions have been given. In one they used interval arithmetic to over-approximate the trajectories, whereas in the second type, they ran simulations at various points to establish transition relations. In the third, they made use of Lipschitz constants for non-linear functions. They implemented their approach in the tool AMCHECK and verified the discrete transition system against CTL properties.

Model checking techniques in continuous domain are based on reach set computation (reachability). Starting in an initial set, reachability techniques in each iteration compute the next set of points. To find the complete tube of trajectories, the pre and post reachable sets are bloated up to get the convex hull of the set of points, comprising pre, post and the sets between them. This way the tube of trajectories is conservatively over approximated. In [81], Mark Greenstreet and Ian Mitchell presented a technique showing reachable sets for a MOS circuit modelled as a system of ODEs. They showed

correctness of analog circuits by reachability considering three cases. First they considered linear ODEs and reachability computation was done based on the convex set representation. This was followed by computing reachable sets for linear ODEs with regions as non convex sets, and for non-linear ODEs with non-convex sets representing state space regions. Safety properties of analog circuits were soundly verified by conservatively over-approximating all reachable sets.

Modelling non-linear circuits as a HDS has gained tremendous interest in the research community during the last decade. This is mainly due to the amount of research that has been going on in formal verification of HDSs [9],[2]. HDSs consist of both continuous and discrete domains, making their verification a difficult task. In [38], Goran et al. verified time domain properties of a tunnel diode oscillator using the tool PHAVER. They showed that variations in amplitude and jitter in the oscillator behaviour are bounded. Modelling oscillator dynamics as a hybrid automata, having modes with affine dynamics of the form $\dot{x} = Ax + b$, PHAVER conservatively over-approximated this with a linear hybrid automata (LHA), where the affine dynamics were replaced with differential inclusion $a_l x \leq \dot{x} \leq a_u x$. They successfully showed that starting in close proximity to a limit cycle of a Tunnel diode oscillator (TDO), it oscillates with a specific fundamental period. A similar but improved approach involving forward/backward reachability has been adopted in [36].

Gupta et al. used the CHECKMATE tool for hybrid systems analysis, and verified time domain properties of the TDO [45]. A MATLAB based tool CHECKMATE can handle hybrid automata having modes with continuous affine dynamics. It uses flow pipe approximation (which is a sequence of polyhedra) and constructs sound abstraction of continuous dynamics. Instead of discretizing the whole state space, it partitions the state space only along the trajectory of the system for a set of its initial conditions. Creating discrete transition systems from the polyhedral invariant hybrid automaton, CHECKMATE uses bi-simulation based model checking and verifies ACTL properties for the given hybrid system. The author in [45], showed oscillation in the state space of the TDO for one set of parameters, and a counterexample for oscillation when a second set of parameters was considered.

In [94] Thao et al, using Mixed Integer Linear Programming (MILP) for discrete hybrid systems, showed worst case safety properties of a $\triangle - \sum$ modulator. Considering the fact that reachability algorithms suffers from time and space explosion, they adopted a bounded horizon reachability concept. Similar to boolean satisfiability for bounded horizon reachability in digital systems, they used concepts of optimal control, and looked for a worst input which induced bad behaviour. They proved safety by proving

safety of the set of worst trajectories. In the same paper, they verified a low pass filter modelled by differential algebraic equations (DAE). They transformed DAEs into ODEs, and computed reachable sets for them on manifolds using the d/dt reachability tool. Steinhorst et al. in [87] showed oscillations in a tunnel diode and a ring oscillator using visualization techniques. Being only applicable to three dimensional space, circuits with higher dimension were projected to three dimensions. Particles were injected in the discrete state space and their tangent vectors were approximated with the nearest point of the discrete vector field. The particles represented independent simulations, and thus gave a picture of the complete state space.

Althoff et al. in [7] formally verified lock time of a CP PLL. They used HDSs theory for the behavioural model of the CP PLL with parameter uncertainties. Faced with the problem of very large numbers of mode switching, they approximated the hybrid switched system with a continuous system, and used reachability computation to find all possible sets of the PLL transient analysis and computed its locking time. Using Labelled Hybrid Petri net (LHPN) analysis tools, Walter et al. in [100], verified switched capacitor integrator circuits. LHPN models have been transformed in to symbolic models, and then these were verified using BDD based model checking. A similar procedure has been adopted in [101], but the verification engine used an SMT solver to verify the symbolic model.

For a given property, a model checking algorithm invokes a decision procedure to traverse the state transition system, and verify whether or not the property is satisfied by that system. To deal with the state explosion problem, the model checking technique has been enhanced by bounded model checking (BMC) where the transition system is verified for a bounded length of state sequences [20]. BMC of hybrid systems involves predicate encoding of the sequential behaviour of transition system and the target formulas, and a decision procedure, e.g. SAT, to find a satisfying instantiation of the target formula. Using BMC, Zaki et al. in [112], proved properties of a $\triangle - \sum$ modulator and oscillator circuits. Representing continuous parts of AMS circuits by differential equation and the digital part by event based models, they used a interval arithmetic based Taylor approximation of the continuous state space to avoid unsoundness.

Recently Satisfiability Modulo Theory (SMT) based techniques have been used for AMS circuit verification. This is because of the recent advancement of SMT solvers to handle Boolean combinations of several thousand linear as well as non linear arithmetic constraints. Thiwary et al. in [57], presented a SAT modulo theory based approach for AMS circuits verification. Based on device (diodes, transistors) voltage current relationships, they tabulated these in the form of linear inequalities in their formulation.

Given these and the KCL/KVL constraints on current and voltages of different nodes in the circuit, they verified DC, steady state, and transient properties of circuits. They used Euler integration method to solve differential equations, which makes their technique less accurate due to soundness issues. In [110], Yin et al. proposed a methodology which is based on Nonlinear-SMT assisted by simulation. They used Bayesian inference rule to trade off between the computational cost of simulation and the number of SMT enquiries. Modelled as hybrid systems, they verified safety properties of PLL using reachability of unsafe states. Ishii et al. in [52], presented a Sat Modulo ODE technique for model checking of non linear hybrid automata, and verified oscillation property of the TDO. They tightly integrated SAT solver with hybrid constraints using an interval solver thus enabling it to deal directly with ODEs without approximating them. Chao et al. in [27] proposed SMT based reachability analysis using implicit integration methods to verify safety and liveness properties of arbiter circuits.

2.6.3 Deductive Methods (Theorem Proving)

Deductive methods, using a set of inference rules, establish proofs of mathematical theorems. Though unlike model checking, they do not suffer from state explosion, deductive methods require user assistance in formulating the problem.

In [41], Gosh et al. using higher order logic in a PVS theorem prover, verified DC and small signal properties of analog circuits. Their methodology used piecewise linearized models of devices, and VHDL for properties specification. They applied the methodology to small circuits consisting of operational amplifiers, resistors and transistors. Hanna in [46], abstracted digital circuits at analog level, and conservatively specified the circuit by rectilinear regions. He then verified the specification against implementation using proof techniques. Al Sammane et al. in [83], converted the circuit differential equations to system of recurrence equations (SRE) using rewriting rules, and verified the circuit properties by an induction proof method. Denman et al, in [30], used the meta Tarski theorem prover, and showed that for certain values of parameters, a TDO does not oscillate if it does not cross certain thresholds of the state variables. Though complete and sound, proof based methods suffer from several problems which debars their use for AMS circuits. It needs extensive inputs from the user to use the required axioms to be able to establish proofs. Secondly, because of the non-linear continuous nature of analog circuits, close form solutions can not be found, and therefore approximate set theoretic methods have to be used.

2.6.4 Run Time Verification

The above AMS circuit verification approaches have proved to be useful only for small block level circuits of few dimensions. The computation time it takes for almost all of these methods increases exponentially with the dimensionality of AMS circuits. Therefore other lightweight verification methods have also been used for AMS circuit verification.

In [28], Dastidar et al. presented a simulation method in which they generated a finite state machine (FSM) from a set of simulation traces, where current, voltage, and time are the state variables. Properties defined in the Ana CTL (CTL like logic) specification language have been checked by simulating the FSM discrete model of the circuit. In [70], Oded et al. presented signal temporal logic (STL) for specifying analog signal temporal properties in dense time. In addition to that they presented a monitoring logic for these properties to verify analog circuits. They implemented their work in AMT tool and verified a flash memory DDR2 DRAM. In [54], Jesser et al. presented a property assertion based approach to verify AMS circuits. Realizing the fact that there is a gap between analog and digital assertions, they came up with a mixed analog-digital signal assertions, and verified $\triangle - \sum$ circuit. Zaki et al. in [111], illustrated an interval arithmetic based simulation approach to verify CTL (Computation Tree Logic) properties for AMS circuits. In [5], Al Sammane et al. presented a monitoring algorithm for PSL properties of analog circuits. Furthermore, probabilistic model checking has been used in [21].

2.6.5 Oscillator Verification

An important property that has been verified for oscillator circuits is the global convergence to a limit cycle. In [38], the authors verified time domain properties of a TDO using the tool PHAVER. They showed that the variations in amplitude and jitter in the oscillator behaviour are bounded. [30] used the meta Tarski theorem prover and showed that for certain values of parameters, a TDO does not oscillate if it does not cross certain thresholds of the state variables; finding such thresholds is difficult for a nonlinear circuit model. A similar approach has been adopted in [45] for proving that a TDO oscillates for all possible initial conditions using an ACTL specification and CHECKMATE. Steinhorst et al, in [87], showed oscillations in TDO and ring oscillators using visualization techniques. Though complex behaviours in the state space can be visualized, absence of higher harmonics in oscillation was not formally verified. The RO start up problem, identified in [56], has been taken up in [44], [57]. They are based

on finding absence of a stable DC equilibrium point. While the former uses small signal analysis around the equilibrium point, the latter puts constraints on node voltages to establish stability of equilibrium points. Both these approaches are very localized and can not encapsulate the global behaviour of non-linear ring oscillator circuits. Chao et al. [108] verified oscillator start up using techniques from dynamical system theory.

Frequency domain approaches on the other hand are limited to small signal AC analysis of a more approximate linearized model around an equilibrium point. In [49], the author computed value sets of a transfer function for parameter variations and a range of frequencies using interval arithmetic. It was shown that the method was computationally very expensive, and its extension to non-linear circuits modelled as piece-wise affine would be a difficult task. Similarly, [64] derived amplitude and phase envelopes of a family of interval rational transfer functions for continuous-time systems. [30] used meta Tarski to prove that the magnitude of the transfer function of a small operational amplifier is bounded for a range of frequencies.

Chapter 3

Verifying the Inevitability of Phase-Locking in CP PLL

This chapter discusses a deductive-bounded verification approach for the inevitability of phase-locking in a CP PLL. In the first part of the chapter, we propose a mixed deductive-bounded methodology for the inevitability verification of a CP PLL. The second part talks about a deductive-only verification methodology for the same purpose.

In the mixed deductive-bounded methodology, we verify the inevitability of phase-locking in a CP PLL by adopting a two-pronged verification approach. Due to the complexity of the property, we essentially divide the inevitability property in to the conjunction of two sub-properties. These two properties determine the truth value of the inevitability property in two disjoint subsets of the state space. The first property specifies, that in a compact attractive invariant set, all system trajectories eventually converge to the equilibrium locking state. The second property is specified such that the set, where the first property holds, is reachable from the set of all outside states. The first property is verified by determining an attractive invariant set utilizing the deductive Lyapunov certificate for the stability of the HDS. This is achieved by constructing multiple Lyapunov certificates for different continuous flow maps of the CP PLL HDS. The maximized level surfaces of these Lyapunov certificates characterize the sub-level sets whose union is the attractive invariant set. We take advantage of both deductive and bounded approaches to verify the second property. Using bounded advection of level sets, introduced in Sec. 2.5.1.2, and a deductive Escape certificate, we show that the attractive invariant set is reachable from every state outside it. The deductive-bounded verification approach involves checking positivity/negativity of real polynomials, which

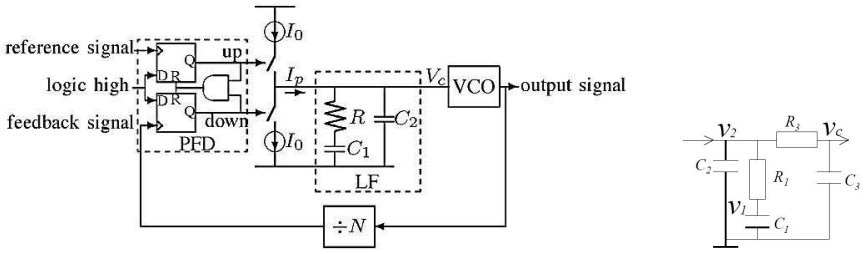


Figure 3.1: *CP PLL, Left: Third Order CP PLL, Right: Fourth order LF*

is an NP-hard problem. We therefore use the sound but incomplete SOS relaxation, discussed in Sec. 2.4.2, for the verification of polynomial positivity/negativity.

The deductive-only verification methodology is a certificate based verification of the inevitability of phase-locking in higher order CP PLL circuits. Similar to the deductive-bounded approach, here too, we divide the inevitability property in to the conjunction of two sub-properties. Verification of these two properties determines the truth value of the inevitability property in two disjoint subsets of the state space. The first property is verified using the Lyapunov certificate based deductive approach, similar to the mixed deductive-bounded methodology. To verify the second property, we use the Escape certificate showing that trajectories in the second set will eventually escape and reach the set where the first property is satisfied.

3.1 Preliminaries of the Verification Methodologies

3.1.1 HDS Modelling of CP PLL

A PLL circuit is responsible for tracking the phase and frequency of the input. In its simple form, a CP PLL circuit consists of a reference signal, a phase frequency detector (PFD), a charge pump (CP), a loop filter (LF), VCO and a frequency divider. In this thesis, we consider a single path higher order CP PLL, discussed in [103], and shown in Fig. 3.1. Here we have shown a third order CP PLL; the fourth order CP PLL is the same, having a fourth order LF instead. Furthermore, though we discuss HDS modelling of a third order CP PLL, it however is applicable to CP PLL of any order. Following the top down modelling strategy of Sec. 2.2, we use a behavioural model of the CP PLL. We consider a linear model for the VCO, a linear model for the LF, and a piece-wise continuous model for the PFD. Due to the low cut off frequency of the LF, the overall bandwidth of the closed loop CP PLL system is very low as compared to the operating frequency of the VCO. Therefore, the high frequency non-linear transients

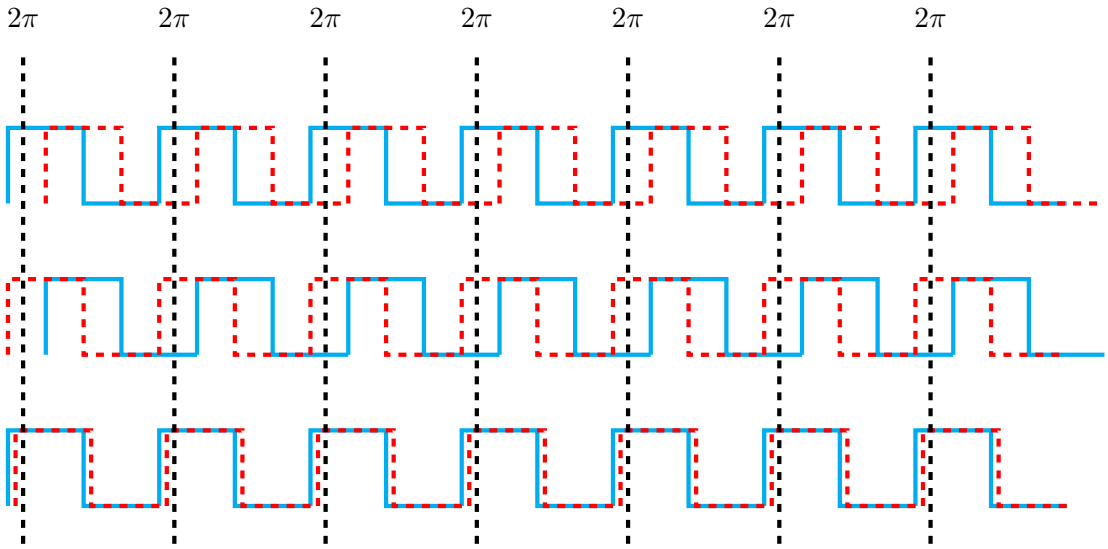


Figure 3.2: *Piece-wise Continuous Behaviour of PFD, Cyan Solid: ϕ_{VCO}, Red Dotted: ϕ_{ref}*

of the VCO has a negligible effect on the overall behaviour of the CP PLL. Therefore, our linear model assumption of the VCO is realistic and reduces the complexity of the overall model. We denote by ϕ_{ref}, and ϕ_{VCO}, the phases of the reference and VCO output feedback signals respectively.

We model the CP PLL using the formal HDS model described in Sec. 2.1.2. This is done such that the non-linearity of the PFD is modelled as a piecewise continuous signal. Ignoring the cycle slip phenomena, the PFD output, in the form of the CP current I_p is given by the following piecewise linear inclusion:

$$I_p = \begin{cases} \in [I_p^U \, I_p^U] & \text{UP=1, Down=0, } 0 \leq \phi_{VCO} < 2\pi \leq \phi_{ref} \\ \in [I_p^D \, I_P^D] & \text{UP=0, Down=1, } 0 \leq \phi_{ref} < 2\pi \leq \phi_{VCO} \\ \in [0^R \; 0^R] & \text{UP=0, Down=0, } 0 \leq \phi_{VCO}, \phi_{ref} < 2\pi \end{cases} \tag{3.1}$$

The three operating modes of the PFD are pictorially shown in Fig. 3.2. Here all phases except 2π are normalized by 2π. The plot at the top shows ϕ_{ref} is leading both ϕ_{VCO} and the 2π threshold. Similarly, the middle plot is the case when ϕ_{VCO} leads both ϕ_{ref} and the 2π threshold, whereas the bottom plot shows both ϕ_{VCO} and ϕ_{ref} lagging the 2π threshold. We denote these three modes as mode1 (UP=0, Down=0), mode2 (UP=1, Down=0) and mode3 (UP=0, Down=1). The transition from one mode to another is based on the reference and feedback signals hitting the 2π threshold. Due to the cyclic behaviour of the PLL, and to keep the analysis modulo 2π, we need to

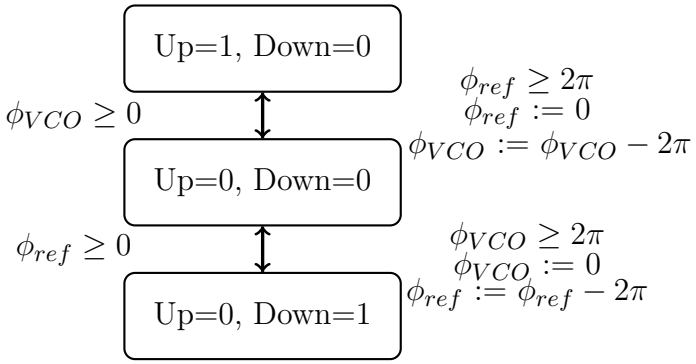


Figure 3.3: *Hybrid Model of CP PLL*

ensure the phases remain in the range $0 \leq \phi_{VCO}, \phi_{ref} < 2\pi$ after resetting the PFD. This is achieved by resetting the two phases such that,

$$\phi_{ref} := 0,\ \phi_{VCO} := \phi_{VCO} - 2\pi, \tag{3.2}$$

$$\phi_{VCO} := 0,\ \phi_{ref} := \phi_{ref} - 2\pi, \tag{3.3}$$

while taking transitions from mode1 to mode2 and mode1 to mode3, respectively. Identity resets are used for transitions from mode2 to mode1 and mode3 to mode1. The HDS model of the CP PLL is shown in Fig. 3.3. Our model consists of the state variables, ϕ_{VCO}, ϕ_{ref}, voltage v_1 across the capacitor C_1, and the voltage v_2 across the capacitor C_2 (fourth order has an additional voltage variable across the third capacitor). Let f_{VCO}, and f_{ref}, represent frequencies of the VCO output and the reference signal respectively. If K_p is the gain of the LF, then,

$$f_{VCO} = K_p v_2/2\pi + f_O \tag{3.4}$$

where f_O is the free running frequency of the VCO. Therefore,

$$\dot{\phi}_{VCO} = 2\pi f_{VCO}/N,\ \dot{\phi}_{ref} = 2\pi f_{ref} \tag{3.5}$$

By Kirchhoff's current law, the two voltages v_1 and v_2 across C_1 and C_2 are given by the following two equations,

$$\dot{v_1} = \frac{-1}{RC_1} v_1 + \frac{1}{RC_1} v_2 \tag{3.6}$$

$$\dot{v_2} = \frac{1}{RC_2} v_1 - \frac{1}{RC_2} v_2 + \frac{I_p}{C_2} \tag{3.7}$$

Therefore, depending on the three modes of the PFD, we get the following HDS model of the third order CP PLL (similarly fourth order),

$$\mathcal{H} = \begin{cases} \dot{\begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix}} = A \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} + BI_p + c & x \in \mathcal{C}, \\ x^+ = G_i(x) & x \in \mathcal{D} \end{cases} \tag{3.8}$$

Here,

$$A = \begin{pmatrix} -1/RC_1 & 1/RC_1 & 0 & 0 \\ 1/RC_2 & -1/RC_2 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & K_p/N & 0 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 0 \\ 1/C_2 \\ 0 \\ 0 \end{pmatrix}, \; c = \begin{pmatrix} 0 \\ 0 \\ 2\pi f_{ref} \\ 2\pi f_o/N \end{pmatrix},$$

$$\mathcal{F} = A \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} + BI_p + c$$

$$F_1(x,u) = \left\{ A \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} + BI_p(\in [0^R 0^R]) + c \;\middle|\; \phi_{ref} \in \phi_{ref} \geq 0 \; or \; \phi_{VCO} \in \phi_{VCO} \geq 0 \right\},$$

$$F_2(x,u) = \left\{ A \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} + BI_p(\in [I_p^U I_P^U]) + c \;\middle|\; \phi_{ref} \in \phi_{ref} \geq 2\pi \right\},$$

$$F_3(x,u) = \left\{ A \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} + BI_p(\in [I_p^D I_P^D]) + c \;\middle|\; \phi_{VCO} \in \phi_{VCO} \geq 2\pi \right\},$$

$$\mathcal{C} = \Bigg\{ [v_1\ v_2\ \phi_{ref}\ \phi_{VCO}] \ \Big|\ -10 \leq v_1 \leq 10 \text{ and } -10 \leq v_2 \leq 10 \text{ and } -2\pi \leq \phi_{ref} \leq 2\pi \text{ and } -2\pi \leq \phi_{VCO} \leq 2\pi \Bigg\},$$

$$\mathcal{D} = \Bigg\{ [v_1\ v_2\ \phi_{ref}\ \phi_{VCO}] \ \Big|\ \phi_{ref} \geq 2\pi \text{ and } \phi_{VCO} \geq 0 \text{ and } \phi_{VCO} \geq 2\pi \text{ and } \phi_{ref} \geq 0 \Bigg\},$$

$$G_1(x) = \Bigg\{ \begin{pmatrix} v_1 \\ v_2 \\ 0 \\ \phi_{VCO} - 2\pi \end{pmatrix} \ \Big|\ \phi_{ref} \in \phi_{ref} \geq 2\pi \Bigg\},$$

$$G_2(x) = \Bigg\{ \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} - 2\pi \\ 0 \end{pmatrix} \ \Big|\ \phi_{VCO} \in \phi_{VCO} \geq 2\pi \Bigg\},$$

$$G_{3 \text{ or } 4}(x) = \Bigg\{ \begin{pmatrix} v_1 \\ v_2 \\ \phi_{ref} \\ \phi_{VCO} \end{pmatrix} \ \Big|\ \phi_{ref} \in \phi_{ref} \geq 0 \ \textit{or}\ \phi_{VCO} \in \phi_{VCO} \geq 0 \Bigg\}.$$

The desired behaviour of the CP PLL output is such that we have a periodic limit cycle in the ϕ_{ref}, ϕ_{VCO} plane. This is shown in the simulation traces in Fig. 3.4. Here ϕ_{ref} and ϕ_{VCO}, are normalized by 2π, and the simulation time is in micro-seconds. As can be seen in Fig. 3.4, state variables ϕ_{ref} and ϕ_{VCO} do not converge to zero, and the system has a limit cycle like behaviour in the (ϕ_{ref}, ϕ_{VCO}) plane. To apply the Lyapunov certificate based stability analysis discussed in Sec. 2.3.2, we need to have equilibrium at the origin. Therefore, we use transformation of axis and introduce a new

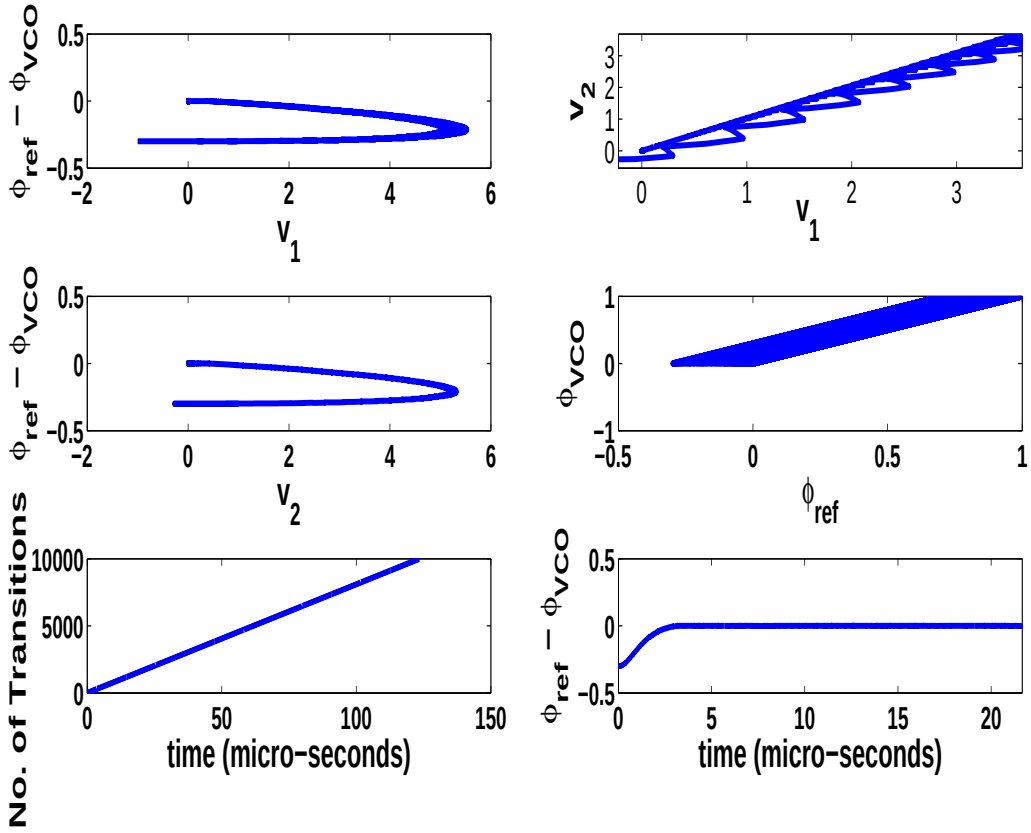


Figure 3.4: *Simulation Plots of the CP PLL Hybrid System*

variable, $\phi_D = \phi_{ref} - \phi_{VCO}$, such that the new HDS $\mathcal{H}'$ has the equilibrium state,

$$x_e = \begin{pmatrix} v_1 \\ v_2 \\ \phi_D \end{pmatrix}^T = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}^T$$

Note that this change of variable not only shifts the equilibrium to the origin for the new HDS $\mathcal{H}'$, but reduces the number of system dimensions by one as well. This is a by-product of this variable transformation which reduces the computation cost of the verification. Accordingly, we also make the necessary changes in I_p, A, B, $\mathcal{C}$, $\mathcal{D}$, and G_i.

Remark 3.1. *This change of state variables transforms all jump maps G_i into identity maps, i.e. $G_i(x) = x$, since the same constant 2π is subtracted from ϕ_{VCO} and ϕ_{ref}, leaving their difference $\phi_{ref} - \phi_{VCO}$ before and after the jumps unchanged.*

Let us denote by $x(t,j)$ the solution of HDS $\mathcal{H}^{'}$. We now define the inevitability property for the HDS $\mathcal{H}^{'}$ in the following definition.

Definition 3.1 (Inevitability of Equilibrium)**.** *The equilibrium point x_e is said to be inevitable, if, $lim_{t+j\to\infty}\ x(t,j) = x_e,\ \forall x(0,j) : x(0,j) \in \mathcal{X}_{\mathcal{H}'}$.*

For all practical purposes, the origin of the hybrid state space $\mathcal{X}_{\mathcal{H}'}$ is not invariant, and in fact it is a small ball, $\mathcal{B}_r = \{y \in \mathcal{X}_{\mathcal{H}'} \mid \|y - x_e\| \leq r\}$, around the origin, $x_e = 0$, which acts as an invariant equilibrium set for the system. In this thesis, we verify the inevitability of the equilibrium state at the origin which can easily be extended to the inevitability of a ball $\mathcal{B}_r$ around the equilibrium.

3.1.2 Attractive Invariance of a Set and Escape of trajectories from a Set

Contrary to the safety properties, where existence of an invariant set having an empty intersection set with the unsafe state is sufficient for proving/dis-proving the property, we use the concept of "attractive invariants" for an inevitability property. Furthermore, we use another important concept which characterizes the escape of solutions of the HDS from a compact set. For the HDS model $\mathcal{H}^{'}$ of the CP PLL, an attractive invariant set is a compact semi-algebraic set where its solutions set remains forever and eventually converge to the equilibrium locking state. Proving that a set is attractive invariant, with respect to an equilibrium, is a difficult problem to solve. We use the Lyapunov certificate, discussed in Sec. 2.3.2 for the HDS, and construct an attractive invariant set with respect to the equilibrium state $x_e = 0$.

Proposition 3.1. *For the HDS $\mathcal{H}^{'}$ of the CP PLL, the set $\mathcal{X}_{AI} \subset \mathcal{X}_{\mathcal{H}'}$ is called an attractive invariant if there are multiple Lyapunov certificates $V_i(x),\ \forall i \in I_C$, satisfying the conditions of Th. 2.3, and the following condition holds,*

$$\mathcal{X}_{AI} = \{\bigcup_i (V_i \leq c_{max})\} \subset \mathcal{X}_{\mathcal{H}},\ c_{max} > 0. \tag{3.9}$$

Proof. Follows directly from Th. 2.3. □

An another important characteristic of HDS solutions in a semi algebraic set is the Escape from the set property. Escape from a set is the dual of invariance of a set such that a set fulfilling this property can not be an invariant set, and all possible solutions of the HDS escape from it in bounded time. Similar to the attractive invariance property

of a set, we verify the Escape property of a set using a Lyapunov like certificate called the Escape certificate. This certificate is illustrated in the following proposition.

Proposition 3.2. *For a compact set $\mathcal{X}_e \subset \mathcal{X}_{\mathcal{H}'}$, if there is a differentiable Escape certificate, $E : \mathbb{R}^n \to \mathbb{R}$, and $\epsilon > 0$, such that*

$$\frac{\partial E}{\partial x}(x)F_i(x,u) \leq -\epsilon,\ \forall x : x \in \mathcal{X}_e,\ \forall u : u \in \mathcal{U},\ \forall i : i \in I_C, \tag{3.10}$$

then $\forall x(t,i) : x(t,i) \in \mathcal{X}_e,\ x(t+T,i) \notin \mathcal{X}_e,$ for $T > t$.

Proof. Assume that there exists $x_0 \in \mathcal{X}_e$ such that $x(t,i)$ starting at x_0 remain in $\mathcal{X}_e$ as $t \to \infty$. From equation. 3.10,

$$E(x) = \int_0^\infty \frac{\partial E}{\partial x}(x)F_i(x,u) \leq -\epsilon$$

As $t \to \infty,\ E(x) \to -\infty$. This contradicts the assumption as $E(x)$ should be bounded from below if $x(t,j)$ has to be in the bounded set $\mathcal{X}_e$. Therefore, $x(t,j)$ has to eventually escape $\mathcal{X}_e$ in finite time. □

While verifying inevitability of CP PLL, we make use of Prop. 3.2 and borrow a lemma from [61, Lemma 4.1] extended for HDS. This is to show that if Escape of solutions from a set is verified, they must eventually reach an invariant set. This is stated in the following lemma for an HDS.

Lemma 3.1. *If the hybrid arc $x(t,j)$ is bounded and belongs to a set $\mathcal{X} \subset \mathcal{X}_{\mathcal{H}'}$ for the hybrid time $(t,j) \geq 0$, then $x(t,j)$ approaches a compact invariant set as $(t,j) \to \infty$.*

Proof. See [61, Lemma 4.1]. □

Proposition 3.3. *Let $\mathcal{X}_{\mathcal{H}'} = \mathcal{X}1 \cup \mathcal{X}2,\ \mathcal{X}1 \cap \mathcal{X}2 = \emptyset$, and assume that $\mathcal{X}1$ is an invariant set. If there is an Escape certificate in the set $\mathcal{X}2$ satisfying conditions of Prop. 3.2, then $\forall x(0,j) : x(0,j) \in \mathcal{X}2,\ \lim_{t \to b(\geq 0)} x(t,j) \in \mathcal{X}1$.*

Proof. Follows directly from Lemma. 3.1. Since the existence of an Escape certificate guarantees that trajectories will leave the set $\mathcal{X}2$, therefore, they must reach the invariant set $\mathcal{X}1$. □

3.1.3 Bounded Advection of Level Sets in HDS

In the mixed deductive-bounded verification methodology, we use bounded advection of level sets, described in Sec. 2.5.1.2, to verify a sub-property of the inevitability. In

this section, we extend this advection of level sets for HDS. We assume the identity reset maps as is the case for the transformed HDS model of the CP PLL.

Let us define a hybrid flow map $\Psi : \mathbb{R}^n \times \mathcal{T} \to \mathbb{R}^n$ for the CP PLL HDS $\mathcal{H}'$. Furthermore, let us denote by $C : \mathbb{R}^n \to \mathbb{R}$ the set of all differentiable maps from Euclidean space to the real number set. For polynomials, $\mathcal{P}_1 \in C(\mathbb{R}^n, \mathbb{R})$ and $\mathcal{P}_2 \in C(\mathbb{R}^n, \mathbb{R})$, an advection operator A_t, for $t : (t, j) \in \mathcal{T}$, is a map

$$A_t : C(\mathbb{R}^n, \mathbb{R}) \ \to C(\mathbb{R}^n, \mathbb{R}),\ t \in \mathcal{I}_j : (t, j) \in \mathcal{T},\ j \in \mathbb{N} \tag{3.11}$$

such that,

$$\mathcal{P}_2 = A_t \mathcal{P}_1 \tag{3.12}$$

and

$$\mathcal{P}_2(x) = \mathcal{P}_1(\Psi_{-t}(x)) \text{ for all } x : x \in \mathcal{X}_{\mathcal{H}'}. \tag{3.13}$$

This advection operator has an important property of linearity. For polynomial functions $U1, U2 \in C(\mathbb{R}^n, \mathbb{R})$, if,

$$U2 = A_t U1,\ t : (t, j) \in \mathcal{T}$$

then

$$\mathcal{Z}(U2) = A_t(\mathcal{Z}(U1)),\ t : (t, j) \in \mathcal{T} \tag{3.14}$$

where $\mathcal{Z}(.)$ is the zero sub-level set as defined in Sec. 2.4.4. Similar to the advection of level sets in the CDS discussed in Sec. 2.5.1.2, we use an approximation to the flow map A_h with an exception, that for each $i \in I_C$, we have different approximations to the advection map A_h such that, $h = t_2 - t_1,\ t_1; (t_1, j) \in \mathcal{T},\ t_2 : (t_2, j) \in \mathcal{T}$. For example, a first order Taylor approximation yields the following advection of a set,

$$U = B_h^i V, \text{ if } U(x) = V(x) - h\frac{\partial}{\partial x} V(x) F_i(x, u),\ x \in C_i,\ i \in I_C \tag{3.15}$$

Here $B_h^i : C(\mathbb{R}^n, \mathbb{R}) \to C(\mathbb{R}^n, \mathbb{R})$ is the first order Taylor approximation to A_h^i. Similar to the CDS, the product $B_h^i V$ results in a polynomial with a degree higher than V, therefore, a conservative approximation to the advected level sets is used. Introducing an approximation parameter $\mu_i \in \mathbb{R}_{\geq 0}$, we conservatively approximate the zero sub-

level set of a polynomial q by backward advecting the polynomial p such that,

$$\mathcal{Z}(q) \subset \mathcal{Z}(B^i_{-h}p) \subset \mathcal{Z}(q - \mu_i),\ i \in I_C \tag{3.16}$$

To incorporate the truncation error due to Taylor approximation, let us introduce η_i such that $\|\nabla^2 p\frac{h^2}{2}\| \leq \eta_i$. This requires that

$$\mathcal{Z}(B^i_{-h}p + \eta) \subset \mathcal{Z}(A^i_{-h}p) \subset \mathcal{Z}(B^i_{-h}p - \eta_i) \tag{3.17}$$

This implies that,

$$\mathcal{Z}(q) \subset \mathcal{Z}(B^i_{-h}p + \eta) \subset \mathcal{Z}(A^i_{-h}p) \subset \mathcal{Z}(B^i_{-h}p - \eta) \subset \mathcal{Z}(q - \mu_i) \tag{3.18}$$

We incorporate these set inclusions as a SOS program which will be discussed later in this chapter.

3.2 Mixed Deductive-Bounded Verification Methodology

The first approach that we present to verify the inevitability of the equilibrium of the HDS model $\mathcal{H}^{'}$ is mixed deductive-bounded verification. The idea is to split the verification task into two smaller tasks and verify them using deductive and mixed deductive-bounded verification approaches. This helps in tractability of the problem and reduces the overall computational cost of the verification. Essentially, we introduce two compact sets $\mathcal{S}1$, and $\mathcal{S}2$, such that $\mathcal{S}1 \cap \mathcal{S}2 = \emptyset$, and $\mathcal{S}1 \cup \mathcal{S}2 = \mathcal{X}_{\mathcal{H}'}$. This is pictorially depicted in Fig. 3.5. We define two sub-properties in these two sets such that the verification of the inevitability of phase-locking is boiled down to the verification of the conjunction of these two properties. These two sub-properties are formally defined as follows,

Property 3.1. $\forall x(0,j) : x(0,j) \in \mathcal{S}1,\ \lim_{t\to\infty} x(t,j) = x_e$

Property 3.2. $\forall x(0,j) : x(0,j) \in \mathcal{S}2 = (\mathcal{X}_{\mathcal{H}'} \setminus \mathcal{S}1),\ \lim_{t\to b(\in\mathbb{R}_{>0})} x(t,j) \in \mathcal{S}1.$

If we denote the inevitability property by φ, Property. 3.1 by $\varphi 1$ and Property. 3.2 by $\varphi 2$, then,

$$\varphi = \varphi 1 \wedge \varphi 2 \tag{3.19}$$

A hybrid arc x satisfies φ, iff, it satisfies $\varphi 1$ in $\mathcal{S}1$ and $\varphi 2$ in $\mathcal{S}2$ i.e.,

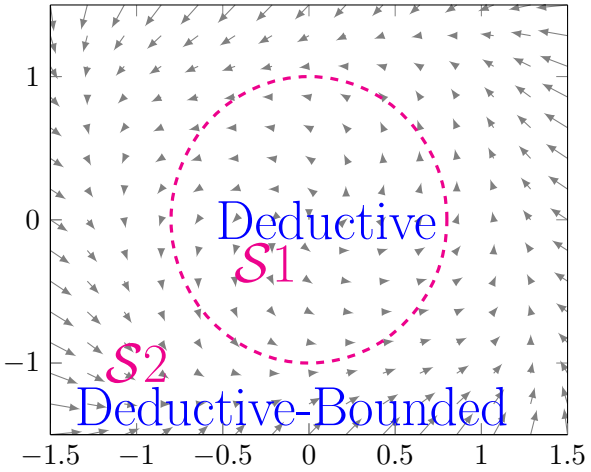


Figure 3.5: *Verification Methodology, Two Properties in Two Disjoint Subsets*

$$\forall x : x \in \mathcal{X}_{\mathcal{H}'},\ x \models \varphi \iff (x \models \varphi 1\ \forall x : x \in \mathcal{S}1)\ \wedge (x \models \varphi 2\ \forall x : x \in \mathcal{S}2) \tag{3.20}$$

3.2.1 Deductive Verification of $\varphi 1$

The property $\varphi 1$ is essentially characterizing attractive invariance of the set $\mathcal{S}1$. As stated in Prop. 3.1, a set can be proven to be an attractive invariant by using multiple Lyapunov certificates for HDS. Therefore, we verify $\varphi 1$ for the CP PLL HDS $\mathcal{H}'$ using the deductive Lyapunov certificate approach. The following theorem describes a necessary condition for this purpose.

Theorem 3.1. *For the HDS $\mathcal{H}'$, If set $\mathcal{S}1$ is an attractive invariant, then*

$$x \models \varphi 1,\ \forall x(0,j) : x(0,j) \in \mathcal{S}1 \tag{3.21}$$

such that

$$\mathcal{S}1 = \bigcup_i (V_i \leq \gamma^i_{max}),\ \gamma^i_{max} > 0,\ i \in \{1,..,\ell\}. \tag{3.22}$$

Proof. Follows directly from Prop. 3.1. Attractive invariance of set $\mathcal{S}1$ ensures that every hybrid arc x of the HDS $\mathcal{H}'$ eventually converges to the equilibrium state x_e, and hence is the model of $\varphi 1$. □

Algorithm 1 Verification of Property $\varphi 1$

INPUT: : HDS Model of CP PLL
OUTPUT: : $\varphi 1$ Verified/No-answer, $\mathcal{S}1$

1: $\mathcal{S}1 \leftarrow \emptyset$
2: **for** $i \leftarrow 1$ **to** $i \leftarrow \ell$ **do** ; Here ℓ is the no. of discrete modes of the HDS
3: $\quad V_i \leftarrow Parametrize(V_i)$; Setting degree d and coefficients of V_i Polynomials
4: **end for**
5: **if** $V_i, \forall i \in \{1, .., \ell\}$, are feasible (fulfilling Th. 4.1) **then**
6: $\quad V_{multiple} \leftarrow \{V_{multiple}, V_i\},\ \forall i \in \{1, .., \ell\}$
7: $\quad \mathcal{S}1 \leftarrow \bigcup_i (V_{multiple}(i) \leq (\gamma_i)_{max}),\ (\gamma_i)_{max} > 0,\ i \in \{1, .., \ell\}$
8: $\quad x \models \varphi 1,\ \forall x \in \mathcal{S}1$
9: **else**
10: $\quad V_i \leftarrow Infeasible$
11: $\quad$ Increase Degree d of V_i ; d is incremented by 2
12: $\quad$ Goto Line(2) if $d < b$; Here b is a user-defined upper bound on degree d
13: **end if**
14: **if** d = b & $x \not\models \varphi 1,\ \forall x \in \mathcal{S}1$ **then**
15: $\quad$ No Answer about $\varphi 1$
16: **end if**
17: **return** $\mathcal{S}1$ and Truth value of $\varphi 1$

Therefore, to verify property $\varphi 1$, we search for multiple Lyapunov certificates satisfying the conditions of Th. 2.3. Formally, the conditions of Th. 2.3 can be formulated as a FOF in the non-linear polynomials over real numbers with universal-existential quantifiers. Though there are techniques to eliminate quantifiers from the quantified FOFs, their worst-case complexity is however doubly exponential in the number of variables and they work for problems of trivial complexity (low dimension). Therefore, we make use of SOS programming and numerically search for feasible Lyapunov certificates.

We verify $\varphi 1$ following the steps underlined in Alg. 1. The truth value of $\varphi 1$ depends on the existence of the attractive invariant set $\mathcal{S}1$. The set $\mathcal{S}1$ is computed from the maximized level sets characterized by the level surfaces of the candidate Lyapunov certificates $V_i,\ i \in \{1, .., \ell\}$. Alg. 1 is encoded as two separate SOS programs. The input of the algorithm is the HDS model $\mathcal{H}'$ of the CP PLL, whereas its output is the truth value of the property $\varphi 1$ and the set $\mathcal{S}1$. The algorithm starts with initializing the set $\mathcal{S}1$ and parametrizing Lyapunov certificates V_i by setting up their degrees and coefficients Line 2-3. Degree d is initially set up to be 2, and it is incremented by 2 in each iteration of the algorithm. This is followed by searching for candidate certificates

V_i, Line-5. We encode this search as a SOS program following the S-procedure discussed in Sec. 2.4.4. A similar procedure for constructing multiple Lyapunov certificates has been given in [78]. Before illustrating the SOS program, we outline how flow sets defined by C_i, and jump sets defined by D_i are represented as semi-algebraic sets.

The C_i of the HDS $\mathcal{H}^{'}$ can be represented as a semi-algebraic set given below,

$$C_i(x) = \{x \in \mathbb{R}^n \ : g_{ik}(x) \geq 0, \text{ for } k \in \{1, .., n_{C_i}\}, \ i \in \{1, .., \ell\}\}. \tag{3.23}$$

Here $g_{ik}(x)$ is a vector of polynomials, and the inequality conditions are held entry wise. For example, for the three dimensional hypercube of the third order CP PLL HDS $\mathcal{H}^{'}$, $g_{ik}(x)$ is given as,

$$g_{ik}(x) = \begin{pmatrix} (v_1 - v_1^L)(v_1^U - v_1) \\ (v_1 - v_1^L)(v_1^U - v_1) \\ (\phi_D - \phi_D^L)(\phi_D - \phi_D^U) \end{pmatrix}$$

Similarly, we represent jump sets D_i by the following semi-algebraic set,

$$D_i(x) = \{x \in \mathbb{R}^n \ : h_{ik}(x) \geq 0, \ h_{i0}(x) = 0, \text{ for } k \in \{1, .., n_{D_i}\}, \ i \in \{1, .., \ell^{'}\}\}. \tag{3.24}$$

Here apart from the inequality constraints $h_{ik}(x) \geq 0$, we have equality constraints $h_{i0}(x) = 0$. Both these constraints are vectors of polynomials and are held entry wise. We also represent the interval bounds on the parameters by the following vector of polynomials,

$$\{a_j(u) \geq 0, \text{ for } j \in \{1, .., m\}\} \tag{3.25}$$

The SOS program that implements Line-5 of Alg. 1 is given below,

$$V_i(0) = 0, \ \forall i \in \mathcal{I}_0 \tag{3.26}$$

$$\left(V_i(x) - \epsilon - \sum_{k=1}^{n_{C_i}} s_1^{(ik)}(x) g_{ik}(x) \right) \in \mathcal{S}_n, \forall x \neq 0, i \in \{1, .., \ell\}, \forall k \in \{1, .., n_{C_i}\}, \quad s_1^{(ik)} \in \mathcal{S}_n, \epsilon > 0, \tag{3.27}$$

$$\left(-\frac{\partial V_i}{\partial x}(x)F_i(x,u) - \epsilon - \sum_{k=1}^{n_{C_i}} s_2^{(ik)}(x)g_{ik}(x) - \sum_{j=1}^{m} s_3^{j}(x)a_j(u) \right) \in \mathcal{S}_n, \\ \forall i \in \{1,..,\ell\}, \forall k \in \{1,..,n_{C_i}\}, \forall j \in \{1,..,m\}, (s_2^{(ik)}, s_3^{j}) \in \mathcal{S}_n, \epsilon > 0, \tag{3.28}$$

$$\left(V_j(x) - V_{j'}(G_i(x)) - s_4^{(i0)}(x)h_{i0}(x) - \sum_{k=1}^{m_{D_i}} s_5^{(ik)}(x)h_{ik}(x) \right) \in \mathcal{S}_n, \forall j \forall j' \in \{1,..,\ell\}, \\ j \neq j', \forall i \in \{1,..,\ell'\}, \forall k \in \{1,..,n_{D_i}\}, s_4^{(i0)} \geq 0,\ s_5^{(ik)} \in \mathcal{S}_n. \tag{3.29}$$

Here $V_i(x)$, $V_j(x)$, $V_{j'}(x)$, $s_1^{(ik)}$, $s_2^{(ik)}$, $s_3^{(j)}$, $s_4^{(i0)}$, $s_5^{(ik)}$, are polynomials of degree d, and $\mathcal{I}_0 \subseteq I_C$ is the set of indices having the equilibrium.

In this SOS program, every constraint is a sound implementation of a condition in Th. 2.3. Equality constraints in Eq. 3.26 make sure that $V_i(x)$ is zero at the origin. Constraints in Eq. 3.27 enforce positive definiteness on Lyapunov certificates, whereas Eq. 3.28 ensures negative definiteness of their Lie-derivatives. Following the S-procedure technique of Sec. 2.4.4, additional flow set constraints have been added to both Eq. 3.27 and Eq. 3.28. Also in Eq. 3.28, we have additional parameter constraints. Constraints in Eq. 3.29 ensure that Lyapunov certificates $V_j(x)$ decrease along the discrete jumps in the set D_i through the mappings $G_i(x)$'s. SOS polynomials $s_1^{(ik)}$, $s_2^{(ik)}$, $s_3^{(j)}$, $s_4^{(i0)}$, $s_5^{(ik)}$ are used to enforce domain constraints through the S-procedure. A feasible solution of the above SOS program results in Lyapunov certificates V_i.

Proposition 3.4. *If the SOS program of Eq. 3.27, Eq. 3.28, and Eq. 3.29 is feasible, then the Lyapunov certificates $V_i(x)$, $i \in I_C$ satisfy the conditions of Th. 2.3.*

Proof. Eq. 3.26 is trivial. In Eq. 3.27, the multipliers $s_1^{(ik)}(x)$ are SOS and $g_{ik}(x) \geq 0$. Also expressions $V_i(x) - \epsilon - \sum_{k=1}^{n_{C_i}} s_1^{(ik)}(x)g_{ik}(x)$ are SOS. Therefore,

$$V_i(x) - \sum_{k=1}^{n_{C_i}} s_1^{(ik)}(x)g_{ik}(x) \geq \epsilon.$$

Since $\epsilon > 0$, we have $V_i(x) > 0$, for $x \neq 0$, which is the second condition, Eq. 2.15, of Th. 2.3. Similar proof can be given for other conditions as well. □

If this SOS program is infeasible, then either the program is repeated for an increased

degree d of the polynomials, or we conclude, if d is equal to the user defined upper bound b, that the truth value of the property $\varphi 1$ can not be established (Line 10-17).

The next step in Alg. 1 is to compute the set $\mathcal{S}1$ (Line 7). This is computed from the maximized level surfaces of Lyapunov certificate $V_i(x)$ and to perform this maximization, we use the following SOS program for every $V_i \leq \gamma_i$.

$$\begin{aligned} &\text{maximize:} && \gamma_i \\ &\text{subject to} && s5(x) + \sum_{k=1}^{n_{C_i}} s6^{ik}(x)(-g_{ik})(x) - (V_i(x) - \gamma_i) + \epsilon = 0, \\ & && (s5, s6^{ik}) \ \in \mathcal{S}_n, \ \gamma_i > 0, \ \epsilon > 0, \ i \in \{1, .., \ell\}, \ k \in \{1, ..., n_{C_i}\}. \end{aligned} \tag{3.30}$$

Proposition 3.5. *If the SOS optimization program in Eq. 1 is feasible, then,*

$$\mathcal{Z}(V_i - (\gamma_i)_{max}) \subset \mathcal{Z}(-g_{ik}), \ \textit{for } k \in \{1, .., n_{C_i}\} \tag{3.31}$$

Proof. Follows directly from Lemma. 2.2. □

For the maximized level curves of the Lyapunov certificates, we compute the set $\mathcal{S}1$ by

$$\mathcal{S}1 = \bigcup_{i=1}^{\ell} (V_i \leq (\gamma_i)_{max}) \tag{3.32}$$

The non-emptiness of the set $\mathcal{S}1$ shows that, $x \models \varphi 1, \ \forall x : x \in \mathcal{S}1$ (Line 8). As mentioned earlier, if for a maximum degree bound b of d, we are unable to find feasible Lyapunov certificates $V_i(x)$, we conclude inconclusiveness about the truth value of $\varphi 1$. This is because the Lyapunov certificate criterion of Th. 2.3 is a sufficiency condition, and it is possible that we get feasible certificates for an even higher degree d parametrization of these certificates.

3.2.2 Deductive-Bounded Verification of $\varphi 2$

To verify $\varphi 2$, we need to show that all hybrid arcs x, starting in set $\mathcal{S}2$, eventually reach the attractive invariant set $\mathcal{S}1$. Towards this goal, we use a mixed deductive-bounded verification approach benefiting from the advection of sets and certificate based deductive verification. Essentially, we use advection of sets for HDS, presented in Sec. 3.1.3, and check whether the advected sets fully submerge in to the set $\mathcal{S}1$

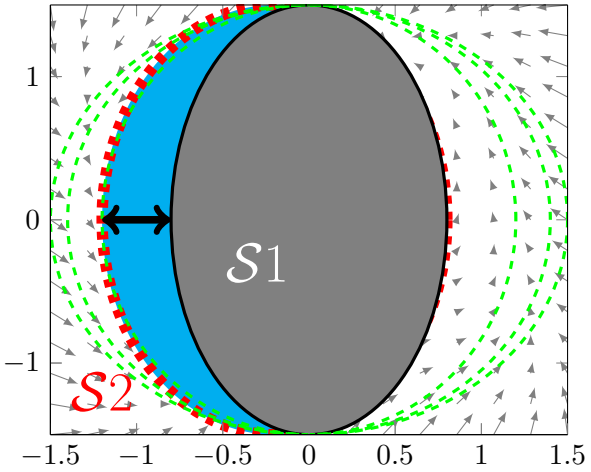


Figure 3.6: *Deductive-Bounded Verification Methodology*

after bounded iterations, as depicted in Fig. 3.6. After a bounded number of advection steps, for a set which is not a proper subset of set $\mathcal{S}1$, we apply the deductive Escape certificate criterion showing trajectories starting in this set will eventually escape and reach $\mathcal{S}1$. This happens when the advection of sets is asymmetrical and submerge in to set $\mathcal{S}1$ from one direction while its progression from another side is very slow. For example, in fourth order CP PLL, we notice that the set advection is inconclusive as sets do not submerge fully in to $\mathcal{S}1$. This scenario is illustrated by the level set of the red dotted curve in Fig. 3.6. We notice that on the right side of set $\mathcal{S}1$, the advected set enclosed by the red curve is fully immersed in $\mathcal{S}1$, its progression however from the left hand side (shown by the double-sided arrow) is stopped after bounded advection steps. This shows that trajectories starting in the left part of the set jump to the right hand side before reaching the set $\mathcal{S}1$. For this part of the set, bounded advection is inconclusive, and we use the deductive Escape certificate criterion.

Following the deductive-bounded approach illustrated above, we verify the property $\varphi 2$ using Alg. 2. The inputs of the algorithm are the sets $\mathcal{S}2$, $\mathcal{S}1$, and the HDS model of the CP PLL. The algorithm determines the truth value of $\varphi 2$ by a combination of deductive Escape certificate and bounded advection of set $\mathcal{S}2$. After initializing different sets, the advection of set $\mathcal{S}2_{advect}$ is performed in Line 5. Following the advection of level sets for HDS, discussed in Sec. 3.1.3, the function "Advect" in Line 5 is performed

Algorithm 2 Verification of Property $\varphi 2$

INPUT: : HDS Model of CP PLL, Sets $\mathcal{S}1,\ \mathcal{S}2$
OUTPUT: : $\varphi 2$ Verified in Bounded Time/No-answer

$\mathcal{S}2_{next} \leftarrow \emptyset$
$\mathcal{S}2_{advect} \leftarrow \emptyset$
$\mathcal{S}2_{advect} \leftarrow \mathcal{S}2$
for $j \leftarrow 1$ **to** $j \leftarrow m$ **do**
  $\mathcal{S}2_{next} \leftarrow \text{Advect}(\mathcal{S}2_{advect})$
  if $\mathcal{S}2_{next} \not\subset \mathcal{S}1$ **then**
    $\mathcal{S}2_{advect} \leftarrow \mathcal{S}2_{next}$
  else
    $x \models \varphi 2,\ \forall x \in \mathcal{S}2$
    break
  end if
end for
Try a large value of m
if $\mathcal{S}2_{next} \not\subset \mathcal{S}1$ **then**
  For $\mathcal{S}2_{next} \setminus (\mathcal{S}2'_{next} = \mathcal{S}1 \cap \mathcal{S}2_{next})$ find the Escape Certificate E.
  if E exists **then**
    $x \models \varphi 2,\ \forall x \in \mathcal{S}2$
    break
  else
    No Answer about $\varphi 2$
  end if
end if

by the following SOS program.

$$
\begin{aligned}
&\text{minimize } \eta_i \\
&\text{s.t.} \quad P_{next}(0) < 0, \\
&\frac{\partial P_{next}}{\partial x}.(v_1,\ v_2,\ \phi_D)^T > 0, \\
&s1_i - s2_i P_{Initial} + B^i_{-h} P_{next} + \eta_i \ + \sum_{k=1}^{m_{C_i}} s3_{ik} g_{ik} + \sum_{j=1}^{m} s4_j(x) a_j(u) = 0, \\
&s5_i + s6_i(P_{Initial} - \mu_i) - B^i_{-h} P_{next} + \eta_i \ + \sum_{k=1}^{m_{C_i}} s7_{ik} g_{ik} + \sum_{j=1}^{m} s8_j(x) a_j(u) = 0, \\
&s9_i - s10_i(P_{Initial} - \mu_i) + \sum_{k=1}^{m_{C_i}} s11_{ik} g_{ik} \ + \frac{\partial^2 P_{next}}{\partial x^2}\frac{h^2}{2} - \eta_i = 0, \\
&s12_i - s13_i(P_{Initial} - \mu_i) + \sum_{k=1}^{m_{C_i}} s14_{ik} g_{ik} \ - \frac{\partial^2 P_{next}}{\partial x^2}\frac{h^2}{2} - \eta_i = 0, \\
&(s1_i,\ s2_i,\ s3_{ik},\ s4_j,\ s5_i,\ s6_i,\ s7_{ik},\ s8_j, s9_i,\ s10_i,\ s11_{ik},\ s12_i,\ s13_i,\ s14_{ik}) \ \in \mathcal{S}_n.
\end{aligned}
\tag{3.33}
$$

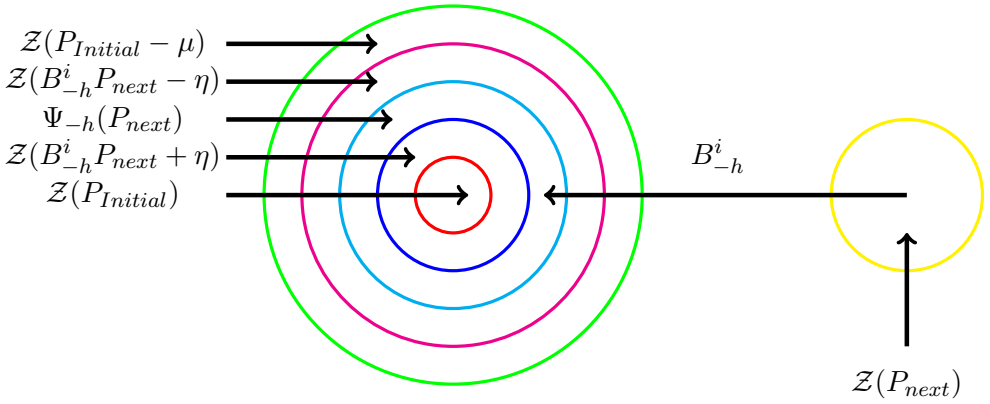


Figure 3.7: *Deductive-Bounded Verification Methodology*

Here P_{next} is of degree d_r, $\mu_i > 0$, $\eta_i > 0$, $h > 0$, $u \in [L\ U]$, and $s1_i$, $s2_i$, $s3_{ik}$, $s4_j$, $s5_i$, $s6_i$, $s7_{ik}$, $s8_j$, $s9_i$, $s10_i$, $s11_{ik}$, $s12_i$, $s13_i$, $s14_{ik}$, are polynomials of degree d.

Let $\mathcal{S}2 = \mathcal{Z}(P_{Initial})$, $\mathcal{S}1 = \mathcal{Z}(P1)$ and $\mathcal{S}2_{next} = \mathcal{Z}(P_{next})$. Here, $P_{Initial}$, $P1$, and P_{next} are differentiable polynomials belonging to the set $C(\mathbb{R}^n, \mathbb{R})$. Similar to the SOS program for Lyapunov certificates, the SOS program for the advection of sets utilizes the S-procedure discussed in Sec. 2.4.4. The first two constraints of this SOS program ensure the advected level sets are closed and connected (see [102] and the references therein). The next two constraints search for a polynomial P_{next}, such that when the set $\mathcal{Z}(P_{next})$ is backward advected by the first order Taylor advection map B^i_{-h}, we obtain a set such that,

$$\begin{aligned}\mathcal{Z}(P_{Initial}) \subset \mathcal{Z}(B^i_{-h}P_{next} + \eta_i) \subset \Psi_{-h}(\mathcal{Z}(P_{next})) \subset \mathcal{Z}(B^i_{-h}P_{next} - \eta_i) \\ \subset \mathcal{Z}(P_{Initial} - \mu_i)\end{aligned} \tag{3.34}$$

Here μ_i is used as a precision parameter determining how closely we want the set $\mathcal{Z}(P_{Initial})$ to be approximated by the set $\mathcal{Z}(B^i_{-h}P_{next} + \|\eta_i\|)$. Constraints for C_i have been added by using SOS multipliers $s3_{ik}$, $s7_{ik}$ and the vector inequality $g_{ik}(x) \leq 0$. Furthermore, parameter constraints are added by using SOS multipliers $s4_j$ and $s8_j$ with the vector inequality $a_j(u) \leq 0$. This advection of zero sub-level sets is illustrated in Fig. 3.7. The last two constraints enforce the truncation error of the first order Taylor approximation such that, $\|\frac{\partial^2 P_{next}}{\partial x^2}\frac{h^2}{2}\| \leq \eta_i$, for all x in the set $\mathcal{Z}(P_{Initial} - \mu)$.

The next step in Alg. 2 is checking the intersection of sets $\mathcal{S}2_{next}$ and $\mathcal{S}1$ Line-6. To be conservative, and use an over-approximation to the set $\Psi_h(\mathcal{Z}(P_{Initial}))$, the set membership is encoded as a SOS program utilizing Lemma 2.2 for the sets $\mathcal{Z}(P_{next}-\eta_i)$ and $\mathcal{S}1$, i.e.,

$$s0 - s1(P_{next} - \eta_i) + P1 = 0,\ \ s0, s1 \in \mathcal{S}_n \tag{3.35}$$

If there are feasible SOS polynomials $s1,$ and $s2,$ then, $\mathcal{Z}(P_{next} - \eta_i) \subset \mathcal{Z}(P1)$.

Remark 3.2. *For the transformed CP PLL HDS, $\mathcal{H}^{'}$, we have identity jump maps, there is therefore no need for constraints on the level sets due to discrete jumps.*

After each iteration of the advection of level sets, if the set inclusion $\mathcal{S}2_{next} \subset \mathcal{S}1$ is true, then property $\varphi 2$ is verified. Alternatively, the algorithm keeps on advecting the set $\mathcal{S}2_{next}$ for a user defined bounded number of iterations (Line 7-13). If the property $\varphi 2$ is still not verified (this can happen when the advection of the level sets is asymmetrical and a subset of the set $\mathcal{S}2_{next}$ is not fully immersed in $\mathcal{S}1$), we compute the Escape certificate E for the set, $\mathcal{S}2_{next} \setminus \mathcal{S}2^{'}_{next} (= \mathcal{S}1 \cap \mathcal{S}2_{next})$. A feasible Escape certificate in the set $\mathcal{S}2_{next} \setminus \mathcal{S}2^{'}_{next}$ shows that trajectories in this set will eventually leave and reach $\mathcal{S}1$ by Prop. 3.3 (Line 14-18). This either results in the verification of property $\varphi 2$ (respectively φ) in set $\mathcal{S}2$, or we conclude inconclusiveness about the truth value of $\varphi 2$ (respectively φ). In Line 15, the Escape certificate is searched by the following SOS program,

$$\begin{aligned} -\frac{\partial E_i}{\partial x}(x)F_i(x,u) - s1(x)g2(x) + s2(x)g2^{'}(x) - \sum_{j=1}^{m} s3_j(x)a_j(u) - \varepsilon \in \mathcal{S}_n, \\ (s1, s2, s3_j) \in \mathcal{S}_n. \end{aligned} \tag{3.36}$$

where, $\mathcal{S}2_{next} := g2(x) \geq 0$, and $\mathcal{S}2^{'}_{next} := g2^{'}(x) \geq 0$.

Proposition 3.6. *If the SOS program of Eq. 3.36 is feasible, then the Escape certificates satisfy the condition in Prop. 3.2.*

Proof. The expression in Eq. 3.36 being SOS is therefore,

$$-\frac{\partial E_i}{\partial x}(x)F_i(x,u) - s1(x)g2(x) + s2(x)g2^{'}(x) - \sum_{j=1}^{m} s3_j(x)a_j(u) - \varepsilon \geq 0$$

Multiplier $s1,\ s2,\ s3_j$ all being SOS, and $g2(x) \geq 0$, $g2^{'}(x) \geq 0$, $a_j(u) \geq 0$, therefore

every product term is positive semi-definite. Therefore,

$$\frac{\partial E_i}{\partial x}(x)F_i(x,u) \leq -\varepsilon.$$

□

If there is a feasible Escape certificate for the set $\mathcal{S}2_{next} \setminus \mathcal{S}2'_{next}$, then we conclude $x \models \varphi 2,\ \forall x : x \in \mathcal{S}2$ Line 17. In case we do not find an Escape certificate of some maximum bounded degree, we declare inconclusiveness about the truth value of $\varphi 2$ Line 20.

3.3 Deductive Verification of Inevitability in CP PLL

In this section, we discuss the certificate based deductive-only verification methodology for the verification of the CP PLL inevitability property. Principally, this methodology is similar to the deductive-bounded verification in that it too uses the divide and rule strategy to verify the complex inevitability property. The difference is that it is purely a certificate based deductive approach and does not use bounded verification. Similar to the mixed deductive-bounded approach, we introduce two compact sets, $\mathcal{S}1$ and $\mathcal{S}2$, such that $\mathcal{S}1 \cap \mathcal{S}2 = \emptyset$, and $\mathcal{S}1 \cup \mathcal{S}2 = \mathcal{C} \cup \mathcal{D}$. We define two properties $\varphi 1$ and $\varphi 2$ in sets $\mathcal{S}1$ and $\mathcal{S}2$ respectively. In this methodology, we use a certificate based deductive approach for the verification of both $\varphi 1$ and $\varphi 2$ as shown in Fig. 3.8. Notice that in both sets $\mathcal{S}1$ and $\mathcal{S}2$, we use the deductive-only approach to verify $\varphi 1$ and $\varphi 2$ respectively. Here we use an approach, similar to that of the deductive-bounded methodology, using Lyapunov certificates to verify property $\varphi 1$. We further show the attractive invariance of $\mathcal{S}1$ from the level curves of the Lyapunov certificates. The difference here is the way we verify property $\varphi 2$ using only the Escape certificates. Therefore, in this section, we only discuss the verification of $\varphi 2$.

Theorem 3.2. *If in a compact set $\mathcal{S}2$, such that $\mathcal{S}1 \cup \mathcal{S}2 = \mathcal{X}_{\mathcal{H}'}$, where $\mathcal{S}1$ is an attractive invariant set, we have Escape certificates, $E_i(x),\ \forall i \in \{1,..,\ell\},\ \forall x : x \in \mathcal{S}2$, then, $\forall x(t,j) : x(t,j) \in \mathcal{S}2$, $\lim_{t+j\to\infty} x(t,j) \in \mathcal{S}1$.*

Proof. Follows directly from Lemma. 3.3. The boundedness of $x(t,j)$ is guaranteed by the supply voltage and ground of the CP PLL circuit. Existence of an Escape certificate for $x(t,j) \in \mathcal{S}2$ (Prop. 3.2), guarantees that trajectories will eventually leave $\mathcal{S}2$, and being the only invariant set, they will eventually reach $\mathcal{S}1$. □

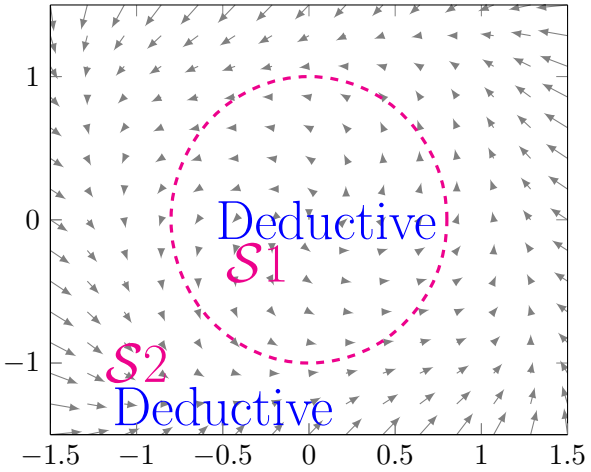


Figure 3.8: *Deductive-Only Verification Methodology*

Following Th. 3.2, we verify property $\varphi 2$ utilizing Alg. 3. We search for ℓ Escape certificates (Prop. 3.2) in ℓ disjoint sets $\mathcal{S}2_i,\ i \in \{1,..,\ell\}$, such that $\mathcal{S}2 = \cup_{i\in 1,..,\ell}\mathcal{S}2_i$. The inputs of Alg. 3 are the set $\mathcal{S}2$ and the CP PLL HDS. After parametrizing these ℓ Escape certificates, by setting their degrees d and coefficients, we establish the feasibility of these Escape certificates by the following SOS program (Line 4),

$$-\frac{\partial E_i}{\partial x}(x)F_i(x,u) - \sum_{k=1}^{n_{C_i}} s_1^{(ik)}(x)g2_{ik}(x) - \sum_{j=1}^{m} s_2^{(j)}(x)a_j(u) - \varepsilon \in \mathcal{S}_n, \qquad s_1^{(ik)}, s_2^{(j)} \in \mathcal{S}_n,\ \varepsilon > 0 \tag{3.37}$$

Note that d is initially set up to be 2 and is incremented by 2 in each iteration. This SOS program ensures that Lie-derivatives of E_i are strictly negative in sets, $\mathcal{S}2_i = \{x \in \mathbb{R}^n\ : g2_{ik} \geq 0,\ \text{for } k \in \{1,..,n_{C_i}\},\ i \in \{1,..,\ell\}\}$. The second constraint in this SOS program is such that the parameters u belong to the set, $\{a(u) \geq 0,\ \text{for } j \in \{1,..,m\}\}$. Here ε is a small positive real number. Feasibility of the SOS program in Eq. 3.37 indicates existence of the Escape certificates for each mode of the CP PLL HDS, and consequently the property $\varphi 2$ is verified, Line(4-6). Alternatively, if the SOS program in Eq. 3.37 is infeasible, we increase the degree d of each Escape certificate by 2 and repeat the process, Line(8-10). If the property $\varphi 2$ is still not verified for a maximum user defined degree b, we conclude inconclusiveness about the truth value of $\varphi 2$ (respectively φ), Line 13.

Algorithm 3 Verification of Property $\varphi 2$

INPUT: : Hybrid System Model of CP PLL, Set $\mathcal{S}2 = \cup_{i \in 1,..,\ell} \mathcal{S}2_i$
OUTPUT: : $\varphi 2$ Verified/No-answer

1: **for** $i \leftarrow 1$ **to** $i \leftarrow \ell$ **do** ; Here ℓ is the no. of discrete modes of the HDS
2: | $E_i \leftarrow Parametrize(E_i)$; Setting degree d and coefficients of E_i Polynomials
3: **end for**
4: **if** $E_i, \forall i \in \{1, .., \ell\}$, are feasible (fulfilling Prop. 3.2) **then**
5: | $E_{multiple} \leftarrow \{E_{multiple}, E_i\}, \ \forall i \in \{1, .., \ell\}$
6: | $x \models \varphi 2, \ \forall x \in \mathcal{S}2 = \cup_{i \in 1,..,\ell} \mathcal{S}2_i$
7: **else**
8: | $E_i \leftarrow Infeasible$
9: | Increase Degree d of E_i's ; d is incremented by 2
10: | Goto Line(1) if d of $E_i < b \ \forall i \in \{1, .., \ell\}$; Here b is a user-defined upper bound on degree d
11: **end if**
12: **if** d = b & $x \not\models \varphi 2, \ \forall x \in \mathcal{S}2$ **then**
13: | No Answer about $\varphi 2$
14: **end if**
15: **return** Truth value of $\varphi 2$

Parameters	Third Order	Fourth Order
C_1	$[1.98\ 2.2]e-12F$	$[31\ 29]e-12F$
C_2	$[6.1\ 6.4]e-12F$	$[3.2\ 3.4]e-12F$
C_3		$[1.8\ 2.2]e-12F$
R	$[7.8\ 8.2]e3\Omega$	$[48\ 52]e3\Omega$
$R2$		$[7\ 9]e3\Omega$
f_{ref}	27MHZ	5MHZ
f_O	27e3MHZ	5MHZ
I_p	[495 505]e-6A	[395 405]e-6A
K_p	[198 202]	[495 502]

Table 3.1: *PLL Parameters used in the Experimentation*

3.4 Experimental Evaluation

We have verified the inevitability of phase locking for a third and fourth order CP PLL. The CP PLL parameters we have used are listed in Table 3.1 ([7]), with all phases normalized by 2π. For all experiments, we used the YALMIP [67] and SeDuMi [88] solvers within MATLAB for the verification of the inevitability property (respectively sub-properties) on a 2.6 GHZ Intel Core i5 machine with 4 GB of memory.

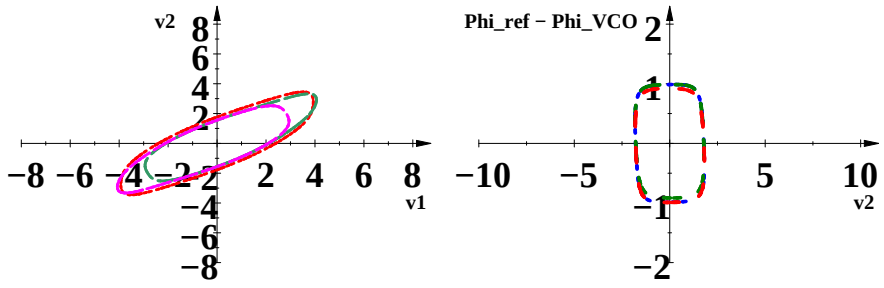


Figure 3.9: *3-Order* $\mathcal{S}1$ *Projected onto* $(v1, v2)$, *and* $(v2, \phi_D)$

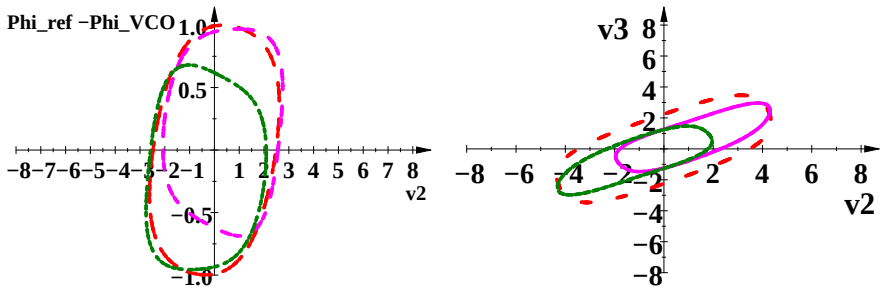


Figure 3.10: *4-Order* $\mathcal{S}1$ *Projected onto* $(v2, v3)$, *and* $(v2, \phi_D)$

3.4.1 Mixed Deductive-Bounded Verification Methodology

In this section results of the inevitability verification, using the mixed deductive-bounded approach, are presented. For a third order CP PLL, we constructed degree-6 multiple Lyapunov certificates while verifying property $\varphi 1$, Appendix B.2.1. Similarly, we found degree-4 multiple Lyapunov certificates for the fourth order CP PLL verifying sub-property $\varphi 1$, Appendix B.2.2. The corresponding attractive invariant sets $\mathcal{S}1$ generated by these Lyapunov certificates are depicted in Fig. 3.9, and Fig. 3.10 for the third and fourth order CP PLL respectively. Note that only projections of the set $\mathcal{S}1$ on different planes have been shown in Fig. 3.9 and Fig. 3.10. This was followed by the verification of the sub-property $\varphi 2$ using advection of sets. This was performed by computing advection of sets using sets $\mathcal{S}1$ as target sets. We considered a circular starting set $\mathcal{S}2$ around the set $\mathcal{S}1$ for both benchmarks. The corresponding results of the advection of sets are shown in Fig. 3.11 and Fig. 3.12 for third and fourth order CP PLL respectively. Note that due to space constraints, we have shown projections on only two planes for each benchmark. The outer set plotted in solid is the initial set inside which we aim to prove the inevitability of the phase-locking in the CP PLL. The

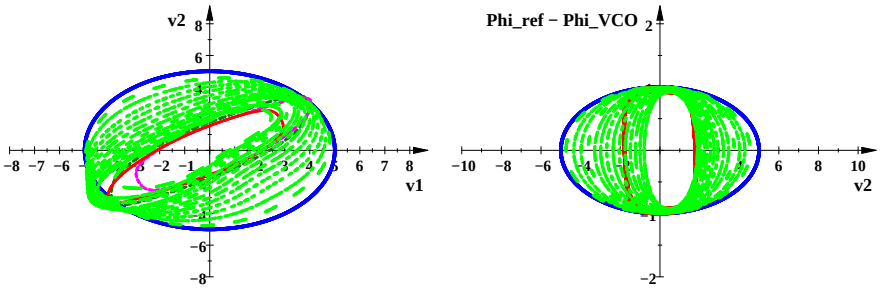


Figure 3.11: *3-Order Advection Projected onto* $(v1, v2)$, *and* $(v2, \phi_D)$

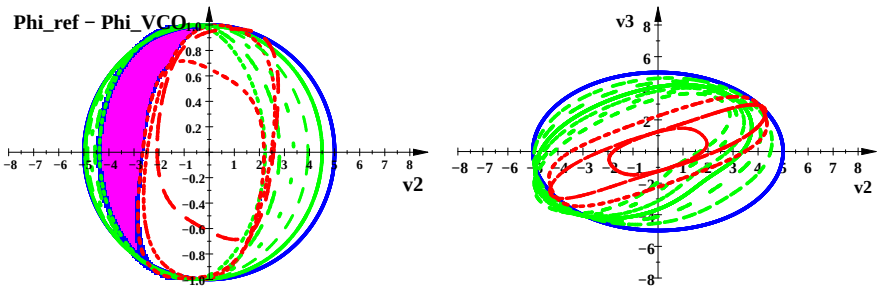


Figure 3.12: *4-Order Advection Projected onto* $(v2, v3)$, *and* $(v2, \phi_D)$

advected level curves are shown in dotted. We used the time step $h = 1e-3$ seconds, and $\mu_i = 1e-4$ in the computation of advected sets. It can be observed that for the third order, the advected zero sub-level sets were eventually symmetrically immersed in the central attractive invariant set $\mathcal{S}1$ after bounded number of iterations. However, for the fourth order CP PLL, the advection of zero sub-level sets is unsymmetrical and the progress in one direction is more abrupt than the other. We therefore have the level sets immersed in the attractive invariant set $\mathcal{S}1$ from one direction, but the advection is inconclusive for a subset in the other direction shown by the pink shaded area in Fig. 3.12. For the inconclusive subset, we searched degree-4 Escape certificates for mode-1 and mode-3 and a degree-2 Escape certificate for mode-2, showing convergence of the trajectories to the attractive invariant set $\mathcal{S}1$ (Appendix B.3.2). The corresponding computation times for different steps of our verification methodology are given in Table 3.2.

Verification Step	3-Order Time(Sec)	4-Order Time(Sec)
Attractive Invariant	1381.7(Degree 6)	1002.1(Degree 4)
Max.Level Curves	15.5	12
Advection	106.8487 (14 iterations)	140.678 (7 Iterations)
Checking Set Inclusion	13	10.2
Escape Certificate		21.6 (3 Certificates)

Table 3.2: *Computation Time of the Inevitability Verification*

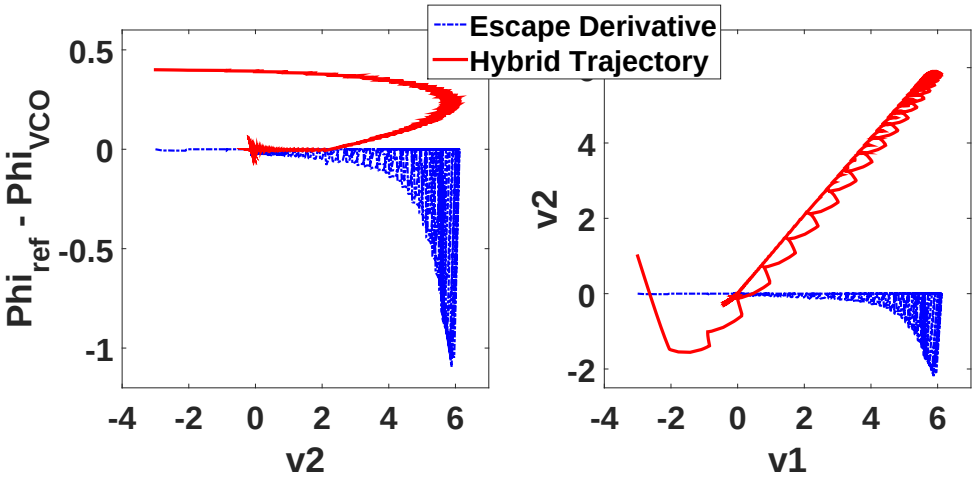


Figure 3.13: *3-Order Derivative of Escape Certificates, Trajectory Trace, Projected onto* $(v1, v2)$, *and* $(v2, \phi_D)$

3.4.2 Deductive-only Verification Methodology

In this section, we present the results of our deductive-only verification methodology. Since the verification of the sub-property $\varphi 1$ is similar to that of the deductive-bounded approach, we followed the same procedure and computed degree-6 multiple Lyapunov certificates for the third order, and degree-4 multiple Lyapunov certificates for the fourth order CP PLL respectively. Their attractive invariant sets $\mathcal{S}1$ as projected onto different planes are shown in Fig. 3.9 and Fig. 3.10 respectively. This was followed by the verification of the sub-property $\varphi 2$ using deductive Escape certificates. We constructed three Escape certificates for each mode of the third order and fourth order CP PLL HDS models, Appendix B.3.1, Appendix B.3.2. For both benchmarks, we computed degree-2 Escape certificates for mode2 and mode3, whereas for mode1, we computed degree-12 and degree-10 Escape certificates for third and fourth order CP PLL respectively. We chose $\varepsilon = 1e - 4$ for the construction of all Escape certificates. We noticed that

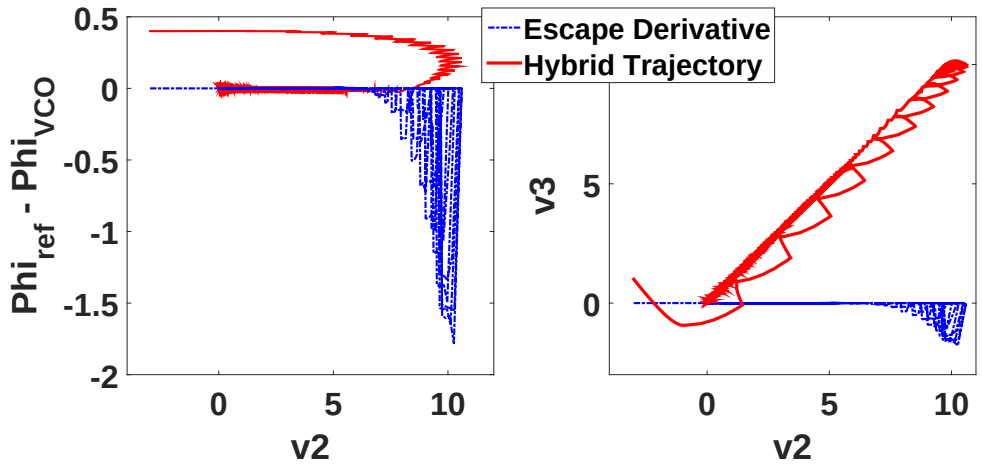


Figure 3.14: *4-Order Derivative of Escape Certificates, Trajectory Trace, Projected onto* $(v2, v3)$, *and* $(v2, \phi_D)$

Verification Step	3-Order Time(Sec)	4-Order Time(Sec)
Attractive Invariants	1381.7(Degree 6)	1002.1(Degree 4)
Max.Level Curves	15.5	12
Escape Certificates	100	900

Table 3.3: *Computation Time of the Inevitability Verification*

decreasing the value of ε resulted in higher degree Escape certificates being needed for both benchmarks. However, this is at the cost of higher computation time. We therefore opted for the value $1e-4$. A simulation trace along with the derivative of the Escape certificate patched up from the three Escape certificates of each benchmark are depicted in Fig. 3.13 and Fig. 3.14 respectively. Note that due to space constraints, we have shown projections on only two planes for each benchmark. These two figures show the value of the derivative of the Escape certificates in blue for the trajectories of the CP PLL shown in red. Simulation traces show that the derivative of the Escape certificates is negative for the entire duration of the two trajectories while they reach the zero locking state (equilibrium), which is the required condition on these Escape certificates. Computation time for different steps of our verification methodology is given in Table 3.3. Though the maximum degree of certificates (Lyapunov,Escape) for the fourth order is less than that of the third order, the dimensionality factor is however dominant as far as the computation time is concerned.

3.5 Related Work

Lyapunov theory has been used for an analog PLL design in [1]. The author has used LaSalle's theorem, which is the generalization of the Lyapunov theorem, and designed parameters for a purely "analog" PLL. Our work is in many ways different from what has been done in [1]. While [1] considered an approximate non-linear continuous model with a "sin" non-linearity in the loop, we consider a CP PLL which is naturally a HDS due its discrete and continuous behaviour. Secondly, the problem tackled in [1] is a design problem, where an analog PLL has been designed such that it is almost globally stable. On the other side, we have worked on an inevitability property which is theoretically completely different from the stability property. Lastly, the construction of the Lyapunov function has been done analytically in [1] as oppose to our SOS based algorithmic approach. While Lyapunov certificate can be found analytically for such simple PLL models, as that in [1], it is generally impossible to be found, analytically, for CDS/HDS like CP PLL. In [66], the authors conservatively converted a digitally extensive PLL into a continuous model using a machine learning technique. They divided the state space into linear and non-linear regions and used hybrid reachability for linear and SMT based reachability for non-linear regions respectively. [105] presented a similar technique to an all digital PLL, where they approximated the behaviour of the PLL with a continuous time piecewise linear hybrid automata to which the quantization effects were added as uncertain parameters. They also divided the state space into linear and non-linear regions, and applied linear Lyapunov stability theory (using Quadratic Lyapunov certificates) for linear and reachability analysis for non-linear regions respectively. In order to reduce the complexity of the model, they neglected a few sub-systems of the PLL circuit. As these neglected sub-systems have a substantial effect on the stability of the overall system, the accuracy of their technique was greatly reduced. [105] also used an SMT solver and computed coefficients of a quadratic Lyapunov certificate without considering the quantifiers alternation we discussed earlier. The approach seems to guess the coefficients of the quadratic Lyapunov certificate, and then check the feasibility of Lyapunov stability constraints over the region of interest using universal quantification. This approach clearly lacks the automation aspect of computing Lyapunov certificates. Though for linear stable systems the existence of a quadratic Lyapunov certificate is a sufficient and necessary condition, for general stable hybrid systems such certificates do not exist and higher order Lyapunov certificates are required. Furthermore, quadratic Lyapunov stability certificates are very conservative and need further splitting of the state space to reduce conservativeness. The authors

in [105], also reported time-outs of the reachability tool SpaceEx [39] while computing reachable sets starting from some remote regions. This is due to the large number of mode transitions needed in PLL systems before trajectories reach the equilibrium state. To avoid discrete jumps, [7] presented a continuization technique and verified the 'time to locking' property of a CP PLL. They have used reachability to verify the time to locking property in a third order CP PLL.

3.6 Summary of the Chapter

In this chapter, we have proposed scalable and computationally tractable methodologies for the verification of an important yet complex inevitability property of a CP PLL. We have come up with two methodologies benefiting from both deductive and bounded verification paradigms. In Sec. 3.1.1, we have given a comprehensive HDS modelling of the higher order CP PLL covering its hybrid discrete and continuous behaviour. We have divided the verification task in to two sub-properties in Sec. 3.2 and proposed deductive and deductive-bounded verification approaches for their verification in Sec. 3.2.1 and Sec. 3.2.2 respectively. We used Lyapunov certificates for the verification of one sub-property in Sec. 3.2.1, and used advection of level sets and an Escape certificate for the verification of the second property in Sec. 3.2.2. A deductive-only verification for both sub-properties was presented in Sec. 3.3. Results show the effectiveness of our approach to the verification of the inevitability property of a complex real circuit. We have proved the inevitability property avoiding hundreds of discrete transitions as well as the complex continuization as in [7]. Computation time is comparable to [7], and in fact is less by an order of at least half considering their approach using gridding of the state space for a third order PLL only. Though user input is needed in the formalization of the problem, our Lyapunov and Escape certificate based deductive methods are applicable to infinite domain (as oppose to bounded) and avoid approximating (under or over) solutions of the differential equations. Furthermore, our bounded advection of level sets has the advantage of dealing with larger sets in a single iteration as compared to the existing bounded model checking approaches. Comparing the two techniques, computation time of the mixed deductive-bounded approach is clearly less than the deductive-only approach. However, considering the number of iterations in the advection of level sets, in the deductive-bounded approach, more user input is required as compare to the deductive-only approach. Furthermore, the computation time of the deductive-only approach can be reduced by dividing the outer set into several subsets, and compute lower degree Escape certificates for these subsets. The whole

process of the certificate computation can further be automated by delegating the task of parametrization and other initializations to a software program with the additional call to a semi-definite program solver.

Chapter 4

Deductive Inevitability Verification of Ring Oscillators using the SOS-QE Approach

Ring oscillators are an integral part of most modern SOC designs. They are used for various purposes — from reference clock generation, data clock recovery to phase modulation etc. They are designed hoping that they will start from all possible voltage conditions on their nodes. Unfortunately, it is practically impossible for an RO to have global start-up property and will start from every possible state of voltages on its nodes. In [56], researchers at Rambus identified start up failure in an even stage RO for a subset of initial conditions and parameters. Recently, several works have been dedicated to the verification of the start-up property mainly based on reachability analysis. These approaches are faced with several issues. Reachability verifies the property for bounded time, and nothing can be established about the behaviour of an RO over the infinite horizon. Secondly, reachability tools rely on over-approximating solutions of the differential equations describing an RO circuit, and are thus subjected to erroneous results. Furthermore, to reduce conservatism, a large number of discrete partitions of the state space is performed, resulting in increased computational complexity.

A periodic set of states is said to be almost globally inevitable, if an RO eventually reaches this set, from all but a negligible dead set of voltages on its nodes. In this chapter, we propose a deductive verification methodology and verify the almost global inevitability of oscillations in ROs. We consider two different topologies of ROs, namely, odd stage and even stage ROs. Due to its layout, the stages of an even RO

operate in differential pairs. This allows division of its operation into differential and common modes. Since oscillations are manifested in the differential mode, we verify the inevitability property for this particular mode of the even RO. Furthermore, we show that its common mode settles to around zero voltage. We treat the odd stage RO similar to the differential mode of even RO. Dividing the even stage RO into differential and common modes reduces the dimensionality of the system by an order of half. Verifying inevitability, we adopt a divide and rule approach and split it into the conjunction of various sub-properties. Verification of these sub-properties determines the truth value of the inevitability property in two disjoint subsets of the state space. We use certificate based deductive verification to verify all these sub-properties. We formulate the construction of these certificates as FOFs having Universal-Existential quantifiers over polynomial inequalities/equalities. Due to the high computational cost of QE in real algebraic theory[1], we present a SOS-QE approach to verify truth values of these FOFs. SOS programming solves these quantified formulas using semi-definite programming in a realistic computational time (utilizing a numerical interior point method), within the limits of numerical precision. To overcome these numerical imprecisions and establish the validity of these certificates, we further verify these certificates (having a fixed structure now) using the symbolic QE approach. This is done by verifying FOFs having only the universal quantifiers.

4.1 Preliminaries

This section discusses the mathematical modelling of RO and the necessary background for the certificate based deductive verification of the inevitability.

4.1.1 Modelling of the Ring Oscillator

We model an RO shown in Fig. 4.1 as a polynomial CDS discussed in Sec. 2.1.1. Let us denote by x the vector of node voltages at inverter outputs. Therefore, the CDS model of an RO is a tuple $(\mathbf{X},\ \mathcal{X}_{initial},\ \mathbf{U},\ f)$. Note that since there are no inputs in ROs we drop the set of inputs $\mathbf{W}$. The vector field f characterising an RO is given by,

$$\dot{x} = f(x,u),\ f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n,\ x \in \mathcal{X},\ u \in \mathcal{U} \tag{4.1}$$

[1] works only for low dimension low complexity problems and is doubly exponential in number of variables

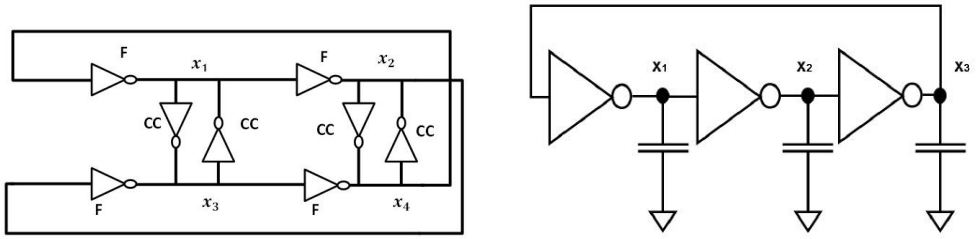


Figure 4.1: *Ring Oscillators, Left: Even Stage, Right: Odd Stage*

where $\mathcal{X} \subset \mathbb{R}^n$ is the state space of the RO. This set is invariant as trajectories of an RO never leave this set. Let us denote by $\Phi(x_0, t),\ x_0 \in \mathcal{X}_{init}$, the solution vector of the system of differential equations Eq. 4.1 (also called trajectory in some literature).

Definition 4.1 (Equilibrium state)**.** *A state $x_e \in \mathcal{X}$ is called an equilibrium of the CDS model of an RO, if $f(x_e) = 0$.*

Definition 4.2 (Limit Cycle)**.** *A set $\gamma \subset \mathcal{X}$ is called a Limit cycle, if $\forall x_0 : x_0 \in \gamma$, $\Phi(x_0, T) = x_0$, for $T > 0$, and $\forall t : 0 < t < T$, $\Phi(x_0, t) \neq x_0$. This is an invariant set.*

Definition 4.3 (Inevitability of the Limit cycle)**.** *The Limit cycle γ is said to be inevitable if, $\forall x_0 : x_0 \in \mathcal{X}_{initial},\ y \in \gamma,\ r > 0,\ b \in \mathbb{R}_{\geq 0}$,*

$$\lim_{t \to b} \|\Phi(x_0, t) - y\| \leq r \tag{4.2}$$

Assumption 4.1. *In this work, we assume that location of γ in the state space $\mathcal{X}$ is known.*

ROs are designed so that on power up, they start to oscillate with the desired frequency from all possible voltages on their nodes. In other words, considering an RO as a CDS, its limit cycle γ is globally inevitable. However, in a practical RO, there are states in $\mathbb{R}^n$ from where it fails to start and reach the limit cycle γ. For example, equilibrium is one such state from where an RO can not start. We call the set of all such states the "dead set".

Definition 4.4 (Dead Set)**.** *A set of states is called a dead set denoted by $\mathcal{X}_{dead}$, such that $\forall x : x \in \mathcal{X}_{dead},\ \lim_{t \to \infty} \|\Phi(x, t) - x_e\| = 0$. Here x_e is an equilibrium state.*

Since the dead set, though of a lower dimension, is unavoidable in an RO state space, we modify the definition of global inevitability and introduce the notion of "almost global inevitability".

Definition 4.5 (Almost Global Inevitability (AGI) of Oscillation in ROs)**.** *The Limit cycle $\gamma \subset \mathcal{X}$, is said to be "AGI", if, $\forall x_0 : x_0 \in \mathcal{X} \setminus \mathcal{X}_{dead},\ y \in \gamma,\ r > 0,\ b \in \mathbb{R}_{\geq 0}$,*

$$\lim_{t \to b} \|\Phi(x_0, t) - y\| \leq r \tag{4.3}$$

4.1.1.1 An Inverter Model

While modelling an RO as CDS the main problem is how to derive the expression for the vector field $f(x, u)$ in Eq. 4.1. Apart from parameters, the vector field $f(x, u)$ is a function of node voltages on capacitors at the output of each inverter. These voltages, if modelled at the transistor level, are non-linear functions of the currents through the transistors. To understand this, we have shown in Fig. 4.2, a CMOS inverter with a capacitor at the output and the internal details of CMOS transistors. We can see that an inverter consists of two MOS transistors, a PMOS and an NMOS with a capacitor at the output. A current model at the transistor level is seemingly very complex due to the non-linear behaviour of a transistor covering three (Linear, Saturation, Cut off) regions of operation. One option is to model this current as a piecewise polynomial function, resulting in a HDS model for the inverter. This will require nine modes in the HDS for a single inverter. Therefore, to reduce the complexity of the model and make it amenable to formal verification, we avoid modelling the inverter at the transistor level. This is in accordance with the top-down modelling strategy we discussed in Sec. 2.2. However, while considering an abstract model for the inverter, we still need to take in to account the non-linearity inherited by the inverter due to its constituent transistors. Therefore, we use an inverter model based on a"$tanh(.)$" non-linearity presented in [40][34]. This non-linear model of the inverter has a "$tanh(.)$" non-linearity connected with a low pass filter as shown by its transfer function $G(s)$ in Fig. 4.2(d). If V_{in} is the input to the "$tanh(.)$" non-linearity block and V_n its output, then we have the non-linear "$tanh(.)$" block represented by the following equation,

$$V_n = V_{sat} \times tanh(V_{in}/V_s) \tag{4.4}$$

Here V_{sat} is the saturation voltage and V_s is a parameter adjusting the slope of the inverter output response. Both these parameters are very important and can be used to have the effect of transistor level parameter variations. For example, [29] has shown

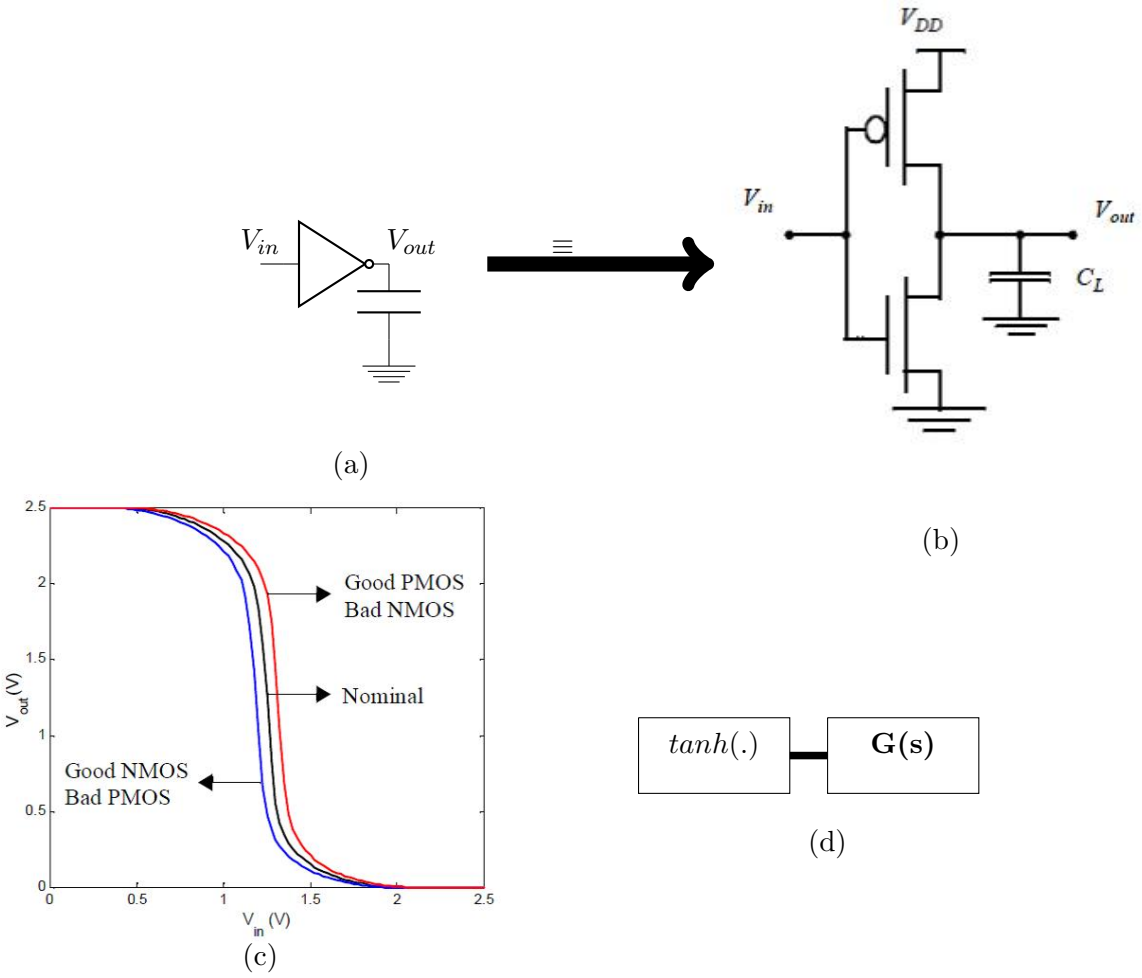


Figure 4.2: *(a)A CMOS Inverter (b) Internal MOS Transistor Circuit of an Inverter (c) Effect of Transistor sizes on Inverter Response (d) Inverter Non-linear Model*

in Fig. 4.2(c) how transistor sizes can affect the slope of the inverter response. Here the "good" device has a smaller oxide thickness (-3nm), a smaller length (-25nm), a higher width (+30nm), and a smaller threshold (-60mV). For a bad device, the opposite of these is true. To cater for the change in the slope due to device variations, we use the parameter V_s in Eq. 4.4. Similarly, bounds on the parameter V_{sat} represents how noise effects change in the saturation voltage. We use a linear model for the low pass filter

described by the transfer function $G(s)$.

$$G(s) = \frac{1}{1+Ts} \tag{4.5}$$

Here, $T = C_L \times R_{inverter}$, is the time constant of the low pass filter with $R_{inverter}$ being the inverter output resistance.

The above model of the inverter, a combination of "tanh(.)" non-linearity and a low pass filter, results in a non-polynomial expression for $f(x)$ in Eq. 4.1. To work around this, we use least-square polynomial approximation of the "$tanh(.)$" non-linearity. This approximation has a maximum error of 0.1% over the range $[-2, 2]$. The input and output of an inverter is thus related by the following equation,

$$V_n(\frac{1}{1+Ts}) = V_{out} \implies V_n = V_{out} + T\dot{V_{out}} \tag{4.6}$$

Here we have used the Laplace transform property $sV_{out} = \dot{V_{out}}$. By rearranging Eq. 4.6 and replacing V_n by Eq. 4.4, we get the following ODE for a CMOS inverter.

$$\dot{V_{out}} = -\frac{V_{out}}{T} - \frac{V_{sat}}{T} tanh(V_{in}/V_s) \tag{4.7}$$

$$\dot{V_{out}} = -\xi V_{out} - \zeta tanh(\rho V_{in}),\ \xi = \frac{1}{T},\ \zeta = \frac{V_{sat}}{T},\ \rho = 1/V_s \tag{4.8}$$

Let us denote by $\widetilde{p}(.)$, the polynomial approximation of the "tanh(.)" non-linearity. Then we have,

$$\dot{V_{out}} = -\xi V_{out} - \zeta \widetilde{p}(\rho V_{in}) \tag{4.9}$$

Let us denote the voltages on the nodes of the odd RO by $x_i,\ i = 1, ...n$, and that on the nodes of the even RO by $x(0, j), x(1, j),\ j = 0, 1...n$. Here n is the number of stages of an RO. Using Eq. 4.9 and applying KCL at each node of the RO, we obtain Eq. 4.10 and Eq. 4.11, representing ODEs for the odd and even stage ROs respectively.

$$\dot{x_i} = -\xi_i x_i - \zeta_i \widetilde{p}(\rho_i x_n),\ i = 1 \tag{4.10a}$$

$$\dot{x_i} = -\xi_i x_i - \zeta_i \widetilde{p}(\rho_i x_{(i-1)}),\ i \neq 1 \tag{4.10b}$$

$$\dot{x(0,j)} = -\xi^{f}_{(0,j)}x(0,j) - \zeta^{f}_{(0,j)}\widetilde{p}(\rho^{f}_{(0,j)}x(1,n-1)) - \xi^{c}_{(0,j)}x(0,j) - \zeta^{c}_{(0,j)}\widetilde{p}(\rho^{c}_{(0,j)}x(1,j)) \tag{4.11a}$$

$$\dot{x(1,j)} = -\xi^{f}_{(1,j)}x(1,j) - \zeta^{f}_{(1,j)}\widetilde{p}(\rho^{f}_{(1,j)}x(0,n-1)) - \xi^{c}_{(1,j)}x(1,j) - \zeta^{f}_{(1,j)}\widetilde{p}(\rho^{f}_{(1,j)}x(0,j)) \tag{4.11b}$$

Note that for the even stage RO, in addition to the path subscripts representing upper and lower paths, we use the f and c superscripts differentiating between the forward and cross-coupled inverters supplying the currents at a particular node.

4.1.1.2 Different Modelling Strategies for Odd and Even Topologies of RO

In this thesis, we consider two different topologies of ROs: odd stage RO and even stage RO as shown in Fig. 4.1. For an odd stage RO, we use the standard modelling approach considering every node voltage as a state variable. On the other hand, for an even stage RO we use the strategy suggested in [109], and instead of individual voltages on the oscillator nodes, we consider the differential and common mode operation. This is useful since it reduces the dimensionality of the system by an order of half and allows the analysis of the two modes to be performed in isolation.

Node voltages $x(0,j)$ and $x(1,j)$ of the even stage RO form differential pairs for all $j = 0, 1...n$. The differential component of the differential pair is $x(0,j) - x(1,j)$, and the common mode component is $x(0,j) + x(1,j)$. The even stage RO, while operating normally, has its oscillation manifested in the differential mode, whereas its common mode settles to the constant zero value. While we treat these two modes separately in the verification process, the overall verification depends on their combined verification. Note further that, while treating these two modes separately, we work with a system of half the dimension of the full even stage RO system. This greatly eases the verification process and reduces the overall computational time. If we assume that inverters are identical then, $\forall j \in [0, n-1]$, $\forall x : x \in \mathcal{X}$ such that $x(0,j) = x(1,j)$, we have, $\lim_{t\to\infty} \Phi(x,t) = x_e$. This means that the set,

$$\{x(0,j) = x(1,j), \forall j \in [0, n-1]\} \in \mathcal{X}_{dead}. \tag{4.12}$$

Similarly, for odd stage RO, if $x_1 = x_2 = x_3$, then, $\lim_{t\to\infty} \Phi(x,t) = x_e$.

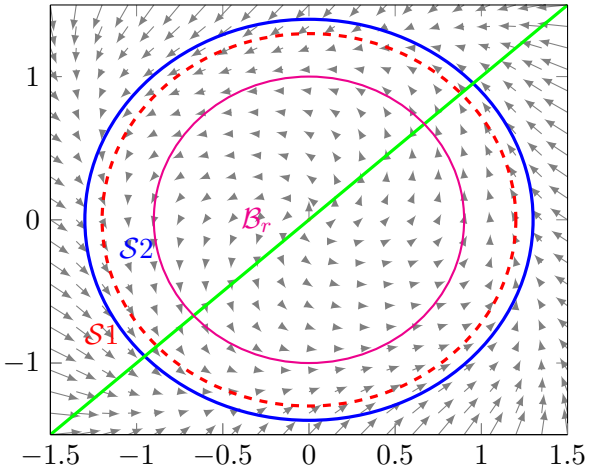


Figure 4.3: *RO Inevitability Verification Methodology, $\mathcal{S}1$, $\mathcal{S}2$ Separated by the Solid Blue circle; Dashed red circle: Limit cycle; Solid Straight line: Dead Set*

4.1.2 RO CDS Properties Verification using Lyapunov-like Certificates

We have seen in Chapter 3 the usefulness of Lyapunov certificates in verifying the attractive invariance of a compact set. Furthermore, an Escape certificate offers its usefulness in disproving invariance of a set. Lyapunov-like certificates have also been used for other interesting properties of CDS such as eventuality, avoidance, instability etc [80]. In this chapter, we use several Lyapunov-like certificates verifying the almost global inevitability of the limit cycle γ. These certificates are used to verify properties of different natures defined in various subsets of the CDS state space as shown in Fig. 4.3. The standard Lyapunov certificate can not be used for showing the attractive invariance of a set containing a limit cycle. Instead, in order to show invariance of a set containing a limit cycle, we use a Lyapunov-like certificate presented in [92].

Definition 4.6. *A set $\mathcal{X}_{\mathbf{AI}} \subset \mathcal{X}$ is called an attractive invariant (AI) with respect to a limit cycle $\gamma \subset \mathcal{X}_{\mathbf{AI}}$, iff, $\forall x_0 : x_0 \in \mathcal{X}_{\mathbf{AI}}$, $\forall t\ \Phi(x_0, t) \in \mathcal{X}_{\mathbf{AI}}$ and $\forall x_0 : x_0 \notin \mathcal{X}_{\mathbf{AI}}$, $\lim_{t\to\infty}\ \Phi(x_0, t) \in \mathcal{X}_{\mathbf{AI}}$.*

The lemma used in [92] shows not only invariance of a set, but convergence of outside trajectories to the invariant set as well. This lemma is stated below.

Lemma 4.1. *If there exists a polynomial with real coefficients $V : \mathbb{R}^n \to \mathbb{R}$, $\epsilon > 0$ and*

a minimum $\eta > 0$ *such that,*

$$V(x) > 0,\ \forall x : x \in \mathbb{R}^n \setminus 0 \tag{4.13a}$$

$$\{V(x) \leq 1\} \subseteq \{q(x) \leq \eta\} \tag{4.13b}$$

$$\{V(x) \geq 1\} \subseteq \{\frac{\partial V}{\partial x}(x) \cdot f(x,u) \leq -\epsilon\} \tag{4.13c}$$

then the set $\mathcal{S}2 := \{V(x) \leq 1\}$ *is an invariant set for Eq. 4.1, and it is contained in the set* $\{q(x) \leq \eta\}$*. Furthermore,* $\forall x_0 : x_0 \in \{V(x) > 1\},\ lim_{t\to\infty}\Phi(x_0,t) \in \{V(x) \leq 1\}$*.*

Proof. [92]. Trajectories starting inside the set $\mathcal{S}2$ (Fig. 4.3) can not leave this set as the derivative of $V(x)$ is strictly negative on the boundary of this set. That shows that the set $\mathcal{S}2$ is an invariant set. For trajectories starting in the set, $\mathcal{S}1 := \{V(x) > 1\}$, suppose $V(x(0)) = k,\ k > 0$, then from the condition,

$$\frac{\partial V}{\partial x}(x) \cdot f(x,u) \leq -\epsilon,$$

we have,

$$V(x(t)) \leq V(x(0)) - \epsilon t$$

This shows that the value of $V(x(t))$ will decrease to 1 in a time interval of $(0, (k-1)/\epsilon]$. This implies that all trajectories starting in the set $V(x) \leq k$, for $k \geq 1$, will eventually enter the set $V(x) \leq 1$. □

Inside the set $\mathcal{S}2$, trajectories may end up either belonging to the dead set $\mathcal{X}_{dead}$, or reach to within a small distance of the limit cycle γ. Let us define a set, $\mathcal{B}_r = V(x) \leq r,\ 0 < r < 1$, as shown by the pink circle in Fig. 4.3. This figure also shows sets $\mathcal{S}1$, $\mathcal{S}2$, $\mathcal{X}_{dead}$, and the limit cycle γ. To show trajectories starting in the set $\mathcal{B}_r \setminus \mathcal{X}_{dead}$ are not trapped in the dead set $\mathcal{X}_{dead}$, and eventually escape to the set $\mathcal{S}2 \setminus \mathcal{B}_r$, we introduce an Escape certificate similar to the Chetaev's instability certificate [61].

Lemma 4.2. *For a compact set* $\mathcal{B}_r \subset \mathcal{S}2$*, if there is a differentiable Escape certificate,* $\mathcal{E} : \mathbb{R}^n \to \mathbb{R}$*, such that*

$$\mathcal{E}(x) = 0,\ \forall x : x \in \mathcal{X}_{dead} \tag{4.14a}$$

$$\mathcal{E}(x) > 0,\ \forall x : x \in \mathcal{B}_r \setminus \mathcal{X}_{dead} \tag{4.14b}$$

$$\frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x,u) > 0,\ \forall x : x \in \mathcal{B}_r \setminus \mathcal{X}_{dead} \tag{4.14c}$$

then $\forall x_0 : x_0 \in \mathcal{B}_r,\ lim_{t\to\infty}\Phi(x_0,t) \notin \mathcal{B}_r$.

Proof. Assume that there exists $x_0 \in \mathcal{B}_r \setminus \mathcal{X}_{dead}$, such that $x(t) = \Phi(x_0,t)$ starting at x_0 remain in $\mathcal{B}_r$ as $t \to \infty$. From Eq. 4.21b of Lemma 4.2,

$$\mathcal{E}(x) = \int_0^\infty \frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x,u) > 0,$$

$$\lim_{t\to\infty} \mathcal{E}(x) = \infty.$$

This contradicts the assumption as $\mathcal{E}(x)$ should be bounded if $x(t)$ has to be in the bounded set $\mathcal{B}_r$. Furthermore, $x(t)$ can not reach the set $\mathcal{X}_{dead}$, since $\mathcal{E}(x) = 0,\ \forall x : x \in \mathcal{X}_{dead}$. Therefore, $x(t)$ has to escape the set $\mathcal{B}_r$ in finite time and reach the set $\mathcal{S}2 \setminus \mathcal{B}_r$. □

Corollary 4.1. *Assuming the RO does not have a chaotic behaviour, the set* $\mathcal{S}2 \setminus \mathcal{B}_r$ *must have a limit cycle.*

Though in this thesis, we assume the location of γ is given, we can use Cor. 4.1 to find a set where there must be a limit cycle.

To show that trajectories in the set $\mathcal{S}2 \setminus \mathcal{B}_r$ reach to a set, in close proximity to the limit cycle γ, we use the Eventuality certificate presented in [80]. Let us have a set $\mathcal{X}_{LC}$, such that, $\|y - x\| \leq \alpha,\ \forall x : x \in \mathcal{X}_{LC},\ y \in \gamma,\ \alpha > 0$.

Theorem 4.1. *If there exists a differentiable certificate of eventuality,* $E : \mathbb{R}^n \to \mathbb{R}$, *satisfying the following conditions*

$$E(x) \leq 0,\ \forall x : x \in (\mathcal{S}2 \setminus \mathcal{B}_r) \setminus \mathcal{X}_{dead} \tag{4.15a}$$

$$E(x) > 0,\ \forall x : x \in Cl(bd\ (\mathcal{S}2) \setminus bd\ (\mathcal{X}_{LC})) \tag{4.15b}$$

$$\frac{\partial E}{\partial x}(x) \cdot f(x,u) < 0,\ \forall x : x \in Cl(\mathcal{S}2 \setminus \mathcal{X}_{LC}) \tag{4.15c}$$

then for all initial conditions x_0 *such that* $x_0 \in \mathcal{S}2 \setminus \mathcal{B}_r$, *the trajectory* $x(t)$ *satisfies,* $x(T) \in \mathcal{X}_{LC}$ *for some* $T \geq 0$, *and for all* $t \in [0,T]$ $x(t) \in \mathcal{X}$. *Here Cl and bd denote closure and boundary of a closed set respectively.*

Proof. [80]. Let us assume $x_0 \in (\mathcal{S}2 \setminus \mathcal{B}_r) \setminus \mathcal{X}_{dead}$. The corresponding trajectory $x(t)$ starting at x_0, must leave $\mathcal{S}2 \setminus \mathcal{X}_{LC}$ in finite time due to Eq. 4.15c being strictly negative and $E(x)$ is bounded from below in this set. Let us suppose $x(t)$ leaves $\mathcal{S}2$ without reaching $\mathcal{X}_{LC}$. Eq. 4.15b of the theorem states that $E(x)$ is strictly positive at

the boundary of the set $(bd(\mathcal{S}2) \setminus bd(\mathcal{X}_{LC}))$, which is a contradiction to the assumption as $E(x)$ has to be non-positive. Therefore, $x(t)$ must reach $\mathcal{X}_{LC}$ before leaving $\mathcal{S}2$. □

We use these three certificates to verify the almost global inevitability of the limit cycle γ, for the odd and the differential mode of the even stage RO. For the common mode of the even stage RO, we further show that common mode voltages settle down to zero in the steady state. We verify this using the Lyapunov certificate restated for the common mode in Th. 4.2.

Theorem 4.2. *For the CDS of the RO with a vector field given in Eq. 4.1, and with the state vector replaced by* $x = \{x(0,0)+x(1,0), x(0,1)+x(1,1), .., x(0,n-1)+x(1,n-1)\}$, *let us assume an invariant set* $\mathcal{X}_{com}$, *which we call a common mode state space. Note that the assumption of this set being invariant is true since the node voltages can not go beyond the supply and ground voltages. If there exists a Lyapunov certificate* $\mathcal{L}(x)$ *such that,*

$$\mathcal{L}(0) = 0 \tag{4.16a}$$

$$\mathcal{L}(x) > 0, \ \forall x : x \in \mathcal{X}_{com} \setminus \{0\} \tag{4.16b}$$

$$\frac{\partial \mathcal{L}}{\partial x}(x) f(x,u) < 0, \ \forall x : x \in \mathcal{X}_{com} \setminus \{0\} \tag{4.16c}$$

then the set $\{x = 0\}$ *is asymptotically stable, and* $\forall x \in \mathcal{X}_{com}, \ \lim_{t\to\infty} \Phi(x,t) = 0$.

We avoid giving proof of Th 4.2 and readers are directed to [61, Ch.4].

4.2 AGI Verification of RO

4.2.1 Formulation of the Verification Problem

An exhaustive search of the complete state space is necessary to verify the almost global inevitability property. This search can not be formulated as a single deductive certificate query. Therefore, we use the divide and rule strategy and split the property in to several sub-properties. Every sub-property is verified in a subset of the state space, thus making the verification task less complex. We introduce two compact sets $\mathcal{S}1$, and $\mathcal{S}2$, such that $\mathcal{S}1 \cap \mathcal{S}2 = \emptyset$, and $\mathcal{S}1 \cup \mathcal{S}2 = \mathcal{X}$. We further define the set $\mathcal{B}_r \subset \mathcal{S}2$. These sets along with the limit cycle γ and the dead set $\mathcal{X}_{dead}$ are shown in Fig. 4.3. For illustration purposes, we have shown the projection of an RO vector field on a two dimensional space. In an actual three/four dimensional space, the dead set does not intersect the limit cycle and in the differential mode it is shifted to the origin. We

formulate verification of the AGI property as the conjunction of several sub-properties defined below.

Property 4.1. $\forall x_0 : x_0 \in \mathcal{S}1,\ lim_{t\to b}\ \Phi(x_0,t) \in \mathcal{S}2,\ b \in \mathbb{R}_{\geq 0}\ \wedge\ \forall x_0 : x_0 \in \mathcal{S}2,\ lim_{t\to\infty}\ \Phi(x_0,t) \in \mathcal{S}2.$

Property 4.2. $\forall x_0 : x_0 \in \mathcal{B}_r \setminus \mathcal{X}_{dead},\ lim_{t\to\infty}\ \Phi(x_0,t) \notin \mathcal{X}_{dead}\ \wedge\ lim_{t\to\infty}\ \Phi(x_0,t) \in \mathcal{S}2 \setminus \mathcal{B}_r.$

Property 4.3. $\forall x_0; x_0 \in \mathcal{S}2 \setminus \mathcal{B}_r, lim_{t\to b}\ \|y - \Phi(x_0,t)\| \leq \alpha,\ y \in \gamma,\ b \in \mathbb{R}_{\geq 0},\ \alpha > 0.$

We define the last property characterizing the common mode behaviour of the even stage RO in the invariant set $\mathcal{X}_{com}$.

Property 4.4. $\forall x_0 : x_0 \in \mathcal{X}_{com},\ \lim_{t\to\infty}\ \Phi(x_0,t) = 0.$

If we denote the almost global inevitability property by φ, Property 4.1 by $\varphi 1$ Property 4.2 by $\varphi 2$, Property 4.3 by $\varphi 3$, and Property 4.4 by $\varphi 4$, then we have

$$\varphi = \varphi 1 \wedge \varphi 2 \wedge \varphi 3 \tag{4.17}$$

for the odd stage RO, and,

$$\varphi = \varphi 1 \wedge \varphi 2 \wedge \varphi 3 \wedge \varphi 4 \tag{4.18}$$

for the even stage RO. A trajectory $x(t)$ of the odd stage RO satisfies φ, iff, it satisfies $\varphi 1$ in $\mathcal{S}1$, $\varphi 2$ in $\mathcal{B}_r$, and $\varphi 3$ in $\mathcal{S}2 \setminus \mathcal{B}_r$, i.e.,

$$\begin{aligned}\forall x : x \in \mathcal{X},\ x \models \varphi \iff (x \models \varphi 1\ \forall x : x \in \mathcal{S}1)\ \wedge (x \models \varphi 2\ \forall x : x \in \mathcal{S}2)\\ \wedge (x \models \varphi 3\ \forall x : x \in \mathcal{S}2 \setminus \mathcal{B}_r).\end{aligned} \tag{4.19}$$

Similarly, for an even stage RO,

$$\begin{aligned}\forall x : x \in \mathcal{X},\ x \models \varphi \iff (x \models \varphi 1\ \forall x : x \in \mathcal{S}1)\ \wedge (x \models \varphi 2\ \forall x : x \in \mathcal{B}_r)\\ \wedge (x \models \varphi 3\ \forall x : x \in \mathcal{S}2 \setminus \mathcal{B}_r) \wedge (x \models \varphi 4\ \forall x : x \in \mathcal{X}_{com}).\end{aligned} \tag{4.20}$$

4.2.2 The SOS-QE Approach to Verify AGI

In what follows, we formulate the properties stated in Sec. 4.2.1 as quantified FOFs over real polynomials. We verify these FOFs by numerically searching for a feasible

certificate satisfying the conditions of these formulas. Having numerical imprecisions, we verify, using symbolic QE, the validity of these certificates by checking falsification of the negation of the universally quantified formulas. We represent the set $\mathcal{X}$ by the semi-algebraic set, $g_k(x) \geq 0,\ k = 1, .., n$, where $g_k(x)$ is a vector of polynomials,

$$g_k(x) = \begin{pmatrix} (x_1 - x_1^L)(x_1^U - x_1) \\ (x_2 - x_2^L)(x_2^U - x_2) \\ . \\ . \\ (x_n - x_n^L)(x_n^U - x_n) \end{pmatrix}$$

Here $x_n = x_n$ for the odd stage RO, and $x_n = x(0,n) - x(1,n)$ for the differential mode of the even stage RO. Similarly, the parameter space $\mathcal{U}$ is represented by the inequality $a_j(u) \geq 0,\ j = 1, .., m$.

4.2.2.1 Verification of $\varphi 1$

The description of the Property $\varphi 1$ states that the set $\mathcal{S}2$ is invariant and that trajectories in the set $\mathcal{S}1$ are bound to reach $\mathcal{S}2$. To verify this property, we use Lemma 4.1 and encode its conditions as a FOF ψ_0, given below.

$$\psi_0 := \exists p^{\mathcal{P}} : \psi_1$$
$$\psi_1 := \forall x^{\mathcal{X}} : \psi_2$$
$$\psi_2 := \Big((x \neq 0 \implies V(p,x) > 0) \wedge \{(1 - V(p,x) \geq 0) \implies (\eta - q(x)) \geq 0\} \wedge$$
$$\{(V(p,x) - 1 \geq 0) \implies (\frac{\partial V}{\partial x}(p,x) \cdot f(x,u) \leq -\epsilon)\} \Big)$$

Here $p \in (\mathcal{P} \subset \mathbb{R})$ represents the coefficients of the certificate V. A sufficient condition for the verification of the property $\varphi 1$ is stated in the following theorem.

Theorem 4.3. *If there is a feasible certificate $V(x)$, fulfilling the conditions in Lemma 4.1, then, $(x \models \psi_0 \iff x \models \varphi 1)$, $\forall x_0 : x_0 \in (\mathcal{S}1 = (V(p,x) - 1 \geq 0) \cap (g_k(x) \geq 0))$, and $\forall x_0 : x_0 \in (\mathcal{S}2 = V(p,x) \leq 1)$.*

Proof. Follows directly from Lemma. 4.1. Existence of $V(x)$ fulfilling the conditions in Lemma. 4.1 verifies ψ_0. Therefore, we have, $\forall x_0 : x_0 \in (V(p,x) - 1 \geq 0)$, $\lim_{t\to\infty}\ \Phi(x_0,t) \in (V(x) \leq 1)$. Also, $\forall x_0 : x_0 \in (V(p,x) \leq 1)$, $\lim_{t\to\infty}\ \Phi(x_0,t) \in$

Algorithm 4 Verification of Property $\varphi 1$

INPUT: : RO CDS
OUTPUT: : $\varphi 1$ Verified/No-answer, $\mathcal{S}2$

1: $\mathcal{S}2 \leftarrow \emptyset$
2: $V \leftarrow Parametrize(V)$; Setting degree d and Parameters p of the Polynomial V
3: **if** V is feasible (fulfilling Lemma 4.1) **then**
4: | **if** $x \not\models \neg\psi_1,\ \forall x : x \in (V \geq 1),\ \forall x : x \in (V \leq 1)$ **then**
5: | | $\mathcal{S}2 \leftarrow (V \leq 1)$
6: | | $\mathcal{S}1 \leftarrow (V \geq 1) \cap (g_k(x) \geq 0)$
7: | | $(x \models \varphi 1 \iff x \models \psi_0)$
8: | | **break**
9: | **else if** d of $V < b$ **then** ; Here b is a user-defined upper bound on degree d
10: | | Increase Degree d of V ; d is incremented by 2
11: | | Goto Line(2)
12: | **else**
13: | | **break**
14: | **end if**
15: **else if** d of $V < b$ **then**
16: | Increase Degree d of V
17: | Goto Line(2)
18: **else**
19: | **break**
20: **end if**
21: **if** d = b & $x \not\models \varphi 1$ **then**
22: | No Answer about $\varphi 1$
23: **end if**
24: **return** $\mathcal{S}2$ and Truth value of $\varphi 1$

$(V(p,x) \leq 1)$. Since we have $\mathcal{S}1 = (V(p,x) - 1 \geq 0) \cap (g_k(x) \geq 0)$, and $\mathcal{S}2 = V(x) \leq 1$, therefore, $x \models \varphi 1$. □

The verification of $\varphi 1$ is associated with the existence of a certificate $V(x)$ that fulfils the condition of Lemma 4.1. Therefore, we start by searching for a feasible certificate $V(x)$ using Alg. 4. The search for $V(x)$ is performed following a numeric-symbolic approach utilizing two algebraic geometry tools, i.e. SOS programming and QE. The input of the algorithm is the CDS representing the RO and its output is the truth value of the property $\varphi 1$ and the set $\mathcal{S}2$. The algorithm starts by initializing the set $\mathcal{S}2$ (Line-1). This is followed by parametrizing the certificate $V(x)$ setting its degree d,

and declaring the coefficient parameters p (Line-2). Note that d is initially set up to be 2. The feasibility of the certificate $V(x)$ is checked in Line-3 of the algorithm. If there are SOS multipliers, $\{s_1^k, s_2, s_3, s_4^k, s_5^j\} \in \mathcal{S}_n,\ \forall k \in \{1,..,n\},\ \forall j \in \{1,..,m\}$, a positive number $\epsilon > 0$ and a minimum $\eta > 0$, then, $\forall x \in \mathcal{X},\ x \neq 0$, we can check the feasibility of $V(x)$ using the following SOS program.

$$\left(V(x) - \epsilon - \sum_{k=1}^{n} s_1^k(x) g_k(x) \right) \in \mathcal{S}_n, \tag{4.21a}$$

$$\left((\eta - q(x)) - s_2(x)(1 - V(x)) \right) \in \mathcal{S}_n, \tag{4.21b}$$

$$\left((-\epsilon - \frac{\partial V}{\partial x}(x) \cdot f(x,u)) - s_3(x)(V(x) - 1) - \sum_{k=1}^{n} s_4^k(x) g_k(x) - \sum_{j=1}^{m} s_5^j(x) a_j(u) \right) \in \mathcal{S}_n, \tag{4.21c}$$

Here $V(x),\ s_1^k,\ s_2,\ s_3,\ s_4^k,\ s_5^j$, are polynomials of degree d.

In this SOS program, constraint 4.21a enforces positive definiteness on the certificate $V(x)$ by introducing a small positive number ϵ. This constraint has to be satisfied in the state space $\mathcal{X}$ defined by the inequality $g_k(x) \geq 0,\ \text{for } k \in \{1,..,n\}\}$. Constraint 4.21b ensures that $\{V(x) \leq 1\} \subseteq \{q(x) \leq \eta\}$. Constraint 4.21c incorporates the set inclusion $\{V(x) \geq 1\} \subseteq \{\frac{\partial V}{\partial x}(x) \cdot f(x,u) \leq -\epsilon\}$. This constraint has additional constraints of state space and parameters defined by $g_k(x) \geq 0$ and $a_j(u) \geq 0$ respectively. The domain and parameters constraints are incorporated using the S-procedure discussed in Sec. 2.4.4. Note that, due to product terms, $s_2(x)(1 - V(x))$, and $s_3(x)(V(x) - 1)$, the above SOS program is non-convex and can not be solved by convex semi-definite programming. To work around this, we use an iterative convexification process; fixing $s_2(x), s_3(x)$, finding $V(x)$ and vice versa.

Proposition 4.1. *If the the SOS program in Eq. 4.21 is feasible, then the certificate $V(x)$ satisfies conditions of Lemma. 4.1.*

Proof. Eq. 4.21a being SOS, therefore, $V(x) - \epsilon - \sum_{k=1}^{n} s_1^k(x) g_k(x) \geq 0$. Since $s_1^k(x)$ is SOS for all $k = 1,..n$, $g_k(x) \geq 0$ for all $k = 1,..n$, and $\epsilon > 0$, we have, $V(x) > 0,\ \forall x : x \in \mathcal{X},\ x \neq 0$. Similarly, Eq. 4.21b being SOS, s_2 is SOS, and $\eta > 0$, therefore, $(\eta - q(x)) \geq (1 - V(x))$. This shows $\{V(x) \leq 1\} \subset \{q(x) \leq \eta\}$. Lastly, since Eq. 4.21c

is SOS, therefore,

$$(-\epsilon - \frac{\partial V}{\partial x}(x) \cdot f(x,u)) - s_3(x)(V(x)-1) - \sum_{k=1}^{n} s_4^k(x) g_k(x) - \sum_{j=1}^{m} s_5^j(x) a_j(u) \geq 0.$$

Since, $\epsilon > 0$, $s_3(x), s_4^k(x), s_5^j$ being SOS for all $k = 1,..n$, for all $j = 1,..m$, $g_k(x) \geq 0$ for all $k = 1,..n$, $a_j(u) \geq 0$ for all $j = 1,..m$, we have, $(V(x)-1) \leq -\frac{\partial V}{\partial x}(x) \cdot f(x,u)) - \epsilon$. This shows $\{V(x) \geq 1\} \subset \{\frac{\partial V}{\partial x}(x) \cdot f(x,u) \leq -\epsilon\}$ □

The above SOS program, if feasible, returns a certificate of invariance $V(x)$ with its parameters p fixed within a limited numerical precision. The numerical inaccuracies, caused by the numerical SOS programming, may change the validity of the certificate $V(x)$ for ill-posed systems. Therefore, adopting a conservative approach, we further verify this certificate using symbolic QE. Note that in QE, coefficients are represented in $\mathbb{Q}^n$. Using QE, we check the falsification of the negation of the formula ψ_1, a boolean combination of polynomial inequalities with universal quantification only. We use the formula ψ_1 and not ψ_0, since the certificate $V(x)$ returned by the SOS program has a fixed structure and can be encoded as a FOF which is only universally quantified. We verify the disjunctive normal form (DNF) of the formula $\neg\psi_1$[1] shown below.

$$\neg\psi_1 := \forall x^{\mathcal{X}} : \neg\psi_2$$

$$\neg\psi_2 := \Big(V(p,0) \neq 0 \vee (x \neq 0 \wedge V(p,x) < 0) \vee \{(1 - V(p,x) \geq 0) \wedge (\eta - q(x)) < 0\} \vee$$
$$\{(V(p,x) - 1 \geq 0) \wedge (\frac{\partial V}{\partial x}(p,x) \cdot f(x,u) > -\epsilon)\} \Big)$$

On the refutation of $\neg\psi_1$, we conclude, $(x \models \varphi 1 \iff x \models \psi_0)$, $\forall x \in \mathcal{S}1$, $\forall x \in \mathcal{S}2$, (Line 4-7). If either the SOS program for a certificate $V(x)$ of degree d is infeasible, or the QE tool returns a true valuation of the formula $\neg\psi_1$, we repeat the process by increasing the degree d of the certificate $V(x)$ (Line 9-15). Note that degree d is incremented by 2 until it reaches a user-defined upper bound b. If a desired certificate $V(x)$ can not be found, the algorithm concludes inconclusiveness about the truth value of $\varphi 1$ (Line 17-18). For a valid certificate $V(x)$, the algorithm returns the invariant set $\mathcal{S}2 = (V(x) \leq 1)$ (Line-5, Line-22). Since our certificate based deductive approach is a sufficient criterion for the verification of property $\varphi 1$, inconclusiveness of the result

[1] $J \implies K = \neg J \vee K$, $\neg(\neg J \vee K) = J \wedge \neg K$

does not imply falsification of the property. It is still quite possible, by searching for an even higher degree certificate $V(x)$, that we are able to verify the property $\varphi 1$.

4.2.2.2 Verification of $\varphi 2$ and $\varphi 3$

The description of $\varphi 2$ and $\varphi 3$ states that no trajectory in the set $\mathcal{B}_r \setminus \mathcal{X}_{dead}$ can be trapped in the set $\mathcal{X}_{dead}$, and that every trajectory will eventually escape the set $\mathcal{B}_r \setminus \mathcal{X}_{dead}$. Furthermore, all trajectories in the set $\mathcal{S}2 \setminus \mathcal{B}_r$ eventually reach to a set within a small distance of the limit cycle γ. These characteristics of trajectories can be described by the Escape certificate of Lemma 4.2, and by the Eventuality certificate of Th. 4.1. Therefore, to verify properties $\varphi 2$ and $\varphi 3$, we use Lemma 4.2, Th. 4.1 and encode their conditions by FOFs Θ_0 and θ_0 respectively.

$$\Theta_0 := \exists p^{\mathcal{P}} : \Theta_1$$
$$\Theta_1 := \forall x^{\mathcal{X}} : \Theta_2$$
$$\Theta_2 := \Big(\{(x \in \mathcal{X}_{dead}) \implies \mathcal{E}(p,x) = 0\} \wedge \{(x \in \mathcal{B}_r \setminus \mathcal{X}_{dead}) \implies \mathcal{E}(p,x) > 0)\} \wedge$$
$$\{(x \in \mathcal{B}_r \setminus \mathcal{X}_{dead}) \implies (\frac{\partial \mathcal{E}}{\partial x}(p,x) \cdot f(x,u) > 0)\} \Big).$$

$$\theta_0 := \exists p^{\mathcal{P}} : \theta_1$$
$$\theta_1 := \forall x^{\mathcal{X}} : \theta_2$$
$$\theta_2 := \Big(\{(x \in (\mathcal{S}2 \setminus \mathcal{B}_r) \setminus \mathcal{X}_{dead}) \implies E(p,x) \leq 0\} \wedge$$
$$\{(x \in Cl(bd\mathcal{S}2 \setminus bd\mathcal{X}_{LC})) \implies E(p,x) > 0\} \wedge$$
$$\{(x \in Cl(\mathcal{S}2 \setminus \mathcal{X}_{LC})) \implies (\frac{\partial E}{\partial x}(p,x) \cdot f(x,u) < 0)\} \Big)$$

A sufficient condition for the verification of properties $\varphi 2$, and $\varphi 3$ is stated in the following theorem.

Theorem 4.4. *If in the set $(\mathcal{B}_r \setminus \mathcal{X}_{dead}) \subset \mathcal{S}2$, there is an Escape certificate $\mathcal{E}(x)$ satisfying the conditions of Lemma 4.2, and there is an Eventuality certificate $E(x)$ in the set $\mathcal{S}2 \setminus \mathcal{B}_r$, satisfying the conditions of Th. 4.1, then $(x \models \Theta_0 \iff x \models \varphi 2)$, $\forall x_0 : x_0 \in \mathcal{B}_r \setminus \mathcal{X}_{dead}$, and $(x \models \theta_0 \iff x \models \varphi 3)$, $\forall x_0 : x_0 \in \mathcal{S}2 \setminus \mathcal{B}_r$.*

Proof. Existence of the Escape certificate fulfilling conditions in Lemma 4.2 verifies Θ_0. Therefore, from conditions of Θ_0, we have $\forall x_0 : x_0 \in \mathcal{B}_r \setminus \mathcal{X}_{dead},\ \forall t,\ \Phi(x_0, t) \notin \mathcal{X}_{dead}$, and, $\lim_{t\to\infty} \Phi(x_0, t) \in (\mathcal{S}2 \setminus \mathcal{B}_r)$. This implies, $x \models \varphi 2,\ \forall x_0 : x_0 \in \mathcal{B}_r \setminus \mathcal{X}_{dead}$. Similarly, an Eventuality certificate obeying conditions of Th. 4.1 verifies θ_0, which consequently, by the same argument as that of Θ_0, verifies $\varphi 3$, $\forall x_0 : x_0 \in \mathcal{S}2 \setminus \mathcal{B}_r$. □

From Th. 4.4, we see that while verifying $\varphi 2$ and $\varphi 3$, we require an Escape certificate in the set $\mathcal{B}_r$, and an Eventuality certificate in the set $\mathcal{S}2 \setminus \mathcal{B}_r$. We search for these two certificates using Alg. 5. The inputs of the algorithm are the RO CSD model and the set $\mathcal{S}2 = (V(x) \leq 1)$. The algorithm starts with parametrizing polynomials $E,\ \mathcal{E}$ and initializing the set $\mathcal{B}_r = (V(x) \leq r),\ 0 < r < 1$ (Line 1). Similar to Alg. 4, we use the numeric-symbolic combination of SOS-QE for the implementation of Alg. 5 (Line 2-5). We search the Escape certificate $\mathcal{E}(x)$ in the set $\mathcal{B}_r$ using the SOS program followed by its validation through QE. The SOS program is given in Eq. 4.22.

$$\left(\mathcal{E}(x) + \sum_{k=1}^{n} s_6^k(x) g_k^{dead}(x) = 0\right) \tag{4.22a}$$

$$\left(\mathcal{E}(x) - \epsilon - s_7(x)(r - V(x)) + \sum_{k=1}^{n} s_8^k(x) g_k^{dead}(x)\right) \in \mathcal{S}_n \tag{4.22b}$$

$$\left(\frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x,u) - \epsilon - s_9(x)(r - V(x)) + \sum_{k=1}^{n} s_{10}^k(x) g_k^{dead}(x) - \sum_{j=1}^{m} s_{11}^j(x) a_j(u)\right) \in \mathcal{S}_n \tag{4.22c}$$

$\forall x \in \mathcal{B}_r, \{s_7, s_9, s_{11}^j\} \in \mathcal{S}_n, \{s_6^k, s_8^k, s_{10}^k\} \in \mathcal{R}_n, \epsilon > 0, 0 < r < 1, \forall k \in \{1, .., n\},\ \forall j \in \{1, .., m\}$. Here, $\mathcal{E},\ \{s_6^k, s_7, s_8^k, s_9, s_{10}^k, s_{11}^j\}$ are polynomials of degree d.

The constraint in Eq. 4.22a of the above SOS program ensures, $\mathcal{E}(x) = 0$, in the dead set represented as $\mathcal{X}_{dead} = \{x \in \mathbb{R}^n \ : g_k^{dead}(x) = 0,\ \text{for } k \in \{1, .., n\}\}$. The positive-definiteness of the certificate $\mathcal{E}(x)$, in the set $(\mathcal{B}_r = V(x) \leq r) \setminus \mathcal{X}_{dead}$, is ensured in the second constraint of Eq. 4.22b. The last constraint of Eq. 4.22c ensures positive-definiteness of the derivative of $\mathcal{E}(x)$ in the set $\mathcal{B}_r \setminus \mathcal{X}_{dead}$. The domain and parameters constraints are incorporated using the S-procedure discussed in Sec. 2.4.4.

Proposition 4.2. *If the the SOS program in Eq. 4.22 is feasible, then the certificate $\mathcal{E}(x)$ satisfies the conditions of Lemma. 4.2.*

Proof. Eq. 4.22a is $\mathcal{E}(x) + \sum_{k=1}^{n} s_6^k(x) g_k^{dead}(x) = 0$. Since, $g_k^{dead}(x) = 0,\ \forall k \in \{1, .., n\}$, we have $\mathcal{E}(x) = 0\ \forall x : x \in \mathcal{X}_{dead}$. The condition Eq. 4.22b is SOS, therefore, $\mathcal{E}(x) -$

Algorithm 5 Verification of Property $\varphi 2$ and $\varphi 3$

INPUT: : System of ODEs for RO, Set $\mathcal{S}2$
OUTPUT: : $\varphi 2$ and $\varphi 3$ Verified/No-answer

1: $\mathcal{B}_r \leftarrow V(x) \leq r$; $0 < r < 1$
2: $\mathcal{E} \leftarrow Parametrize(\mathcal{E})$; Setting degree d and parameters p for $\mathcal{E}$
3: **if** $\mathcal{E}$ is feasible (fulfilling Lemma. 4.2) **then**
4: | **if** $x \not\models \neg\Theta_1, \forall x \in \mathcal{B}_r$ **then**
5: | | $(x \models \Theta_0 \iff x \models \varphi 2),\ \forall x \in \mathcal{B}_r$
6: | | $E \leftarrow Parametrize(E)$; Setting degree d and parameters p for E
7: | | **if** E is feasible (fulfilling Th. 4.1) **then**
8: | | | **if** $x \not\models \neg\theta_1, \forall x \in (\mathcal{S}2 \setminus \mathcal{B}_r)$ **then**
9: | | | | $x \models \theta_0 \iff x \models \varphi 3,\ \forall x \in \mathcal{S}2 \setminus \mathcal{B}_r$
10: | | | | **break**
11: | | | **else if** d of $E < b$ **then** ; Here b is a user-defined upper bound on degree d
12: | | | | Increase Degree d of E ; d is incremented by 2
13: | | | | Goto Line(6)
14: | | | **else**
15: | | | | **break**
16: | | | **end if**
17: | | **else if** d of $E < b$ **then**
18: | | | Increase Degree d of E
19: | | | Goto Line(6)
20: | | **else**
21: | | | **break**
22: | | **end if**
23: | **else if** d of $\mathcal{E} < b$ **then** ; Here b is a user-defined upper bound on degree d
24: | | Increase Degree d of $\mathcal{E}$; d is incremented by 2
25: | | Goto Line(2)
26: | **else**
27: | | **break**
28: | **end if**
29: **else if** d of $\mathcal{E} < b$ **then**
30: | Increase Degree d of $\mathcal{E}$
31: | Goto Line(2)
32: **else**
33: | **break**
34: **end if**
35: **if** $\nexists\, \mathcal{E}$ of $d \leq b$ in $\mathcal{B}_r$ **then**
36: | No Answer about $\varphi 2$
37: **end if**
38: **if** $\nexists\, E$ $d \leq b$ in $\mathcal{S}2 \setminus \mathcal{B}_r$ **then**
39: | No Answer about $\varphi 3$
40: **end if**
41: **return** Truth value of $\varphi 2$ and $\varphi 3$

$\epsilon - s_7(x)(r - V(x)) + \sum_{k=1}^{n} s_8^k(x) g_k^{dead}(x) \geq 0$. Since, $\epsilon > 0$, $s_7(x) \geq 0$, and $g_k^{dead}(x) = 0$, $\forall k \in \{1, .., n\}$, we have, $\mathcal{E}(x) \geq (r - V(x)) - \sum_{k=1}^{n} g_k^{dead}(x) + \epsilon$. Therefore, $\{(V(x) \leq r) \setminus (\sum_{k=1}^{n} g_k^{dead}(x) = 0)\} \subset (\mathcal{E}(x) > 0)$. The last constraint in Eq. 4.22b is also SOS, therefore,

$$\frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x, u) - \epsilon - s_9(x)(r - V(x)) + \sum_{k=1}^{n} s_{10}^k(x) g_k^{dead}(x) - \sum_{j=1}^{m} s_{11}^j(x) a_j(u) \geq 0.$$

Since s_9, s_{11}^j $\forall j \in \{1, .., m\}$ are SOS multipliers, and $g_k^{dead}(x) = 0$ $\forall k \in \{1, .., n\}$, $a_j(u) \geq 0$ $\forall j \in \{1, .., m\}$, $\epsilon > 0$, we have, $\frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x, u) \geq (r - V(x)) - \sum_{k=1}^{n} g_k^{dead}(x)$. Therefore, we have, $\{(V(x) \leq r) \setminus (\mathcal{X}_{dead})\} \subset (\frac{\partial \mathcal{E}}{\partial x}(x) \cdot f(x, u) > 0)$. □

The numerically constructed Escape certificate $\mathcal{E}(x)$ is further validated by checking the falsification of the formula $\neg\Theta_1$ using the symbolic QE (Line 4-5). The QE checks the falsification of the DNF of $\neg\Theta_1$ given below.

$$\neg\Theta_1 := \forall x^{\mathcal{X}} : \neg\Theta_2$$

$$\neg\Theta_2 := \Bigg(\{(x \in \mathcal{X}_{dead}) \wedge (\mathcal{E}(p, x) \neq 0)\} \vee (x \in \mathcal{B}_r \setminus \mathcal{X}_{dead} \wedge \mathcal{E}(p, x) < 0)$$
$$\vee \{(x \in \mathcal{B}_r \setminus \mathcal{X}_{dead}) \wedge (\frac{\partial \mathcal{E}}{\partial x}(p, x) \cdot f(x, u) < 0)\} \Bigg)$$

On falsification of the formula $\neg\Theta_1$, we conclude that $x \models \varphi 2$, $\forall x : x \in \mathcal{B}_r$, Line 5. If either the SOS program results in an infeasible certificate, or the QE returns a "true" answer for the formula $\neg\Theta_1$, we repeat the process for an increased degree d of the certificate $\mathcal{E}(x)$ (Line 26-27, Line 21-22). Note that d is incremented by 2 in each iteration until it reaches a user defined upper bound b. Similarly, the structure of the Eventuality certificate $E(x)$ is identified through the SOS programming, followed by checking its validity using symbolic QE (Line 6-20). The SOS program for the

construction of $E(x)$ is given below.

$$\Bigg(-E(x) - s_{12}(x)(1-V(x)) + s_{13}(x)(r-V(x)) + \sum_{k=1}^{n} s_{14}^{k}(x) g_k^{dead}(x) + \sum_{k=1}^{n} s_{15}^{k}(x) g_k^{LC}(x) \Bigg) \in \mathcal{S}_n \tag{4.23a}$$

$$\Big(E(x) - \epsilon - s_{16}(x)(1-V(x)) \Big) \in \mathcal{S}_n \tag{4.23b}$$

$$\Bigg(-\epsilon - \frac{\partial E}{\partial x}(x) \cdot f(x,u) - s_{17}(x)(1-V(x)) + \sum_{k=1}^{n} s_{18}^{k}(x) g_k^{LC}(x) - \sum_{j=1}^{m} s_{19}^{j}(x) a_j(u) \Bigg) \in \mathcal{S}_n \tag{4.23c}$$

$\forall x \in \mathcal{X}, \{s_{12}, s_{13}, s_{15}^k, s_{17}, s_{18}^k, s_{19}^j\} \in \mathcal{S}_n, s_{14}^k \in \mathcal{R}_n, \forall k \in \{1,..,n\}, \forall j \in \{1,..,m\}, \epsilon > 0$. Here, $E(x)$, $s_{12}, s_{13}, s_{15}^k, s_{17}, s_{18}^k, s_{19}^j, s_{14}^k$, are polynomials of degree d.

The first two constraints of the SOS program in Eq. 4.23 ensure positive-definiteness of $-E(x)$ and $E(x)$ in sets, $(\mathcal{S}2 \setminus \mathcal{B}_r) \setminus \mathcal{X}_{dead} \setminus \mathcal{X}_{LC}$, and, $\partial\mathcal{S}2 = (V(x) = 1)$, respectively. The last constraint is responsible for ensuring negative-definiteness of the derivative of $E(x)$ in the set $\mathcal{S}2 \setminus \mathcal{X}_{dead}$. Again, the domain and parameters constraints are incorporated using the S-procedure discussed in Sec. 2.4.4.

Proposition 4.3. *If the the SOS program in Eq. 4.23 is feasible, then the certificate $E(x)$ satisfies conditions of Th. 4.1.*

Proof. Constraint in Eq. 4.23a is SOS, therefore, $-E(x) - s_{12}(x)(1-V(x)) + s_{13}(x)(r - V(x)) + \sum_{k=1}^{n} s_{14}^{k}(x) g_k^{dead}(x) + \sum_{k=1}^{n} s_{15}^{k}(x) g_k^{LC}(x) \geq 0$.
Since, $g_k^{dead}(x) = 0 \ \forall k \in \{1,..,n\}$, $s_{12}, s_{13}, s_{15}^k \ \forall k \in \{1,..,n\}$ are SOS, therefore, $-E(x) \geq 0, \ \forall x : x \in \{((V(x) \leq 1) \setminus \mathcal{X}_{dead}) \setminus (g_k^{LC} \geq 0 \ \forall k \in \{1,..,n\})\}$. The constraint in Eq. 4.23a is SOS, therefore, $E(x) - \epsilon - s_{16}(x)(1 - V(x)) \geq 0$. Since s_{16} is SOS, $\epsilon > 0$, $(1 - V(x)) \geq 0$, therefore, $E(x) > 0 \ \forall x : x \in (V(x) \leq 1)$. The last constraint in Eq. 4.23c as well as $s_{17}, s_{18}^k \ \forall k \in \{1,..,n\}$, $s_{19}^j \ \forall j \in \{1,..,m\}$ are SOS. Furthermore, $\epsilon > 0$, $(1 - V(x)) \geq 0$, $g_k^{LC}(x) \geq 0$, $a_j(u) \geq 0$, therefore, $\frac{\partial E}{\partial x}(x) \cdot f(x,u) < 0 \ \forall x : x \in \{(V(x) \leq 1) \setminus (g_k^{LC}(x) \geq 0)\}$. □

To verify the validity of the Eventuality certificate $E(x)$, constructed by the SOS

program, we check the truth value of the formula $\neg\theta_1$ given in its DNF below.

$$\neg\theta_1 := \forall x^{\mathcal{X}} : \neg\theta_2$$

$$\neg\theta_2 := \Big(\{(x \in \mathcal{S}2 \setminus \mathcal{B}_r \setminus \mathcal{X}_{dead} \setminus \mathcal{X}_{LC}) \wedge E(p,x) > 0)\} \vee \{(x \in \partial\mathcal{S}2) \wedge E(p,x) \leq 0) \\ \vee \{(x \in \mathcal{S}2 \setminus \mathcal{X}_{LC}) \wedge (\frac{\partial E}{\partial x}(p,x) \geq 0)\} \Big)$$

Failing to either construct the Eventuality certificate $E(x)$ numerically, using the SOS program, or getting a "True" model of the formula $\neg\theta_1$, we repeat the process for an increased degree d of $E(x)$ (Line 16-17, 11-12). If for a maximum degree d, the SOS-QE approach is not able to construct validated certificates $\mathcal{E}(x)$ or $E(x)$ in their respective sets, the corresponding property $\varphi 2$ or $\varphi 3$ is declared inconclusive (Line 31-36). Note that d is incremented by 2 in each iteration until it reaches a user defined upper bound b.

Remark 4.1. *The small positive number ϵ in the above SOS programs relaxes the strict positivity/negativity conditions conservatively, and does not contradict the validity of our results. Multipliers s of the polynomial equalities are not constrained to be $\mathcal{S}_n$.*

4.2.2.3 Verification of $\varphi 4$

Property $\varphi 4$ is concerned with showing that in the common mode of the even stage RO, all trajectories converge to the zero common mode voltage. This property is closely associated with the Lyapunov stability theorem which has been stated in Th. 4.2 for the common mode of the even stage RO. Therefore, we verify $\varphi 4$ by searching for a Lyapunov certificate $\mathcal{L}$ satisfying the conditions of Th. 4.2. We follow a similar procedure as that for property $\varphi 1$, and search for the Lyapunov certificate using the SOS-QE approach. The FOF encoding of the conditions of Th. 4.2 is given below.

$$\Psi_0 := \exists p^{\mathcal{P}} : \Psi_1$$
$$\Psi_1 := \forall x^{\mathcal{X}} : \Psi_2$$

$$\Psi_2 := \Big(\{(x = 0) \implies \mathcal{L}(p,x) = 0\} \wedge \{(x \neq 0 \wedge x : x \in \mathcal{X}_{com}) \implies \mathcal{L}(p,x) > 0\} \wedge \\ \{(x \neq 0 \wedge x : x \in \mathcal{X}_{com}) \implies (\frac{\partial \mathcal{L}}{\partial x}(p,x) \cdot f(x,u) < 0)\} \Big)$$

We state the sufficient condition for the verification of property $\varphi 4$ in the following theorem.

Theorem 4.5. *If in the invariant set $\mathcal{X}_{com} \subset \mathbb{R}^n$ there is a Lyapunov certificate $\mathcal{L}(x)$ satisfying the conditions of Th. 4.2, then, $x \models \Psi_0 \iff x \models \varphi 4,\ \forall x : x \in \mathcal{X}_{com}$, where $x = \{x(0,0) + x(1,0), x(0,1) + x(1,1), .., x(0,n-1) + x(1,n-1)\}$.*

Proof. A feasible Lyapunov certificate $\mathcal{L}(x)$ according to Th. 4.2 is a true model of the FOF Ψ_0. Since the set $\mathcal{X}_{com}$ is invariant, we have, $\forall x_0 : x_0 \in \mathcal{X}_{com},\ \forall t,\ \Phi(x_0, t) \in \mathcal{X}_{com}$. Furthermore, since, $\frac{\partial \mathcal{L}}{\partial x}(p,x) \cdot f(x,u) < 0,\ \mathcal{L}(p,x) > 0,\ \forall x : x \in \mathcal{X}_{com},\ x \neq 0$, therefore, $lim_{t\to\infty}\ \Phi(x_0,t) = 0$. This shows $x \models \varphi 4$. □

The above theorem illustrates that the sufficient condition for the verification of property $\varphi 4$ is the existence of a Lyapunov certificate in the common mode invariant state space $\mathcal{X}_{com}$. We use an algorithm, similar to to that of Alg. 4, and search for the certificate $\mathcal{L}$ following the numeric-symbolic approach. A SOS program numerically constructing the Lyapunov certificate is given below.

$$\mathcal{L}(0) = 0 \tag{4.24a}$$

$$\left(\mathcal{L}(x) - \sum_{k=1}^{n} s_{20}^k(x) g_{diff}^k(x) + \epsilon \right) \in \mathcal{S}_n \tag{4.24b}$$

$$\left(-\epsilon - \frac{\partial \mathcal{L}}{\partial x}(x) \cdot f(x,u) - \sum_{k=1}^{n} s_{21}^k(x) g_{diff}^k(x) - \sum_{j=1}^{m} s_{22}^j(x) a_j(u) \right) \in \mathcal{S}_n \tag{4.24c}$$

$\forall x \in \mathcal{X}_{com}, \{s_{20}^k, s_{21}^k, s_{22}^j\} \in \mathcal{S}_n, \forall k \in \{1, .., n-1\}, \forall j \in \{1, .., m\}, \epsilon > 0$. Here, $\mathcal{L}(x),\ s_{20}^k,\ s_{21}^k,\ s_{22}^j$, are polynomials of degree d.

The constraints in Eq. 4.24b and Eq. 4.24c, ensure the positive and negative definiteness of the Lyapunov certificate $\mathcal{L}(x)$ in the set $\mathcal{X}_{com} \setminus \{0\}$. The domain and parameter constraints have been added following the S-procedure. If the above SOS program is feasible, then the constructed Lyapunov certificate $\mathcal{L}(x)$ satisfies the conditions of Th. 4.2. Since the certificate constructed by the above SOS program suffers from numerical inaccuracies, we further validate it using symbolic QE. This is done by checking the falsification of the formula $\neg\Psi_1$ as previously discussed for all other certificates. This alternating application of SOS-QE is performed until either a valid Lyapunov certificate $\mathcal{L}(x)$ is found, or we reach a point where we conclude inconclusiveness of the approach to find the truth value of $\varphi 4$ for a maximum degree d of $\mathcal{L}(x)$.

Parameters	Values
$R_{inverter}$	$[0.98\ 1.2]e3\Omega$
C_L	$[0.98\ 1.2]e-12F$
V_{sat}	$[0.98\ 1.2]V$
V_s	$[0.23\ 0.27]$

Table 4.1: *Inverter Parameters*

4.3 Experimental Evaluation

We have demonstrated the applicability and effectiveness of our approach by applying it to three stage odd and two stage even ROs. The parameters we have used for an inverter model are given in Table. 4.1. Note that we have ranges of values for different parameters. Furthermore, these ranges have been chosen randomly and can be adjusted by the designers according to their needs of checking the design for various ranges of different parameters. We used the YALMIP [67] solver within MATLAB for SOS programming, and REDLOG [31] for QE, on a 2.6 GHZ Intel Core i5 machine with 4 GB of memory.

For an odd RO, we were able to compute a degree-4 AI certificate, Appendix B.4. The AI set, marked by the level set $V(x) \leq 1$, is shown in Fig. 4.4 (see this set in three-dimension in Appendix B.6). For the purpose of illustration, we have also shown an invariant set generated from a degree-10 AI certificate. This emphasises that higher degree certificates have the ability to closely over-approximate the set containing the limit cycle as it has higher degree of freedom and align itself along the limit cycle, however at a high computation/verification cost. Inside the AI set, we showed that trajectories escape the set $V \leq r$, by computing a degree-2 Escape certificate. Similarly, further convergence of the trajectories, to within a small distance of the limit cycle, has been shown by computing a degree-4 Eventuality certificate in the set $\{V \leq 1 \wedge V \geq r\}$. Time taken by the SOS solver to compute these certificates is listed in the second column of Table. 4.2. Verification of these certificates in REDLOG, given how large a formula it can handle, has been divided in to the verification of the individual clauses of the FOFs benefiting from its DNF. Since we were interested in the negation of FOFs in the DNF, we verified whether each clause was "false". The verification times of the QE are listed in the third column of Table 4.2. For AI and Escape certificates, REDLOG successfully verified the negation of their universally quantified FOFs. Derivatives of these

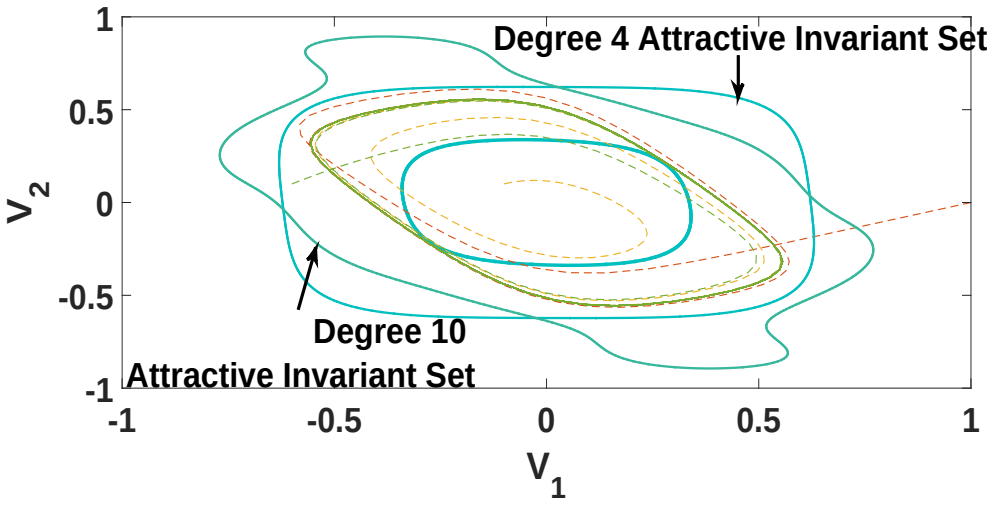


Figure 4.4: *ODD RO Attractive Invariant Set, defined by* $\{V <= 1\}$*: Outer Solid plots, Degree 4 and Degree 10,* $\{V = r\}$*: Inner Solid plot of degree 4, Trajectories: Dashed plots*

certificates were symbolically calculated using the MATLAB symbolic toolbox before verifying it using QE. A time out was reported by the REDLOG tool for all clauses of the eventuality FOF of the odd RO. The reason for these time outs is the set, an intersection of two level curves of the AI certificate, that puts an additional burden on the solver resulting in its time out. To overcome this issue, we instead conservatively over-under approximate the set $\{V \leq 1 \wedge V \geq r\}$, by a quadratic polynomial, and construct the Eventuality certificate for this new set. This solved our problem and REDLOG has been able to verify the Eventuality certificate in this conservative approximation of the set $\{V \leq 1 \wedge V \geq r\}$. Similarly, for the even stage RO, we computed a degree-10 AI, a degree-4 Escape, a degree-6 Eventuality and a degree-4 Lyapunov certificate (Appendix B.5). Their SOS computation times are listed in the second column of Table 4.3. The AI set, represented by the level curve $V(x) \leq 1$, is shown in Fig. 4.5. Three trajectories in different sets have also been shown. Though of degree-10, from the QE point of view, the AI certificate has fewer monomials, and was thus easily verified by the REDLOG. All clauses of the AI, Escape and Lyapunov certificates were verified by the QE. For the Eventuality certificate, we followed the same procedure as we did for odd stage, over-under approximating the set $\{V \leq 1 \wedge V \geq r\}$ with quadratic polynomial level curves, and verified the negation of the corresponding FOF. The QE verification times for these certificates are reported in the third column of Table 4.3.

Certificate	YALMIP-SOS Time(Sec)	REDLOG-QE Time(Sec)
Attractive Invariants	**824.8 (Degree 4)**	**Clause 1 =0.219** **Clause 2 =0.047** **Clause 3 =8.222**
Escape	**6.3 (Degree 2)**	**Clause 1 = 0.060** **Clause 2 = 0.026** **Clause 3 = 0.320**
Eventuality	**31.5 (Degree 4)**	**Clause 1 = 0.070** **Clause 2 = 0.025** **Clause 3 = 0.636**

Table 4.2: *ODD RO Inevitability Verification Time*

Certificate	YALMIP-SOS Time(Sec)	REDLOG-QE Time(Sec)
Attractive Invariants	**6127.6 (Degree 10)**	**Clause 1 =5.24** **Clause 2 =0.33** **Clause 3 =1.56**
Escape	**320.6757 (Degree 4)**	**Clause 1 = 0.01** **Clause 2 = 0.30** **Clause 3 = 2.50**
Eventuality	**4128.8 (Degree 6)**	**Clause 1 = 0.349** **Clause 2 = 0.300** **Clause 3 = 0.615**
Lyapunov	**55.24(Degree 4)**	**Clause 1 = 0.02** **Clause 2 = 0.75** **Clause 3 = 0.57**

Table 4.3: *Even RO Inevitability Verification Time*

Results show the effectiveness of our approach to verify the complex inevitability property of a real world AMS circuit. Though it needs user input, formalizing the problem as SOS and then as QE, our approach offers a comparable computation time to [109]. It is in fact less by an order of at least half when considering their approach using partitioning of the state space in to boxes. We have proved the inevitability property avoiding hundreds of reachability computations as was done in previous approaches. Secondly, our approach is less conservative compared to other reach set computation methods, where ODEs are explicitly solved and trajectories are conservatively approximated. Our certificate based deductive method is applicable to infinite horizon (as opposed to bounded) and avoids approximating solutions of the differential equations. SOS based relaxation, in addition to solving the NP-hard problem of positivity check, offers an easy way of incorporating parameter variations as well. Given the existing state of the art QE tools, we have further provided formal proofs of these numerically

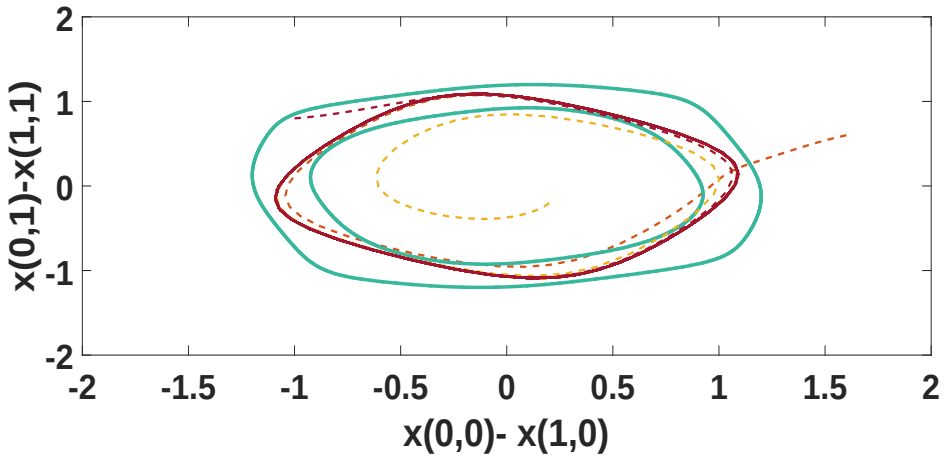


Figure 4.5: *Even RO: Attractive Invariant Set, defined by* $\{V = 1\}$*: Outer Solid plot,* $\{V = r\}$*: Inner Solid plot, Trajectories: Dashed*

calculated certificates. We believe that in future, foreseeing advancements in the QE of FOF over non-linear polynomials, the accuracy and efficiency of our methodology can be furthered.

4.4 Related Work

In [43], the authors attempted to show start-up by finding the DC equilibria and showing its instability. Their approach did not take in to consideration convergence to the limit cycle. A SAT based approach has been used in [57]. They showed stable DC equilibria using a crude approach which can not be generalized. A particle visualization approach has been discussed in [87]. Though the approach could show a broad range of circuit behaviour, it however can easily neglect the failure set from where the circuit fails to start. In [86], the authors showed a model checking approach, where the state space has been discretized, and temporal properties have been verified for the discrete transition system. The most comprehensive work on an even stage RO has been presented in [109]. In this work, the authors showed convergence to the oscillation with probability one. They showed zero measure probability for the failure set using a cone argument. They further showed convergence to the desired limit cycle using reachability analysis. While the approach is comprehensive, it has two disadvantages. They used

an expensive paper-pencil argument about the zero measure probability of the failure set. Secondly, they used approximate but sound reachability computations, which apart from being of a bounded time nature, need partitions of the state space and are thus intractable for higher dimensional systems. A SOS-QE approach has also been used for non-linear gain analysis in [51]. In [47], the author used SOS in a HOL theorem prover to verify positivity of polynomials which are universally quantified. In [85], the authors used a SOS-QE approach for stability analysis of the switched hybrid system. Inevitability of an invariant set is closely related to the global asymptotic stability of a dynamical system. [104] used QE for stability of equilibrium point analysis of non-linear differential equations. In [92], the author introduced various convex/non-convex programs for stability of equilibrium and other invariant sets in non-linear dynamical systems using SOS programming. In the last decade, SOS programming has been the major tool used in the algorithmic construction of Lyapunov certificates for continuous as well as hybrid systems [77]. Deductive verification of continuous and hybrid systems has been demonstrated in [91], [90].

4.5 Summary of the Chapter

We have presented a scalable deductive verification methodology for the inevitability verification of the RO. We have benefited from Lyapunov-like certificates, from non-linear continuous dynamical systems theory, and have come up with some interesting local properties. By verifying these local properties using a combination of SOS-QE, we have successfully verified the global inevitability property of an RO with two different topologies. Experimental results show the effectiveness of our approach avoiding expensive discretization and reach set computations.

Chapter 5

Verifying Frequency Domain Properties of Oscillators using SMODE

A non-linear oscillator can have multiple limit cycles with different frequencies. It is of great importance to verify that an oscillator oscillates with the desired frequency and does not have undesired harmonics. In this chapter, we introduce a robust frequency domain properties specification for the behaviour of the oscillator in close proximity to the limit cycle. We make use of the frequency domain periodogram, and specify the behaviour of the oscillator such that it oscillates with the desired frequency despite parameter and process variations. Towards this goal, we use a robust periodogram specification allowing lower and upper bounds on the sum of squares of Fourier series coefficients for the desired limit cycle.

The verification of frequency domain properties is not straightforward. There are two options to perform this task. One is to carry out the verification in the frequency domain by having both the system and properties in the frequency domain and performing the decision procedure in this domain. This is beyond the capabilities of the current state of the art solvers/approaches. Therefore, we employ a mixed time-frequency domain technique, where we have our properties specified in the frequency domain and we carry out the verification task in the time domain. This approach is illustrated in Fig. 5.1. As shown, frequency domain properties are verified by checking the distance of the timed traces of the oscillator model from the traces, generated from the frequency domain properties. If this distance is less than a user defined small number, we conclude

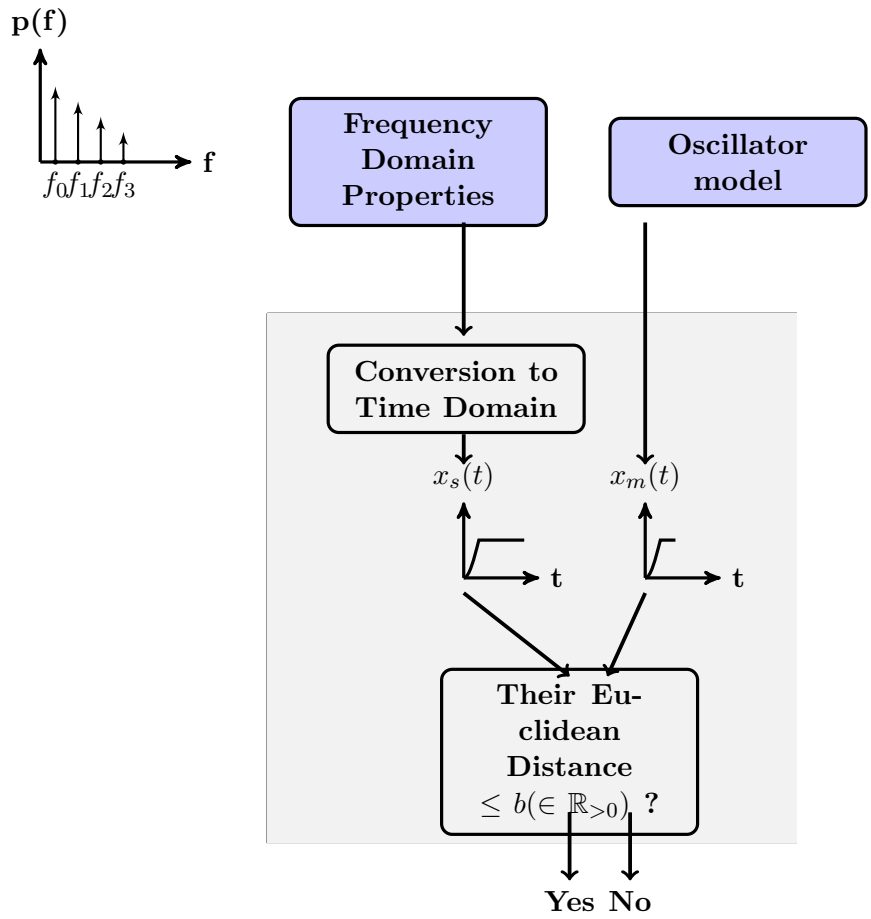


Figure 5.1: *Frequency Domain Property Verification*

satisfaction of the frequency domain with a degree of robustness and vice versa.

5.1 Preliminaries

This section discusses the mathematical modelling of analog oscillators at the device level. Furthermore, we give background of the frequency domain concept used for the frequency domain properties specification.

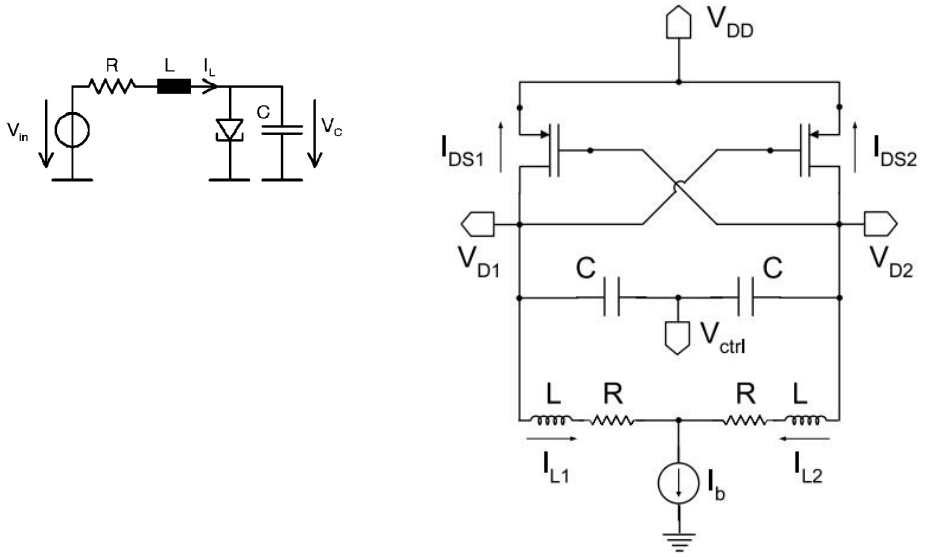


Figure 5.2: *Oscillators Circuit Diagrams, Left: TDO, Right: VCO*

5.1.1 Modelling of analog oscillators as HDS

In this chapter, we model analog oscillators as HDS, as discussed in Sec. 2.1.2. The HDS model of an oscillator is a tuple, $\mathcal{H} = (\mathcal{C}, \mathcal{F}, \mathcal{D}, \mathcal{G})$. Let us denote by x, the vector of continuous variables. In this chapter, we consider two types of oscillators: Voltage Controlled Oscillator (VCO) and Tunnel Diode Oscillator (TDO) as shown in Fig. 5.2. To find $\mathcal{C}$, $\mathcal{F}$, $\mathcal{D}$, and $\mathcal{G}$, we model these oscillators at the device level by treating their input-output responses as piece-wise polynomials.

The VCO consists of an LC tank energised by currents through two PMOS transistors. The non-linearity in the circuit is predominately caused by the non-linear PMOS transistors, and non-linearities due to capacitors and inductors are assumed negligible. To get a HDS model of the VCO, we model the PMOS transistor using the Schichman-Hodges PMOS model [72] given in Eq. 2.9. This model describes the current $I_{DS}(V_{GS}, V_{DS})$ through each PMOS as a function of drain-to-source and gate-to-source voltages. This is a piece-wise polynomial model of the current spanning three different regions of transistor operation: cut off, active, and saturation. The state variables are, V_{D1}, V_{D2}, and I_{L1} or I_{L2}. To get the HDS for VCO, we have nine possible combinations of the three regions of the two PMOS transistors. Let us denote these three regions for each transistor by $C1(C2)$, $L1(L2)$, $S1(S2)$, where C stands for cut off, L for linear and

S for saturation region respectively. We denote $V_{GS} = V_{D2} - V_{DD},\ V_{DG} = V_{D1} - V_{D2}$ for transistor 1, and $V_{GS} = V_{D1} - VDD,\ V_{DG} = V_{D2} - V_{D1}$ for transistor 2 respectively. Accordingly, following are the nine possible regions in the (V_{D1}, V_{D2}) plane based on the I_{DS} model of current through each transistor.

$$C1/C2 = V_{D2} - V_{DD} > V_{tp} \ \wedge V_{D1} - V_{DD} > V_{tp}$$

$$S1/C2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} \leq -V_{tp} \ \wedge V_{D1} - V_{DD} > V_{tp}$$

$$L1/C2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} > -V_{tp} \ \wedge V_{D1} - V_{DD} > V_{tp}$$

$$C1/S2 = V_{D2} - V_{DD} > V_{tp} \wedge V_{D1} - V_{DD} \leq V_{tp} \ \wedge V_{D2} - V_{D1} \leq -V_{tp}$$

$$C1/L2 = V_{D2} - V_{DD} > V_{tp} \wedge V_{D1} - V_{DD} \leq V_{tp} \ \wedge V_{D2} - V_{D1} > -V_{tp}$$

$$S1/L2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} \leq -V_{tp} \ \wedge V_{D1} - V_{DD} \leq V_{tp} \wedge V_{D2} - V_{D1} > -V_{tp}$$

$$S1/S2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} \leq -V_{tp} \ \wedge V_{D1} - V_{DD} \leq V_{tp} \wedge V_{D2} - V_{D1} \leq -V_{tp}$$

$$L1/S2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} > -V_{tp} \ \wedge V_{D1} - V_{DD} \leq V_{tp} \wedge V_{D2} - V_{D1} \leq -V_{tp}$$

$$L1/L2 = V_{D2} - V_{DD} \leq V_{tp} \wedge V_{D1} - V_{D2} > -V_{tp} \ \wedge V_{D1} - V_{DD} \leq V_{tp} \wedge V_{D2} - V_{D1} > -V_{tp}$$

It can easily be shown that the $L1/L2$ mode is infeasible, and we are left with eight possible combinations of the two transistor operating modes. The region in the (V_{D1}, V_{D2}) plane corresponding to different operating modes of the two transistors is shown in

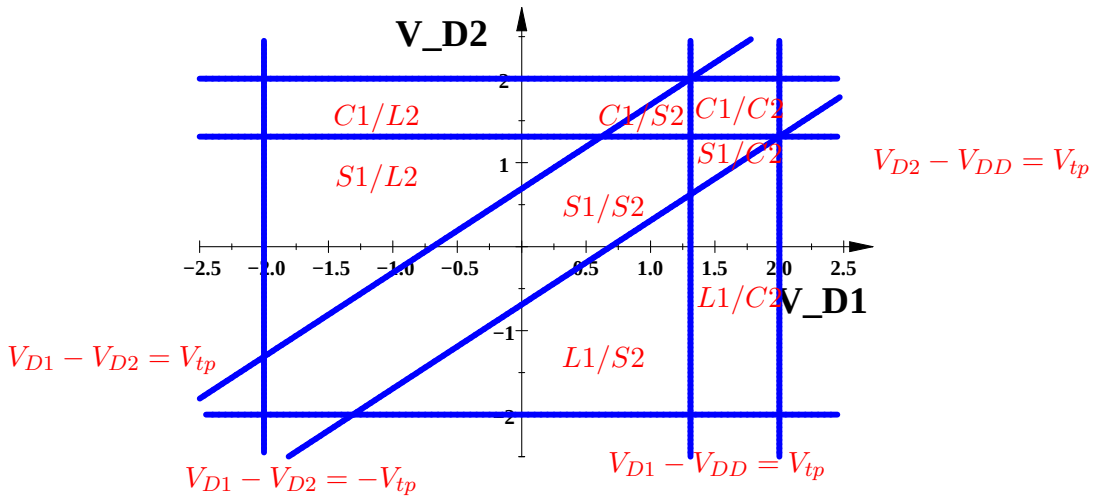


Figure 5.3: *VCO eight possible modes of operation*

Fig. 5.3.

Applying KVL to the VCO circuit, we get the following ODE equations for each state V_{D1}, V_{D2} and I_{L1}.

$$\dot{V}_{D1} = \frac{-1}{C}(I_{DS1}(V_{D2} - V_{DD}, V_{D1} - V_{DD}) + I_{L1}) \tag{5.1}$$

$$\dot{V}_{D2} = \frac{-1}{C}(I_{DS2}(V_{D1} - V_{DD}, V_{D2} - V_{DD}) + I_b - I_{L1}) \tag{5.2}$$

$$\dot{I}_{L1} = \frac{1}{2L}(V_{D1} - V_{D2} - R(2I_{L1} - I_b)) \tag{5.3}$$

Here currents I_{DS1} and I_{DS2} are piece-wise polynomials depending on which eight modes the two transistors operate in. Accordingly, the HDS $\mathcal{H}$ of the VCO based on transistor currents I_{DS1} and I_{DS2} is,

$$\mathcal{H} = \begin{cases} \begin{pmatrix} \dot{V}_{D1} \\ V_{D2} \\ I_{L1} \end{pmatrix} = \begin{pmatrix} \frac{-1}{C}(I_{DS1}(V_{D2} - V_{DD}, V_{D1} - V_{DD}) + I_{L1}) \\ \frac{-1}{C}(I_{DS2}(V_{D1} - V_{DD}, V_{D2} - V_{DD}) + I_b - I_{L1}) \\ \frac{1}{2L}(V_{D1} - V_{D2} - R(2I_{L1} - I_b)) \end{pmatrix} & x \in \mathcal{C}, \\ x^+ = G_i(x) & x \in \mathcal{D} \end{cases} \tag{5.4}$$

Here,

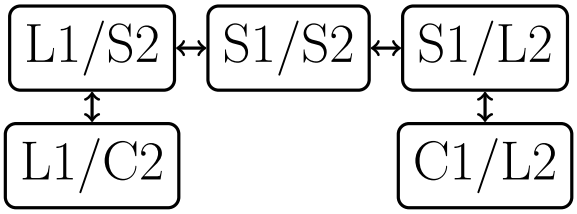


Figure 5.4: *VCO Periodic Limit Cycle*

$$\mathcal{F} = \begin{pmatrix} \frac{-1}{C}(I_{DS1}(V_{D2} - V_{DD}, V_{D1} - V_{DD}) + I_{L1}) \\ \frac{-1}{C}(I_{DS2}(V_{D1} - V_{DD}, V_{D2} - V_{DD}) + I_b - I_{L1}) \\ \frac{1}{2L}(V_{D1} - V_{D2} - R(2I_{L1} - I_b)) \end{pmatrix},$$

$$F_i(x,u) = \begin{pmatrix} \frac{-1}{C}(I^i_{DS1}(V_{D2} - V_{DD}, V_{D1} - V_{DD}) + I_{L1}) \\ \frac{-1}{C}(I^i_{DS2}(V_{D1} - V_{DD}, V_{D2} - V_{DD}) + I_b - I_{L1}) \\ \frac{1}{2L}(V_{D1} - V_{D2} - R(2I_{L1} - I_b)) \end{pmatrix}, \ \forall i \in \{1, .., 8\},$$

$$\mathcal{C} = \left\{ [V_{D1} \ V_{D2} \ I_{L1}] \ \middle| \ -2 \leq V_{D1} \leq 2 \text{ and } -2 \leq V_{D1} \leq 2 \text{ and } -1 \leq I_{L1} \leq 1 \right\},$$

$\mathcal{D} = \Big\{ [V_{D1} \ V_{D2} \ I_{L1}] \ \Big| \ V_{D2} - V_{DD} = V_{tp}$ and $V_{D1} - V_{DD} = V_{tp}$ and $V_{D1} - V_{D2} = -V_{tp}$ and $V_{D2} - V_{D1} = -V_{tp} \Big\}$, $G_i(x) := x, \ \forall x \in \mathcal{D}, \ \forall i \in I_D$. I^i_{DS1} and I^i_{DS2} are as given by Eq. 2.9.

Note that we have identity reset maps for all jumps and we have not shown every jump map explicitly. As will be shown in the next section, the limit cycle of the VCO periodically visits only five modes (regions) of the (V_{D1}, V_{D2}) plane, and therefore only these five modes are considered for frequency domain properties verification. The HDS formed by these five modes is shown in Fig. 5.4.

Similarly, we model the TDO as HDS using the piece-wise polynomial model of the tunnel diode [38]. This model represents the current I_d through the tunnel diode as a

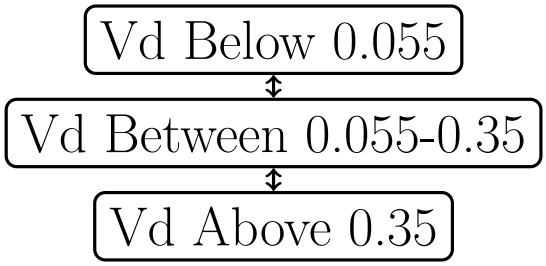


Figure 5.5: *TDO Hybrid Automata*

function of voltage V_d across it given in Eq. 5.5.

$$I_d = \begin{cases} 6.01V_d^3 - 0.992V_d^2 + 0.0545V_d & V_d \leq 0.055, \\ 0.0692V_d^3 - 0.0421V_d^2 + 0.004V_d + 8.96.10^{-4} & 0.055 \leq V_d \leq 0.35, \\ 0.263V_d^3 - 0.277V_d^2 + 0.0968V_d - 0.0112 & 0.35 \leq V_d. \end{cases} \tag{5.5}$$

Therefore, the HDS for the TDO has three modes of operation as shown in Fig. 5.5. Considering V_d and I_L as state variables and applying KVL to the TDO circuit (Fig. 5.2), we get the following ODEs for the circuit,

$$\dot{V_d} = \frac{1}{C}(-I_d(V_d) + I_L) \tag{5.6}$$

$$\dot{I_L} = \frac{1}{L}(-V_d + I_L.R + V_{in}) \tag{5.7}$$

Therefore,

$$\mathcal{H} = \begin{cases} \begin{pmatrix} \dot{V_d} \\ \dot{I_L} \end{pmatrix} = \begin{pmatrix} \frac{1}{C}(-I_d(V_d) + I_L) \\ \frac{1}{L}(-V_d + I_L.R + V_{in}) \end{pmatrix} & x \in \mathcal{C}, \\ x^+ = G_i(x) & x \in \mathcal{D} \end{cases} \tag{5.8}$$

Here,

$$\mathcal{F} = \begin{pmatrix} \frac{1}{C}(-I_d(V_d) + I_L) \\ \frac{1}{L}(-V_d + I_L.R + V_{in}) \end{pmatrix},$$

$$F_1(x,u) = \begin{pmatrix} \frac{1}{C}(-(6.01V_d^3 - 0.992V_d^2 + 0.0545V_d) + I_L) \\ \frac{1}{L}(-V_d + I_L.R + V_{in}) \end{pmatrix},$$

$$F_2(x,u) = \begin{pmatrix} \frac{1}{C}(-(0.0692V_d^3 - 0.0421V_d^2 + 0.004V_d + 8.96.10^{-4}) + I_L) \\ \frac{1}{L}(-V_d + I_L.R + V_{in}) \end{pmatrix},$$

$$F_3(x,u) = \begin{pmatrix} \frac{1}{C}(-(0.263V_d^3 - 0.277V_d^2 + 0.0968V_d - 0.0112) + I_L) \\ \frac{1}{L}(-V_d + I_L.R + V_{in}) \end{pmatrix}$$

$$\mathcal{C} = \left\{ [V_d \ I_L] \;\middle|\; 0 \leq V_d \leq 1 \text{ and } -10 \leq I_L \leq 10 \right\},$$

$$\mathcal{D} = \left\{ [V_d \ I_L] \;\middle|\; V_d = 0.055 \text{ and } V_d = 0.35 \right\},\ G_i(x) := x,\ \forall x \in \mathcal{D},\ \forall i \in I_D.$$

Similarly, C_i can be easily found.

Let us define the flow map $\Psi_{\mathcal{H}} : \mathcal{T} \times \mathcal{X}_{\mathcal{H}} \to \mathcal{X}_{\mathcal{H}}$. We now define hybrid limit sets and hybrid limit cycles.

Definition 5.1 (Hybrid Limit Sets)**.** *A point $z \in \mathcal{X}_{\mathcal{H}}$ is called an Ω-limit point of $y \in \mathcal{X}_{\mathcal{H}}$ if, $lim_{(t+j)\to\infty}\Psi_{\mathcal{H}}((t,j),y) = z$. The set of all such points z is the hybrid Ω-limit set.*

Definition 5.2 (Hybrid periodic Orbits)**.** *An orbit Γ is a closed hybrid periodic orbit if it is not an equilibrium, and for some j and t $\Psi_{\mathcal{H}}((T,j),x) = x$, for some smallest $T \neq 0$. $T \in [t_j, t_{j+1}]$ is called the fundamental period of Γ.*

Definition 5.3 (Hybrid Limit cycle)**.** *A closed hybrid orbit Γ is called a hybrid limit cycle if, $\forall x : x \in \mathcal{X}_H \setminus \Gamma,\ y \in \Gamma,\ \alpha > 0,\ \lim_{(t+j)\to\infty} \|\Psi_{\mathcal{H}}((t,j),x) - y\| \leq \alpha$.*

Lemma 5.1. *If Z is a compact set of states that constitute a periodic orbit and its close proximity, then for any $Y \subset Z$, we have $Q = \bigcup_k Reach(Y)$, such that for an arbitrary point $x \in Q$, and $y \in \Gamma$, $\alpha > 0$, either $\lim_{(t+j)\to\infty} \|\Psi_{\mathcal{H}}((t,j),x) - y\| \leq \alpha$, or $\lim_{(t+j)\to b} \|\Psi_{\mathcal{H}}((t,j),x) - y\| = 0$, for an arbitrary small $b > 0$.*

Proof. Clearly for $x \in Q$, if also $x \in \Gamma$, then $\lim_{(t+j)\to b} \|\Psi_{\mathcal{H}}((t,j),x) - y\| = 0$, $b > 0$ as every state on Γ is reachable from every other state on it. For $x \in Q$, and $x \notin \Gamma$, $\lim_{(t+j)\to\infty} \|\Psi_{\mathcal{H}}((t,j),x) - y\| \leq \alpha$, since a stable Γ attracts every trajectory to its close proximity. □

5.2 Frequency Domain Properties Specification of the Hybrid Limit Cycle

This section introduces the robust frequency domain properties specification of the hybrid limit cycle, using a periodogram-based power spectral envelope.

5.2.1 Robust Specification of a Periodic Function in the Frequency Domain

A function g is periodic with period T if $g(t) = g(t + mT), \forall t : t \in \mathbb{R}$ and $\forall m : m \in \mathbb{Z}$. We denote by $\mathcal{P}$ the set of all functions which in addition to being T periodic, also have the property of square sumability over a period T, i.e., $\mathcal{P} \subset L^2[0,T]$. All such periodic functions $g(t) \in \mathcal{P}$ can be represented by the sum of an infinite number of T-periodic sinusoids as,

$$g(t) = \sum_{k=0}^{\infty} (a_k \cos \omega_k t + b_k \sin \omega_k t) \tag{5.9}$$

where $\omega_k = 2\pi k/T$, $a_k, b_k \in \mathbb{R}$. Instead of an infinite series representation of exact periodic functions, we use the notion of almost periodic functions [62]. These are the functions which are represented by at most a countable number of sinusoids. We denote such sets of almost periodic functions by $\mathcal{AP}$, and therefore $g(t) \in \mathcal{P}$ is represented by its approximation $S_k(t) \in \mathcal{AP}$,

$$S_K(t) = \sum_{\omega_k \in \Omega_K} (a_k \cos \omega_k t + b_k \sin \omega_k t), \ k \in \{1, ...K\}. \tag{5.10}$$

where Ω_K is the set of $K \in \mathbb{N}$ frequencies. The finite series representation $S_K(t)$ is the best approximation of $g(t)$, and it has a least mean square error property. Let $\varepsilon_K = \max(\|g(t) - S_K(t)\|)$ represent the maximum approximation error, then $g(t)$ can be conservatively represented by

$$S_K(t) - \varepsilon_K \leq g(t) \leq S_K(t) + \varepsilon_K \tag{5.11}$$

Let us $\mathcal{F} = \{(a_0, b_0), ...(a_k, b_k)\}$, the set of all $k+1$ pairs of Fourier coefficients. This set $\mathcal{F}$ is called the frequency domain representation of an almost periodic function $S_K(t)$. Instead of specifying a periodic function $S_K(t)$ in the frequency domain in terms of the set $\mathcal{F}$, we use the periodogram specification which is defined below.

Definition 5.4 (Periodogram)**.** *The energy content of a signal at each frequency ω_k is called a periodogram, and is given by $\mathsf{p}_k = (a_k^2 + b_k^2)$, $\mathsf{p}_k \in \mathbb{R}_{\geq 0}$. We denote by $P = \{\mathsf{p}_0, ...\mathsf{p}_K\}$, the set of all periodograms at frequencies $\omega_k \in \Omega_K$.*

To cater for parameter variations, temperature and uncertainty in initial conditions, we introduce the idea of robust periodogram specification.

Definition 5.5 (Robustness of Periodogram)**.** *We specify P such that pairs of the Fourier series coefficients (a_k, b_k) $\forall k \in \{1, ...K\}$, for all $\omega_k \in \Omega_K$, result in the function $S_K(t)$ (Eq. 5.10) which is the approximate representation of the periodic function $g(t)$, and satisfies the inequality constraint of Eq. 5.11. We say that $\mathsf{p}_k \in P$ has ϵ_k degree of robustness, if it can tolerate an ϵ_k amount of perturbation such that, $\exists \mathsf{p}'_k : ||\mathsf{p}_k - \mathsf{p}'_k|| \leq \epsilon_k$ (Fig. 5.6), without altering the validity of the condition in Eq. 5.11.*

5.2.2 Encoding Membership of the Limit Cycle in the Robust Power Spectral Envelope

Let there exists a periodic hybrid arc $P_{\mathcal{H}} : \mathcal{T} \to \mathcal{X}_{\mathcal{H}}$. Let us define a power spectral envelope, $H(\omega_k) : \Omega_K \to \mathbb{R}_{\geq 0}$, which maps each discrete frequency $\omega_k \in \Omega_K$ to a periodogram p_k for all $k \in \{1, .., K\}$. The set $A\mathcal{P}_{\epsilon_k}$ of almost periodic functions belongs to the power spectral envelope $H(\omega_k)$ with ϵ_k degree of robustness, if the Fourier series coefficients of Eq. 5.10, representing the set $A\mathcal{P}_{\epsilon_k}$ in the frequency domain, satisfy the following constraints [18],

- for $k > K$, $(\omega_k > \omega_K) \implies \mathsf{p}_k = 0$,
- $\forall k : k \in \{1, .., K\}$, $H(\omega_k) - \epsilon_k \leq \mathsf{p}_k \leq H(\omega_k) + \epsilon_k$, such that $0 \leq \omega_k \leq \omega_K$.

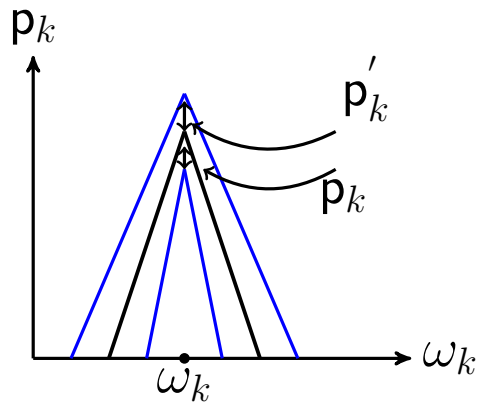


Figure 5.6: *Robust Periodogram Specification*

We require that for $S_K(t) \in cl(A\mathcal{P}_{\epsilon_k})$ the hybrid periodic orbit $P_{\mathcal{H}}$ satisfies the constraint

$$S_K(t) - \varepsilon_K \leq P_{\mathcal{H}}(t,j) \leq S_K(t) + \varepsilon_K \tag{5.12}$$

Here $cl(A\mathcal{P}_{\epsilon_k})$ denotes closure of $A\mathcal{P}_{\epsilon_k}$. We encode this by introducing the following set of constraints for the hybrid periodic arc $P_{\mathcal{H}}$,

$$\psi_1 := \bigwedge_{\ell=1}^{n} \left(\bigwedge_{k=0}^{K} (H_\ell(\omega_k) - \epsilon_k^\ell \leq \mathsf{p}_k^\ell \leq H_\ell(\omega_k) + \epsilon_k^\ell) \right), \tag{5.13}$$

$$\psi_2 := \bigwedge_{\ell=1}^{n} \left(\forall t : t \in [t_{min}, t_{max}] \left(S_K^\ell(t) = \sum_{k=0}^{K} (a_k^\ell \cos \omega_k t + b_k^\ell \sin \omega_k t) \right) \right), \tag{5.14}$$

$$\psi_3 := \bigwedge_{\ell=1}^{n} \left(\forall t : t \in [t_{min}, t_{max}] \left(S_K^\ell(t) - \varepsilon_K^\ell \leq P_{\mathcal{H}}^\ell(t,j) \leq S_K^\ell(t) + \varepsilon_K^\ell \right) \right). \tag{5.15}$$

Here the constraint ψ_1 puts upper and lower bounds on the periodograms at K frequencies in the presence of ϵ_k^ℓ perturbations. Note that here n is the dimension of the system. The second constraint ψ_2 ensures that for all time t, each N periodic scalar variable is approximated by K sinusoids. The last constraint ψ_3 conservatively

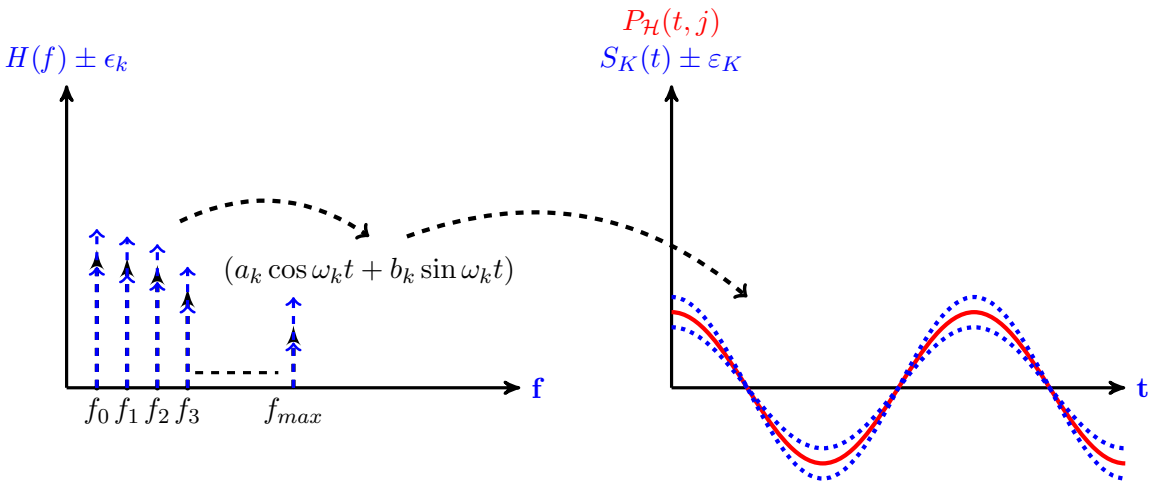


Figure 5.7: *Frequency Domain Specification*

over-approximates the periodic function $P_{\mathcal{H}}(t, j)$ taking in to consideration the error generated by the almost approximate periodic function S_K. This encoding of the frequency domain specification of a periodic hybrid arc is pictorially shown in the Fig. 5.7.

5.3 Verification of the Frequency Domain Properties

The task of verification of the frequency domain properties has two parts. In the first part, we verify the existence of a unique limit cycle in the hybrid state space of the oscillator. The second part verifies membership of the limit cycle in the robust frequency domain envelope.

To verify the frequency domain properties discussed in the last section, we need to identify the location of the limit cycle in the oscillator hybrid state space. This search of the limit cycle is performed utilizing BMC following the Pseudo code given in Alg. 6. In order to show that there is a unique limit cycle in the hybrid state space of the oscillator HDS, we show the presence of a periodic arc in the hybrid state space and show its reachability from all but the equilibrium state. The input of the Alg. 6 is the HDS model of an oscillator whereas its output is the set of all boxes constituting a limit cycle. We divide the hybrid state space in boxes $[x_l, x_u]_1 \times [y_l, y_u]_1, ..., [x_l, x_u]_n \times [y_l, y_u]_n$. The search starts from a box with the equilibrium state x_e and goes radially outward as

Algorithm 6 Locating Limit Cycle

INPUT: : HDS model of Oscillator
OUTPUT: : Hybrid Limit Cycle

$B_{periodic} \leftarrow \emptyset$
$\mathbb{BR}_{limit} \leftarrow \emptyset$
$\Gamma \leftarrow \emptyset$
for $i = 0 \rightarrow i = i_{max}$ **do**
  $B_{initial} \leftarrow [x_l, x_u]_i \times [y_l, y_u]_i$
  repeat
    $B_{next} \leftarrow \text{Reach}_t^{\mathcal{H}}(B_{initial})$
  until $(B_{next} \cap B_{initial} \neq \emptyset \wedge t > 0) \vee (t = t_{max})$
  if $(B_{next} \cap B_{initial} \neq \emptyset$ **then**
    $B_{periodic} \leftarrow B_{initial}$
  else
    $B_{periodic} \leftarrow \emptyset$
  end if
  if $B_{periodic} \neq \emptyset$ **then**
    break
  end if
end for
if $B_{periodic} \neq \emptyset$ **then**
  repeat
    $B_{limit_{next}} \leftarrow Reach_t^{\text{H}}(B_{periodic})$
    $\mathbb{BR}_{limit} \leftarrow \{B_{periodic}, B_{limit_{next}}\}$
  until $B_{limit_{next}} \in \mathbb{BR}_{limit}$
  for $k = 0 \rightarrow k = k_{max}$ **do**
    repeat
      $B_{next} \leftarrow \text{Reach}_t^{\text{H}}(B_k)$; $B_k \notin \mathbb{BR}_{limit}$
    until $(d(y \in B_{next}, z \in (B \in \mathbb{BR}_{limit})) \leq \epsilon) \vee (t = t_{max})$
  end for
end if
$\Gamma \leftarrow \mathbb{BR}_{limit}$
return Γ

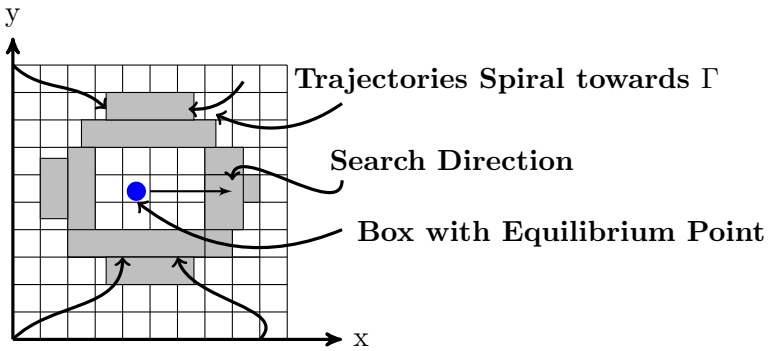


Figure 5.8: *Locating the Global Positive Limit Cycle*

depicted in Fig. 5.8. An arbitrary small box which does not contain an equilibrium point, and is part of a closed orbit, should satisfy $\Psi_{\mathcal{H}}((t,j),x) = x$ for some $x \in ([x_l, x_u]_n \times [y_l, y_u]_n)$ and $t > 0$. Following this strategy, we start in $B_{initial}$ and check if the reachable sets, $\text{Reach}_t^H(B_{initial})$, have a non-empty intersection with $B_{initial}$ for $t > 0$ (Lines 4-8). If there is a box $B_{periodic}$ with the periodicity property, the search is stopped (Lines 9-15), otherwise all i_{max} boxes are searched for periodicity for a maximum time t_{max} (Line 4-16). Utilizing Lemma 5.1, the algorithm finds all boxes that are reachable from $B_{periodic}$ through a series of reachability operations (Lines 18-22). This results in the set of all boxes $\mathbb{BR}_{limit}$, which contains set of states constituting a T periodic closed orbit and it's close proximity. From Lemma 5.1, the set of boxes $\mathbb{BR}_{limit}$ contains all states that are either part of the periodic limit cycle, or are in close proximity of it. To show that $\mathbb{BR}_{limit}$ is the global positive limit set, we further show that those trajectories that start in the set of boxes $(\mathbb{BR} \setminus \mathbb{BR}_{limit})$ converge to $\mathbb{BR}_{limit}$ for $t \to \infty$. Here, $\mathbb{BR}$ denotes the set of all boxes. This can formally be verified by showing that,

$$lim_{t \to m} d(\Psi_{\mathcal{H}}(\mathcal{T}, x), z) \leq \beta$$

$\forall x \in \mathbb{BR} \setminus \mathbb{BR}_{limit}$, and $\exists z \in B(\in \mathbb{BR}_{limit})$. Here d is the distance from the hybrid trajectory $\Psi_{\mathcal{H}}(\mathcal{T}, x)$ to the nearest point $z \in B(\in \mathbb{BR}_{limit})$. This is performed by the reachability of a box in close proximity of $\mathbb{BR}_{limit}$, from all non-periodic boxes B_k (Lines 24-27). This procedure is repeated for all k_{max} non-periodic boxes. Note that this reachability is not performed for the equilibrium state as the limit cycle is non-

reachable from it. Note that to verify the frequency domain properties, we need to have a single box with the periodicity property and which is part of the unique limit cycle Γ. However, to reduce the time of the reachability queries for all outside boxes, we need to find all boxes with the periodicity property so that to be able to use the closest box of a limit cycle as a target box.

To verify that the unique limit cycle Γ satisfies frequency domain properties of Eq. 5.13, Eq. 5.14, and Eq. 5.15, the euclidean distance of Γ from the specified periodic arc $P_{\mathcal{H}}(t, j)$, must be less than an arbitrary small positive number σ. This is performed by introducing the following constraint,

$$\|P_{\mathcal{H}}(t,j) - \Psi_{\mathcal{H}}(\mathcal{T}, x)\| \leq \sigma,\ x \in B(\in \mathbb{BR}_{limit}) \tag{5.16}$$

5.3.1 Encoding the Frequency Domain Properties Verification as a BMC Problem

We use SMODE, discussed in Sec. 2.5.1.1, to implement the reachability instances in Alg. 6. Predicative encoding is used to define the state space by bounding variables $x := \{x_1, ...x_n\}$, such that $x \in ([x_L, x_U] \times [y_L, y_U])$. At step 0, of the K unwindings of the transition system, and for an initial $i \in I_C$, x can be set to be in an initial box, i.e. $(i \in I_C) \implies x \in B_{initial}$. Each C_i is defined by a set $\{c_1, ...c_n\}$ of invariants on the real variables and this can be added as a constraint $C_i \implies \bigwedge_n c_n,\ \forall i \in I_C$. In a flow set C_i when time elapses, the real variables x_n update according to the set of flow maps $F_i := \{f_1, ...f_n\}$ for n variables respectively. For each C_i we add the constraint $(C_i \wedge t_{elapse}) \implies \bigwedge_n f_n$. For each jump set, $D_i,\ i \in I_D$, $D_i := \{d_1, ...d_n\}$, the jump from C_j to C_{j+1} is encoded as a predicate, $jump\wedge(C_j, C_{j+1}) \implies \bigwedge_n d_n$. For each jump set, we add identity jump maps, i.e. $jump\wedge(C_j, C_{j+1}) \implies \bigwedge_n g_n^{j+1}(x_n) = x_n^j$. The first property that we verify is the existence of periodicity for a certain initial box $B_{initial}$. We encode this as a target predicate, $time > 0 \wedge x \notin B_{initial}$. This property essentially states that, starting in $B_{initial}$ and after some time, trajectories traversed by the continuous valuation of variable x, do not return to the box $B_{initial}$. A counterexample of this formula shows that, $time > 0 \wedge x \in B_{initial}$ is true. Consequently, this shows that trajectories starting in $B_{initial}$ are periodic. All other reachability queries are encoded similarly. This results in a constraint system Π consisting of the following constraints,

- $x := \{x_1, ...x_n\} \in [x_L, x_U] \times [y_L, y_U]$
- $C_i \implies x \in B_{initial}$

- $C_i \implies \bigwedge_n c_n$
- $(C_i \wedge t_{elapse}) \implies \bigwedge_n f_n$
- $jump \wedge (C_j, C_{j+1}) \implies \bigwedge_n d_n$
- $jump \wedge (C_j, C_{j+1}) \implies \bigwedge_n g_n^{j+1}(x_n) = x_n^j$
- $time > 0 \wedge x \notin B_{initial}$

The decision of the BMC problem to check periodicity/reachability is reduced to the satisfaction of the Π constraint system.

Similarly, we determine membership of the periodic hybrid arcs of the oscillator HDS, in the robust power spectral envelope, by incorporating the additional set of constraints ψ_1, ψ_2, ψ_3, in a transition system similar to Π. The initial conditions of the BMC is given in the form of a box $B_{initial} \in \mathbb{BR}_{limit}$. Apart from the ODE constraints, we add the set of constraints $\psi_1, \psi_2, , \psi_3$ for each scalar variable x_n to the BMC algorithm. As a 'Target' state of the BMC, we introduce the following predicate,

$$\neg(time > 0 \wedge time <= t_{max} \wedge x_n \in B_{initial}) \vee \neg(\|P_{\mathcal{H}}^n(t,j) - x_n(t,j)\| \leq \sigma) \qquad (5.17)$$

This target predicate is a disjunction of two predicates. The predicate, $\neg(time > 0 \wedge time <= t_{max} \wedge x_n \in B_{initial})$, ensures that starting in the box $B_{initial}$, trajectories will return to the same box before the maximum time limit is elapsed. A satisfiable valuation of this predicate is a counterexample of the periodicity property. The second predicate, $\neg(\|P_{\mathcal{H}}^n(t,j) - x_n(t,j)\| \leq \sigma)$, ensures that for all time, the distance of the hybrid arc $x(t,j)$ from the specified time domain periodic hybrid arc $P_{\mathcal{H}}^n(t,j)$, obtained from the frequency domain specification, must be less than an arbitrary small positive number σ. A satisfiable valuation of this predicate indicates the violation of the frequency domain specification implicitly.

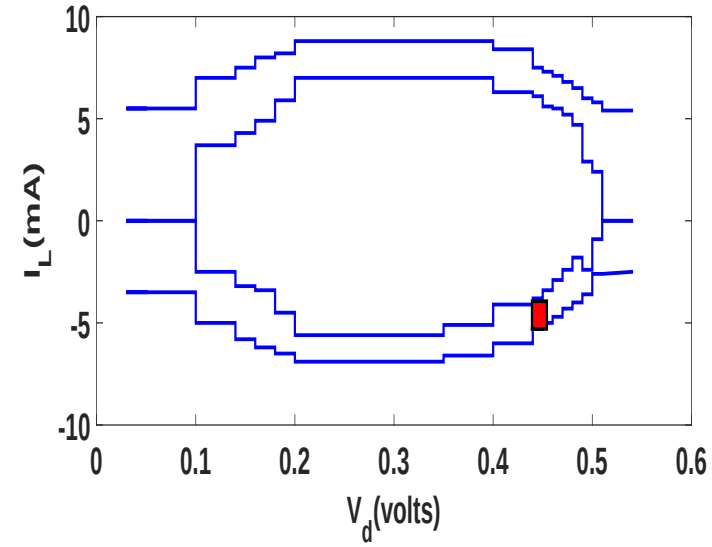
5.4 Experimental Evaluation

We have verified frequency domain properties for TDO and VCO benchmarks. The list of parameters with their ranges are given in the Table 5.1 [37]. We used the SMODE solver, iSAT-ODE [33], for the verification of BMC-based reachability/periodicity and frequency domain properties. We used Matlab to compute periodogram specifications for both benchmarks [71]. We used a 2.6 GHZ Intel(R) Core(TM) i5 machine with 4 GB of memory for all experiments.

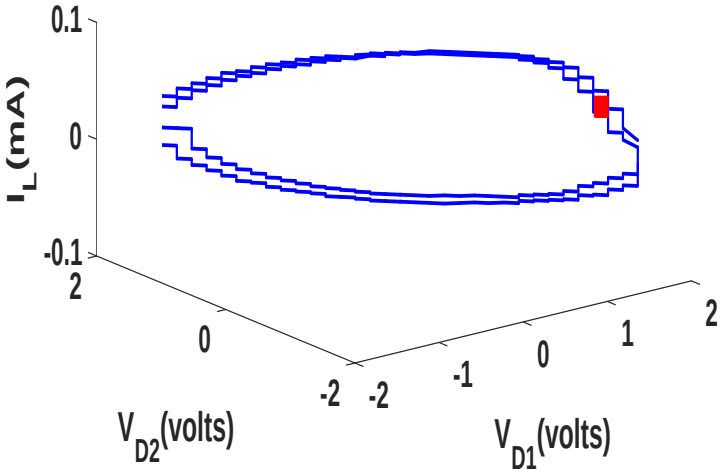
Parameters	VCO	TDO
C	$3.43nF \pm 2\%$	
L	$2.85mH \pm 2\%$	
V_{ctr}	**0**	
R	$3.7\Omega \pm 2\%$	
V_{tp}	-0.69	
K_p	$86\mu A/V^2$	
W	$240\mu m$	
L	$0.25\mu m$	
λ	$-0.07V^{-1}$	
V_{DD}	$1.8V$	
I_b	$18mA$	
C		$1nF \pm 2\%$
L		$1mH \pm 2\%$
R		$0.2\Omega \pm 2\%$

Table 5.1: *Benchmark Oscillator Parameters*

To find all boxes with equilibrium states, we used Z3 SMT solver [50] to find the roots of the right hand side of Eq. 5.6 and Eq. 5.7. The equilibrium point analysis for the TDO shows that there are two equilibrium points, i.e. $x_e \in [0.34, 0.35]$ and $x_e \in [0.35, 0.38]$. Note that these two boxes belong to different continuous flow sets. To determine the location of the limit cycle in the TDO hybrid space, we divided the space ($V_d \in [0, 1], I_L \in [-10, 10]$), into 400 boxes. Starting from the box(es) having equilibrium states, we performed our search in an arbitrary direction keeping the axis I_L fixed and varying V_d iteratively checking each box for the periodicity property. Once we identified such a box, we found all boxes which were reachable from this box. To show that this is the unique limit cycle, we verified that the boxes, being part of the periodic arc, are reachable from all other boxes. This resulted in an isolated limit cycle in the hybrid state space of the TDO HDS as depicted in Fig. 5.9a. Note that, to verify the frequency domain properties, we needed only a single isolated box having periodicity and global reachability properties. However, to reduce the computation time for reachability queries, we found all boxes belonging to the periodic arc and reachability were verified for a nearest box for different regions. The processes of splitting intervals and formulating SMODE properties have been done manually, supported by around 5500 seconds of CPU time. This includes approximately 100 SMODE enquiries for periodicity and approximately 380 reachability properties. The given time is for all those formulas, Periodicity/Reachability, for which a "Satisfiable" result was given by iSAT-



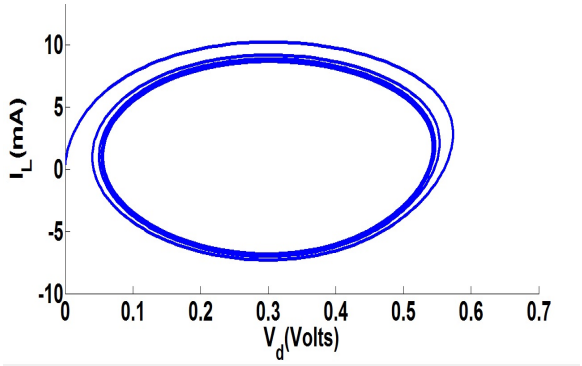
(a) TDO Limit Cycle Simulation



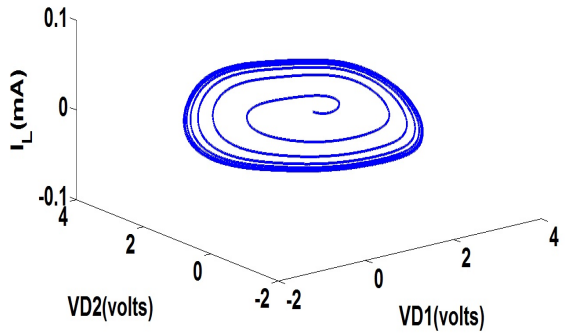
(b) VCO Limit Cycle Simulation

Figure 5.9: *Locating Limit Cycle in Hybrid State Space*

ODE, and does not include the instances when a property was declared "Unsatisfiable" by the solver for a particular box.



(a) TDO Limit Cycle Simulation

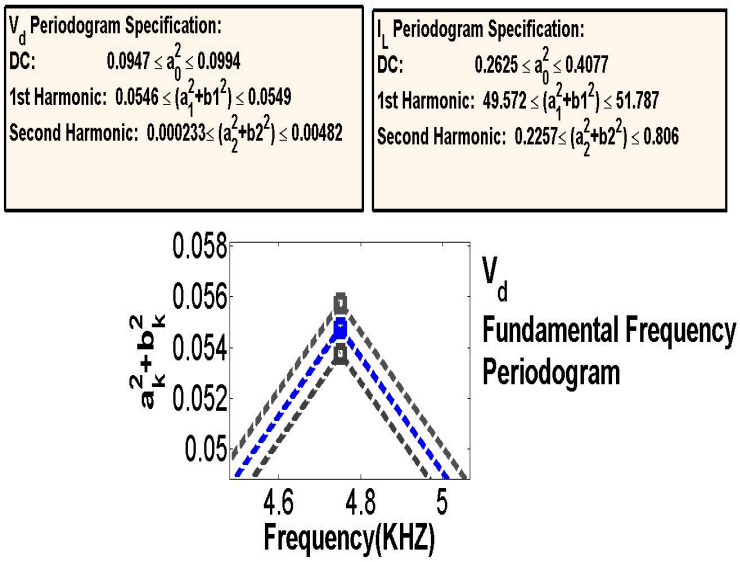


(b) VCO Limit Cycle Simulation

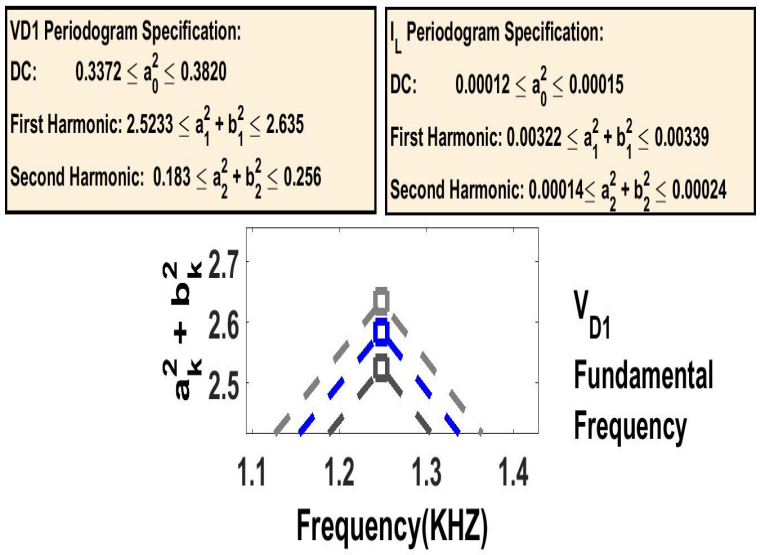
Figure 5.10: *Oscillators Hybrid Systems Simulation Traces*

We have used a similar procedure for locating the isolated limit cycle in the VCO hybrid state space. However, while performing the periodicity and reachability verification of boxes using iSAT-ODE, we needed to transform some of the flow sets, where there were multiple variables, e.g. $V_{D1} - V_{D2} \leq -V_{tp}$, to simpler inequalities involving a single variable. This is because encoding of hybrid systems in iSAT-ODE requires simple bounds on the variables as elaborated by Andreas et al. in [33]. Therefore, for the VCO HDS, we have introduced an auxiliary variable, $V = V_{D1} - V_{D2}$, with its derivative with respect to time t given below,

$$\dot{V} = \frac{\partial V}{\partial V_{D1}} \dot{V_{D1}} + \frac{\partial V}{\partial V_{D2}} \dot{V_{D2}} = \dot{V_{D1}} - \dot{V_{D2}} \tag{5.18}$$



(a) TDO Robust Periodogram Specification



(b) VCO Robust Periodogram Specification

Figure 5.11: *Frequency Domain Properties Specifications*

From the equilibrium point analysis, we have found two boxes with equilibrium in the VCO hybrid state space. These are, $V_{D1} \in [0.62, 0.67]$, $V_{D2} \in [0.62, 0.67]$, $I_L \in [0, 0]$,

Depth	Decision	Time(Seconds)
0	**Unsatisfiable**	**0**
1	**Unsatisfiable**	**81.07**
2	**Unsatisfiable**	**83.22**
3	**Unsatisfiable**	**304.37**
4	**Unsatisfiable**	**352.44**
5	**Unsatisfiable**	**1299.64**
6	**Unsatisfiable**	**1448.71**
7	**Unsatisfiable**	**26779.75**
8	**Unsatisfiable**	**27096.21**

(a) TDO Verification Results

Depth	Decision	Time(Seconds)
0	**Unsatisfiable**	**0**
1	**Unsatisfiable**	**6.13**
2	**Unsatisfiable**	**206.45**
3	**Unsatisfiable**	**538.39**
4	**Unsatisfiable**	**947.10**
5	**Unsatisfiable**	**2237.89**
6	**Unsatisfiable**	**3457.43**
7	**Unsatisfiable**	**11672.11**
8	**Unsatisfiable**	**15892.13**

(b) VCO Verification Results

Table 5.2: *Experimental Results*

and $V_{D1} \in [1.5, 1.7]$, $V_{D2} \in [1.5, 1.7]$, $I_L \in [0, 0]$. We have divided the hybrid state space of the VCO, $V_{D1} \in [-2, 2]$, $V_{D2} \in [-2, 2]$, $I_L \in [-0.1, 0.1]$), in to 1000 boxes and followed the same procedure as we did for the TDO limit cycle identification. Starting from the boxes with the equilibrium state, we have searched the state space for periodicity in an arbitrary direction, this time fixing two axis and varying the third one, identifying the box satisfying the periodicity property. Furthermore, all boxes reachable from the box with the periodicity property have been found and later reachability of these from all others was ensured. The isolated limit cycle for the VCO is depicted in 3-D in Fig. 5.9b. The computation time for this limit cycle identification was noted to be approximately 11500 seconds of CPU time. Note that we have recognized that the limit cycle of the VCO spans over five regions of the V_{D1}, V_{D2} plane. Consequently, for the frequency domain properties of VCO (discussed later), we have used a hybrid system having five continuous flow sets as shown in Fig. 5.4.

To specify robust frequency domain properties, we have used simulation traces as

shown in Fig. 5.10a and Fig. 5.10b. The corresponding periodogram specifications for these traces are depicted in Fig. 5.11a, and Fig. 5.11b, for TDO and VCO respectively. Here we have only shown specifications for the fundamental frequency of the variables (V_d for TDO, and $VD1$ for VCO). The upper and lower bounds on these periodograms have been found based on designer judgement, i.e. we chose random values in the parameter spaces and correspondingly varied the "power spectral envelope" and arrived at these bounds. Considering the box, as shown in red in Fig. 5.9a, as the initial conditions for the state variables, we model checked the TDO HDS for eight unwindings of the BMC formula. We got "Unsat" results for the formula for all unwindings showing the satisfaction of the frequency domain properties. Verification results with computation times are given in Table 5.2a. Similarly for VCO, we considered the box shown in red in Fig. 5.9b as initial conditions, and verified the frequency domain property for eight unwindings of the BMC formula. We obtained "Unsat" for eight unwindings showing that the frequency domain property is satisfied by the VCO for the given ranges of parameters. Results have been given in Table 5.2b. These results show that the distance between the traces from the HDS models and that from the robust frequency domain specification has been less than the user defined bound for atleast eight unwindings of the BMC formula. Note that in a BMC algorithm, one unwinding corresponds to one discrete jump of the HDS model. Since BMC is a bounded approach, the results show that the model meets the robust frequency domain properties specification for a bounded time, corresponding to the eight unwindings.

5.5 Related Work

In [38], the authors verified time domain properties of a tunnel diode oscillator using the tool PHAVER, showing that the variations in amplitude and jitter in the oscillator behaviour are bounded, but it did not show that there are no undesired harmonics in the behaviour. [30] used the meta Tarski theorem prover, and showed that for certain values of parameters, a tunnel diode oscillator does not oscillate if it does not cross certain thresholds of the state variables; finding such thresholds is difficult for a nonlinear circuit model. A similar approach has been adopted in [45] proving that a TDO oscillates for all possible initial conditions using an ACTL specification and CHECKMATE. It does state that the presence of an unstable equilibrium point in the state space shows the presence of a limit cycle; this work however does not talk about the period or the frequency of the periodic orbit. Steinhorst et al. in [87], showed oscillations in TDO and ROs using visualization techniques. Though complex behaviours in the state

space can be visualized, absence of higher harmonics in oscillation was not formally verified. The Ring oscillator start up problem has been taken up in [44], [57]. They are based on finding absence of a stable DC equilibrium point. While the former uses small signal analysis around the equilibrium point, the later puts constraints on node voltages to establish stability of equilibrium points. Both these approaches are very localized and can not encapsulate the global behaviour of non-linear ring oscillator circuits; neither prove the absence of higher harmonic oscillations. Chao et al. [108] verified oscillator start up using techniques from dynamical system theory. Donze et al. [32] demonstrated time and frequency logic (TFL) for monitoring properties of music signals, but its extension to set based AMS circuit analysis needs further research.

Frequency domain approaches on the other hand are limited to small signal AC analysis of a more approximate linearized model around an equilibrium point. In [49], the author computed value sets of a transfer function for parameter variations and a range of frequencies using interval arithmetic. It was shown that the method was computationally very expensive, and its extension to non-linear circuits modelled as piece-wise affine would be a difficult task. Similarly, [64] derived amplitude and phase envelopes of a family of interval rational transfer functions for continuous-time systems. [30] used meta Tarski to prove that the magnitude of transfer function of a small operational amplifier is bounded for a range of frequencies.

5.6 Chapter Summary

In this chapter, we have shown frequency domain property specification and its verification using the SMODE technique for BMC of hybrid systems. Dynamical systems (continuous/hybrid) are generally verified for their frequency domain properties by linearizing these systems and transforming them in to frequency domain Laplace transforms. While this is theoretically simple, the loss of the non-linear behaviour of systems like oscillators is however significant. Furthermore, limit cycle is an inherent behaviour of non-linear systems and can not be approximated by any linearisation. For example, a non-linear system with a stable limit cycle when perturbed from its limit cycle, will always attain its state of oscillation with the same frequency. On the other side, a linear system when disturbed from its state of oscillation will start to oscillate with a different frequency. We therefore use an approach, borrowing techniques from both time and frequency domains, to specify properties in the frequency domain and verify them in the time domain. We have used a robust periodogram to specify the oscillatory behaviour of the analog oscillator for any parameter and process variations. Since ver-

ifying non-linear systems in the frequency domain is intractable, we have used a time domain BMC for the verification of these properties. Membership of the limit cycle in a set of the time domain trajectories generated from the power spectral envelope (defined by the periodogram specifications) have been shown using SMODE. Results are sound for bounded time and has been shown to be practical for simple circuits. In the future, as tools continue to develop and are capable of providing more efficient solvers and deduction methods giving improvements in computation time, we are confident that this approach will scale to more significant real world examples of AMS circuits. We have shown a successful first step in developing property-based formal verification in the frequency domain. In the future we intend to extend this methodology to cover transient region of circuits by using Fourier transform and short time Fourier transform. We also intend to extend the work to allow deduction directly in the frequency domain.

Chapter 6

Conclusion and Future Work

In this thesis, we have verified time and frequency domain properties of AMS circuits. Inevitability property has been the main focus of this thesis as oppose to the safety property in previous works. Proving Inevitability is somewhat more difficult than the safety property, as long term behaviour of the AMS circuit needs to be considered for the verification of Inevitability. Moreover, convergence to a "particular set" by all possible system trajectories has to be verified. Therefore, verifying Inevitability is intractable using the explicit time domain reachability analysis, where the continuous/hybrid state space is divided in to small partitions and reachability of the desired state is verified iteratively. Reachability being undecidable, for general continuous/hybrid systems, generates less accurate results. This is because conservative over-approximation is used to counter the undecidability issue. To reduce conservatism, the partitioning of the state space is refined to obtain more accurate results. However, no matter how high the granularity of these partitions is, reachability, being of an approximate nature, they can not give exact results to verify the Inevitability property.

In this thesis, we have presented mixed deductive-bounded and deductive-only verification methodologies benefiting from the exact certificate based deductive verification approach and improving upon what we can achieve with the bounded only verification approach. There are two major advantages of the deductive verification approach. It verifies a property for infinite time as oppose to the bounded time nature of the bounded reachability analysis, such as BMC of the continuous/hybrid systems. Secondly, for systems modelled as continuous/hybrid dynamical systems, deductive methods do not explicitly solve the ODEs encountered in these models, and consequently are free from any approximations of the trajectories. Certificate based deductive methods are an abstract way of deducing the long term behaviour of systems from the existence of a certificate having certain properties in semi-algebraic sets. We have used Lyapunov

theory and came up with some interesting local properties which were verified by different flavours of Lyapunov-like certificates. Involving quantifiers (both Universal and Existential), construction of Lyapunov-like certificates is an NP-Hard problem. Though there are symbolic tools (such as QEPCAD, Redlog) which can solve these formulas exactly, their computation time is such that they are not practically useful for realistic problems. Therefore, we have used numeric and numeric-symbolic approaches to deal with this complexity issue. On the numeric side, we have used SOS programming for numerically verification of positivity of polynomial functions in realistic computational time. Associated with numerical errors, results of the SOS programming are further validated in the symbolic QE tool. We have also used SOS programming based bounded advection of sets for partial verification of the Inevitability property.

Lastly, realizing the significance of verifying AMS circuits in the frequency domain, we have proposed frequency domain property specification. This has been followed by the verification of these frequency domain properties using a mixed time and frequency domain approach. We have verified two types of circuit in this thesis: pure analog (RO, TDO, VCO) and AMS circuits (CP PLL).

6.1 Verifying the Inevitability of Phase-Locking in CP PLL

In Chapter 3, we verified the Inevitability of phase locking in CP PLL AMS circuits. We have modelled the CP PLL as a HDS taking in to account both the continuous and discrete behaviours of the circuit. We have proposed two approaches for the verification of the Inevitability property of the CP PLL circuit. These are mixed deductive-bounded and deductive-only approaches. The Inevitability property has been divided into two sub-properties. One of the properties is verified by the deductive-only approach using Lyapunov stability certificates, and it is how we verify the second property that differentiates the two approaches from each other. The mixed deductive-bounded approach benefits from both deductive and bounded verification approaches, whereas the deductive-only approach uses an Escape certificate to verify the inevitability property. The first property essentially specifies the long term behaviour of the CP PLL HDS to converge to the locking state in a sub-set of the hybrid state space. Verification of this property is equivalent to verifying attractive invariance of a set with respect to the equilibrium locking state. We have shown attractive invariance of a set by construct-

ing multiple Lyapunov certificates and using their maximized level curves. The second property ensures that eventually all trajectories in the outer space converge to the attractive invariant set. This convergence to the attractive invariant set has been verified following two different approaches of mixed deductive-bounded and deductive-only. In the mixed deductive-bounded approach, convergence to the attractive invariant set has been partly verified by using Bounded advection of sets and partly using Escape certificate. Only the Escape certificate has been used showing convergence to the attractive invariant set in the deductive only approach. We have used semi-algebraic sets defined by polynomial inequalities to represent sets. Furthermore, our Inevitability verification involves FOFs with quantifiers (Universal and Existential). Verification of these formulas belongs to the set of NP-Hard problems, and though there are few symbolic tools available to eliminate these quantifiers, they work only for low dimension academic problems. We have therefore used numeric SOS programming technique to solve the problem in realistic computation time. We have verified the Inevitability property of a third and fourth order CP PLL.

Results show valuable insight in to the idea of using mixed deductive-bounded and deductive-only approaches to the verification of AMS circuits like CP PLL. Computation time is comparable to previous reachability based approaches with the added advantage of being applicable to an infinite horizon. Comparing deductive-bounded and deductive-only approaches, we have noticed that although the deductive-only approach is simple in the number of SOS programming queries, it is computationally more expensive than the deductive-bounded approach. Since the difference is in how we verify the second property in the set outside the attractive invariance, it clearly indicates that finding a single Escape certificate for a large set is more expensive than having one for a small set. However, due to the iterative nature of the bounded advection of sets using Euler maps for ODEs, large numbers of SOS programming queries have been used in the deductive-bounded approach. We therefore conclude that, where it is possible, deductive-only verification offers more accurate results with less user effort. We have also observed that higher the order of the CP PLL, a Lyapunov certificate with a lower degree is needed as has been demonstrated in the case of fourth order CP PLL. We therefore conclude that our approach, though it needs some expertise in the formulation of SOS programming, is more accurate and comparable in computation time to previous reachability based approaches. On the modelling side, we have observed that at the system level (CP PLL), it is more tractable to use an abstract model of sub-systems while conserving the significant non-linear behaviour of the real circuit.

6.2 Deductive Inevitability Verification of Ring Oscillators using the SOS-QE Approach

In Chapter 4, we extended the deductive-only verification methodology to the verification of inevitability of oscillations in ROs. To keep the verification task tractable, we modelled the RO at an abstract level, where we have ignored the circuit behaviour at the transistor level. To cater for the transistor parameters' (Length,width) effect on the system, we have introduced parameters such that they reflect these changes. We modelled the RO as a CDS. Recognizing the fact that the inevitability of an RO is a complex property, we divided it into several sub-properties defined in different subsets. We used a certificate based deductive approach for the verification of all these properties. Theoretically, these certificates are similar in nature to the Lyapunov certificate irrespective of the fact that they have been used for different purposes. Three different certificates have been used in three subsets showing attractive invariance of a set, Escape of trajectories from a set, and Eventuality of trajectories to reach to within an arbitrary small distance of the limit cycle. We have verified inevitability of ROs with even and odd topologies. Benefiting from the structural layout of the even stage RO, we have treated its differential and common modes separately. For the common mode operation, we have only used a Lyapunov certificate showing convergence of common mode trajectories to zero.

We have formulated the construction of these certificates as FOFs over polynomial inequalities, equations, and universal/existential quantifiers. Recognizing the fact that QE tools work only for problems of low dimensions, we have used a numeric-symbolic approach for the construction of different certificates needed to verify various sub-properties. We have used SOS programming, which uses semi-definite programming at the back end, and soundly constructed the certificates needed for the verification of different properties. The results of the numerically generated SOS polynomials, being error prone, have been further validated by using symbolic QE tool. Therefore, we can claim that as oppose to the results of Chapter 3, results of Chapter 4 are exact. However, this has been possible at a cost. The polynomial certificates resulting from the SOS programmer have been generally of degree higher than 4, and the resulting FOFs were of a length not possible to be given as an input to the QE tool. Therefore, we have had had to use some heuristics to verify these formulas in the symbolic QE tool, mainly by representing them in DNF and verifying each clause separately.

Results of the deductive-only verification methodology for the verification of in-

evitability in ROs has been very encouraging. In fact they have been better than previous approaches in several aspects. They are less conservative as compared to the reachability analysis as we do not need to solve ODEs and over-approximate their solutions. We do not use any partitioning of the state space and this reduces the computation time. Our results are for infinite horizon as oppose to the bounded time nature of the reachability approach. Lastly, computation times are comparable to previous approaches considering their iterative discretization of the state space and verifying reachability for each location.

6.3 Verifying Frequency Domain Properties of Oscillators using SMODE

In Chapter. 5, we have proposed a novel technique of robust frequency domain specification and verification. We modelled oscillators as CDS considering devices at transistor level. Recognizing the fact that the limit cycle is an inherent non-linear characteristic, we avoided linearization and Laplace transform techniques. Instead, we specified the non-linear behaviour of oscillators operating in close proximity to the periodic limit cycle. Towards this goal, we used a Fourier series based periodogram specification representing the energy content of each discrete frequency as a sum of squares of the Fourier series coefficients. To show that the oscillator has a particular frequency, and does not have undesired harmonics, we considered harmonic frequencies as well. Furthermore, catering for changes in frequency response of the oscillator as a result of parameter and process variations, we used robust periodogram specifications.

We verified frequency domain properties using a mixed time and frequency domain approach. This is because of the fact that verifying a non-linear system in the frequency domain is practically intractable with existing state of the art solvers. Therefore we verified, using the time domain SMODE technique for BMC of hybrid systems, that the distance of the hybrid arcs from the time domain periodic arcs, generated from the frequency domain specifications, is less than a user defined positive number for all time. We have verified frequency domain properties for benchmark TDO and VCO circuits. Results of our analysis are promising and a step in a novel direction of frequency domain properties specification and verification. Though computation times are relatively large, state of the art solvers could have only provided these results. We believe, as progress is being made in SMT solvers to be able to solve instances with transcendental functions,

in future, our research will make a considerable contribution in the formal verification of frequency domain properties.

6.4 Conclusion

We have successfully verified the inevitability property for various AMS circuits by proposing a divide and rule strategy, and divided the property in to several sub-properties. These properties were verified using deductive-bounded and deductive-only verification methodologies. The deductive technique made use of the Lyapunov control theory, and verified different properties via Lyapunov-like certificates. The task of the construction of these certificates was delegated to solving various SOS programs, followed by symbolic analysis. The results are very encouraging and are better than the previous reachability based approaches, in terms of accuracy and scalability.

Similarly, we have come up with a novel technique of expressing the behaviour of analog oscillators when they operate near to their limit cycles. We have proposed a methodology of how to take in to account the effects of changing parameters on the oscillator frequency. We have successfully verified frequency domain properties using a mixed time-frequency domain approach, employing the recent SMODE solver. This is a significant and novel step and as these tools progress, we expect to improve upon these results in future.

6.5 Future Work

Several research directions stem from the work presented in this thesis. In Chapter 3, we presented Lyapunov certificate-based attractive invariance of a set. The size of the set depends on the level surfaces of these certificates. Certificates with different structures can be found to enlarge the area of the attractive invariant set. For example, in [92], the author presented various techniques for enlarging this area using multiple Lyapunov functions. Similar technique can be used in AMS circuit verification of the inevitability property. The techniques we proposed for the inevitability verification of the CP PLL can be extended to digital extensive PLLs. This can be done by difference equation based modelling techniques for which Lyapunov theory has already been used in [3]. The use of an Escape certificate in verifying "Escape from a set property" can be improved such that several such certificates are computed for a decomposition of the large set where we verify convergence to the invariance set. The numerically generated

certificates in Chapter 3 can be symbolically validated similar to Chapter 4. The only difficulty is the size of the FOF representing Boolean combination of the conditions of these certificates. To reduce the size of these formulas, and reduce the size of the certificates themselves, lower degree certificates can be used such as proposed in [92] and [4].

While verifying the inevitability of ROs, we faced the problem of dealing with FOFs of sizes such that it was not possible for the symbolic tool to solve it as a single query. We had to split these FOFs in to clauses and verify them individually. We intend to improve these results of the SOS-QE technique in Chapter 4 by using lower degree certificates (piecewise or point wise maximum) as proposed in [92] and [4]. Using certificates (may be more in number) of lower degrees lend themselves to symbolic verification. One interesting direction that we aim to pursue in a future research project is the formalization and automation of our SOS-QE procedure, similar to [47], [6], [11]. We aim to extend our work, specifically for AMS circuit verification, and come up with an automatic decision procedure in a theorem proving engine dealing with non-linear polynomial optimization. As has been described in [4], SOS programming, relying on semi-definite programming, suffers from scalability issues as the programs are exponential in the dimension n and degree d of the certificate. To overcome this issue, and apply our methodology to higher order PLLs/ROs, an interesting direction is to use a combination of linear programming (LP) and second order cone programming (SOCP), discussed in [4]. This can greatly reduce the degree and overall size of certificates which are more amenable to symbolic analysis. Similarly, instead of using a numeric-symbolic approach, a research direction could be to identify feasible certificates with coefficients in the rational number set [75], [59], [58]. This eliminates the need for re-validating the numerically calculated certificates with floating point coefficients.

We suggest several future extensions to our research on frequency domain properties specification and verification. This technique can be extended to verify frequency domain properties of transients in AMS circuits. Towards this goal, we can use a Fourier transform to specify the complete behaviour of an AMS circuit in the frequency domain such that the circuit will not have a frequency content higher than a specified cut off. The other extension is to carry out the complete verification process in the frequency domain. Other circuits such as PLLs can be verified in the frequency domain to verify their frequency variations around a nominal value for variations in parameters. Lastly, we have considered fixed fundamental and harmonic frequencies, and to consider phase noise in our analysis, we can add constraints on the frequencies. This can be done in the existing methodology but at the cost of additional computation time by the SMODE

solver.

Appendix A

A.1 Semi-definite Programming

A semi-definite program is an optimization problem of the following form [14],

$$\begin{aligned} &\text{minimize:} && c^T x \\ &\text{subject to} && x_1 F_1 + .. + x_n F_n + G \preceq 0, \\ & && Ax = b, \end{aligned} \tag{1}$$

where $G,\ F_i \in \mathbb{R}^{m \times m}$, for $i = 1, ..n$, $A \in \mathbb{R}^{m \times n}$. The inequality in the above optimization program is affine in x, and therefore this problem is also termed Linear Matrix Inequality (LMI). An important property of the SDP feasibility problem is its convexity which can be efficiently solved using numerical methods, such as the interior point method.

Appendix B

B.2 Lyapunov Certificates

B.2.1 Third Order CP PLL

$0.0045x1^6 - 0.0287x1^5x2 + 0.0856x1^4x2^2 - 0.1482x1^3x2^3 + 0.1585x1^2x2^4 - 0.0986x1x2^5 + 0.0274x2^6 - 0.0094x1^5x3 + 0.0441x1^4x2x3 - 0.0949x1^3x2^2x3 + 0.1147x1^2x2^3x3 - 0.0761x1x2^4x3 + 0.0209x2^5x3 + 0.0290x1^4x3^2 - 0.0933x1^3x2x3^2 + 0.1480x1^2x2^2x3^2 - 0.1188x1x2^3x3^2 + 0.0419x2^4x3^2 - 0.0306x1^3x3^3 + 0.0538x1^2x2x3^3 - 0.0344x1x2^2x3^3 + 1.8004e-04x2^3x3^3 + 0.0492x1^2x3^4 - 0.0771x1x2x3^4 + 0.0442x2^2x3^4 - 0.0561x1x3^5 - 0.0204x2x3^5 + 0.9642x3^6.$

$0.001341331582 - 9.1670e-04x1 - 0.0099x3 + 1.8925e-04x1^2 + 0.0016x2^2 + 0.0589x3^2 + 9.1968e-05x1x2 + 0.0048x1x3 - 8.2196e-04x2x3 - 7.1721e-04x1^3 - 8.3907e-04x1^2x2 + 8.6911e-04x1x2^2 + 1.1350e-04x2^3 - 0.0019x1^2x3 + 9.7889e-05x1x2x3 - 0.0090x2^2x3 - 0.0294x1x3^2 - 0.0011x2x3^2 - 0.1495x3^3 + 4.9324e-04x1^4 - 4.9549e-04x1^3x2 + 8.4285e-04x1^2x2^2 + 0.0011x1x2^3 + 6.1185e-04x2^4 + 0.0037x1^3x3 + 0.0052x1^2x2x3 - 0.0019x1x2^2x3 - 7.0112e-04x2^3x3 + 0.0015x1^2x3^2 - 9.1061e-04x1x2x3^2 + 0.0209x2^2x3^2 + 0.0713x1x3^3 + 0.0141x2x3^3 + 0.1583x3^4 - 1.7346e-04x1^5 - 5.3615e-04x1^4x2 - 5.8522e-04x1^3x2^2 - 1.3701e-04x1^2x2^3 - 5.3559e-05x1x2^4 - 0.0014x1^4x3 + 6.8525e-04x1^3x2x3 + 0.0011x1^2x2^2x3 - 0.0012x1x2^3x3 - 7.1250e-04x2^4x3 - 0.0074x1^3x3^2 - 0.0126x1^2x2x3^2 + 0.0011x1x2^2x3^2 + 0.0012x2^3x3^2 + 0.0075x1^2x3^3 + 0.0046x1x2x3^3 - 0.0152x2^2x3^3 - 0.0665x1x3^4 - 0.0213x2x3^4 - 0.0580x3^5 + 0.0045x1^6 - 0.0287x1^5x2 + 0.0856x1^4x2^2 - 0.1482x1^3x2^3 + 0.1586x1^2x2^4 - 0.0986x1x2^5 + 0.0274x2^6 - 0.0094x1^5x3 + 0.0442x1^4x2x3 - 0.0947x1^3x2^2x3 + 0.1147x1^2x2^3x3 - 0.0761x1x2^4x3 + 0.0209x2^5x3 + 0.0290x1^4x3^2 - 0.0947x1^3x2x3^2 + 0.1448x1^2x2^2x3^2 - 0.1185x1x2^3x3^2 + 0.0415x2^4x3^2 - 0.0264x1^3x3^3 + 0.0622x1^2x2x3^3 - 0.0342x1x2^2x3^3 - 3.2896e-04x2^3x3^3 + 0.0411x1^2x3^4 -$

$0.0813x1x2x3^4 + 0.0450x2^2x3^4 - 0.0357x1x3^5 - 0.0116x2x3^5 + 0.9629x3^6.$
$0.00134023928 + 9.1689e-04x1 + 0.0097x3 + 1.8505e-04x1^2 + 0.0016x2^2 + 0.0576x3^2 +$
$8.7490e-05x1x2 + 0.0047x1x3 - 9.3516e-04x2x3 + 7.1432e-04x1^3 + 8.3253e-04x1^2x2 -$
$8.8106e-04x1x2^2 - 1.1727e-04x2^3 + 0.0019x1^2x3 - 1.3486e-04x1x2x3 + 0.0090x2^2x3 +$
$0.0290x1x3^2 + 7.1032e-04x2x3^2 + 0.1470x3^3 + 4.9454e-04x1^4 - 5.0103e-04x1^3x2 +$
$8.5568e-04x1^2x2^2 + 0.0011x1x2^3 + 6.1944e-04x2^4 + 0.0037x1^3x3 + 0.0052x1^2x2x3 -$
$0.0020x1x2^2x3 - 7.0485e-04x2^3x3 + 0.0015x1^2x3^2 - 9.9430e-04x1x2x3^2 + 0.0208x2^2x3^2 +$
$0.0709x1x3^3 + 0.0137x2x3^3 + 0.1574x3^4 + 1.7553e-04x1^5 + 5.4016e-04x1^4x2 +$
$5.9084e-04x1^3x2^2 + 1.4129e-04x1^2x2^3 + 5.4219e-05x1x2^4 + 0.0014x1^4x3 - 7.1054e-$
$04x1^3x2x3 - 0.0011x1^2x2^2x3 + 0.0012x1x2^3x3 + 7.4397e-04x2^4x3 + 0.0075x1^3x3^2 +$
$0.0127x1^2x2x3^2 - 0.0012x1x2^2x3^2 - 0.0012x2^3x3^2 - 0.0075x1^2x3^3 - 0.0046x1x2x3^3 +$
$0.0150x2^2x3^3 + 0.0668x1x3^4 + 0.0213x2x3^4 + 0.0594x3^5 + 0.0045x1^6 - 0.0287x1^5x2 +$
$0.0856x1^4x2^2 - 0.1482x1^3x2^3 + 0.1586x1^2x2^4 - 0.0986x1x2^5 + 0.0274x2^6 - 0.0094x1^5x3 +$
$0.0442x1^4x2x3 - 0.0947x1^3x2^2x3 + 0.1147x1^2x2^3x3 - 0.0761x1x2^4x3 + 0.0209x2^5x3 +$
$0.0290x1^4x3^2 - 0.0948x1^3x2x3^2 + 0.1448x1^2x2^2x3^2 - 0.1185x1x2^3x3^2 + 0.0415x2^4x3^2 -$
$0.0264x1^3x3^3 + 0.0623x1^2x2x3^3 - 0.0343x1x2^2x3^3 - 3.5304e-04x2^3x3^3 + 0.0412x1^2x3^4 -$
$0.0813x1x2x3^4 + 0.0449x2^2x3^4 - 0.0354x1x3^5 - 0.0114x2x3^5 + 0.9638x3^6.$

B.2.2 Fourth Order CP PLL

$0.0032x1^4 + 2.7182e-05x1^3x2 + 5.8767e-05x1^2x2^2 + 4.1039e-05x2^4 - 2.1590e-$
$04x1^3x3 - 1.7776e-04x1^2x2x3 - 2.0303e-05x1x2^2x3 - 1.7521e-04x2^3x3 + 1.3634e-$
$04x1^2x3^2 + 8.1972e-05x1x2x3^2 + 4.5238e-04x2^2x3^2 - 8.3780e-05x1x3^3 - 5.1037e-$
$04x2x3^3 + 2.3540e-04x3^4 + 5.1681e-04x1^3x4 - 1.4012e-04x1^2x2x4 + 1.2315e-$
$05x1x2^2x4 - 1.1398e-04x2^3x4 - 1.4829e-04x1^2x3x4 + 1.0921e-05x1x2x3x4 +$
$2.6429e-04x2^2x3x4 + 5.4511e-05x1x3^2x4 - 3.2835e-04x2x3^2x4 + 7.7165e-05x3^3x4 +$
$0.0039x1^2x4^2 + 3.7459e-05x1x2x4^2 + 4.0877e-04x2^2x4^2 - 2.4536e-04x1x3x4^2 -$
$3.4247e-04x2x3x4^2 + 4.2323e-04x3^2x4^2 + 2.4831e-04x1x4^3 - 2.4512e-04x2x4^3 -$
$1.9282e-04x3x4^3 + 0.0016x4^4.$
$6.679107934e-05 - 5.7263e-05x2 + 1.1257e-05x3 - 3.8488e-04x4 + 5.5263e-05x1^2 +$
$1.2369e-05x2^2 + 9.2214e-05x3^2 + 0.0011x4^2 + 1.2009e-05x1x3 + 3.5790e-05x2x3 -$
$1.6631e-05x1x4 + 2.5997e-04x2x4 + 4.1801e-05x3x4 - 5.7937e-05x1^2x2 - 3.0019e-$
$05x2^3 - 2.4677e-05x1^2x3 - 2.5902e-05x2^2x3 - 2.5371e-05x2x3^2 - 1.7793e-05x3^3 -$
$1.1448e-04x1^2x4 + 1.5853e-05x1x2x4 - 1.3636e-05x2^2x4 - 5.7279e-05x1x3x4 +$
$8.2492e-05x2x3x4 - 1.2598e-04x3^2x4 + 3.0465e-05x1x4^2 - 6.0710e-04x2x4^2 -$
$2.0224e-04x3x4^2 - 0.0011x4^3 + 0.0032x1^4 + 2.7346e-05x1^3x2 + 6.3534e-05x1^2x2^2 +$

$4.1430e-05x2^4-2.1497e-04x1^3x3-1.7448e-04x1^2x2x3-2.3497e-05x1x2^2x3-1.8607e-04x2^3x3+1.3684e-04x1^2x3^2+8.2992e-05x1x2x3^2+4.4749e-04x2^2x3^2-8.1842e-05x1x3^3-5.0485e-04x2x3^3+2.3848e-04x3^4+5.1769e-04x1^3x4-5.3379e-05x1^2x2x4-8.9272e-05x2^3x4-1.1457e-04x1^2x3x4+2.7328e-04x2^2x3x4+5.0106e-05x1x3^2x4-2.8968e-04x2x3^2x4+9.8729e-05x3^3x4+0.0039x1^2x4^2+1.9440e-05x1x2x4^2+2.4539e-04x2^2x4^2-1.8283e-04x1x3x4^2-5.4663e-04x2x3x4^2+4.2292e-04x3^2x4^2+2.3680e-04x1x4^3+1.1141e-04x2x4^3-3.7303e-05x3x4^3+0.0019x4^4.$

$6.646611473e-05+5.7376e-05x2-1.1035e-05x3+3.8929e-04x4+5.6130e-05x1^2+1.2645e-05x2^2+9.2813e-05x3^2+0.0012x4^2+1.2057e-05x1x3+3.5138e-05x2x3+2.6132e-04x2x4+4.6449e-05x3x4+5.7592e-05x1^2x2+2.9821e-05x2^3+2.4456e-05x1^2x3+2.5912e-05x2^2x3+2.5411e-05x2x3^2+1.7738e-05x3^3+1.1901e-04x1^2x4-1.5425e-05x1x2x4+1.5139e-05x2^2x4+5.8450e-05x1x3x4-8.4558e-05x2x3x4+1.2776e-04x3^2x4+6.0917e-04x2x4^2+2.1249e-04x3x4^2+0.0011x4^3+0.0032x1^4+2.7378e-05x1^3x2+6.3099e-05x1^2x2^2+4.1232e-05x2^4-2.1497e-04x1^3x3-1.7435e-04x1^2x2x3-2.3380e-05x1x2^2x3-1.8564e-04x2^3x3+1.3627e-04x1^2x3^2+8.2980e-05x1x2x3^2+4.4725e-04x2^2x3^2-8.1890e-05x1x3^3-5.0490e-04x2x3^3+2.3820e-04x3^4+5.1698e-04x1^3x4-5.3947e-05x1^2x2x4-8.9342e-05x2^3x4-1.1532e-04x1^2x3x4+2.7342e-04x2^2x3x4+4.9577e-05x1x3^2x4-2.8976e-04x2x3^2x4+9.8389e-05x3^3x4+0.0039x1^2x4^2+1.9872e-05x1x2x4^2+2.4707e-04x2^2x4^2-1.8213e-04x1x3x4^2-5.4883e-04x2x3x4^2+4.2503e-04x3^2x4^2+2.6135e-04x1x4^3+1.1198e-04x2x4^3-3.1518e-05x3x4^3+0.0019x4^4.$

B.3 Escape Certificates

B.3.1 Third Order CP PLL

$1.69e+03x1^2+971.57x1x2+289.91x2^2-0.30x1x3-0.10x2x3-0.25x3^2+692.37x1^4+693.57x1^3x2+410.51x1^2x2^2+115.45x1x2^3+88.27x2^4-2.13x1^3x3-2.41x1^2x2x3-0.841x1x2^2x3-0.10x2^3x3+803.18x1^2x3^2+362.79x1x2x3^2+170.45x2^2x3^2-0.33x1x3^3-0.09x2x3^3-0.67x3^4+394.27x1^6+508.80x1^5x2+439.31x1^4x2^2+190.89x1^3x2^3+129.13x1^2x2^4+50.41x1x2^5+59.55x2^6-1.07x1^5x3-1.86x1^4x2x3-1.30x1^3x2^2x3-0.5087x1^2x2^3x3-0.1413x1x2^4x3-0.0351x2^5x3+383.52x1^4x3^2+277.21x1^3x2x3^2+177.07x1^2x2^2x3^2+50.12x1x2^3x3^2+74.59x2^4x3^2-2.74x1^3x3^3-3.01x1^2x2x3^3-0.98x1x2^2x3^3-0.11x2^3x3^3+664.43x1^2x3^4+267.71x1x2x3^4+152.25x2^2x3^4+2.16x1x3^5+$

$$0.71x2x3^5 - 1.18x3^6 + 263.73x1^8 + 385.50x1^7x2 + 426.48x1^6x2^2 + 254.81x1^5x2^3 + 173.64x1^4x2^4 + 78.86x1^3x2^5 + 83.07x1^2x2^6 + 37.48x1x2^7 + 44.65x2^8 - 0.69x1^7x3 - 1.24x1^6x2x3 - 1.55x1^5x2^2x3 - 0.61x1^4x2^3x3 - 0.69x1^3x2^4x3 + 0.002x1^2x2^5x3 - 0.22x1x2^6x3 + 0.01x2^7x3 + 261.19x1^6x3^2 + 245.66x1^5x2x3^2 + 205.46x1^4x2^2x3^2 + 70.71x1^3x2^3x3^2 + 84.25x1^2x2^4x3^2 + 35.97x1x2^5x3^2 + 54.66x2^6x3^2 - 1.92x1^5x3^3 - 2.64x1^4x2x3^3 - 1.93x1^3x2^2x3^3 - 0.60x1^2x2^3x3^3 - 0.22x1x2^4x3^3 - 0.0260x2^5x3^3 + 351.07x1^4x3^4 + 232.19x1^3x2x3^4 + 153.67x1^2x2^2x3^4 + 43.55x1x2^3x3^4 + 73.39x2^4x3^4 - 4.11x1^3x3^5 - 4.10x1^2x2x3^5 - 1.28x1x2^2x3^5 - 0.09x2^3x3^5 + 698.87x1^2x3^6 + 281.88x1x2x3^6 + 160.40x2^2x3^6 + 1.13x1x3^7 + 0.37x2x3^7 - 2.51x3^8 + 199.36x1^{10} + 332.14x1^9x2 + 448.63x1^8x2^2 + 342.81x1^7x2^3 + 254.37x1^6x2^4 + 135.76x1^5x2^5 + 111.64x1^4x2^6 + 63.52x1^3x2^7 + 69.80x1^2x2^8 + 30.84x1x2^9 + 35.4981x2^{10} - 0.41x1^9x3 - 0.9725x1^8x2x3 - 1.43x1^7x2^2x3 - 0.9982x1^6x2^3x3 - 0.90x1^5x2^4x3 - 0.39x1^4x2^5x3 - 0.41x1^3x2^6x3 - 0.09x1^2x2^7x3 - 0.16x1x2^8x3 - 0.03x2^9x3 + 210.97x1^8x3^2 + 248.60x1^7x2x3^2 + 256.68x1^6x2^2x3^2 + 125.32x1^5x2^3x3^2 + 106.05x1^4x2^4x3^2 + 52.71x1^3x2^5x3^2 + 75.47x1^2x2^6x3^2 + 35.03x1x2^7x3^2 + 44.25x2^8x3^2 - 1.35x1^7x3^3 - 2.38x1^6x2x3^3 - 2.53x1^5x2^2x3^3 - 1.22x1^4x2^3x3^3 - 0.84x1^3x2^4x3^3 - 0.10x1^2x2^5x3^3 - 0.23x1x2^6x3^3 - 0.03x2^7x3^3 + 270.01x1^6x3^4 + 254.29x1^5x2x3^4 + 211.24x1^4x2^2x3^4 + 72.89x1^3x2^3x3^4 + 87.55x1^2x2^4x3^4 + 37.54x1x2^5x3^4 + 55.90x2^6x3^4 - 3.02x1^5x3^5 - 4.44x1^4x2x3^5 - 3.08x1^3x2^2x3^5 - 0.89x1^2x2^3x3^5 - 0.37x1x2^4x3^5 - 0.02x2^5x3^5 + 411.29x1^4x3^6 + 297.49x1^3x2x3^6 + 190.23x1^2x2^2x3^6 + 53.71x1x2^3x3^6 + 79.27x2^4x3^6 - 5.52x1^3x3^7 - 5.89x1^2x2x3^7 - 1.86x1x2^2x3^7 - 0.07x2^3x3^7 + 881.67x1^2x3^8 + 415.43x1x2x3^8 + 180.79x2^2x3^8 - 4.81x1x3^9 - 1.57x2x3^9 - 6.02x3^{10} + 480.72x1^{12} + 735.96x1^11x2 + 1.1e + 03x1^{10}x2^2 + 989.85x1^9x2^3 + 841.38x1^8x2^4 + 467.31x1^7x2^5 + 398.69x1^6x2^6 + 243.35x1^5x2^7 + 291.64x1^4x2^8 + 175.85x1^3x2^9 + 205.99x1^2x2^{10} + 96.90x1x2^11 + 118.56x2^{12} - 1.88x1^11x3 - 0.59x1^{10}x2x3 - 1.96x1^9x2^2x3 - 0.13x1^8x2^3x3 - 1.27x1^7x2^4x3 + 0.38x1^6x2^5x3 - 0.93x1^5x2^6x3 + 0.60x1^4x2^7x3 - 0.30x1^3x2^8x3 + 0.76x1^2x2^9x3 - 0.0370x1x2^{10}x3 + 0.68x2^11x3 + 224.54x1^{10}x3^2 + 323.88x1^9x2x3^2 + 429.14x1^8x2^2x3^2 + 296.36x1^7x2^3x3^2 + 235.93x1^6x2^4x3^2 + 117.22x1^5x2^5x3^2 + 120.10x1^4x2^6x3^2 + 63.64x1^3x2^7x3^2 + 84.67x1^2x2^8x3^2 + 35.93x1x2^9x3^2 + 47.27x2^{10}x3^2 - 1.35x1^9x3^3 - 2.13x1^8x2x3^3 - 3.60x1^7x2^2x3^3 - 2.23x1^6x2^3x3^3 - 2.15x1^5x2^4x3^3 - 0.66x1^4x2^5x3^3 - 0.87x1^3x2^6x3^3 + 0.05x1^2x2^7x3^3 - 0.32x1x2^8x3^3 + 0.13x2^9x3^3 + 275.49x1^8x3^4 + 353.87x1^7x2x3^4 + 384.16x1^6x2^2x3^4 + 203.99x1^5x2^3x3^4 + 157.70x1^4x2^4x3^4 + 73.31x1^3x2^5x3^4 + 95.54x1^2x2^6x3^4 + 40.79x1x2^7x3^4 + 54.29x2^8x3^4 - 3.11x1^7x3^5 - 4.97x1^6x2x3^5 - 5.46x1^5x2^2x3^5 - 2.44x1^4x2^3x3^5 - 1.67x1^3x2^4x3^5 - 0.22x1^2x2^5x3^5 - 0.61x1x2^6x3^5 + 0.11x2^7x3^5 + 396.45x1^6x3^6 + 455.06x1^5x2x3^6 + 370.07x1^4x2^2x3^6 +$$

$146.94x1^3x2^3x3^6 + 129.66x1^2x2^4x3^6 + 60.56x1x2^5x3^6 + 70.00x2^6x3^6 - 5.22x1^5x3^7 -$
$8.18x1^4x2x3^7 - 5.35x1^3x2^2x3^7 - 1.54x1^2x2^3x3^7 - 0.53x1x2^4x3^7 - 0.11x2^5x3^7 + 658.88x1^4x3^8 +$
$586.84x1^3x2x3^8 + 339.031x1^2x2^2x3^8 + 107.50x1x2^3x3^8 + 101.76x2^4x3^8 - 6.97x1^3x3^9 -$
$8.56x1^2x2x3^9 - 2.30x1x2^2x3^9 - 0.01x2^3x3^9 + 1.53e + 03x1^2x3^{10} + 822.51x1x2x3^{10} +$
$283.09x2^2x3^{10} + 4.73x1x3^{11} + 1.63x2x3^{11} - 13.84x3^{12}.$
$-1.1170x1 - 0.6735x2 + 0.0102x3 + 0.1212x1^2 - 0.2695x1x2 + 0.1365x2^2 + 0.0161x1x3 +$
$0.0145x2x3 - 1.3295e - 04x3^2.$
$1.1346x1 + 0.6416x2 - 0.0105x3 + 0.1035x1^2 - 0.2350x1x2 + 0.1183x2^2 + 0.0165x1x3 +$
$0.0136x2x3 - 1.3840e - 04x3^2.$

B.3.2 Fourth Order CP PLL

Deductive-Bounded Verification:
$0.5716x1^2 - 1.8978x1x2 + 0.5970x2^2 - 1.2771x1x3 + 0.5142x2x3 + 0.3226x3^2 + 6.0960x1x4 +$
$0.6704x2x4 + 0.4063x3x4 + 1.2297e - 04x4^2 + 1.2343x1^4 - 1.8361x1^3x2 - 0.4136x1^2x2^2 -$
$0.2727x1x2^3 + 0.3537x2^4 - 1.2257x1^3x3 - 0.6465x1^2x2x3 - 0.5960x1x2^2x3 + 0.6499x2^3x3 -$
$0.1219x1^2x3^2 - 0.4352x1x2x3^2 + 0.6644x2^2x3^2 - 0.1134x1x3^3 + 0.3196x2x3^3 + 0.1341x3^4 +$
$4.9357x1^3x4 + 2.6858x1^2x2x4 + 0.5378x1x2^2x4 - 0.1053x2^3x4 + 1.6851x1^2x3x4 +$
$0.6833x1x2x3x4 - 0.2173x2^2x3x4 + 0.2305x1x3^2x4 - 0.1527x2x3^2x4 - 0.0390x3^3x4 +$
$0.7354x1^2x4^2 - 1.5554x1x2x4^2 + 0.5123x2^2x4^2 - 1.0560x1x3x4^2 + 0.4251x2x3x4^2 +$
$0.2784x3^2x4^2 + 5.0210x1x4^3 + 0.5525x2x4^3 + 0.3349x3x4^3.$
$-0.1048 * x1 - 0.6358 * x2 - 0.6033 * x3 + 0.0074 * x4 + 1.9974 * x1^2 + 0.0032 * x1 * x2 +$
$0.0223 * x2^2 - 0.0253 * x1 * x3 - 0.0511 * x2 * x3 + 0.0238 * x3^2 + 0.1682 * x1 * x4 +$
$0.0081 * x2 * x4 + 0.0116 * x3 * x4 - 3.7127e - 04 * x4^2.$
$0.0061x1 + 0.1893x2 + 1.4139x3 + 0.0011x4 + 0.0061x1^2 + 0.0052x1x2 + 0.1709x2^2 +$
$0.0103x1x3 - 0.3438x2x3 + 0.6271x3^2 - 7.9986e - 04x3x4 + 0.0857x1^3 + 0.2965x1^2x2 +$
$0.0392x1x2^2 + 0.0719x2^3 + 0.6592x1^2x3 + 0.0623x1x2x3 + 0.4477x2^2x3 - 0.0015x1x3^2 +$
$0.0441x2x3^2 - 0.0743x3^3 - 4.0447e - 04x1^2x4 - 1.2140e - 04x2^2x4 - 0.0017x1x3x4 +$
$3.0000e - 04x2x3x4 - 0.0011x3^2x4 + 0.0062x1x4^2 + 0.2591x2x4^2 + 0.8073x3x4^2 + 0.0016x4^3 +$
$3.1308x1^4 - 0.0022x1^3x2 + 0.0684x1^2x2^2 + 0.0048x1x2^3 + 0.0393x2^4 - 0.0853x1^3x3 -$
$0.1137x1^2x2x3 - 0.0064x1x2^2x3 - 0.0724x2^3x3 + 0.0557x1^2x3^2 + 0.0051x1x2x3^2 - 0.0122x2^2x3^2 +$
$0.0097x1x3^3 + 0.0373x2x3^3 + 0.0873x3^4 + 0.0194x1^3x4 + 4.6850e - 04x1^2x2x4 + 1.4614e -$
$04x2^3x4 + 9.0770e - 04x1^2x3x4 + 8.3574e - 04x1x2x3x4 + 4.6533e - 04x2x3^2x4 + 1.6312e -$
$04x3^3x4 + 0.1540x1^2x4^2 - 2.9423e - 04x1x2x4^2 + 0.1027x2^2x4^2 - 0.0207x1x3x4^2 -$
$0.1627x2x3x4^2 + 0.2086x3^2x4^2 - 6.7642e - 04x1x4^3 + 8.9842e - 04x3x4^3.$

Deductive-Only Verification:

$$218.2054x1^2-724.2974x1x2+225.8941x2^2-487.4403x1x3+194.7218x2x3+122.1302x3^2+2.3142e+03x1x4+254.5501x2x4+154.2719x3x4+0.3149x4^2+274.8406x1^4-408.1116x1^3x2-62.9006x1^2x2^2-130.2242x1x2^3+130.3355x2^4-271.3946x1^3x3-158.3718x1^2x2x3-275.7396x1x2^2x3+201.3793x2^3x3+0.4527x1^2x3^2-197.3733x1x2x3^2+209.9656x2^2x3^2-50.8696x1x3^3+98.6859x2x3^3+55.4381x3^4+810.3546x1^3x4+461.0312x1^2x2x4+112.8572x1x2^2x4-27.7462x2^3x4+292.5319x1^2x3x4+143.5278x1x2x3x4-56.4005x2^2x3x4+49.9083x1x3^2x4-40.1189x2x3^2x4-10.7445x3^3x4+175.5234x1^2x4^2-349.4142x1x2x4^2+135.6410x2^2x4^2-237.1636x1x3x4^2+79.5019x2x3x4^2+78.5147x3^2x4^2+1.3522e+03x1x4^3+148.7329x2x4^3+90.1417x3x4^3-0.5151x4^4+331.5602x1^6-289.7337x1^5x2-44.5056x1^4x2^2-148.9686x1^3x2^3+24.8828x1^2x2^4-22.3955x1x2^5+76.3346x2^6-198.9197x1^5x3-131.0286x1^4x2x3-289.4706x1^3x2^2x3-61.2132x1^2x2^3x3-84.7176x1x2^4x3+138.0442x2^5x3+10.5631x1^4x3^2-193.7018x1^3x2x3^2-46.9663x1^2x2^2x3^2-128.6070x1x2^3x3^2+208.3036x2^4x3^2-48.0819x1^3x3^3-21.9327x1^2x2x3^3-104.2390x1x2^2x3^3+162.2451x2^3x3^3+31.5554x1^2x3^4-45.6940x1x2x3^4+117.1682x2^2x3^4-10.6668x1x3^5+48.1018x2x3^5+34.6423x3^6+575.4837x1^5x4+404.1483x1^4x2x4+189.7930x1^3x2^2x4+65.0483x1^2x2^3x4-0.4517x1x2^4x4-13.4718x2^5x4+252.2362x1^4x3x4+233.8666x1^3x2x3x4+118.1129x1^2x2^2x3x4-1.8981x1x2^3x3x4-41.6152x2^4x3x4+80.5990x1^3x3^2x4+80.3749x1^2x2x3^2x4-1.1124x1x2^2x3^2x4-56.0022x2^3x3^2x4+21.0157x1^2x3^3x4+0.9352x1x2x3^3x4-40.5361x2^2x3^3x4+0.4200x1x3^4x4-16.6576x2x3^4x4-3.7262x3^5x4+199.6751x1^4x4^2-178.3625x1^3x2x4^2-13.7995x1^2x2^2x4^2-76.7466x1x2^3x4^2+88.6200x2^4x4^2-120.1260x1^3x3x4^2-96.4895x1^2x2x3x4^2-161.2453x1x2^2x3x4^2+99.2932x2^3x3x4^2+13.8720x1^2x3^2x4^2-114.2433x1x2x3^2x4^2+104.5764x2^2x3^2x4^2-29.2313x1x3^3x4^2+49.0719x2x3^3x4^2+43.5367x3^4x4^2+549.3334x1^3x4^3+298.8656x1^2x2x4^3+69.3973x1x2^2x4^3-20.5004x2^3x4^3+189.9144x1^2x3x4^3+87.5125x1x2x3x4^3-40.7043x2^2x3x4^3+31.1981x1x3^2x4^3-28.7995x2x3^2x4^3-7.9001x3^3x4^3+169.3557x1^2x4^4-299.3768x1x2x4^4+112.5971x2^2x4^4-204.0243x1x3x4^4+49.6121x2x3x4^4+67.5033x3^2x4^4+1.0978e+03x1x4^5+120.8098x2x4^5+73.2190x3x4^5+0.3335x4^6+476.3272x1^8-260.7127x1^7x2-42.6157x1^6x2^2-161.3527x1^5x2^3-3.5140x1^4x2^4-51.2348x1^3x2^5+38.4569x1^2x2^6-1.7833x1x2^7+51.7535x2^8-190.2391x1^7x3-127.5069x1^6x2x3-317.0168x1^5x2^2x3-124.1348x1^4x2^3x3-167.6342x1^3x2^4x3+3.5474x1^2x2^5x3-18.5926x1x2^6x3+96.6752x2^7x3+14.0310x1^6x3^2-209.2511x1^5x2x3^2-101.9569x1^4x2^2x3^2-222.1974x1^3x2^3x3^2-8.9005x1^2x2^4x3^2-47.5353x1x2^5x3^2+$$

$$183.5610x2^6x3^2 - 49.3684x1^5x3^3 - 43.8805x1^4x2x3^3 - 157.9611x1^3x2^2x3^3 - 33.5924x1^2x2^3x3^3 - 72.9076x1x2^4x3^3 + 198.0452x2^5x3^3 + 27.3078x1^4x3^4 - 61.4988x1^3x2x3^4 + 7.9665x1^2x2^2x3^4 - 68.5388x1x2^3x3^4 + 182.6346x2^4x3^4 - 11.9509x1^3x3^5 + 11.8897x1^2x2x3^5 - 42.8518x1x2^2x3^5 + 112.3370x2^3x3^5 + 27.7625x1^2x3^6 - 17.9673x1x2x3^6 + 77.0185x2^2x3^6 - 4.7233x1x3^7 + 32.8676x2x3^7 + 26.2538x3^8 + 618.2642x1^7x4 + 490.0785x1^6x2x4 + 253.1415x1^5x2^2x4 + 112.6347x1^4x2^3x4 + 33.2777x1^3x2^4x4 - 4.3474x1^2x2^5x4 - 18.9821x1x2^6x4 - 14.2245x2^7x4 + 300.2779x1^6x3x4 + 310.5358x1^5x2x3x4 + 202.5881x1^4x2^2x3x4 + 77.0198x1^3x2^3x3x4 - 12.9004x1^2x2^4x3x4 - 65.3111x1x2^5x3x4 - 56.0309x2^6x3x4 + 102.3173x1^5x3^2x4 + 135.5875x1^4x2x3^2x4 + 76.9260x1^3x2^2x3^2x4 - 13.5137x1^2x2^3x3^2x4 - 100.4228x1x2^4x3^2x4 - 105.4344x2^5x3^2x4 + 35.1780x1^4x3^3x4 + 41.2020x1^3x2x3^3x4 - 3.6048x1^2x2^2x3^3x4 - 86.4994x1x2^3x3^3x4 - 119.0832x2^4x3^3x4 + 10.3238x1^3x3^4x4 + 2.3025x1^2x2x3^4x4 - 44.5137x1x2^2x3^4x4 - 87.7610x2^3x3^4x4 + 1.1315x1^2x3^5x4 - 14.0371x1x2x3^5x4 - 44.1023x2^2x3^5x4 - 2.6123x1x3^6x4 - 15.5330x2x3^6x4 - 3.5153x3^7x4 + 262.3711x1^6x4^2 - 126.4232x1^5x2x4^2 + 4.7792x1^4x2^2x4^2 - 89.9603x1^3x2^3x4^2 + 23.4212x1^2x2^4x4^2 - 21.1358x1x2^5x4^2 + 61.1843x2^6x4^2 - 89.5554x1^5x3x4^2 - 74.3173x1^4x2x3x4^2 - 170.4745x1^3x2^2x3x4^2 - 55.4967x1^2x2^3x3x4^2 - 72.3507x1x2^4x3x4^2 + 85.5217x2^5x3x4^2 + 24.3410x1^4x3^2x4^2 - 114.8027x1^3x2x3^2x4^2 - 42.1625x1^2x2^2x3^2x4^2 - 101.0906x1x2^3x3^2x4^2 + 120.8688x2^4x3^2x4^2 - 30.9054x1^3x3^3x4^2 - 23.2906x1^2x2x3^3x4^2 - 79.1370x1x2^2x3^3x4^2 + 82.6234x2^3x3^3x4^2 + 28.3436x1^2x3^4x4^2 - 34.7211x1x2x3^4x4^2 + 72.6219x2^2x3^4x4^2 - 8.2703x1x3^5x4^2 + 33.2024x2x3^5x4^2 + 31.0307x3^6x4^2 + 474.7801x1^5x4^3 + 326.5877x1^4x2x4^3 + 152.3460x1^3x2^2x4^3 + 54.6197x1^2x2^3x4^3 + 2.9068x1x2^4x4^3 - 8.6576x2^5x4^3 + 204.3578x1^4x3x4^3 + 186.8368x1^3x2x3x4^3 + 96.6558x1^2x2^2x3x4^3 + 6.4647x1x2^3x3x4^3 - 26.2066x2^4x3x4^3 + 65.3455x1^3x3^2x4^3 + 65.8111x1^2x2x3^2x4^3 + 6.5228x1x2^2x3^2x4^3 - 35.0387x2^3x3^2x4^3 + 17.5726x1^2x3^3x4^3 + 4.1005x1x2x3^3x4^3 - 25.4951x2^2x3^3x4^3 + 0.8266x1x3^4x4^3 - 10.7606x2x3^4x4^3 - 2.5383x3^5x4^3 + 214.0616x1^4x4^4 - 168.8956x1^3x2x4^4 - 12.8639x1^2x2^2x4^4 - 82.9423x1x2^3x4^4 + 82.1409x2^4x4^4 - 114.3588x1^3x3x4^4 - 95.7791x1^2x2x3x4^4 - 168.6399x1x2^2x3x4^4 + 82.5304x2^3x3x4^4 + 14.3646x1^2x3^2x4^4 - 118.8212x1x2x3^2x4^4 + 87.8279x2^2x3^2x4^4 - 31.5465x1x3^3x4^4 + 40.4057x2x3^3x4^4 + 41.5105x3^4x4^4 + 509.9118x1^3x4^5 + 277.3872x1^2x2x4^5 + 74.0394x1x2^2x4^5 - 6.1597x2^3x4^5 + 174.9528x1^2x3x4^5 + 93.8800x1x2x3x4^5 - 13.2903x2^2x3x4^5 + 32.4891x1x3^2x4^5 - 9.8985x2x3^2x4^5 - 3.0093x3^3x4^5 + 200.6662x1^2x4^6 - 353.5433x1x2x4^6 + 126.4191x2^2x4^6 - 241.5683x1x3x4^6 + 65.1974x2x3x4^6 + 74.1257x3^2x4^6 + 1.0773e + 03x1x4^7 + 118.4684x2x4^7 + 71.7986x3x4^7 - 0.0548x4^8 + 1.0471e + 03x1^{10} - 294.8721x1^9x2 - 131.8876x1^8x2^2 - 284.2325x1^7x2^3 - 53.9646x1^6x2^4 - 95.5097x1^5x2^5 + 15.0800x1^4x2^6 - 20.2359x1^3x2^7 + 41.8349x1^2x2^8 + 6.1929x1x2^9 +$$

$$44.0009x2^{10}-246.1546x1^{9}x3-239.9259x1^{8}x2x3-557.5429x1^{7}x2^{2}x3-257.7532x1^{6}x2^{3}x3-313.0259x1^{5}x2^{4}x3-80.5002x1^{4}x2^{5}x3-94.6854x1^{3}x2^{6}x3+51.2696x1^{2}x2^{7}x3+20.4916x1x2^{8}x3+83.5515x2^{9}x3-11.6017x1^{8}x3^{2}-363.2798x1^{7}x2x3^{2}-221.5096x1^{6}x2^{2}x3^{2}-411.5605x1^{5}x2^{3}x3^{2}-140.2560x1^{4}x2^{4}x3^{2}-188.6823x1^{3}x2^{5}x3^{2}+80.4621x1^{2}x2^{6}x3^{2}+41.2416x1x2^{7}x3^{2}+183.8698x2^{8}x3^{2}-82.3559x1^{7}x3^{3}-83.9296x1^{6}x2x3^{3}-279.9516x1^{5}x2^{2}x3^{3}-139.6142x1^{4}x2^{3}x3^{3}-223.4046x1^{3}x2^{4}x3^{3}+56.4809x1^{2}x2^{5}x3^{3}+42.3207x1x2^{6}x3^{3}+256.0967x2^{7}x3^{3}+26.2092x1^{6}x3^{4}-100.2571x1^{5}x2x3^{4}-40.6189x1^{4}x2^{2}x3^{4}-169.8808x1^{3}x2^{3}x3^{4}+49.6966x1^{2}x2^{4}x3^{4}+16.4174x1x2^{5}x3^{4}+297.9395x2^{6}x3^{4}-17.3955x1^{5}x3^{5}+1.6346x1^{4}x2x3^{5}-87.5979x1^{3}x2^{2}x3^{5}+25.7888x1^{2}x2^{3}x3^{5}-15.0604x1x2^{4}x3^{5}+245.0054x2^{5}x3^{5}+26.5451x1^{4}x3^{6}-30.9394x1^{3}x2x3^{6}+35.9868x1^{2}x2^{2}x3^{6}-26.9544x1x2^{3}x3^{6}+181.8587x2^{4}x3^{6}-7.0820x1^{3}x3^{7}+19.9245x1^{2}x2x3^{7}-21.5624x1x2^{2}x3^{7}+102.9701x2^{3}x3^{7}+23.6190x1^{2}x3^{8}-10.6197x1x2x3^{8}+68.7360x2^{2}x3^{8}-3.1702x1x3^{9}+31.2975x2x3^{9}+24.3983x3^{10}+1.3118e+03x1^{9}x4+1.2451e+03x1^{8}x2x4+749.3986x1^{7}x2^{2}x4+416.1998x1^{6}x2^{3}x4+209.3495x1^{5}x2^{4}x4+94.4972x1^{4}x2^{5}x4+29.5292x1^{3}x2^{6}x4-0.8553x1^{2}x2^{7}x4-11.1397x1x2^{8}x4-7.0368x2^{9}x4+753.2963x1^{8}x3x4+915.4588x1^{7}x2x3x4+759.6865x1^{6}x2^{2}x3x4+501.3879x1^{5}x2^{3}x3x4+273.9288x1^{4}x2^{4}x3x4+101.7762x1^{3}x2^{5}x3x4-3.4506x1^{2}x2^{6}x3x4-50.1737x1x2^{7}x3x4-34.3754x2^{8}x3x4+285.6347x1^{7}x3^{2}x4+484.0426x1^{6}x2x3^{2}x4+476.9220x1^{5}x2^{2}x3^{2}x4+341.1515x1^{4}x2^{3}x3^{2}x4+155.4489x1^{3}x2^{4}x3^{2}x4-5.6380x1^{2}x2^{5}x3^{2}x4-107.6696x1x2^{6}x3^{2}x4-84.2270x2^{7}x3^{2}x4+109.8593x1^{6}x3^{3}x4+219.4084x1^{5}x2x3^{3}x4+235.8187x1^{4}x2^{2}x3^{3}x4+142.1727x1^{3}x2^{3}x3^{3}x4-1.0916x1^{2}x2^{4}x3^{3}x4-138.2233x1x2^{5}x3^{3}x4-129.8647x2^{6}x3^{3}x4+43.1603x1^{5}x3^{4}x4+94.8470x1^{4}x2x3^{4}x4+85.8055x1^{3}x2^{2}x3^{4}x4+8.2611x1^{2}x2^{3}x3^{4}x4-115.3366x1x2^{4}x3^{4}x4-137.2496x2^{5}x3^{4}x4+18.8001x1^{4}x3^{5}x4+32.6346x1^{3}x2x3^{5}x4+10.2701x1^{2}x2^{2}x3^{5}x4-65.3080x1x2^{3}x3^{5}x4-104.1004x2^{4}x3^{5}x4+5.9441x1^{3}x3^{6}x4+4.5918x1^{2}x2x3^{6}x4-26.5151x1x2^{2}x3^{6}x4-58.8889x2^{3}x3^{6}x4+0.6512x1^{2}x3^{7}x4-8.3669x1x2x3^{7}x4-26.0781x2^{2}x3^{7}x4-1.9016x1x3^{8}x4-9.2209x2x3^{8}x4-2.1612x3^{9}x4+479.7409x1^{8}x4^{2}-129.7670x1^{7}x2x4^{2}+41.8244x1^{6}x2^{2}x4^{2}-106.3779x1^{5}x2^{3}x4^{2}+17.0831x1^{4}x2^{4}x4^{2}-46.7454x1^{3}x2^{5}x4^{2}+34.2367x1^{2}x2^{6}x4^{2}-5.6784x1x2^{7}x4^{2}+51.0936x2^{8}x4^{2}-104.8584x1^{7}x3x4^{2}-44.6854x1^{6}x2x3x4^{2}-222.0849x1^{5}x2^{2}x3x4^{2}-84.7276x1^{4}x2^{3}x3x4^{2}-148.5037x1^{3}x2^{4}x3x4^{2}-12.7304x1^{2}x2^{5}x3x4^{2}-35.1080x1x2^{6}x3x4^{2}+86.1585x2^{7}x3x4^{2}+49.2890x1^{6}x3^{2}x4^{2}-147.3796x1^{5}x2x3^{2}x4^{2}-72.7719x1^{4}x2^{2}x3^{2}x4^{2}-193.7191x1^{3}x2^{3}x3^{2}x4^{2}-32.7166x1^{2}x2^{4}x3^{2}x4^{2}-75.4965x1x2^{5}x3^{2}x4^{2}+155.1909x2^{6}x3^{2}x4^{2}-33.7770x1^{5}x3^{3}x4^{2}-27.1517x1^{4}x2x3^{3}x4^{2}-137.9635x1^{3}x2^{2}x3^{3}x4^{2}-53.8955x1^{2}x2^{3}x3^{3}x4^{2}-99.6789x1x2^{4}x3^{3}x4^{2}+156.4952x2^{5}x3^{3}x4^{2}+32.5209x1^{4}x3^{4}x4^{2}-55.0190x1^{3}x2x3^{4}x4^{2}-2.5629x1^{2}x2^{2}x3^{4}x4^{2}-84.7448x1x2^{3}x3^{4}x4^{2}+145.9554x2^{4}x3^{4}x4^{2}-11.7232x1^{3}x3^{5}x4^{2}+8.7647x1^{2}x2x3^{5}x4^{2}-49.3078x1x2^{2}x3^{5}x4^{2}+90.4472x2^{3}x3^{5}x4^{2}+27.4737x1^{2}x3^{6}x4^{2}-19.8454x1x2x3^{6}x4^{2}+69.1798x2^{2}x3^{6}x4^{2}-5.0018x1x3^{7}x4^{2}+$$

$31.4284x2x3^7x4^2+26.7792x3^8x4^2+621.6310x1^7x4^3+505.8009x1^6x2x4^3+275.4191x1^5x2^2x4^3+$
$141.2129x1^4x2^3x4^3+61.6971x1^3x2^4x4^3+19.6650x1^2x2^5x4^3+2.1155x1x2^6x4^3-0.1963x2^7x4^3+$
$307.2358x1^6x3x4^3+336.1259x1^5x2x3x4^3+250.7507x1^4x2^2x3x4^3+143.8505x1^3x2^3x3x4^3+$
$58.9071x1^2x2^4x3x4^3 + 8.0781x1x2^5x3x4^3 - 0.8110x2^6x3x4^3 + 108.1374x1^5x3^2x4^3 +$
$162.8151x1^4x2x3^2x4^3+135.7852x1^3x2^2x3^2x4^3+72.9916x1^2x2^3x3^2x4^3+11.8475x1x2^4x3^2x4^3-$
$1.9416x2^5x3^2x4^3 + 39.7125x1^4x3^3x4^3 + 65.5018x1^3x2x3^3x4^3 + 51.2702x1^2x2^2x3^3x4^3 +$
$10.2407x1x2^3x3^3x4^3 - 2.9254x2^4x3^3x4^3 + 13.1411x1^3x3^4x4^3 + 20.2786x1^2x2x3^4x4^3 +$
$6.2636x1x2^2x3^4x4^3 - 2.0849x2^3x3^4x4^3 + 3.0535x1^2x3^5x4^3 + 1.7441x1x2x3^5x4^3 -$
$0.9082x2^2x3^5x4^3-0.0105x1x3^6x4^3-0.5683x2x3^6x4^3-0.2265x3^7x4^3+354.5270x1^6x4^4-$
$173.9076x1^5x2x4^4 + 20.9055x1^4x2^2x4^4 - 109.4406x1^3x2^3x4^4 + 33.0015x1^2x2^4x4^4 -$
$23.5176x1x2^5x4^4 + 76.4197x2^6x4^4 - 124.5860x1^5x3x4^4 - 72.5904x1^4x2x3x4^4 -$
$216.7772x1^3x2^2x3x4^4-53.2677x1^2x2^3x3x4^4-88.4653x1x2^4x3x4^4+122.9875x2^5x3x4^4+$
$39.4904x1^4x3^2x4^4-145.3118x1^3x2x3^2x4^4-45.0424x1^2x2^2x3^2x4^4-132.7526x1x2^3x3^2x4^4+$
$174.4420x2^4x3^2x4^4-37.1420x1^3x3^3x4^4-18.2266x1^2x2x3^3x4^4-105.1766x1x2^2x3^3x4^4+$
$125.8199x2^3x3^3x4^4 + 34.9956x1^2x3^4x4^4 - 44.4952x1x2x3^4x4^4 + 100.9224x2^2x3^4x4^4 -$
$10.4051x1x3^5x4^4+45.5495x2x3^5x4^4+36.3399x3^6x4^4+498.2646x1^5x4^5+341.4446x1^4x2x4^5+$
$165.2771x1^3x2^2x4^5 + 69.7009x1^2x2^3x4^5 + 19.4615x1x2^4x4^5 + 4.0135x2^5x4^5 +$
$208.5187x1^4x3x4^5+202.7438x1^3x2x3x4^5+126.0987x1^2x2^2x3x4^5+48.8458x1x2^3x3x4^5+$
$12.6267x2^4x3x4^5 + 67.1793x1^3x3^2x4^5 + 82.3780x1^2x2x3^2x4^5 + 47.1103x1x2^2x3^2x4^5 +$
$16.4851x2^3x3^2x4^5 + 19.6988x1^2x3^3x4^5 + 21.8505x1x2x3^3x4^5 + 11.7089x2^2x3^3x4^5 +$
$3.6219x1x3^4x4^5+4.5499x2x3^4x4^5+0.7144x3^5x4^5+313.5899x1^4x4^6-286.1430x1^3x2x4^6-$
$2.6618x1^2x2^2x4^6 - 107.3658x1x2^3x4^6 + 133.6331x2^4x4^6 - 194.4420x1^3x3x4^6 -$
$106.5621x1^2x2x3x4^6 - 235.0333x1x2^2x3x4^6 + 187.9999x2^3x3x4^6 + 27.5339x1^2x3^2x4^6 -$
$168.5253x1x2x3^2x4^6 + 191.0340x2^2x3^2x4^6 - 42.7780x1x3^3x4^6 + 92.4147x2x3^3x4^6 +$
$58.9857x3^4x4^6+547.8917x1^3x4^7+292.0432x1^2x2x4^7+89.5708x1x2^2x4^7+8.0896x2^3x4^7+$
$180.4107x1^2x3x4^7 + 115.4957x1x2x3x4^7 + 14.7312x2^2x3x4^7 + 38.9382x1x3^2x4^7 +$
$9.4980x2x3^2x4^7+1.9467x3^3x4^7+304.2022x1^2x4^8-630.2108x1x2x4^8+232.7135x2^2x4^8-$
$431.0513x1x3x4^8+195.4685x2x3x4^8+125.1845x3^2x4^8+1.2033e+03x1x4^9+132.3637x2x4^9+$
$80.2202x3x4^9 - 0.0139x4^{10}.$

$-0.1048x1 - 0.6361x2 - 0.6036x3 + 0.0074x4 + 1.9981x1^2 + 0.0032x1x2 + 0.0223x2^2 -$
$0.0253x1x3-0.0511x2x3+0.0238x3^2+0.1683x1x4+0.0081x2x4+0.0116x3x4-3.7141e-$
$04x4^2.$

$0.1048x1 + 0.6361x2 + 0.6036x3 - 0.0074x4 + 1.9981x1^2 + 0.0032x1x2 + 0.0223x2^2 -$
$0.0253x1x3-0.0511x2x3+0.0238x3^2+0.1683x1x4+0.0081x2x4+0.0116x3x4-3.7151e-$
$04x4^2.$

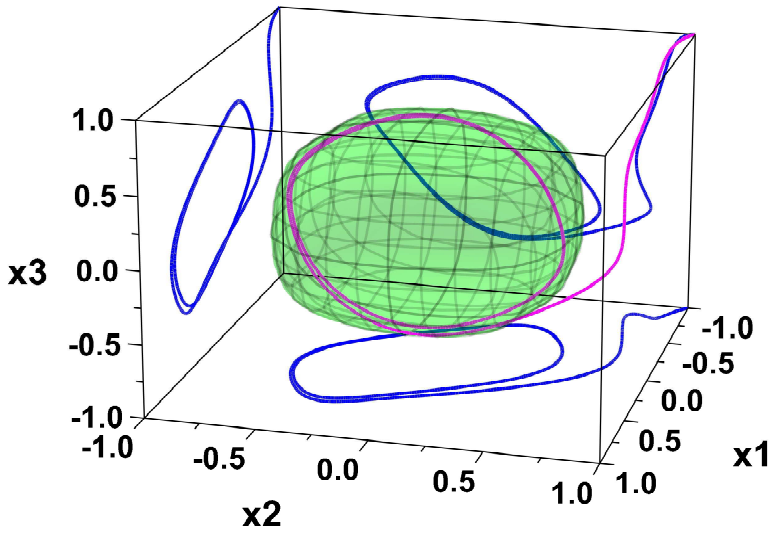
B.4 Odd Stage RO Certificates

Attractive Invariant:
$0.9203860029+0.0190x1^2+0.0190x2^2+0.0190x3^2+0.0751x3^4+0.0751x1^4+0.0751x2^4+$
$0.0065x1x2+0.0065x1x3+0.0065x2x3+0.0046x1^3x2-0.0423x1^2x2^2-0.0143x1x2^3-$
$0.0143x1^3x3-0.0231x1^2x2x3-$
$0.0231x1x2^2x3+0.0046x2^3x3-0.0423x1^2x3^2-0.0231x1x2x3^2-0.0423x2^2x3^2+$
$0.0046x1x3^3-0.0143x2x3^3.$
Escape:
$13.4165x1^2+13.4165x2^2+13.4165x3^2-13.4165x1x2-13.4165x1x3-13.4165x2x3.$
Eventuality:
$2.8838x1^2+2.8836x2^2+2.8979x3^2+2.9322x3^4+3.1808x1^4+3.2589x2^4-2.1068x1x2-$
$2.0682x1x3-2.0671x2x3+29.7895x1^3x2-23.0744x1^2x2^2+24.1883x1x2^3+24.0572x1^3x3-$
$35.3446x1^2x2x3-35.3397x1x2^2x3+29.8880x2^3x3-22.6803x1^2x3^2-34.9999x1x2x3^2-$
$22.9830x2^2x3^2+29.8238x1x3^3+23.7623x2x3^3.$

B.5 Even Stage RO Certificates

Attractive Invariant:
$0.9990128927+1.8133e-04xp1^2+1.8134e-04xp2^2-3.3157e-04xp1^4+2.3598e-$
$04xp1^3xp2-0.0017xp1^2xp2^2-2.3591e-04xp1xp2^3-3.3158e-04xp2^4+4.4111e-$
$04xp1^6-4.4058e-04xp1^5xp2+0.0071xp1^4xp2^2+0.0071xp1^2xp2^4+4.4052e-04xp1xp2^5+$
$4.4113e-04xp2^6-2.1259e-04xp1^8+1.1636e-04xp1^7xp2-0.0084xp1^6xp2^2+8.0223e-$
$05xp1^5xp2^3-0.0200xp1^4xp2^4-8.0579e-05xp1^3xp2^5-0.0084xp1^2xp2^6-1.1639e-$
$04xp1xp2^7-2.1260e-04xp2^8+5.5367e-05xp1^{10}+4.3193e-05xp1^9xp2+0.0031xp1^8xp2^2+$
$8.1649e-05xp1^7xp2^3+0.0129xp1^6xp2^4+0.0129xp1^4xp2^6-8.1366e-05xp1^3xp2^7+$
$0.0031xp1^2xp2^8-4.3166e-05xp1xp2^9+5.5368e-05xp2^{10}.$
Escape:
$0.0124xp1^2+0.0124xp2^2+0.0034xp1^4+0.0034xp2^4.$
Eventuality:
$60.15925038+0.0152xp1^2+0.0149xp2^2-0.0140xp1^4-0.0318xp1^2xp2^2-0.0139xp2^4+$
$0.0259xp1^4xp2^2+0.0258xp1^2xp2^4.$
Lyapunov:
$0.0430xp1^2+0.0430xp2^2-0.0024xp1^4-0.0024xp2^4.$

B.6 Odd Stage RO Attractive Invariant Set



References

[1] Daniel Y Abramovitch. Lyapunov redesign of analog phase-lock loops. *IEEE Transactions on Communications*, 38(12):2197–2202, 1990. 68

[2] Thomas A.Henzinger, Pei-Hsin Ho, and Howard Wong-Toi. Algorithmic analysis of nonlinear hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):540–554, Apr 1998. 13, 36

[3] Amir Ali Ahmadi. *Algebraic relaxations and hardness results in polynomial optimization and Lyapunov analysis*. PhD thesis, Massachusetts Institute of Technology, 2011. 128

[4] Amir Ali Ahmadi, Pablo Parrilo, et al. Towards scalable algorithms with formal guarantees for lyapunov analysis of control systems via algebraic optimization. In *Decision and Control CDC, 2014 IEEE 53rd Annual Conference on*, pages 2272–2281. IEEE, 2014. 129

[5] G. Al-Sammane, M. H. Zaki, Z. J. Dong, and S. Tahar. Towards assertion based verification of analog and mixed signal designs using PSL. In *Proceedings of Languages for Formal Specification and Verification, Forum on Specification and Design Languages (FDL)*, pages 293–298, 2007. 39

[6] Xavier Allamigeon, Stéphane Gaubert, Victor Magron, and Benjamin Werner. Formal proofs for nonlinear optimization. *CoRR*, abs/1404.7282, 2014. 129

[7] Matthias Althoff, Akshay Rajhans, Bruce H. Krogh, Soner Yaldiz, Xin Li, and Larry Pileggi. Formal verification of phase-locked loops using reachability analysis and continuization. In *Proceedings of the International Conference on Computer-Aided Design ICCAD*, pages 659–666. IEEE, 2011. 4, 37, 63, 69

[8] Rajeev Alur. Formal verification of hybrid systems. In *Embedded Software (EMSOFT), 2011 Proceedings of the International Conference on*, pages 273–278. IEEE, 2011. 32

[9] 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. *Theoretical Computer Science*, 138(1):3–34, February 1995. 36

[10] Rajeev Alur and David L Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994. 29

[11] Michaël Armand, Germain Faure, Benjamin Grégoire, Chantal Keller, Laurent Théry, and Benjamin Werner. A modular integration of SAT/SMT solvers to Coq through proof witnesses. In *Certified Programs and Proofs*, volume 7086 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2011. 129

[12] Ashok Balivada, Yatin Hoskote, and Jacob A.Abraham. Verification of transient response of linear analog circuits. In *Proceedings 13th IEEE VLSI Test Symposium*, pages 42–47, April-May 1995. 34

[13] Armin Biere, Alessandro Cimatti, Edmund Clarke, and Yunshan Zhu. *Symbolic Model Checking without BDDs*, volume 1579 of *Lecture Notes in Computer Science*. Springer, 1999. 29

[14] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 131

[15] Stephen P Boyd, Laurent El Ghaoui, Eric Feron, and Venkataramanan Balakrishnan. *Linear matrix inequalities in system and control theory*, volume 15. SIAM, 1994. 27

[16] Michael S Branicky. Studies in hybrid systems: Modeling, analysis, and control. Technical report, DTIC Document, 1995. 13, 22

[17] M.S. Branicky. Stability of switched and hybrid systems. In *Decision and Control CDC, Proceedings of the 33rd IEEE Conference on*, volume 4, pages 3498 – 3503, 1994. 13

[18] Aleksandar Chakarov, Sriram Sankaranarayanan, and Georgios Fainekos. Combining time and frequency domain specifications for periodic signals. In *Runtime*

Verification, volume 7186 of *Lecture Notes in Computer Science*, pages 294–309. Springer, 2012. 108

[19] Man-Duen Choi, Tsit Yuen Lam, and Bruce Reznick. Sums of squares of real polynomials. In *Proceedings of Symposia in Pure mathematics*, volume 58, pages 103–126. American Mathematical Society, 1995. 24

[20] Edmund Clarke, Armin Biere, Richard Raimi, and Yunshan Zhu. *Bounded Model Checking Using Satisfiability Solving*. Kluwer Academic Publishers, 2001. 37

[21] Edmund Clarke, Alexandre Donzé, and Axel Legay. On simulation-based probabilistic model checking of mixed-analog circuits. *Formal Methods in System Design*, 36:97 – 113, 2010. 39

[22] Edmund M Clarke and E Allen Emerson. *Design and synthesis of synchronization skeletons using branching time temporal logic*, volume 131 of *Lecture Notes in Computer Science*. Springer, 1982. 29

[23] Edmund M Clarke, Orna Grumberg, and Doron Peled. *Model Checking*. MIT press, 1999. 29

[24] George E Collins. *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, volume 33 of *Lecture Notes in Computer Science*. Springer, 1998. 34

[25] Wikipedia contributors. Pentium fdiv bug. Wikipedia, The Free Encyclopedia. 2

[26] Wikipedia contributors. Therac- 25. Wikipedia, The Free Encyclopedia., February 2007. 2

[27] C.Yan, M.Greenstreet, and Jochen Eisinger. Formal verification of an arbiter circuit. In *IEEE Symposium on Asynchronous Circuits and Systems (ASYNC)*, pages 165 – 175, May 2010. 38

[28] T. R. Dastidar and P. P Chakrabarti. A verification system for transient response of analog circuits using model checking. In *VLSI Design (VLSID)*, pages 195–200. IEEE Computer Society Press, 2005. 39

[29] Thomas A DeMassa and Zack Ciccone. *Digital integrated circuits*. Wiley New York, 1996. 74

[30] William Denman, Behzad Akbarpour, Sofiene Tahar, Mohamed H Zaki, and Lawrence C Paulson. Formal verification of analog designs using metitarski. In *Formal Methods in Computer-Aided Design, 2009. FMCAD 2009*, pages 93–100. IEEE, 2009. 38, 39, 40, 120, 121

[31] Andreas Dolzmann and Thomas Sturm. Redlog: Computer algebra meets computer logic. *ACM SIGSAM Bulletin*, 31(2):2–9, 1997. 94

[32] Alexandre Donzé, Oded Maler, Ezio Bartocci, Dejan Nickovic, Radu Grosu, and Scott Smolka. On temporal logic and signal processing. In *Automated Technology for Verification and Analysis*, pages 92–106. Springer, 2012. 121

[33] Andreas Eggers, Martin Fränzle, and Christian Herde. SAT modulo ODE: A direct SAT approach to hybrid systems. In *Automated Technology for Verification and Analysis*, volume 5311 of *Lecture Notes in Computer Science*, pages 171–185. Springer, 2008. 30, 114, 117

[34] Ahmed S Elwakil and Khaled N Salama. On the nonlinear modeling of ring oscillators. *Journal of Circuits, Systems, and Computers*, 18(04):681–696, 2009. 74

[35] Fulvio Forni. *Analysis of Hybrid Systems and Design of Hybrid Controllers*. PhD thesis, Università di Roma, 2010. 14

[36] Goran Frehse, Bruce H Krogh, and Rob A Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, pages 257–262. European Design and Automation Association, 2006. 36

[37] Goran Frehse, Bruce H Krogh, and Rob A Rutenbar. Verifying analog oscillator circuits using forward/backward abstraction refinement. In *Proceedings of the conference on Design, automation and test in Europe: Proceedings*, pages 257–262. European Design and Automation Association, 2006. 114

[38] Goran Frehse, Bruce H Krogh, Rob A Rutenbar, and Oded Maler. Time domain verification of oscillator circuit properties. *Electronic Notes in Theoretical Computer Science*, 153(3):9–22, 2006. 36, 39, 104, 120

[39] 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*, volume 6806 of *Lecture Notes in Computer Science*, pages 379–395. Springer, 2011. 69

[40] Xiaoqing Ge, Murat Arcak, and Khaled Nabil Salama. Nonlinear analysis of ring oscillator and cross-coupled oscillator circuits. *Dynamics of Continuous, Discrete and Impulsive Systems Series B: Applications and Algorithms*, pages 959–977, 2010. 74

[41] A. Ghosh and R Vemuri. Formal verification of synthesized analog designs. In *Proceedings of International Conference on Computer Design ICCD*, pages 40–45. IEEE Computer Society Press, 1999. 38

[42] Rafal Goebel, Ricardo G Sanfelice, and Andrew R Teel. *Hybrid dynamical systems: modeling, stability, and robustness*. Princeton University Press, 2012. 13, 21

[43] Mark R Greenstreet and Suwen Yang. Verifying start-up conditions for a ring oscillator. In *Proceedings of the 18th ACM Great Lakes symposium on VLSI*, pages 201–206. ACM, 2008. 97

[44] M.R. Greenstreet and S. Yang. Verifying start-up conditions for a ring oscillator. In *18th Great Lakes Symposium on VLSI (GLSVLSI'08)*, pages 201–206. ACM, May 2008. 39, 121

[45] S. Gupta, B.H. Krogh, and R.A. Rutenbar. Towards formal verification of analog designs. In *International Conference on Computer Aided Design ICCAD*, pages 210–217, San Jose,CA (USA), November 7-11 2004. IEEE/ACM. 36, 39, 120

[46] K Hanna. Reasoning about real circuits. In *Higher Order Logic Theorem Proving and Its Applications TPHOLs*, volume 859 of *Lecture Notes in Computer Science*, pages 235–253. Springer, 1994. 38

[47] John Harrison. Verifying nonlinear real formulas via sums of squares. In *Theorem Proving in Higher Order Logics*, volume 4732 of *Lecture Notes in Computer Science*, pages 102–118. Springer, 2007. 34, 98, 129

[48] Walter Hartong, Ralf Klausen, and Lars Hedrich. *Formal Verification for Nonlinear Analog systems:Approaches to Model and Equivalence Checking*, chapter 6, pages 205–243. Kluwer Academic Publishers, Netherland, 2004. 35

[49] Lars Hedrich and Erich Barke. A formal approach to verification of linear analog circuits wth parameter tolerances. In *Proceedings of the conference on Design, automation and test in Europe*, pages 649–655. IEEE Computer Society, 1998. 34, 40, 121

[50] http://z3.codeplex.com. 115

[51] Hiroyuki Ichihara and Hirokazu Anai. An SOS-QE approach to nonlinear gain analysis for polynomial dynamical systems. *Mathematics in Computer Science*, 5(3):303–314, 2011. 33, 34, 98

[52] Daisuke Ishii, Kazunori Ueda, and Hiroshi Hosobe. An interval-based SAT modulo ODE solver for model checking nonlinear hybrid systems. *Software Tools for Technology Transfer*, 13:449–461, 2011. 38

[53] Zachary William Jarvis-Wloszek. *Lyapunov based analysis and controller synthesis for polynomial systems using sum-of-squares optimization.* PhD thesis, University of California, 2003. 27, 28

[54] Alexander Jesser, Stefan Laemmermann, Roland Weiss, Alexander Pacholik, Lars Hedrich, Juergen Ruf, Thomas Kropf, Wolfgang Fengler, and Wolfgang Rosenstiel. Analog simulation meets digital verification-a formal assertion approach for mixed-signal verification. In *Synthesis And System Integration of Mixed Information Technologies*, 2007. 39

[55] Mikael Johansson and Anders Rantzer. Computation of piecewise quadratic lyapunov functions for hybrid systems. *IEEE transactions on automatic control*, 43(4):555–559, 1998. 22

[56] Kevin D Jones, Jeha Kim, and V Konrad. Some real world problems in the analog and mixed signal domains. In *Designing Correct Circuits*, 2008. 2, 39, 71

[57] Saurabh K Tiwary, Anubhav Gupta, Joel R Phillips, Claudio Pinello, and Radu Zlatanovici. First steps towards SAT-based formal analog verification. In *International Conference on Computer-Aided Design*, pages 1–8, San Jose, California, USA, November 2009. 37, 39, 97, 121

[58] Erich Kaltofen, Bin Li, Zhengfeng Yang, and Lihong Zhi. Exact certification of global optimality of approximate factorizations via rationalizing sums-of-squares with floating point scalars. In *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, pages 155–164. ACM, 2008. 129

[59] Erich L Kaltofen, Bin Li, Zhengfeng Yang, and Lihong Zhi. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. *Journal of Symbolic Computation*, 47(1):1–15, 2012. 129

[60] Gustav Kirchhoff. Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Vertheilung galvanischer Ströme geführt wird. *Annalen der Physik*, 148(12):497–508, 1847. 16

[61] Hassan K.Khalil. *Nonlinear Systems*. Prentice Hall, third edition, 2002. 19, 49, 79, 81

[62] Kenneth S Kundert and Alberto Sangiovanni-Vincentelli. Finding the steady-state response of analog and microwave circuits. In *Custom Integrated Circuits Conference, 1988., Proceedings of the IEEE 1988*, pages 6–1. IEEE, 1988. 107

[63] Robert P. Kurshan and K. L. McMillan. Analysis of digital circuits through symbolic reduction. *IEEE Transaction on Computer Aided Design of Integrated Circuits and Systems*, 10(11):1356–1371, November 1991. 35

[64] Avraham Levkovich, Ezra Zeheb, and Nir Cohen. Frequency response envelopes of a family of uncertain continuous-time systems. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 42(3):156–165, 1995. 40, 121

[65] Daniel Liberzon. *Switching in systems and control*. Springer Science & Business Media, 2012. 21

[66] Honghuang Lin, Peng Li, and Chris J Myers. Verification of digitally-intensive analog circuits via kernel ridge regression and hybrid reachability analysis. In *Proceedings of the 50th Annual Design Automation Conference*, page 66. ACM, 2013. 4, 68

[67] Johan Lofberg. Yalmip: A toolbox for modeling and optimization in matlab. In *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pages 284–289. IEEE, 2004. 63, 94

[68] John Lygeros, Karl Henrik Johansson, Slobodan N. Simić, Jun Zhang, and S. Shankar Sastry. Dynamical properties of hybrid automata. *IEEE TRANSACTIONS ON AUTOMATIC CONTROL*, 48(1):2–17, January 2003. 13

[69] Oded Maler. *Algorithmic Verification of Continuous and Hybrid Systems*, volume 140 of *EPTCS*, pages 48–69. 2014. 30

[70] Oded Maler and Dejan Nickovic‘. Monitoring properties of analog and mixed-signal circuits. *Software Tools for Technology Transfer*, 15(3):247–268, June 2013. 39

[71] MATLAB. *Version 8.2 (R2013b)*. The MathWorks Inc., 2013. 114

[72] Paolo Nenzi and Holger Vogt. Ngspice users manual version 23, 2011. 18, 101

[73] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology Pasadena, California, 2000. 23, 24, 25, 26, 27

[74] Stefan Pettersson and Bengt Lennartson. Stability and robustness for hybrid systems. In *Decision and Control, 1996., Proceedings of the 35th IEEE Conference on*, volume 2, pages 1202–1207. IEEE, 1996. 22

[75] Helfried Peyrl and Pablo A Parrilo. Computing sum of squares decompositions with rational coefficients. *Theoretical Computer Science*, 409(2):269–281, 2008. 129

[76] Amir Pnueli. The temporal logic of programs. In *Foundations of Computer Science, 1977., 18th Annual Symposium on*, pages 46–57. IEEE, 1977. 29

[77] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 477–492. Springer, 2004. 32, 98

[78] Stephen Prajna and Antonis Papachristodoulou. Analysis of switched and hybrid systems-beyond piecewise quadratic methods. In *American Control Conference, 2003. Proceedings of the 2003*, volume 4, pages 2779–2784. IEEE, 2003. 24, 54

[79] Stephen Prajna, Antonis Papachristodoulou, Pablo Parrilo, et al. Introducing sostools: A general purpose sum of squares programming solver. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 1, pages 741–746. IEEE, 2002. 26

[80] Stephen Prajna and Anders Rantzer. Convex programs for temporal verification of nonlinear dynamical systems. *SIAM Journal on Control and Optimization*, 46(3):999–1021, 2007. 78, 80

[81] Mark R. Greenstreet and Ian Mitchell. Integrating projections. In *First International Workshop, HSCC'98*, pages 159–174, 1998. 35

[82] A. Salem. Semi-formal verification of VHDL-AMS descriptions. In *IEEE International Symposium on Circuits and Systems ISCAS.*, volume 5, pages 333–336, 2002. 35

[83] Ghiath Al Sammane, Mohamed H. Zaki, and Sofiéne Tahar. A symbolic methodology for the verification of analog and mixed signal designs. In *DATE*, pages 249–254. ACM, 2007. 38

[84] Sriram Sankaranarayanan, Henny B Sipma, and Zohar Manna. Constructing invariants for hybrid systems. In *Hybrid Systems: Computation and Control*, volume 32, pages 539–554. Springer, 2004. 34

[85] Zhikun She and Bai Xue. Algebraic analysis on asymptotic stability of switched hybrid systems. In *Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control*, pages 187–196. ACM, 2012. 34, 98

[86] S. Steinhorst and L. Hedrich. Trajectory-directed discrete state space modeling for formal verification of nonlinear analog circuits. In *Computer-Aided Design ICCAD, 2012 IEEE/ACM International Conference on*, pages 202–209, Nov 2012. 97

[87] S. Steinhorst, M. Peter, and L. Hedrich. State space exploration analog circuits by visualized multi-parallel particle simulation. In *International Conference on Signal Processing Systems (ICSPS'09)*, pages 858–862, Washington, DC, USA, May 2009. IEEE Computer Society. 37, 39, 97, 120

[88] Jos F Sturm. Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 1999. 63

[89] Thomas Sturm. *Real Quantifier Elimination in Geometry*. Fak. für Math. und Inf., 1999. 33

[90] Thomas Sturm and Ashish Tiwari. Verification and synthesis using real quantifier elimination. In *Proceedings of the 36th international symposium on Symbolic and algebraic computation*, pages 329–336. ACM, 2011. 33, 98

[91] Ankur Taly and Ashish Tiwari. Deductive verification of continuous dynamical systems. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 4. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009. 33, 98

[92] Weehong Tan. *Nonlinear Control Analysis and Synthesis using Sum-of-Squares Programming*. PhD thesis, University of California, Berkeley, 2006. 78, 79, 98, 128, 129

[93] Alfred Tarski. A decision method for elementary algebra and geometry. 1951. 24

[94] Dang Thao, Alexandre Donzé, and Oded Male. Verification of analog and mixed-signal circuits using hybrid system techniques. In Alan J. Hu and Andrew K.Martin, editors, *FMCAD*, volume 3312 of *Lecture Notes in Computer Science*, pages 21–36. Springer, 2004. 36

[95] Ashish Tiwari and Gaurav Khanna. Nonlinear systems: Approximating reach sets. In *Hybrid Systems: Computation and Control*, volume 2993 of *Lecture Notes in Computer Science*, pages 600–614. Springer, 2004. 34

[96] Yannis P.. Tsividis and Colin McAndrew. *Operation and Modeling of the MOS Transistor*. Oxford University Press, 2011. 17

[97] Hafiz Ul Asad and Kevin D Jones. Inevitability of phase-locking in a charge pump phase lock loop using deductive verification. In *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pages 295–300. ACM, 2015. 9

[98] Hafiz Ul Asad and Kevin D Jones. Verifying inevitability of phase-locking in a charge pump phase lock loop using sum of squares programming. In *Proceedings of the 25th edition on Great Lakes Symposium on VLSI*, pages 295–300. ACM, 2015. 9

[99] Hafiz Ul Asad, Kevin D Jones, and Frederic Surre. Verifying robust frequency domain properties of non linear oscillators using smt. In *Design and Diagnostics of Electronic Circuits & Systems, 17th International Symposium on*, pages 306–309. IEEE, 2014. 9

[100] D. Walter, S. Little, N. Seegmiller, C. J. Myers, and T Yoneda. Symbolic model checking of analog/mixed-signal circuits. In *Proc. of Asia and South Pacific Design Automation Conference ASPDAC*, pages 316–323. IEEE Computer Society, 2007. 37

[101] David Walter, Scott Little, and Chris Myers. Bounded model checking of analog and mixed-signal circuits using an SMT solver. In Kedar S. Namjoshi, Tomohiro Yoneda, Teruo Higashino, and Yoshio Okamura, editors, *Proceedings of the 5th*

international conference on Automated technology for verification and analysis ATVA, volume 4762 of *Lecture Notes in Computer Science*, pages 66–81, Berlin, 2007. Springer-Verlag. 37

[102] T Wang, Sanjay Lall, and Matthew West. Polynomial level-set method for polynomial system reachable set estimation. *Automatic Control, IEEE Transactions on*, 58:2508 – 2521, 2013. 28, 30, 32, 59

[103] Zuoding Wang. An analysis of charge-pump phase-locked loops. *Circuits and Systems I: Regular Papers, IEEE Transactions on*, 52(10):2128–2138, 2005. 17, 42

[104] Andreas Weber. Quantifier elimination on real closed fields and differential equations. *Algebra, Logic, Set Theory–Festschrift für Ulrich Felgner zum*, 65:291–315. 98

[105] Jijie Wei, Yan Peng, Ge Yu, and Mark Greenstreet. Verifying global convergence for a digital phase-locked loop. In *Formal Methods in Computer-Aided Design FMCAD, 2013*, pages 113–120. IEEE, 2013. 4, 68, 69

[106] Volker Weispfenning. *A new approach to quantifier elimination for real algebra.* Springer, 1998. 34

[107] Moris W.Hirsch and Stephen Smale. *Differential Equations, Dynamical Systems, and Linear Algebra.* Academic Press, INC., San Diego, CA, 1974. 13

[108] C. Yan and Mark Greensreet. Oscillator verification with probablity one. In *FMCAD*, 12, pages 165–172, Cambridge, October 2012. 40, 121

[109] Chao Yan, Mark R Greenstreet, and Suwen Yang. Verifying global start-up for a Möbius ring-oscillator. *Formal Methods in System Design*, 45(2):246–272, 2014. 77, 96, 97

[110] Leyi Yin, Yue Deng, and Peng Li. Verifying dynamic properties of nonlinear mixed-signal circuits via efficient smt-based techniques. In *International Conference on Computer-Aided Design ICCAD*, pages 436–442. IEEE, November 2012. 38

[111] M. H. Zaki, S. Tahar, and G. Bois. A practical approach for monitoring analog circuits. In *In ACM Great Lakes Symposium on VLSI (GLS-VLSI)*, pages 330–335, 2006. 39

[112] M. H. Zaki, S. Tahar, and G. Bois. Combining symbolic simulation and interval arithmetic for the verification of ams designs. In *Formal Methods for Computer Aided Design FMCAD*, pages 207–215. IEEE Computer Society Press, 2007. 37

[113] Mohamed H Zaki, Sofiène Tahar, and Guy Bois. Formal verification of analog and mixed signal designs: A survey. *Microelectronics Journal*, 39(12):1395–1404, 2008. 34