Mingshuai Chen · ✉ chenms@cs.rwth-aachen.de

I am a postdoctoral researcher at the Software Modeling and Verification Group headed by Prof. Dr. Joost-Pieter Katoen, at RWTH Aachen, associated with the ERC Advanced Grant Project FRAPPANT: Formal Analysis of Probabilistic Programs. *My research interest lies in the general scope of formal verification and synthesis, broadly construed in mathematical logic and theoretical computer science. In particular, I am interested in formal reasoning of programs and hybrid discrete-continuous systems, for ensuring the reliability and effectiveness of safety-critical software systems.* I appreciate —and have been constantly amazed by— the elegance of mathematical aspects of computer science, as well as its impact on real-world applications.

Conventional computer systems have over the past few decades vividly evolved into an open, interconnected form, now known as *cyber-physical systems* (CPS), that integrates capabilities of computing, communication, and control, thereby triggering yet another round of global revolution of the information technology. The underlying safety-critical feature, however, impels the community to tackle the grand challenge concerning analysis, verification, and design of reliable CPS. *Hybrid systems* that seamlessly integrate continuous dynamics with discrete programs have been extensively used as mathematical models for CPS, wherein prominent formal techniques, e.g., reachability-based model checking, deduction-based theorem proving and correct-by-construction synthesis, have been developed for ensuring correctness of hybrid systems.

This research statement first identifies in Section A several key challenges in the verification and design of safety-critical CPS, and then sketches out in Section B my, and my colleagues', efforts over the recent years in tackling these challenges, particularly addressing the inherent features thereof like intricacy, randomness, and time-delayed behaviors. The statement is concluded by Section C with a list of future research projects consisting of key research questions, promising approaches, their risk assessments, as well as the major anticipated research results.

A Challenges in Designing Safety-Critical CPS

> *"How can we provide people with CPS they can bet their lives on?"*
>
> — Jeannette M. Wing, former AD for CISE at NSF

Cyber-physical systems have witnessed an increasing number of safety-critical applications particularly in major scientific projects. Prominent examples include automotive electronics, health care, nuclear reactors, high-speed transportations, manned spaceflight, etc., in which a malfunction of any software or hardware component would potentially lead to catastrophic consequences like significant casualties and economic losses. In the meantime, with the rapid development of computer control, sensor techniques, and AI/ML algorithms, inevitable features, e.g., heavy intricacy, probabilistic/nondeterministic behaviors, and time delays have become increasingly essential underneath both the continuous evolution of physical plants and the discrete transition of computer programs, which may well annihilate the safety certificate and control performance of CPS.

Traditional engineering methods, e.g., testing and simulations, are nevertheless argued insufficient for the zero-tolerance of failures incurred in safety-critical systems. Therefore, how to rigorously verify and design reliable CPS involving the aforementioned features tends to be a formidable challenge in computer science and the control community. More specifically, the following key challenges have been commonly agreed on by the community:

- *Real-time:* the system should not only be logically correct, but also satisfy physical time constraints;
- *Hybridization*: a deep coupling between the discrete computing process and the continuous physical dynamics;
- *Intricacy*: an incremental integration of a mass of complex, heterogeneous subcomponents;
- *Uncertainty*: the system is often deployed in an open, dynamic, distributed environment subject to multiple forms of uncertainty, e.g., nondeterminism, randomness, time delays, etc.;
- *Efficiency and cost*: the abovementioned factors complicate the entire software life-cycle, thus yielding increasingly severe issues in both efficiency and cost.

In response to these challenges, formal methods have been acknowledged as effective techniques to guarantee the correctness and performance of hardware/software systems. *My research aims to develop techniques for designing and verifying safety-critical cyber-physical systems based on rigorous mathematical models and formal reasoning.* This amounts to, as depicted in Figure 1, either *verifying* (or refuting, otherwise) that the system satisfies a given property (specified in a certain logic) in its environment, or *synthesizing* a correct-by-construction system model that satisfies the property when being composed with the environment. Meanwhile, *I devote a significant amount of efforts to pushing the limits of automation as far as possible such that the procedure can be fully (or partially) mechanized*, and hence benefits from various well-established techniques, e.g., convex optimization and SAT/SMT-solving, in computer-aided verification and synthesis.

B Research Achievements – Towards Theories of Designing Safety-Critical CPS

This section summarizes my major research outcomes in a topic-wise manner. It demonstrates how I approached the challenges (RT = real-time, HD = hybridization, IT = intricacy, UC = uncertainty, EC = efficiency & cost) through different lenses, and worked towards a spectrum of diverse, yet interconnected techniques for the design of safety-critical CPS.

Figure 1: Illustration of formal verification and synthesis: A verifier generates as output a YES/NO answer, indicating whether the systems S satisfies the property Φ in its environment E; In contrast, a synthesizer generates a system model S from a class of systems that satisfy Φ when composed with E.

Decidability of Reachability for Solvable Dynamical Systems. The safety-critical feature of CPS —with increasingly complex behaviors— has initiated automatic safety or, dually, reachability verification of hybrid systems. This amounts to answering questions like "will the robot (eventually) grasp the target while avoiding the traps?" Alur et al. showed in the mid-1990s that the reachability problem of hybrid systems is undecidable in general. The hardcore of the verification problem lies in reasoning about the continuous dynamics, which are often characterized by ordinary differential equations (ODEs), or vector fields. I was intrigued by the question "is there a (preferably large) fragment of vector fields which admits a decidable reachability problem?" RT

I co-identified two families of linear vector fields with possibly transcendental inputs and proved the decidability result by leveraging a reduction to (the extension of) Tarski's algebra [GCD+15, GCL+16]. The decidable families were later extended to the nonlinear case featuring the so-called solvable dynamics [GCL+18, FCK19]. *These families of vector fields constitute a fragment of continuous differential dynamics which has been well recognized as hitherto the most expressive one whose reachability problem is proven decidable.* Based on the reduction, I developed a novel decision procedure called LinR to decide reachability, and further proposed a tight abstraction of general solvable dynamics. Our recent work [WCX+21, WCX+22] followed by proposing a weakly complete method on synthesizing inductive invariants via difference-of-convex programming for dynamics beyond solvability. According to the reviewer at CAV '21, *"the quality, quantity, and thoroughness of the work are very impressive."*

Verification and Synthesis of Time-Delayed Dynamical Systems. Time-delayed behaviors are omnipresent, especially in computer-controlled CPS. For instance, how can we safely steer a self-driving vehicle in case its observations via numerous sensors are subject to small, yet critical time delays? In contrast to delay-free systems, time-delayed systems yield substantially higher theoretical complexity (i.e., leaving from a finite state space to an infinite functional domain), thus rendering the design and verification tasks notoriously harder. I took this topic for my doctoral dissertation, back to when automatic synthesis (or even verification) of time-delayed systems was still in its infancy. RT, UC, EC

Together with my co-authors, I developed safety games played under discrete delays to model delayed interactions between a controller and its environment [CFL+18, CFL+21]. The controllability problem of such games was proven decidable w.r.t. safety objectives via an exponential reduction to delay-free games. I further developed an incremental technique for synthesizing delay-resilient control that mitigates the effect of exponential blow-up and proved the equivalence of qualitative controllability under different delay patterns, thereby fulfilling the request of efficient synthesis due to Tripakis and others. *This line of work received the Distinguished Paper Award at ATVA 2018, and according to the reviewer, "[it] provides a much needed applied vantage point, and is a first step towards practical studies."*

Delayed coupling between state variables in continuous dynamics is often modelled as delay differential equations (DDEs). I co-developed a validated simulation-based method [CFL+16] and a boundary propagation-based method [XMF+17] for automatic bounded-time verification of dynamical systems governed by DDEs. These results were further generalized to unbounded-time verification by leveraging spectral analysis and linearization [FCZ+19]. *These findings enabled fully automatic verification of delayed differential dynamics, and according to the reviewer, "this paper clearly deserves a space [at CAV]."*

Our results in the verification and synthesis of time-delayed dynamical systems have catalyzed a mounting interest in the communities of CPS and computer-aided verification. *In recognition of my contributions, I received the CAS-President Special Award in 2019 (as the first awardee from ISCAS ever since its inception in 1985), which is the most prestigious award for graduate students across over a hundred institutes in Chinese Academy of Sciences (CAS); My Ph.D. thesis [Che19] on this topic has been nominated for the CAS Excellent Doctoral Dissertation Award; and I was invited to co-present a tutorial on the same topic at RTSS 2020.*

Verification of Probabilistic and Nondeterministic Systems. Our universe is stochastic, uncertain, and even chaotic. It is hardly the case —if not impossible— that CPS can be deployed in a perfectly predictable environment. Meanwhile, randomness emerges as a key feature in security mechanisms, randomised algorithms, and is rapidly encroaching AI. Therefore, probabilistic and nondeterministic behaviors have undergone a recent surge of interest among computer scientists. Prominent models include multi-path programs encoding nondeterminism, probabilistic programs featuring probabilistic choices and conditioning, and stochastic differential equations (SDEs) modelling continuous dynamics coupled with Gaussian white noises. Reasoning over these models amounts to answering quantitative questions like "what is the expected value when the program terminates?" "what is the probability that the program terminates?" and "what is the probability that the SDE system visits a bad state?" RT, UC, EC

In cooperation with my colleagues, I developed a constructive method, based on the synthesis of stochastic barrier certificates, for computing an exponentially decreasing upper bound on the tail probability that an SDE system violates a given safety specification over an infinite time horizon [FCX+20]. I co-developed techniques for proving decidability of the termination problem for a family of multi-path polynomial programs (MPPs) [LZC+22]. We presented an explicit recursive function, which is essentially Ackermannian, to compute the maximal length of ascending chains of polynomial ideals under a control function, and thereby obtained a complete answer to the questions raised by Seidenberg. *The identified family of MPPs is hitherto the largest one for which termination is known decidable.*

More recently, I co-developed κ-induction [BCK+21], a generalization of classical k-induction to arbitrary complete lattices, and —together with a complementary bounded model-checking approach— obtained a fully automated technique for verifying infinite-state probabilistic programs. This research work went through a clear acceptance to the CAV community, and thereafter, Satnam Singh, one of the key contributors to the seminal work on k-induction, twittered that *"good to see that k-induction lives on! A nice talk at CAV 2021 [...]"* Our recent work following this line of research —generatingfunctionology-based verification of probabilistic programs [CKKW22] and inductive synthesis of inductive invariants [BCJ+22]— are currently under review.

Synthesis of Nonlinear Craig Interpolants. Scalability is clearly a bottleneck of existing formal verification techniques due to the pivotal challenges of intricacy and efficiency. Craig interpolation has been recognized as a promising technique in scaling these verification techniques, e.g., theorem proving, model checking, and abstract interpretation, due to its inherent capability of local and modular reasoning. An interpolant serves as a witness of the unsatisfiability of two logical formulas. How to efficiently synthesize interpolants, especially for nonlinear theories, is a central but difficult task in interpolation-based techniques. IT, EC

Based on a key observation that quadratic polynomial inequalities can be linearized if they are concave, I co-developed a polynomial-time algorithm for generating interpolants for the combined theory of concave quadratic polynomial inequalities and the equality theory over uninterpreted functions [GDX+16]. *This is the first polynomial-time algorithm for generating interpolants over nonlinear theories via a reduction to semi-definite programming.* To surmount the limitation on concave-quadratic polynomial theories, I further developed NIL [CWA+19], a unified, counterexample-guided method for synthesizing polynomial interpolants over the general quantifier-free theory of nonlinear arithmetic, possibly with transcendental functions.

Modelling, Verification, and Synthesis of Hybrid Systems. I co-developed MARS [CHT+17, CRW+16], a toolchain for modelling, analyzing, and verifying hybrid systems. MARS integrates the aforementioned techniques into a unified framework that supports hierarchical modelling, compositional reasoning, and refinement. *This toolchain has been successfully applied in the verification of control programs of the Chinese lunar lander Chang'e-3 and the Chinese high-speed railway system.* I co-developed an active learning method for deterministic timed systems [ACZ+20], which received the *Best Paper Award at FMAC 2019* and has been selected as a *High-Impact Publication in CS by Chinese researchers across from Springer Nature*. Moreover, I co-developed NAPL [WAC+20], a domain-specific language that enables rapid network-algorithm programming. HD

C Future Research Projects and Plans

Despite the aforementioned achievements towards theories of designing safety-critical CPS, there remains a spectrum of fundamental research problems along the way. In particular, I am interested to establish my future research program within three main tracks: (I) *verification and synthesis of probabilistic programs*, (II) *verification and identification of hybrid systems*, and (III) *(semi-)automation, tool support, and applications*. I brief as follows for each track a list of key research questions, promising approaches, their risk assessments (HR = high risk, LR = low risk), as well as the major anticipated research outcomes.

Track I Verification and Synthesis of Probabilistic Programs[1]

Key Research Questions

I-1. How to decide if two probabilistic programs are equivalent? What does (probabilistic) equivalence even mean?
I-2. How to combine forward- and backward-reasoning of probabilistic programs?
I-3. How to synthesize (k-)inductive invariants and witnesses of almost-sure termination for probabilistic programs?
I-4. How to synthesize (or repair) probabilistic programs against given specifications?

Promising Approaches

Deciding Equivalence Using Generating Functions. In order to answer question I-1 on probabilistic equivalence, we first have to assign meanings (aka, semantics) to probabilistic programs. There is a variety of operational or denotational semantics existing in the literature which differ from each other in many aspects, e.g., generality, simplicity, and reasoning directions. I plan to develop a forward denotational semantics based on generating-function (GF) representations of (sub-)distributions, where HR

[1] This research track can be viewed as a derivative of the FRAPPANT project, with a focus on related but significantly orthogonal problems and/or approaches.

a probabilistic program (possibly with continuous random variables) acts as a transformer that transforms an input GF into an output GF. The GF semantics shines because there are often closed-form representations of possibly infinite-support distributions. I will then define a proper notation of program equivalence, e.g., two programs are equivalent iff they transform every possible input GF into the same output GF. The aim is to show a potential decidability result by reducing the program-equivalence problem to the problem of checking equalities of GFs over a decidable theory fragment. The key challenges include obtaining closed-form GFs capturing loop executions and reasoning about infinitely-many input GFs in one shot. This line of research shall also produce interesting insights on (infinite-state) probabilistic model checking, i.e., to check whether a (possibly infinite-state) probabilistic program P meets its specification S: the specification S can be encoded as a probabilistic program P_S, and hence the model-checking problem boils down to checking the equivalence of P and P_S.

Combining Forward- and Backward-Reasoning via Craig Interpolation. For non-probabilistic programs, promising methods have been developed to combine forward- and backward-style reasoning, yielding many interesting results in program analysis and verification, e.g., modular/compositional reasoning and automated invariant generation. The intuition is that one can squeeze a candidate inductive invariant based on Craig interpolants between forward and backward predicate transformers, i.e., strongest postconditions (forward) and weakest preconditions (backward). However, extending this idea to probabilistic programs is extremely non-trivial: Despite the well-known weakest-preexpectation calculus for probabilistic programs, the dual notion of strongest postexpectation is, unfortunately, ill-defined; moreover, Craig interpolation works only in the qualitative setting (where a formula is either true or false). I plan to address these challenges by (1) developing a quantitative extension of Craig interpolation, (2) exploiting the distribution transformers (e.g., the GF transformers as described above) as the forward semantics, and (3) connecting it with the backward weakest-preexpectation semantics via the so-obtained quantitative Craig interpolants. As explained above, this line of research will also bring insights to the research question I-3 on (quantitative) invariant synthesis. HR

Synthesizing Loop Invariants, Termination Proofs, and Programs. Recall Figure 1: the synthesis problem is arguably significantly harder than the corresponding verification problem, since from a logical perspective the encoding of a synthesis problem often has one more level of quantification (in fact, quantifier alternation) than that of the verification problem. This applies also to our problems described in questions I-3 and I-4. It is hence not surprising that existing synthesis techniques are usually subject to a trade-off between expressiveness and efficiency: they either can synthesize only a very restrictive class of models, or have to invoke inefficient procedures, e.g., quantifier elimination, as a back-end solver. My aim is to substantially enlarge the class of models —here: quantitative loop invariants, almost-sure termination proofs, and probabilistic programs— for which the synthesis problems still admit efficient algorithms. I will explore various techniques established in the non-probabilistic setting, e.g., counterexample-guided inductive synthesis, syntax/semantics-guided synthesis, constraint solving, and learning-based synthesis, and determine the limit I can reach in the probabilistic setting. To this end, our recent work [WCX+21, WCX+22] on synthesizing inductive invariants for hybrid systems could be a good start, as the synthesis of both (k-)inductive loop invariants and witnesses of almost-sure termination (question I-3) can be encoded as a similar (bilinear) optimization problem. LR

Expected Results

1. A novel forward denotational semantics using GFs for probabilistic programs with continuous distributions.
2. A decision procedure for checking equivalence of probabilistic programs based on the GF semantics.
3. A quantitative extension of Craig interpolation with an application in combing forward- and backward-reasoning.
4. Powerful techniques for synthesizing loop invariants, (almost-sure) termination proofs, and probabilistic programs.

Track II Verification and Identification of Hybrid Systems

Key Research Questions

- II-1. Are there vector fields beyond solvability whose reachability problem is still decidable?
- II-2. What would be a proper continuous analogy to k-inductive invariants for hybrid systems? How can we synthesize it?
- II-3. How to identify (possibly nonlinear) hybrid-system models from finite observations or active queries?

Promising Approaches

Relating Decidability of the Reachability Problem to Algebraic Integrability. Recall that our solvable dynamics [GCL+18, FCK19] identified a fragment of continuous differential dynamics which has been generally acknowledged as hitherto the most expressive one whose reachability problem is proven decidable. But can we go further, e.g., to show a decidability result for vector fields beyond solvability? An interesting observation is that the class of the so-called integrable systems —dynamical systems with sufficiently many conserved quantities, or first integrals— recognizes a larger family of vector fields than solvability (at least for the linear case). Moreover, one may hopefully construct the system orbits out of a basis of first integrals and thereby obtain HR decidability of the underlying reachability problem. I, therefore, plan to first relate our decidability results for solvable dynamics to the notion of integrability and then try to identify an extended class of systems whose reachability problem is still decidable. The key challenge is to construct a basis of independent, algebraic first integrals which could be highly non-trivial for general integrable systems, but we will start with the easy case concerning linear vector fields.

Discovering (the Continuous Analogy to) k-Inductive Invariants via Finite-Time Stability. k-induction is a well-established verification technique for (discrete) programs, and I am curious to see its application in the verification of hybrid discrete-continuous systems. There have been research efforts along this line, e.g., by either performing classical k-induction over a discretized system dynamics, or working with a purely continuous analogy, known as t-barrier certificates. The latter is close to my expectation, however, the way that t-barrier certificates are formulated is not amenable to automated synthesis: the definition of t-barrier certificates explicitly involves system trajectories, rendering no straightforward method to find an appropriate time bound t and the associated barrier function. I plan to develop an alternative to t-barrier certificates with the help of a promising concept in control theory called finite-time stability, which will hopefully lead to an efficient (semi-)automated synthesis algorithm leveraging, e.g., convex programming techniques. LR

Learning (Possibly Nonlinear) Hybrid Automata through Active Queries. There is recently a mounting interest amongst the verification community in identifying hybrid-system models (often characterized by hybrid automata) from input/output system traces. Most of the proposed techniques are confined to linear models with a passive learning framework, i.e., to learn a linear model that agrees (up to a specified tolerance) with a finite collection of trace data. I am interested in extending these techniques to the identification of nonlinear hybrid automata, where the key challenge is to provide bounds on the errors of the learned model against the given data. Furthermore, I plan to investigate the possibility of applying active learning as per Angluin's L^* framework, where a learner tries to infer a correct model via membership and equivalence queries to a teacher. I foresee many challenging yet interesting problems to be addressed, e.g., (1) how to define an (approximate) equivalence relation for nonlinear hybrid automata such that two models can be claimed ϵ-approximately equivalent (under some confidence of at least δ)? The theory of probably approximately correct learning could be of interest here; (2) how to produce a counterexample (by the teacher) and how the learner can use it to refine a hypothesis? and (3) how to measure and control the size of the identified model? HR

Expected Results

1. A decision procedure for deciding reachability for an extended class of vector fields (beyond solvability).
2. A proper continuous analogy to k-induction for hybrid systems that admits automated synthesis.
3. An active learning method for identifying hybrid-system models with possibly nonlinear dynamics.

Track III (Semi-)Automation, Tool Support, and Applications

Key Research Questions

III-1. To what extent can we automate and implement our verification and synthesis techniques in Track I and Track II?
III-2. Can our techniques and the corresponding implementations be applied to industry-level systems?

Promising Approach

My team will devote a substantial amount of efforts to pushing the limits of automation as far as possible for our techniques in both Track I and Track II. This will include verification techniques such as equivalence checking (I-1), bidirectional reasoning (I-2), and reachability analysis (II-1), as well as synthesis techniques for, e.g., loop invariants and termination proofs (I-3), probabilistic programs (I-4), continuous k-inductive invariants (II-2), and hybrid automata (II-3). The tools will be open-sourced and make use of various prototypes that we have developed over the past years, e.g., KIPRO2 [BCK+21] (k-induction verifier), NIL [CWA+19] (interpolants synthesizer), BMI-DC [WCX+21] (invariant synthesizer), and LINR [GCL+18] (reachability analyser), whereas incorporate substantial improvements with focuses on automaticity, modularity, scalability, and usability. LR

Expected Results

1. Open-source software tools for formally verifying and synthesizing probabilistic programs and hybrid systems.
2. Applications of the software tools in industry-level CPS, e.g., autonomous vehicles and robust control systems.

Conclusion. *My three research tracks are scheduled in parallel, yet will definitely share various insights —e.g., on (k-)invariant synthesis, model identification, and (semi-)automation— and hence benefit from each other. I have been (and will continue to be) working on these projects in extensive collaborations with research groups led by, e.g., J.-P. Katoen at RWTH Aachen, N. Zhan at ISCAS, M. Fränzle at Uni. of Oldenburg, D. Kapur at UNM, S. Sankaranarayanan at CU Boulder, B. Xia at PKU, A. Sogokon at Uni. of Southampton, C. Fan at MIT, B. L. Kaminski at Saarland Uni., S. Junges at Radboud Uni., and C. Matheja at DTU. I foresee also promising collaborations with the faculty at the College of Computer Science and Technology at Zhejiang University, e.g., with Prof. Jianwei Yin on quantum computing and services computing, and with Prof. Yongwang Zhao on program semantics, logic, and verification. Expected outcomes include pivotal breakthroughs in computer-aided verification and synthesis, and promising applications in industry-level systems, e.g., autonomous vehicles, quantum computers, operating systems, robust control, compilers, chips, and AI/ML-driven systems.*

References

⋆ indicates equal contributions

[ACZ+20] J. An, **M. Chen**, B. Zhan, N. Zhan, and M. Zhang. Learning one-clock timed automata. In *Proc. of TACAS '20*, pages 444–462, 2020. [Arfifact Evaluated · Best Paper Award at FMAC 2019].

[BCJ+22] K. Batz, **M. Chen**✉, S. Junges, B. L. Kaminski, J.-P. Katoen, and C. Matheja. Probabilistic program verification via inductive synthesis of inductive invariants. 2022. Under review.

[BCK+21] K. Batz, **M. Chen**✉, B. L. Kaminski, J.-P. Katoen, C. Matheja, and P. Schröer. Latticed k-induction with an application to probabilistic programs. In *Proc. of CAV '21*, pages 524–549, 2021. [Arfifact Evaluated].

[CFL+16] **M. Chen**, M. Fränzle, Y. Li, P. N. Mosaad, and N. Zhan. Validated simulation-based verification of delayed differential dynamics. In *Proc. of FM '16*, pages 137–154, 2016.

[CFL+18] **M. Chen**✉, M. Fränzle, Y. Li, P. N. Mosaad, and N. Zhan. What's to come is still unsure - Synthesizing controllers resilient to delayed interaction. In *Proc. of ATVA '18*, pages 56–74, 2018. [Distinguished Paper Award].

[CFL+21] **M. Chen**✉, M. Fränzle, Y. Li, P. N. Mosaad, and N. Zhan. Indecision and delays are the parents of failure - Taming them algorithmically by synthesizing delay-resilient control. *Acta Informatica*, 58(5):497–528, 2021.

[Che19] **M. Chen**✉. Verification and synthesis of time-delayed dynamical systems. *PhD Dissertation*, Institute of Software, Chinese Academy of Sciences, China, 2019.

[CHT+17] **M. Chen**✉, X. Han, T. Tang, S. Wang, M. Yang, N. Zhan, H. Zhao, and L. Zou. MARS: A toolchain for modelling, analysis and verification of hybrid systems. In *Provably Correct Systems*, pages 39–58. Springer, 2017.

[CKKW22] **M. Chen**✉, J.-P. Katoen, L. Klinkenberg, and T. Winkler. Does a program yield the right distribution? Verifying probabilistic programs via generating functions. 2022. Under review.

[CRW+16] **M. Chen**, A. P. Ravn, S. Wang, M. Yang, and N. Zhan. A two-way path between formal and informal design of embedded systems. In *Proc. of UTP '16*, pages 65–92, 2016.

[CWA+19] **M. Chen**✉, J. Wang, J. An, B. Zhan, D. Kapur, and N. Zhan. NIL: Learning nonlinear interpolants. In *Proc. of CADE '19*, pages 178–196, 2019.

[FCK19] M. Fränzle, **M. Chen**, and P. Kröger. In memory of Oded Maler: Automatic reachability analysis of hybrid-state automata. *ACM SIGLOG News*, 6(1):19–39, 2019.

[FCX+20] S. Feng⋆, **M. Chen**✉⋆, B. Xue, S. Sankaranarayanan, and N. Zhan. Unbounded-time safety verification of stochastic differential dynamics. In *Proc. of CAV '20*, pages 327–348, 2020. [Arfifact Evaluated].

[FCZ+19] S. Feng⋆, **M. Chen**✉⋆, N. Zhan, M. Fränzle, and B. Xue. Taming delays in dynamical systems - Unbounded verification of delay differential equations. In *Proc. of CAV '19*, pages 650–669, 2019. [Arfifact Evaluated].

[GCD+15] T. Gan, **M. Chen**, L. Dai, B. Xia, and N. Zhan. Decidability of the reachability for a family of linear vector fields. In *Proc. of ATVA '15*, pages 482–499, 2015.

[GCL+16] T. Gan, **M. Chen**, Y. Li, B. Xia, and N. Zhan. Computing reachable sets of linear vector fields revisited. In *Proc. of ECC '16*, pages 419–426, 2016.

[GCL+18] T. Gan, **M. Chen**, Y. Li, B. Xia, and N. Zhan. Reachability analysis for solvable dynamical systems. *IEEE Trans. Automat. Contr.*, 63(7):2003–2018, 2018.

[GDX+16] T. Gan, L. Dai, B. Xia, N. Zhan, D. Kapur, and **M. Chen**. Interpolant synthesis for quadratic polynomial inequalities and combination with EUF. In *Proc. of IJCAR '16*, pages 195–212, 2016.

[LZC+22] Y. Li, N. Zhan, **M. Chen**✉, H. Lu, G. Wu, and J.-P. Katoen. On termination of polynomial programs with equality conditions. *Journal of Computer and System Sciences*, 2022. Under review.

[WAC+20] J. Wang, J. An, **M. Chen**, N. Zhan, L. Wang, M. Zhang, and T. Gan. From model to implementation: A network-algorithm programming language. *SCIENCE CHINA Information Sciences*, 63(7), 2020.

[WCX+21] Q. Wang⋆, **M. Chen**✉⋆, B. Xue, N. Zhan, and J.-P. Katoen. Synthesizing invariant barrier certificates via difference-of-convex programming. In *Proc. of CAV '21*, pages 443–466, 2021. [Arfifact Evaluated].

[WCX+22] Q. Wang⋆, **M. Chen**✉⋆, B. Xue, N. Zhan, and J.-P. Katoen. Invariant barrier certificates. *Information and Computation*, 2022. Under review (invited contribution).

[XMF+17] B. Xue, P. N. Mosaad, M. Fränzle, **M. Chen**, Y. Li, and N. Zhan. Safe over- and under-approximation of reachable sets for delay differential equations. In *Proc. of FORMATS '17*, pages 281–299, 2017.