Aerospace, robotics, health care, transportation, and military are just a few areas in which Cyber-Physical Systems (CPS) are ubiquitous. They are becoming an indispensable part of our lives, and companies no longer have the luxury of time to spend a year testing them. For instance, right now, in the era of “groundbreaking first,” companies endeavor to create a drone delivery system. In May 2019, the University of Maryland Medical Center in Baltimore delivered a kidney to a patient using an unmanned drone. And, in June 2019, CEO of Amazon Worldwide Consumer announced that they plan to deliver packages via drones within months. On the other hand, failures in these systems are often catastrophic financially and in terms of human life. In October 2018 and March 2019, two Boeing 737 MAX crashed and killed all 346 people on board. In both accidents, automated commands issued by a new system pushed the airplane nose down in response to erroneous sensor data. In the first quarter of 2018, more than 208 million medical devices were recalled, which is the highest number since 2005. Experts believe that software problems account for 23% of them. In March 2018, a software bug in one of Uber’s self-driving cars caused the death of a pedestrian. It is more critically important than ever to be able to verify these systems automatically.

Considering tragedies caused by failures in these systems and knowing that testing can never, even in theory, cover all the uncountably many possible behaviors, there should be no doubt that formal verification must be a crucial part of the production. Unfortunately, verification of all but very restricted classes of cyber-physical systems is known to be undecidable. Even an innocent-looking problem, like whether a particular behavior (e.g., server responds to requests) always happens in the future, is known to be at a much higher level of undecidability. Therefore, it is very natural that researchers have explored different approaches to partially solve the CPS verification problem.

My research spans three themes. The first two, automatic verification and statistical verification, are about advancing state-of-the-art techniques specifically designed to overcome challenges in solving practical problems. While solving these problems generally hinges on building powerful tools, fundamental advances often rely on our better understanding of the theoretical foundation, which is precisely the third theme in my research.

1. **Automatic Verification.** Industrial cyber-physical systems often exhibit non-polynomial dynamics. To establish unbounded safety and stability in these systems, we introduced the first-ever inductive proof rules that are amenable to automatic verification [2]. We also used automated abstraction refinement techniques to formally verify unbounded safety in cyber-physical systems with non-linear dynamics, polynomial or otherwise [4,8,9]. Our implementation outperforms state-of-the-art tools on quite a few examples and is freely available at [https://nima-roohi.github.io/HARE/](https://nima-roohi.github.io/HARE/). To the best of our knowledge, the progress guarantee that is offered by our algorithm applies to one of the largest classes of cyber-physical systems that a tool ever supported for formal verification [11].

2. **Statistical Verification.** When a system exhibits probabilistic behavior, like when there is a noise involved, we cannot always apply our previous techniques. Furthermore, in practice, most of these systems are too large for symbolic model checkers. This is where statistical methods shine, where the verification is performed by looking at random behaviors. Using them, we became the first who verified a real-world problem proposed by engineers at Toyota motor corporation [15]. Also, by being smarter about how to choose the random executions, we have been able to verify properties in systems with more than $1.35 \times 10^{14}$ states, two orders of magnitude faster than a state-of-the-art tool [16]. Our tool is freely available at [https://nima-roohi.github.io/STMC/](https://nima-roohi.github.io/STMC/). Finally, we developed algorithms to verify systems with more than $4.39 \times 10^{52}$ abstract states in just a few minutes [18].

3. **Theoretical Foundation.** Besides our effort to solve practical problems, we have extended theoretical aspects of model checking cyber-physical systems in a few areas. We discovered a 20+ years old bug in a well-studied topic like metric temporal logic and fixed it [14]. We developed a new monitoring algorithm and proved its performance does not depend on unknown parameters, which is particularly useful in medical devices since there are many unknowns in the human body [6]. We developed new algorithms for robust model checking of timed automata with direct application in the implementability of distributed algorithms [10]. We introduced a much more general notion of stability in switching systems and showed how to reduce it to simpler ones [17]. We critically evaluated the performance of different algorithms for probabilistic model checking of unbounded temporal formulas [13]. We introduced a new class of systems for which the unbounded safety problem is decidable [3]. Finally, we proved the decidability of bounded safety model checking for an extension of what is known as a boundary of decidability [12].
Research Accomplishments

Automatic Verification of Non-Probabilistic Cyber-Physical Systems. In software and hardware verification domains, Satisfiability Modulo Theories (SMT) solvers have been successfully used in practice. Therefore, it should be of no surprise that SMT solvers are widely used in formal verification of CPS as well. However, due to high computational complexity, practical applications in this area are much more limited.

In [11], we showed that by merely using non-strict inequalities, continuous functions (including exponential and trigonometric), and bounded quantifiers, all CPS bounded safety problems written using the first-order logic are numerically robust. This is significantly more general than what is supported by almost any tool. We even showed that as long as the amount of perturbation is small enough, hybrid automaton and its perturbation simulate the behavior of each other for an arbitrarily adjustable finite number of steps. Therefore, bounded safety can be automatically proven using bounded verification tools that are δ-complete. However, as it is well-known, the computational complexity will be the bottleneck in practice. To tackle the resource problem and to handle other crucially important non-robust properties (like stability), in [2], we introduced three numerically robust inductive proof rules for unbounded safety and Lyapunov stability. For example, in the case of stability, the general idea is to require it only outside of an arbitrarily small neighborhood around the origin (at most $10^{-10}$ in our examples). A simple change like this allowed us to define what is now called ε-Lyapunov function and prove that ε-stability in the sense of ε-Lyapunov function is always robust. Therefore, current decision procedures can be used to establish those properties automatically. As another example, using our inductive proof rules, we were able to prove an unbounded safety property for a powertrain control system with non-polynomial dynamics suggested by engineers at Toyota motor corporation.

A complementary approach to SMT-based model checking is the abstraction, which, due to high complexity, is often required in practical applications. The main problem of the abstraction is finding a model with simpler dynamics that is amenable to automated analysis and satisfies the property of interest. In [8, 9], we developed a new Counter-Example Guided Abstraction Refinement (CEGAR) algorithm for hybrid automata with non-linear dynamics. The CEGAR framework tries to automatically discover the right abstraction through a process of progressive refinement based on analyzing spurious counter-examples in the abstract models. For instance, in the case of an example called Jet Engine, our tool, HARE, automatically created 189 locations and connected them using 1330 edges to establish unbounded safety. We compared HARE with six other state-of-the-art tools on 85 examples from literature. Our results show that HARE proves the unbounded safety of more models almost always in a shorter amount of time. Using our later results in [11], we realized that not only our algorithm is relatively complete, but also, to the best of our knowledge, its progress guarantee applies to one of the largest classes of cyber-physical systems that a tool ever supported for formal verification. Our tool is open source and available at https://nima-roohi.github.io/HARE/.

Statistical Verification of Probabilistic Cyber Physical Systems. Scalability has been the number one objection against formal verification since the beginning, and for probabilistic systems, statistical model checking is the number one solution to the scalability issue of formal verification. Furthermore, in CPS systems, due to the mixture of continuous and discrete dynamics, probability distributions most often cannot be even expressed in closed forms. Therefore, for all but the most trivial systems, automatic symbolic model checking is a nonstarter. In [18], we used abstraction to represent non-linear stochastic dynamics of discrete, continuous, and hybrid systems using discrete and continuous-time Markov chains. These abstractions made it possible to encode systems with infinitely many states using only finite memory. Finally, statistical techniques made it possible for us to successfully model check those abstract systems with more than $4.39 \times 10^{52}$ states in just a few minutes.

The most fundamental requirement for statistical verification is the ability to take samples. In [16], we used smarter sampling techniques to reduce the number of samples required to verify temporal formulas. The most popular tool in this area is PRISM. We compared our algorithms with a PRISM’s statistical algorithm and four different symbolic ones (not all of them survived the state space explosion). The result shows that our methods use significantly fewer samples
and execution time than PRISM. For example, in the case of the EGL Contract Signing Protocol with more than $1.35 \times 10^{14}$ states, our algorithm runs about two orders of magnitude faster than PRISM's statistical algorithm by taking two orders of magnitude fewer samples. And in the case of the Crowds protocol with more than $7.5 \times 10^{13}$ states, after we conservatively set error bounds to $10^{-4}$ and indifference region to $8 \times 10^{-4}$, our algorithm runs 13 times faster than one of the symbolic algorithms, while all the other ones either ran out of memory or did not even start, complaining the model has too many states. Our tool is open source and available at https://nima-roohi.github.io/STMC/.

Last but not least, in 2014, engineers at Toyota motor corporation published a real-world problem on verifying a power-train (the mechanism that transmits the drive from the engine to its axle). Their problem includes realistic automotive design features like delayed differential and difference equations, lookup tables, and highly non-linear dynamics. In [15], we statistically verified the most complicated version of these problems, by simulating the C++ code generated from the Simulink\textsuperscript{TM} model of the design. Our statistical results show that with the error probability of less than 0.001%, at least 98% of the possible initial operating conditions result in safe executions. It took our tool just about two minutes to terminate. This is the first formal verification result for this problem, statistical or otherwise. Note that the inductive proof rules that we developed for verifying the safety and stability of non-linear systems do not handle delayed differential equations nor lookup tables. That is why, for those rules, we used a simplified version that is proposed by the same engineers at Toyota.

**Theoretical Foundation.** Parallel to our efforts in solving practical problems, we continue to extend theoretical results in the verification of cyber-physical systems. In [14], we show, despite being an active area of research for more than 20 years, automata-based model checking algorithms for Metric Interval Temporal Logic (MITL), a prominent specification formalism for real-time systems, are flawed. In addition to fixing this problem, we also found a new subclass of these formulas for which model checking is exponentially faster. In [6], we show how to extend any hypothesis testing algorithm invariant to unknown parameters to a parameter invariant monitoring algorithm for Signal Temporal Logic (STL), which is an extension of MITL with real-valued constraints. Our algorithm also has provable error bound guarantees and has direct application in medical devices, as the human body has many parameters that cannot be measured. In [10], we consider robust model checking of timed automata, with direct application in distributed algorithms and their implementation. This is where one considers the fact that, in practice, sensors are noisy, and clocks are imperfect. We show (in)equivalency between seven alternations to the problem of robust model checking ω-regular properties, and prove that they are all PSPACE-complete. In [17], we consider a much more general notion of stability for switching systems. We introduce an efficient procedure to reduce verification of stability under our general notion into simpler ones. In [13], we critically evaluate the performance of several algorithms for model checking probabilistic unbounded until formulas and argue that some of these algorithms are indeed incorrect. This is perhaps the most challenging type of formulas in probabilistic temporal logic. In [3], we consider subtleties caused by high dimensionality in cyber-physical systems (i.e., large number of continuous variables in the model) and introduce a new class of hybrid automata for which safety problem is decidable, with no restriction on the dimension. Finally, in [12], we prove that bounded time safety problem is decidable for a class of hybrid automata that strictly subsumes another class known as a boundary for decidability of safety problems.

**Future Directions**

Like many other systems, at a very high level, building a cyber-physical system involves three phases: modeling, verification, and implementation. However, there are a few challenges that often are specific to cyber-physical systems: 1. The most famous tool currently used in the industry for modeling a cyber-physical system is Simulink\textsuperscript{TM}, which does not even have formal semantics. 2. While we know most of the current formal models cannot represent actual systems precisely [7], even safety verification is undecidable for all but very restricted classes of these models. 3. When we can positively verify a CPS model, the controller part of this model is rarely implementable. These problems make building a cyber-physical system like creating art; there is no established methodology that addresses all those issues. Let us not forget that the only future one can expect is the one in which cyber-physical systems are even more dominantly exist in our lives. My ultimate research goal is to transform building industrial cyber-physical systems from art into an engineering activity. I believe learning-based search methods, along with robust and statistical model checking, are three promising approaches toward solving the challenges mentioned above and making the formal verification of cyber-physical systems more favorable/feasible in the industry.
Learning-Based Search Methods. Proving safety, establishing stability, and synthesizing policies, are among the most fundamental problems in formal verification of cyber-physical systems. Despite the apparent differences in how these problems are defined, solved, or applied in practice, they can all be looked at as search problems: searching for a counter-example, Lyapunov functions, or different policies that satisfy some constraints. Furthermore, knowing these are challenging problems, it is of no surprise that they are all \( \text{NP} \)-hard. Therefore, unless complexity classes \( \text{P} \) and \( \text{NP} \) are equal, in theory, one can never solve them efficiently. Fortunately, this has not stopped the research community and the search for better algorithms for more specific, restricted, and yet still practical applications continues.

Recent advances in neural networks have been beneficial in finding better search algorithms. I would like to leverage AI in finding more effective search strategies. For example, in [1], we use neural network Lyapunov functions to learn control policies for non-linear control problems. Our experiments suggest that this could be used to obtain provably correct solutions for challenging robot control problems such as humanoid robot balancing and wheeled vehicle path-following. We plan to take this approach one step further in [5]. Not only we use neural networks to represent the controller, but we also use conventional learning algorithms like proximal policy optimization for training. In this approach, we use the full power of machine learning algorithms to train a network and only stop after we formally verified a candidate solution. Similarly, in [19], we plan to use reinforcement learning to verify Markov decision processes with unknown probability distributions against temporal specifications. Our hierarchal algorithm automatically learns how to optimize the search such that only those states that are believed could be useful, will be explored (the belief is formed through higher levels of the hierarchy).

Robust Model Checking. A promising approach to deal with the current difficulties in the verification of cyber-physical systems is robust model checking. This is where we address issues like uncertainty in design parameters, the validity of synthesis after small changes in settings, a little noise in the physical environment, impreciseness of sensors, distributed controllers, implementability of verified designs, and complexity of model checking.

Considering our experience in [4,8–10,12] and similar results by other authors, I would like to explore robustness in more detail. Even though robust control is an active field of research in control theory, surprisingly, not much has been done in the robust verification of cyber-physical systems. I would like to understand the underlying properties of cyber-physical systems regarding different notions of robustness. An incomplete understanding of these concepts can lead to a significant waste of resources in model checking. For example, according to the most common semantics that is used for STL, a trivially valid formula like \( \square (p \lor \neg p) \) is neither false nor true on almost all continuous signals.

I would also like to investigate interval arithmetic, which is another venue with immediate application in SMT solvers. It is because of robustness that interval arithmetic can be used to solve non-linear constraints. Although a lot has been done in this field, it certainly deserves more attention. For example, based on my experience, even a famous interval arithmetic library like Gaol is still far from being perfect; \( \cos(x) \) is not necessarily inside \([-1,1]\), and other functions can return smaller intervals without any compromise in their performance. Another famous library that uses interval arithmetic is IBEX, which does not allow shared terms in inequality constraints. This simple restriction causes the encoding of a simple constraint on extended Kalman filter with less-than-a-page definition to produce a one-million-characters-long formula with no shared term. And finally, the most critical example is the ability to use GPUs in SMT solvers. The most used sub-algorithm in solving non-linear constraints is called forward/backward propagation, which applies a fixed sequence of non-linear functions to a very long sequence of intervals. Any success in moving interval arithmetic to GPUs would have an immediate application in model checking cyber-physical systems.

Statistical Model Checking. I would like to explore statistical techniques in more detail. We already released some of our algorithms to the public [16]. However, there are many exciting directions that one can follow. For instance, I would like to design better sampling methods when there are many steps in temporal specifications. Although our approach in [16] is straightforward, we don’t know if that is the best approach. We may even change the representation of the probability transition matrix to have a better sampling method. As another example, I would like to work on the probabilistic error guarantees in the statistical verification of temporal formulas. When truth values of atomic propositions at different steps are a priori unknown, one often divide the desired error over different steps. But this is not always required. For instance, to prove \( \square [1,100] p \), one does not need to split the error probability at all. When the simulation is expensive (like in [15]), this could significantly improve performance.
The correctness of many of our statistical algorithms relies on some notion of robustness. For example, suppose an execution of a system is given by a sequence of distributions. This is particularly useful in the field of Robot Perception, where a robot only has a probabilistic belief about its surrendering, and it could change over time. In [18], we require that a small change in this perception does not change whether or not it satisfies a specific temporal specification. This simple yet still realistic requirement made it possible for us to use statistical methods for the verification. So robot has a probabilistic dynamic, our problem is defined in a deterministic sense (there is only one signal), and we used statistical techniques for the verification. I would like to understand the connection between robustness and statistical verification better. Was our experience just a lucky accident, or under some notion of robustness, we can transform a robust deterministic problem into a stochastic one and solve it more efficiently?

References