Distribution-Free Runtime Monitoring of Stochastic Contracts via Conformal Prediction and Scenario Optimization

Joon Kim, Wanyue Lin

Abstract— **Cyber-physical systems (CPS) often operate in stochastic environments where precise modeling is difficult and design-time assumptions may not hold at runtime. While runtime monitoring is essential for safety, verifying probabilistic specifications in Stochastic Signal Temporal Logic (StSTL) contracts typically relies on strong distributional assumptions, such as Gaussian noise. This paper presents a distribution-free framework for runtime monitoring of stochastic contracts, leveraging Conformal Prediction (CP) and Scenario Optimization (SO). We address the challenge of monitoring system constraints under unknown stochasticity by formulating two distinct problems: maintaining safety distance using CP and optimizing performance using SO. Our approach utilizes CP to construct valid prediction regions for unknown noise distributions and adapts SO to derive dynamic lower tolerance limits for runtime fault detection. Experimental results demonstrate that these methods provide rigorous probabilistic guarantees and competitive performance compared to baseline methods that assume known distributions.**

I. Introduction

Cyber-physical systems (CPS), such as autonomous vehicles, aircraft, and robotic platforms, operate in stochastic and uncertain environments [1]. Noise and disturbances introduce uncertainty that makes precise modeling and prediction challenging. Assumptions made during design or verification may no longer hold at runtime, especially when the system is deployed in real-world settings. Therefore, runtime monitoring of system constraints is essential to ensure safe and reliable operation.

However, verifying or monitoring probabilistic predicates—such as whether a safety requirement holds with high probability—is a challenging problem [2]. Classical runtime verification techniques are often deterministic and assume exact knowledge of the system model or disturbance bounds [3]–[5]. These assumptions become restrictive when the underlying system is stochastic, high-dimensional, or learned from data.

To address this, several frameworks have been proposed for the design and verification of CPS under uncertainty. Among them, **assume–guarantee contracts** have proven particularly powerful for compositional reasoning [6]–[8]. Contracts specify assumptions on a component's environment and guarantees on its behavior, typically expressed as logical formulae over a set of atomic propositions such as Signal Temporal Logic (STL) [9] and Linear Temporal Logic (LTL) [10]. To extend formal reasoning into probabilistic domains, Stochastic Signal Temporal Logic (StSTL) has been proposed [11], [12]. StSTL augments STL with chance constraints to express statements such as "the state satisfies the predicate with probability at least 0.95." This enables formal reasoning about system properties under uncertainty.

Yet, existing StSTL-based verification approaches often rely on strong distributional assumptions, such as Gaussian noise models, and are typically performed offline, using precomputed models or simulation data [11]–[13]. As a result, these methods cannot adapt to changing environments or distribution shifts that naturally occur during system operation. When strong assumptions on the distribution of stochasticity are impossible, we must resort to distribution-free methods. Fortunately, advances in data-driven uncertainty quantification, such as **conformal prediction** and **scenario optimization**, offer new opportunities for runtime verification.

Conformal prediction(CP) provides distribution-free probabilistic guarantees on future predictions by calibrating them with observed data [14]–[16]. Unlike traditional StSTL-based methods that assume a fixed noise model, conformal prediction can adapt its prediction regions dynamically and remains valid under unknown or shifting data distributions [2], [17], [18]. This property makes it particularly attractive for monitoring stochastic systems operating in nonstationary or partially modeled environments.

Orthogonal but not unrelated to conformal prediction, scenario optimization(SO) is a powerful framework that quantifies the risk and confidence of chance constraints in a distribution-free, data-driven manner [19]. Many problems in StSTL design-time verification can be cast as a scenario optimization problem. While scenario optimization is not developed in the context of runtime verification, we propose an adaptation that gives opportunities for further exploration.

Other approaches, such as hypothesis testing, moment generating functions, and dynamic programming methods, were also considered throughout the investigation. Notably, these approaches are not distribution-free but rather attempt to explicitly infer the underlying distribution of stochasticity. However, they are not discussed further in this paper, as they lack statistical power and rigor compared to distribution-free approaches.

In this paper, we aim to develop a framework for monitoring stochastic contracts at runtime that combines the structure of assume–guarantee contracts with the robustness of data-driven uncertainty quantification. Specifically, our goal is to enable high-confidence, distribution-free runtime verification of stochastic system properties.

Our main contributions are summarized as follows:

J.Kim and W.Lin are with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, CA, USA. Email: {joonkim1, jennylinwanyue}@berkeley.edu

- We introduce CP and SO and motivate how they are relevant in the context of StSTL runtime verification. Two concrete StSTL problems are formulated for experimentation.
- We develop two CP-based runtime monitoring algorithms that detect contract violations in an offline and online manner, respectively.
- We compare SO to a baseline StSTL verification algorithm(PYCASSE [12]) that has strong distribution assumptions to show its competitiveness, and develop a SO-based runtime monitoring algorithm.

II. Related Work

A. Runtime Verification and Conformal Prediction

The goal of [20] is to check real-time properties by incorporating empirical stochastic analysis into contracts. The method sets a probabilistic bound on each component's execution time to ensure it meets timing requirements with probability π, defined as

$$\hat{\tau}_{\pi} = \mu_n + \gamma s_n,$$

where μ_n and s_n are the mean and standard deviation of execution times. At *design time*, γ is computed from the empirical cumulative distribution function (CDF). At *runtime*, μ_n and s_n are continuously updated from new data using a sliding window, while γ remains fixed.

The work extends Design by Contract to real-time properties, unifying functional and timing contracts within a single framework. However, the approach mainly focuses on per-component timing constraints without addressing cross-component temporal relations or compositional contracts. Moreover, when the execution time distribution drifts significantly, a fixed threshold may become less robust. Although the paper provides empirical probabilistic justification, it lacks stronger formal guarantees.

Furthermore, [21] proposes quantitative predictive monitoring (QPM), the first predictive monitoring method to support stochastic processes and the quantitative semantics of arbitrary STL specifications, avoiding expensive Monte Carlo simulations at runtime. During operation, the monitor observes the current system state and uses a machine learning model to predict future behavior. It then applies *Conformal Quantile Regression* to generate a confidence interval for the future satisfaction of the STL specification. If the interval lies entirely above zero (robustness > 0), the system is likely safe; if it crosses or falls below zero, the monitor raises a warning for a potential violation. However, it is task-specific and assumes Markovian dynamics, and its performance may degrade under temporal correlations or distribution shifts.

Similarly, [2] introduces a conformal-prediction–based framework for runtime verification with statistical confidence guarantees. The authors propose (i) a direct method that constructs prediction regions for the robust satisfaction value of a temporal logic specification, and (ii) an indirect method that first builds prediction regions for future system states and then derives a worst-case robustness bound from them. Its strength is that it provides the first formal, distribution-free guarantees for predictive runtime verification using common trajectory predictors like RNNs and LSTMs, while remaining computationally simple. However, it assumes that calibration, test, and runtime data are i.i.d. from the same distribution, which is rarely true in real systems with distribution shift, and its coverage guarantee is only marginal rather than conditional, so reliability can drop significantly in specific high-risk scenarios.

B. Scenario Optimization

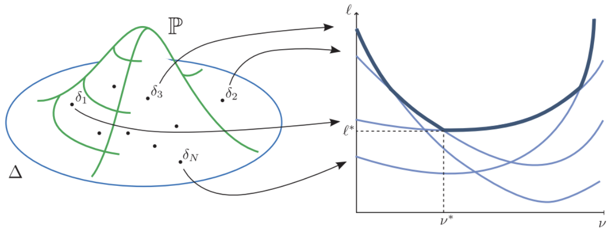

Fig. 1: Pictorial representation of SO [22].

This section establishes the theoretical foundations of Scenario Optimization [19], [22]. Consider an optimization problem governed by a decision parameter vector ν and a random variable δ representing uncertainty. Let Δ denote the probability space of all possible outcomes equipped with a probability measure $\mathbb{P}$. Each realization of δ is termed a "scenario." The objective is to minimize a convex objective function $\ell(\nu, \delta)$ subject to constraints imposed by randomly sampled δ_i. Since the underlying distribution of δ is assumed unknown, an exact solution is intractable. SO circumvents this by adopting a data-driven strategy: sampling N independent and identically distributed (i.i.d.) scenarios $\delta_1, \ldots, \delta_N$ from Δ and solving for the worst-case instance among them. Formally, the scenario program is defined as $\min_\nu \max_{i \in [N]} \ell(\nu, \delta_i)$ where ℓ represents the convex loss function (over ν if δ_i is fixed). This min-max problem is also convex, and SO solves for a tuple (ν^*, ℓ^*) such that $\ell(\nu^*, \delta_i) \leq \ell^* \; \forall \delta_i$.

The efficacy of the computed solution (ν^*, ℓ^*) is evaluated across three interdependent dimensions:

- **Performance** (ℓ^*): The optimality of (ν^*, ℓ^*) relative to the true optimal parameters.
- **Risk** (ϵ): The probability that the computed parameters ν^* will violate a newly sampled constraint. Formally, this is the measure of the set of bad scenarios: $\mathbb{P}[\delta \in \Delta : \ell(\nu^*, \delta) > \ell^*]$.
- **Confidence** $(1 - \beta)$: The probability that the risk bound holds over the random sampling of the N scenarios.

The confidence parameter is necessary due to only observing a finite number of samples; there always exists a nonzero probability that the observed samples do not properly reflect the underlying distribution Δ. In that case, SO guarantees nothing. In practice, β is fixed to be a very small constant($\leq 10^{-5}$), and such edge cases in sample collection is ignored.

There exist inherent trade-offs among these dimensions; specifically, achieving lower risk often necessitates a larger sample size N or compromised performance. The fundamental theorem of Scenario Optimization quantifies the relationship between the sample size N, risk ϵ, and confidence parameter β.

Theorem 1: For any $\epsilon \in (0, 1)$ (risk parameter) and $\beta \in (0, 1)$ (confidence parameter), if the number of scenarios N satisfies $N \geq \frac{2}{\epsilon}(\ln(\frac{1}{\beta})+d-1)$, then, with probability $\geq 1-\beta$, it holds that l^* is ϵ-risk guaranteed, that is,

$$\mathbb{P}[\delta \in \Delta : l(\nu^*, \delta) > l^*] \leq \epsilon. \tag{1}$$

This theorem provides a rigorous probabilistic certificate, making SO particularly relevant for verifying stochastic contracts where distribution-free guarantees are required.

While Theorem 1 provides rigorous bounds on Risk and Confidence, it does not address the conservatism of the found solution with respect to Performance. In robust optimization, satisfying all sampled constraints, which might include outliers, often leads to solutions with poor performance cost. To mitigate this, the Sample-and-Discard (SD) framework [23] introduces a mechanism that trades risk for performance.

The procedure involves sampling N scenarios and discarding a subset of k samples according to a specified removal rule (e.g., greedy removal of the most restrictive constraints). Optimization is then performed on the remaining $N-k$ samples. Because the constraint set is relaxed, the performance of the post-discarding solution, ν_k^*, improves (or remains constant) relative to the full scenario solution (Fig. 2). This relaxation comes at the cost of increased risk, which is bounded by the following theorem:

Theorem 2: With probability $\geq 1-\beta$, the solution (ν_k^*, ℓ_k^*) derived after discarding k samples is ϵ_k-risk guaranteed, that is, $\mathbb{P}[\delta \in \Delta : \ell(\nu_k^*, \delta) > \ell_k^*] \leq \epsilon_k$, where:

$$\epsilon_k = \frac{k}{N} + [\frac{\sqrt{k}}{N} + \frac{\sqrt{k}+1}{N}((d-1)\ln(k+d-1) + \frac{d-1}{\sqrt{k}} + \ln\frac{1}{\beta})]. \tag{2}$$

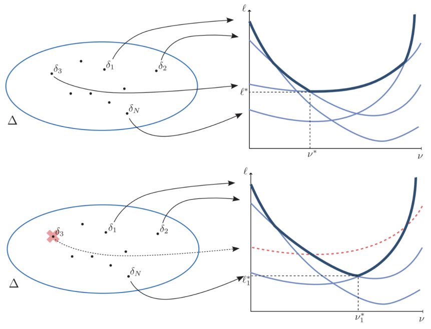

Fig. 2: The effect of discarding [22].

An alternative perspective on the relationship between discarding and risk is provided by the binomial tail bound.

Theorem 3: Let $\beta \in (0, 1)$ be any small confidence parameter value. If N and k satisfy:

$$\binom{k+d-1}{k} \sum_{i=0}^{k+d-1} \binom{N}{i} \epsilon^i (1-\epsilon)^{N-i} \leq \beta \tag{3}$$

then the violation probability of the solution is bounded by ϵ with confidence $1-\beta$.

Theorem 3 is of independent interest as it explicitly models the probability of observing exactly i violations. This formulation allows for the construction of a Performance-Risk trade-off curve, enabling system designers to select an optimal operating point by varying k. The validity of the curve is maintained via a union bound argument on confidence, which is negligible for sufficiently small β (e.g., 10^{-9}).

III. Problem Formulation

We consider two different StSTL problems for CP and SO. Both are related to vehicle dynamics, but CP solves a $\mathbb{G}$ (Always) contract related to safety and comfort, while SO solves a $\mathbb{F}$ (Eventually) contract related to performance. It remains as future work to validate whether the methods would work without loss of generality when the contracts are swapped.

A. Conformal Prediction

For CP, we study the safety distance maintenance problem in an Adaptive Cruise Control (ACC) system. Our goal is to verify whether the control–contract parameters optimized by PYCASSE [12] ensure that the inter-vehicle distance remains above a required safety threshold under both process and measurement noise, and to evaluate the satisfaction rate through repeated stochastic simulations.

The vehicle dynamics are modeled as a discrete-time linear system with sampling period $\Delta t = 0.5$ s. The state vector is $x = [x_e,\, v_e,\, x_l,\, v_l]^\top$, where (x_e, v_e) represent the ego vehicle's position and velocity, and (x_l, v_l) represent the lead vehicle's position and velocity. The dynamics follow

$$x_{k+1} = Ax_k + Bu_k + w_k, A = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \Delta t \\ 0 \\ 0 \end{bmatrix},$$

The measurement model is $z_k = Cx_k + v_k$ and

$$C = \begin{bmatrix} -1 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix},$$

where the first row extracts the distance $d = x_l - x_e$, second row the relative velocity, and third row the ego velocity.

The MILP synthesis performed by PYCASSE yields the optimal safety contract parameters $p = 0.99375$ and $c_s = 5.625$. The safety constraint requires the inter-vehicle distance to satisfy $d_k = x_l(k) - x_e(k) \geq c_s = 5.625$ m for all time steps. To enforce the contract, we adopt the synthesized control law $u_k = Dz_k + E$ with $D = [K,\; K,\; -\tau K]$ and $E = -Kd_{\text{safe}}$, using the simulation parameters $K = 0.5$, $\tau = 1.6$, and $d_{\text{safe}} = 10$ m.

Closely following the model in [12], we introduce process noise to reflect uncertainty in real driving conditions, modeled as $w_k \sim \mathcal{N}(0, Q)$ with $Q = \mathrm{diag}(0, 0, 0, (\sigma_\alpha \Delta t)^2)$ and $\sigma_\alpha = 0.5$. The measurement noise is modeled as $v_k \sim \mathcal{N}(0, R)$ with $R = \mathrm{diag}(1^2, 1^2, 0.5^2)$. However, in our setting, we make a more stochastic assumption by not presuming knowledge of the exact noise distributions; instead, we treat the disturbances as unknown but bounded stochastic variations that are sampled during simulation.

To evaluate the probability that the synthesized control law satisfies the safety distance requirement under stochastic disturbances, we perform a Monte-Carlo simulation study. Each simulation run starts from the initial condition $x_0 = [0,\, 0,\, 50,\, 0]^\top$ and evolves for a total duration of $T = 10$ s with sampling period $\Delta t = 0.5$, yielding $H = 20$ discrete time steps. For each trajectory, we compute the inter-vehicle distance $d_k = x_l(k) - x_e(k)$ and check whether the safety constraint $d_k \geq c_s$ holds for all k. A trajectory is classified as violating the contract if any time step satisfies $d_k < c_s$. We repeat this procedure over many independent stochastic realizations, count the number of satisfying and violating trajectories, and estimate the empirical safety probability.

B. Scenario Optimization

We again draw an example system from PYCASSE for comparison with a solution that assumes a known distribution. Consider a discretized vehicle dynamics model where the position x is updated based on constant acceleration $a = 1$ and a velocity term subject to additive Gaussian noise. The system is initialized at $x = 0, v = 0$. The noise is sampled from a 0-mean normal distribution $\mathcal{N}(0, 0.5^2)$ at each timestep. After 10 unit timesteps, the final position is a random variable dependent on the noise at each timestep.

The verification objective is to identify parameters (p, c) satisfying the stochastic contract:

$$\mathbb{F}_{[0:9]}\mathbb{P}[x \geq c] \geq p.$$

This requires that at least p percent of all trajectories reach a distance c within 10 timesteps. Higher values of c represent better system performance; thus, the loss function is to minimize $-c$ (or equivalently, to maximize c).

IV. Conformal Prediction Results

A. Design time Verification

1) Noise Approximation Without Distributional Assumptions: Since we do not assume prior knowledge of the true noise distribution, the simplest way to handle unknown disturbances is to approximate their distribution using samples. We adopt two methods. The first is a plug-in Gaussian approximation: we generate raw "unknown-distribution" noise samples, estimate their empirical mean and standard deviation, and then use these estimates as parameters for a Gaussian model in the simulation—that is, a parametric approximation of the underlying noise. The second method is a non-parametric bootstrap procedure, in which no distributional form is imposed; instead, at each simulation step, we resample directly from the stored pool of raw noise samples and inject the resampled values into the system.

The results are similar for both cases with Gaussian-generated samples: for 10,000 Monte Carlo rollouts, the estimated violation probabilities are 0.0074 and 0.011, which are very close to Chanwook's original result (0.0101).

2) Conformal Prediction for Distribution-Free Noise: To obtain disturbance bounds without assuming any parametric noise model, we use distribution-free conformal prediction (CP). A large set of raw process- and measurement-noise samples is first generated and split into calibration and test subsets. Using the calibration samples, we compute absolute deviations from the empirical median and apply the conformal quantile rule to obtain a one-dimensional interval $I_w = [-q_w, q_w]$ for the process noise and a three-dimensional hyper-rectangular set $I_v = [-q_{nd}, q_{nd}] \times [-q_{nv}, q_{nv}] \times [-q_{nve}, q_{nve}]$ for the measurement noise. These sets guarantee finite-sample coverage at level $1 - \alpha$ without relying on Gaussianity. During simulation, noise is resampled from the raw noise pool and clamping each sample to the CP bounds.

We evaluate the effect of the distribution-free conformal bounds under heavy-tailed Cauchy disturbances. The estimated violation probability depends on the choice of the miscoverage level α. For $\alpha = 0.3$, 0.2, and 0.1, the corresponding violation probabilities are 0.0973, 0.1858, and 0.3425, respectively. A smaller α leads to a wider conformal interval, which in turn produces a less restrictive noise set and therefore a higher chance of simulated violations.

We also examine the influence of the raw sample size used to construct the CP bounds. For $\alpha = 0.1$, using $N = 1{,}000$, 10,000, and 10^5 samples yields violation probabilities of 0.3725, 0.3538, and 0.3425. As N increases, the conformal quantile estimates stabilize and the violation probability converges, demonstrating the finite-sample reliability of CP even under heavy-tailed noise.

Figure 3 shows the conformal interval obtained from Cauchy process-noise samples, Figure 4 illustrates the three-dimensional measurement-noise conformal set, and Figure 5 depicts Monte Carlo trajectories color-coded by satisfaction and violation outcomes.

B. Runtime Verification

Our monitoring procedure consists of an offline conformal calibration stage and an online distribution-free risk evaluation stage. First, we generate a large pool of historical process and measurement noise samples and apply conformal quantiles. During execution, the real system evolves under its own stochastic disturbances, while at every time step the monitor (i) records the realized distance and checks for rule violations, (ii) detects whether the realized noise lies inside the conformal sets, and (iii) performs a distribution-free Monte Carlo prediction from the current state by sampling future disturbances from I_w and I_v. Based on the estimated violation probability, the monitor assigns a risk level—low, medium, or high—according to predefined thresholds, and reports elevated risk to issue early warnings.

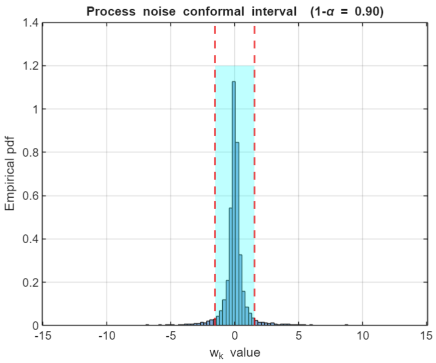

Fig. 3: Conformal interval for Cauchy process noise ($1-\alpha = 0.80$). The histogram shows heavy-tailed samples with CP bounds indicated by dashed vertical lines.

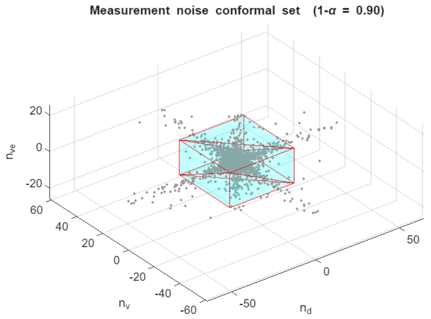

Fig. 4: Three-dimensional conformal set for Cauchy measurement noise. The hyper-rectangular CP region captures the raw noise cloud without assuming Gaussianity.

Figure 6 shows that the monitor overlays the real distance trajectory with risk markers. The middle panel plots the estimated violation probability over the prediction window, revealing short periods where the predicted risk momentarily rises; these correspond precisely to the high-risk markers in the trajectory view. The bottom panel indicates when the realized process or measurement noise falls outside the conformal prediction sets, meaning the environment is deviating from the historical conditions.

Figure 7 further provides the rolling predictions that generate these warnings. At each time step, the monitor propagates multiple future trajectories for 5 time steps and plots the average predicted distance.

V. Scenario Optimization Results

A. Design-Time Verification

Using $N = 5000$ sample trajectories, we apply the Sample-and-Discard framework to determine the boundary parameters.[1] Rather than fixing a single risk ϵ, we utilize Theorem 3 to map the relationship between the probability bound p (denoted as α) and the distance threshold c (denoted as T_{lower}) based on the number of discarded samples k.

The discarding rule employed is a greedy algorithm that removes the k lowest position samples, identifying the lowest remaining sample as the robust lower bound. For a target violation probability $\alpha = 0.05$ (hence $p = 0.95$) and confidence $1 - \beta = 1 - 10^{-5}$, the SD framework yields a lower bound of $c = 30.12$. Moreover, an entire trade-off curve generated by the discarding rule is shown in Figure 8. This data-driven trade-off curve does not stray far from the analytic solutions derived by PYCASSE, $(p^*, c^*) = (0.9438, 31.2500)$, demonstrating the efficacy of SO even when the distribution is unknown.

[1] Experimentation code can be found in Dynamics_SO.ipynb.

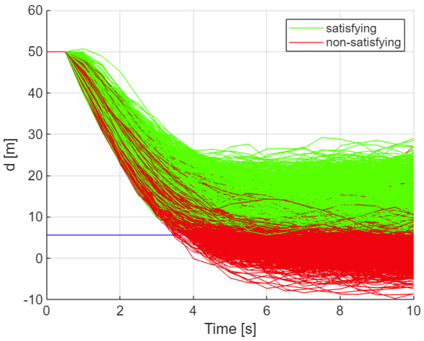

Fig. 5: Monte Carlo trajectories for evaluating safety violations under CP noise bounds. Green trajectories satisfy the distance requirement, while red trajectories violate it.

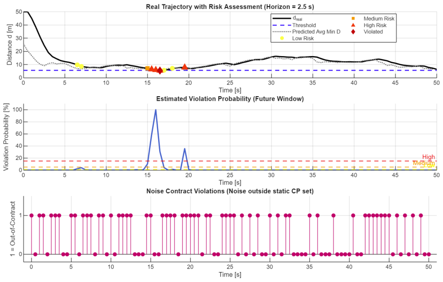

Fig. 6: Online conformal monitoring results: real trajectory with risk levels, predicted violation probability over the future horizon, and detected noise contract violations.

B. Runtime Verification

We extend the static analysis to Runtime Verification (RV) to enable fault detection. Since the contract involves the temporal operator $\mathbb{F}$, the goal is to detect potential contract violations before the trajectory concludes. Here, we present a modification of the static algorithm that yields a highly conservative runtime monitoring algorithm.

Instead of focusing solely on $t = 10$, we derive a dynamic lower tolerance limit for each timestep $t \in [1, 9]$ by applying the SD framework with $N = 5000$ and $\alpha = 0.05$ (Fig. 9). During runtime, a violation is flagged if the system state at **any timestep** drops below these pre-computed thresholds.

The performance of this monitor was evaluated on a test set of 5000 new trajectories (Table I). The empirical violation probability was observed to be $\alpha_{emp} \approx 4.3\%$, satisfying the design constraint $\alpha_{emp} \lesssim \alpha = 5\%$. The results can be interpreted in the following sense:

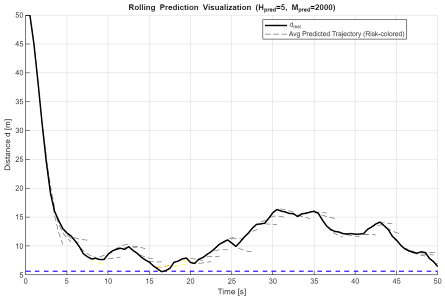

Fig. 7: Rolling prediction visualization: real trajectory and risk-colored average predicted trajectories over the future horizon ($H_{\text{pred}} = 5$, $M_{\text{pred}} = 2000$).

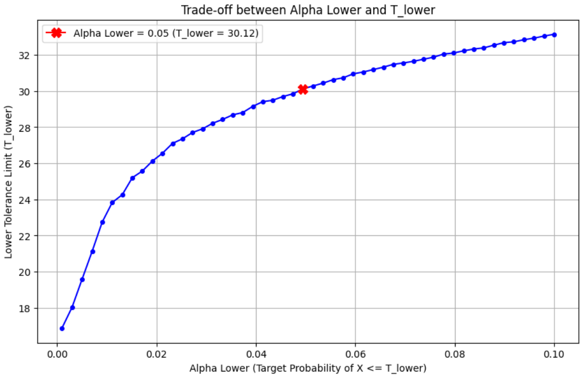

Fig. 8: α vs T_{lower}

- **False Positives (FP)**: 234 cases where the monitor flagged a failure, but the system recovered to satisfy the contract.
- **False Negatives (FN)**: 27 "sudden death" cases where the system remained within bounds until the final timestep, then failed to reach the threshold at the last timestep.

For a more involved analysis, we observe the ratios: Precision, False Positive Rate(FPR), and False Negative Rate(NPR).

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{190}{190 + 234} \approx \mathbf{44.8}\%$$

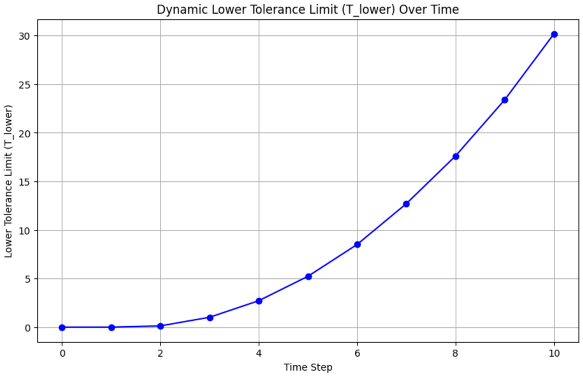

Fig. 9: Lower Tolerance Limit for each timestep

TABLE I: Confusion Matrix for Runtime Verification ($N = 5000, \alpha = 0.05$). TP: True Positive (Correct Detection), FP: False Positive (False Alarm), FN: False Negative (Missed Detection), TN: True Negative (Correct Benign).

		Actual Condition	
		Fail	**Safe**
Pred.	**Fail**	$TP = 190$	$FP = 234$
	Safe	$FN = 27$	$TN = 4549$

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} = \frac{234}{4549 + 234} \approx \mathbf{4.9}\%.$$

$$\text{FNR} = \frac{\text{FN}}{\text{FN+TP}} = \frac{27}{27 + 190} \approx \mathbf{12.4}\%.$$

The first two statistics can be interpreted through the Reflection Principle of random walks. The principle states that "for an unbiased random walk (Brownian motion), the probability of ever crossing a safety barrier is exactly twice the probability of ending up across the barrier." Recall that the randomness of the system is an additive Gaussian with a mean of 0. While our system is not a pure Brownian motion, we can interpret the lower tolerance limits as a moving barrier such that 5% of all trajectories end up crossing it, if we stop at that timestep. This suggests that the probability of a path crossing a barrier and returning (FP) should be similar to the probability of crossing and remaining (TF). Since FN is small compared to TP,

$$\frac{\text{TP}}{N} \lesssim \frac{\text{TP+FN}}{N} = \alpha_{emp}.$$

And by applying the Reflection Principle, we expect

$$\frac{\text{FP}}{N} \approx \frac{\text{TP+FN}}{N} \gtrsim \frac{\text{TP}}{N} \approx \alpha_{emp},$$

implying Precision $\approx \frac{1}{2}$ and FPR $= \frac{\alpha_{emp}}{1-\alpha_{emp}} \approx \alpha_{emp}$ (when $\alpha_{emp} << 1$). The upshot is that with this naive algorithm, we cannot expect precision to be significantly lower than $\frac{1}{2}$. In contrast, the FPR is expected to be in the same regime as the risk parameter α, at least for randomness that simulates an unbiased random walk. Finally, we acknowledge that while FN=27 might seem reasonable, FNR=12.4% could be deemed too high for safety-critical applications. At this time, there is no rigorous justification for the exact numerical values of FN, and we report it as an empirical finding. Finding a mathematical interpretation of FN and developing methods to control it is left as crucial future work.

Another line of future work may investigate varying the risk parameter α across timesteps to increase precision. As illustrated in Figure 10, the majority of FP originate in early timesteps. Relaxing the tolerance limits early in the trajectory (allocating lower α thresholds) could reduce false alarms without significantly compromising other statistics.

VI. Conclusion

We introduced conformal prediction and scenario optimization as two approaches to resolving runtime monitoring

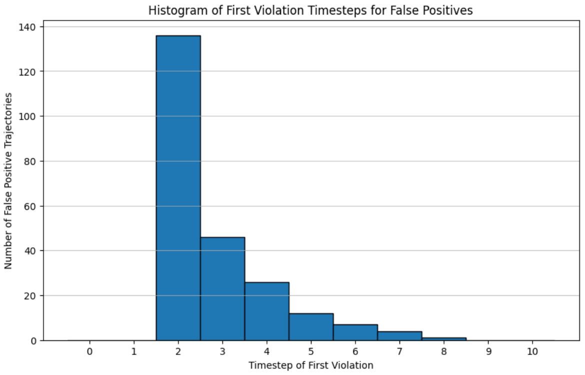

Fig. 10: Histogram of when a False Positive alarm is first raised

for stochastic contracts. Both assume a distribution-free setting, achieving a higher level of flexibility. We investigated both online and offline variants of conformal prediction. The results demonstrate that these methods effectively capture violation probabilities and enable a direct comparison of their performance. The scenario optimization-based approach showed remarkable performance in imitating the solution of a non-distribution-free algorithm. Extension to runtime verification showed nascent but promising results that give some directions for controlling False Positives and False Negatives.

VII. Acknowledgments

We thank GSI/Project Mentor Matthew Low for helpful discussions and guidance on experimental design throughout the semester.

VIII. Project Roles

Wanyue Lin primarily investigated conformal prediction. Joon Kim primarily investigated hypothesis testing and scenario optimization. Both contributed equally in presentations and writing.

IX. Course Relevance/Feedback

This project directly extends the temporal logic portion of the class to StSTL. Lectures on LTL/STL helped us appreciate and understand the background of this project. The first discussion on LTL was useful for jump-starting us into contracts and temporal logic earlier than when they were formally covered in class.

References

[1] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2104–2132, 2015.

[2] L. Lindemann, X. Qin, J. V. Deshmukh, and G. J. Pappas, "Conformal prediction for stl runtime verification," in *Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023)*, 2023, pp. 142–153.

[3] E. Bartocci, Y. Falcone, A. Francalanza, and G. Reger, "Introduction to runtime verification," in *Lectures on Runtime Verification: Introductory and Advanced Topics*. Springer, 2018, pp. 1–33.

[4] A. Bauer, M. Leucker, and C. Schallhart, "Runtime verification for ltl and tltl," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 20, no. 4, pp. 1–64, 2011.

[5] M. Jaeger, K. G. Larsen, and A. Tibo, "From statistical model checking to run-time monitoring using a bayesian network approach," in *International Conference on Runtime Verification*. Springer, 2020, pp. 517–535.

[6] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems," *European journal of control*, vol. 18, no. 3, pp. 217–238, 2012.

[7] P. Nuzzo, H. Xu, N. Ozay, J. B. Finn, A. L. Sangiovanni-Vincentelli, R. M. Murray, A. Donzé, and S. A. Seshia, "A contract-based methodology for aircraft electric power system design," *IEEE Access*, vol. 2, pp. 1–25, 2013.

[8] P. Nuzzo, J. B. Finn, A. Iannopollo, and A. L. Sangiovanni-Vincentelli, "Contract-based design of control protocols for safety-critical cyber-physical systems," in *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2014, pp. 1–4.

[9] O. Maler and D. Nickovic, "Monitoring temporal properties of continuous signals," in *International symposium on formal techniques in real-time and fault-tolerant systems*. Springer, 2004, pp. 152–166.

[10] X. Qin, Y. Xia, A. Zutshi, C. Fan, and J. V. Deshmukh, "Statistical verification of cyber-physical systems using surrogate models and conformal inference," in *2022 ACM/IEEE 13th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2022, pp. 116–126.

[11] P. Nuzzo, J. Li, A. L. Sangiovanni-Vincentelli, Y. Xi, and D. Li, "Stochastic assume-guarantee contracts for cyber-physical system design," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 1, pp. 1–26, 2019.

[12] C. Oh, M. Lora, and P. Nuzzo, "Quantitative verification and design space exploration under uncertainty with parametric stochastic contracts," in *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, 2022, pp. 1–9.

[13] A. Salamati, S. Soudjani, and M. Zamani, "Data-driven verification of stochastic linear systems with signal temporal logic constraints," *Automatica*, vol. 131, p. 109781, 2021.

[14] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[15] D. Boursinos and X. Koutsoukos, "Assurance monitoring of learning-enabled cyber-physical systems using inductive conformal prediction based on distance learning," *AI EDAM*, vol. 35, no. 2, pp. 251–264, 2021.

[16] M. Fontana, G. Zeni, and S. Vantini, "Conformal prediction: a unified review of theory and new challenges," *Bernoulli*, vol. 29, no. 1, pp. 1–23, 2023.

[17] Y. Zhao, B. Hoxha, G. Fainekos, J. V. Deshmukh, and L. Lindemann, "Robust conformal prediction for stl runtime verification under distribution shift," *arXiv preprint arXiv:2311.09482*, 2023.

[18] Z. Mao, C. Sobolewski, and I. Ruchkin, "How safe am i given what i see? calibrated prediction of safety chances for image-controlled autonomy," in *6th Annual Learning for Dynamics & Control Conference*. PMLR, 2024, pp. 1370–1387.

[19] M. C. Campi, S. Garatti, and M. Prandini, "The scenario approach for systems and control design," *Annual Reviews in Control*, vol. 33, no. 2, pp. 149–157, Dec. 2009. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1367578809000479

[20] C. Nandi, A. Monot, and M. Oriol, "Stochastic contracts for runtime checking of component-based real-time systems," in *Proceedings of the 18th International ACM SIGSOFT Symposium on Component-Based Software Engineering*, 2015, pp. 111–116.

[21] F. Cairoli, N. Paoletti, and L. Bortolussi, "Conformal quantitative predictive monitoring of stl requirements for stochastic processes," 2023.

[22] M. C. Campi and S. Garatti, *Introduction to the Scenario Approach*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2018, _eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9781611975444. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611975444

[23] ——, "A Sampling-and-Discarding Approach to Chance-Constrained Optimization: Feasibility and Optimality," *Journal of Optimization Theory and Applications*, vol. 148, no. 2, pp. 257–280, Feb. 2011. [Online]. Available: http://link.springer.com/10.1007/s10957-010-9754-6