

Predictive Zonotope Reduction: Precise Runtime Monitoring under Uncertainty

Vladimir Krsmanović^{1*}, Florian Kohn¹, Bernd Finkbeiner^{1,2}, and Milan Simović³

¹CISPA Helmholtz Center for Information Security ²Technical University of Munich ³beanTech

Abstract—Robots operating in physical environments make control decisions based on uncertain sensor measurements, which can lead to unsafe or suboptimal actions. Runtime monitors that check their behavior against safety specifications must represent this uncertainty soundly. Zonotopes are a widely used representation, but continuously incorporating new measurements grows their order unboundedly, so monitors must periodically apply an over-approximating reduction. The choice of the reduction method substantially affects the zonotope’s precision, yet existing approaches typically utilize a fixed method throughout the run, even though the optimal choice depends on the current state. This paper presents a Predictive Zonotope Reduction (PZR) approach, which frames reducer selection as an optimal control problem and solves it using beam-search model predictive control. Policy distillation into a small neural policy further provides substantially higher execution speed than model predictive control while maintaining improved performance, enabling uncertainty-aware runtime monitoring on resource-constrained real-time systems. We implement our approach in the RLola runtime monitoring framework and evaluate it on a 5-degree-of-freedom robotic arm simulated in MuJoCo, with sensor uncertainty modeled according to ISO 5725. Experiments on a Raspberry Pi 5 show that dynamic reduction significantly lowers false-positive rates in monitoring compared with static reduction strategies.

I. INTRODUCTION

Robots operating in the physical world make control decisions based on sensor inputs. Environmental conditions, such as weather, affect the precision with which sensors capture the true physical state of the measured property. A robot that makes control decisions based on uncertain or imprecise sensor inputs can therefore take suboptimal or unsafe control actions.

One solution to ensure the safety of robots operating in uncertain environments is runtime monitoring [1], [2], [3]. In runtime monitoring, a dedicated system component, called a monitor, observes the robot’s behavior and validates it against a specification of known good or bad behaviors while the robot is operating. This monitor can then be used to intervene in the event of dangerous robot behavior. As the monitor bases its assessment of the robot’s state on its (uncertain) sensor inputs, it has to track measurement uncertainty to provide sound, precise verdicts.

A common approach for representing interval-bounded measurement uncertainty in such systems is the use of zonotopes [4], [5], [6]. Zonotopes are convex, point-symmetric sets used to efficiently represent the set of system states admissible by the measurements.

However, their use in online settings, that is, while the robot is operating, requires continuously integrating new sensor measurements into the zonotope [7], [8]. As more information accumulates in the zonotope, its computational representation grows without bound. This is a major limitation for their application on resource-constrained hardware. In the literature, several zonotope reduction methods address this problem by computing an over-approximation of fixed size [4], [5], [7]. In practice, however, the choice of reduction operator is not obvious and is situation-dependent [9], while strongly affecting the precision of the result.

In this paper, we improve the precision of monitoring verdicts by treating the choice of the reduction method as a sequence problem. Unlike existing works, which restrict the reduction method to remain the same throughout the system’s runtime, we allow different reduction methods to be used over time. As a first solution, we frame this selection process as an optimal control problem, in which the control action at each step is the selection of the optimal zonotope-reduction method. This approach, based on Model Predictive Control (MPC), improves the false-positive rates of monitoring while additionally removing the need for manual method selection or tuning.

To enable use in real-time systems, we refine the selection method through several optimizations. We enable online operation by introducing causal input prediction and reducing computational overhead through the use of beam search for the control policy. To obtain real-time throughput, we further distill the control policy into a neural policy. To improve robustness to covariate shifts, we combine multiple neural policies with DAGger [10] augmentations into an ensemble. Lastly, we augment this ensemble with a control policy to handle ambiguous situations and further improve robustness to extreme events while minimally affecting performance. We show that distilling the control policy into a neural one yields a better policy that runs significantly faster, enabling deployment in real-time systems.

We build our implementation on the RLola monitoring framework [6], which has seen extensive use in cyber-physical systems [1], [3], [11], [12], and provides formal guarantees about its memory bounds, semantics, and soundness [13], [14], [15]. We evaluate our method on a 5-degree-of-freedom robotic arm simulated in MuJoCo [16], with sensor noise and uncertainty modeled in accordance with ISO 5725 [17]. We benchmark our method’s performance on a Raspberry Pi 5 equipped with 16 GB of memory.

The results show that our approach outperforms the state

*Correspondence to vladimir.krsmanovic@cispa.de

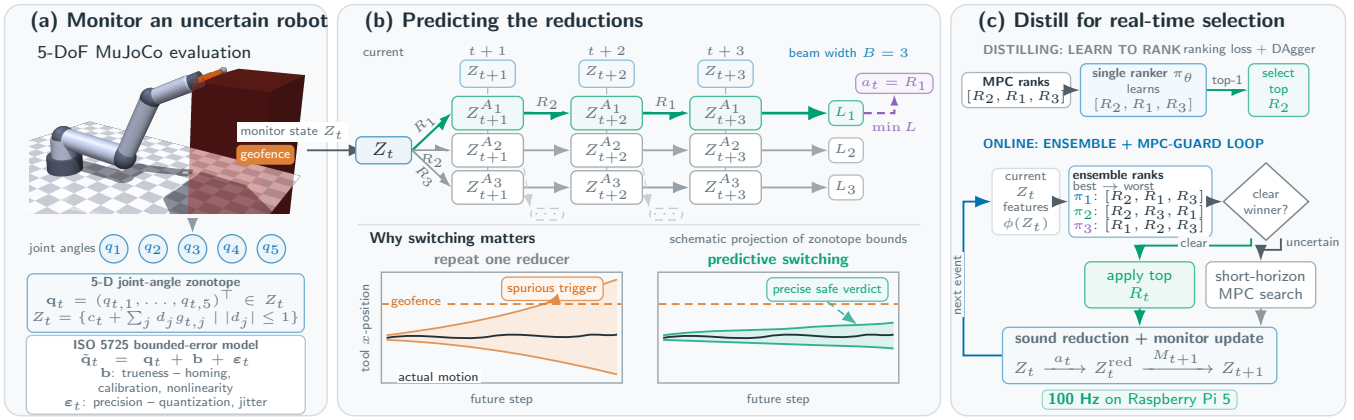


Fig. 1. Overview of *Predictive Zonotope Reduction* (PZR) for uncertainty-aware runtime monitoring. (a) Bounded measurement errors induce a joint-angle zonotope that the monitor checks against a geofence. (b) The PZR evaluates sequences of sound zonotope reduction methods and applies the first operator from the sequence with the lowest predicted terminal approximation loss, which improves accuracy and the false-positive rate (c) We distill control policy rankings into lightweight ranking policies. At runtime, an ensemble selects a reduction method by voting and consults a control policy in difficult situations.

of the art by several orders of magnitude. It produces tighter zonotope approximations overall, resulting in a significantly lower false-positive rate (FPR) for the monitor. In practice, a lower FPR means fewer unnecessary emergency stops or task aborts, which is important in scenarios such as robot learning in uncertain environments, where mistakes during exploration can cause harm. In summary, this paper presents the following contributions:

- We formulate *Predictive Zonotope Reduction* (PZR) as an optimal control problem in which the monitor’s action corresponds to choosing an optimal reduction operator. This captures the central observation that zonotope reduction is not merely a local compression step but an operation that affects the future shape of the uncertainty.
- We instantiate this formulation with a receding-horizon MPC policy that selects reduction methods by optimizing predicted future set tightness. To meet real-time monitoring budgets, we distill the control policy into a DAGger-trained neural ranking policy that selects among sound reduction methods at runtime, composes multiple policies into an ensemble, and integrates the neural and control policies together to improve robustness.
- We implement predictive reducer selection in the RLola monitoring framework, and evaluate it on robotic systems under realistic measurement uncertainty. Compared with the fixed-reduction-method baselines, our method produces tighter over-approximations, reduces the false-positive rate, and achieves real-time inference speeds.

II. BACKGROUND AND RELATED WORK

A. Zonotopes

A zonotope is a convex, point-symmetric set defined by a center ($c \in \mathbb{R}^d$) and a set of generators ($g_1, \dots, g_k \in \mathbb{R}^d$). Its points are given by the Minkowski sum of its generators

translated by its center:

$$\mathbb{Z} \triangleq \{c + d_1 g_1 + \dots + d_k g_k \mid d_1, \dots, d_k \in [-1, 1]\}.$$

Zonotopes are widely used in reachability analysis and control theory due to their computational efficiency [4], [18]. In reachability analysis, zonotopes represent sets of states reachable under all possible input sequences. In contrast, this paper considers runtime monitoring, which assesses the safety of a single system execution at runtime. Rather than propagating zonotopes solely according to a system model, runtime monitors continuously refine their state estimates using new sensor measurements. This is related to set-based observers from control theory [19], [20], which estimate system states from uncertain observations. The key difference is that runtime monitors aim to provide sound Boolean verdicts about system health rather than precise estimates of an unobservable system state.

All of these applications face the challenge of controlling the order of the zonotope to keep computations feasible. Adaptive reduction methods [21], [22], for example, tune their parameters for a given zonotope according to a cost function. Our approach addresses a different problem: Rather than optimizing the reduction of an individual zonotope, we select reduction methods sequentially across a stream of zonotopes. This allows the choice of reduction method to depend on the system’s evolving state and the anticipated effects of future reductions.

B. Runtime Verification

Runtime monitoring [23] is a safety-assurance mechanism widely applied to cyber-physical systems, particularly in safety-critical settings [24], [25]. In stream-based runtime monitoring [26], a monitor observes a system through input streams representing sensor measurements. Output streams define aggregations and transformations of input and other output streams, while triggers specify safety properties that express Boolean verdicts about the system’s behavior.

Most stream-based monitoring tools, such as TeSSLa [2], Striver [27], or Copilot [28] do not explicitly account for uncertainty in sensor measurements and may therefore produce verdicts with false confidence. The extension of Lola [26] to symbolic inputs [29] models uncertainty through symbolic inputs but relies on interval-based methods, providing overly pessimistic verdicts.

In this paper, we focus on RLola [6], the robust extension of RTLola [30] that has seen extensive use in autonomous and safety-critical systems, including UAVs [1], [11], [31], as well as deep integration into the robotics software stack via a ROS adapter [12]. The tool is validated in practice, supported on different hardware [32], [33], well documented [30], [34], and provides strong memory guarantees and formally verified compilers and semantics [13], [14], [15].

C. Model Predictive Control

Model Predictive Control (MPC) is a receding-horizon approach to optimal control. For a system with state x_k , control action u_k , and dynamics $x_{k+1} = f(x_k, u_k)$, MPC solves the finite-horizon problem

$$\min_{u_k, \dots, u_{k+H-1}} \sum_{j=0}^{H-1} \ell(x_{k+j}, u_{k+j}) + V_f(x_{k+H}), \quad (1)$$

where H is the horizon, ℓ is the stage cost, and V_f is an optional terminal cost. Only the first action of the optimal sequence is applied; after the system advances, the optimization is solved again from the new state. This feedback loop lets the controller account for delayed consequences of its current action, but it requires a model of how the state evolves under candidate actions and, when relevant, a forecast or preview of exogenous inputs.

Robust Model Predictive Control is a broad field that focuses on optimizing physical control inputs subject to constraints while accounting for bounded disturbances and model uncertainty [35]. While these approaches focus on control *under uncertainty*, we use MPC to control *the evolution of the uncertainty* representation itself - the control actions do not actuate a robot but instead shape the zonotopes.

III. PREDICTIVE ZONOTOPE REDUCTIONS

Existing work treats zonotope reduction as a continuous application of the same method. This paper views the problem of reducing the order of zonotopes as a sequential decision problem and formulates it as a control problem. Intuitively, every zonotope reduction affects the future because it shapes the resulting zonotope, which will serve as the basis for future reductions. Viewing this as a sequential problem, we focus on finding the best reduction methods not just for the current timestep but for the system's entire runtime.

Varying the zonotope reduction method during the monitor's execution does not affect the approach's validity [6]. The underlying monitoring algorithm is agnostic of the applied reduction method, and its soundness guarantee for the emitted verdicts relies only on the reduction method chosen at each step to correctly over-approximate the zonotope. It

is then guaranteed that the monitor never misses an actual violation of the specification; that is, false negatives are excluded by construction. The number of false positives, i.e., the monitor reports a violation even though the specification is not violated, largely depends on the quality of the over-approximation.

We use these insights to formulate zonotope reduction over the full measurement trace as an optimal control problem: We select actions aiming to minimize the final size of the zonotope representing the monitor's state, thereby reducing the uncertainty within it. The problem of *Predictive Zonotope Reduction* (PZR) is the optimal control problem of minimizing the loss $\bar{L}$ in the last step of a sequence of zonotope reduction operations. Such a monitoring sequence is depicted in Fig. 2. There, the reference states $Z_0, \dots, Z_k$ represent the most accurate representation of the system state obtainable. The approximate states $Z^A_0, \dots, Z^A_k$ represent over-approximations of these states as a result of a sequence of zonotope reduction operations $a_0, \dots, a_k$.

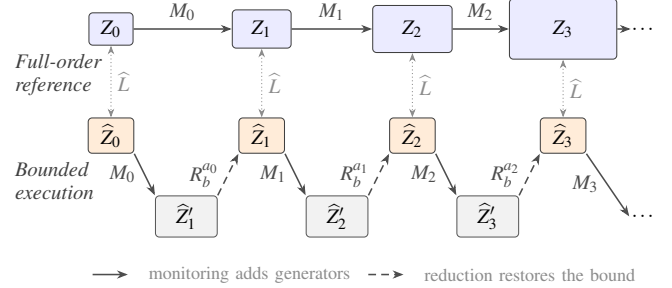


Fig. 2. The sequence problem visualized. $M_0, \dots, M_k$ are monitoring steps that transform the state zonotopes Z_i and Z^A_i given the input measurements. $a_0, \dots, a_k$ is the sequence of zonotope reduction operators.

Framed as a control problem, the system action at each step is the *choice* of a zonotope reduction method from a predefined set. The state of the control problem is the full symbolic monitor state, including its zonotope geometry, and the control action is the selection of a sound reduction method. We use zero-stage cost and the Squared Hull Error relative to an unreduced monitor rollout as the cost, while reducer feasibility and the transform bound provide the constraints. The action is chosen via an MPC loop that optimizes the loss function, which is directly linked to the zonotope's resulting shape.

To quantify the precision of an over-approximation, we adapt a previously proposed monitoring-specific method [6]. The metric directly estimates the false-positive rate of trigger conditions using zonotopes representing the monitor state. Since trigger conditions correspond to axis-aligned half-space queries over the zonotope, the resulting loss is defined as the dimension-wise mean squared error between the interval hulls of two zonotopes:

Definition 1 (Squared Hull Error): Let $Z = (c, g_1, \dots, g_k)$ and $\hat{Z} = (\hat{c}, \hat{g}_1, \dots, \hat{g}_k)$ be zonotopes in $\mathbb{R}^d$ with $c = \hat{c}$. The

squared hull error of Z relative to $\hat{Z}$ is

$$\bar{L}(Z, \hat{Z}) \triangleq \sum_{i=1}^d \left(\sum_j^k |g_{j,i}| - \sum_j^{\hat{k}} |\hat{g}_{j,i}| \right)^2$$

At each time step, Predictive Zonotope Reductions selects the next reduction action by minimizing predicted Squared Hull Error over a future finite sequence of decisions. For a decision depth H that counts the current event, let Z_{t+H}^A be the zonotope state obtained by rolling out a sequence of zonotope reduction actions $A = (a_t, \dots, a_{t+H-1})$ from a set of all possible method combinations A^H over the fixed finite horizon H . Let Z_{t+H} be the reference zonotope on which no reduction method was applied. We pick the next zonotope reduction method a_t by finding the optimal sequence of reductions A^* over the horizon, and select the starting action of that sequence as $a_t = (A^*)_t$.

$$A^* \in \arg \min_{A \in A^H} \bar{L}(Z_{t+H}^A, Z_{t+H}), \quad (2)$$

We perform this optimization search in every state of the system in which the zonotope size exceeds the bound b , with no reduction being applied if the size is within the bound. The search discards infeasible candidates when the method fails due to broken assumptions and ensures that a feasible selection method always exists. The selected action a_t gets applied, and the search is repeated in the next timestep.

A. Recorded-Future Reductions

In offline monitoring, where the traces are prerecorded, or where the zonotope-reduction decision is staggered (effectively giving us a Perfect Event Preview), the future events are precisely known. Importantly, known measurements of the system do not eliminate uncertainty, as each recorded value remains subject to bounded measurement error within the monitor. In this case, the control decision on which zonotope reduction method to select can be made precisely based on the known future inputs. Starting from the current monitor state, we roll each candidate schedule through the recorded current and future events, with an unreduced branch (with no zonotope reductions applied) serving as the reference. We refer to this method, which uses a full rollout with the known three future inputs, as *MPC-F*.

However, enumerating all possible combinations of method selections over the horizon is computationally prohibitive due to the exponential size of the space. To address this and enable the use of more methods and a longer horizon, we adapt beam search [36] as a constrained search method. After each event, the search retains the W prefixes with the lowest current terminal loss, expands them with every reducer, and finally commits the first action of the best complete schedule. We refer to this variant as *MPC-B*, which considers the next five future input events and explores them using a beam search with width $W = 4$.

B. Causal Predictive Reductions

For a more realistic scenario, we have to consider an online setting in which future events are not precisely known and must be predicted. We adapt our approach to this setting by

implementing a causal prediction step that predicts future events for the MPC rollout from previously observed inputs. While these predictions can lead to different reducer choices and cannot predict outlier events, they provide a strong basis for the online setting of our method without sacrificing soundness and safety, as these guarantees are preserved by our zonotope reduction methods. The method *MPC-L* performs Predictive Zonotope Reductions with a Horizon Length of $H = 5$ and a beam search width of $W = 4$. Future inputs are predicted by linearly extrapolating from previously seen inputs.

C. Learned Policy Distillation

While the causal prediction of future inputs enables our method to be used in the online setting, where these inputs are not known in advance, the computational bottleneck of MPC remains restrictive for realistic deployments. To enable this, we *distill* the control policies into a small neural network. The neural policies act as a student network, learning from the teacher outputs of the control policies, as determined by one of the previously mentioned Predictive Zonotope Reduction methods.

We set up the teacher-student setting with the same possible reduction actions, but focus the neural policy on minimal resource requirements and maximal decision throughput. While the control policies can roll out future inputs and search for the optimal sequence of reductions, we limit the student inputs to scenario-agnostic features of the current zonotope. While this leads to an information-asymmetric setting, which can limit performance, several adaptations preserve our approach's performance.

Crucially, the student policy is not trained on a classification objective ("what is the optimal reduction method for this step") but on a current-state ranking policy. In each state, the student policy must provide a pairwise ranking of possible reducers that the network can choose. These pairwise rankings yield a ranked list of preferred reduction methods for each state. This learning objective provides a significantly denser learning signal, especially given the imbalance in method-selection choices produced by the control policies (as discussed and shown in Sec. IV). Because the student network can choose only among sound reduction methods, it preserves the same soundness guarantees as the teacher-control policies.

As the learned policies are trained via distillation on sampled traces, the resulting policies are vulnerable to the well-known issue of covariate shift. To improve their robustness, we combine several learned policies into an ensemble. At each step, the policies independently rank the reduction operators and vote for the operator to apply, with ties resolved using rank and confidence. To improve overall robustness and introduce diversity into the ensemble, each policy is independently augmented with sampled traces using DAgger [10]. We refer to these variants as *Vote*. To further improve robustness and reduce the covariate shift, we extend the ensemble with a control policy *Vote-Guarded*. In uncertain cases, where the ensemble votes do not select

a clear winner, rather than breaking ties based on secondary metrics, the next reduction method is selected based on a prediction made by one of the control policies using a short horizon and beam width.

IV. EVALUATION

We evaluate our methods on the robot-arm monitoring scenario illustrated in Fig. 1. We simulate a 5-DoF MuJoCo robot arm [16] at 10Hz and model measurement error according to ISO 5725 [17]. The error model includes quantization error, jitter, integral nonlinearity due to magnet displacement, and homing and calibration errors. The robot arm is monitored by an RLola monitor against a specification that prevents it from entering a forbidden region. We run the experiments on a Raspberry Pi 5 with 16GB of memory running Raspbian 64-bit.

A. Experimental Setup

a) Methods: Predictive Zonotope Reduction policies choose from four reduction methods described in existing literature - namely Girard [18], Scott [8], PCA [9], and Combastel [19] reducers. While many other methods exist, experiments have shown that they are rarely selected by policies and that their standalone performance is significantly worse than that of the four mentioned methods. We implement these methods in RLola, with our Rust implementations tracking their implementation in the CORA tool [5].

b) Controllers: For control policies, we evaluate on the previously mentioned variations of MPC policies. MPC-B and MPC-F have perfect previews of future inputs, while MPC-L has to predict them causally. The policies are evaluated over the horizon length of $H = 5$, and MPC-L and MPC-B search over the beam width of $W = 4$. We introduce the MPC-Greedy ($H = 1$) baseline, which selects the best next action at the current state and can be viewed as a restricted version of our methods, highlighting the importance of switching between zonotope reduction methods.

We construct the neural policies as two dense layers of 32 neurons each, with ReLU activation, mapping the 15 input zonotope properties to 4 lower-is-better preference scores, thereby creating a ranking of methods. As discussed in Sec. III-C, the policies do not have access to information such as stream values, history, or predicted rollouts. They are distilled over 128 randomly sampled traces using MPC-F as the teacher policy. We refer to this neurally distilled policy as G15. For the ensemble method described before, we combine three G15 policies, each augmented with 6 additional independently sampled random traces. The Vote3Guard method follows the control policy guard in case the three policies do not unanimously select a reduction method.

c) Cohorts and Metrics: We evaluate the performance of our methods across a range of sampled robot arm behaviors and zonotope budget bounds (the maximal shape of the zonotope representation), showcasing a connection between method selection and memory bounds $b \in \{40, 80, 120, 150, 200, 250, 500\}$. We generate 20 random-waypoint executions and monitor the traces at 10Hz using the geofence specification.

Our primary metric is the Squared Hull Error $\bar{L}$ (from Definition 1), computed with respect to a ground-truth reference zonotope. We also report the false-positive rate (FPR), whose denominator is the number of negative reference events, and the p99 latency for reduction method selection and execution. While $\bar{L}$ and the FPR are correlated, they need not coincide, as a larger zonotope can still be better aligned with the monitor predicate. We note that the false-negative rate is 0 across all methods, and we do not report it in tables or figures for conciseness. All controllers measured use sound overapproximation methods; thus, reductions can affect tightness and false interventions but not soundness¹.

B. Results and Analysis

In total, we performed 1,400 monitoring runs, and present the results and their analysis in the following section.

Table I summarizes the performance of all methods at $b = 150$, while Fig. 3 reports approximation loss, FPR, and selection-plus-commit latency across all transform bounds. Table II and Fig. 4 summarize reducer choices, Fig. 5 shows the behaviour of the control policy guard, and Fig. 6 reports Raspberry Pi 5 latency across budget bounds.

a) Dynamic Selection beats Fixed Reduction Methods:

Table I shows that Predictive Zonotope Reductions produce tighter zonotopes and fewer false positives than every fixed reducer. At $b = 150$, the best fixed reducer has an FPR of 29.68 %, compared with 1.76 % to 5.08 % for the dynamic selectors. Fig. 3 shows that this advantage persists across the transform bounds. Furthermore, the performance and throughput plot in Fig. 3 shows that fixed schedules are fast but loose, that PZR control policies are precise but computationally expensive, and the neural PZR policies retain MPC-scale loss while achieving the speed required for real deployment.

TABLE I
MEDIAN $\bar{L}$ AND FPR OVER 20 HELD-OUT TRACES AT $b = 150$.

Method	$\bar{L} \downarrow$	FPR (%) $\downarrow$
<i>Fixed reducer</i>		
Girard	6.7×10^{-4}	32.60
Scott	7.6×10^{-7}	39.70
PCA	1.4×10^{-1}	62.11
Combastel	6.4×10^{-4}	29.68
<i>Recorded-Future Reductions</i>		
MPC-F	2.1×10^{-7}	1.80
MPC-B	1.5×10^{-7}	1.76
<i>Causal Predictive Reductions</i>		
MPC-Greedy ($H = 1$)	2.6×10^{-7}	1.96
MPC-L	1.6×10^{-7}	1.78
G15/Clean148	1.9×10^{-7}	5.08
Vote3	2.1×10^{-7}	1.92
Vote3-Guarded	1.9×10^{-7}	1.79

MPC-Greedy ($H = 1$) is the simplest PZR instance: it exhaustively scores the four reducers for the current decision

¹Nevertheless, we track the metric during evaluations and confirm that no evaluated trace produces a false negative over approximately 700,000 monitored events

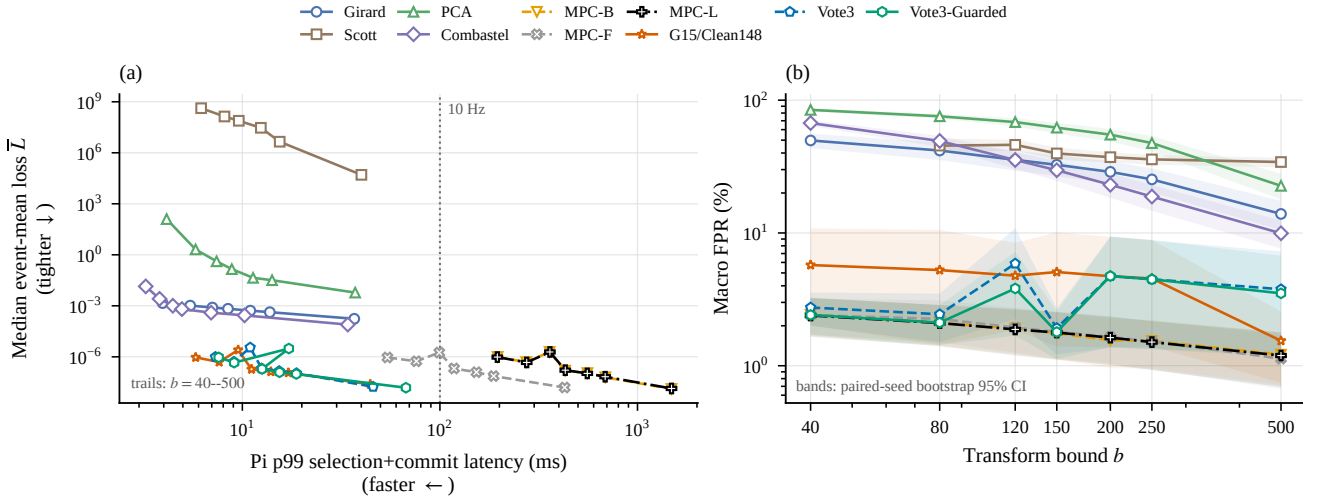


Fig. 3. Quality-latency trade-off across transform bounds; lower is better on every vertical axis. (a) Median event-mean approximation loss over 20 traces against median p99 selection-plus-commit latency over five Raspberry Pi 5 traces. (b) FPR over 20 traces with paired-seed bootstrap 95 % intervals.

without a future rollout. At $b = 150$, switching alone reduces the best fixed median loss from 6.4×10^{-4} to 2.6×10^{-7} and the FPR from 29.68 % to 1.96 %. MPC-L then scores four causally predicted future events in addition to the current decision, reaching 1.6×10^{-7} loss and 1.78 % FPR; across the full paired sweep, it improves over Greedy in 107 of 140 cells.

b) Reduction Methods act complementary over long traces: The analysis of the selected methods shows that the performance of reduction methods does not strictly correlate with their performance when combined alongside other methods over the full execution of the system. Table II shows the frequency with which the PZR policies select specific reduction methods.

TABLE II
REDUCER-SELECTION FREQUENCIES (%) OF DIFFERENT POLICIES.

Controller	Girard	Scott	PCA	Combastel
MPC-B	6.73	85.55	3.03	4.69
MPC-F	8.65	84.24	2.38	4.73
MPC-L	6.72	85.56	3.20	4.52
G15/Clean148	10.99	88.00	0.46	0.56
Vote3	10.41	87.69	0.25	1.66
Vote3-Guarded	9.13	89.12	0.06	1.69

Perhaps surprisingly, the dynamic controllers choose Scott most often of all methods despite its poor standalone performance. We attribute this to the Scott method being vulnerable to poorly shaped matrices, which often lead to poor performance or even complete failure². However, the application of the Girard or PCA methods at the beginning of the trace, and their occasional use throughout (as shown in Fig. 4), is crucial. We hypothesize that their use conditions the zonotope representation into a form optimal for applying

²Scott's method assumes the zonotope matrix is invertible, which often fails during longer executions

Scott's methods, leading to significant performance gains observed in Fig. 3. This shows the contrast between the dynamic and static zonotope reduction settings, as well as the complementary interactions among the various methods.

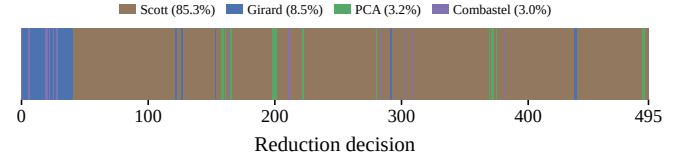


Fig. 4. Method selections of MPC-L on a selected trace showing the ration and distribution of reduction method selection decisions

c) Learned Policies can learn the control behavior with minimal inputs: The results in Tables I and Fig. 3 show that learned policies distilled from the control policies can retain strong performance despite having access to only a limited set of information compared to the control policies, and that they significantly outperform the static methods at minimal runtime cost.

However, a deeper analysis of the method's performance reveals that simple distillation occasionally fails under co-variate shift. These occasional failures motivate the ensemble and control policy guard measures introduced in Sec. III-C. As shown in Table I and Fig. 3, these additions help mitigate these issues, although they do not completely resolve them. We find that Vote3-Guarded helps recover performance on the problematic traces compared to the plain learned policies.

Table II and Fig. 5 show how the control policy guard helps resolve part of these issues. An interesting analysis emerges from examining the Guard method's actions, which override those of the ensembles when the ensembles fail to select a method unanimously. We find it turns on rarely and does not always strictly replace the method that would have been selected via tie-breaking, as in the pure ensemble variations. Looking at 69,120 method selection decisions, 94.02 % were unanimous, while the guard was activated on

the remaining 5.98%, of which 2.65% led to a method change compared to the tie-breaker method.

As shown in Fig. 5, most of the overrides seem to encourage exploitation of the Scott methods’ performance, with occasional redirection from Scott to Girard or PCA. These actions help stabilize the policy behavior by encouraging active exploitation of Scott when available, while also helping to avoid failure modes by occasionally intervening with PCA and Girard overrides to condition the matrices for follow-on reductions.

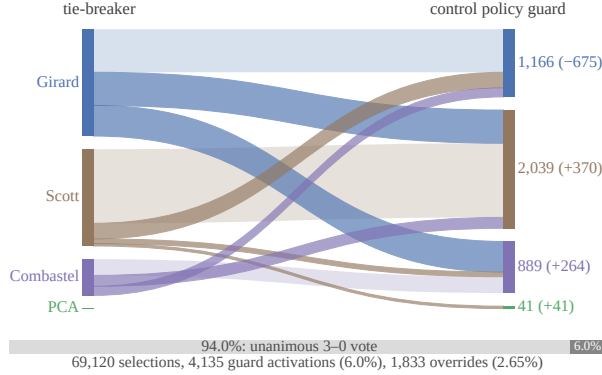


Fig. 5. Changes in selected reduction methods between tie-breaking and a control policy guard over 69,120 selections. Pale bands preserve the vote winner, saturated bands are overrides, and brackets indicate net changes.

While the occasional failures of the learned policies are a limitation of the method, several ways of fixing it remain open for future work, such as shadow backup copies, the addition of history, or, most directly, removing the information asymmetry by adopting a more powerful input policy. Importantly, this limitation never threatens the system’s safety, as the construction of the Predictive Zonotope Reductions ensures safety-by-construction.

d) Learned Polies Can Run Fast: We demonstrate that our method can perform runtime monitoring of embedded systems, with minimal resources and high throughput. We profile the latency of our methods on a single isolated Cortex-A76 core of a Raspberry Pi 5. Figs. 6 and 3 show that the learned selectors combine low approximation loss of the control policies with low latency. Control policies can sustain the 10Hz throughput at its lower bound, but, more importantly, we show that the learned policies enable a significant boost in throughput while maintaining high accuracy as can be observed in Fig. 3.

This also applies to the Vote3 Guarded method, which occasionally runs a control policy, but its selective use still keeps its performance close to that of static methods such as Scott or Girard. Overall, neural PZR policies distilled from the control PZR can combine the performance and high throughput required for deployments. The distilled neural policies introduce negligible overhead and are primarily bottlenecked by the throughput of the static reduction methods they select. This shows potential for even faster performance on hardware with greater processing power or multi-core capabilities.

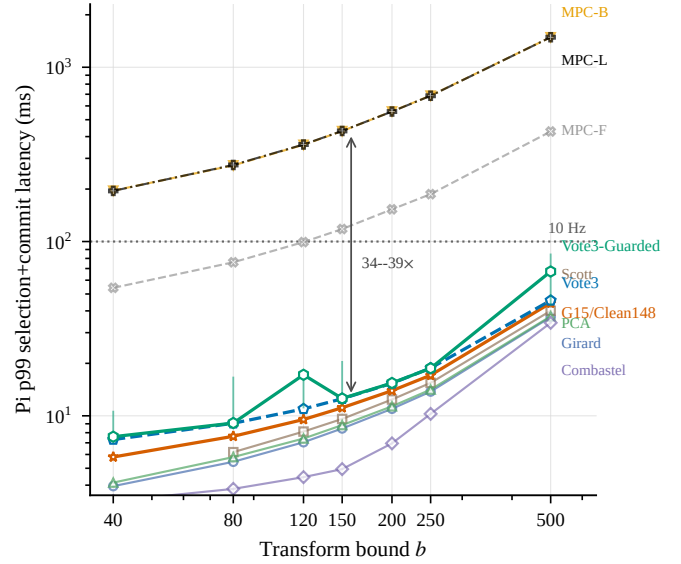


Fig. 6. Pi p99 latency across bounds. Points are five-seed medians, whiskers are observed ranges, and the dotted line marks 100 ms.

V. LIMITATIONS

Our policies select among existing zonotope-reduction methods (we do not construct new ones), and our method’s performance is tied to theirs. Our work shows that switching between methods, as well as predicting future inputs, provides significant benefits, and we view it as deeply complementary. This, however, comes at a computational cost, and the balance among horizon, beam width, and size bounds depends on the specific problem and deployment. As the physical ground truth is generally unknown, we compute the loss metrics against a predicted ground truth derived online from unreduced rollouts - this is an unavoidable part of working with uncertainty. The learned policies presented aim to maximize throughput on resource-constrained embedded hardware, which is prone to covariate shift. Importantly, these issues never risk loss of soundness or safety - they only affect the zonotope tightness.

VI. CONCLUSION

In this paper, we introduced Predictive Zonotope Reduction, a sequential control framework for online uncertainty monitoring in autonomous systems. By treating zonotope reduction as an optimal control problem rather than applying a static method throughout the run, our approach significantly improves representation accuracy by orders of magnitude and reduces vacuous alarms arising from false positives.

To satisfy embedded real-time constraints, we distilled receding-horizon MPC controllers into lightweight neural ranking policies augmented with a DAgger ensemble and a short-horizon control guard. We evaluated our method on a 5-DoF robotic arm simulated in MuJoCo, with measurement uncertainty per ISO 5725. Deployed on a Raspberry Pi 5, we sustain high-frequency monitoring while preserving formal safety guarantees by construction.

VII. ACKNOWLEDGMENTS

This work was partially supported by the German Research Foundation (DFG) as part of PreCePT (FI 936/7-1; FR 2715/6-1) and partially supported by the European Research Council (ERC) Grant HYPER (No. 101055412). Views and opinions expressed are, however, those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Florian Kohn carried out this work as a member of the Saarbrücken Graduate School of Computer Science.

REFERENCES

- [1] J. Baumeister et al., “Monitoring unmanned aircraft: Specification, integration, and lessons-learned,” in *Computer Aided Verification - 36th International Conference, CAV 2024, Montreal, QC, Canada, July 24-27, 2024, Proceedings, Part II*, A. Gurfinkel and V. Ganesh, Eds., ser. Lecture Notes in Computer Science, vol. 14682, Springer, 2024, pp. 207–218. DOI: 10.1007/978-3-031-65630-9_10 [Online]. Available: https://doi.org/10.1007/978-3-031-65630-9_10
- [2] L. Convent, S. Hungerecker, M. Leucker, T. Schefel, M. Schmitz, and D. Thoma, “Tessla: Temporal stream-based specification language,” in *Formal Methods: Foundations and Applications - 21st Brazilian Symposium, SBMF 2018, Salvador, Brazil, November 26-30, 2018, Proceedings*, T. Massoni and M. R. Mousavi, Eds., ser. Lecture Notes in Computer Science, vol. 11254, Springer, 2018, pp. 144–162. DOI: 10.1007/978-3-030-03044-5_10 [Online]. Available: https://doi.org/10.1007/978-3-030-03044-5_10
- [3] S. Biewer, B. Finkbeiner, H. Hermanns, M. A. Köhl, Y. Schnitzer, and M. Schwenger, “Rtlola on board: Testing real driving emissions on your phone,” in *Tools and Algorithms for the Construction and Analysis of Systems - 27th International Conference, TACAS 2021, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2021, Luxembourg City, Luxembourg, March 27 - April 1, 2021, Proceedings, Part II*, J. F. Groote and K. G. Larsen, Eds., ser. Lecture Notes in Computer Science, vol. 12652, Springer, 2021, pp. 365–372. DOI: 10.1007/978-3-030-72013-1_20 [Online]. Available: https://doi.org/10.1007/978-3-030-72013-1_20
- [4] M. Althoff, G. Frehse, and A. Girard, “Set propagation techniques for reachability analysis,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 369–395, 2021.
- [5] M. Althoff, “An introduction to CORA 2015,” in *Proc. of the 1st and 2nd Workshop on Applied Verification for Continuous and Hybrid Systems*, EasyChair, 2015, pp. 120–151. DOI: 10.29007/zbkv [Online]. Available: <https://easychair.org/publications/paper/xMm>
- [6] B. Finkbeiner, M. Fränzle, F. Kohn, and P. Kröger, “Cutting corners on uncertainty: Zonotope abstractions for stream-based runtime monitoring,” *CoRR*, vol. abs/2601.11358, 2026. DOI: 10.48550/ARXIV.2601.11358 arXiv: 2601.11358. [Online]. Available: <https://doi.org/10.48550/arXiv.2601.11358>
- [7] B. Finkbeiner, M. Fränzle, F. Kohn, and P. Kröger, “Stream-based monitoring under measurement noise,” in *Runtime Verification - 24th International Conference, RV 2024, Istanbul, Turkey, October 15-17, 2024, Proceedings*, E. Ábrahám and H. Abbas, Eds., ser. Lecture Notes in Computer Science, vol. 15191, Springer, 2024, pp. 22–39. DOI: 10.1007/978-3-031-74234-7_2
- [8] J. K. Scott, D. M. Raimondo, G. R. Marseglia, and R. D. Braatz, “Constrained zonotopes: A new tool for set-based estimation and fault detection,” *Autom.*, vol. 69, pp. 126–136, 2016. DOI: 10.1016/J.AUTOMATICA.2016.02.036 [Online]. Available: <https://doi.org/10.1016/j.automatica.2016.02.036>
- [9] A. Kopetzki, B. Schürmann, and M. Althoff, “Methods for order reduction of zonotopes,” in *56th IEEE Annual Conference on Decision and Control, CDC 2017, Melbourne, Australia, December 12-15, 2017*, IEEE, 2017, pp. 5626–5633. DOI: 10.1109/CDC.2017.8264508 [Online]. Available: <https://doi.org/10.1109/CDC.2017.8264508>
- [10] S. Ross, G. J. Gordon, and D. Bagnell, “A reduction of imitation learning and structured prediction to no-regret online learning,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, G. J. Gordon, D. B. Dunson, and M. Dudík, Eds., ser. JMLR Proceedings, vol. 15, JMLR.org, 2011, pp. 627–635. [Online]. Available: <http://proceedings.mlr.press/v15/ross11a/ross11a.pdf>
- [11] J. Baumeister, B. Finkbeiner, S. Schirmer, M. Schwenger, and C. Torens, “Rtlola cleared for take-off: Monitoring autonomous aircraft,” in *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II*, S. K. Lahiri and C. Wang, Eds., ser. Lecture Notes in Computer Science, vol. 12225, Springer, 2020, pp. 28–39. DOI: 10.1007/978-3-030-53291-8_3 [Online]. Available: https://doi.org/10.1007/978-3-030-53291-8_3

- [12] J. Baumeister, B. Finkbeiner, F. Jünger, F. Kohn, S. Schirmer, and C. Torens, "A ROS adapter for rtlola," in *Runtime Verification - 25th International Conference, RV 2025, Graz, Austria, September 15-19, 2025, Proceedings*, B. Könighofer and H. Torfah, Eds., ser. Lecture Notes in Computer Science, vol. 16087, Springer, 2025, pp. 92–100. DOI: 10.1007/978-3-032-05435-7_6 [Online]. Available: https://doi.org/10.1007/978-3-032-05435-7_6
- [13] J. Baumeister, J. C. Dauer, B. Finkbeiner, and S. Schirmer, "Monitoring with verified guarantees," *Int. J. Softw. Tools Technol. Transf.*, vol. 25, no. 4, pp. 593–616, 2023. DOI: 10.1007/S10009-023-00712-3 [Online]. Available: <https://doi.org/10.1007/s10009-023-00712-3>
- [14] F. Kohn, A. Correnson, J. Baumeister, and B. Finkbeiner, "Pacing types for asynchronous stream equations," in *Formal Methods - 27th International Symposium, FM 2026, Tokyo, Japan, May 18-22, 2026, Proceedings, Part II*, A. Sampaio and M. Stoelinga, Eds., ser. Lecture Notes in Computer Science, vol. 16557, Springer, 2026, pp. 154–173. DOI: 10.1007/978-3-032-26220-2_8 [Online]. Available: https://doi.org/10.1007/978-3-032-26220-2_8
- [15] B. Finkbeiner, S. Oswald, N. Passing, and M. Schwenger, "Verified rust monitors for lola specifications," in *Runtime Verification - 20th International Conference, RV 2020, Los Angeles, CA, USA, October 6-9, 2020, Proceedings*, J. Deshmukh and D. Nickovic, Eds., ser. Lecture Notes in Computer Science, vol. 12399, Springer, 2020, pp. 431–450. DOI: 10.1007/978-3-030-60508-7_24 [Online]. Available: https://doi.org/10.1007/978-3-030-60508-7_24
- [16] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109
- [17] International Organization for Standardization, "Accuracy (trueness and precision) of measurement methods and results — part 1: General principles and definitions," International Organization for Standardization, Geneva, CH, Standard ISO 5725-1:2023, Jul. 2023.
- [18] A. Girard, "Reachability of uncertain linear systems using zonotopes," in *Hybrid Systems: Computation and Control, 8th International Workshop, HSCC 2005, Zurich, Switzerland, March 9-11, 2005, Proceedings*, M. Morari and L. Thiele, Eds., ser. Lecture Notes in Computer Science, vol. 3414, Springer, 2005, pp. 291–305. DOI: 10.1007/978-3-540-31954-2_19
- [19] C. Combastel, "A state bounding observer based on zonotopes," in *7th European Control Conference, ECC 2003, Cambridge, UK, September 1-4, 2003*, IEEE, 2003, pp. 2589–2594. DOI: 10.23919/ECC.2003.7085991
- [20] T. Alamo, J. M. Bravo, and E. F. Camacho, "Guaranteed state estimation by zonotopes," in *42nd IEEE Conference on Decision and Control, CDC 2003, Maui, Hawaii, USA, December 9-12, 2003*, IEEE, 2003, pp. 5831–5836. DOI: 10.1109/CDC.2003.1271935
- [21] M. Wetzlinger, A. Kulmburg, A. Le Penven, and M. Althoff, "Adaptive reachability algorithms for nonlinear systems using abstraction error analysis," *Nonlinear Analysis: Hybrid Systems*, vol. 46, p. 101 252, 2022.
- [22] M. Wetzlinger, N. Kochdumper, and M. Althoff, "Adaptive parameter tuning for reachability analysis of linear systems," in *2020 59th IEEE conference on decision and control (CDC)*, IEEE, 2020, pp. 5145–5152.
- [23] M. Leucker and C. Schallhart, "A brief account of runtime verification," *J. Log. Algebraic Methods Program.*, vol. 78, no. 5, pp. 293–303, 2009. DOI: 10.1016/J.JLAP.2008.08.004 [Online]. Available: <https://doi.org/10.1016/j.jlap.2008.08.004>
- [24] E. Bartocci et al., "Specification-based monitoring of cyber-physical systems: A survey on theory, tools and applications," in *Lectures on Runtime Verification - Introductory and Advanced Topics*, ser. Lecture Notes in Computer Science, E. Bartocci and Y. Falcone, Eds., vol. 10457, Springer, 2018, pp. 135–175. DOI: 10.1007/978-3-319-75632-5_5 [Online]. Available: https://doi.org/10.1007/978-3-319-75632-5_5
- [25] P. Moosbrugger, K. Y. Rozier, and J. Schumann, "R2U2: monitoring and diagnosis of security threats for unmanned aerial systems," *Formal Methods Syst. Des.*, vol. 51, no. 1, pp. 31–61, 2017. DOI: 10.1007/S10703-017-0275-X [Online]. Available: <https://doi.org/10.1007/s10703-017-0275-x>
- [26] B. D'Angelo et al., "LOLA: runtime monitoring of synchronous systems," in *12th International Symposium on Temporal Representation and Reasoning (TIME 2005), 23-25 June 2005, Burlington, Vermont, USA*, IEEE Computer Society, 2005, pp. 166–174. DOI: 10.1109/TIME.2005.26 [Online]. Available: <https://doi.org/10.1109/TIME.2005.26>
- [27] F. Gorostiaga and C. Sánchez, "Striver: Stream runtime verification for real-time event-streams," in *Runtime Verification - 18th International Conference, RV 2018, Limassol, Cyprus, November 10-13, 2018, Proceedings*, C. Colombo and M. Leucker, Eds., ser. Lecture Notes in Computer Science, vol. 11237, Springer, 2018, pp. 282–298. DOI: 10.1007/978-3-030-03769-7_16 [Online]. Available: https://doi.org/10.1007/978-3-030-03769-7_16

doi.org/10.1007/978-3-030-03769-7%5C_16

- [28] I. Perez, A. E. Goodloe, and F. Dedden, “Runtime verification in real-time with the copilot language: A tutorial,” in *Formal Methods - 26th International Symposium, FM 2024, Milan, Italy, September 9-13, 2024, Proceedings, Part II*, A. Platzer, K. Y. Rozier, M. Pradella, and M. Rossi, Eds., ser. Lecture Notes in Computer Science, vol. 14934, Springer, 2024, pp. 469–491. DOI: 10.1007/978-3-031-71177-0_27
- [29] R. Hippler, H. Kallwies, M. Leucker, M. Montali, C. Sánchez, and S. Winkler, “Symbolic runtime verification for monitoring under uncertainties and assumptions,” *Inf. Softw. Technol.*, vol. 191, p. 108004, 2026. DOI: 10.1016/J.INFSOF.2025.108004
- [30] J. Baumeister, B. Finkbeiner, F. Kohn, and F. Scheerer, “A tutorial on stream-based monitoring,” in *Formal Methods - 26th International Symposium, FM 2024, Milan, Italy, September 9-13, 2024, Proceedings, Part II*, A. Platzer, K. Y. Rozier, M. Pradella, and M. Rossi, Eds., ser. Lecture Notes in Computer Science, vol. 14934, Springer, 2024, pp. 624–648. DOI: 10.1007/978-3-031-71177-0_33 [Online]. Available: https://doi.org/10.1007/978-3-031-71177-0%5C_33
- [31] C. Torens et al., “Certification aspects of runtime assurance for urban air mobility,” in *AIAA SciTech 2024 Forum*, Luftfahrtforschungsprogramm des Bundesministeriums für Wirtschaft und Energie (LuFo), AIAA, Jan. 2024, pp. 1–17, ISBN: 978-162410711-5. DOI: 10.2514/6.2024-1464 [Online]. Available: <https://elib.dlr.de/201362/>
- [32] J. Baumeister, B. Finkbeiner, M. Schwenger, and H. Torfah, “FPGA stream-monitoring of real-time properties,” *ACM Trans. Embed. Comput. Syst.*, vol. 18, no. 5s, 88:1–88:24, 2019. DOI: 10.1145/3358220 [Online]. Available: <https://doi.org/10.1145/3358220>
- [33] J. Baumeister, A. Correnson, B. Finkbeiner, and F. Scheerer, “An intermediate program representation for optimizing stream-based languages,” in *37th International Conference on Computer Aided Verification, CAV 2025, Croatia, July 21-25, 2025*, 2025.
- [34] M. Schwenger, “Monitoring cyber-physical systems: From design to integration,” in *Runtime Verification - 20th International Conference, RV 2020, Los Angeles, CA, USA, October 6-9, 2020, Proceedings*, J. Deshmukh and D. Nickovic, Eds., ser. Lecture Notes in Computer Science, vol. 12399, Springer, 2020, pp. 87–106. DOI: 10.1007/978-3-030-60508-7_5 [Online]. Available: https://doi.org/10.1007/978-3-030-60508-7%5C_5
- [35] M. B. Saltık, L. Özkan, J. H. Ludlage, S. Weiland, and P. M. Van den Hof, “An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects,” *Journal of Process Control*, vol. 61, pp. 77–102, 2018.
- [36] B. T. Lowerre, “The harpy speech recognition system,” Ph.D. dissertation, Carnegie Mellon University, 1976.