Robust Runtime Monitoring with Slack Variables

Bernd Finkbeiner[1], Martin Fränzle[2], Florian Kohn[1], and Paul Kröger[2]

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
{finkbeiner,florian.kohn}@cispa.de
[2] Carl von Ossietzky Universität, Oldenburg, Germany
{martin.fraenzle,paul.kroeger}@uol.de

Abstract. Stream-based monitoring is a runtime assurance technique for cyber-physical systems, where monitors are synthesized from formal specifications. These monitors evaluate system behavior during execution, typically using data from physical sensors. However, sensor measurements are subject to noise and calibration errors, introducing uncertainty in the monitored data.
Robustness is the hyperproperty that requires that small variations in the input - such as those caused by sensor noise - result in only small variations in the monitor's verdicts. In this talk, we present methods for constructing robust monitors using symbolic slack variables and affine arithmetic, and discuss the practical and theoretical challenges involved.

1 Robustness against Measurement Noise

Hyperproperties are properties that relate multiple execution traces of one or more systems. Cyber-physical systems are safety-critical systems that observe the physical environment through sensors. These sensors are subject to calibration errors and random measurement noise. As a result, each sequence of sensor measurements corresponds to a set of possible ground-truth traces that are consistent with the observed data.

Measurement Noise. Measurement noise is a well-known issue in cyber-physical systems [4] and signal processing [11]. ISO Standard 5725 [9] models measurement error as the combination of an unknown but fixed calibration error and a random error for each sample. Sensor specifications typically provide bounds for both error types.

Following [6,5], we define the set of input traces consistent with a sequence of sensor measurements m_s at discrete time points $T \subset \mathbb{N}$. Let $\delta \geq 0$ be the maximum calibration offset and $\epsilon \geq 0$ the maximum random error. The set of consistent input traces is:

$$\mathbb{C}(m_s) \triangleq \{\pi^{in} \in S \mid \exists \Delta \in [-\delta, \delta] : \forall t \in T : \exists \varepsilon \in [-\epsilon, \epsilon] : \pi^{in}(t) + \varepsilon + \Delta = m_S(t).\}$$

Robustness. Robustness is a hyperproperty that requires small differences in input traces to result in small differences in output traces. It has, for example, been studied in the context of real-valued signals [10] and timed I/O systems [8].

In the context of safety-critical systems, runtime monitors typically produce Boolean verdicts to indicate system health. For clarity, we define the monitor's verdict to be true if at least one ground-truth trace that is consistent with the observed measurements satisfies the monitoring condition. However, Boolean verdicts do not align well with distance-based robustness definitions. Instead, we define robustness against measurement noise as follows: For any ground-truth trace that violates the monitoring condition, the monitor must return false for any measurement sequence that is consistent with that trace. Intuitively, this states that for a fixed, unsatisfying ground truth trace, the monitor is robust against any admissible measurement noise imposed onto the trace.

The challenge in constructing such monitors lies in their real-time and memory constraints.

2 Constructing Robust Monitors

Interval Methods. A common approach to handle measurement noise in temporal logics is to use robust semantics based on interval arithmetic [3,2,7]. In this method, input measurements are interpreted as intervals centered around the observed values to account for uncertainty. However, interval arithmetic suffers from the aliasing problem. For example, if a measured value i is represented by the interval $[-10, 10]$, then computing $i - i$ yields: $[-10, 10] - [-10, 10] = [-20, 20]$. Instead of the expected result 0, the operation doubles the uncertainty. This over-approximation propagates through computations, resulting in overly pessimistic error bounds. As a consequence, interval-based methods often yield monitors with inaccurate or inconclusive verdicts.

Slack Variables. Instead we propose to track measurement noise throughout monitoring computations through symbolic slack variables bounded by the interval $[-1, 1]$. Input values are represented as an affine form: $i = m_s + 5\varepsilon + 5\Delta$. Here, m_s is the scalar sensor measurement, ε is a slack variable representing per-sample random error, and Δ is a constant slack variable representing the sensor's calibration error. This formulation captures the measurement noise model described above without introducing the aliasing problem of interval arithmetic.

In our approach [6], we incorporated slack variables into the stream-based monitoring language Lola [1]. Consider the following simple Lola example that sums input measurements:

```
1 input i
2
3 constant d: Variable
4 output e: Variable
5
6 output ia := i + 5*e + 5*d
7 output sum := sum.last(or: 0) + ia
```

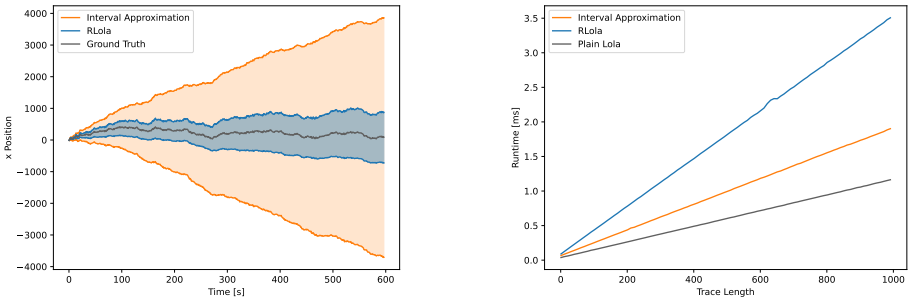
The input stream on line one represents scalar sensor measurements. These measurements are augmented with slack variables in the stream `ia`. The slack variables are declared in lines three and four: `d` is a constant slack variable, and `e` is a stream of slack variables, producing a new symbolic variable for each measurement. The stream `sum` aggregates the affine forms using the `last` operator to reference the previous value, with an initial default.

Generating bounded-memory monitors for such specifications is non-trivial. During execution, slack variables accumulate in memory because their con-

crete values are unknown. For instance, given input measurements i_1, i_2, i_3, the monitor must retain the following affine expression for the `sum` stream: $sum_3 = i_1 + 5e_1 + 5d + i_2 + 5e_2 + 5d + i_3 + 5e_3 + 5d$

3 Experimental Evaluation

To solve this, we identified a fragment of the Lola with slack variables that can indeed be monitored using only bounded memory. To evaluate its practicality, we tested the approach on randomly generated traces.



The left figure compares error margins for a single output stream. The grey line represents the ground truth value. The orange area shows the error bounds computed using interval arithmetic, while the blue area shows the tighter bounds produced by our slack-variable-based method. The right figure shows the runtime overhead of using slack variables. While some performance cost is incurred, it is minor relative to the improvement in accuracy.

4 Talk Outline

The talk will be structured in three parts:

1. **Robustness against Measurement Noise.** We introduce the problem of measurement noise in cyber-physical system monitoring and relate it to the hyperproperty of robustness.
2. **Robust Lola with Slack Variables.** We present the integration of slack variables into Lola, including its benefits and implementation challenges. We explain the simplifications and approximations that are neccesary to achieve constant memory monitors.
3. **Future Directions.** We discuss open questions and ongoing challenges in managing slack variables within runtime monitors such as precise offline monitoring and error estimation.

5 Conclusion

We presented an extension of the Lola monitoring framework to support robustness against measurement noise through the use of slack variables. These symbolic variables allow uncertainty from sensor measurements to be tracked throughout computation. While the approach can lead to unbounded memory usage, we identified a fragment of the logic for which constant memory monitors can be generated without loss of precision. Experimental results show that this method significantly improves accuracy with minimal runtime overhead.

References

1. D'Angelo, B., Sankaranarayanan, S., Sánchez, C., Robinson, W., Finkbeiner, B., Sipma, H.B., Mehrotra, S., Manna, Z.: LOLA: runtime monitoring of synchronous systems. In: 12th International Symposium on Temporal Representation and Reasoning (TIME 2005), 23-25 June 2005, Burlington, Vermont, USA. pp. 166–174. IEEE Computer Society (2005). https://doi.org/10.1109/TIME.2005.26
2. Donzé, A., Ferrère, T., Maler, O.: Efficient robust monitoring for STL. In: Sharygina, N., Veith, H. (eds.) Computer Aided Verification - 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings. Lecture Notes in Computer Science, vol. 8044, pp. 264–279. Springer (2013). https://doi.org/10.1007/978-3-642-39799-8_19
3. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) 8th International Conference on Formal Modeling and Analysis of Timed Systems, FORMATS 2010, Klosterneuburg, Austria, September 8-10, 2010. Proceedings. Lecture Notes in Computer Science, vol. 6246, pp. 92–106. Springer (2010). https://doi.org/10.1007/978-3-642-15297-9_9
4. Duo, W., Zhou, M., Abusorrah, A.: A survey of cyber attacks on cyber physical systems: Recent advances and challenges. IEEE CAA J. Autom. Sinica **9**(5), 784–800 (2022). https://doi.org/10.1109/JAS.2022.105548, https://doi.org/10.1109/JAS.2022.105548
5. Finkbeiner, B., Fränzle, M., Kohn, F., Kröger, P.: A truly robust signal temporal logic: Monitoring safety properties of interacting cyber-physical systems under uncertain observation. Algorithms **15**(4), 126 (2022). https://doi.org/10.3390/A15040126
6. Finkbeiner, B., Fränzle, M., Kohn, F., Kröger, P.: Stream-based monitoring under measurement noise. In: Ábrahám, E., Abbas, H. (eds.) Runtime Verification - 24th International Conference, RV 2024, Istanbul, Turkey, October 15-17, 2024, Proceedings. Lecture Notes in Computer Science, vol. 15191, pp. 22–39. Springer (2024), https://doi.org/10.1007/978-3-031-74234-7_2
7. Fränzle, M., Hansen, M.R.: A robust interpretation of duration calculus. In: Hung, D.V., Wirsing, M. (eds.) Theoretical Aspects of Computing - ICTAC 2005, Second International Colloquium, Hanoi, Vietnam, October 17-21, 2005, Proceedings. Lecture Notes in Computer Science, vol. 3722, pp. 257–271. Springer (2005). https://doi.org/10.1007/11560647_17
8. Henzinger, T.A., Otop, J., Samanta, R.: Lipschitz robustness of timed I/O systems. In: Jobstmann, B., Leino, K.R.M. (eds.) Verification, Model Checking, and Abstract Interpretation - 17th International Conference, VMCAI 2016, St. Petersburg, FL, USA, January 17-19, 2016. Proceedings. Lecture Notes in Computer Science, vol. 9583, pp. 250–267. Springer (2016), https://doi.org/10.1007/978-3-662-49122-5_12
9. ISO: ISO/IEC 5725:2023: Accuracy (trueness and precision) of measurement methods and results - Part 1: General principles and definitions. International Organization for Standardization, Geneva, Switzerland (July 2023)
10. Nguyen, L.V., Kapinski, J., Jin, X., Deshmukh, J.V., Johnson, T.T.: Hyperproperties of real-valued signals. In: Talpin, J., Derler, P., Schneider, K. (eds.) Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design, MEMOCODE 2017, Vienna, Austria, September 29 - October 02, 2017. pp. 104–113. ACM (2017), https://doi.org/10.1145/3127041.3127058

11. Tuzlukov, V.: Signal processing noise. CRC Press (2018)