The Eleventh International Conference on
Engineering Computational Technology
23–25 August, 2022 | Montpellier, France

Proceedings of the Eleventh International Conference on
Engineering Computational Technology
Edited by B.H.V. Topping and P. Iványi
Civil-Comp Conferences, Volume 2, Paper 6.7
Civil-Comp Press, Edinburgh, United Kingdom, 2022, doi: 10.4203/ccc.2.6.7

Runtime Monitoring for Unmanned Aerospace Systems with Neural Network Components

Y. He[1] and J. Schumann[2]

[1]NASA Ames Research Center, USA
[2]KBR/Wyle, NASA ARC, USA

Abstract

AI components (e.g., Deep Neural Networks) are increasingly used in unmanned Aerospace systems for safety-relevant applications. Rigorous Verification and Validation methods for such components are still in their infancy and thus, monitoring of the AI's behavior during runtime is essential. In this paper, we will present a runtime-monitoring architecture, which combines the advanced statistical analysis framework SYSAI (System Analysis using Statistical AI) with temporal and probabilistic runtime monitoring carried out by R2U2 (Realizable, Responsive, and Unobtrusive Unit). Learned statistical models of complex systems with AI components are produced by the SYSAI framework and provide detailed information to enable the R2U2 runtime monitor to efficiently perform advanced safety and performance checks in nominal and off-nominal conditions. We will present initial results of our tool set and architecture on a case study, a DNN-based autonomous centerline tracking system (ACT).

Keywords: deep neural network, runtime monitoring, statistical analysis

1 Introduction

AI components such as Deep Neural Networks (DNNs) have found their way into many complex aerospace systems. In particular in autonomous aircraft, such applications are safety-critical, and failures might led to loss of vehicle and mission, or even to loss of life. Yet, such systems need to operate properly under a wide variety

of different operational and environmental conditions, as well as under failures. Rigorous Verification and Validation (V&V) is mandatory, yet V&V techniques for DNNs are still in their infancy and can often only provide relatively weak guarantees.

In this paper, we will present a runtime-monitoring approach, which combines advanced statistical DNN analysis techniques with temporal, model-based, and probabilistic runtime monitoring. For the statistical analysis and model-building, we use our statistical framework SYSAI (System Analysis using Statistical AI) [3,2,5,4] our flexible statistical learning framework for V&V and analysis of complex and high-dimensional cyber-physical systems with AI components. The R2U2 (Realizable, Responsive, and Unobtrusive Unit) [8,7,1] is an on-board monitoring system to continuously monitor system and safety properties of a cyber-physical system or its components using Temporal Logics and Bayesian reasoning.

Learned statistical models of the complex system and its AI components, which are produced by SYSAI during V&V provide detailed information to enable the R2U2 runtime monitor to efficiently perform advanced safety and performance checks in nominal and off-nominal conditions and to perform model-based prognostics. R2U2 results can, for example, be used to switch between different AI components for better performance. Within an assurance architecture, R2U2 can function as a safety monitor, which can initiate a mitigation from a poorly performing or unsafe DNN toward a more safely operating component.

2 Methods

Our architecture for monitoring of the complex AI component, using the R2U2 runtime monitor. Important information for the R2U2 properties are produced by SYSAI during statistical analysis of the system at design and V&V time. For a synergistic combination of both tools, we propose a draft of a process as shown in the figure.

Based upon detailed system requirements, the system with AI components is developed and the DNN(s) are trained using training data. At this V&V stage, SYSAI can be used for analysis of training data, characterization of safety regions in a high-dimensional state space, as well as analysis of the system's behaviour under failures [5,4]. Analysis results also provide feedback to the designer. The SYSAI framework and the underlying models and algorithms are described in detail in [3]. SYSAI has been used for the analysis of several complex and safety-critical aerospace systems [2,5,4].

After system development and testing, the system is being deployed. At this stage, our R2U2 runtime monitoring architecture (lower half of the figure; inspired by [6], Fig. 1) is active while the system is in operation.

The Deep Neural Network (grey box) receives inputs from the AC and processes them. The results (e.g., estimated position of the AC on the runway) are then passed through the RTA switch back to the system, e.g., the aircraft controller. In nominal operations, the RTA switch is set to route the signals from the AI component to the system. In parallel, R2U2 receives the system signals, as well as signals from the AI

component. Without affecting the overall system behaviour (unobtrusiveness), a multitude of temporal and probabilistic properties can be checked in real time.

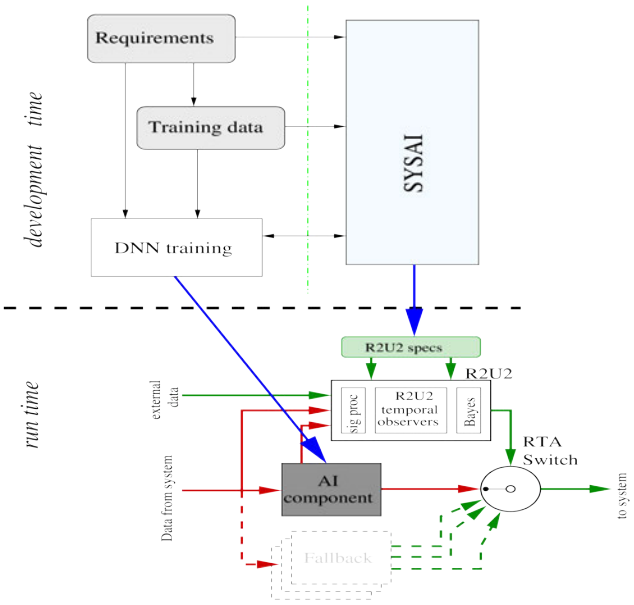

Figure 1: Tool chain and process for the combination of SYSAI and R2U2.

The statistical models and results, produced by SYSAI, are used to define and customize properties to be checked by R2U2 (vertical red arrow). The R2U2 output is used to control the RTA switch: in case, R2U2 detects a violation of important safety/performance properties, the RTA switch can be turned to use a fallback component instead of the AI component to retain system safety and (at least limited) performance. Multiple fallback methods might be provided, ranging from algorithmic components (e.g., simple dead reckoning) to entering a fail-safe mode, stopping the AC, and contact a remote operator.

The information passed can range from simple threshold parameters, whose values have been determined by SYSAI's safety-boundary characterization. In that case, SYSAI's advanced capabilities for the geometric characterization of safety boundaries can be used for setting up efficient R2U2 property checking.

3 Results

For our experiments, we used an Autonomous Centerline Tracking (ACT) system. Here, a forward-facing camera mounted on the wing provides input to a DNN. Its outputs, the estimated distance from the centerline *CTE* and the heading (*he*) relative to the runway are then fed into a standard fixed-gain controller that operates the front

wheel of the aircraft as it rolls down the runway at a slow speed. Experiments were carried out using the X-Plane Flight Simulator (`www.xplane.com`).

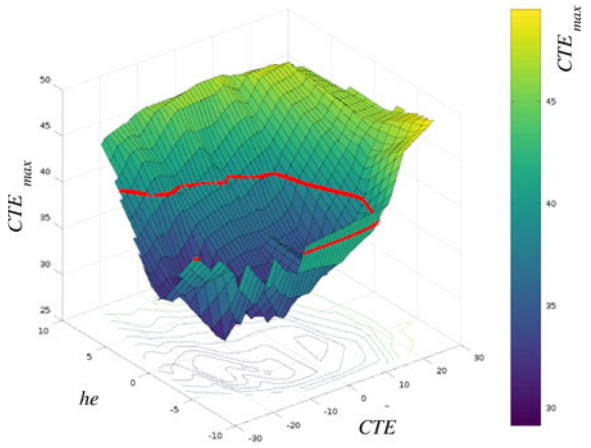

Figure 2: Safety-envelope: surface shows estimated maximal CTE value during a run over initial position CTE and heading of the aircraft he. The safety envelope at a given threshold of 40ft is shown as a red line.

The DNN is a multi-layer feed-forward network with ReLU nodes and implemented in TensorFlow. For the setup of the R2U2 specifications for nominal operation, it is, among others, important to establish reasonable safety thresholds for the neural network outputs using SYSAI.

Figure 2 shows how the safety envelope for ACT under different initial conditions *CTE* and *he* develop. For a small threshold, only runs with initial values close to *CTE*=0 and *he*=0 are successful, i.e., that our safety conditions is never violated during a run. This safety envelope becomes larger as the value for Θ increases. The red line in Figure 2 shows its boundary for Θ=40*ft*. SYSAI has been used to effectively create a model of this surface; a geometric characterization of the boundary can be obtained from SYSAI.

The information obtained during the SYSAI analysis is then used to set up the R2U2 properties and monitors. We can distinguish between three different categories of R2U2 properties: (a) universal properties, (b) temporal properties, and (c) probabilistic properties. Universal properties are supposed to be valid throughout the entire operation and are necessary to define many safety properties. For example, makes sure that the speed of the aircraft is always limited and that the aircraft never rolls backward. Within the R2U2 monitor, such properties are usually linked to conditions or system modes. In our example, this condition is only to be checked if

the ATC system is on and the AC in taxi mode. This will yield the temporal formula

```
H(ATC_on & AC_mode = taxi -> v_w < 5m/s & 0 <= v_w)   (1)
```

Temporal R2U2 properties can be used to specify include (a) overall performance properties, e.g., the end of the runway should be reached within 4-5 minutes, (b) filtering of transients. For example, the outputs of the DNN should always lie in a certain range, e.g., $he \in [-10,10]$. However, transients yielding values outside that range should be tolerated if they are short enough, for example, less than 2 seconds, (c) limiting the number of occurrences of events. For example, it can be specified that no more than 3 transients occur within a period of 20 seconds. Such a property can also be seen as a discrete form of specifying error rates.

Such properties can be defined using the original signals (e.g., *cte*, *he*) or results of signal processing. In our architecture, we can use following algorithms and methods to process the input signals before discretizing them into a Boolean value and handing them to the Temporal logic engine. Signal rates, as approximation of signal derivatives are used to help monitor the system dynamics and integration of signals over time is used to check, for example, for changes of biases over time. Fast Fourier Transformation of signals are helpful in detection of oscillations. Such effects, like pilot-induced oscillations can lead to dangerous situations that need to be avoided. Finally, Kalman filtering can be used for sensor fusion or to check the behaviour of a signal against a given dynamical model. In this case study, Kalman filters have not been used.

4 Conclusions and Contributions

In this paper, we have presented an advanced architecture to monitor the safety and performance of a complex AI component (e.g., a DNN) within an aerospace system. Inspired by the ASTM runtime assurance architecture, we are using the R2U2 runtime monitoring system to dynamically check numerous properties, using temporal logic observers, Bayesian reasoners, and signal processing.

Our SYSAI statistical analysis framework can provide models, parameters, and other information to the R2U2. This information is used to define, formulate, and customize complex, yet justified properties that go ways beyond traditional range and rate checking monitors.

The information passed can range from simple threshold parameters, whose values have been determined by SYSAI's safety-boundary characterization. In that case, SYSAI's advanced capabilities for the geometric characterization of safety boundaries can be used for setting up efficient R2U2 property checking.

During system deployment, the R2U2 continually monitors the system, and its output is used to control the RTA switch: in case, R2U2 detects a violation of important safety/performance properties, the RTA switch can be turned to use a fallback component instead of the AI component to retain system safety and (at least limited) performance.

Future work will include the use of dynamic statistical reasoners and prognostic engines to extend this architecture into a fully statistical monitoring system, which can reason and decide with probabilities and confidence levels—a prerequisite for

monitoring systems like Deep Neural Networks. We are also planning to work toward the use of this architecture and process in certification and risk management.

References

[1] J. Geist, K.Y. Rozier, J. Schumann. Runtime Observer Pairs and Bayesian Network Reasoners On-board FPGAs: Flight-Certifiable System Health Management for Embedded Systems. In *Proceedings Runtime Verification (RV14)*, pages 215–230. Springer, 2014. doi: 10.1007/978-3-319-11164-3_18

[2] Y. He. Online detection and modeling of safety boundaries for aerospace applications using active learning and Bayesian statistics. In *2015 International Joint Conference on Neural Networks, IJCNN 2015, Killarney, Ireland, July 12-17, 2015*, pages 1–8. IEEE, 2015. doi: 10.1109/IJCNN.2015.7280595

[3] Y. He, J. Schumann. A framework for the analysis of deep neural networks in aerospace applications using Bayesian Statistics. In *Proc Int Joint Conf on Neural Networks IJCNN,* 2020. doi: 10.1109/IJCNN48605.2020.9207228

[4] Y. He, H. Yu, G. Brat, M. Davies. Statistical learning framework for safety and failure analysis of a DNN-based autonomous aircraft system. In *Proc. International Conference on Machine Learning Applications (ICMLA)*. IEEE, 2021.

[5] Y. He, H. Yu, G. Brat, M. Davies. System and safety analysis for autonomous center line tracking with SYSAI. In *Proc AIAA SciTech Forum 2022.* 2022. doi: 10.2514/6.2022-2399

[6] P. Nagarajan, S.K. Kannan, C. Torens, M.E. Vukas, G.F. Wilber. *ASTM F3269 - An Industry Standard on Run Time Assurance for Aircraft Systems*. In *Proc AIAA SciTech Forum.* 2021. doi: 10.2514/6.2021-0525

[7] T. Reinbacher, K.Y. Rozier, J. Schumann. Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems. In *Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS*, volume 8413 of *LNCS*, pages 357–372. Springer, 2014. doi: 10.1007/978-3-642-54862-8_24

[8] K.Y. Rozier, J. Schumann. R2U2: tool overview. In *Proceedings RV-CuBES 2017*, pages 138–156, In *Proc. RV-CuBES,* 2017.