Runtime Monitoring with R2U2 for Aircraft Systems with Neural Networks

Johann Schumann†

†KBR/Wyle, NASA Ames Research Center

Johann.M.Schumann@nasa.gov
$R_1: \text{BeginPresentation} \rightarrow \diamond [45\text{min}, 48\text{min}] "\text{THE END}"$
Runtime Monitoring

\[ R_1 : \text{BeginPresentation} \rightarrow \diamond [45\text{min}, 48\text{min}]" \text{THE END}" \]

R2U2

- **RESPONSIVENESS**: respond in “real time”
- **REALIZABILITY**: plug-and-play
- **UNOBTUSIVENESS**: do not “mess up” the flight software
- **UNIT**

R2U2 is a run-time monitoring and V&V tool that combines *Metric Temporal Logic* observers, *Bayesian Network* reasoners, and *model-based prognostics*. 
On a flight from Boston, Mass. to SFO...

Software Error? Sensor Failure? Operational Error?
Pilot: “The sun is on the left hand side, so we are OK”
R2U2 is a run-time monitoring and V&V tool that combines Metric Temporal Logic observers, Bayesian Network reasoners, and model-based prognostics.
Why all the R2U2 Ingredients?

Different health management approaches focus on specific properties

- Boolean Logic (assertions)
  - value and rate checkers
  - thresholding
Why all the R2U2 Ingredients?

Different health management approaches focus on specific properties

- Boolean Logic (assertions)
  - value and rate checkers
  - thresholding
- Model-Based Diagnostics and Monitoring
  - TEAMS
  - Prognostics
Different health management approaches focus on specific properties

- Boolean Logic (assertions)
  - value and rate checkers
  - thresholding
- Model-Based Diagnostics and Monitoring
  - TEAMS
  - Prognostics
- Temporal
  - TFPG, checkers with delays
Why all the R2U2 Ingredients?

Different health management approaches focus on specific properties

- Boolean Logic (assertions)
  - value and rate checkers
  - thresholding
- Model-Based Diagnostics and Monitoring
  - TEAMS
  - Prognostics
- Temporal
  - TFPG, checkers with delays
- Probabilistic
  - Bayesian Networks
Why all the R2U2 Ingredients?

Different health management approaches focus on specific properties

- Boolean Logic (assertions)
  - value and rate checkers
  - thresholding
- Model-Based Diagnostics and Monitoring
  - TEAMS
  - Prognostics
- Temporal
  - TFPG, checkers with delays
- Probabilistic
  - Bayesian Networks

Our R2U2 framework provides powerful mechanisms to enable temporal, probabilistic diagnostic models integrating advanced prognostics models.
R2U2 Architecture

- R2U2 as software node for ROS
- R2U2 as software app for NASA cFS/cFE Core Flight System
- R2U2 as Simulink Block
- R2U2 as Field Programmable Gate Array (FPGA) configuration
Future Time Metric Temporal Logic (MTL) reasons about bounded timelines:

- Atomic propositions: $p, q$
- Operators: $\neg, \land, \lor, \rightarrow, \leftrightarrow, \circ, \Box_I, \Diamond_I, U_I, R_I$

### Examples of MTL operators:

**$\Box_{[2,6]}p$**

- **Always** $[2,6]$

- Transition diagram:

  - States: 0, 1, 2, 3, 4, 5, 6, 7, 8
  - $p$ at states 1, 2, 4, 5, 6, 8

**$\Diamond_{[2,6]}p$**

- **Eventually** $[0,7]$

- Transition diagram:

  - States: 0, 1, 2, 3, 4, 5, 6, 7, 8
  - $p$ at state 4

**$pUq$**

- **Until**

- Transition diagram:

  - States: 0, 1, 2, 3, 4, 5, 6, 7, 8
  - $p$ at states 0, 1, 2, 3, 4
  - $q$ at state 5

**Past Time**: similar with temporal operators Historically, Once, Since
Observer Pairs

For each future time MTL formula, we create two observers:

- **asynchronous** observers return \( \{ T, F \} \)
  - results are not instantaneous
  - observer has considerable complexity and needs local memory

- **synchronous** observer returns \( \{ T, F, \text{maybe} \} \) at each timestamp
  - results immediately available
  - observer has low complexity
  - three-valued logic can be useful for reasoning

We do not translate MTL formulas into finite state machines; rather we use synchronisation queues [TACAS’2014]
Example: Safety Rule

After receiving a command for takeoff, the UAS shall reach an altitude of 600ft within 60 seconds.
Example: Safety Rule

After receiving a command for takeoff, the UAS shall reach an altitude of 600ft within 60 seconds.

\( \square((cmd == \text{takeoff}) \rightarrow \lozenge_{[0,60s]}(alt \geq 600 \text{ ft})) \)
Example: Safety Rule

After receiving a command for takeoff, the UAS shall reach an altitude of 600ft within 60 seconds.

\[
\square((\text{cmd} == \text{takeoff}) \rightarrow \Diamond_{[0,60s]}(alt \geq 600 \text{ ft}))
\]
Example: Safety Rule

After receiving a command for takeoff, the UAS shall reach an altitude of 600ft within 60 seconds.

\[ \square((\text{cmd} == \text{takeoff}) \rightarrow \Diamond_{[0,60s]}(alt \geq 600 \text{ ft})) \]
Example: Safety Rule

After receiving a command for takeoff, the UAS shall reach an altitude of 600ft within 60 seconds.

$$\Box((\text{cmd} == \text{takeoff}) \rightarrow \Diamond_{[0, 60s]}(alt \geq 600 \text{ ft}))$$
Is there enough battery to fly over the hill? Flight time is 10 minutes

\[
\text{start-climb} \rightarrow \square_{[10\, \text{min}]}\text{battery-OK}
\]
Ingredient III: Model-based Prognostics

Is there enough battery to fly over the hill? Flight time is 10 minutes

\[
\text{start-climb} \rightarrow \square_{[10\,\text{min}]} \text{battery-OK}
\]

- **Not Good:** valuation only available after 10 minutes
Ingredient III: Model-based Prognostics

Is there enough battery to fly over the hill? Flight time is 10 minutes

\[ \text{start-climb} \rightarrow \Box_{[10\text{min}]} \text{battery-OK} \]

- **Not Good**: valuation only available after 10 minutes
- **Not much better**: \textbf{Maybe} Valuation available now
Ingredient III: Model-based Prognostics

Is there enough battery to fly over the hill? Flight time is 10 minutes

\[
\text{start-climb} \rightarrow \square_{[10\text{min}]} \text{battery-OK}
\]

- **Not Good**: valuation only available after 10 minutes
- **Not much better**: Maybe Valuation available now

\[
\text{start-climb} \rightarrow (RUL > 10\text{min})
\]

- **Good**: model-based prognostics. Result available now

R2U2 uses electro-chemical model for LiPo batteries and a UKF-based algorithm
Bayesian Reasoning

- Bayesian models for diagnostic reasoning and health management are well established
- Our Bayesian Networks (BNs) contain
  - health nodes $H$ ("output")
  - behavior nodes
  - observable sensor nodes $S$

R2U2 provides efficient constant-footprint implementation and tight integration with temporal properties
Bayesian models for diagnostic reasoning and health management are well established.

Our Bayesian Networks (BNs) contain:
- health nodes $H$ ("output")
- behavior nodes
- observable sensor nodes $S$

R2U2 provides efficient constant-footprint implementation and tight integration with temporal properties.
Bayesian Reasoning

- Bayesian models for diagnostic reasoning and health management are well established
- Our Bayesian Networks (BNs) contain
  - health nodes $H$ ("output")
  - behavior nodes
  - observable sensor nodes $S$

R2U2 provides efficient constant-footprint implementation and tight integration with temporal properties
R2U2 Capabilities and Applications

- Signal Processing
- Past Time Temporal Logic
- Future Time Temporal Logic
- Bayesian Reasoning
- Prognostics

- Safety monitoring
- Performance monitoring
- Security monitoring
- Failure diagnosis
- Prognostics
- Autonomous decision making

Swift, ARC
Dragoneye, ARC
Edge 540, LaRC
AOS, ARC
R2U2 Capabilities and Applications

- Signal Processing
- Past Time Temporal Logic
- Future Time Temporal Logic
- Bayesian Reasoning
- Prognostics

- Safety monitoring
- Performance monitoring
- Security monitoring
- Failure diagnosis
- Prognostics
- Autonomous decision making
R2U2 Capabilities and Applications

- Signal Processing
- Past Time Temporal Logic
- Future Time Temporal Logic
- Bayesian Reasoning
- Prognostics
- safety monitoring
- performance monitoring
- security monitoring
- failure diagnosis
- prognostics
- autonomous decision making

Swift, ARC
Dragoneye, ARC
Edge 540, LaRC
AOS, ARC
R2U2 for Monitoring of Neural Networks

Case Study: Autonomous Center Line Tracking

- NN can produce noisy or arbitrary result
- NN can produce wrong result
- NN is inherently probabilistic
- NN lacks explainability and interpretability

Runtime monitoring on NN level and system level is necessary
R2U2 support

R2U2 can

- analyze and incorporate signals from NN-specific sensors (e.g., from Prophecy runtime)
- use results of Bayesian NN calculations for safety reasoning (e.g., confidence of outputs or weights)
- detect and diagnose NN related failures (e.g., camera sensor problems) using model-based diagnostics
- merge system sensor signals with NN signals (sensor fusion)
- monitor temporal behavior on system level and on component level
- monitor behavior for onboard learning applications (N/A here)
R2U2 on ACT

R2U2 as a ROS node monitoring the behavior of ACT

video/video_1.mp4
Toward a Runtime Assurance Architecture (Future Work)

R2U2 can be Safety Monitor in an ASTM F3269 style system

underlying figure from P. Nagarajan, S. Kannan, et.al. SciTech 2021
R2U2 is a Responsive, Realizable, and Unobtrusive Unit for system and component runtime monitoring.

R2U2 combines Metric Temporal Logic observers, Bayesian reasoners, and model-based prognostics.

R2U2 implementations for ROS, Simulink, cFS, and FPGA.

R2U2 applicable for safety monitoring, performance monitoring, security monitoring, failure diagnosis, prognostics, and autonomous decision making.

R2U2 useful for monitoring of systems with Neural Networks.


Team

- Johannes Geist
- Timmy Mbaya
- Thomas Reinbacher
- Julian Rhein
- Kristin Rozier
- Johann Schumann

PoC: johann.m.schumann@nasa.gov or johann.schumann@us.kbr.com
Thank You!