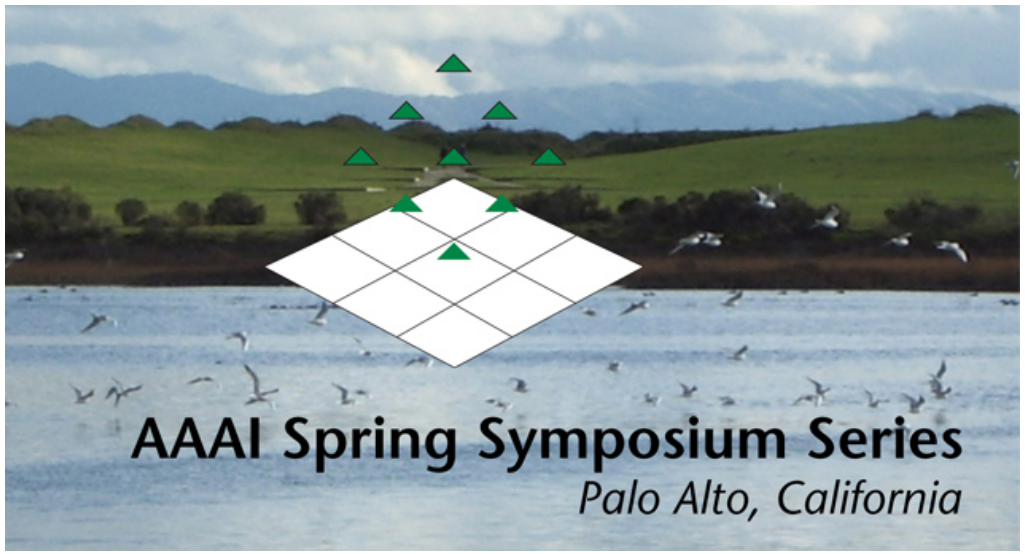
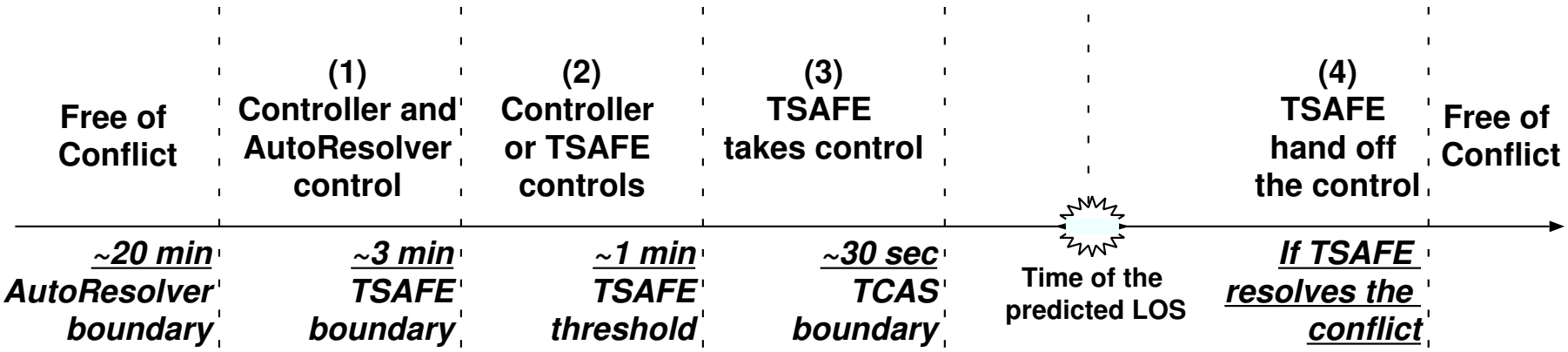
On the Effectiveness of Mission-time Linear Temporal Logic (MLTL) in AI Applications



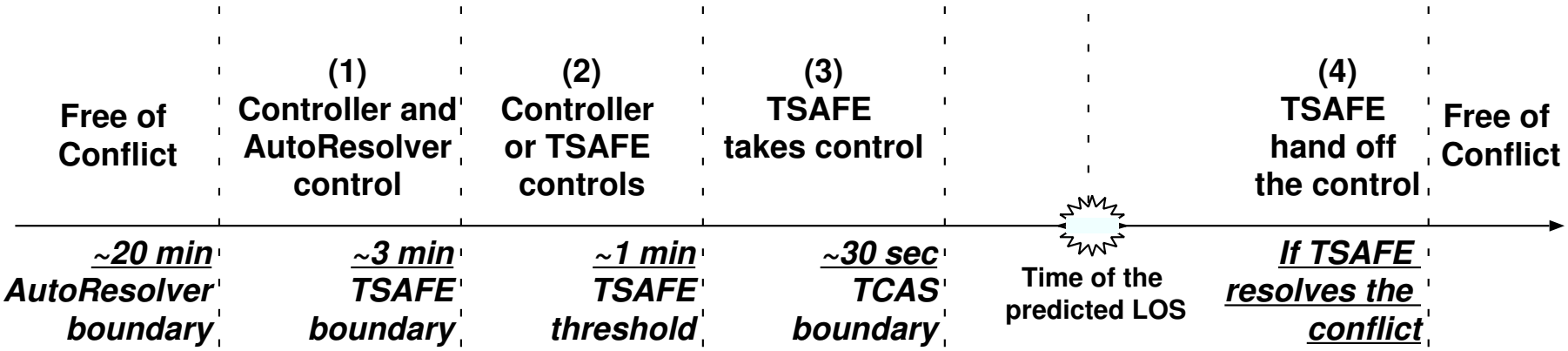
Kristin Yvonne Rozier
Iowa State University of Science and Technology
March 28, 2023

AAC Operational Concept[1]



[1] H Erzberger, K Heere. "Algorithm and operational concept for resolving short-range conflicts." Proc. IMechE G J. Aerosp. Eng. 224 (2) (2010) 225–243.

AAC Operational Concept[2]



LTL Model Checking triggered system design changes[1]

[1] Y. Zhao and K.Y. Rozier. "Formal Specification and Verification of a Coordination Protocol for an Automated Air Traffic Control System." SCP Journal, vol-96, no-3, pg 337-353, 2014.

[2] H Erzberger, K Heere. "Algorithm and operational concept for resolving short-range conflicts." Proc. IMechE G J. Aerosp. Eng. 224 (2) (2010) 225–243.

Is LTL All We Need?

ATC never turns off . . .

Is LTL All We Need?

ATC never turns off . . .

LTL intuitively describes it

Is LTL All We Need?

ATC never turns off . . .

LTL intuitively describes it

Aircraft have finite missions; ATC has finite modes. . .

Is LTL All We Need?

ATC never turns off . . .

LTL intuitively describes it

Aircraft have finite missions; ATC has finite modes. . .

LTLf?

Is LTL All We Need?

ATC never turns off . . .

LTL intuitively describes it

Aircraft have finite missions; ATC has finite modes. . .

LTLf?

But there are numerical bounds on the timelines. . .

Is LTL All We Need?

ATC never turns off . . .
LTL intuitively describes it

Aircraft have finite missions; ATC has finite modes. . .
LTLf?

But there are numerical bounds on the timelines. . .
MTL? STL?

MTL: Many Variations[3]

- **Semantics:** continuous vs pointwise
- **Traces:** finite vs infinite
- **Intervals:**
  - infinite vs finite vs bounded (specific bounds)
  - open, closed, half-open
  - punctual (singleton allowed) or not
  - start with 0 or end with ∞: MTL_0; $MTL_{0,\infty}$
- **Interval types:** integer vs real numbers

[3] Ouaknine & Worrell. "Some Recent Results in Metric Temporal Logic." FORMATS 2008.

STL: Made for Describing CPS[5]

STL adds an **analog layer** to MTL,
reasons over **real-valued predicates** with **real-time intervals**[4]

STL Semantics: the satisfaction of an STL formula φ by a signal $\mathrm{x} = (x_1, \ldots, x_n)$ at time t is

$$(\mathrm{x}, t) \models \mu \Leftrightarrow f(x_1[t], \ldots, x_n[t]) > 0$$
$$(\mathrm{x}, t) \models \varphi \wedge \psi \Leftrightarrow (\mathrm{x}, t) \models \varphi \wedge (\mathrm{x}, t) \models \psi$$
$$(\mathrm{x}, t) \models \neg\varphi \Leftrightarrow \neg((x, t) \models \varphi)$$
$$(\mathrm{x}, t) \models \varphi \mathcal{U}_{[a,b]} \psi \Leftrightarrow \exists t' \in [t + a,\ t + b] \text{ such that } (\mathrm{x},\ t') \models \psi$$
$$\wedge \forall t'' \in [t,\ t'], (\mathrm{x},\ t'') \models \varphi$$

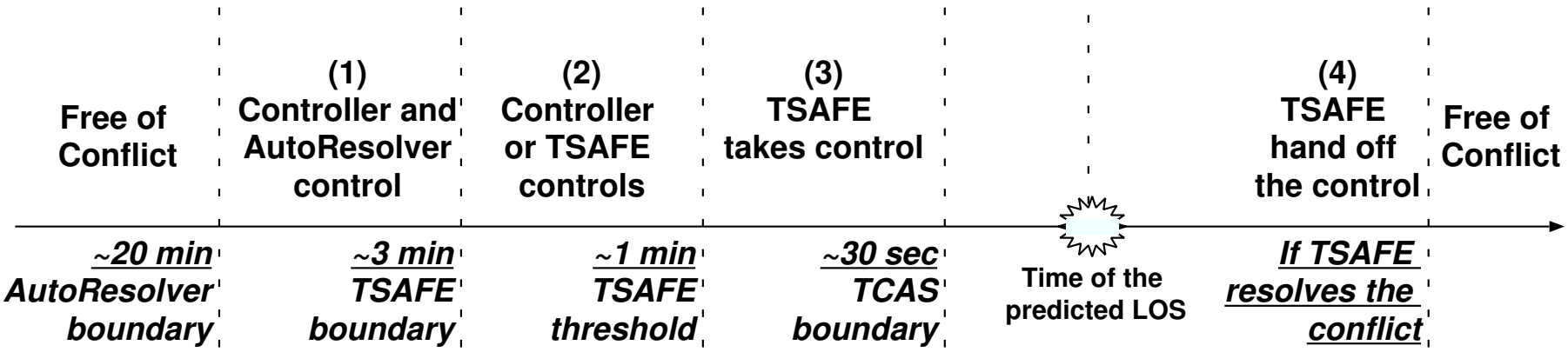
[4] Donzè. "On Signal Temporal Logic." RV, 2013.

[5] Maler & Nickovic. "Monitoring Temporal Properties of Continuous Signals." FORMATS 2004.

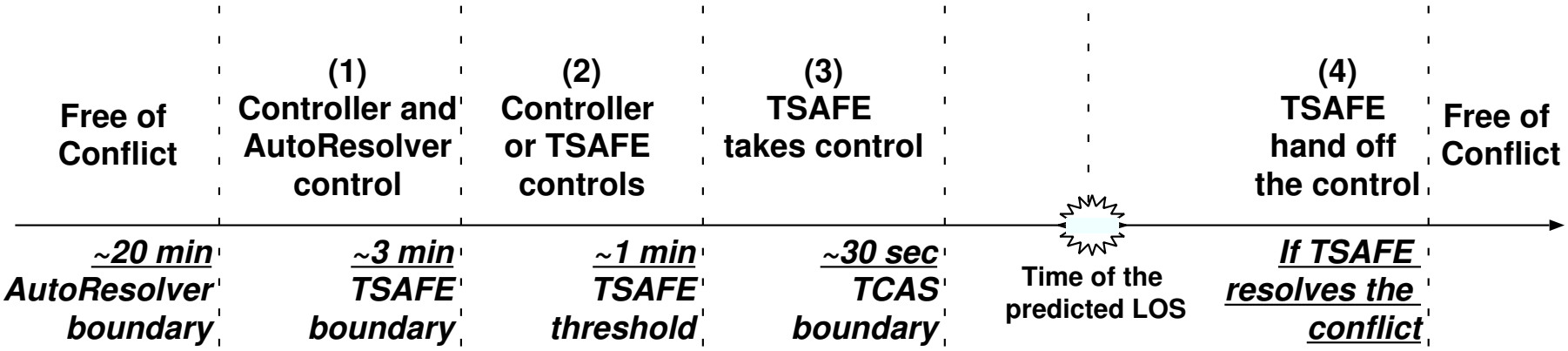
STL: With Great Power Comes...

- **Complexity**: MTL satisfiability and model checking are **undecidable**; STL?
- **Confusion**: LTL is hard to write correctly, validate; STL?
- **Precision**: STL needs details not present in the system

AAC Operational Concept



AAC Operational Concept



Times are discrete, rough estimates. . .

STL: Not a Good Fit

- Humans have to write the specifications; **writing formal properties is hard.**[6]
- Humans have to **validate** the specifications; need to check satisfiability efficiently
- Certification requires **explainability**
- Many domains (ISS) require **adaptabiity**

[6] K. Y. Rozier. "Specification: The Biggest Bottleneck in Formal Methods and Autonomy." VSTTE Keynote, 2016.

STL continued . . .

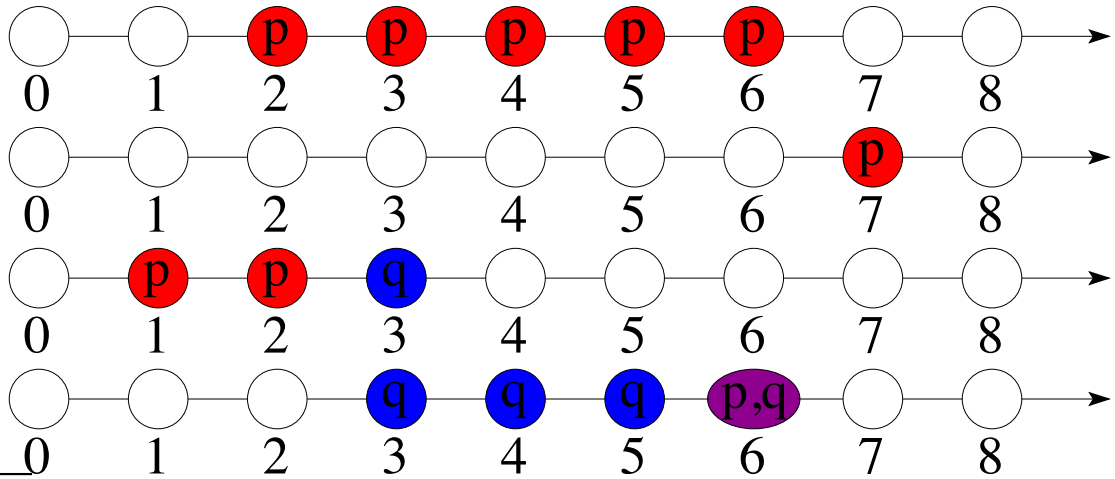
- Requires or assumes details not present in the system
- Continuous-time reasoning might not match up with discrete systems
- "Preciseness" $\rightarrow \uparrow$ complexity; won't fit in tight spaces
- Formulas are too specific; not re-useable
- Hard to validate (hard for humans to understand)
- Specifications not robust to realistic system changes

MLTL: A Good Specification Language[7]

Mission-Time Temporal Logic (MLTL) reasons about *integer-bounded* timelines:

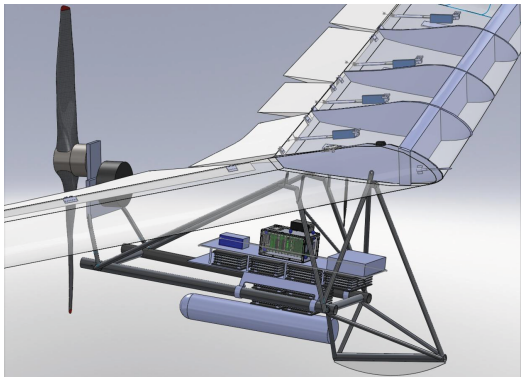
- finite set of atomic propositions {p q}
- Boolean connectives: $\neg$, $\wedge$, $\vee$, and $\rightarrow$
- temporal connectives *with time bounds*:

Symbol	Operator	Timeline
$\Box_{[2,6]} p$	$\text{ALWAYS}_{[2,6]}$	
$\Diamond_{[0,7]} p$	$\text{EVENTUALLY}_{[0,7]}$	
$p \mathcal{U}_{[1,5]} q$	$\text{UNTIL}_{[1,5]}$	
$p \mathcal{R}_{[3,8]} q$	$\text{RELEASE}_{[3,8]}$	



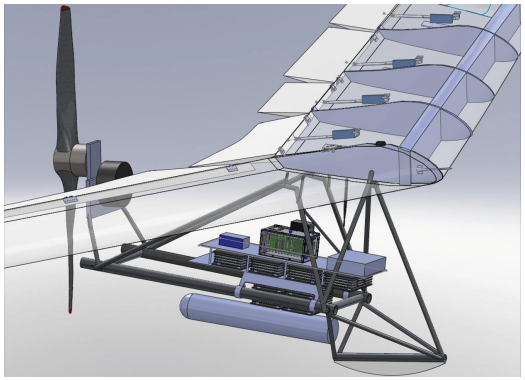
[7] T. Reinbacher, K.Y. Rozier, J. Schumann. "Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems." TACAS 2014.

Runtime Monitoring for the Swift UAS



After receiving a command (cmd) for takeoff, the Swift UAS must reach an altitude of 600ft within 40 seconds.

Runtime Monitoring for the Swift UAS

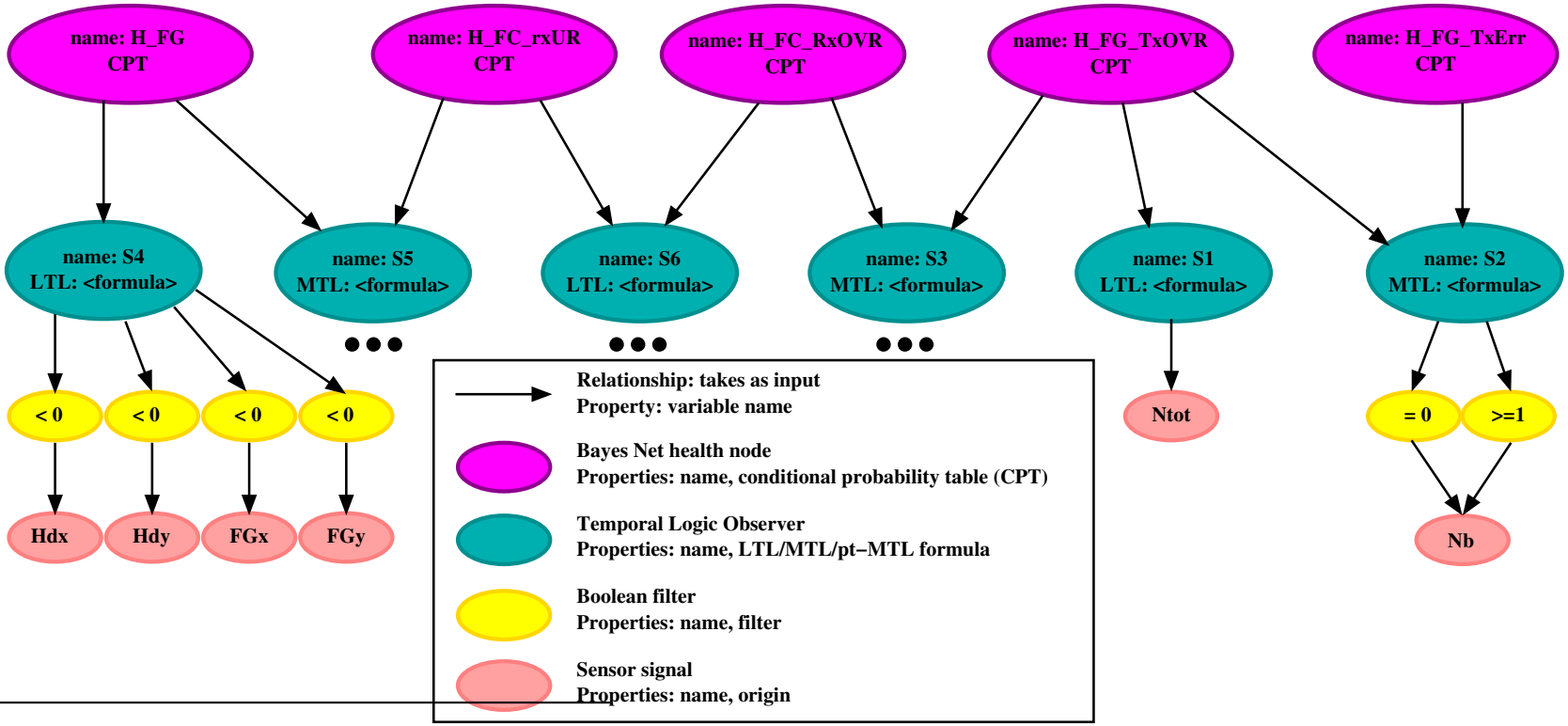


After receiving a command (cmd) for takeoff, the Swift UAS must reach an altitude of 600ft within 40 seconds.

$$(\text{cmd} == \text{takeoff}) \rightarrow \Diamond_{[0,40]}(alt \geq 600 \text{ ft})$$

R2U2 Observation Tree (Specification)[8]

Health Nodes / Failure Modes	
H_FG	**Magnetometer sensor**
H_FC_RxUR	Receiver underrun
H_FC_RxOVR	Receiver overrun
H_FG_TxOVR	**Transmitter overrun in sensor**
H_FG_TxErr	Transmitter error in in sensor



[8] Rozier & Schumann. "R2U2: Tool Overview." In *International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools (RV-CUBES)*, 2017.

MLTL: Not MTL-over-naturals

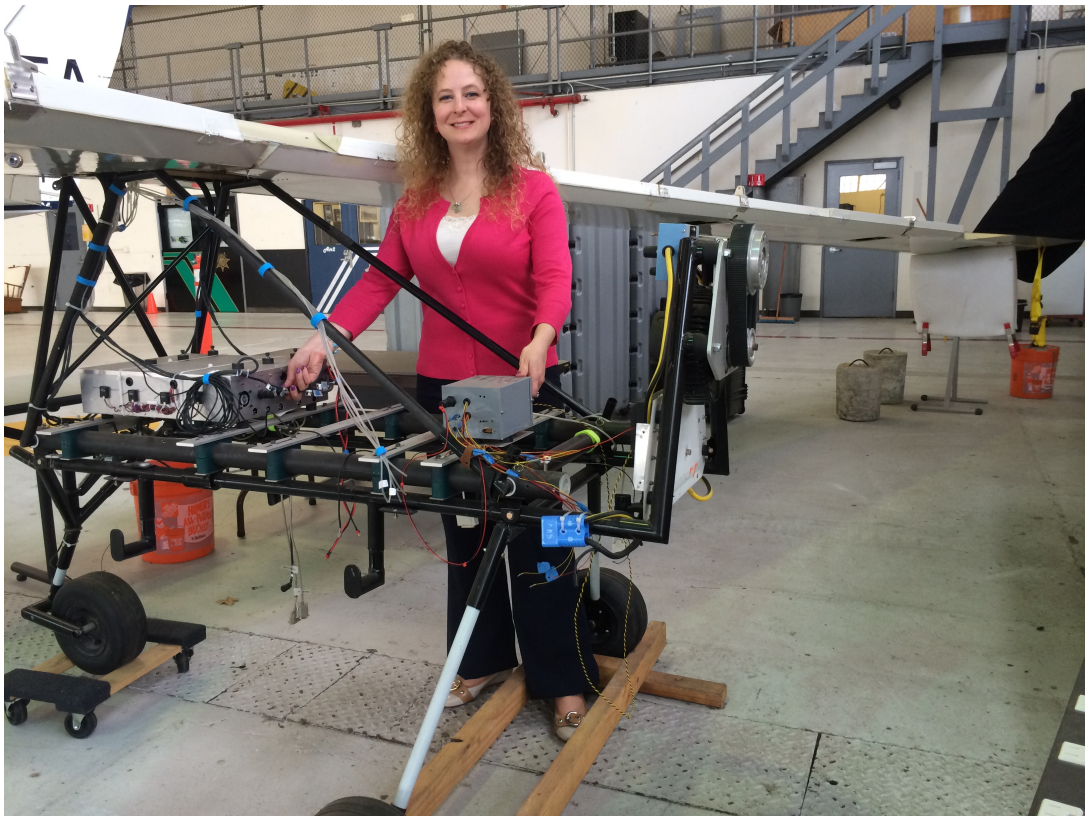
Some important differences:

- Finite traces
- Finite intervals
- **U-semantics**: $\pi \models \varphi \, \mathcal{U}_{[a,b]} \psi$ iff $|\pi| > a$ and, $\exists i \in [a,\ b], i < |\pi|$ such that $\pi, i \models \psi$ and $\forall j \in [a,\ b], j < i$ it holds that $\pi, j \models \varphi$
- Intervals are closed, unit-less (generic)
- Signal processing compartmentalized

Satisfying Requirements

RESPONSIVE
REALIZABLE
UNOBTRUSIVE
Unit

R2U2

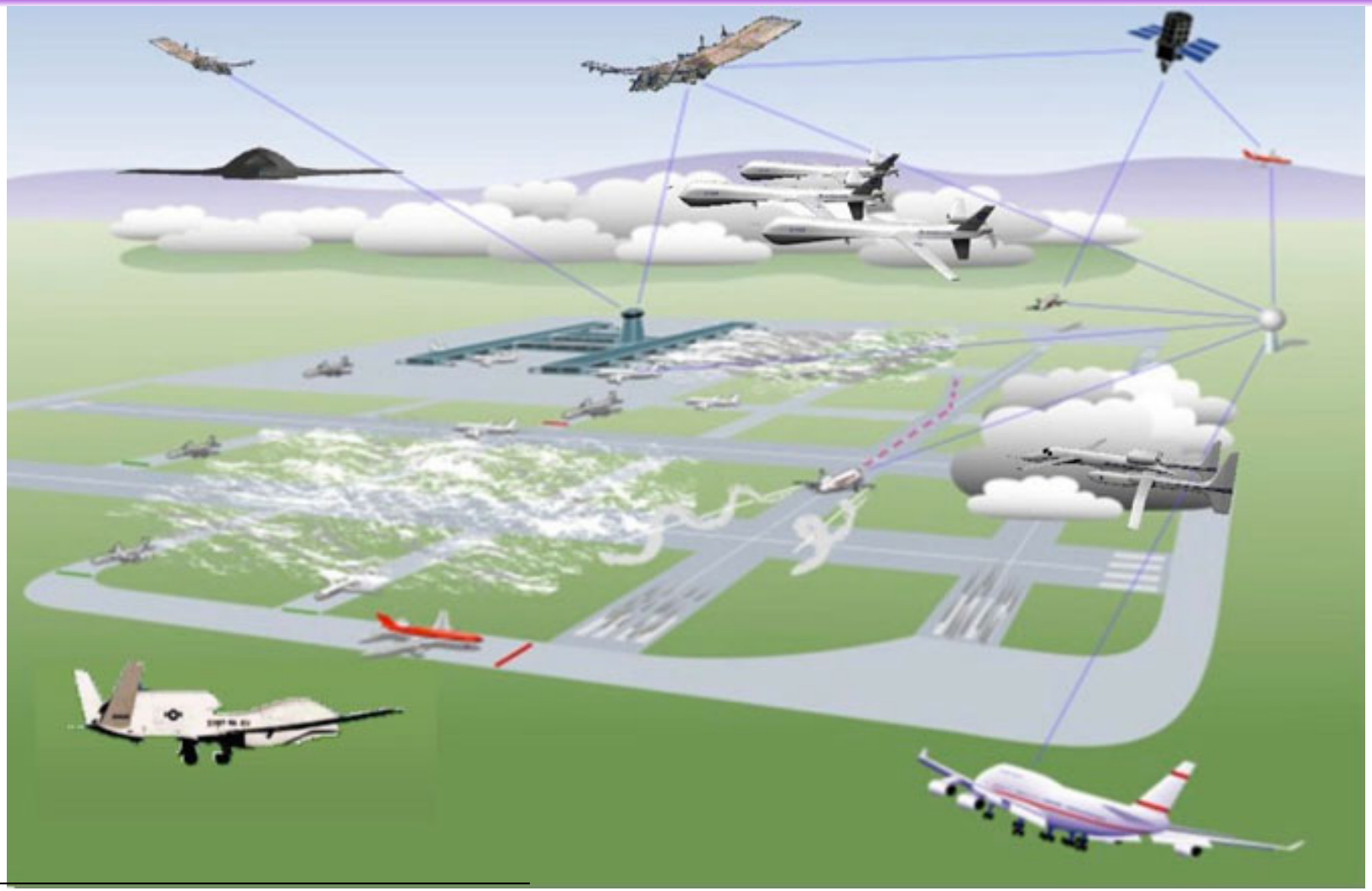


R2U2: Realizable, Responsive, Unobtrusive[9]

1. **Signal Processing**: Preparation of sensor readings
2. **Temporal Logic (TL) Observers**: Efficient temporal reasoning
   1. **Asynchronous**: output $\langle t, \{0, 1\} \rangle$
   2. **Synchronous**: output $\langle t, \{0, 1, ?\} \rangle$
   - **Logics**: Mission-time LTL (MLTL) (plus pt-MLTL, set-wise reasoning)
3. **Bayes Nets**: Efficient decision making
   - **Output**: most-likely status + probability

[9] Kristin Yvonne Rozier, and Johann Schumann. "R2U2: Tool Overview." In International Workshop on Competitions, Usability, Benchmarks, Evaluation, and Standardisation for Runtime Verification Tools (RV-CUBES), held in conjunction with the 17th International Conference on Runtime Verification (RV), Kalpa Publications, Seattle, Washington, USA, September 13-16, 2017.

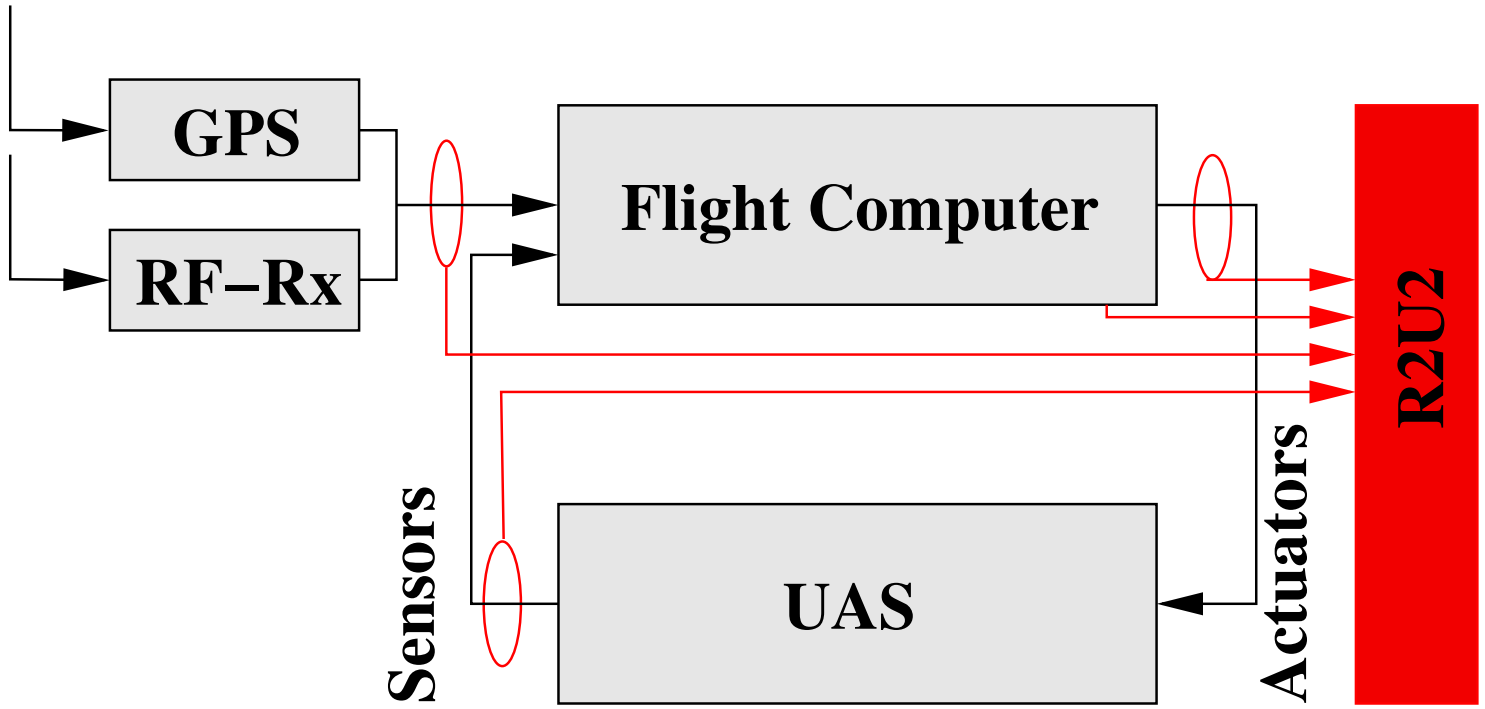
Adding UAS into the NAS: A UTM First Step[10]



[10] Hammer, Cauwels, Hertz, Jones, Rozier. "Integrating runtime verification into an automated UAS traffic management system." *Innovations in Systems and Software Engineering*, 2021

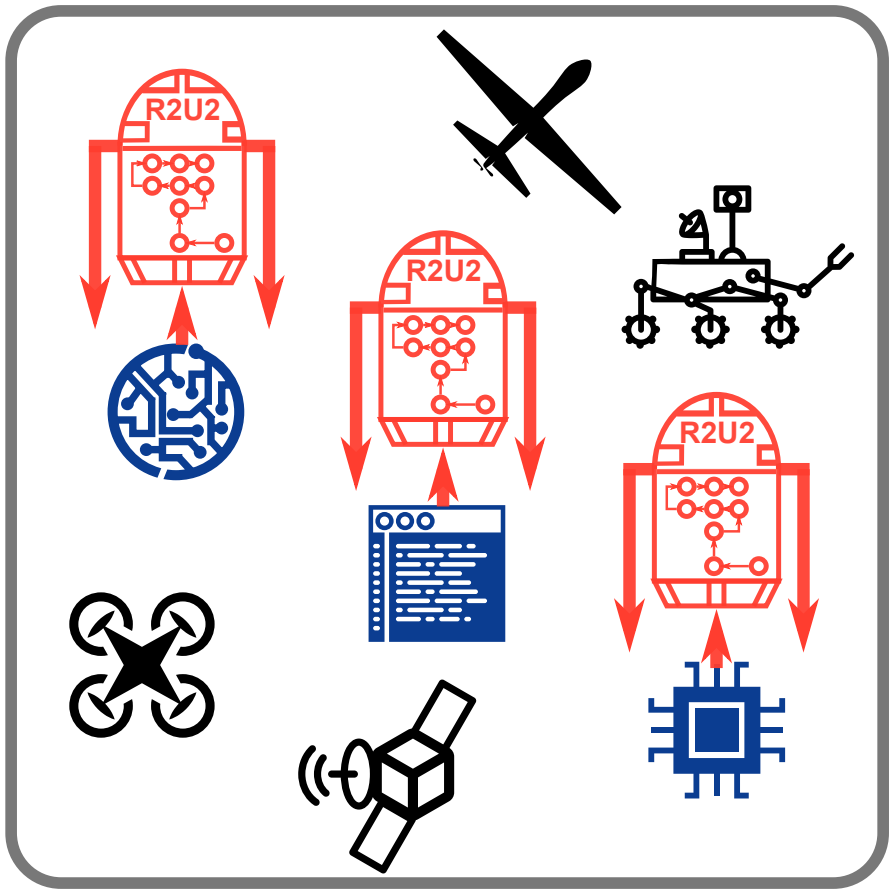
Monitoring and Diagnosis of Security Threats[11]

Threat detection: *attack monitoring*, *post-attack system behavior monitoring*, and *diagnosis*.

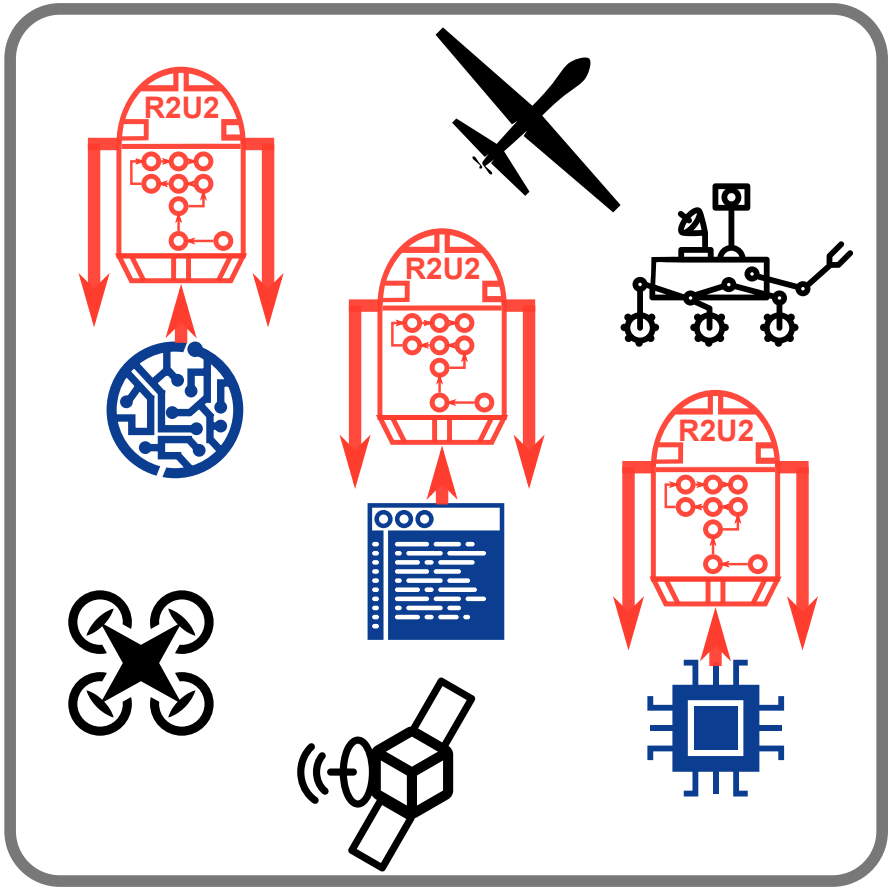


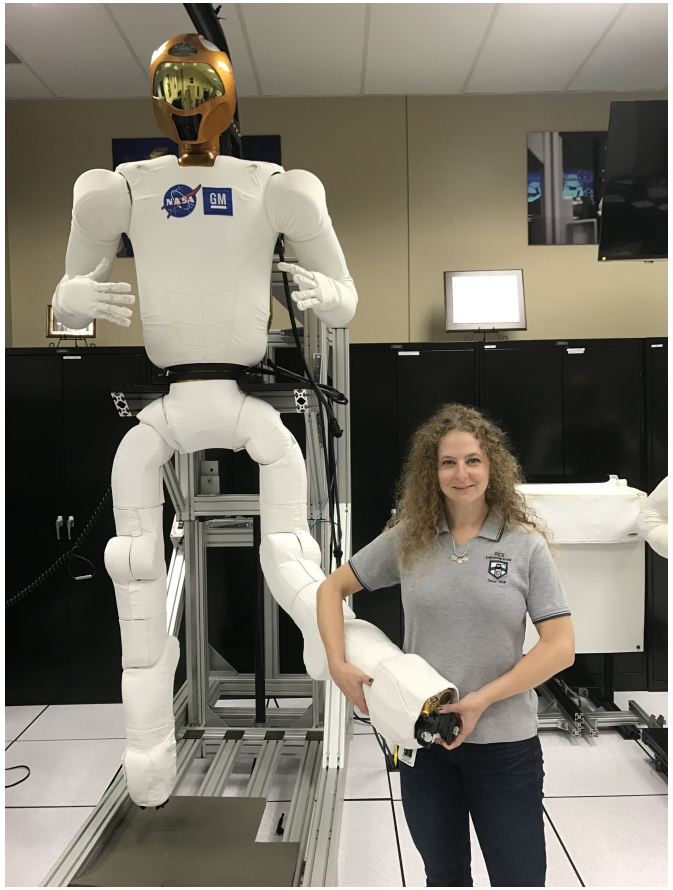
[11] Johann Schumann, Patrick Moosbrugger, Kristin Y. Rozier. "R2U2: Monitoring and Diagnosis of Security Threats for Unmanned Aerial Systems." In *Runtime Verification (RV15)*, Springer-Verlag, September, 2015.

Multi-Platform, Multi-Architecture Runtime Verification of Autonomous Space Systems



Multi-Platform, Multi-Architecture Runtime Verification of Autonomous Space Systems

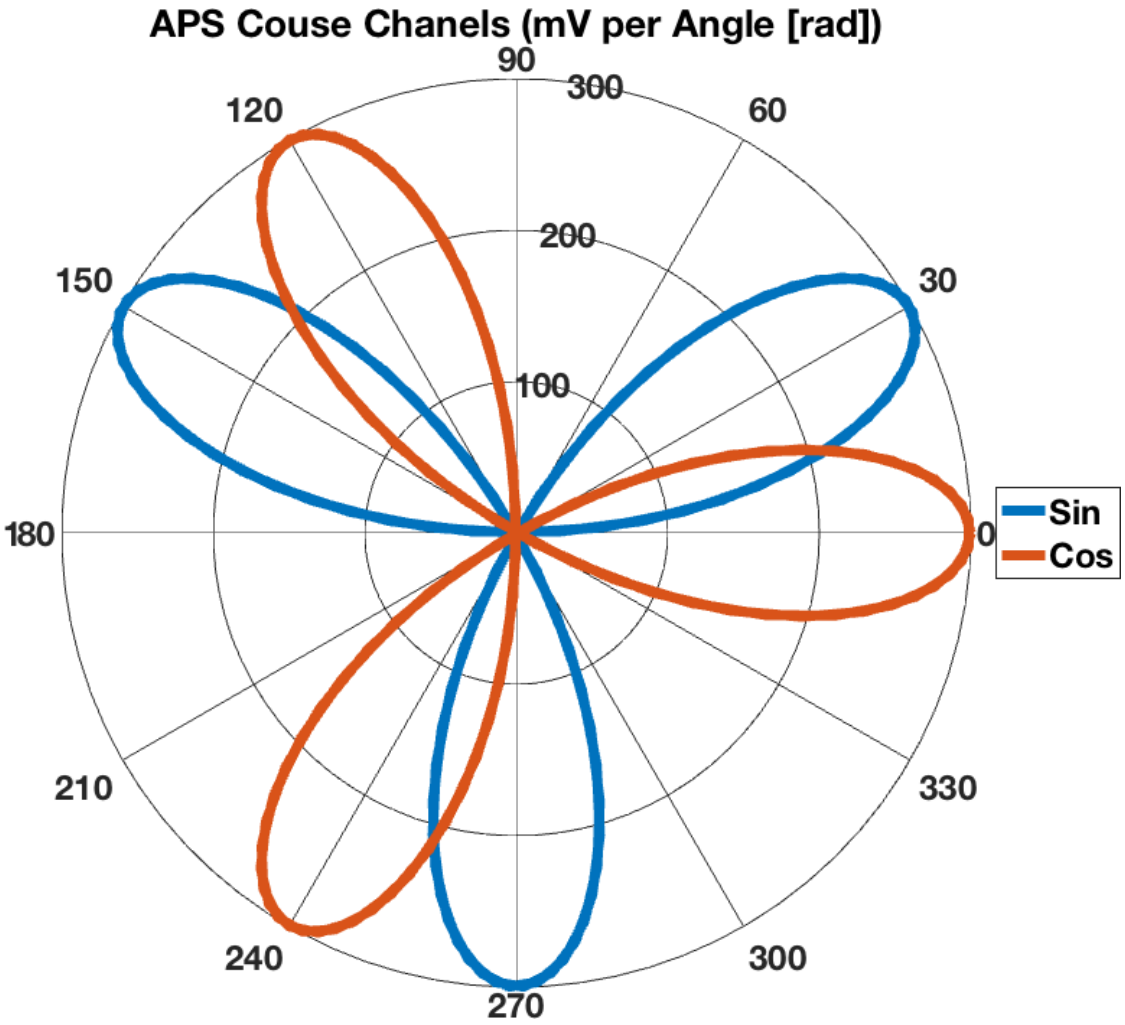




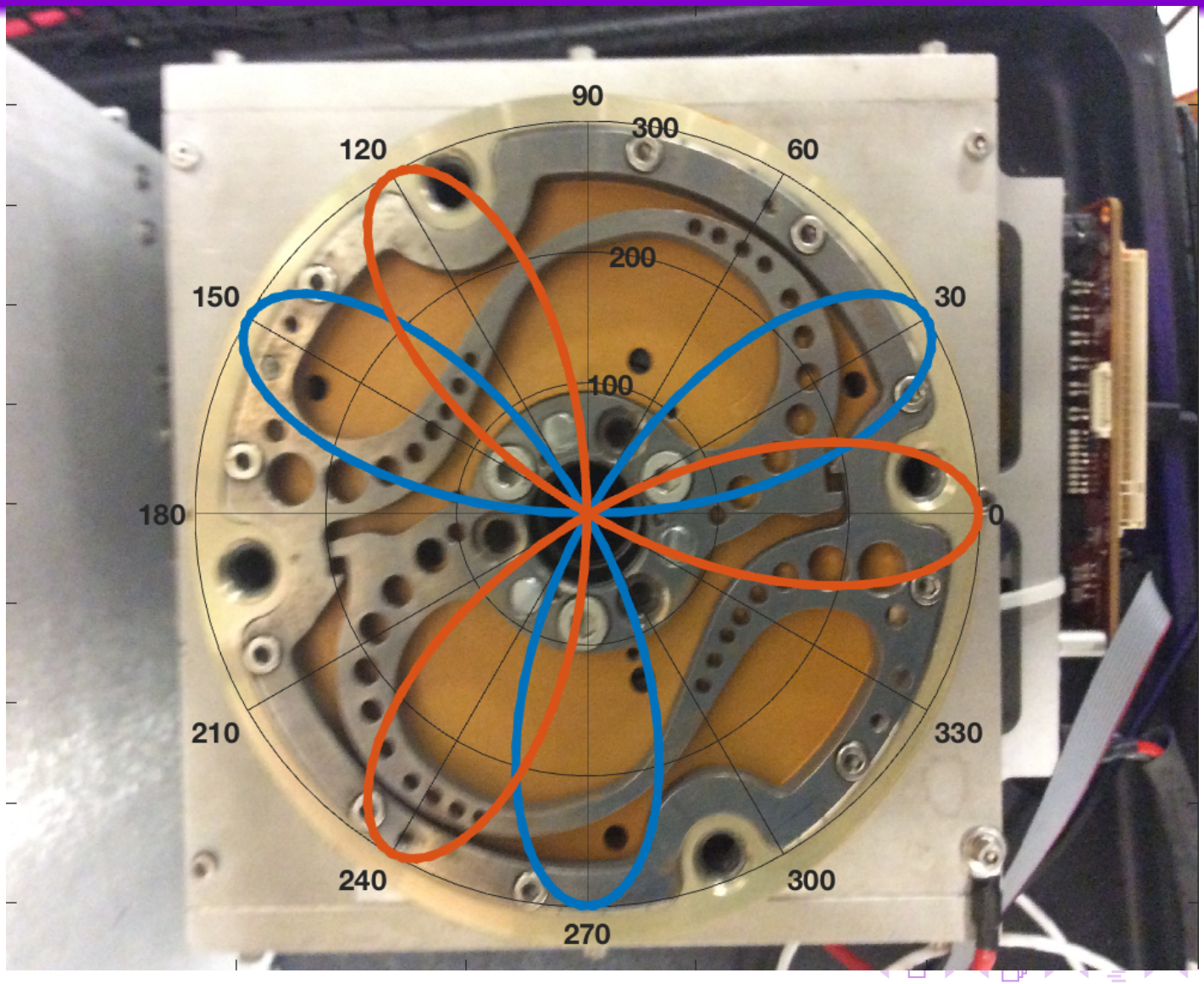
https://temporallogic.org/research/R2U2/FORMATS_18_teaser_BrianKempa.mp4

Robonaut2

Robonaut2's Knee

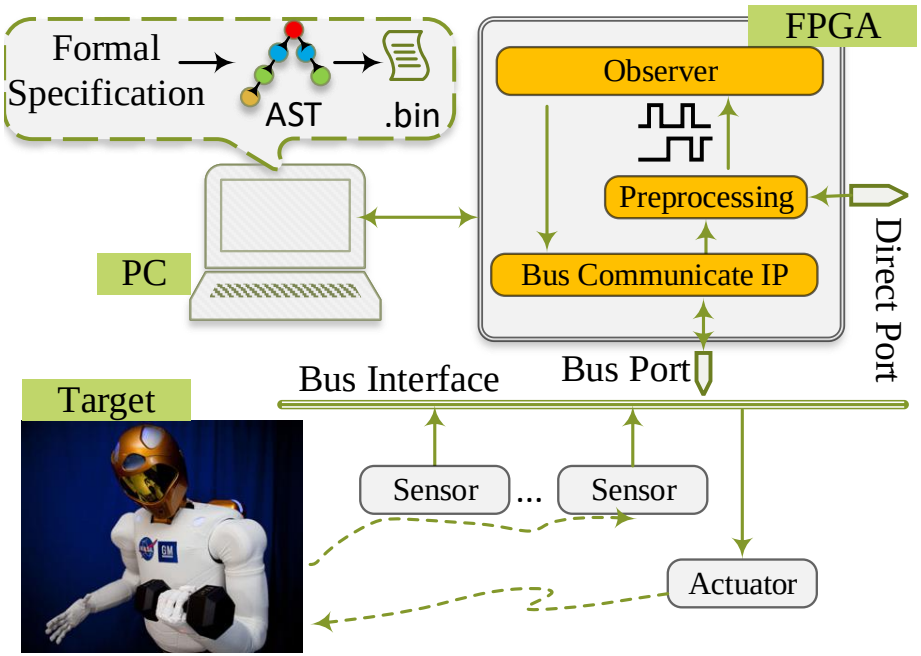


Robonaut2's Knee



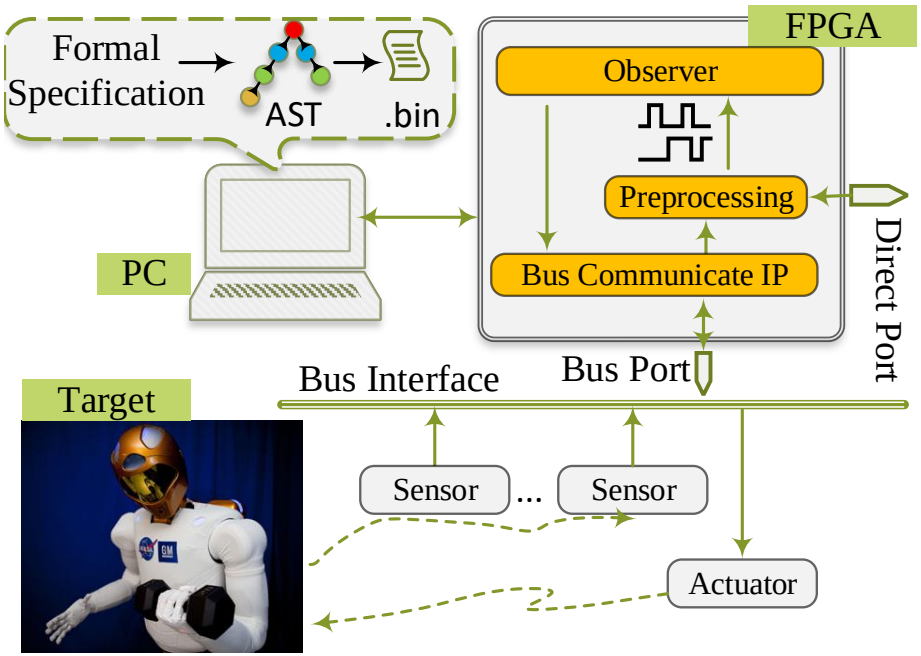
http://temporallogic.org/research/R2U2/R2U2-on-R2_demo.mp4

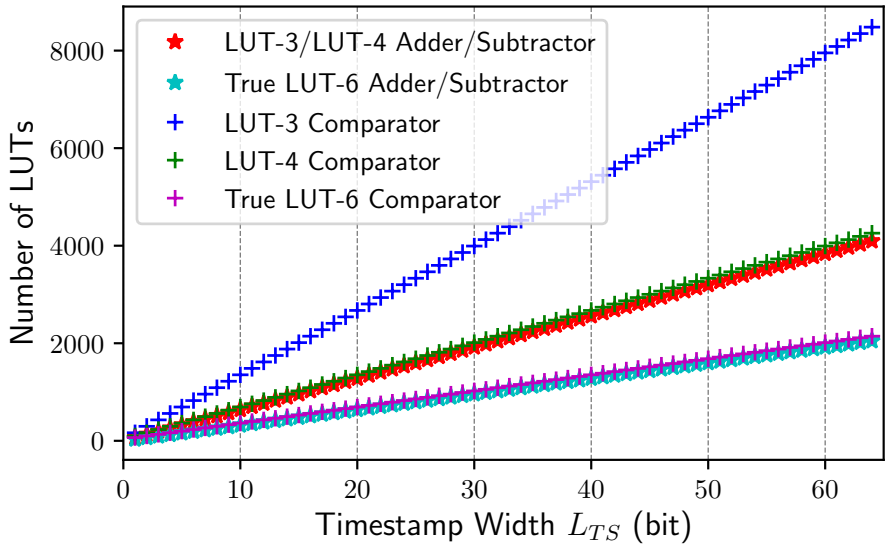
Resource Estimation and Improved Encoding Algorithms [12]



[12] B.Kempa, P.Zhang, P.H.Jones, J.Zambreno, K.Y.Rozier. "Embedding Online Runtime Verification for Fault Disambiguation on Robonaut2." FORMATS, LNCS vol 12288, 2020.

Resource Estimation and Improved Encoding Algorithms [12]





[12] B.Kempa, P.Zhang, P.H.Jones, J.Zambreno, K.Y.Rozier. "Embedding Online Runtime Verification for Fault Disambiguation on Robonaut2." FORMATS, LNCS vol 12288, 2020.

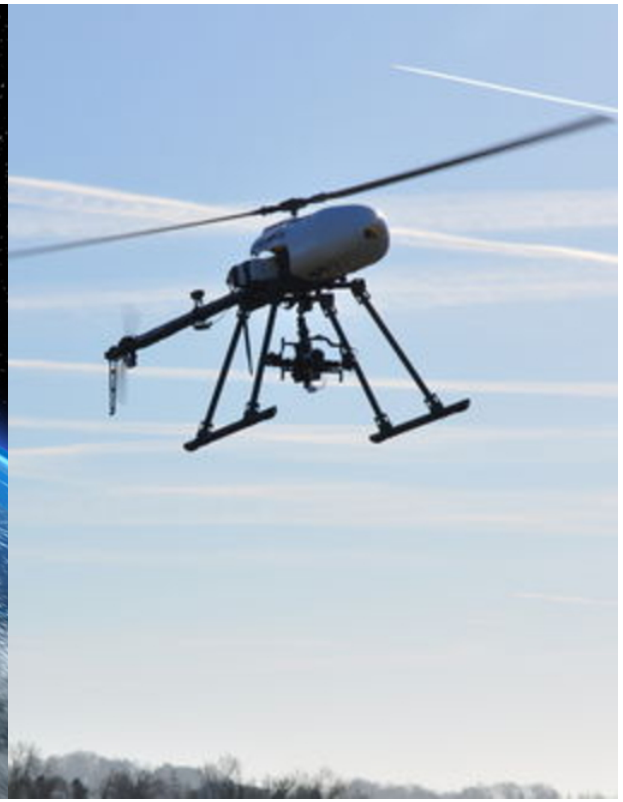
Cyclone Sounding Rocket!

https://www.youtube.com/watch?v=p6dwT0sTdH0&t=158s

Flight-Certification == Proofs that Fly! [13] [14] [15] [16]







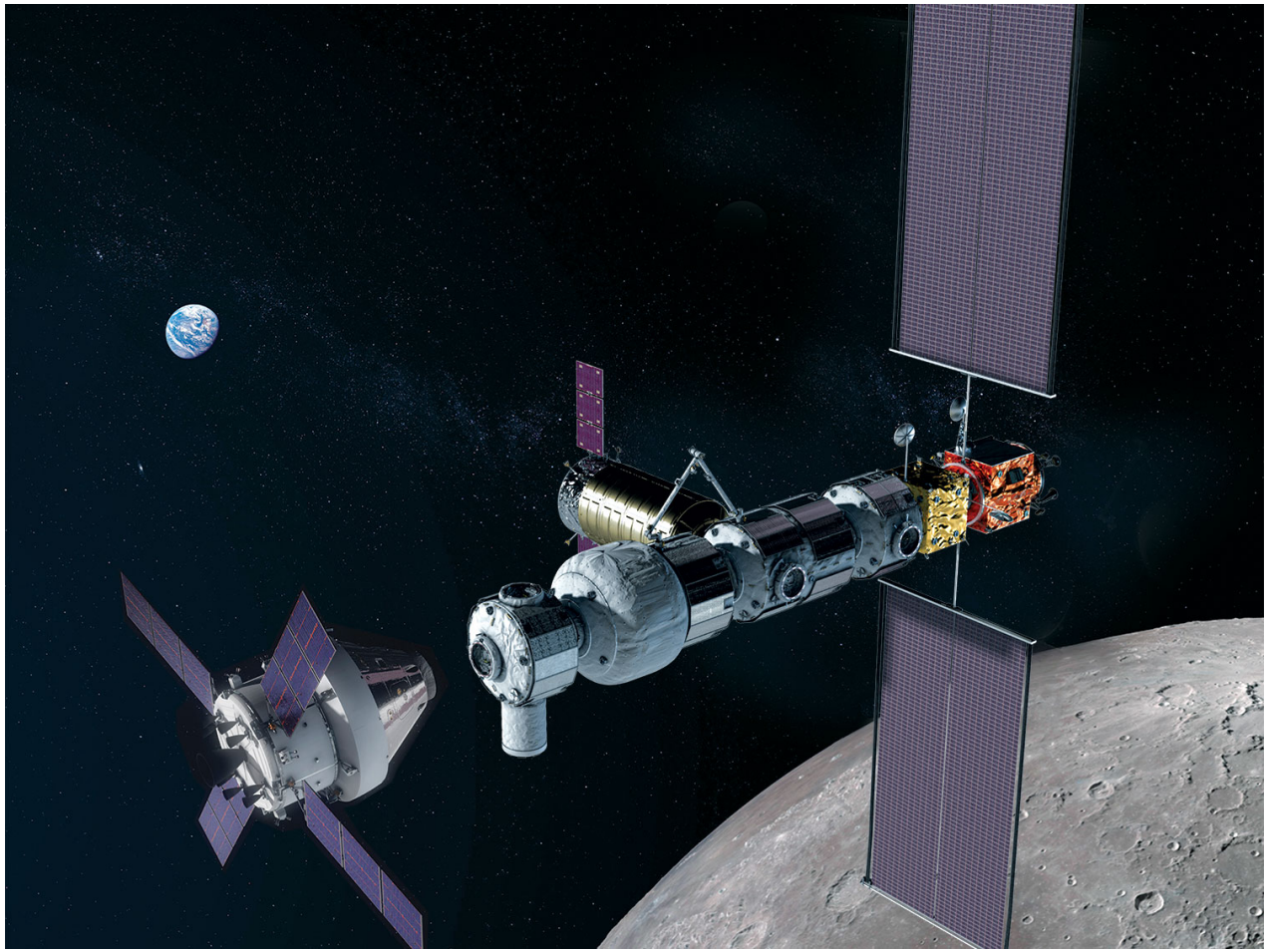
[13] Hariharan, Kempa, Wongpiromsarn, Jones, Rozier. "MLTL Multi-type (MLTLM): A Logic for Reasoning about Signals of Different Types." NSV 2022.

[14] Luppen, Jacks, Baughman, Hertz, Cutler, Lee, Rozier. "Elucidation and Analysis of Specification Patterns in Aerospace System Telemetry." NFM 2022.

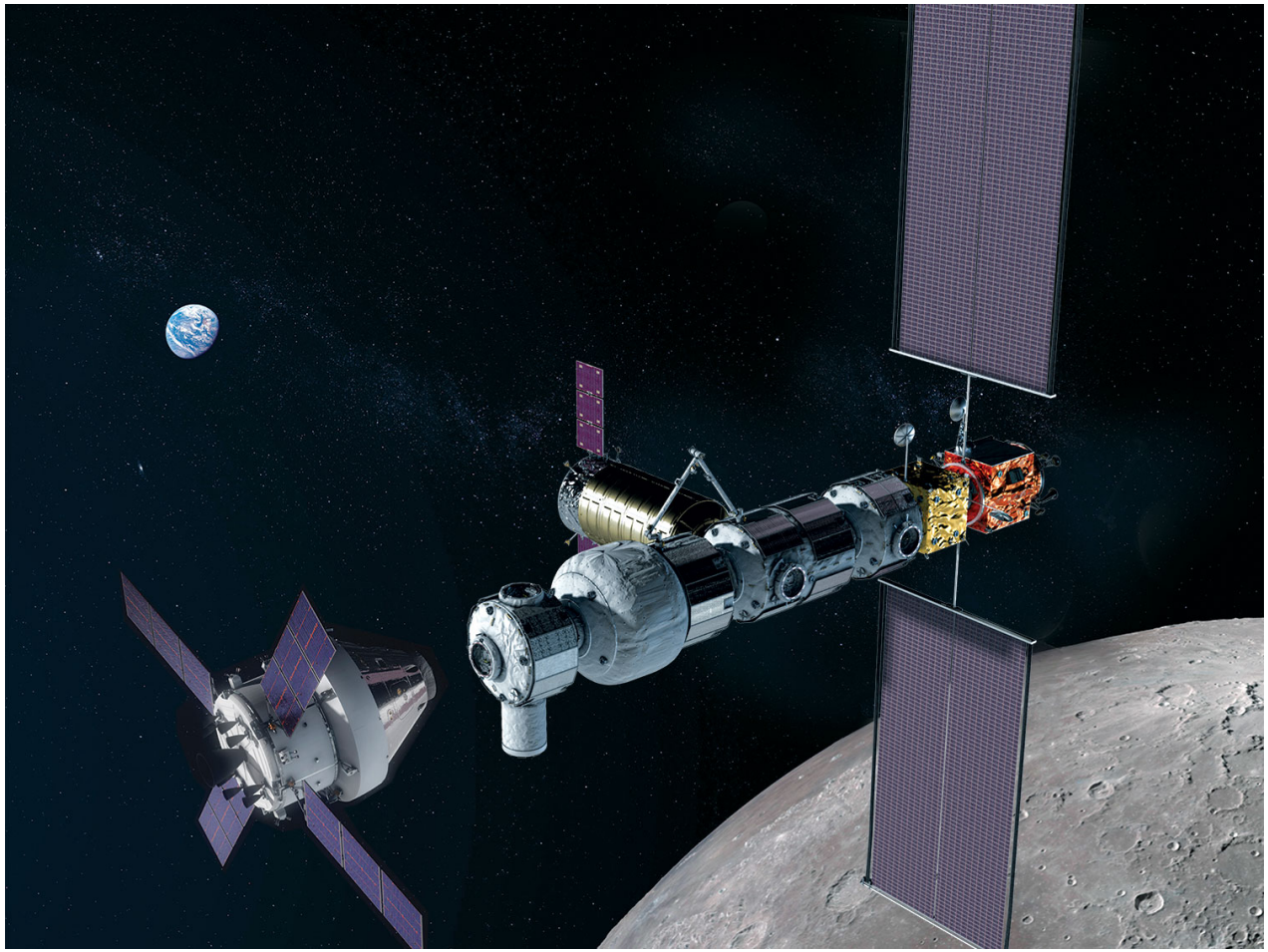
[15] Hertz, Luppen, Rozier. "Integrating Runtime Verification into a Sounding Rocket Control System." NFM 2021.

[16] Hammer, Cauwels, Hertz, Jones, Rozier. "Integrating Runtime Verification into an Automated UAS Traffic Management System." *Innovations in Systems and Software Engineering: A NASA Journal* 2021.

NASA Lunar Gateway Vehicle System Manager V&V



NASA Lunar Gateway Vehicle System Manager V&V



$$(CMD = START) \rightarrow (\Diamond_{[0,5]}(ActionHappens \ \& \Diamond_{[0,2]}(CMD = END)))$$

NASA Lunar Gateway V&V Using MLTL

ARC AEROSPACE RESEARCH CENTRAL Adding a Verification View for an Autonomous Real-Time System Architecture

James B. Dabney, Julia M. Badger, Pavan Rajagopal

AIAA 2021-0566; SE-05:Systems Engineering V, 12 January 2021

https://doi.org/10.2514/6.2021-0566

Video: https://doi.org/10.2514/6.2021-0566.vid

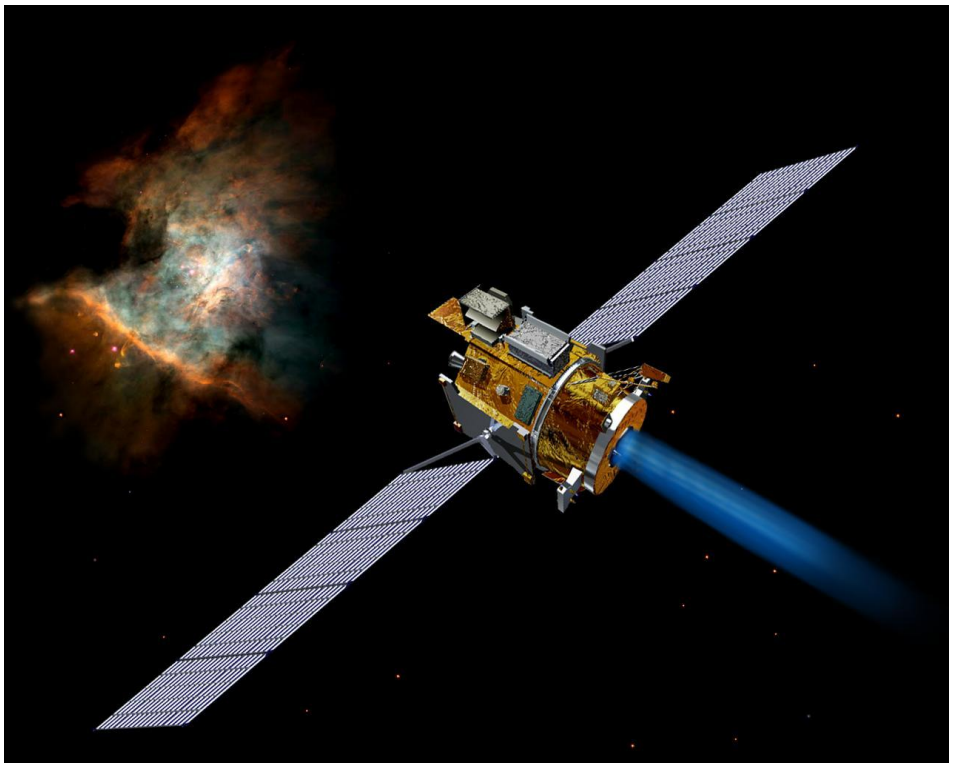
FSW 2021: Using Assume-Guarantee Contracts In Autonomous Spacecraft - James Bruster Dabney

https://www.youtube.com/watch?v=zrtyiyNf674

FSW 2022: Using Assume-Guarantee Contracts for Developmental Verification of Autonomous Spacecraft

https://www.youtube.com/watch?v=HFnn6TzblPg

MLTL Multi-type (MLTLM): A Logic for Reasoning About Signals of Different Types[17]



The spacecraft maintenance cycle runs at least once a month over the five-year mission.

Monthly course corrections never involve burning the thrusters more than 3 seconds at a time.

$$\Box_{[0,5,year]}[(\Diamond_{[0,30,day]} maintenance) \wedge (\neg\Box_{[0,3,sec]} thrusters)]$$

[17] Hariharan, Kempa, Wongpiromsarn, Jones, Rozier, NSV 2022

MLTL is Unusually Effective!

- Easier to **accurately represent timing constraints** of real systems (e.g., ATC bounds)
- Easier to **validate** (matches real system better)
- Low **complexity/memory** to check
- **Generic, reusable** specifications are robust to hardware substitutions/clock changes
  - Can **tune timescales** for resource trade offs on embedded systems
- **Separation of concerns** (Boolean testers, temporal logic, intervals)
  - Allows **re-use of processed signals** outside of logic
  - Retain **validation/complexity/size benefits** while separating out extensions
- **Fits into business processes for real CPS development**

Future Directions:Make MLTL Even More Effective!

- MLTL **model checking**
- MLTL $\rightarrow$ **automata**
- Better MLTL **satisfiability checking**
- MLTL **synthesis**
- MLTL **planning?**
- Better MLTL **elicitation** and **validation**
- MLTL **explainability**