Asteria: Space and Satellites Seminar

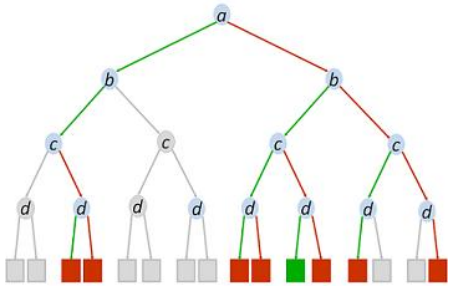
Kristin Yvonne Rozier, Iowa State University
http://laboratory.temporallogic.org
May 19, 2022

Research Interests

Automated Reasoning

- Avionics/Flight Software
- Satisfiability (SAT)/SMT
- AI/Algorithms
- Explainability

Formal Specification

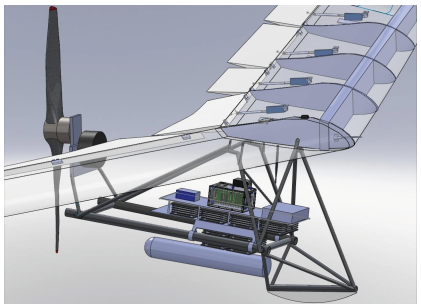


- Specification Patterns
- Specification Debugging
- Consistency/Temporal Satisfiability Checking

Design-time Safety Analysis

- Model Checking (Explicit and Symbolic)
- Model Based Design
- Requirements Elicitation
- Temporal Logic Encoding

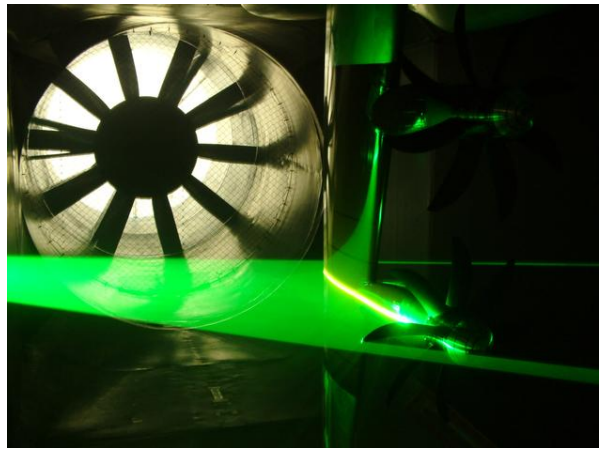
Runtime Verification



- R2U2 Engine
- System Health Management
- Resource-limited Sanity Checking
- Automated Diagnostics/Prognostics
- Real-time Intelligent Sensor Fusion

Path to Iowa State University

WILLIAM & MARY



NASA LANGLEY 90 YEARS ON THE LEADING EDGE SINCE 1917

LANGLEY FORMAL METHODS

RICE

DISCOVERY • INNOVATION • SOLUTIONS 75th Anniversary 1939-2014 Ames Research Center

IOWA STATE UNIVERSITY OF SCIENCE AND TECHNOLOGY

STATE

Formal Methods Research

Intuitively, the system does what you think it should do
and nothing else.

Design-Time Verification

- produces automated, replay-able proofs of the absence of behaviors we don't want (in addition to the presence of behaviors we want)

Runtime Verification

- checks on-board, in real time, during flight that the system is still upholding its requirements even if there were some off-nominal conditions we couldn't anticipate during design time
- R2U2 is the only flight-certifiable runtime verification engine currently in the literature

On the front lines against software bugs and unexpected emergent behaviors

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks

How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks

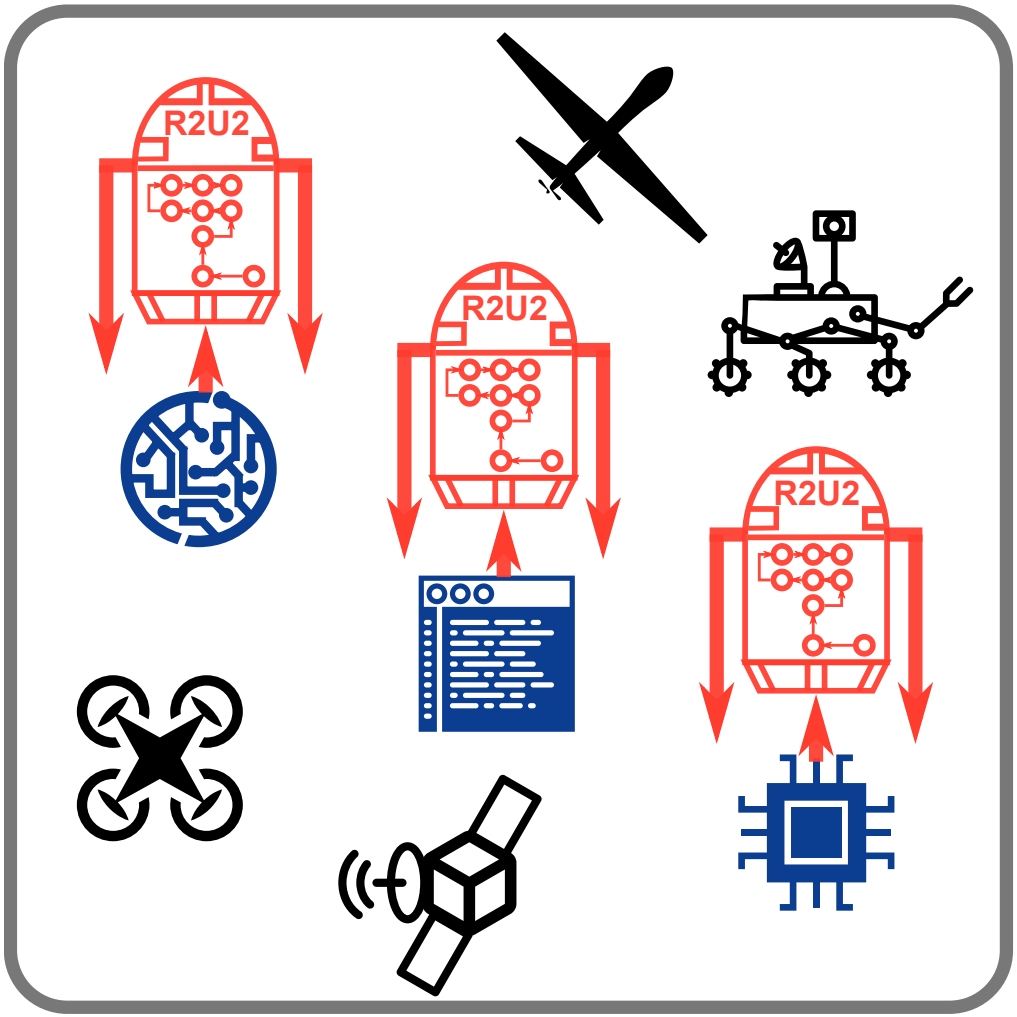
How is **Flight Software** Different from **Software**?

- **Has to** work
- Need capabilities for **independent checks**
- Need **transparent** ties to **verification** tasks

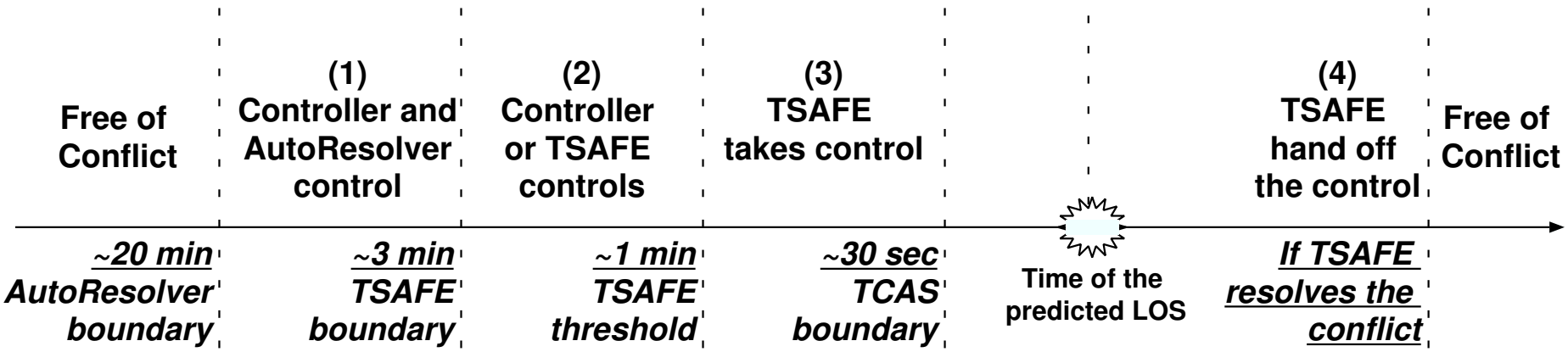
Satisfying Requirements for Flight Software

RESPONSIVE
REALIZABLE
UNOBTRUSIVE
Unit

R2U2

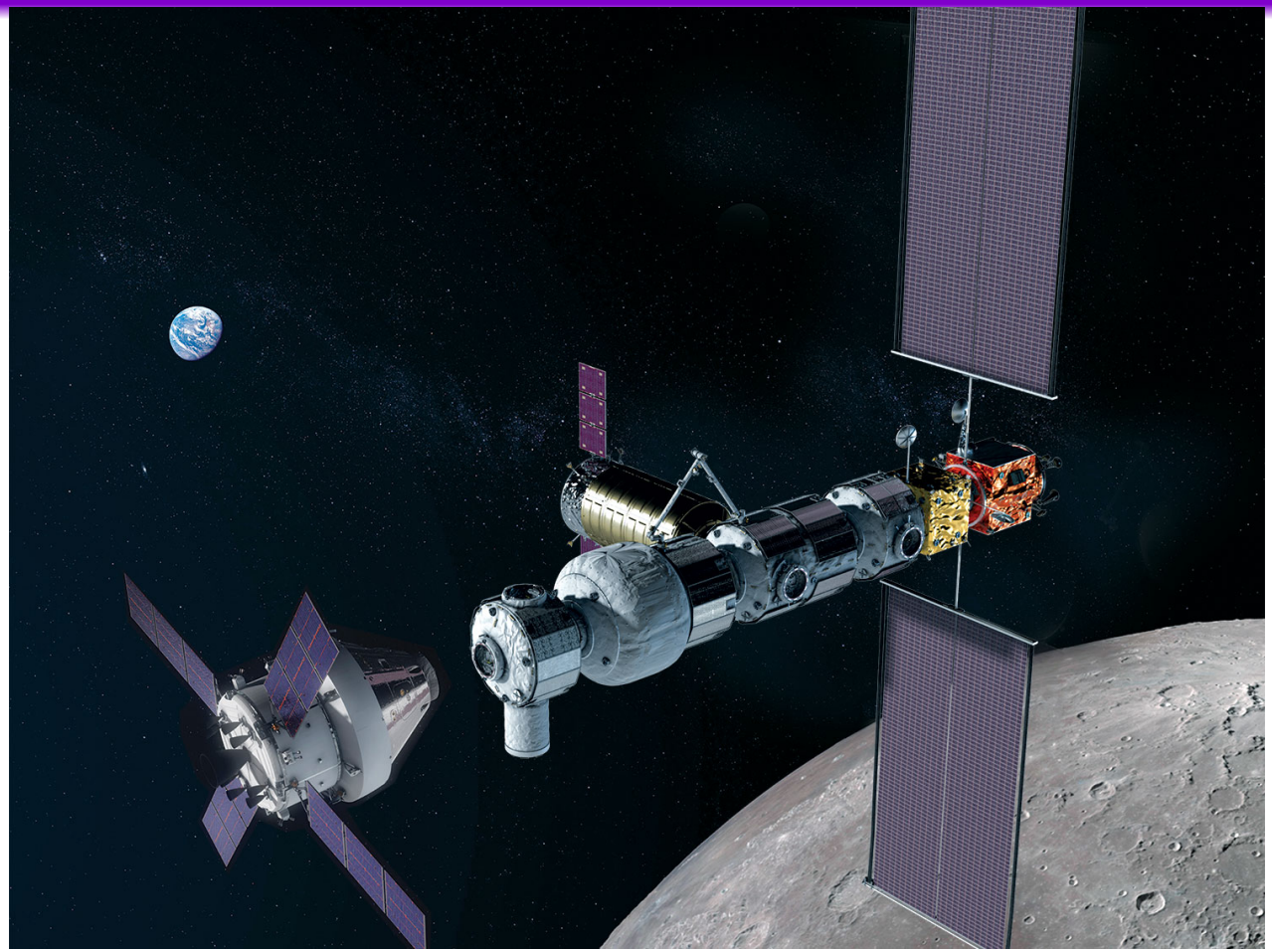


Automated Airspace Concept High-Level Architecture[1]



[1] H. Erzberger, K. Heere, Algorithm and operational concept for resolving short-range conflicts, Proc. IMechE G J. Aerosp. Eng. 224 (2) (2010) 225243

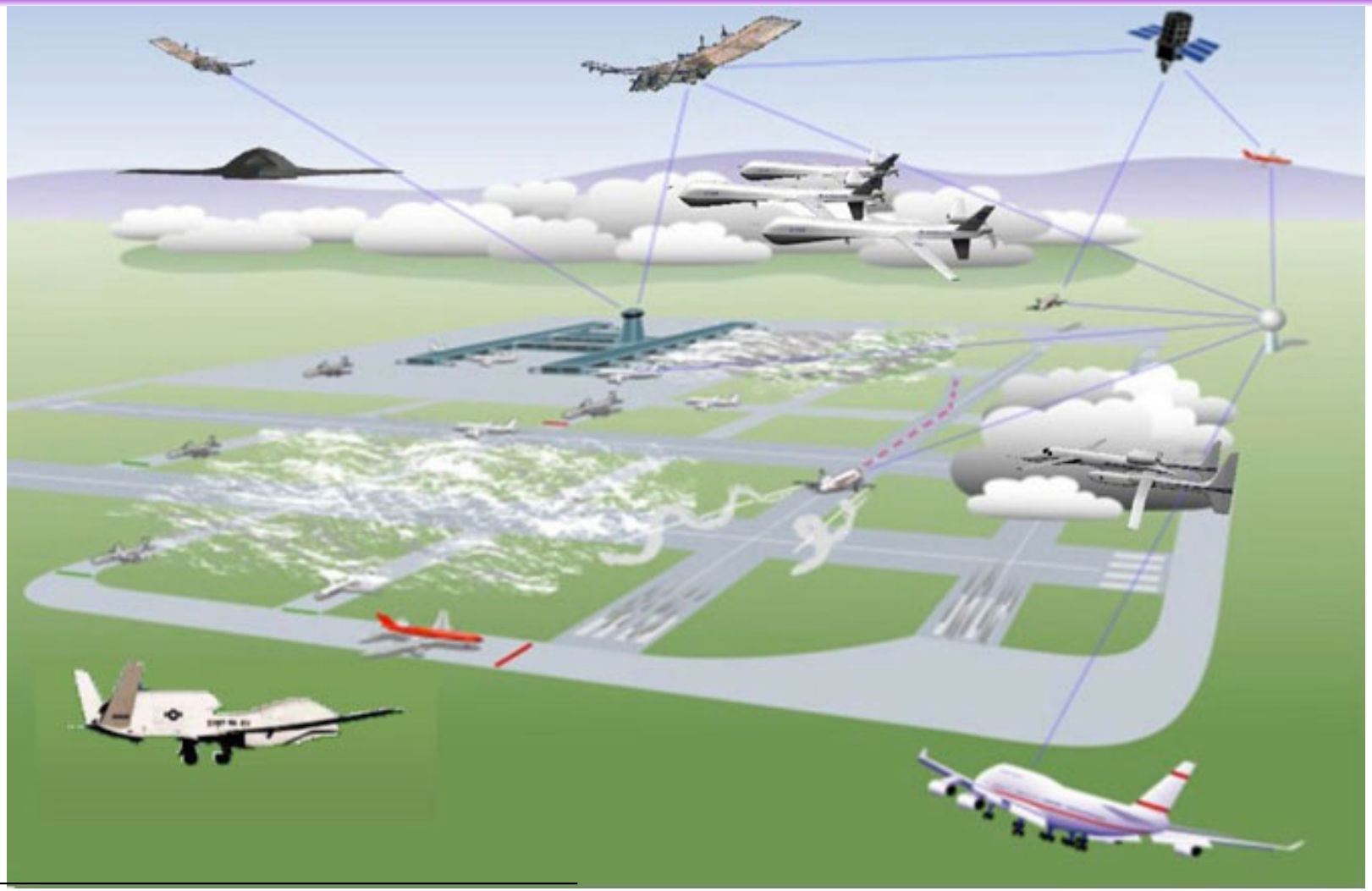
NASA Lunar Gateway: Assume-Guarantee Contracts[2]



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

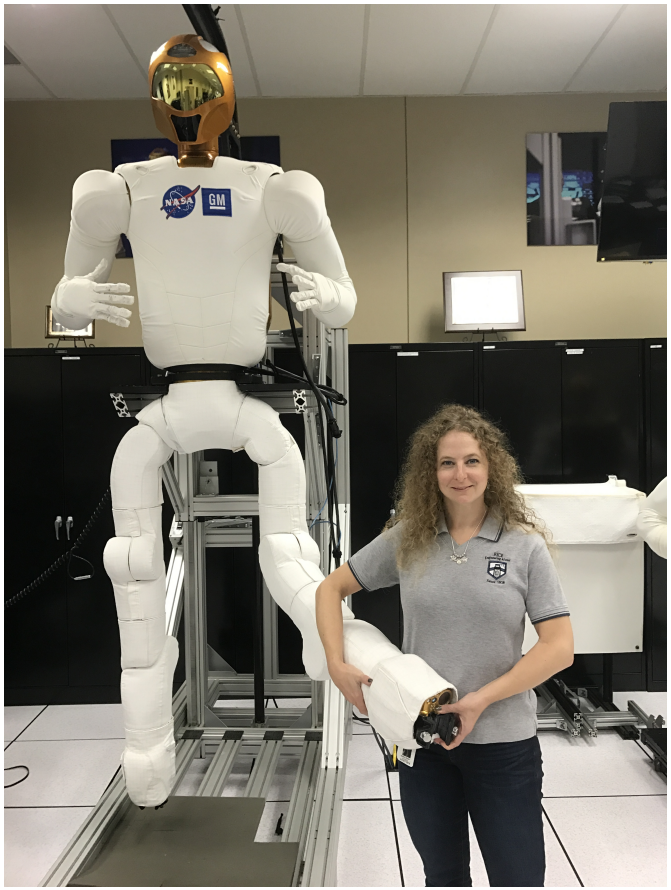
[2] Dabney, James B., Julia M. Badger, and Pavan Rajagopal. "Adding a Verification View for an Autonomous Real-Time System Architecture." In AIAA Scitech 2021 Forum, p. 0566. 2021.

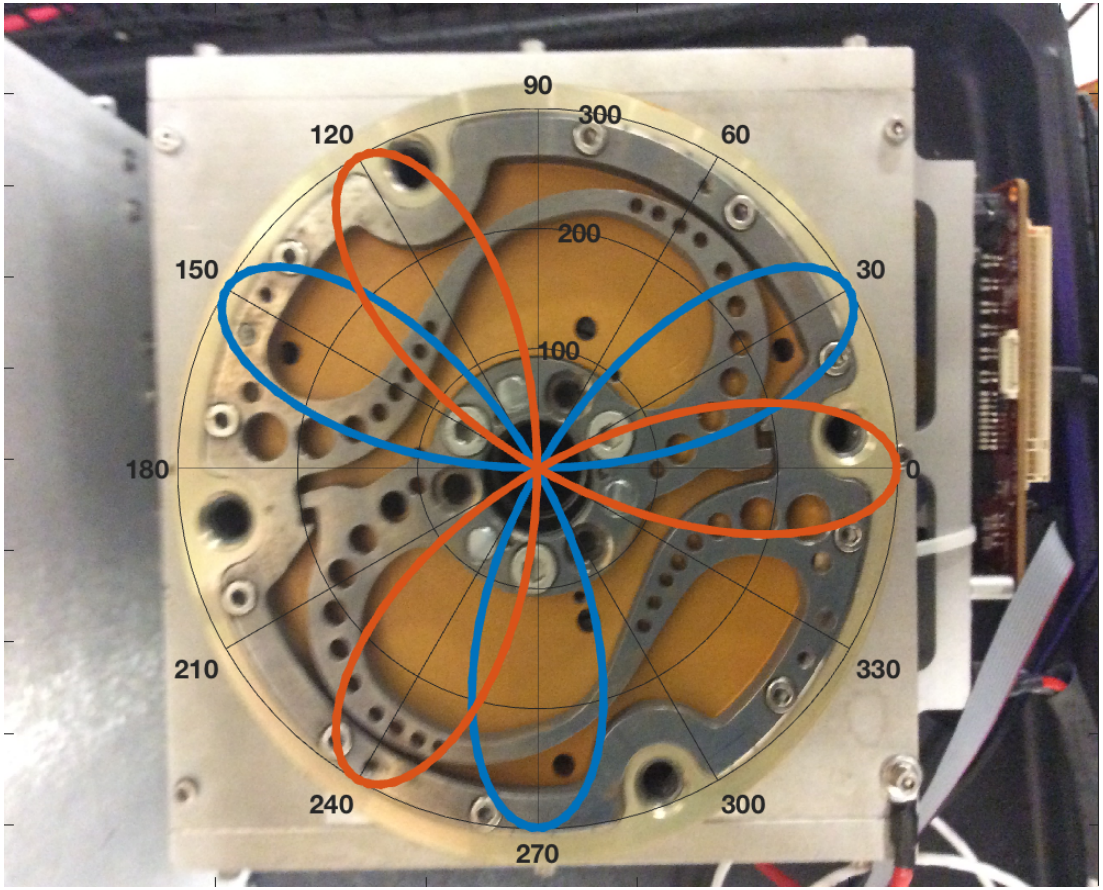
Adding UAS into the NAS?[3]



[3] Matthew Cauwels, Abigail Hammer, Benjamin Hertz, Phillip Jones, and Kristin Yvonne Rozier. "Integrating Runtime Verification into an Automated UAS Traffic Management System." *DETECT* 2020

Robonaut2's Knee[4]





[4] Kempa, Brian, Pei Zhang, Phillip H. Jones, Joseph Zambreno, and Kristin Yvonne Rozier. "Embedding online runtime verification for fault disambiguation on Robonaut2." *FORMATS* 2020.

Formal Methods Research

Intuitively, the system does what you think it should do
and nothing else.

Design-Time Verification

- produces automated, replay-able proofs of the absence of behaviors we don't want (in addition to the presence of behaviors we want)

Runtime Verification

- checks on-board, in real time, during flight that the system is still upholding its requirements even if there were some off-nominal conditions we couldn't anticipate during design time
- R2U2 is the only flight-certifiable runtime verification engine currently in the literature

On the front lines against software bugs and unexpected emergent behaviors