Applications of F′ for Split-Computing

Alex Mariano, Jerold Emansala
Professor Thomas Huang
CS3700 Parallel Processing

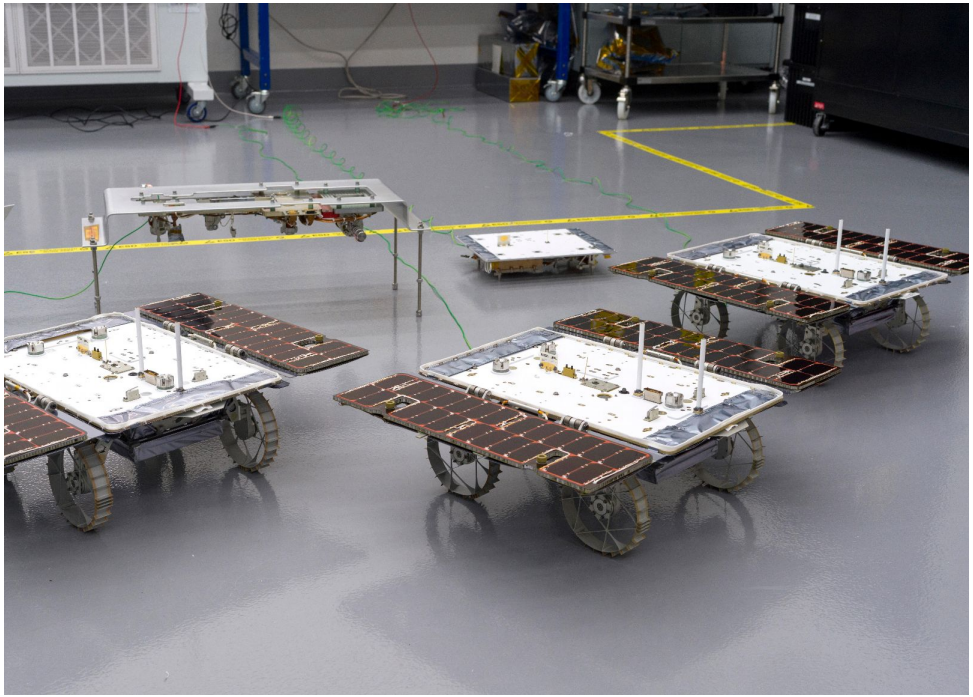
Motivation

- More than 2,300 small satellites launched world-wide.
- Rise in AI applications lead to potential usage in space.
- Enable efficiency and reusability.
- CubeSats offer low cost solution for experimentation.
- Terabytes of satellite imagery is generated every day.

What is F′

- NASA-JPL’s open-source flight software framework.
- Built to support a greater network of space flight missions.
- Emphasis on versatility, reusability, and portability without sacrificing reliability and performance.

Relevance and Usage



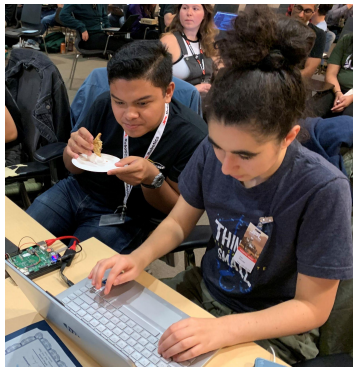
- Heightened need for autonomous functionality to address physical and fiscal constraints.
- Terabytes of instrument data are generated per orbit, requiring need for in-situ tasking.
- Split computing architectures are in experiment phase and may lead to faster task completion.

- CADRE: Network of rovers working together using F′ to explore the Moon.

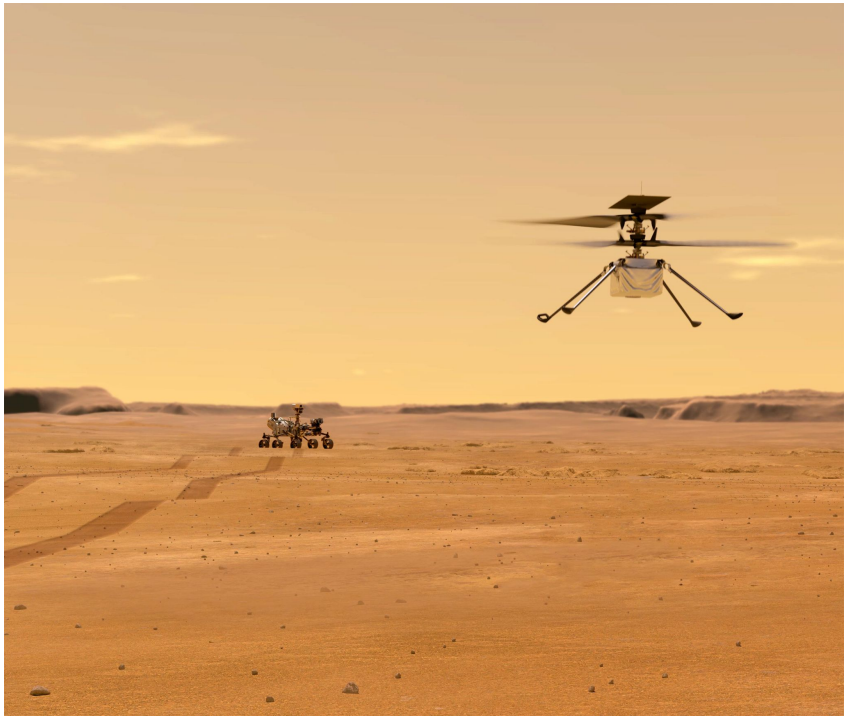
Impact

- NASA-JPL F′ team showcases framework as an educational tool to learn about FSW and apply them in university CubeSat projects.
- Open-Source Community with high developer interaction and ease of support.

- Current Research at CPP Bronco-STAR (Bronco Space) for split-computing architectures and machine learning.



Real “Space” Applications and Architectures

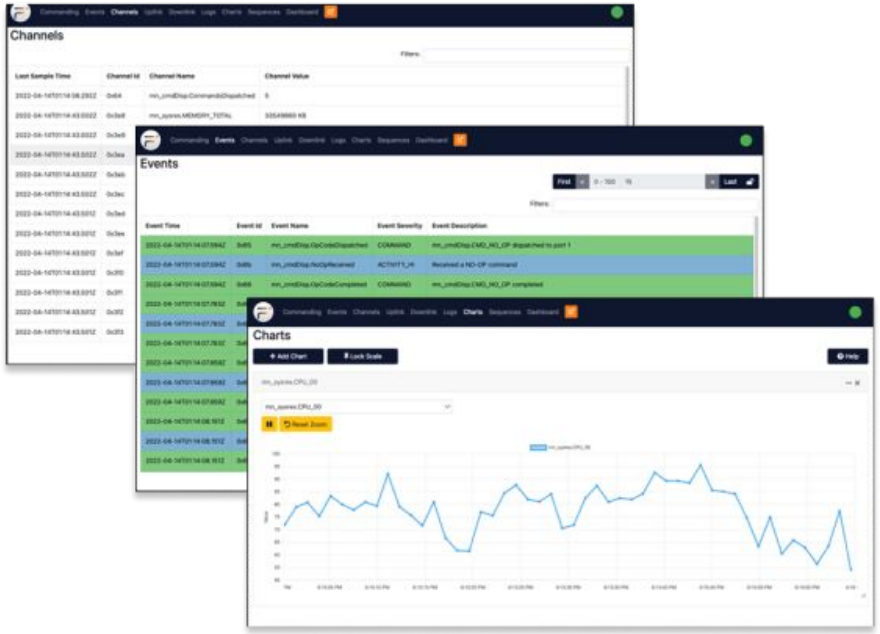


- ISS-RapidScat Instrument (2014-2018)
- ASTERIA CubeSat (2017-2020)
- Ingenuity Mars Helicopter (2020-)
  - Had two processors running F′ to handle flight controls and navigation.
- Lunar Flashlight CubeSat (2022-)
- NEAScout CubeSat (2022-)
- OWLS Instrument
- CADRE Rovers (prosp. 2025)
  - Group of rovers (3) working cooperatively to explore the lunar surface.
- COLDArm Robotic Arm

Performance Statistics

- Modularity and reusability
- Low resource usage
- F′ utilizes threading for precise timing and scheduling of component port invocations

- Split-Computing architectures allow a computer framework to make use of heavy-lifting processors and hardware while keeping main FSW available.
- Use of multiple machines in a single mission allows for larger collection of data and bigger scientific instruments.



Strengths and Limitations

Strengths

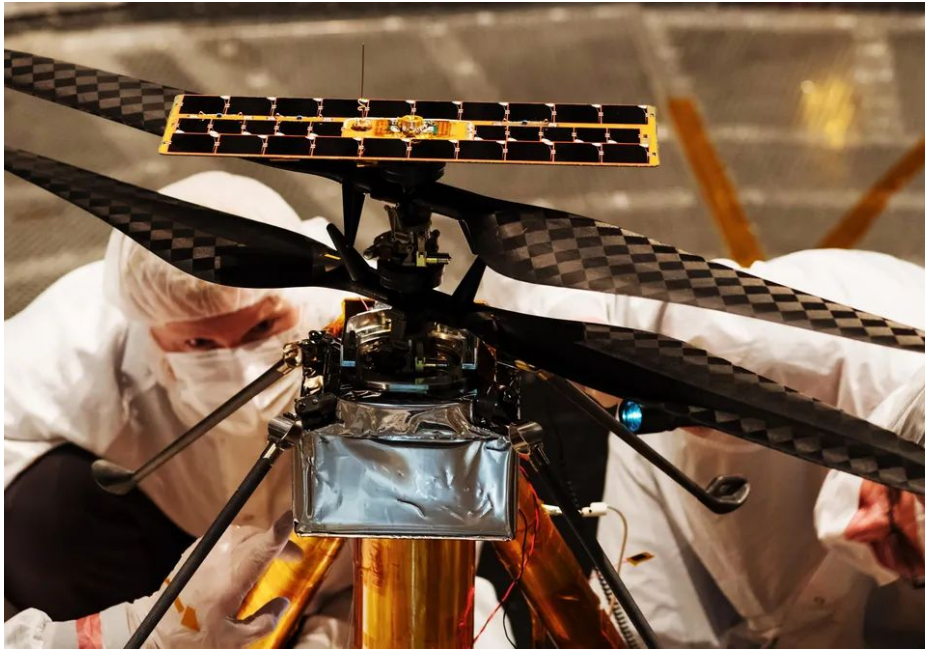
- Flight-Proven Reliability
- Easier to learn compared to other open-source FSWs.
- Portability to other projects.
- Active development and issue tracking

Limitations

- Steep Initial Learning Curve for learning FSW
- Limited Flexibility Outside Paradigm
- Smaller Ecosystem compared to other open-source communities.
    - Few Tutorials

Technology Maturity

- TRL 8-9.
- Flight Heritage on Numerous Missions.
- Active development and issue tracking
- Advertised for other Researchers, Projects, and Universities to pick up.
- Existing flight-light simulation and testbed development.
- Streamlined FSW Development, from design to integration.



Demo