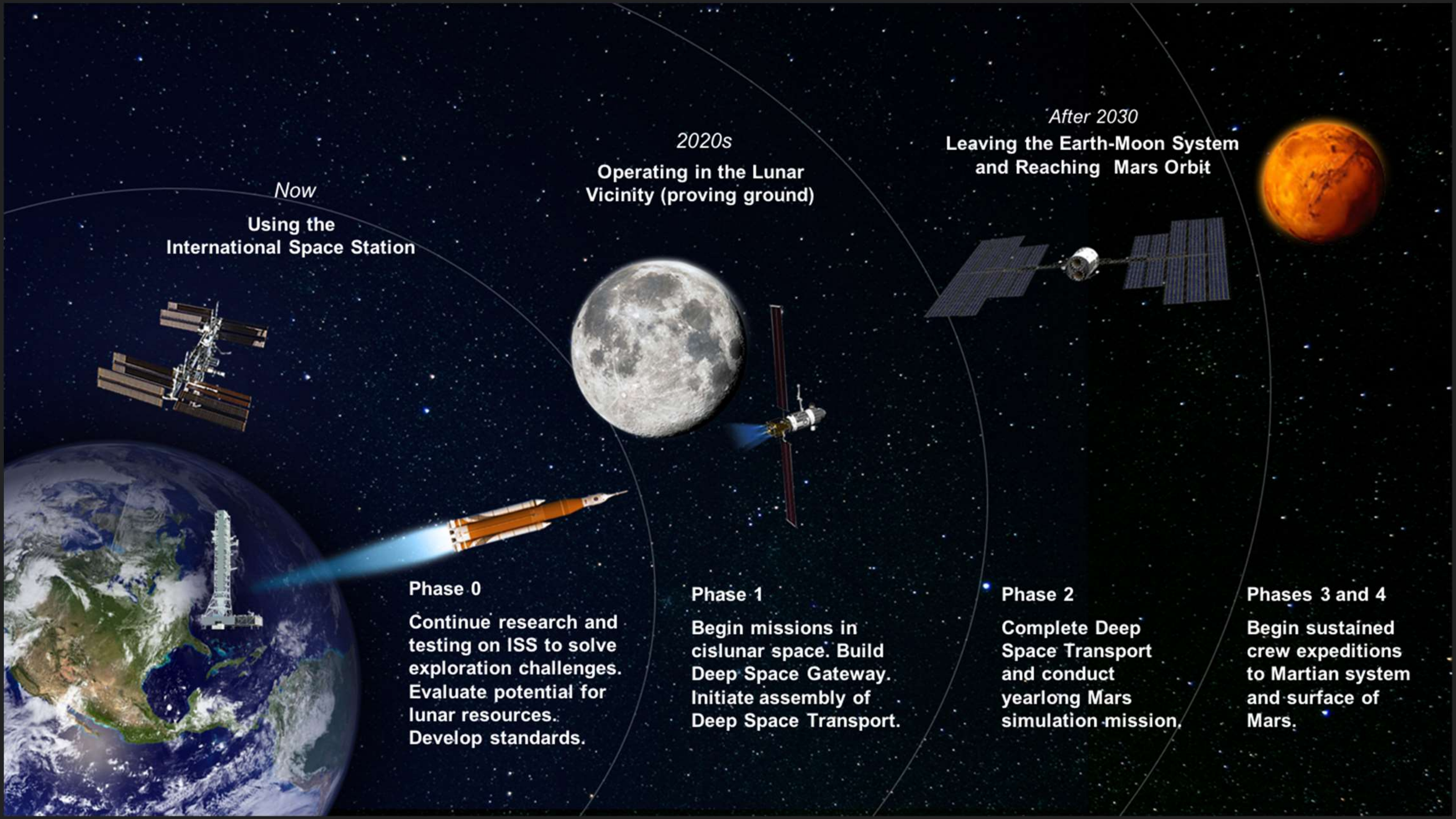
DUSTIN GOODING, NASA/JSC

# INTEGRATING ROS INTO NASA SPACE EXPLORATION

# INTEGRATING ROS INTO NASA SPACE EXPLORATION MISSIONS



*Now*

Using the International Space Station

*2020s*

Operating in the Lunar Vicinity (proving ground)

*After 2030*

Leaving the Earth-Moon System and Reaching Mars Orbit

**Phase 0**

Continue research and testing on ISS to solve exploration challenges. Evaluate potential for lunar resources. Develop standards.

**Phase 1**

Begin missions in cislunar space. Build Deep Space Gateway. Initiate assembly of Deep Space Transport.

**Phase 2**

Complete Deep Space Transport and conduct yearlong Mars simulation mission.

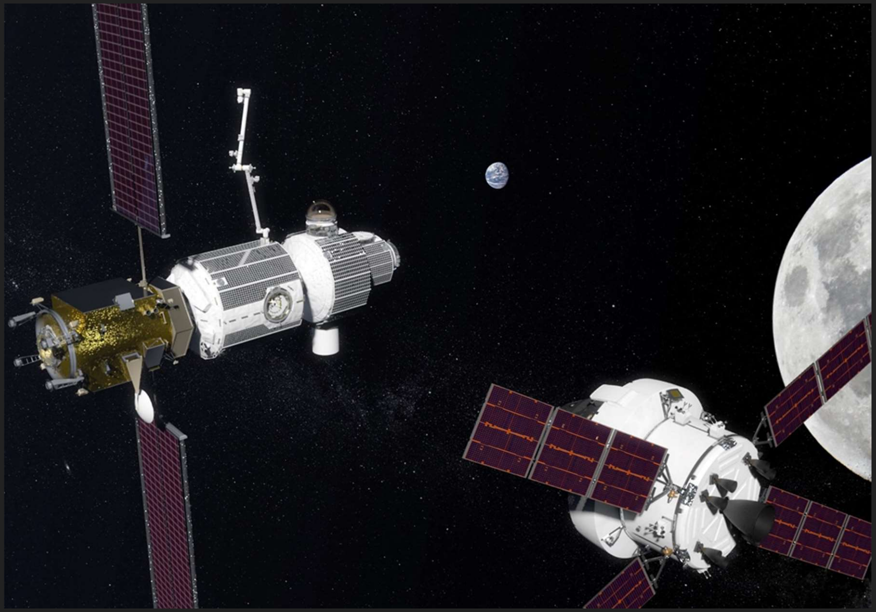
**Phases 3 and 4**

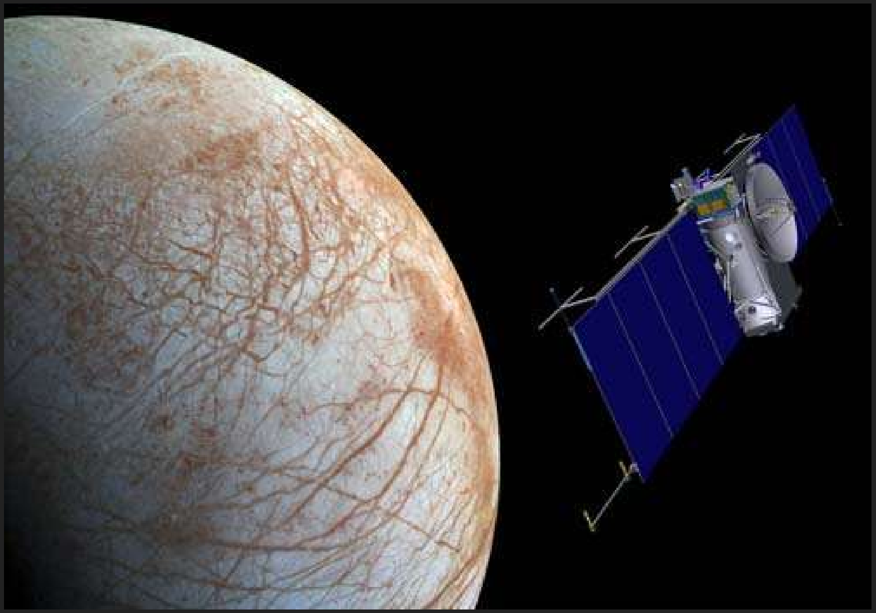
Begin sustained crew expeditions to Martian system and surface of Mars.

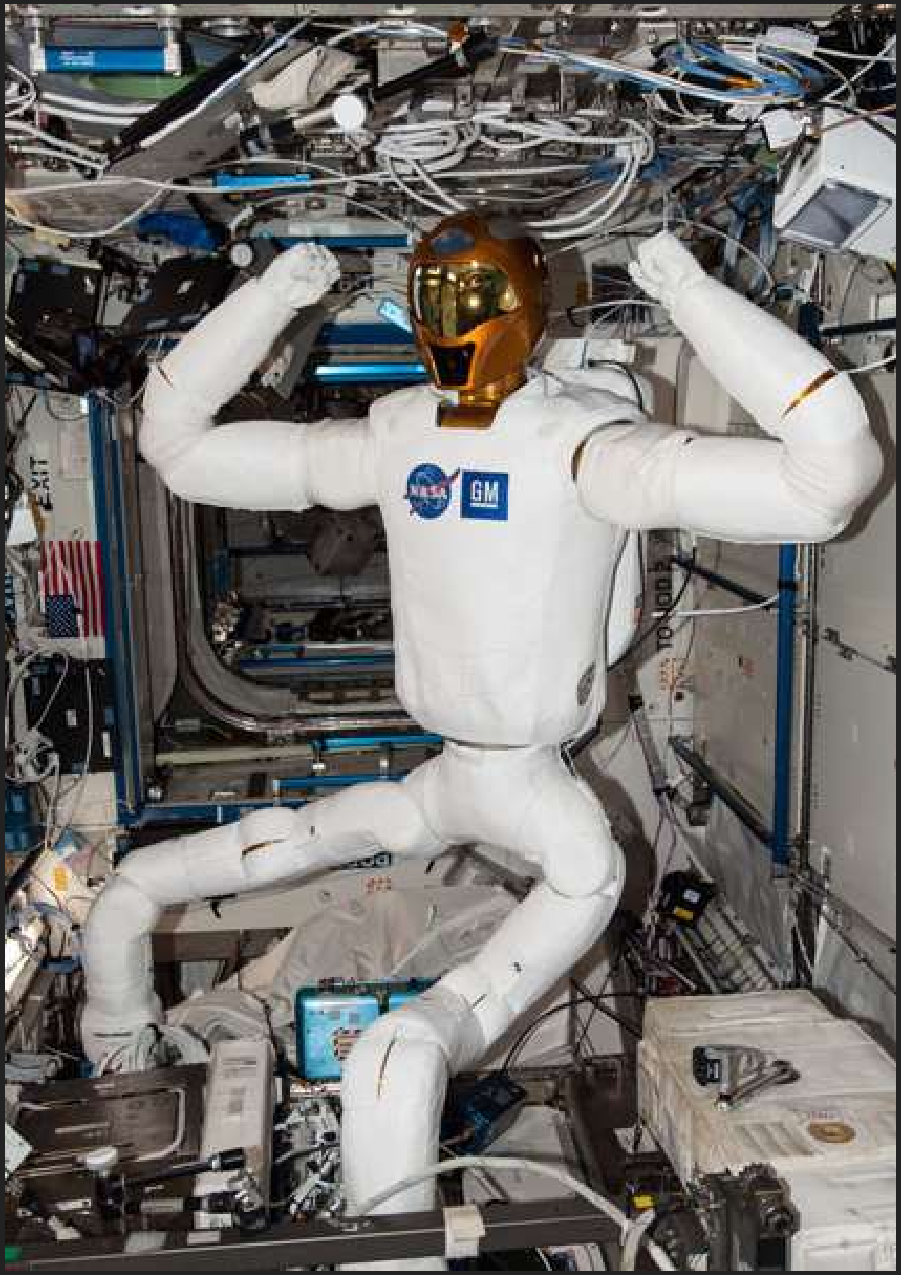
HTTPS://WWW.NASA.GOV/JOHNSON/EXPLORATION/DEEP-SPACE

# FUTURE EXPLORATION MISSIONS

- Lunar Proving Grounds
  - Deep Space Gateway
  - Surface Operations
- Mars and Other Worlds
  - In-Orbit Habitats
  - On-Surface Habitats
  - Science Scouting





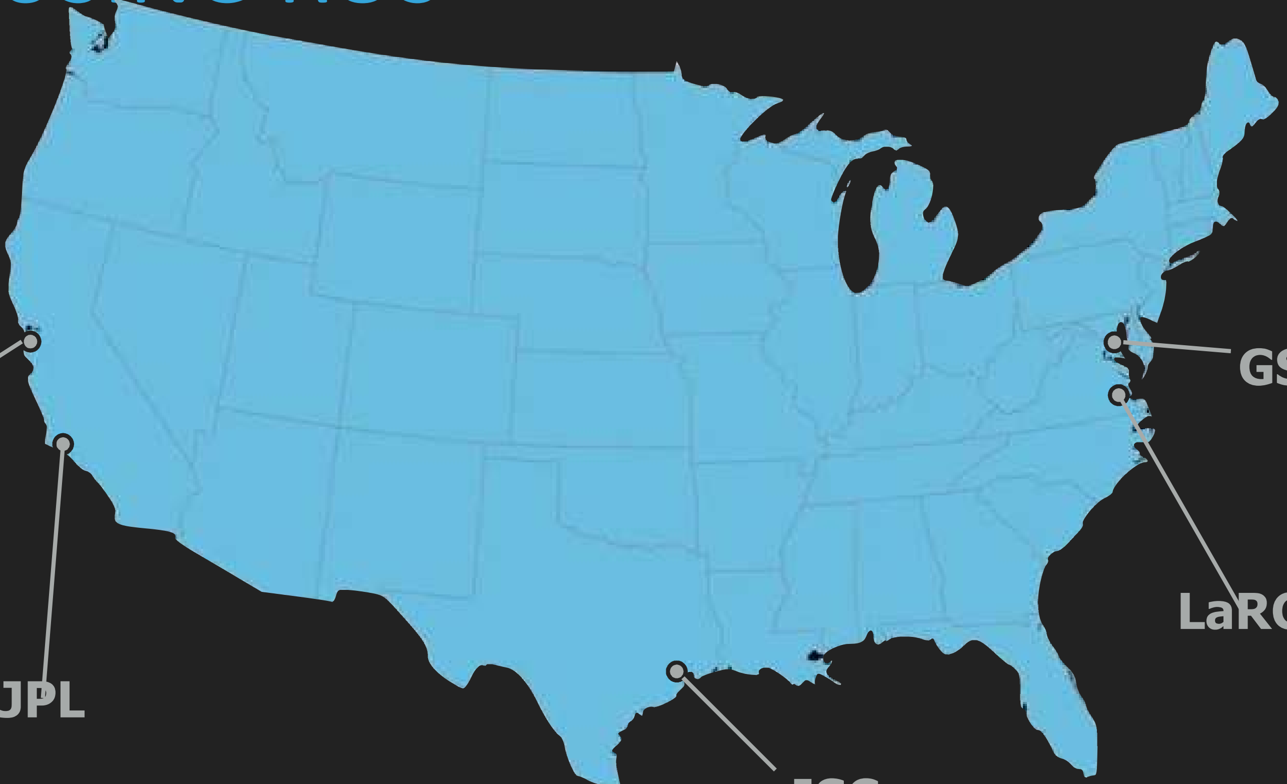


# ROBOTICS AND AUTONOMOUS SYSTEMS

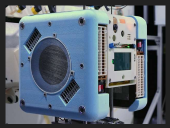
- As precursors to crewed missions
- As crew helpers in space
- As caretakers of assets sent ahead or left behind
- Capabilities will be extended in:
  - Sensing and Perception
  - Mobility
  - Manipulation
  - Human-System Interaction
  - System-Level Autonomy

HTTPS://WWW.NASA.GOV/JOHNSON/EXPLORATION/DEEP-SPACE

# CURRENT ROBOTICS AND AUTONOMOUS SYSTEMS USING ROS



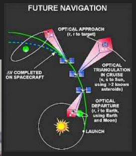
Astrobee



Dragonfly

RP

AutoNav

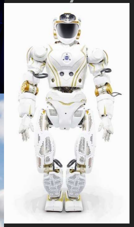

FUTURE NAVIGATION
OPTICAL APPROACH (r, ṙ to target)
ΔV COMPLETED ON SPACECRAFT
OPTICAL TRIANGULATION IN CRUISE (s, ṡ to Sun, using >2 known asteroids)
OPTICAL DEPARTURE (r, ṙ to Earth, using Earth and Moon)
LAUNCH

µAVs

Rovers

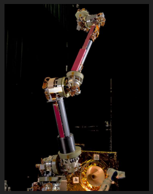
Robonaut2

Valkyrie



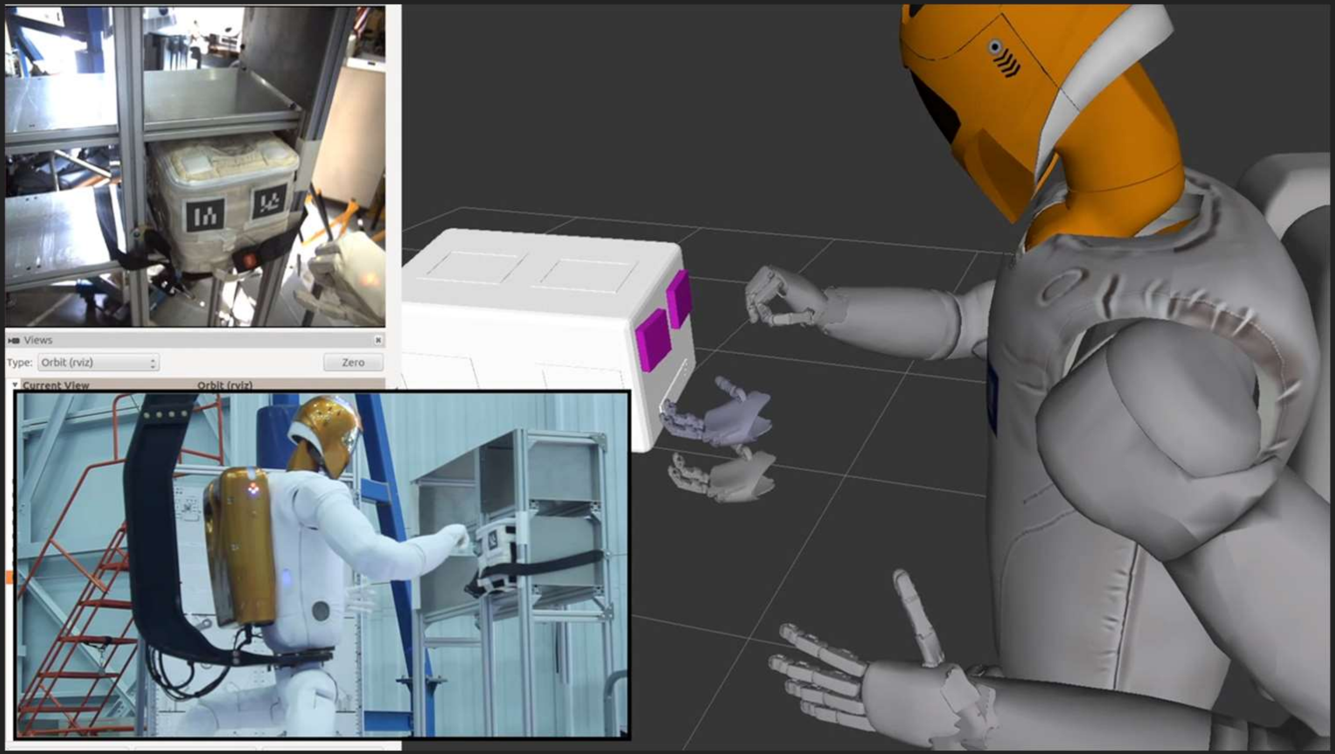
RP

ADP



TALISMAN

# HIGHLIGHT: AFFORDANCE TEMPLATES



- Originally developed to support *Valkyrie* during the DARPA Robotics Challenge.
- Improved in recent *Robonaut2* autonomous resupply demonstration.

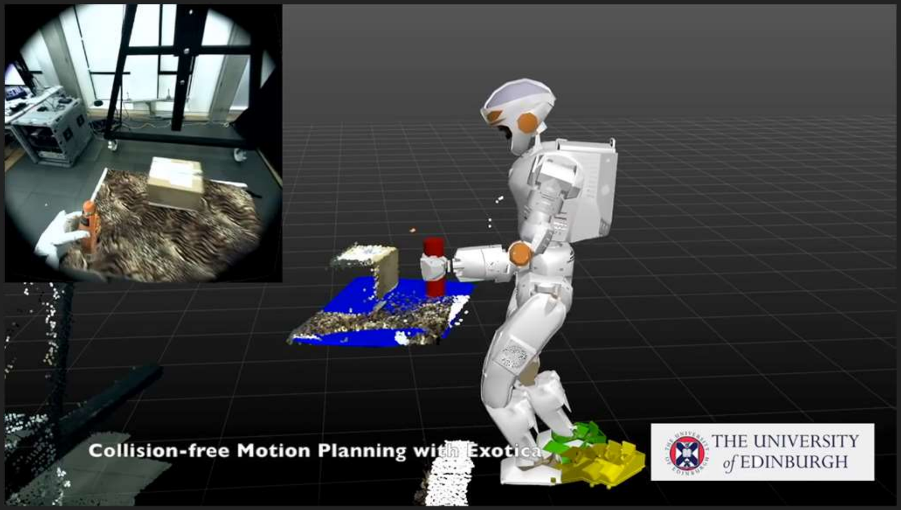
HTTP://IEEEXPLORE.IEEE.ORG/DOCUMENT/7140073 — HTTP://IEEEXPLORE.IEEE.ORG/DOCUMENT/8206229

# HIGHLIGHT: RESOURCE PROSPECTOR

- Joint project between ARC and JSC, developing future science rover technologies.
- Mobility and sensor development based on advanced Gazebo simulations.

# ROS INTEGRATION: ADVANTAGES

- Enables external organizations to collaborate more easily with NASA labs
  - ROS is easy to get, use, and it's free!
  - Students and researchers are already familiar with ROS
- Plenty of functionality right out of the box
  - "Core ROS" packages provide solutions to common problems
  - Large, active community of developers and researchers provide solutions for many areas of research
- Adding functionality is natural
  - ROS is designed to support adding functionality in a modular way
  - Projects can easily mix community and custom solutions, enabling rapid prototyping and development


Collision-free Motion Planning with Exotica
THE UNIVERSITY of EDINBURGH

HTTPS://WWW.YOUTUBE.COM/CHANNEL/UC0PRF6EBGFCFHCGGMUCG3RG

# ROS INTEGRATION: DISADVANTAGES

**NPR 7150.2B, Section 3.9**, regarding the use of COTS/GOTS/MOTS software.

a. The requirements to be met by the software component are identified.
b. The software component includes documentation to fulfill its intended purpose (e.g., usage instructions).
c. Proprietary rights, usage rights, ownership, warranty, licensing rights, and transfer rights have been addressed.
d. Future support for the software product is planned and adequate for project needs.
e. The software component is verified and validated to the same level required to accept a similar developed software component for its intended use.
f. The project has a plan to perform periodic assessments of vendor reported defects to ensure the defects do not impact the selected software components.

- NASA software development requirements can make using ROS on safety-critical systems a challenge.
  - NPR 7150.2B and NASA-GB-8719.13
- Some products in “Core ROS” simply don’t meet the needs of the flight community.
  - No reliable communication over degraded networks.
  - Real-time determinism is questionable.
  - Limited or no support for embedded hardware and operating systems.
- Some existing domain solutions aren’t easily portable to ROS’ way of doing business.

HTTPS://NODIS3.GSFC.NASA.GOV/DISPLAYDIR.CFM?INTERNAL_ID=N_PR_7150_002B_

# ADDRESSING THE “FLIGHT SYSTEM” CONCERNS

- The “up and coming” flight software system at NASA is the in-house and open-source *Core Flight Software*.
  - Provides an environment for real-time execution of C and C++ code, plus some common utilities.
- Prototypes in work to get CFS and ROS to share data.
  - Initial success with the “bridge” approach. Next up is native support for ROS comm within CFS.

HTTPS://GITHUB.COM/NASA/CFE

# THANK YOU