RLSO2 Considerations

What has **not** changed since the original study?

- **"Similar" computing** (RAD750) and **sensors** (cameras) – still waiting for HPSC and lidar for rovers
- Have only performed field geology robotically in space – **no experience with robotic "civil engineering"** in space
- Have only operated planetary rovers on Mars – **no experience with rovers on the Moon**
- Have not maintained a human spacecraft **without** humans aboard – unclear what autonomous systems (including robots), architecture, etc. are needed
- Very limited use of autonomy in space – due to limited need (to date), mission risk posture, etc.

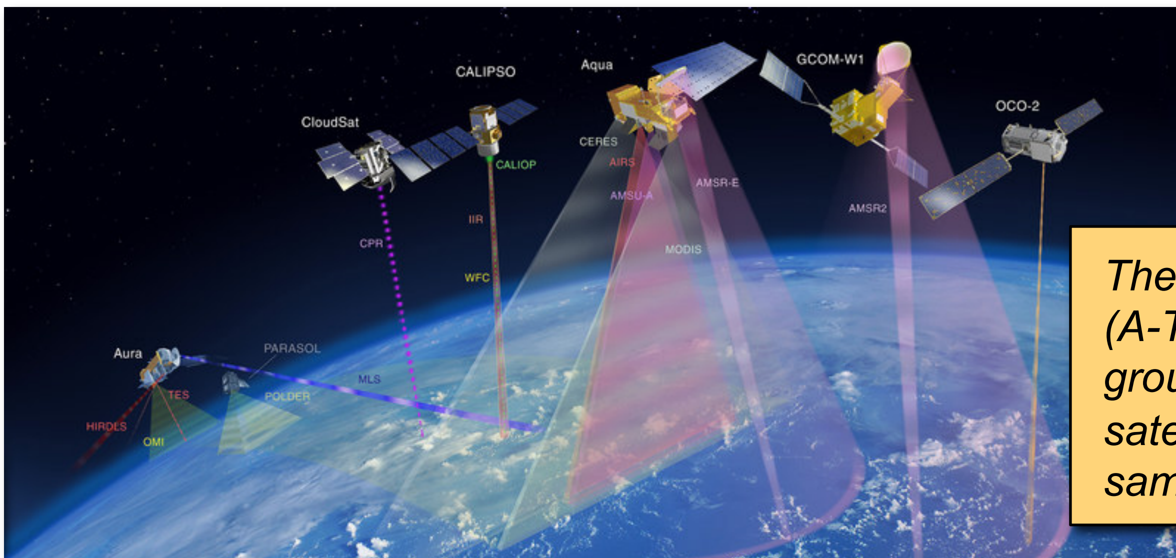
RLSO2 Considerations

What has changed in robotics since the original study?

- **Terrestrial robotics**: significant fielded advances in robot competency / capabilities due to algorithms, software, computing, and sensors.
- **NASA R&D:** P2P-HRI project, Desert RATS, Human-Robotic Systems project, Human Exploration Telerobotics project, PSTAR projects
- **MER** (landed January 2004): good understanding of how to keep rovers functioning **on Mars**, tactical ops process, multiple software uploads, limited autonomy
- **MSL** (landed August 2012): experience with a “large” rover
- **Robonaut 2** (launched February 2011): dexterous manipulation tests in microgravity (within ISS IVA environment) both teleop and supervised
- **Resource Prospector rover**: design / risk reduction for lunar polar operation

Automation

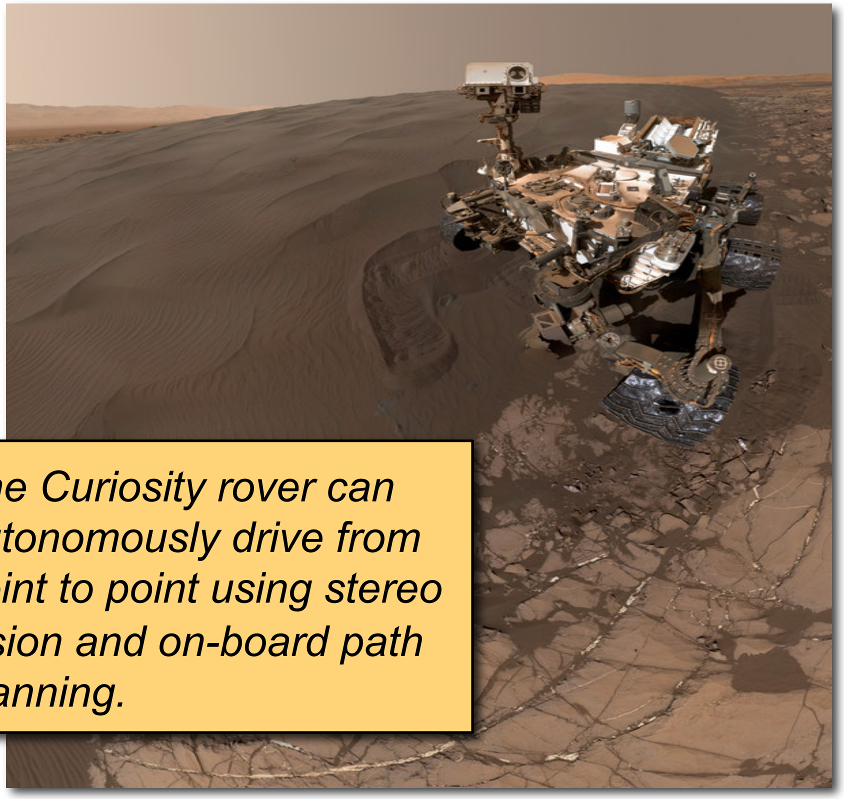
- **Automation is the automatically-controlled operation of an apparatus, process, or system by mechanical or electronic devices that take the place of human labor**
  – *Merriam-Webster*
- Automation is not “self-directed”, but instead requires control (e.g., command sequence)



The “Afternoon Train” (A-Train) is a coordinated group of Earth observing satellites that follows the same orbital “track”.

Autonomy

- **Autonomy is the ability of a system to achieve goals while operating independently of external control.**
  – *2015 NASA Technology Roadmaps*
    - Requires **self-directedness** (to achieve goals)
    - Requires **self-sufficiency** (to operate independently)
- A **system** is the combination of elements that function together to produce the capability required to meet a need. The elements include all hardware, software, equipment, facilities, personnel, processes, and procedures needed for this purpose
  – *2016 NASA System Eng. Handbook*

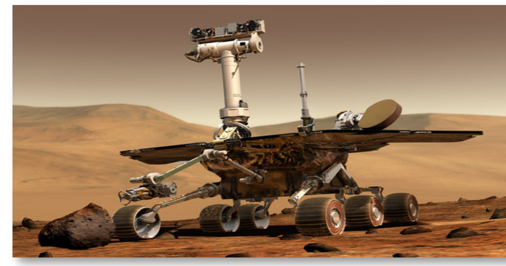


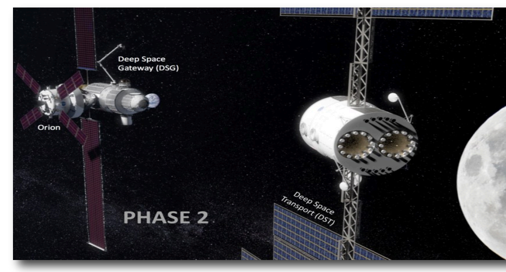
The Curiosity rover can autonomously drive from point to point using stereo vision and on-board path planning.

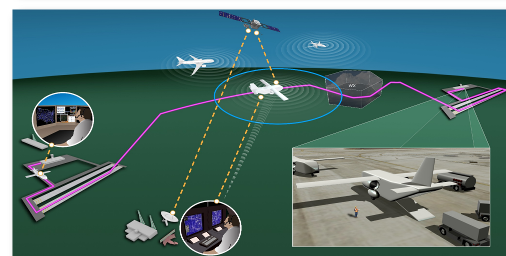
Why autonomy?

Autonomy is needed …

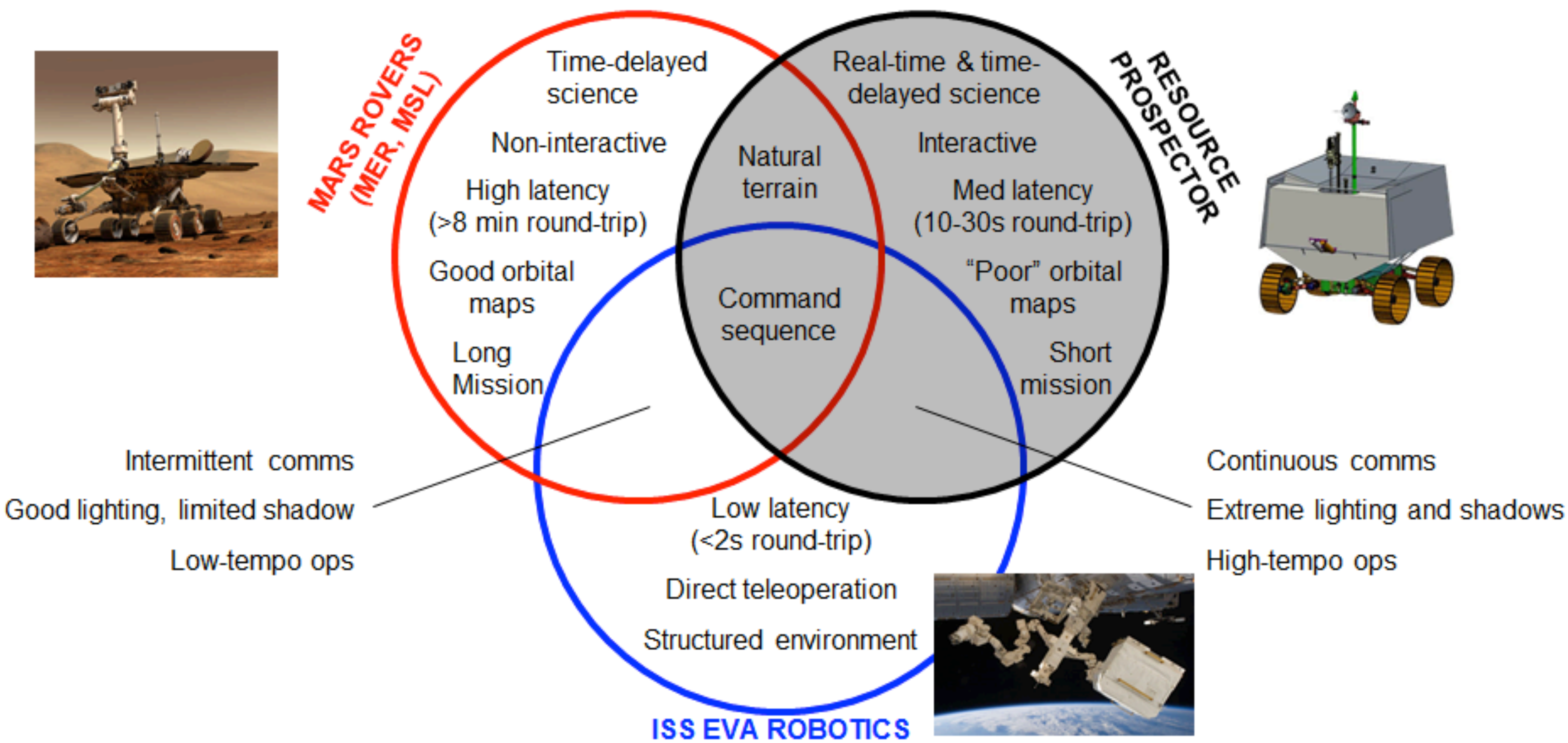
- When the cadence of decision making **exceeds communication constraints** (delays, bandwidth, and/or communication windows)
- When **time-critical decisions** (control, health, life-support, etc) must be made on-board the system, vehicle, etc.
- When decisions can be better made using **rich on-board data** compared to limited downlinked data
- When local decisions **improve robustness** and **reduce complexity** of system architecture
- When autonomous decision making can re**duce system cost** or im**prove performance**
- When **manual control is unacceptable** (due to variability in training, proficiency, etc.)



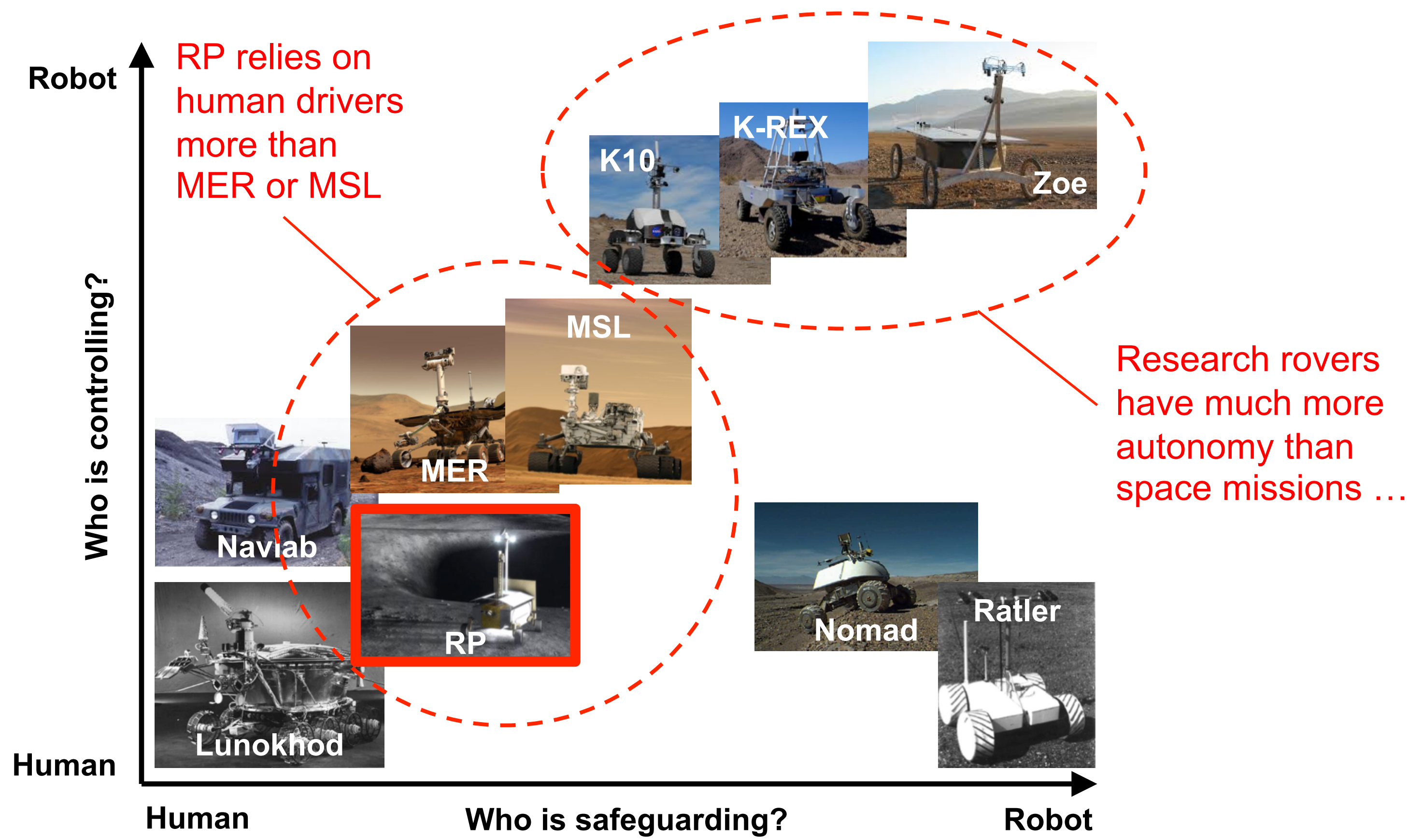




RP vs. ISS vs. MER/MSL

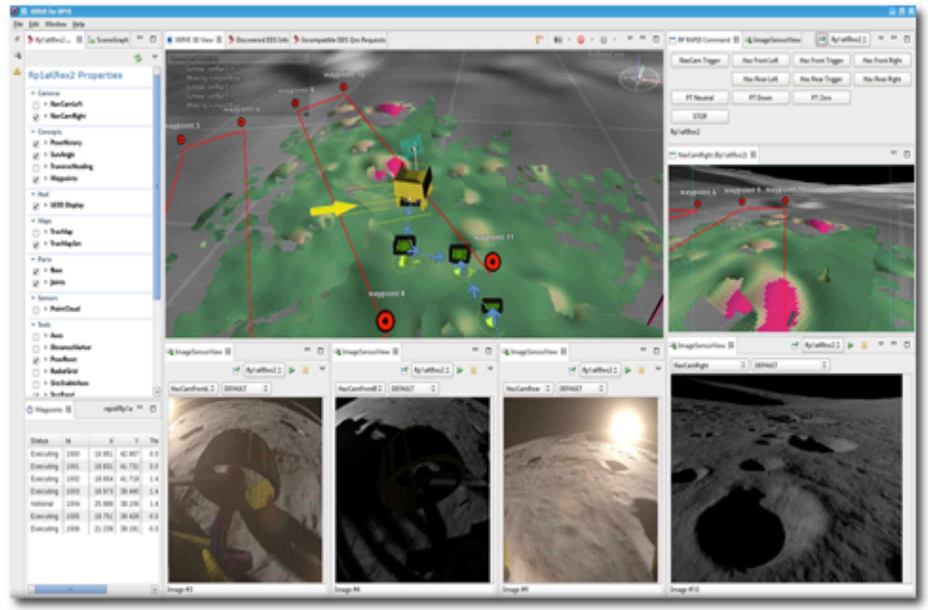


Robot Operations

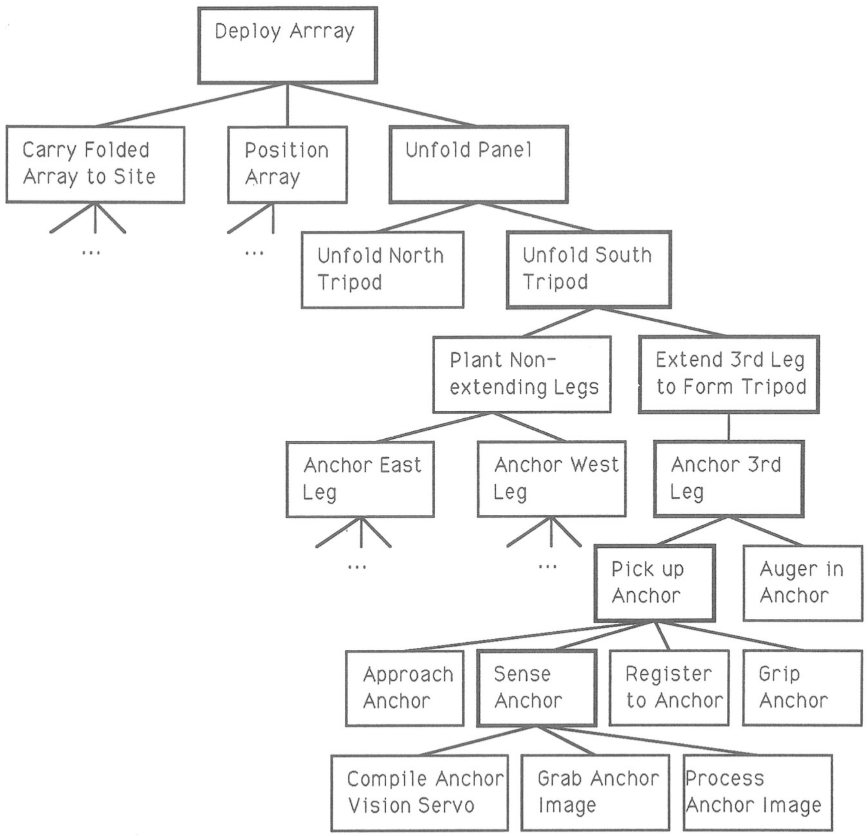


RP Rover Operations

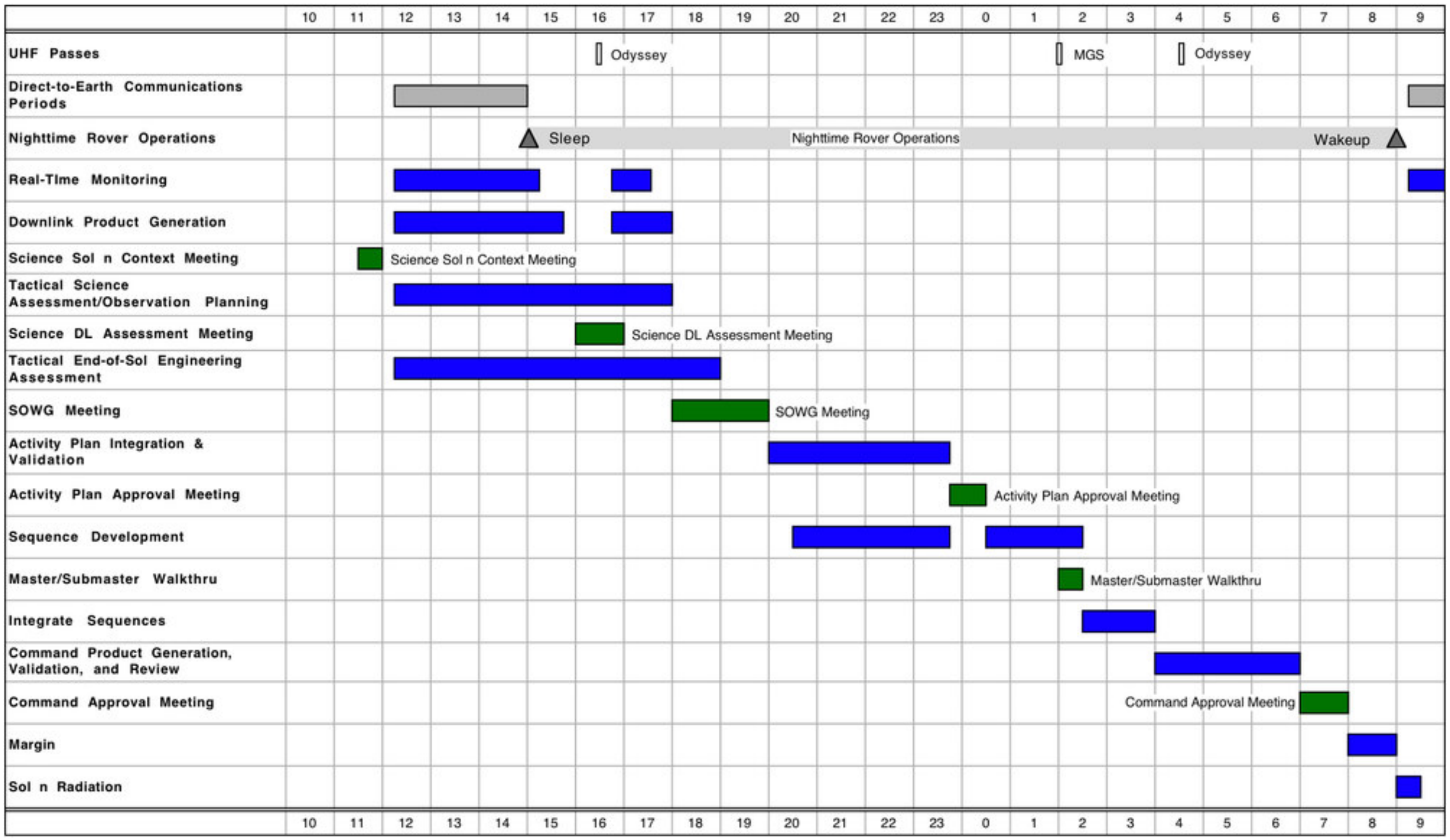
- Direct-to-Earth communications
  - High availability during entire mission
  - 10 sec (one-way latency)
  - 236 kbps downlink (all telemetry, not just rover)
- Driving is interactive, not sequenced
- Short mission, high tempo ops
- Science is involved in driving ops
- Challenging lighting environment
- Uncertain terrain
  - Unknown hazard density
  - Unknown hazard location
- Semi-automatic navigation
  - Automatic (stereo process) and manual hazard assessment
  - Map-matching + limited on-board sensors for localization



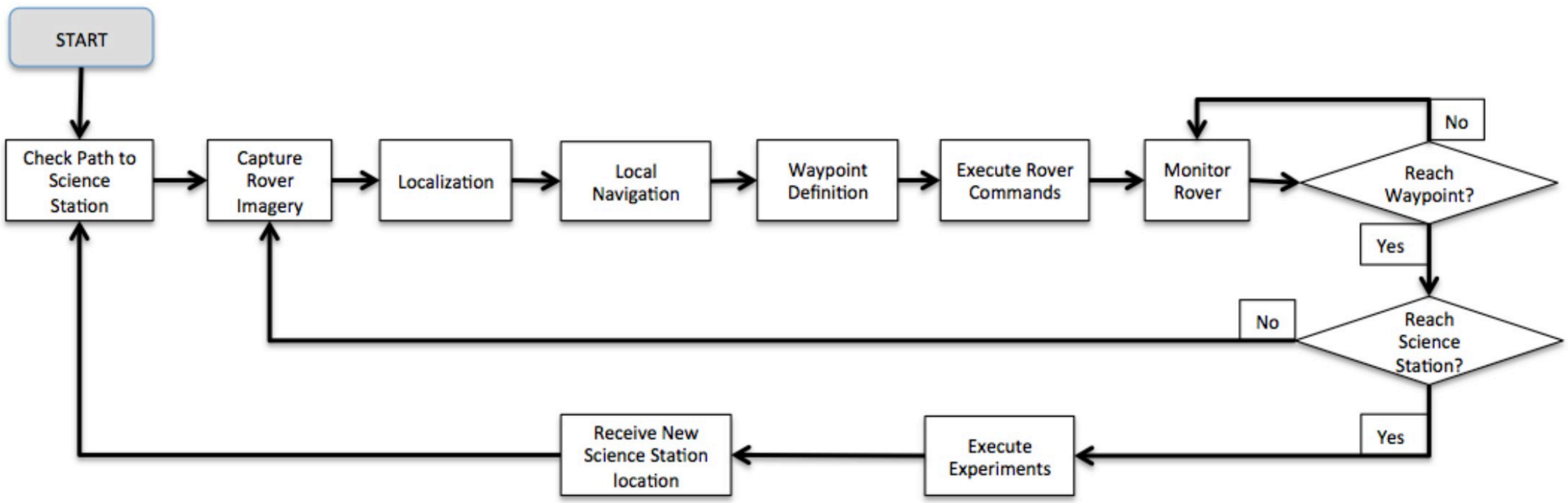
RLSO1 Supervisory Control



MER Tactical Ops Process (2004)

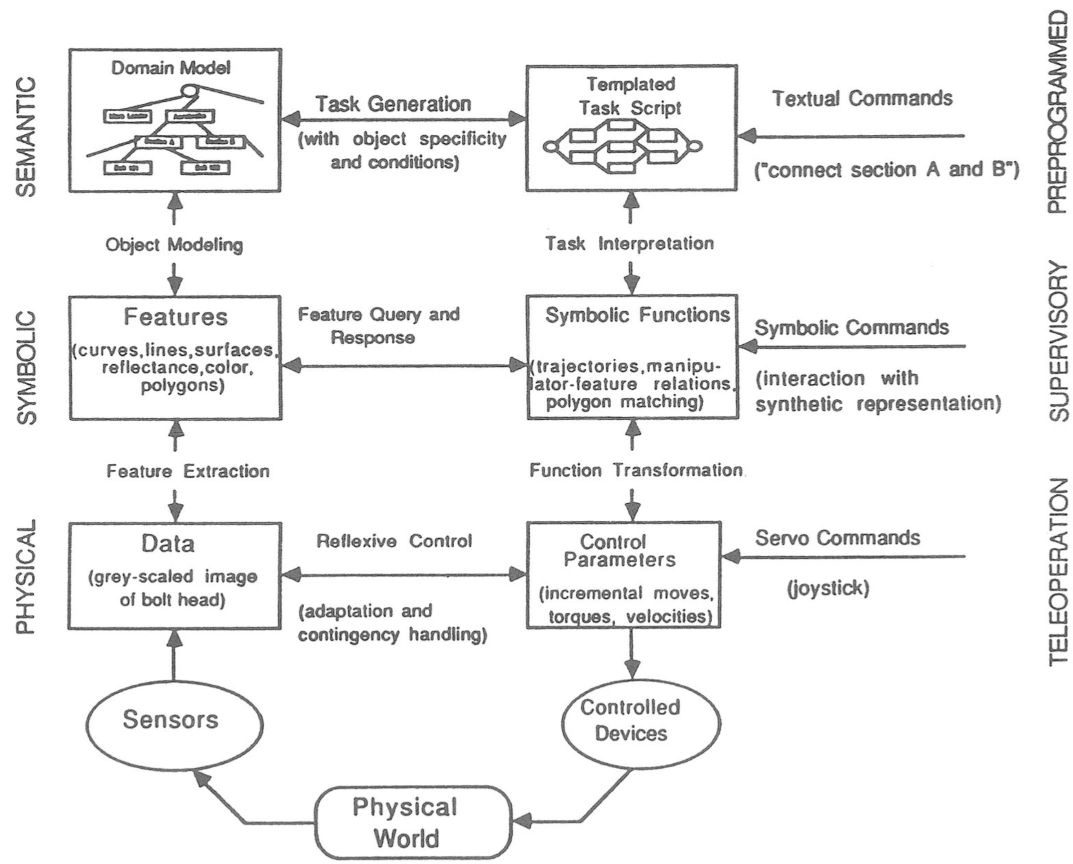


RP Remote Driving Process



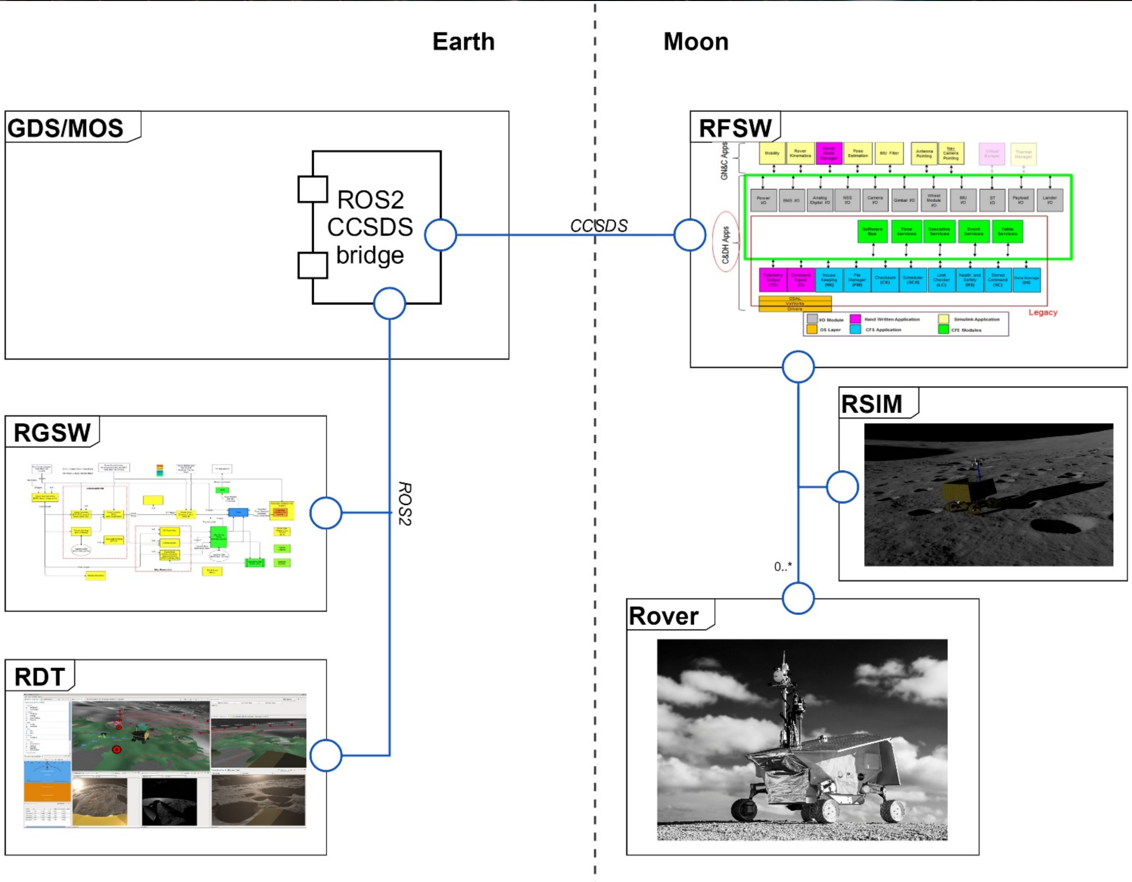
RLSO1 Robot Software Architecture

- Hierarchical control architecture
  - Safeguarded teleop (bottom)
  - Supervisory control (middle)
  - Automated task control (top)
- Assumes fully-specified domain and task model
- Inspired by NASREM (NIST), TCA (CMU), etc. architecture concepts from the late 1980s



RP Rover Software Architecture

- Rover software components split between on-board and off-board (ground) components
- Simulation (including processor-in-the-loop) for testing, validation, & training purposes



RLSO2 Recommendations

- A few key points from the original study
  - Emplace a local positioning system – **but do this for both IVA and EVA use**
  - Robot(s) should have some amount of on-board safeguarding
  - Primary mode of operations should be supervisory control – **but take RP approach (few commands with interactive monitoring)**
  - Assume that humans will need to intervene (crew/local and ground control/remote) – **so design the system to both support and take advantage of this**
- Design principles
  - Follow Resource Prospector approach – computing, sensors, & conops
  - Deploy "high performance" software modules on the ground
  - Consider tasks that have to be performed during **both** base build up and ops (sustaining) phases
  - Support both IVA and EVA tasks
  - Robot software/ops architecture is not standalone – **robot + base should be designed together**