Case Studies in Verifying Spacecraft Autonomy

Lorraine E. Prokop & Daniel J. Dorney: NASA – Langley Research Center

Martin S. Feather: Jet Propulsion Laboratory, California Institute of Technology

Stephen B. Johnson: NASA Marshall Space Flight Center, Jacobs Space Exploration Group

The Authors

Dr. Stephen Johnson
stephen.b.johnson@nasa.gov

Ascent Phase –
Space Launch System
(SLS) Fault Management

Dr. Daniel Dorney
daniel.j.dorney@nasa.gov

Ascent Phase –
SpaceX Autonomous
Flight Termination System
(AFTS)

Dr. Lorraine Prokop
lorraine.e.prokop@nasa.gov

Transit Phase –
Orion Optical Navigation
(OpNav)

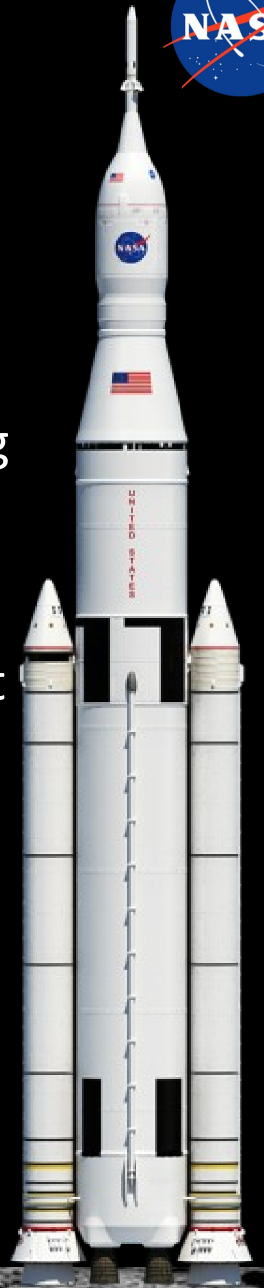
Dr. Martin Feather
martin.s.feather@jpl.nasa.gov

Descent/Landing Phase –
MER-DIMES
Science Phase –
Curiosity/Opportunity AEGIS

This research was carried out at NASA - Langley Research Center, Jacobs Space Exploration, and the Jet Propulsion Laboratory, California Institute of Technology under a contract with the National Aeronautics and Space Administration.

Ascent - Human-Rated Launcher Fault Management (FM): NASA Space Launch System (SLS)

SLS – Key Autonomy and FM Requirements



- NASA’s next-generation human-rated deep space launch vehicle
  - Intended to boost humans and cargo beyond Earth orbit
- SLS autonomy needed for nominal flight sequencing and control, and for FM due to very fast times-to-criticality
  - Rapid failure propagation - engine combustion, thrust vector control, propellant mixing
- Key Autonomy & Fault Management requirements derived from NASA NPR 8705.2C Human Rating Requirements for Space Systems
  - Autonomous operation for functions, which if lost, would result in a catastrophic event
  - Isolate and recover from faults … that would result in a catastrophic event
  - Failure tolerance to catastrophic events, with levels of failure tolerance and implementation (similar or dissimilar redundancy) derived from integration of the design and safety analysis
  - Abort capability from the launch pad to Earth-orbit insertion to protect for the following ascent failure scenarios (minimum list): a. Complete loss of ascent thrust propulsion; b. Loss of attitude or flight path control

Autonomy Verification Strategy

- Fault Management Architecture is “monitor-respond”
- NASA process uses “Detailed Verification Objectives” (DVOs) for top-level requirements
- Nominal autonomy verified by Systems Integration Laboratory (SIL) test for nominal flight operations.
- Fault Management verification is much harder due to the large number of algorithms (>200) and very large number of failure scenarios (>1,600) and modes (> 30,000)
  - Each algorithm is executed for risk reduction via State Analysis Model (SAM) execution (state machine) and physics-based simulation in Vehicle Management End-to-End Testbed (VMET), and formally tested in the Software Development Facility (SDF) and SIL.
  - The need for algorithms (which ones are valuable and needed), and their performance is verified by analysis. Analysis performed via failure scenarios.
    - Quantitative estimates of the value of abort algorithms against the metric of “Loss of Crew Benefit” (Loss of Crew avoided due to existence of the algorithm), compared to the risks of those algorithms activating when they should not (false positives, causes Loss of Mission). Estimated per scenario and then results “slice and diced” to get the total estimates per algorithm.
    - Qualitative judgments of algorithm need & value via task teams that include SLS engineering, Systems Engineering & Integration, Mission Operations and Crew Office (both part of Flight Operations Directorate).

Ascent - Launch Termination : SpaceX Falcon 9

Automated Flight Termination System (AFTS)

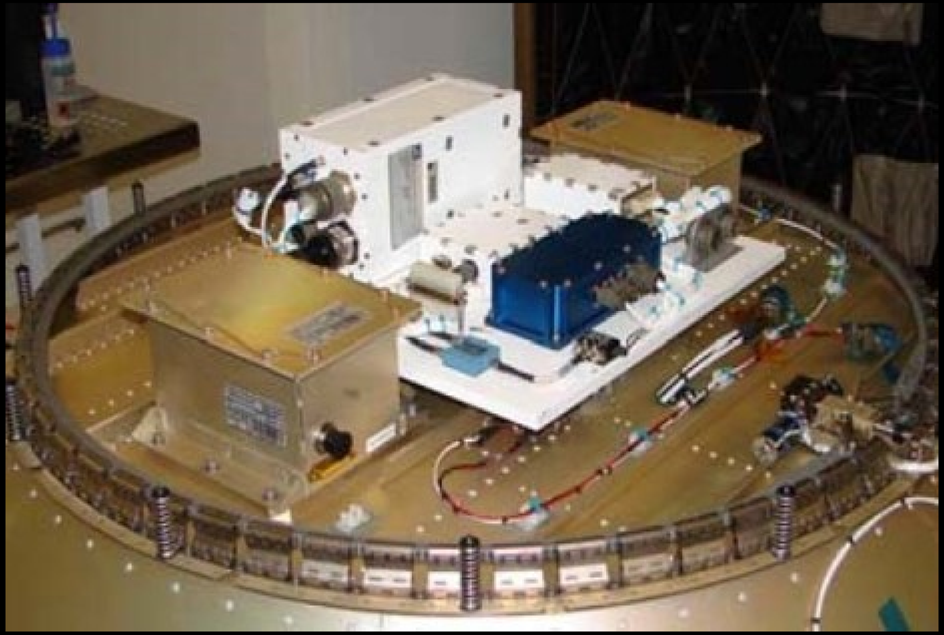
- The launching of flight vehicles is a complex and energetic process necessitating many safety precautions and protocols
- The time-to-criticality can be below human reaction time, necessitating the need for increased flight system autonomy
- Active monitoring , historically provided by range-safety personnel, is becoming more automated
- The Falcon 9 originally used a traditional flight termination system but transitioned to an AFTS to enable the simultaneous return of multiple boosters and to support crewed launches

AFTS Development

- The AFTS system uses a core software package developed by NASA, the Air Force and DARPA ▭ with a customized software wrapper added by SpaceX
- AFTS design features include (not limited to):
  - Complete functional independence from the launch vehicle
  - Reliability of 99.9% with a 95% confidence level
  - Single fault tolerance with regard to performance degradation during flight and inadvertent initiation of the termination command
  - Capability to issue telemetry throughout all mission phases by land-based connections or by interfacing with a space based telemetry system
- AFTS can use multiple sensor types to provide position, velocity, attitude, and other state vector data ▭ inertial navigation system (INS), global positioning system (GPS), inertial measurement units (IMUs), accelerometers, etc.
- Configurable rule-based algorithms make the flight termination decisions
- Mission rules are developed using the inventory of rule types from current human-in-the-loop operational flight safety practices

AFTS Hardware and Applications

- The AFTS uses redundant chassis (each with redundant sensor inputs), the Core Autonomous Safety Software (CASS), and a launch vehicle specific software wrapper ▭ either chassis can initiate flight termination
- AFTS built and tested to NASA design and construction standards
    - Hazard report process to identify catastrophic failure modes for crewed flights
    - Tested on sounding rockets and operated in shadow mode on the Falcon 9 for thirteen flights
- AFTS first flown as primary flight termination system for the Falcon 9 on 2/19/17 and used on all Falcon 9/Falcon Heavy launches since
- SpaceX estimates that AFTS provides a 50% reduction in range-related expenses

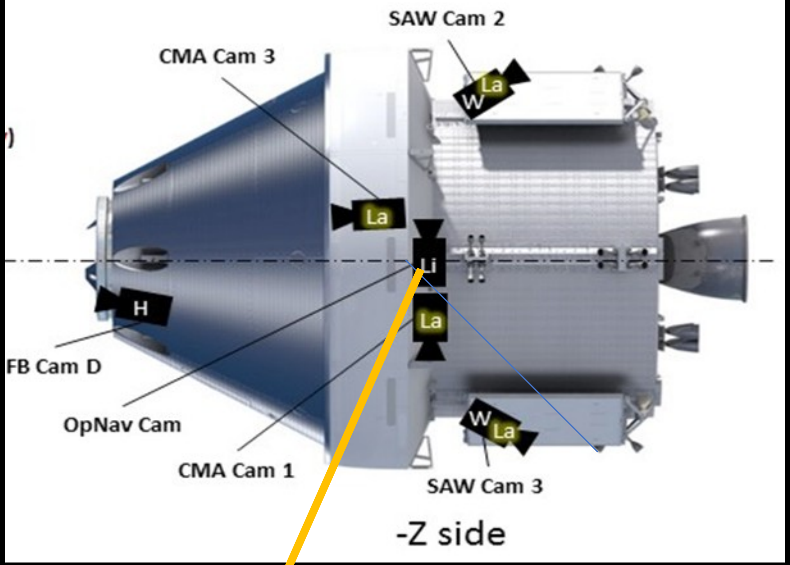


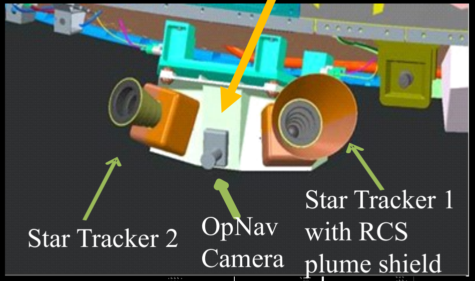
Early Test Hardware on SpaceX Falcon 1

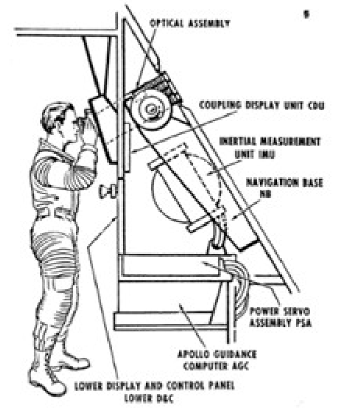
https://ses.gsfc.nasa.gov/ses_data_2011/110913_Esper.pdf

Transit – Orion Optical Navigation

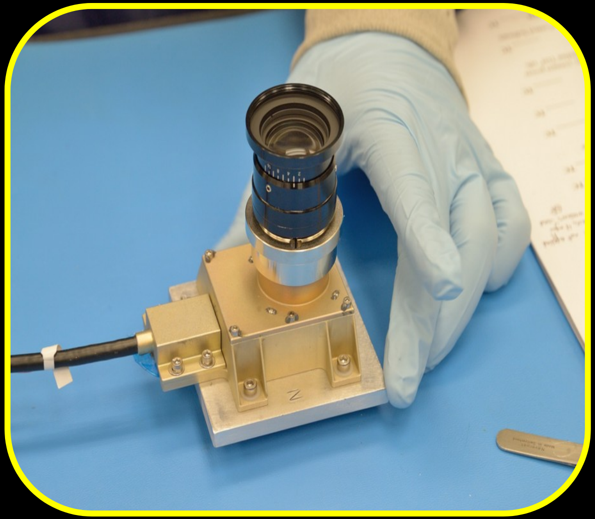
Orion Optical Navigation (OpNav)





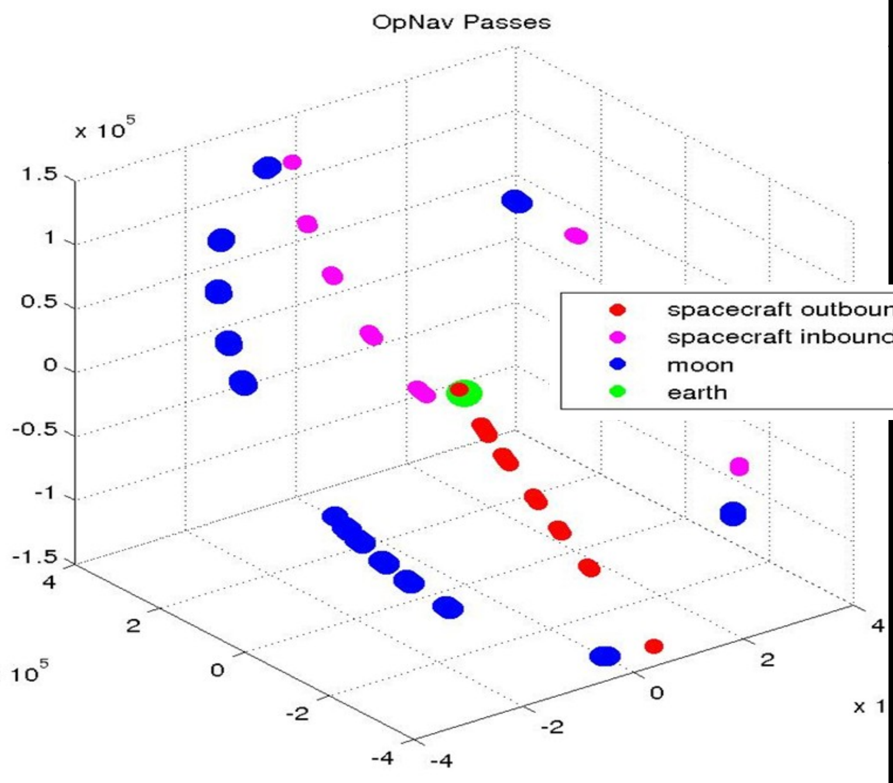

OPTICAL ASSEMBLY
COUPLING DISPLAY UNIT CDU
INERTIAL MEASUREMENT UNIT IMU
NAVIGATION BASE NB
POWER SERVO ASSEMBLY PSA
APOLLO GUIDANCE COMPUTER AGC
LOWER DISPLAY AND CONTROL PANEL LOWER D&C

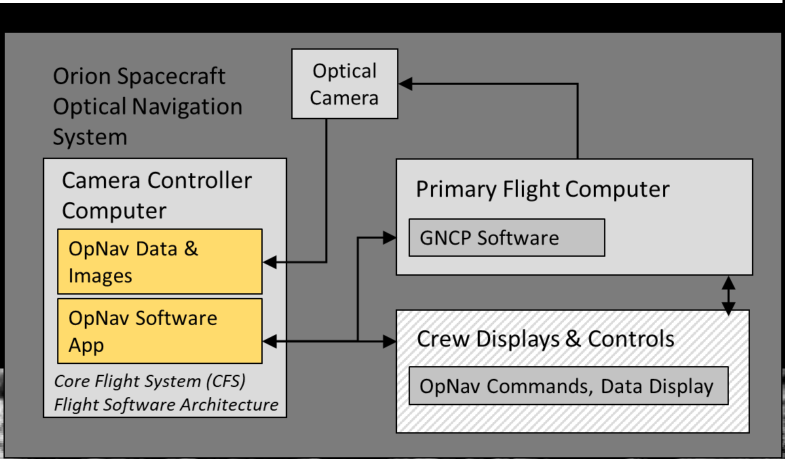
LOCATION OF GUIDANCE AND NAVIGATION EQUIPMENT IN SPACECRAFT GN-9005



Optical Images processed for autonomous navigation under communications loss – safety critical

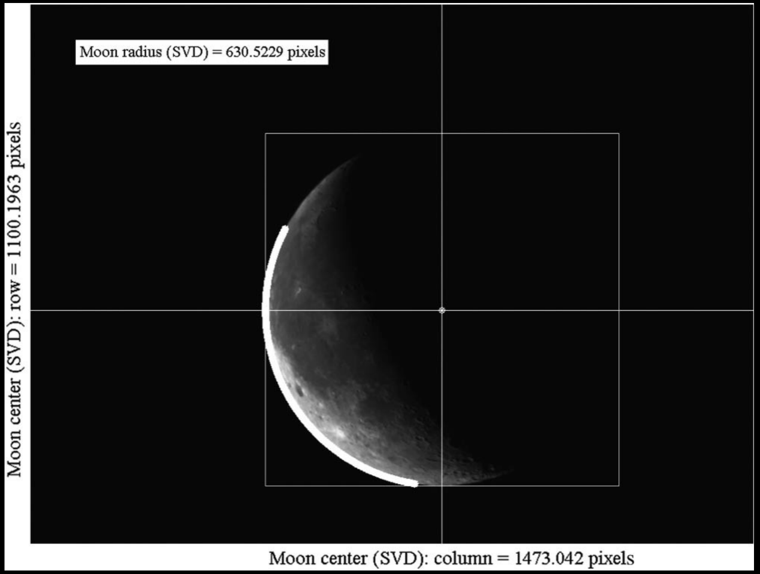
Similar role as Apollo Space Sextant



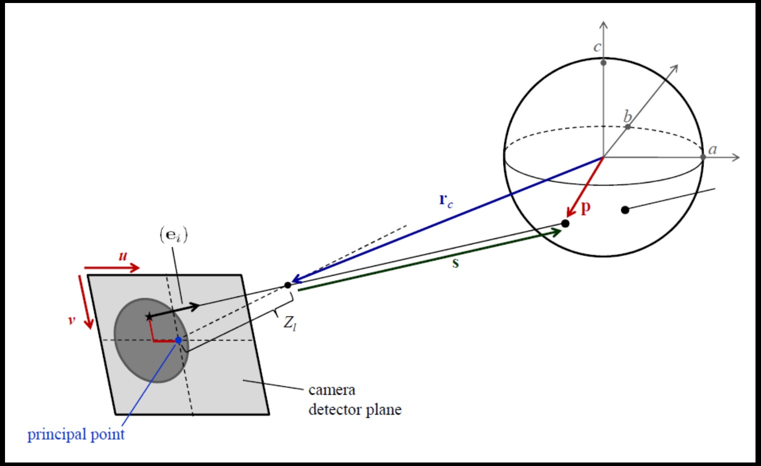

Orion Spacecraft Optical Navigation System
Optical Camera
Camera Controller Computer
OpNav Data & Images
OpNav Software App
Core Flight System (CFS) Flight Software Architecture
Primary Flight Computer
GNCP Software
Crew Displays & Controls
OpNav Commands, Data Display

OpNav Software Basics

- Images processed onboard in real time during “opnav passes”
- Still images of *Moon or Earth* -> apparent angular diameter/centroid -> range/bearing
  - Artemis 1 – range/bearing with star tracker attitude
  - Artemis 2 – self attitude determination from star centroiding
- Calibrates and undistorts images in real-time based on star centroids
- Class A Safety-Critical Software, developed according to the following process standards:
  - NPR 7150.2B
  - CMMI Level 3



Moon radius (SVD) = 630.5229 pixels
Moon center (SVD): row = 1100.1963 pixels
Moon center (SVD): column = 1473.042 pixels



OpNav Software Test Campaign

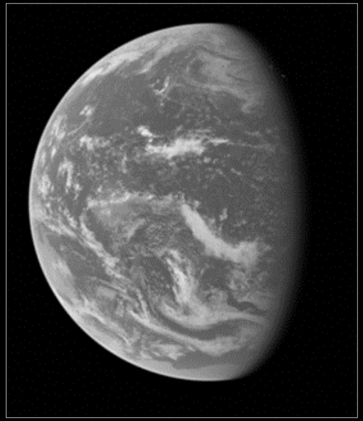
- Key performance requirements on accuracy of calibration, undistortion, range/bearing, and attitude drove test success criteria
  - No output data allowed out of spec
- Test campaign consisted of the following
  - Lifecycle reviews
  - Unit testing (100% Multiple Condition Coverage – MCC)
  - Code inspections & static code analysis
  - Verification testing – synthetic imagery
    - Verification test peer reviews
    - Extensive test cases
  - Off-nominal and robustness testing
  - Continuous integration & regression
  - Integration testing – higher fidelity labs
  - Validation – final validation with actual imagery in-flight during outbound leg of Artemis I

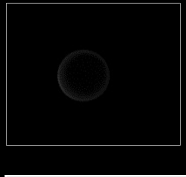
Item	Quantification
Flight Software Size (SLOC)	8,800
Test Software Size (SLOC)	35,400
Individual Pass/Fail Test Cases (Total)	21,322
Unit Test Cases	2,769
Verification Test Cases (Nominal)	16,618
Verification Test Cases (Off Nominal)	1,935
Nominal Synthetic Test Images	3,382
Off Nominal Test Images	2,299

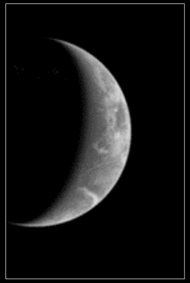
Review Type / Product	Major Issues	Minor Issues
Software Development Plan (SDP)	1	6
Software Requirements Specification (SRS)	4	21
Critical Design Review (CDR)	7	11
Verification Test Review (VTR)	0	7
Test Readiness Review (TRR)	0	0
Code Inspections	49	305

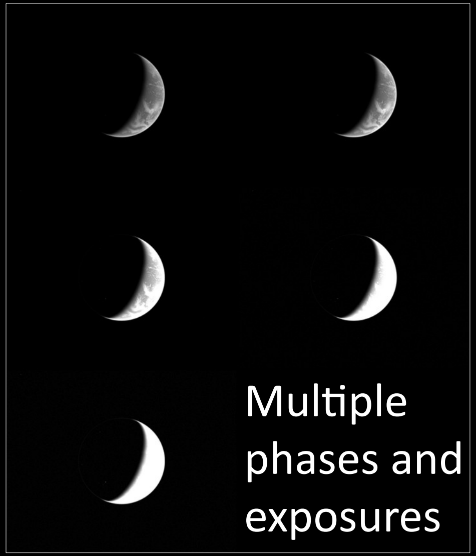
Sample OpNav Synthetic Test Imagery

Nominal Images expected during mission timeline

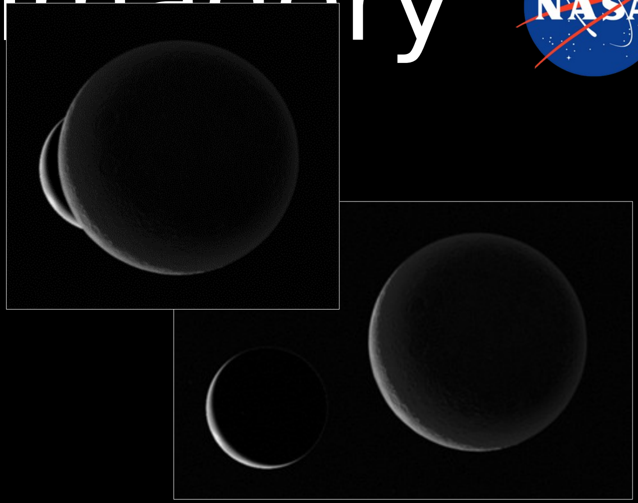


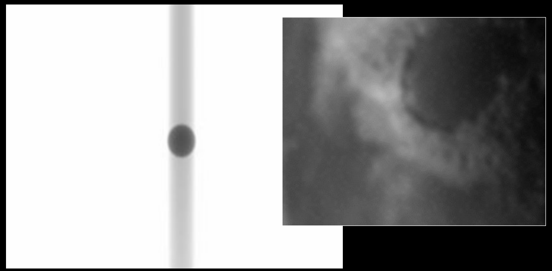




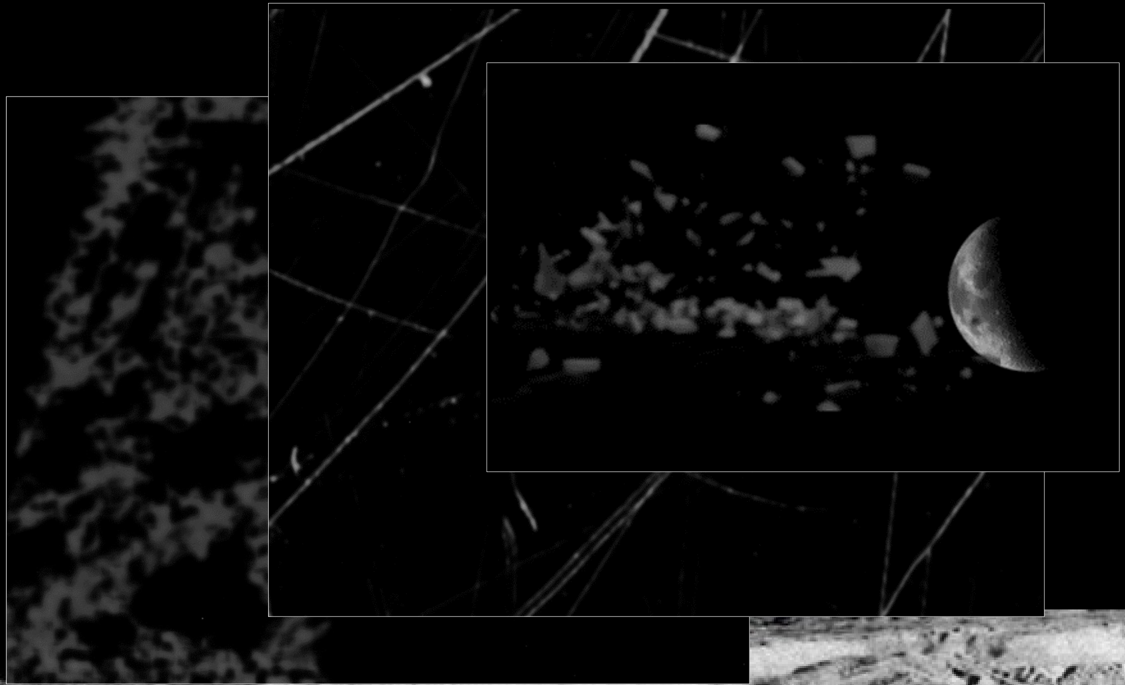


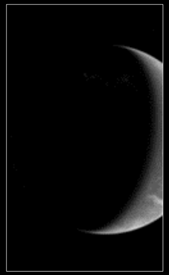
Multiple phases and exposures

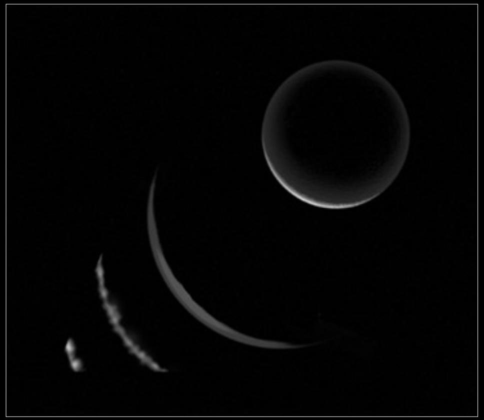




Off Nominal Images

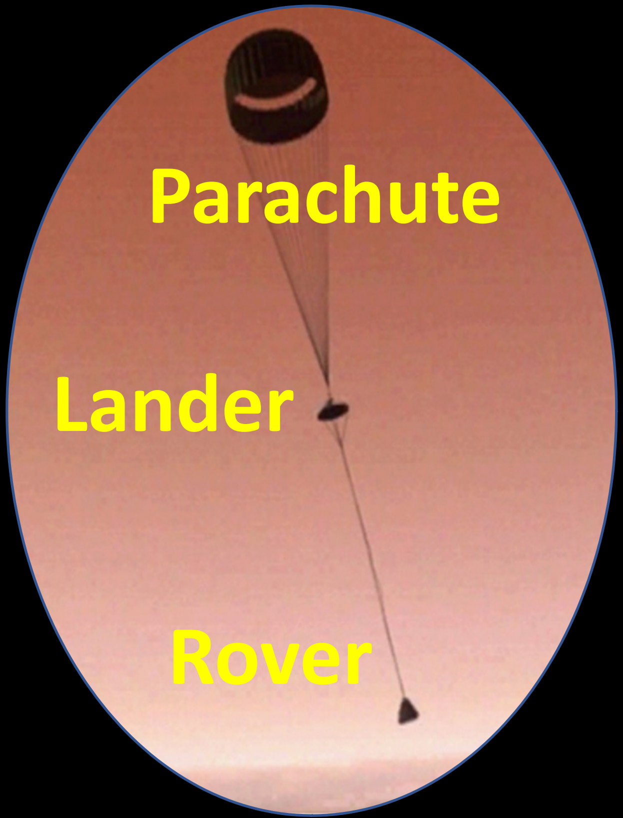




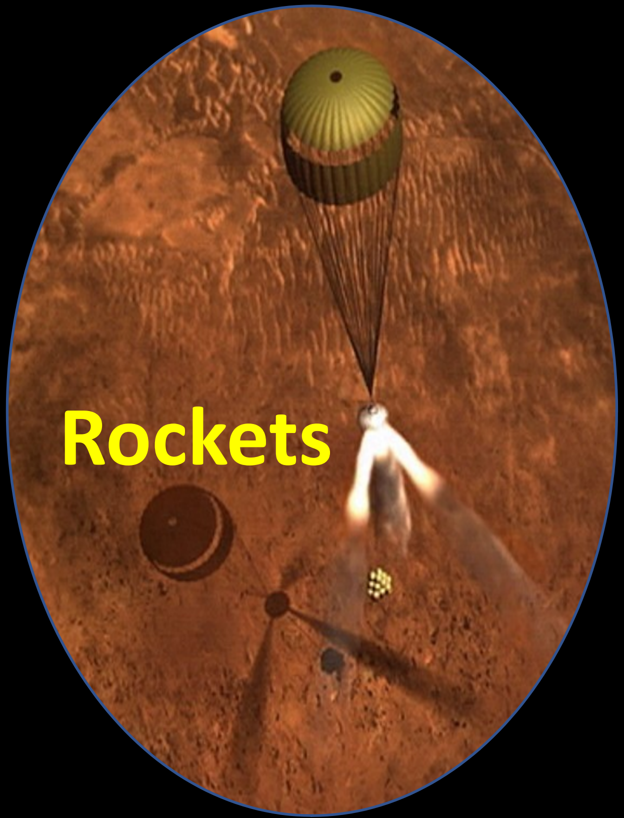


Entry, Descent and Landing - Mars Exploration Rovers (MERs) DIMES

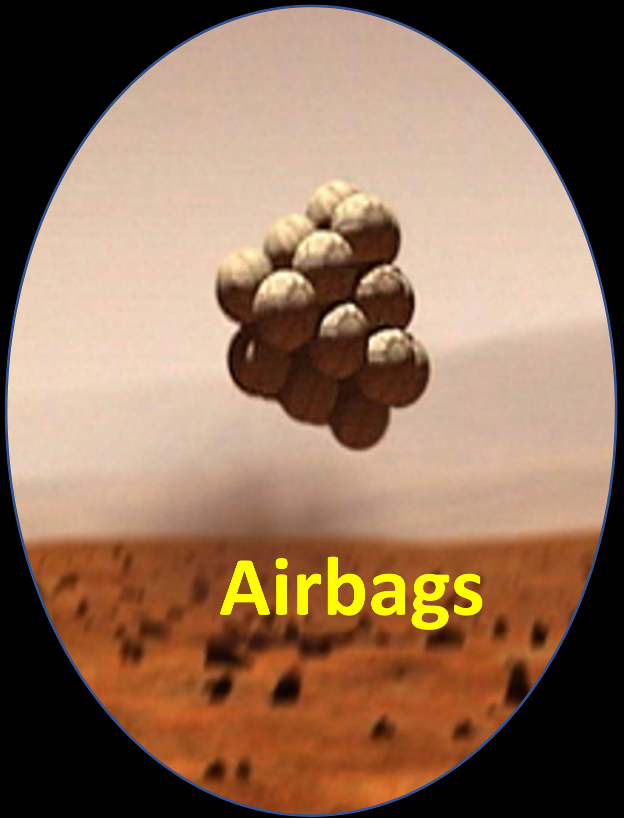
Successful Entry, Descent & Landing of both Mars Exploration Rovers in 2004



Parachute slows descent



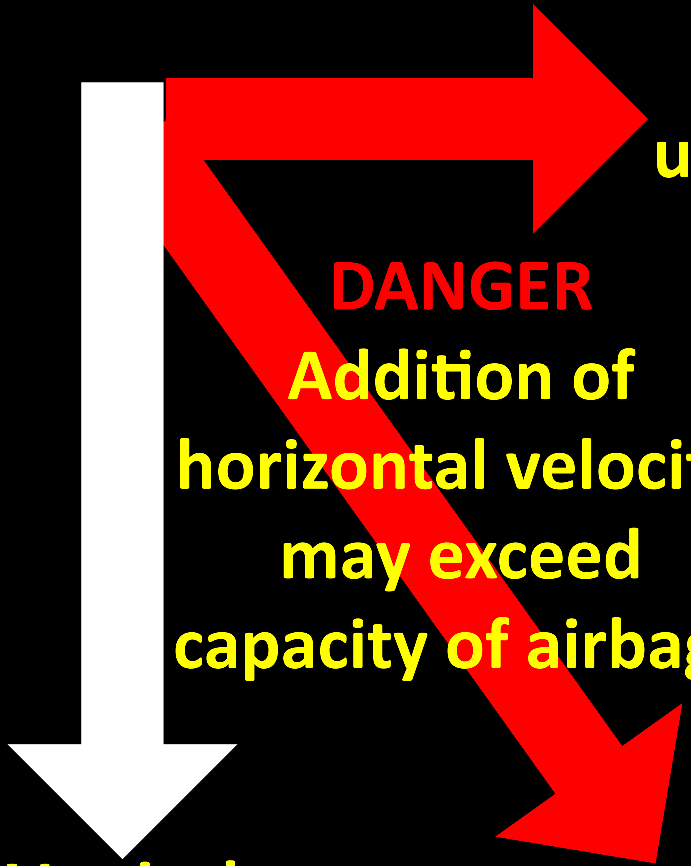
Rockets cancel velocity



Airbags cushion final drop

https://mars.nasa.gov/mer/mission/timeline/edl/steps/

DIMES (Descent Image Motion Estimation System)



UNKNOWN horizontal velocity due to unpredictable cross-winds in Martian atmosphere

DANGER
Addition of horizontal velocity may exceed capacity of airbags

Vertical velocity predicted

DIMES estimates horizontal velocity
Rockets fired to compensate if need be
Estimation:

- During descent, take photos of ground
- Pick a feature in first photo
- Identify that feature in second photo
- Deduce horizontal velocity from motion of feature between photos

https://mars.nasa.gov/mer/mission/timeline/edl/steps/

DIMES requirements

Perform within 20 seconds using 40% allocation of flight processor

Probability of reporting an ***incorrect*** velocity estimate < 0.1%

Probability of ***not*** reporting a velocity estimate ≤ 10%

"Do no harm" - avoid an incorrect estimate causing failure when the landing might have succeeded anyway

DIMES had to be autonomous – control from Earth would be too slow

The challenge: how to have confidence in DIMES?

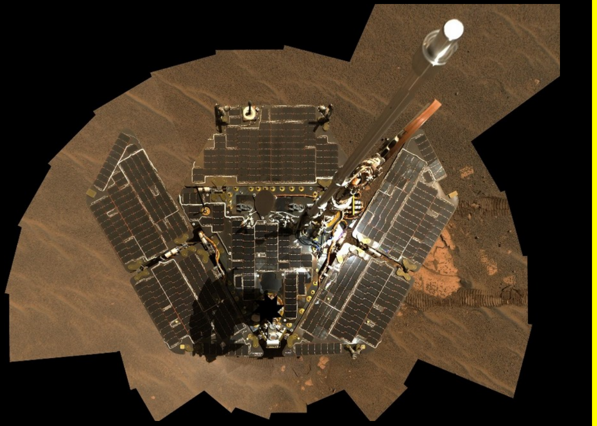
Design: predict hazards & avoid by design if possible, otherwise recognize when an estimate is not to be trusted; provide multiple (4) chances to produce a trusted velocity estimate

Test: simulations and field tests made as representative as possible

Science - (Opportunity & Curiosity)

“Laser-targeting A.I. Yields More Mars Science”

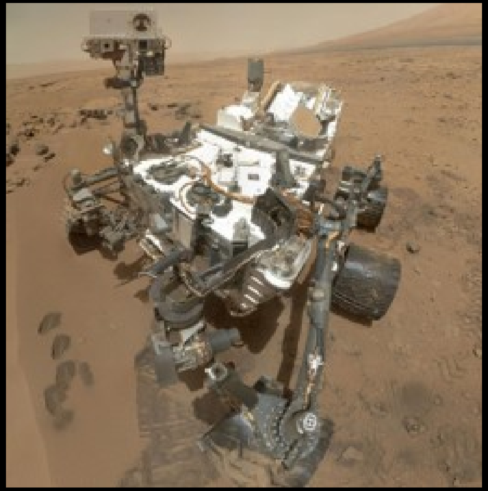
https://www.jpl.nasa.gov/news/news.php?feature=6879



https://mars.nasa.gov/resources/5852/opportunity-self-portrait/

At the end of Opportunity’s day’s drive

- Take a panoramic picture
- Identify the most scientifically interesting target
- Point the narrow field-of-view camera at it
- Take a photograph to then send back to Earth



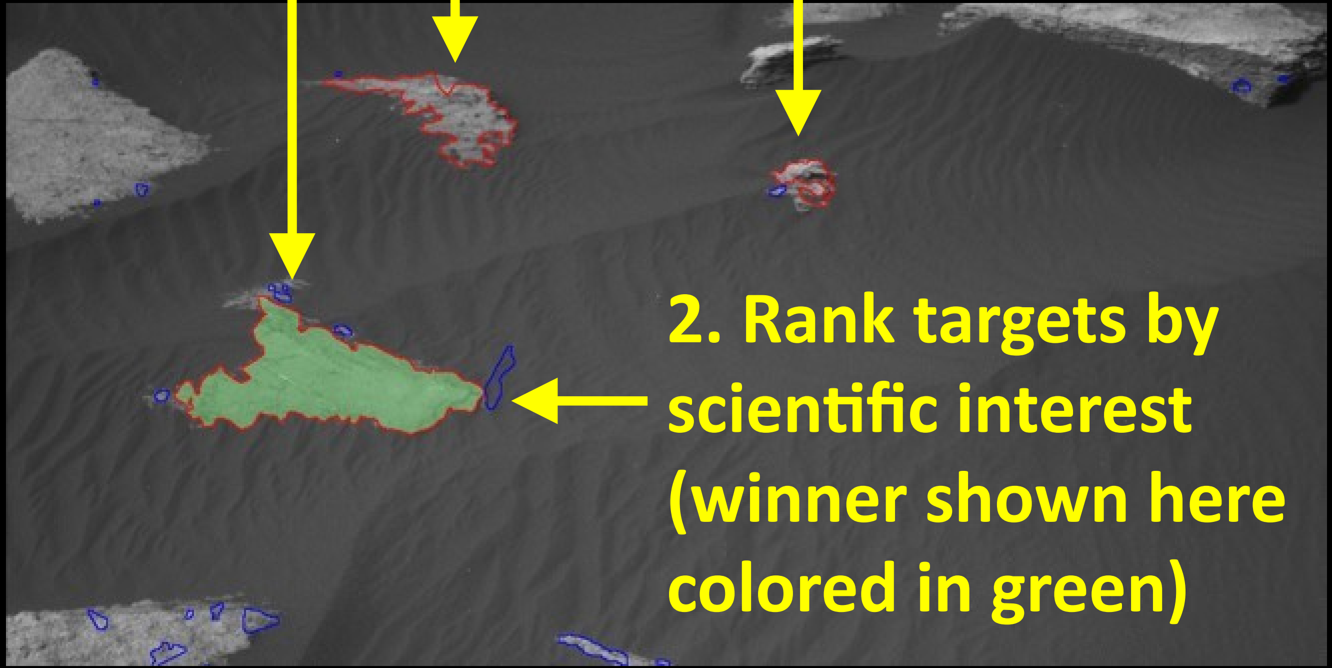
https://mars.nasa.gov/resources/4845/high-resolution-self-portrait-by-curiosity-rover-arm-camera/

At the end of Curiosity's day’s drive

- Take a panoramic picture
- Identify the most scientifically interesting target
- Point the laser & spectrometer at it
- “Zap” it with the laser to yield measurements

AEGIS (Autonomous Exploration for Gathering Increased Science)

1. Identify targets (rocks) outlines by image processing



2. Rank targets by scientific interest (winner shown here colored in green)

DANGER

- Do NOT point the camera at the sun
- Do NOT “Zap” the rover with the laser (!)
- Stop gracefully if interrupted or anything odd happens

3. Plan and execute trajectory to point camera/laser at selected target

4. Take science measurement (photo or laser zap)

https://www.jpl.nasa.gov/news/news.php?feature=6879

AEGIS (Autonomous Exploration for Gathering Increased Science)

Note: AEGIS only *enhanced*, it was not *essential* → crucial it be low risk

Design: • heritage algorithm for identifying rocks
• criteria & formulae for ranking rocks defined by scientists
• perform its own safety checks (don’t rely on rover to do so)

Testing: • using images of a variety of Martian terrains
• in avionics simulators
• on hardware rover in large Mars-like terrain sandbox
• nominal & off-nominal (e.g., image with no rocks)

Deployment: • progressive checkout – incrementally do more steps
• success on Opportunity paved way for use on Curiosity

Benefit clear: • additional science measurement not otherwise obtained
• much better than purely random targeting

Conclusions

- Autonomous systems have been successfully employed and verified in launch vehicles and spacecraft

- Automation is especially necessary with the presence of the following:
  - Time criticality
  - Communications delay or absence

- These case studies provide examples of successful verification strategies across a broad range of domains:
  - Multiple mission phases (ascent, transit, decent/landing, science operations)
  - Multiple launch vehicles and spacecraft
  - Differing criticality levels

- Some Common Themes
  - Lots of testing (!) - analysis / tests in environments with increasing fidelities
  - Concerted team effort and much time devoted not only to duplicate nominal, but also identify extensive off-nominal conditions to minimize “false positives”

Thank you for your attention