National Aeronautics and Space Administration

Integrated System for Autonomous and Adaptive Caretaking (ISAAC)

Gateway T&V September Demo

Oleg Alexandrov, Mark Allan, Julia Badger, Chuck Claunch, Matt Deans, Lorenzo Flückiger, Patrick Knauth, Khaled Sharif, Trey Smith, Terry Thurk

26 Sept 2019

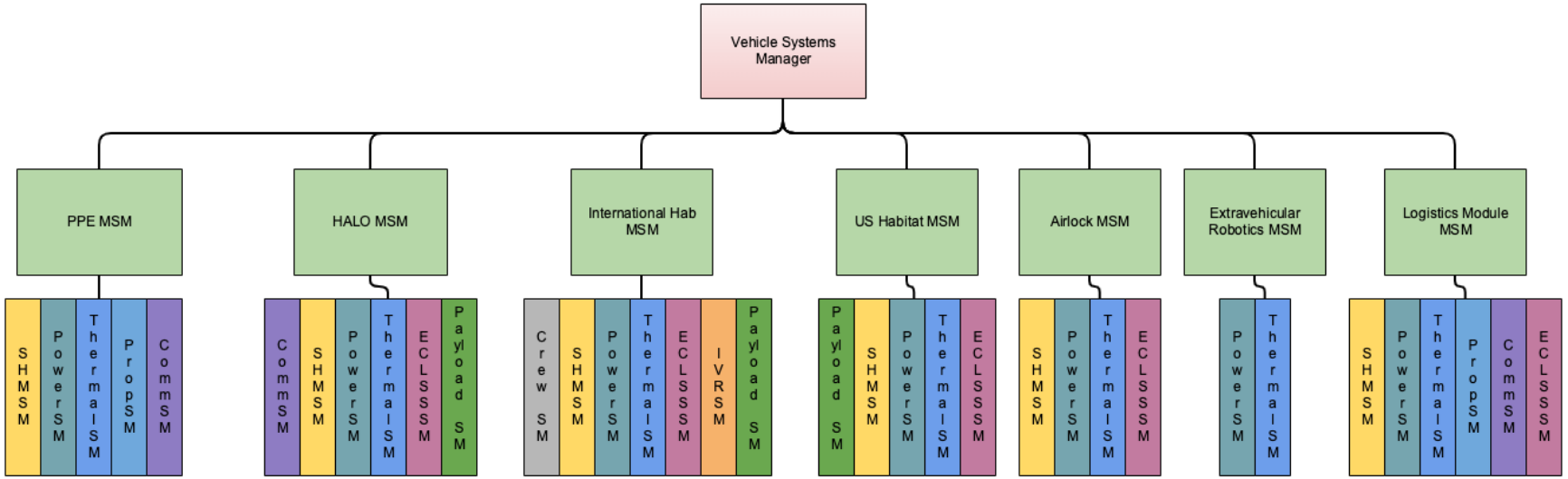
ISAAC FY19 Technical Results

- **MAST integrated with robots**
  - Astrobee in simulation
- **MAST integrated with Gateway FSW**
  - Now using the same cFS SBN communication architecture
- **Adaptive data management**
  - Change logging and downlink allocation based on what is important right now
- **Acoustic mapping**
  - With Astrobee in simulation
- **Leak detection and localization demo in iPAS**
- **Full MAST integration with GiaB***
  - GiaB running sim, FSW
  - *Full MAST integration expected in the next 2 weeks

Vehicle System Management

- **Vehicle systems management** software provides in situ operational autonomy
  - Distributed, hierarchical architecture
  - Clear definitions of interfaces and interdependencies
  - Careful design of locus of authority

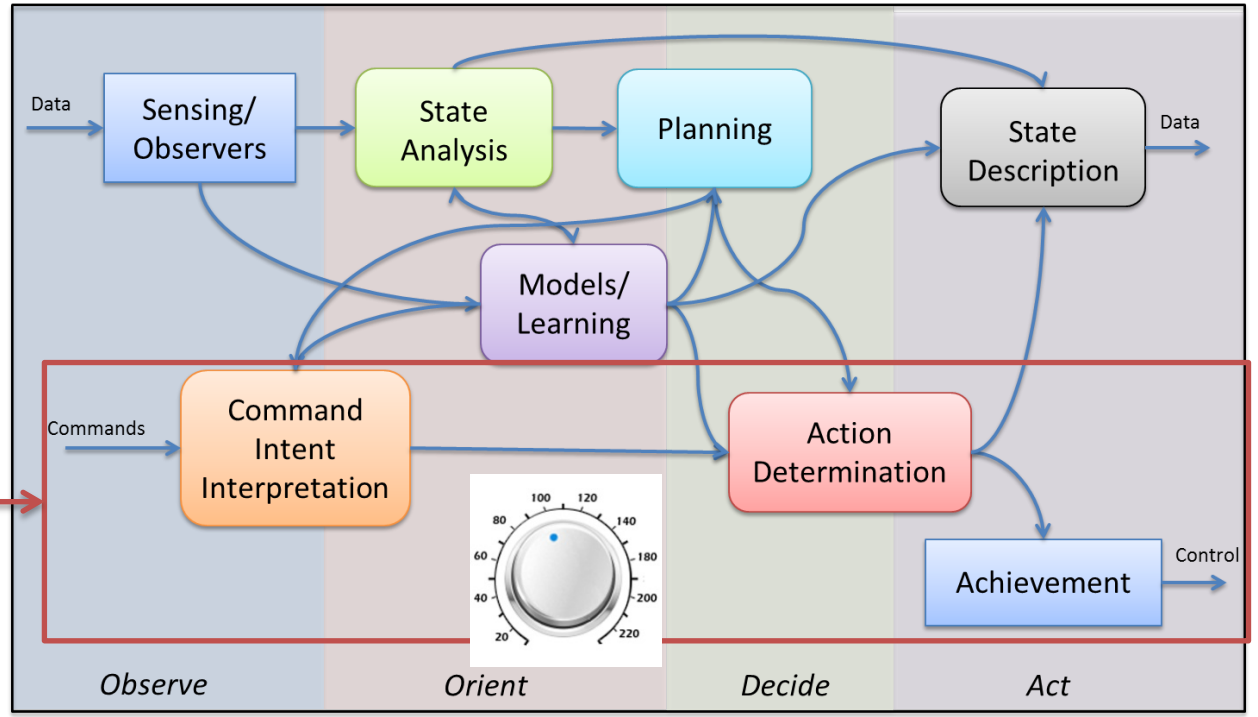
Conceptual diagram of vehicle following the Autonomous Systems Management Architecture (ASMA).



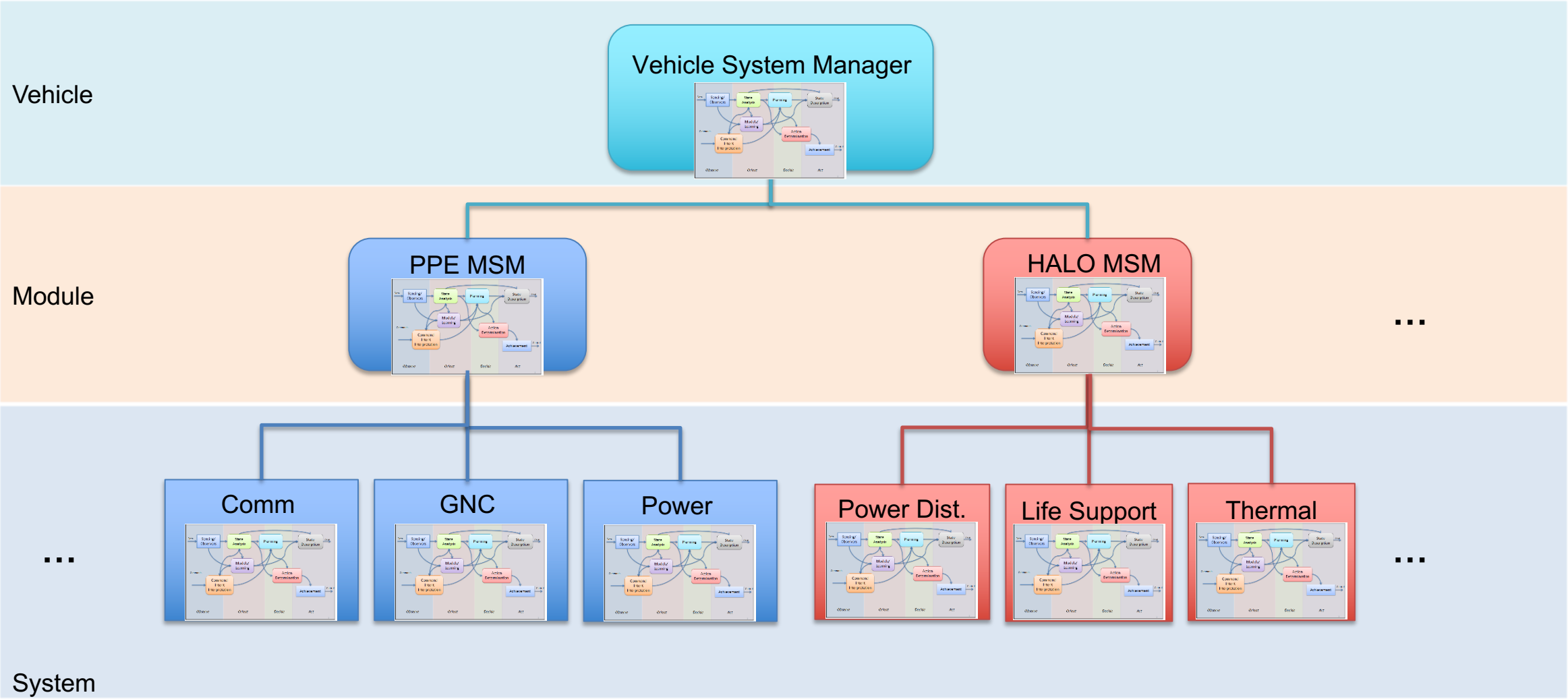
Modular Autonomous Systems Technology

- The Modular Autonomous Systems Technology (MAST) framework implements the ASMA:
  - Can be used for all classes of autonomous systems
  - Standardizes information sharing and interfaces between technologies
  - Designed around formal verification and validation principles

- Variable autonomy possible by figuring out how to throttle "actions"

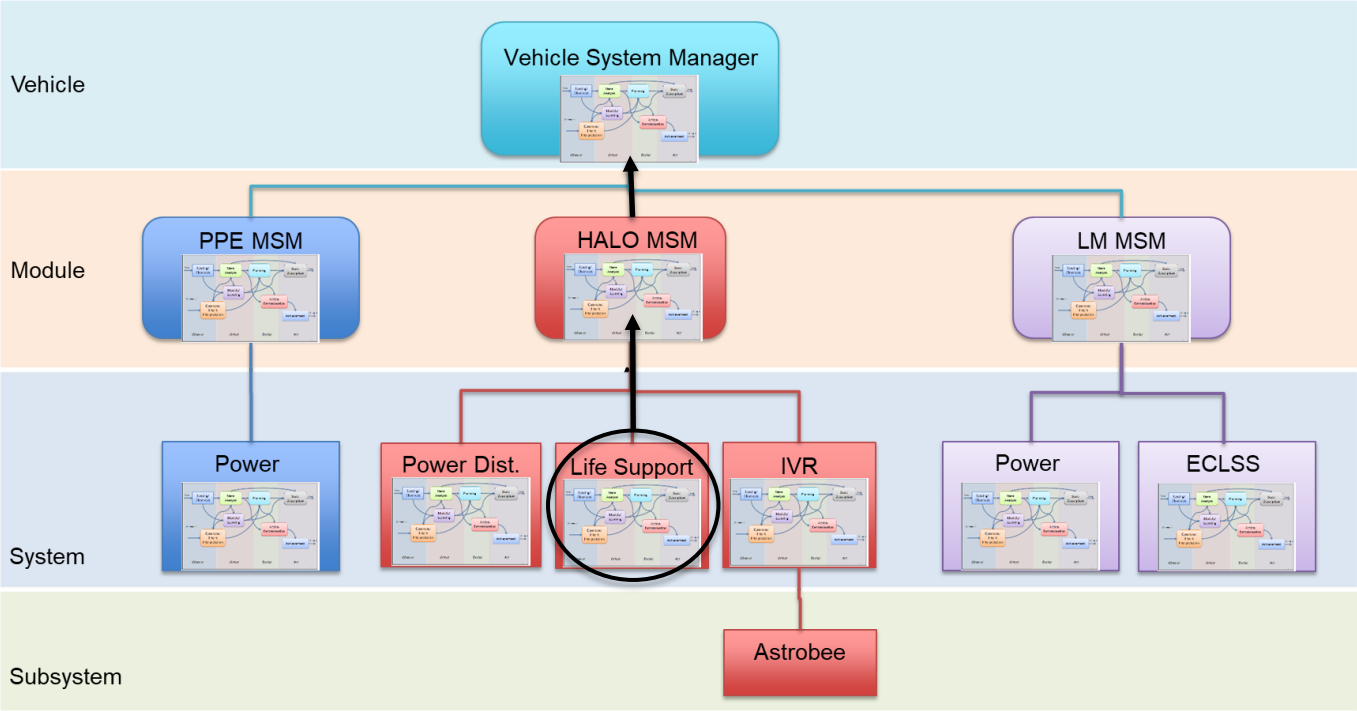


MAST ASMA Emulation



Distributed Hierarchical Control Example- Fault Detection

State Analysis:



Life Support

Pressure drop detected

System Responses:

Module Response:

HALO MSM

Life Support

- Reports status up

- Does not diagnose due to LM ECLSS (missing info)
- Communicates state up to VSM

Vehicle Response:

Vehicle System Manager

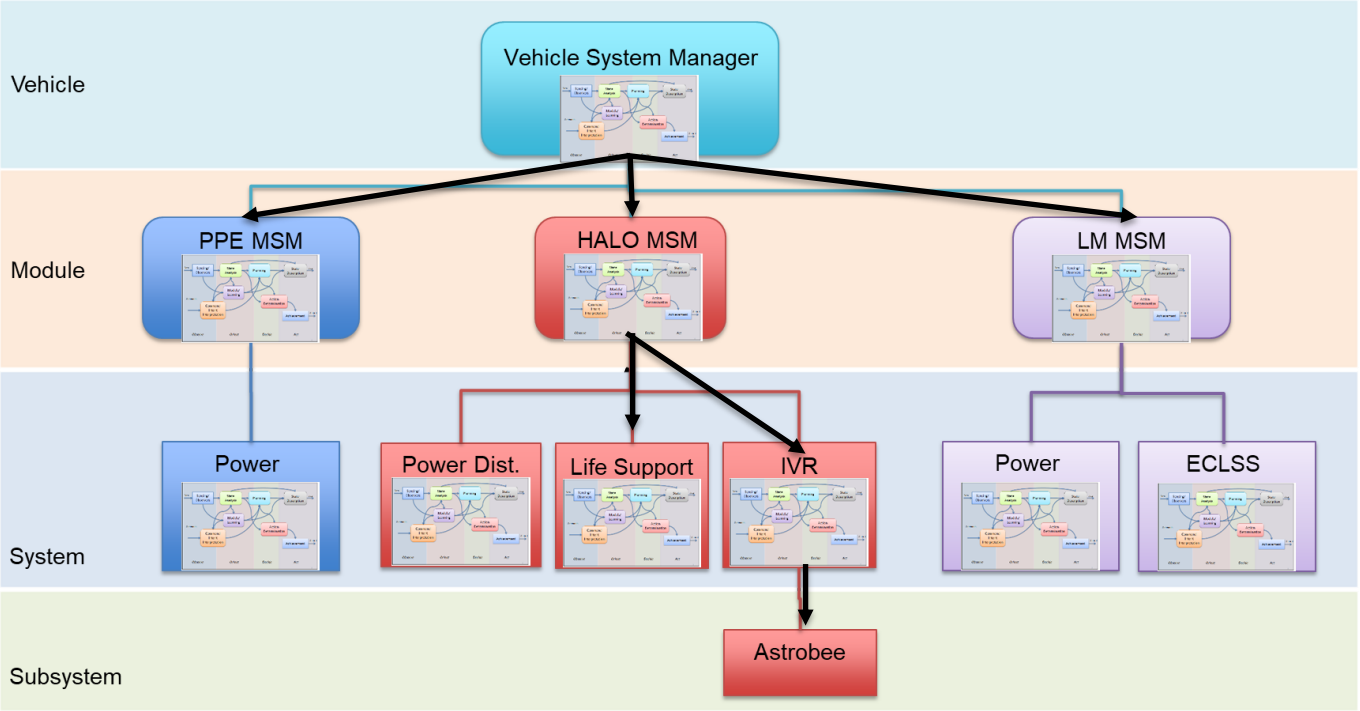
- Confirms leak
- Vehicle status is reported to crew/ground

Distributed Hierarchical Control Example- Fault Isolation

Plan and Execute:

Astrobee

Search pattern initiated



System Responses:

Module Response:

HALO MSM

- Adjusts telemetry plan, sends commands
- Commands IVR SM to initiate leak search
- Commands ECLSS SM to shut off fan

Vehicle Response:

Vehicle System Manager

- Sends command to adjust telemetry
- VSM sends tasks to modules to find and mitigate leak effects

IVR

- Throttles data
- Commands Astrobee subsystem to initiate leak search

Life Support

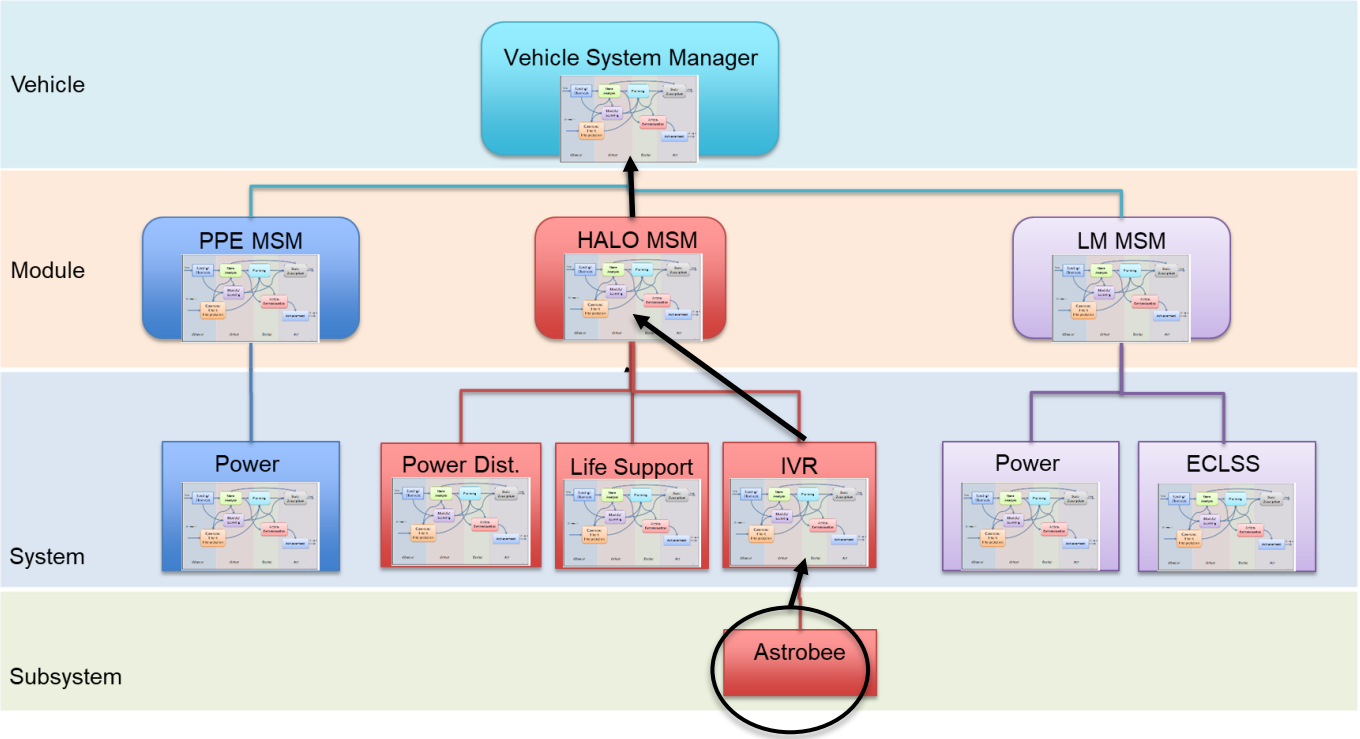
- Throttles data
- Commands fan off

Distributed Hierarchical Control Example- Closing the Loop

State Analysis:

Astrobee

Acoustic sensing data sent up



System Responses:

IVR

- Generate report
- Send status/data up

Module Response:

HALO MSM

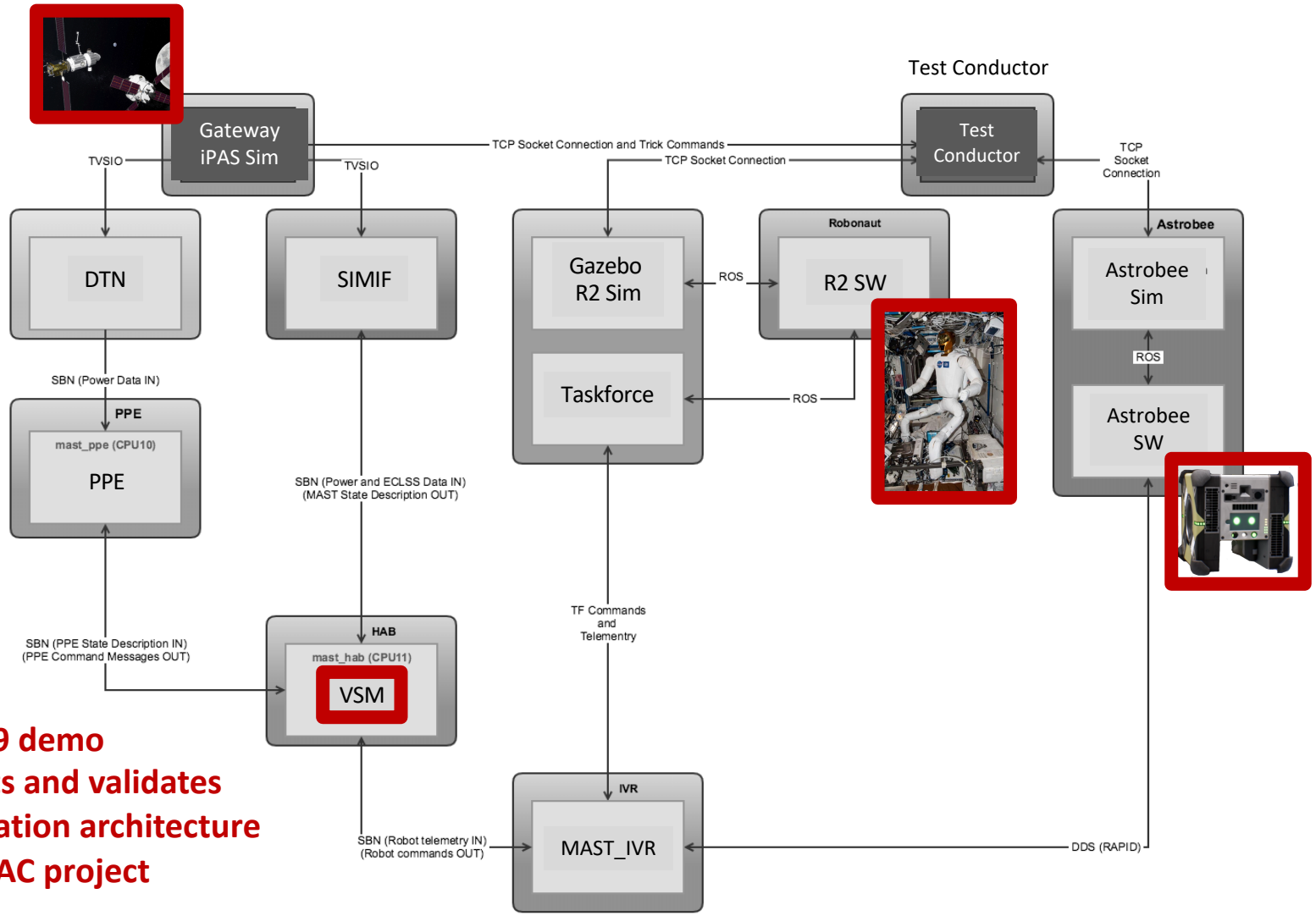
- Consolidates IVR status with other telemetry
- Send status/data up

Vehicle Response:

Vehicle System Manager

- Vehicle status is reported to crew/ground

Integrating vehicle and robots



ISAAC-FY19 demo implements and validates communication architecture for full ISAAC project

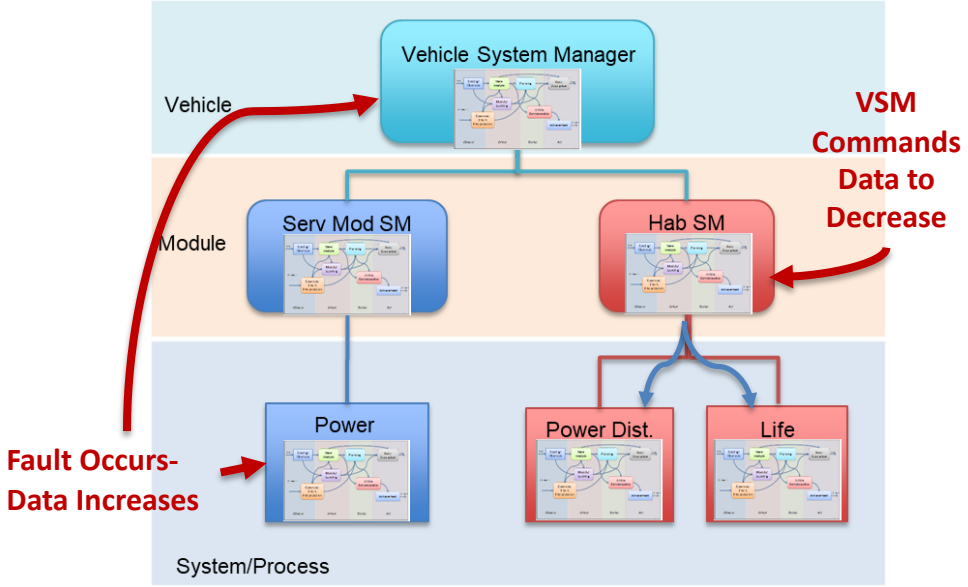
Adaptive data management

- **Need**
  - Bandwidth, latency dictate how much data we can downlink
  - Data is generated faster than we can downlink and store (especially when robots and payloads are included)
  - **Respond to GTW-L2-047**: *The Gateway shall curate data based on priorities and events.*

Accomplishments

- Developed test infrastructure to vary network bandwidth and latency
- Mast now supports classifying data into categories, then prioritizing and throttling based on these categories

Benefit: *Provides method to control data downlink based on events*



End-of-year demo

- **Milestone: September iPAS demonstration**
  - Test distributed framework with Vehicle System Manager (SM), Module SMs, and System SMs (including an IVR SM)
  - Use case: leak detection and localization scenario
  - Additions of several SMs (including ECLSS) to fill out data integration efforts
    - Requires sufficient integration efforts with other groups
- **Demonstrate:**
  - Scaling of ASM architecture, MAST
  - Data throttling over larger network
  - Data/command interface to ground through VSM
  - Robot data passing through ASM architecture (spatial integration is incomplete)
  - Applicability to Gateway autonomy architecture

Leak Management Scenario Overview

- **ISAAC is primarily about building integrated autonomy technology that is applicable across:**
  - Many capabilities (e.g. state assessment, logistics management, fault management)
  - Multiple future missions
- **However, ISAAC will do fairly deep dives on particular Gateway-relevant scenarios in order to validate the technology – leak management is one**
- **Leak scenario assumptions:**
  - A micrometeoroid strike has caused a leak in a Gateway module during an uncrewed mission phase
  - The leak must be patched quickly (hours) to avoid major impacts, such as losing pressurized payloads that can't tolerate depressurization
- **Leak management phases:**
  - Leak detection
  - Leak localization
  - Leak mitigation [not modeled in demo]

1. Leak Detection

- **Our current implementation detects the leak based on pressure sensor trend analysis**
- **Complementary methods could also be evaluated later:**
  - Detecting the sound of the impact itself
  - Detecting thrust generated by the leak

2. Leak Localization

- **Initial localization**
  - Assuming closed hatches, pressure sensing determines location at the level of which module.
  - If Gateway has embedded structural vibration sensors (not 100% clear yet), they may be able to provide coarse localization of the sound of impact or the sound of escaping air, “within a rack or two”.
    - Partly to keep the demo time duration conveniently short, our current implementation assumes the leak location is initially known to within about a 2 x 2 meter area
- **Survey preparation [not modeled in demo]**
  - Prior to searching for the leak, some site prep may be needed
    - Example: Turn off noisy systems that could mask leak noise
    - Example: Swing racks out of the way to expose possible leak locations on the pressure hull
- **Survey**
  - Use a mobile inspection robot with ultrasound sensors to find noise sources that could be the leak
  - **More on this later!**
- **Refinement and confirmation [not modeled in demo]**
  - Once candidate leak locations have been identified, the robot can approach and dwell near those locations to collect better data
  - In addition to noise sensing, other lines of evidence can help confirm a leak
    - Example: Visible imaging to see the hole
    - Example: Thermal imaging to see localized cooling from air expansion

3. Leak Mitigation [not modeled in demo]

- **Depending on the location and type of leak, a mobile manipulator robot may be able to patch it using a patch kit similar to those currently present on the ISS**
- **Otherwise, mitigation would focus on steps like moving sensitive equipment out of the affected module, and closing hatches to isolate it from the rest of the Gateway**

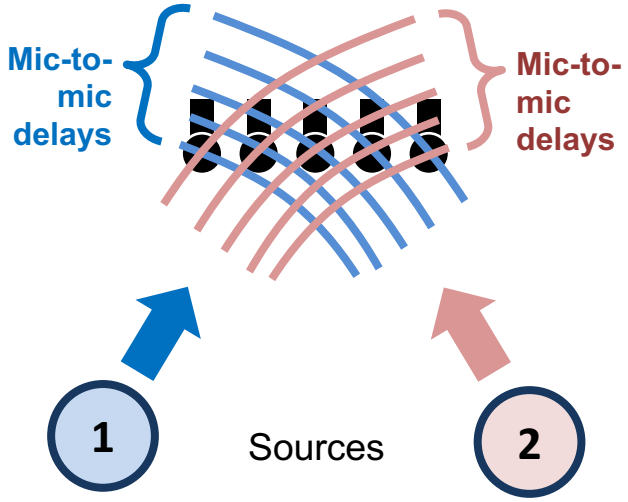
Leak Localization Survey

- **Experience from use of the ultrasonic leak detector on the ISS shows that:**
  - There are many confounding sources of ultrasound noise, some louder than a small leak
  - At least when using a single-microphone sensor, a leak is hard to detect until you are in close proximity (e.g. < 1 meter)
- **To address these challenges, we use the following survey data analysis steps:**
  - Direction-of-arrival (DOA) estimation
  - Spatial response imaging
  - Map building
  - Source separation
  - Source rejection filters

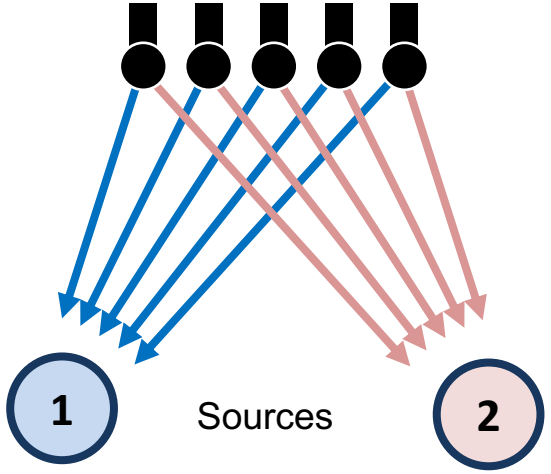
Leak Localization Survey – Direction of Arrival (DOA) Estimation

- **We can sample an (ultra)sound field with a compact array of multiple microphones**
- **Depending on the location of a sound source, the sound arrival time will be offset between different microphones in the array**
  - (Delays can be computed from source location, microphone array geometry, and speed of sound)
- **By cross-correlating microphones and calculating relative arrival times, we can estimate the direction of arrival (DOA) for a source**
- **With a sufficient number of microphones, multiple sources can be separated and localized simultaneously**
- **With a 1D array of microphones, we can solve for 1 degree-of-freedom (DOF) DOA**

1. Measure differential delay

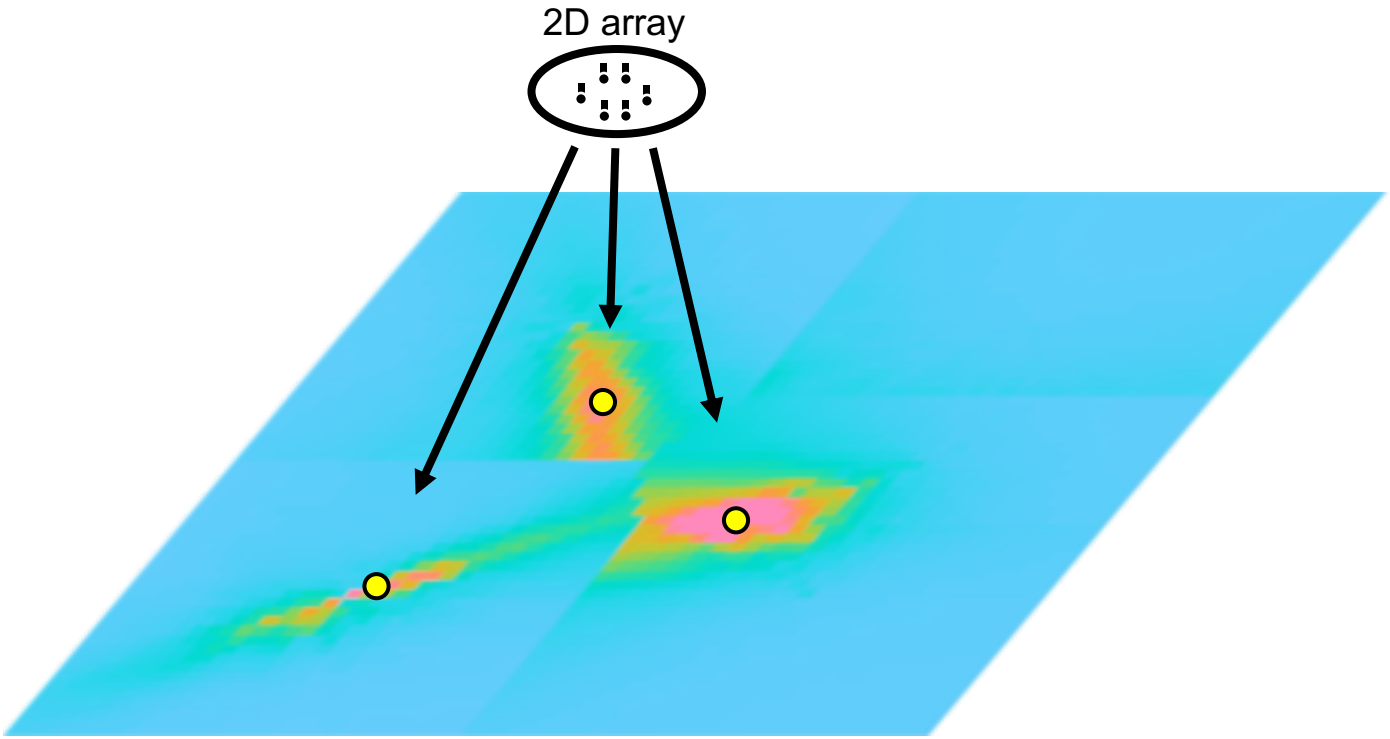


2. Solve for direction of arrival (DOA)



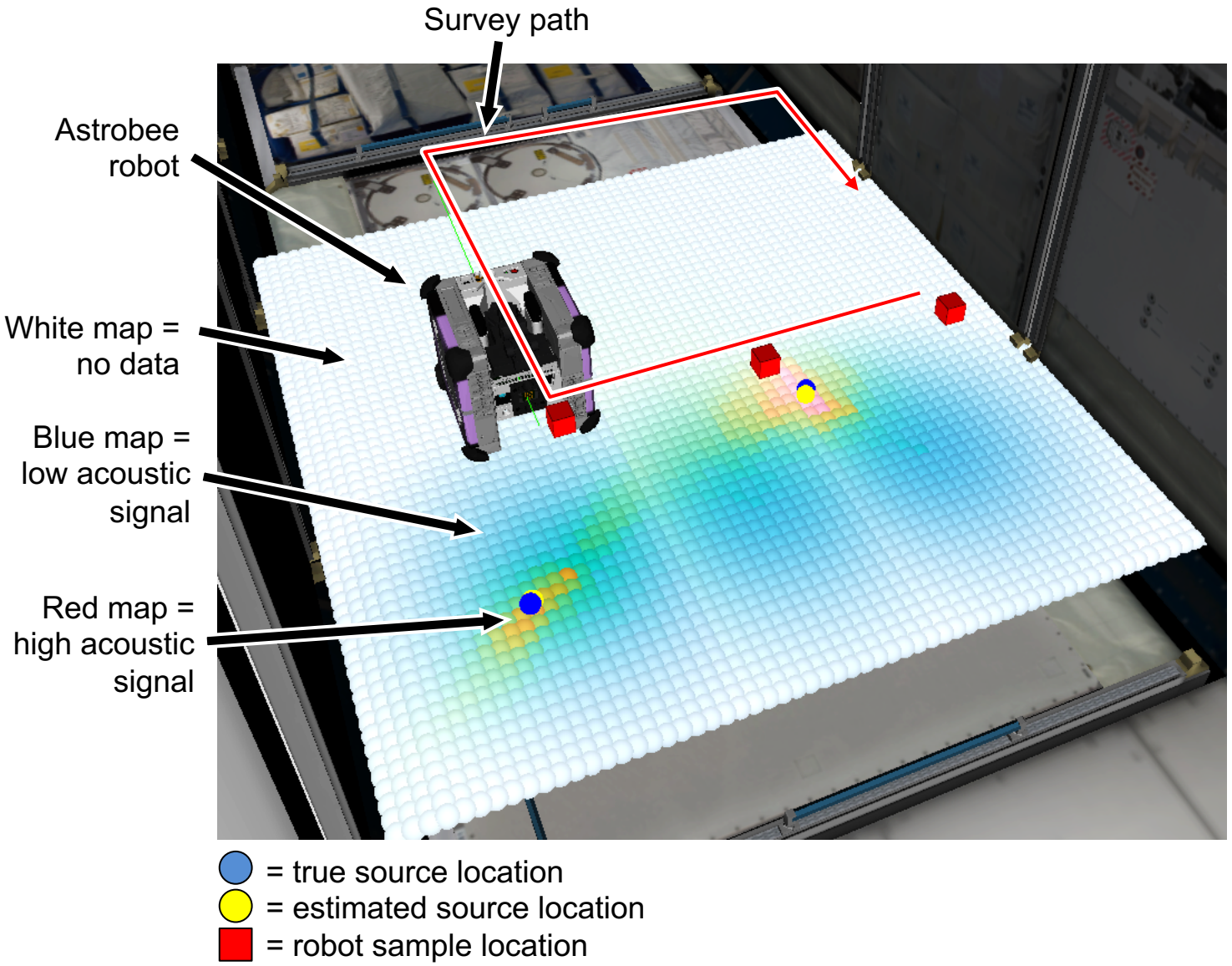
Leak Localization Survey – Spatial Response Imaging

- **With a 2D microphone array, we can solve for 2 DOF DOA**
- **We can also visualize the spatial response of the microphone array as an image**
  - Hot colors in areas that are producing more noise



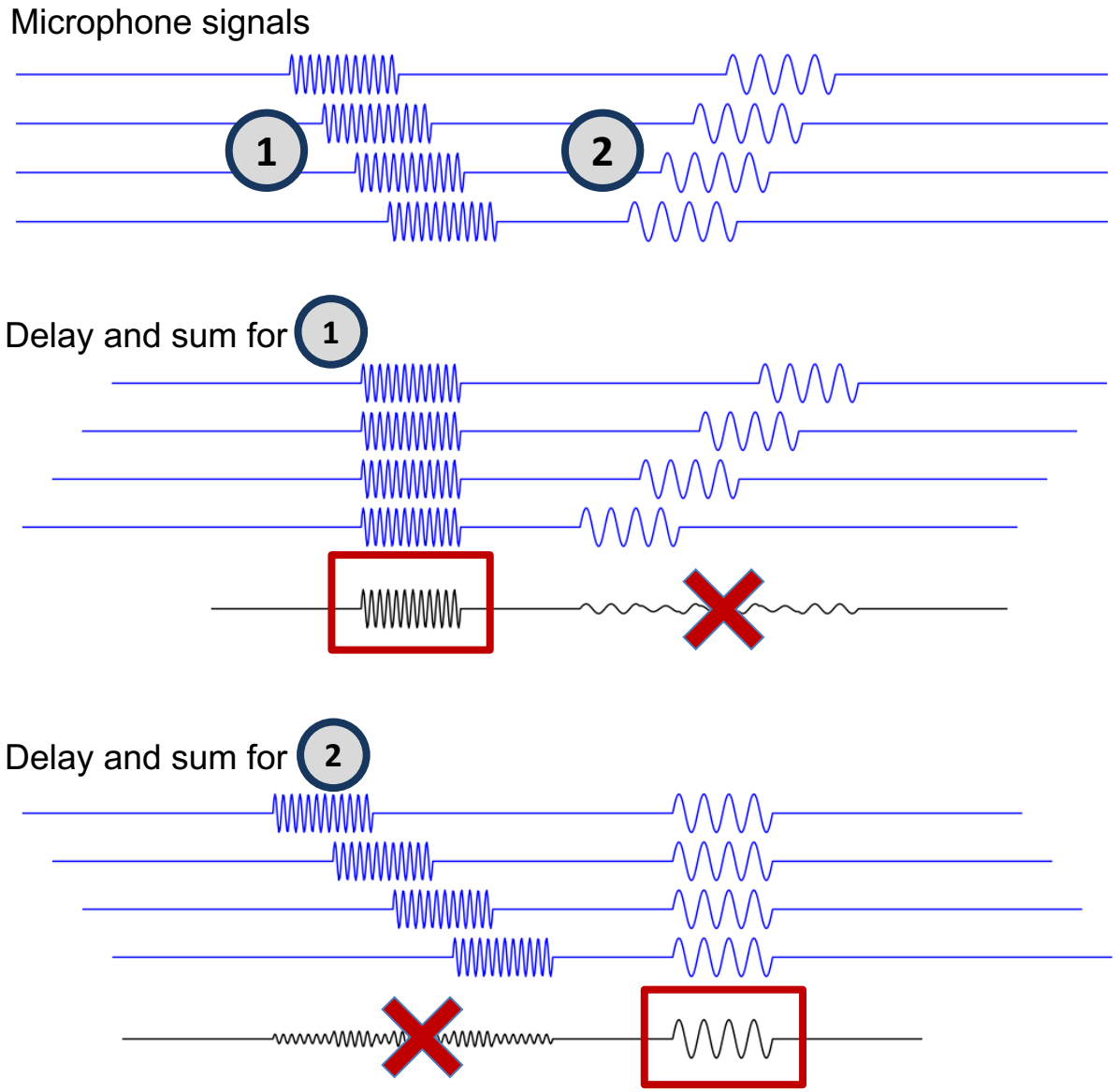
Leak Localization Survey – Map Building

- **The robot survey collects multiple samples in different locations**
- **The robot knows its own position and the surrounding 3D geometry**
- **Therefore, we can build a map by projecting the local spatial response onto nearby surfaces (in this case, the deck)**
- **Then we can use peak detection in the map to estimate source locations**



Leak Localization Survey – Source Separation

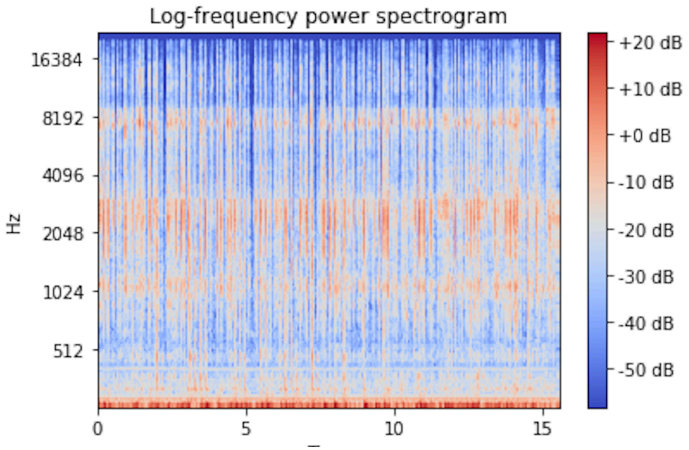
- **Once source locations are identified from the map, we:**
  - Pick the closest sound sample to each source
  - Use delay-and-sum beamforming to separate out and amplify the sound signal from that source
- **This is especially important when there are loud confounding sources**



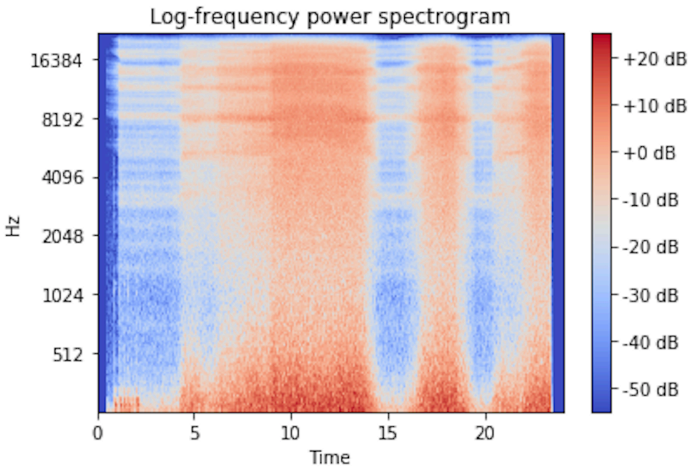
Sound classification – “non-leak” filter

- **A sound classifier can distinguish sounds, for example:**
  - Leak: Should sound like hissing or whistling
  - Non-leak: Other familiar sounds, such as motors or lights buzzing or clicking
- **ISAAC’s demo classifier is designed to be applied to the separated signal from each source, using a standard approach:**
  1. Generate a spectrogram image representation of each sound (examples at right)
  2. Apply a deep learning image classifier
- **Classifier performance was good, evaluated across a wide variety of sound samples from freesound.org**
- ***** This sophisticated sound classifier didn’t quite make it into the integrated demo on Sep 26 due to last-minute compatibility issues with the development environment**
  - As a fallback, we made the simulated leak and non-leak type sounds have different pitches, so that a simple frequency-based classifier could distinguish between them

Keyboard typing

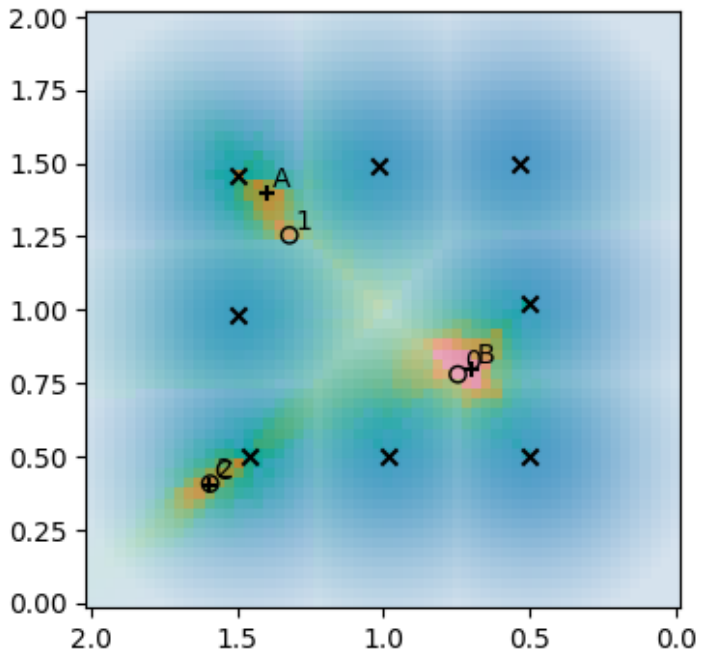


Steam hissing



Leak Localization Survey – Source Rejection Filters

- **We currently use two main principles to rule out ultrasound sources as leaks**
- **Non-leak filter:**
  - The sound classifier from the previous page: it returns a high score if the reconstructed sound signal sounds like a “non-leak” (motor buzz, clicking, etc.)
- **Baseline filter:**
  - A comparison filter that returns a high score if the source properties (both location and sound signal) are similar to a source observed in past baseline surveys
  - (A baseline source is unlikely to be a leak)
- **If a source has high intensity and passes the non-leak and baseline filters, it is considered a likely leak**
- **For the demo, we simulate three sound sources:**
  A. The actual leak (weakest source)
  B. A source with the wrong sound power spectrum to be a leak
  C. A source that was observed in the baseline survey
- **Despite louder confounding sources, the system can identify peak 1 (near source A) as the most likely leak location**



Peak	Intensity	Non-leak	Baseline	Leak
0	0.54	0.99	0.00	0.01
1	0.22	0.00	0.00	**0.53**
2	0.21	0.00	0.79	0.11

x = Robot sample location
+ = Actual sound source (A is the leak)
o = Detected sound source

Peak 1 passes the two filters, correctly identified as the most likely leak

Leak Management Scenario - Reusability

- **Technology developed for this scenario is reusable**
  - Acoustic mapping has uses other than leak localization
    - Example: Prognostics for impending mechanism failure
  - Projecting imagery onto module interior 3D geometry for analysis applies to other sensors
    - Examples: Visual imagery, thermal imagery
  - The same filtering logic will also re-occur in other scenarios
    - Baseline vs. anomaly detection
    - Machine learning to classify anomaly type and determine if action is needed

Demo highlights video

Demo highlights video

https://ti.arc.nasa.gov/ti/isaacdemo/

Summary & Future Work

- **MAST developed directly for the architecture proposed for Gateway Autonomous Systems Management**
  - Will be used as preliminary emulations of command/data architecture for delivery to module partners (Maxar, NGIS)
  - Understand data flow and model development across elements- explore mission use cases garnered from Gateway VSM WG
  - Added robots to the mix – special use case of commanding a resource that needs to have a fair amount of self-autonomy
  - MAST will continue to be basis for Gateway VSM development
- **ISAAC project will focus on the following areas over the next 3 years:**
  - Integrate disparate types of data to solve problems (includes spatial and temporal registration)
  - Develop integrated control interfaces to promote ground operator situational awareness (special emphasis on controlling remote robots to optimize data efficacy for SA)
  - Use planning technology to coordinate execution of procedures requiring multiple assets, particularly those needing spatial and temporal constraints/goals
- **ISAAC aims to extend MAST capabilities in order to enable Phase 2 Gateway VSM functionalities**