Distributed Spacecraft Autonomy (DSA):
Development of Swarm Autonomy Capability and Scalability for Spacecraft

Jason Fugate

February 10, 2021
Outline

• Project Overview
• Motivation
• Notional Autonomy Use Case
• System Design
• Project Overview
• Motivation
• Notional Autonomy Use Case
• System Design
• Project Goals:
  – Advancing command and control methodologies for controlling a swarm of spacecraft as a single entity
  – Developing, maturing, and demonstrating autonomous coordination between multiple spacecraft in the swarm
  – Developing, maturing, and demonstrating approaches for adaptive reconfiguration and distributed consistency planning across a swarm of spacecraft

• Demonstration
  – **Starling Flight Mission**: DSA will demonstrate flight-relevant autonomy capabilities in a multi-spacecraft mission as a software payload
    • Scale: 4, Timeframe: Years 1-3
  – **DSA Simulation Mission**: DSA will demonstrate scalability of the same architecture driven by a Lunar Positioning, Navigation, and Timing (PNT) application
    • Scale: 100, Timeframe: Year 4

**DSA will establish scalability as a core design option for space missions**

Ionospheric results from CHAMP Mission. DSA-Starling will be reacting to the same physical phenomena

• Project Overview
• **Motivation**
• Notional Autonomy Use Case
• System Design
• Autonomous decision-making is needed for multi-spacecraft missions due to latency, bandwidth constraints, and mission complexity

• Autonomy fundamentally brings independent adaptability to missions
  – Increases capabilities of an asset
  – Reduces risk to a mission

• Distributed assets provide robustness
  – Increases survivability
  – Enhances scalability and maintainability
  – Provides reliability

**DSA is developing and maturing multi-asset autonomy**
Motivation

What Does DSA Enable?

Simultaneous multipoint data collection

Increased availability

Enhanced system capabilities

n to m vehicle operations

n < m

Increased throughput and reduced cost

reduced down time and graceful degradation
Outline

• Project Overview
• Motivation
• Notional Autonomy Use Case
• System Design
**Goal:** Balance the reactivity of the swarm to changing conditions against the need to focus on identified high-value targets

**Explorative vs. Exploitative**

**Explorative:**
- Allows the swarm to retain “situational awareness”
- Encourages the monitoring of unobserved observations
- Demonstrates the ability of the swarm to **balance** observations across assets

**Exploitative:**
- Allows the swarm to focus on high-value phenomena
- Encourages the capture of "interesting" observations
- Demonstrates the ability of the swarm to **prioritize** observations across assets
Case Study: GPS Channel Selection/Allocation

Spacecraft perform live channel selection to analyze for features of interest.

Example Scenario

- Each spacecraft can see 4 GPS, but is constrained to listen to only 2
  - Blue TEC values correspond to explorative rewards – spacecraft explore for new features by listening to as many different GPS as possible (e.g. assignments A-1, C-4)
  - Red TEC values correspond to exploitative rewards – i.e., multiple spacecraft listen to the same GPS (e.g. assignments A-3, B-3)
Notional Autonomy Use Case

GPS Radio Tomography of the Topside Ionosphere

- Previous missions (e.g. SWARM, CHAMP) have established the science that will drive DSA-Starling’s autonomy algorithms.
- Swarm Assets will collect Total Electron Content (TEC) information using an 8-channel dual-frequency GPS receiver also used for precise orbit determination (similar to [1]).
- SWARM data products provide testing and training sets for DSA-Starling:
  - Same Data rate (1 Hz)
  - Similar altitude (450 km vs. ~500 km)
  - Similar formation (pushbroom)

We are therefore performing an autonomy demonstration with a science use case

Notional Autonomy Use Case

Explorative Demonstration:

Definition:
Characterization of the swarm’s ability to observe a large, homogenous feature (e.g. The Equatorial Ionization Anomaly [EIA])

Purpose:
Ensures maximum EIA coverage via partitioning the set of GPS channels in view with the goal of producing as few as possible overlaps
Exploitative Demonstration:

**Definition:**
Characterization of the swarm’s ability to react to transient, localized phenomena (i.e. polar patches)

**Purpose:**
Focuses the swarm on GPS channels that are showing evidence of passing through a polar patch

---

Notional Autonomy Use Case

---

Occurrence of polar patch phenomena over time for the GPS instrument of the SWARM satellites.

---

Polar Patch Example

\[
\frac{(\text{TEC}_u - \text{TEC}_{\text{ref}})}{\text{TEC}_{\text{ref}}} \geq 1.2
\]

---

NASA DSA: AI & Data Science Workshop 2021
Distributed Agent Architecture

Each **Swarm Sat** is an autonomous agent
- with its own sense/plan/act cycle.
- with a local **tecApp** that collects TEC *observations* for visible GPS

**Observation:** \(<\text{gpsId}, \text{exploreReward}, \text{exploitReward}>\)
- *explore reward* \(\propto\) distance(sat sat, GPS sat)
- *exploit reward* \(\propto\) TEC(swarm sat, GPS sat)

**Shared States:**
- Swarm sats share their local *observations* to achieve global shared state

**Job Scheduling:** Assign GPS observation "jobs" to swarm sat "machines"

**Distributed Decision Making:**
- Locally-generated global plans (job assignments for the whole swarm)
- Each swarm sat uses the same planning algorithm & same shared state information to *generate and execute* the same plans
Autonomous agent cycle

Each swarm agent performs its autonomy cycle on each tick event (triggered by cFS msg)

- On each tick, each agent generates a global plan to be executed on the next tick.
- The global plan is a set of GPS assignments for each agent in the swarm.
• Project Overview
• Motivation
• Notional Autonomy Use Case
• System Design
• System employs NASA’s open-source Core Flight Software (cFS)
  – cFS is primarily designed for single-board computers and single spacecraft
• Communication between mission applications on the internal cFS is extended to utilize Data Distribution Service (DDS) for vehicle-to-vehicle networking
  – The DDS middleware provides reliable delivery, routing, and topic subscription features over User Datagram Protocol (UDP).
  – This requires interface between cFS and DDS
cFS Layout

DSA Dataflow

Key
- Non-cFS Messages
- cFS Messages
- DSA Apps
- Non-DSA Apps
- Non-DSA Apps

NASA DSA: AI & Data Science Workshop 2021
Questions?
• Leveraging Linux containerization, a networked set of satellite instances are generated by script to simulate swarm behavior.

• Swarm commanding and synchronization through the network is demonstrated under various topologies and data-loss conditions.
• Concepts of Operations:

- Swarms are relatively resilient and robust against faults; however, they could demonstrate suboptimal performance due to decentralized control.
- Other entities such as ESA and NASA JPL are also investigating the applicability of swarms in space.
• Why is scalability hard?
• How we address scalability

Agent Complexity

Simple platform with one mission system → Highly capable platform with multiple mission systems, able to make complex decisions in unstructured environments

Collective Complexity

Leader-follower formations → Fluid, distributed maneuvers
Static comms → Dynamic, resilient networking
Scripted, rule-based behaviors → Adaptive collaboration, flexible coordination

Heterogeneity

Identical platforms and payloads in a single domain → Heterogeneous platforms and payloads in multiple domains

Human-Swarm Interaction

Simple, fixed functions or remotely operated → Freestyle intuitive human-swarm interactions
• How we address scalability
Mission Control setting goals and priorities

**Goals**

- Swarm assigning, balancing, executing, accessing, and reporting

**Results**

**DSA human swarm interaction issues and challenges:**
- What is an appropriate level of interaction between humans and swarms?
- How can optimization for autonomy be balanced with optimization for collaboration and cooperation?
- How can controls, displays, and decision support be designed to support central control of a distributed system?
Resource Constrained Missions

Almost all spacecraft are resource constrained due to launch costs.

The DSA mission problem involves addressing collective behavior in the presence of resource constraints.

Examples include:
• choosing a ground station for downlink (e.g. satellite internet constellations)
• choosing which target to image (e.g. Earth observation)
• choosing when to collect data (e.g. atmospheric sampling)
Explorative Demonstration:

Pertinent Parameters

Explorative Reward: \( r^1 \)
Distance between swarm sat & GPS: \( D \)
Portion of \( D \) containing the EIA: \( d \)
Portion of \( d \) not detected by alt. channels/sats: \( x \)
# of swarm sats detecting the GPS channel: \( S \)
Total GPS channels for a given swarm satellite: \( C \)
Weighting Factor: \( \delta \)

Implementation Strategies

1. \( r_1 = D \); [Current Placeholding Formula]
2. \( r_1 = d \cdot \frac{1}{C} \cdot \frac{1}{S} \cdot \delta \)
3. \( r_1 = x \cdot \frac{1}{C} \cdot \frac{1}{S} \cdot \delta \)
4. \( r_1 = \begin{cases} d \cdot \frac{1}{C} \cdot \delta, & S = 1 \\ 0, & S > 1 \end{cases} \)

Ideal implementation strategy to be determined via simulation
Exploitative Demonstration:

Pertinent Parameters

Exploitative Reward: \( r^2 \)
Absolute Slant TEC: \( TEC \)
Predicted/True patch maximum: \( TEC_p \)
Predicted/True patch peak-to-peak STEC: \( \Delta TEC \)
Predicted/True patch peak-to-peak time: \( \Delta t \)
Probability of patch being detected: \( P(p) \)
Number of channels detecting the patch: \( N \)
Weighting Factor: \( \gamma \)

Implementation Strategies

1. \( r_2 = f(TEC) \); [Current Placeholder Formula]
2. \( r_2 = P(p) \cdot TEC_p \cdot N \cdot \gamma \)
3. \( r_2 = P(p) \cdot \Delta TEC \cdot N \cdot \gamma \)
4. \( r_2 = P(p) \cdot \Delta TEC / \Delta t \cdot N \cdot \gamma \)
5. \( r_2 = P(p) \cdot (\Delta TEC)^2 / \Delta t \cdot N \cdot \gamma \)

Ideal implementation strategy to be determined via simulation