a.i. solutions SmallSat Formation
Flying Testbed

CubeSat Workshop 2014
The Big Picture – Goals

• **Ultimate Goal:**
  - Create Onboard Autonomous Formation Flight Capability for Small Spacecraft

• **Goals Along The Way:**
  - Encapsulate existing knowledge and gain new knowledge in:
    - Formation Control
    - Attitude Control
    - Flight Software
The Big Picture – How We Are Getting There

• Developing a Closed-Loop Formation Flying Testbed to:
  – Simulate spacecraft dynamics for N-spacecraft
  – Simulate RTOS Flight Software (FSW) for N-spacecraft with realistic memory and processor constraints
  – Simulate inter-spacecraft communications
Potential Applications of …

• Closed-Loop Formation Flying Testbed:
  – Formation Design & Analysis
  – FSW Design & Testing

• Onboard Autonomous Formation Flight Capability for Small Spacecraft:
  – Inexpensive stereo imaging of objects of interest
    – Asteroids
    – Space Debris
    – Earth
  – Distributed spatial measurement experiments
    – Sparse-aperture telescopes
    – Gravity mapping
    – Magnetic field mapping
    – Lower Thermosphere/Upper Mesosphere atmospheric research
  – Advanced maneuvering
    – Autonomous collision avoidance
    – Autonomous docking
Dynamics Simulator (DSim)
Flight Software (FSW)
Shared Object Server (SOS)

R&D Activities 2014
DYNAMICS SIMULATOR (DSIM)
DSim Features

**Purpose:** DSim is a software application that enables simulation of rigid body dynamics with a task-based interface.

- Coded in Python
- Extensibility and Optimization with Cython
- Symbolic equations of motion (EoM)’s using SymPy
- Dynamic EoM’s with Kane’s method
- Task execution framework

```python
sc.task_manager.addTask(task_applySLMPIDControl, 'applySLMPIDControl')
```
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Scene
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Scene

SC with RWA Model
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object

- SC with RWA Mode

Scene
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Scene

SC Object

- SC with RWA Mode
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object

Scene

SC with RWA Mode
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object

Scene

- Earth
- Sun
- Planets

SC with RWA Mode
DSim Use Case – Simulate a Formation

Component Library
- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object
- SC with RWA Mode

Scene
- Earth
- Sun
- Planets
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object

SC with RWA Mode

Scene

Earth
Sun
Planets
Propagator
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Scene

SC Object

- SC with RWA Mode

Earth

Sun

Planets

Propagator
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors
- Actuators

Scene

SC Object

- SC with RWA Mode
- RWA
- IRU
- Controller
- TAM
- Estimator
- GPS

- Set initial state
- Define tasks

Earth
Sun
Planets
Propagator
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Scene

SC Object

(SC with RWA Mode) RWA
IRU
TAM
GPS

- Set initial state
- Define tasks

SC Object

(SC with RWA Mode) RWA
IRU
TAM
GPS

- Set initial state
- Define tasks

SC Object

(SC with RWA Mode) RWA
IRU
TAM
GPS

- Set initial state
- Define tasks

Earth
Sun
Planets

Propagator
DSim Use Case – Simulate a Formation

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

SC Object

- SC with RWA Mode
- RWA
- IRU
- TAM
- GPS
- Controller
- Estimator

• Set initial state
• Define tasks

SC Object

- SC with RWA Mode
- RWA
- IRU
- TAM
- GPS
- Controller
- Estimator

• Set initial state
• Define tasks

SC Object

- SC with RWA Mode
- RWA
- IRU
- TAM
- GPS
- Controller
- Estimator

• Set initial state
• Define tasks

Scene

Earth
Sun
Planets
Propagator

Run

ai-solutions.com
DSim Use Case – Simulate Inverted Pendulum

Component Library

- Scenes
- Bodies
- Controllers
- Estimators
- Dynamics Models
- Propagators
- Force Models
- Sensors Actuators

Body (Parent Class)

Inverted Pendulum (Custom Class)

Custom MSD Model

Controller

Propagator

Run
DSim Use Case – Simulate Inverted Pendulum

Uncontrolled Case

Controlled Case

Coordinates

Velocity

Uncontrolled Case

Controlled Case

Coordinates

Velocity
NAV FLIGHT SOFTWARE (FSW)
**NAV FSW – High Level Architecture**

**Purpose:** The NAV FSW is the navigation system prototype that is the first component of the GN&C flight software system.

- Overall design complete
- High level prototype working with estimation of simple harmonic oscillator
- Implemented in Python and running in Linux for prototyping purposes
SHARED OBJECT SERVER (SOS)
Purpose: The SOS is a networking architecture designed to enable communication between FSW instances.

- Powered by node.js
- Uses ZeroRPC for remote Python-to-Python communication
- Provides network visualization feature to graphically represent the formation
DSIM + FSW + SOS = FORMATION FLYING TESTBED
Formation Flying Testbed Vision

- **Goals:**
  - Simulate a formation of spacecraft by spawning N virtualized FSWs
  - Model the formation dynamics with DSim
  - Enable communication between nodes with SOS
Autonomous Operations
Swarm Dynamics
FSW Testing

Planned
R&D Activities 2015
Planned R&D Activities 2015

• Implement autonomous operations algorithms

• FSW in the Loop Testing

• Investigate Swarm Dynamics