

autoNGC

A Flight Software Framework for Autonomous
Navigation, Guidance and Control

Mike Romeo

autoNGC Flight Software Lead

NASA Goddard Space Flight Center



Why Autonomy?

- Reduced Operations Costs
- Reduced reliance on ground-based assets
 - Earth Independent Operations (EIO)
- Enable new mission capabilities
 - Distributed Systems Missions (DSM)
 - Communications Limited Missions



The autoNGC Project

- Enables autonomous Navigation, Guidance and Control to be run onboard as flight software using NASA's core Flight System (cFS)
- Designed for modularity and reuse through a “plug-and-play” architecture in both software and hardware
- Key Components of autoNGC Include:
 - Flight Software Framework
 - Simulation Testbed and Analysis Tools
 - Reference Hardware Platform



Reference Mission

- Currently targeting a lunar navigation scenario as reference mission for autoNGC development
- Incorporates multiple navigation measurement sources
 - GPS, including Weak-Signal
 - Terrain-Relative Nav (TRN)
 - Limb-based Optical Nav
 - 1-Way Direct to Earth (DTE)
 - 2-Way DTE
 - Accelerometer
 - Rubidium Clock



autoNGC Flight Software

- Implemented as a collection of cFS apps and libraries, including:
 - Hardware Interfaces
 - Navigation Library Interfaces
 - Ephemeris Library
 - Ephemeris Prediction
 - Fault Management
- Targeting NPR 7150.2D Class B
- Leverages existing NASA Goddard software products



core Flight System (cFS)



Goddard Enhanced Onboard Navigation
System (GEONS)



Goddard Image Analysis and Navigation
Tool (GIANT)

core Flight System (cFS)



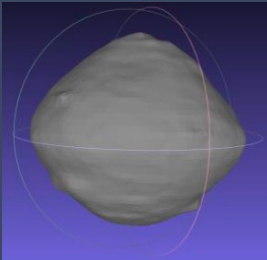
- Open-source NASA Goddard-developed flight software framework designed for extensive reuse with “plug-and-play” apps
 - Many existing NASA-developed apps available from open-source releases
- Include core Flight Executive (cFE), Operating System Abstraction Layer (OSAL) and Platform Support Package (PSP)
- Extensive heritage both inside and outside of NASA

Goddard Enhanced Onboard Navigation System (GEONS)

- Navigation Filter used by autoNGC
- Heritage on Terra, MMS, GPM and NICER/SEXTANT
- Supports multiple navigation measurement types
 - Nominal/Weak Signal GNSS
 - 1-Way/2-Way forward DTE
 - Crosslink
 - Celestial object bearing angles
 - Limb/Centroiding
 - Accelerometer
 - X-Ray Pulsar
- Includes ground tools, such as simulation and analysis capabilities



Goddard Image Analysis and Navigation Tool (GIANT)

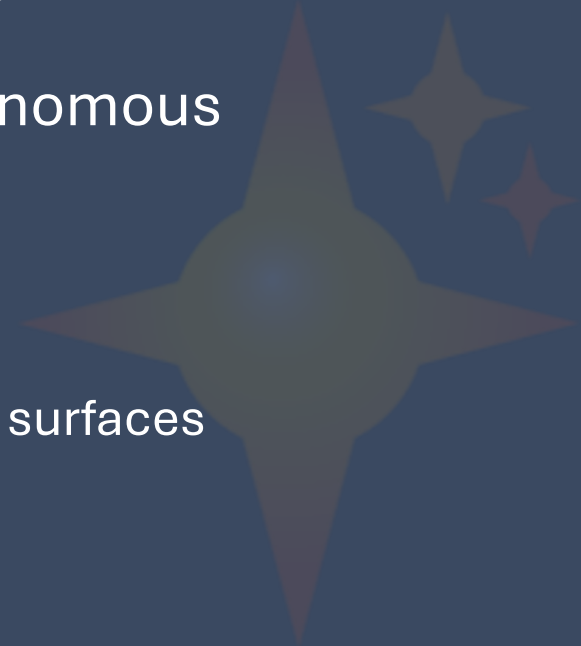


Shape Model of Bennu built autonomously using cGIANT



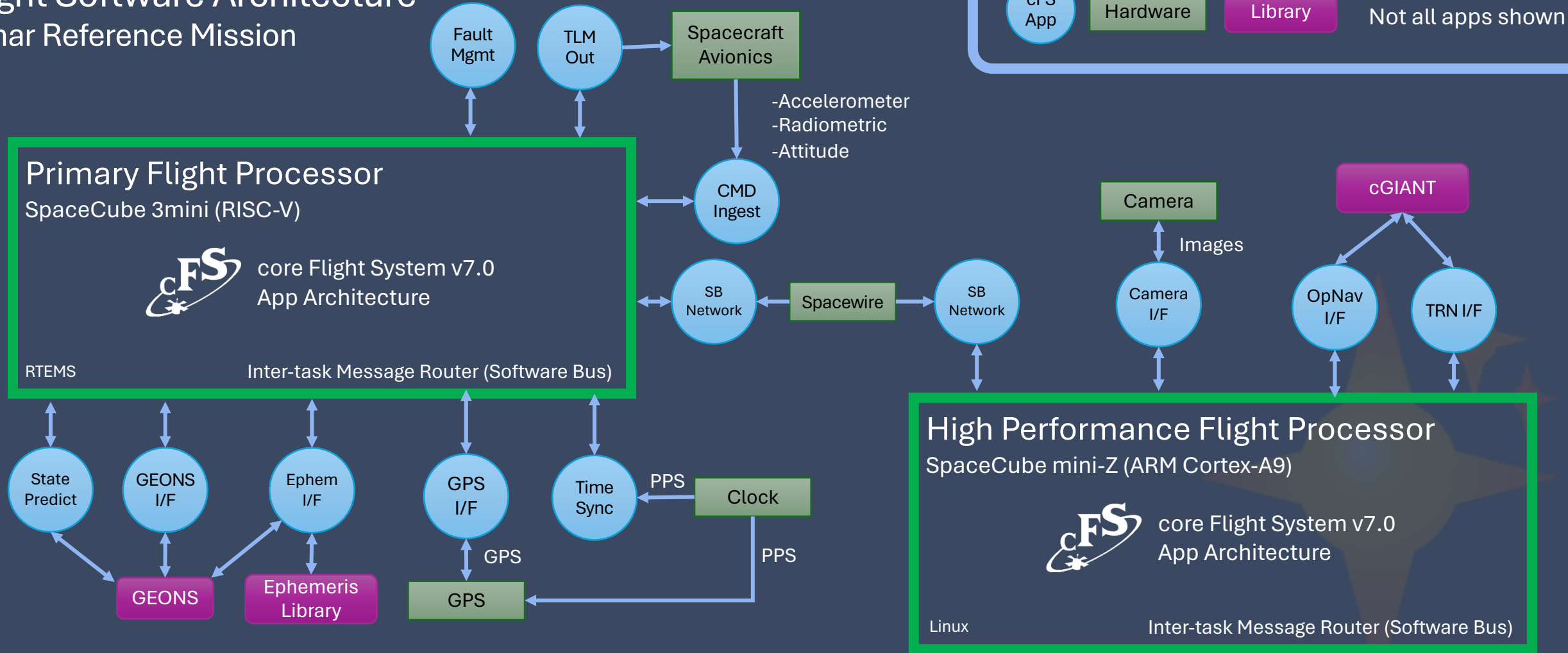
Moon limb observations extracted using cGIANT

- Implemented with state-of-the-art image processing algorithms for precision optical navigation
 - Ground heritage on OSIRIS-Rex
- Enables navigation relative to bodies whose trajectories and shapes are not well known
- cFS Giant (cGIANT) was created to enable autonomous onboard navigation
- Supports sub-pixel level measurements for:
 - Center-finding line of sight
 - Limb-based line of sight and range to target
 - Surface feature/terrain-relative navigation for known surfaces
 - Celestial navigation
 - Autonomous limb-based shape modeling



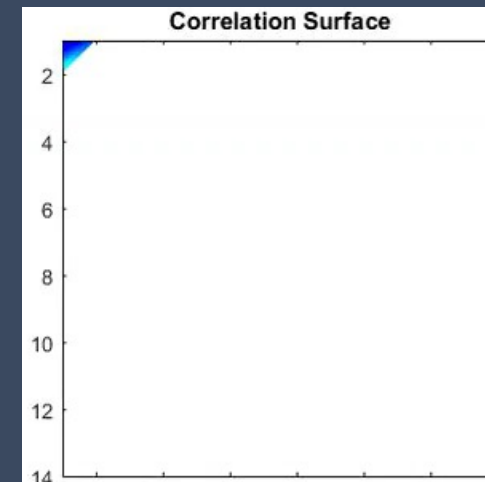
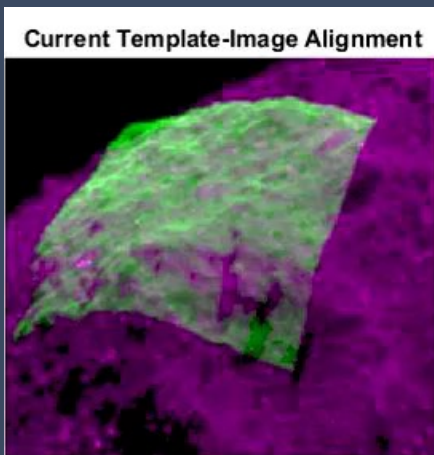
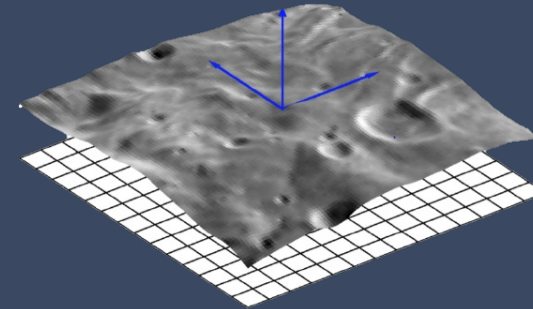
Flight Software Architecture

Lunar Reference Mission



Terrain-Relative Navigation (TRN)

- Images for TRN captured by onboard camera
- Uses where we think we are to predict what we think surface features should look like in an image and then locate the features in the image
- Surface maps provided by LuNaMaps
- Image processing done with cGIANT



Lunar DEM, Mapping, Modeling, and Validation (LuNaMaps)



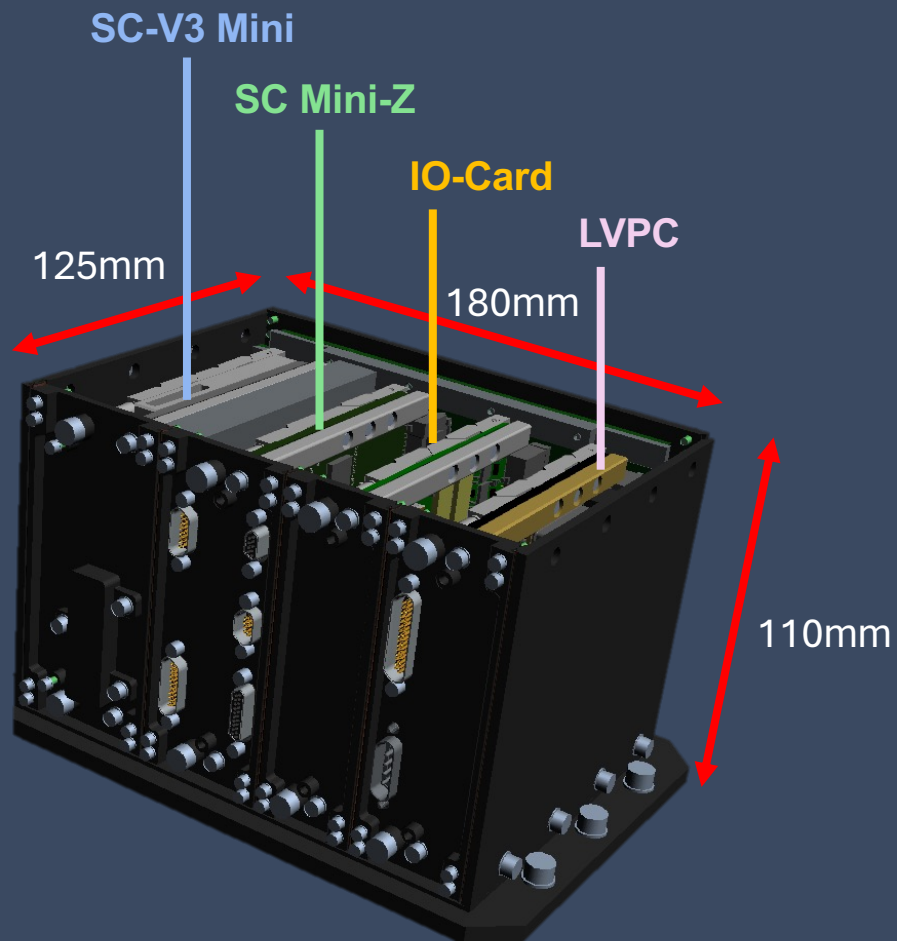
- NASA Space Technology Mission Directorate (STMD) Game Changing Development program focused on developing lunar maps suitable for Terrain-Relative Navigation (TRN) and Hazard Detection
- Digital Elevation Maps (DEMs) developed primarily with data from Lunar Reconnaissance Orbiter (LRO) instruments
- LuNaMaps team is currently working on releasing DEMs and software to build DEM products

autoNGC Reference Processors

- Leverages NASA's SpaceCube line of flight processing systems
- SpaceCube 3mini serves as the primary flight computer
 - RISC-V Soft CPU on the AMD Kintex FPGA
 - RTEMS
 - Prioritize reliability and uptime
- SpaceCube mini-Z serves as the high-performance flight computer
 - AMD Zynq-7020 FPGA with ARM Cortex-A9 Processor
 - Linux
 - Used for optical navigation image processing using cGIANT
 - Uses Software Bus Network (SBN) app over SpaceWire to send optical navigation measurements to GEONS navigation filter

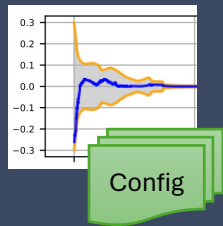


autoNGC Reference Hardware Design



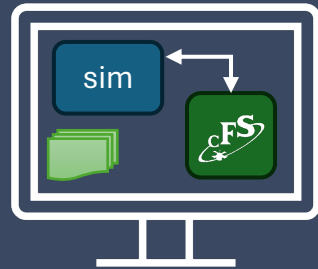
- Box Contains:
 - SpaceCube 3mini (primary processor)
 - SpaceCube mini-Z (high-performance processor)
 - Low Voltage Power Card (LVPC)
 - Configurable I/O Card
 - Backplane
- Mass of < 2.5 kg
- 10W idle, 15 W Nominal and 20 W Peak Power Usage

Simulation Pipeline



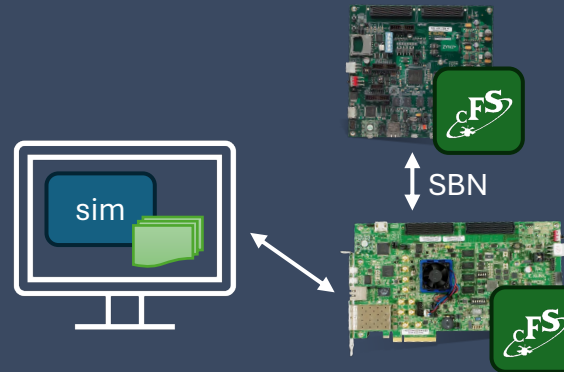
Filter Tuning and Configuration

- Uses MATLAB to tune GEONS for the set of measurement sources being tested
- Generates config files used in other simulations



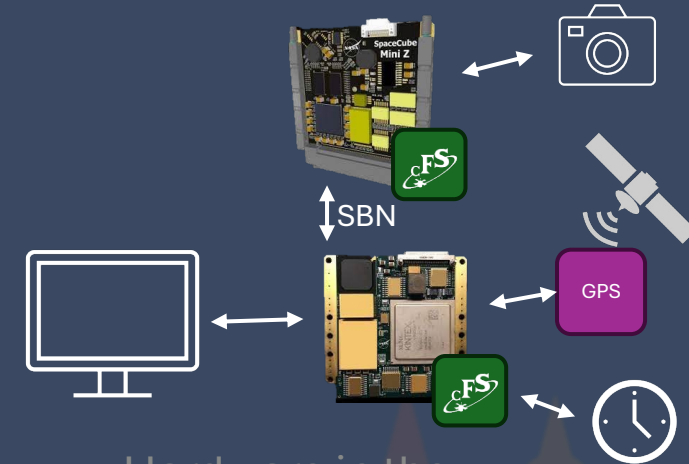
Software in the Loop

- Run simulation with cFS and autoNGC apps in the loop using configuration from filter tuning simulations
- Goal is to reproduce navigation performance results from MATLAB simulation



Processor in the Loop

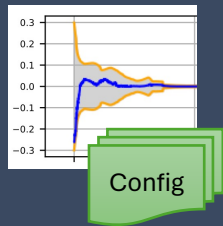
- Reproduce the software-in-the-loop simulation with cFS and autoNGC apps running on development boards
- Tests the performance of autoNGC on flight-like processors



Hardware in the Loop

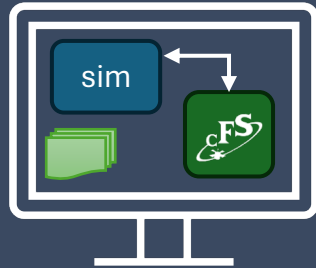
- Runs simulations on hardware ETUs using high fidelity navigation simulators
- Simulates autoNGC in a flight-like configuration

Simulation Pipeline



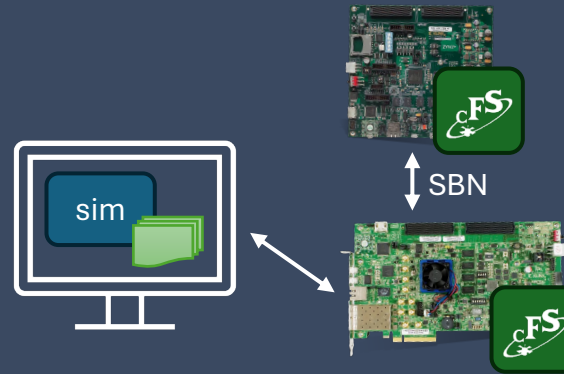
Filter Tuning and Configuration

- Uses MATLAB to tune GEONS for the set of measurement sources being tested
- Generates config files used in other simulations



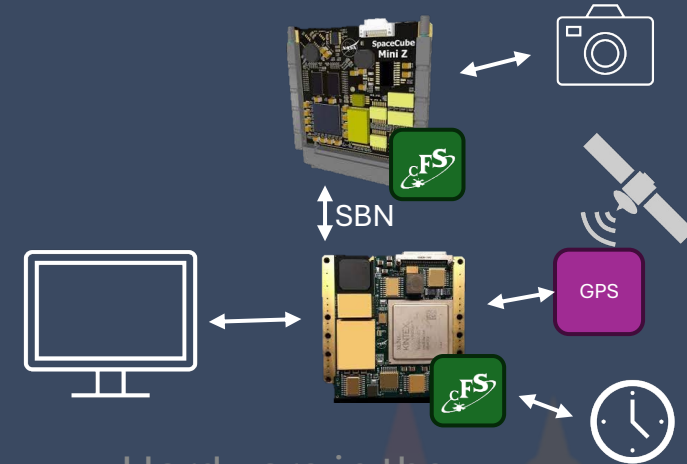
Software in the Loop

- Run simulation with cFS and autoNGC apps in the loop using configuration from filter tuning simulations
- Goal is to reproduce navigation performance results from MATLAB simulation



Processor in the Loop

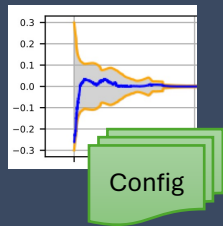
- Reproduce the software-in-the-loop simulation with cFS and autoNGC apps running on development boards
- Tests the performance of autoNGC on flight-like processors



Hardware in the Loop

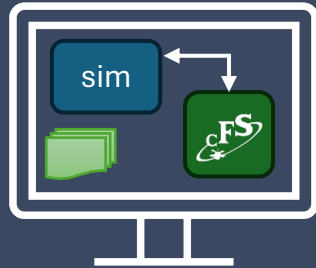
- Runs simulations on hardware ETUs using high fidelity navigation simulators
- Simulates autoNGC in a flight-like configuration

Simulation Pipeline



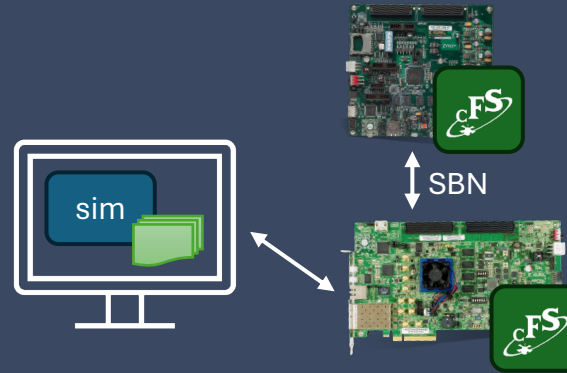
Filter Tuning and Configuration

- Uses MATLAB to tune GEONS for the set of measurement sources being tested
- Generates config files used in other simulations



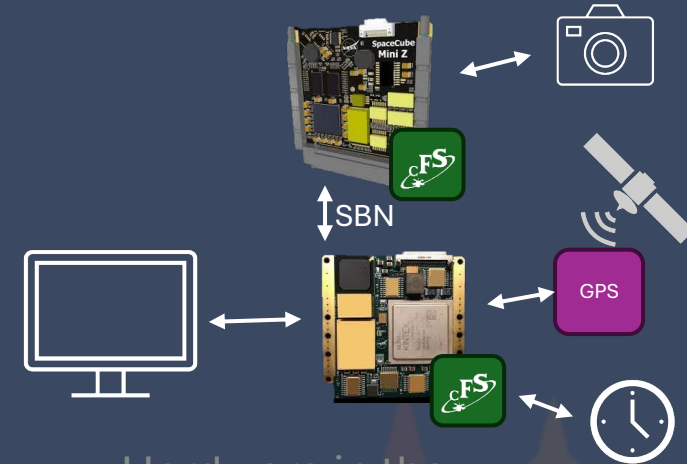
Software in the Loop

- Run simulation with cFS and autoNGC apps in the loop using configuration from filter tuning simulations
- Goal is to reproduce navigation performance results from MATLAB simulation



Processor in the Loop

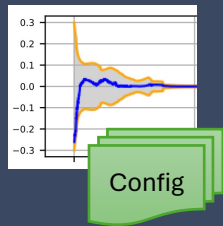
- Reproduce the software-in-the-loop simulation with cFS and autoNGC apps running on development boards
- Tests the performance of autoNGC on flight-like processors



Hardware in the Loop

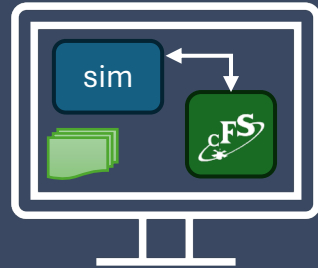
- Runs simulations on hardware ETUs using high fidelity navigation simulators
- Simulates autoNGC in a flight-like configuration

Simulation Pipeline



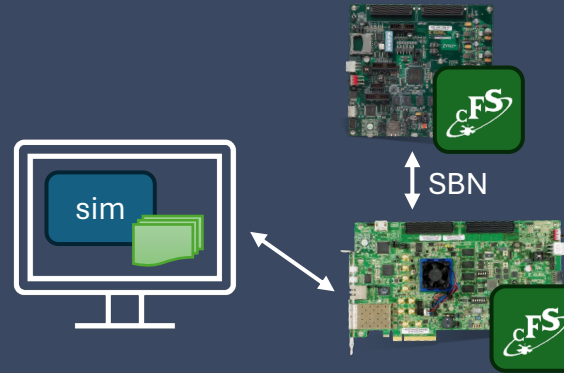
Filter Tuning and Configuration

- Uses MatLab to tune GEONS for the set of measurement sources being tested
- Generates config files used in other simulations



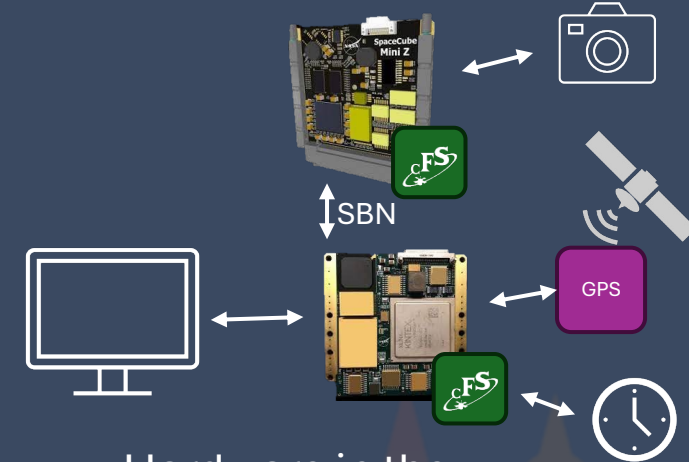
Software in the Loop

- Run simulation with cFS and autoNGC apps in the loop using configuration from filter tuning simulations
- Goal is to reproduce navigation performance results from MATLAB simulation



Processor in the Loop

- Reproduce the software-in-the-loop simulation with cFS and autoNGC apps running on development boards
- Tests the performance of autoNGC on flight-like processors



Hardware in the Loop

- Runs simulations on hardware ETUs using high fidelity navigation simulators
- Simulates autoNGC in a flight-like configuration

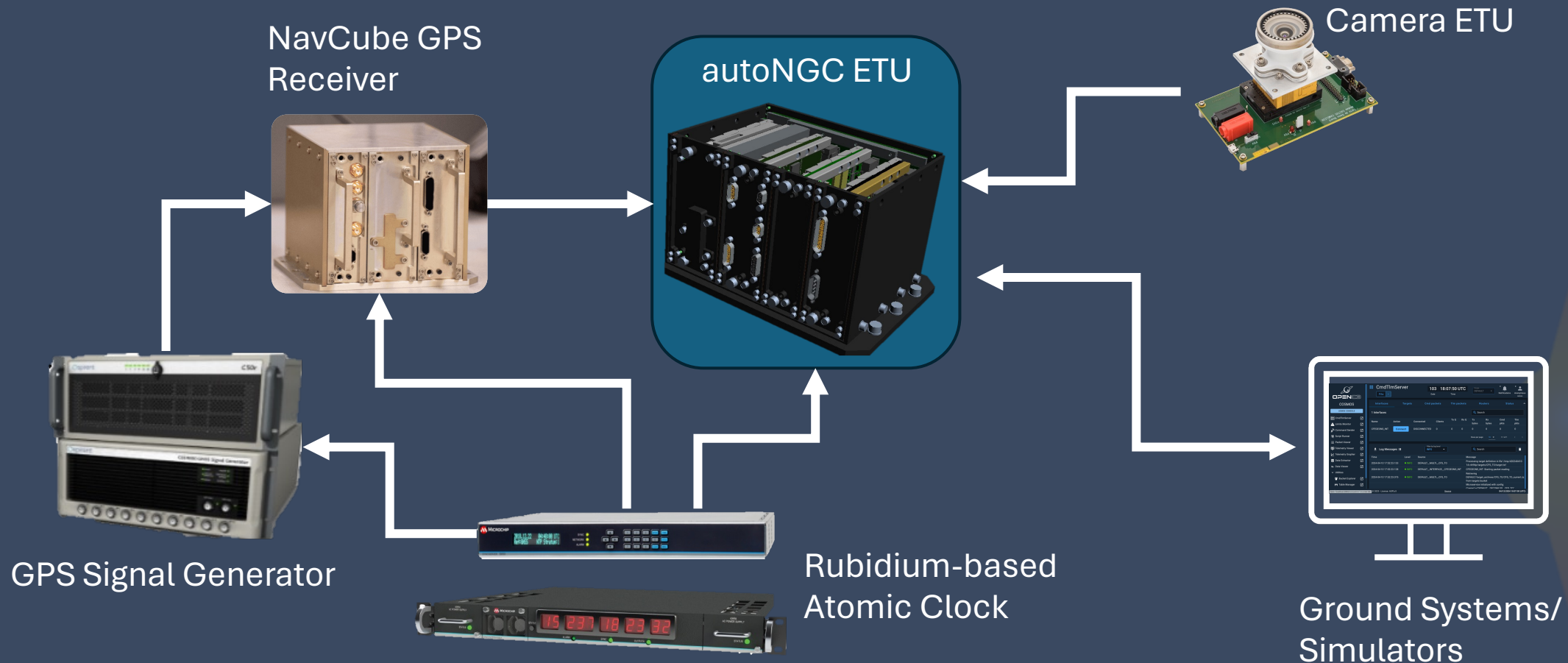
Simulation Environment

- Simulation Data
 - Orbit simulated using 42 and saved to flat files
 - TRN Data provided by LuNaMaps
 - GIANT used to generate synthetic moon images to simulate camera images
 - GPS measurements simulated using GEONS DataGen
- Simulation is driven by autoNGC-developed sim_all Python tool
 - Modular simulation tool developed as Python packages
 - Sends navigation measurements and commands to flight software via UDP CCSDS packets
 - Provides data to camera simulator
- Efforts to integrate autoNGC with NASA's Operational Simulator for Small Satellites (NOS3) currently underway

autoNGC Lab

- Located at NASA Goddard
- Focused on high-fidelity autoNGC simulations for TRL advancement
- Includes high-fidelity simulators for a variety of navigation sensors
 - GPS Simulation
 - Camera/Image simulation
 - Rubidium-based Atomic Clock
- Includes computing hardware to test on flight-like hardware
 - Development boards
 - Computing hardware ETUs

Hardware-in-the-Loop Simulations



autoNGC Release Roadmap

We are here

Research and Development

- Prototype and prove out the technologies
- Focus on experimentation at low TRL
- Multiple use-cases explored

TRL Advancement

- Reuse/rewrite apps and libraries following FSW standards
- Higher fidelity simulations
- Focus on lunar navigation reference mission

Potential Releases

- Explore release options for each component of autoNGC
- Open-source releases will be explored where possible
- Releases for government use will be explored for export-controlled components

autoNGC Release Roadmap

We are here

Research and Development

- Prototype and prove out the technologies
- Focus on experimentation at low TRL
- Multiple use-cases explored

TRL Advancement

- Reuse/rewrite apps and libraries following FSW standards
- Higher fidelity simulations
- Focus on lunar navigation reference mission

Potential Releases

- Explore release options for each component of autoNGC
- Open-source releases will be explored where possible
- Releases for government use will be explored for export-controlled components

autoNGC Release Roadmap

We are here

Research and Development

- Prototype and prove out the technologies
- Focus on experimentation at low TRL
- Multiple use-cases explored

TRL Advancement

- Reuse/rewrite apps and libraries following FSW standards
- Higher fidelity simulations
- Focus on lunar navigation reference mission

Potential Releases

- Explore release options for each component of autoNGC
- Open-source releases will be explored where possible
- Releases for government use will be explored for export-controlled components

What's Next

- Explore new capabilities and partnerships for autoNGC
- Explore flight demonstration opportunities
- Support more measurement types
 - LiDAR, Celestial Navigation, Crosslink, etc
- Onboard Guidance and Maneuver Planning Support
- Onboard Planning and Scheduling Capabilities



autoNGC Team

- Flight Software
 - Daniel Berdugo
 - Jaiden Ferraccioli
 - Liam Greenlee
 - Richard LeBoeuf
 - Andrew Liounis
 - Jose Martinez-Pedraza
 - Joseph Patton
 - Damon Stewart
 - Justin Trulear
- Leads
 - Sun Hur-Diaz, autoNGC PI/Navigation Lead <sun.h.hur-diaz@nasa.gov>
 - Behnam Azimi, Hardware Lead <behnam.azimi@nasa.gov>
 - Mike Romeo, Flight Software Lead <michael.j.romeo@nasa.gov>
- Testbed/Hardware
 - Nick Cason
 - Justin Clavette
 - Alan Cudmore
 - Kate Fowee Gasaway
 - Miguel Hernandez
 - Mohammadkhan Hossein
 - Kartik Pathak
 - Sean Semper
- Navigation
 - Grant Hecht
 - Nathan Stacey
 - Wayne Yu
- Flight Computing
 - Gary Crum
 - Sarah Dangelo
 - Sarah Flanagan
 - Justin Goodwill
 - James Marshall
 - Sebastian Sabogal
 - Dave Wilson
- Guidance and Maneuver Planning
 - Robert Pritchett
 - Ian Hammer