10/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

51 days before May 1st

Updates

- **Estimation**:
  - Studied MEKF resources and papers
  - MEKF for estimating attitude using IMU and sun sensor data
    - First-pass implementation completed
    - Sim integration and testing in progress
    - Recorded time stamped sensor data to use in MEKF implementation
- **FSW**
  - Architecture separation between the hardware interface layer (HAL) and the application layer
  - Configuration and state machine early architecture
  - Preliminary camera payload requirements

Blockers

Weekly Plan

- FSW development
  - PyCubed Time Distribution and Configuration
  - Jetson-PyCubed board communication
  - Sun Vector module (processing, calibration)
  - Camera interface improvement
  - Trained vision models ⇒ FSW implementation (vision)
- Estimation
  - Complete MEKF testing in Sim
  - Sun sensor calibration

Interface dependencies

- Mechanical:
  Inertia measurements for the updated CAD model
  - Need it now to update the sim

26/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

65 days before May 1st

Updates

- **FSW** development:
  - **Onboard File Storage**
    - Single interface to SD Card
    - File Management Services for every logging tasks
    - Interface for telemetry downlink
  - **Camera interface**
    - Read and configure all 6 cameras
    - Access to camera's status, latest image and live feed
    - Storage of time stamped images of each camera is created
  - **IMU interface**
    - Sample sensor at pre-set frequency
    - Support Moving Averaging Filter for smoothing the data
  - **Jetson-PyCubed Inter-communication**:
    - Complete protocol spec
    - Helper library built for packet parsing and creation
    - Initial implementation to read and send messages done
  - **State Machine Manager**
    - Preliminary design, implementation in progress

Blockers

Weekly Plan

- FSW development
  - PyCubed Configuration and Time Management & State Machines
  - Jetson-PyCubed command and control
  - Sun Vector module (processing, calibration)
  - Continue development on existing modules
- Estimation
  - MEKF for attitude estimation using IMU, magnetometer and sun sensor
  - Record time stamped sensor data to use in MEKF implementation.

Interface dependencies

- Mechanical:
  Inertia measurements for the updated CAD model, updated CAD of the CubeSat for test bed design

19/02/24

Demonstrating Visual-Inertial A&OD & On-Orbit Edge Computing

Progress summary

72 days before May 1st

Updates

- Vision:
  - Dataset download from 23 most salient regions
  - Landmark pruning for to identify ideal landmark size
  - Started hyperparameter tuning for LD
  - Looking into custom loss function focusing on pixel error
- Estimation integrated testing:
  - Generated camera vectors using landmark and satellite ground-truth through vector transformations
  - Tested batch optimizer to validate attitude estimation
  - Created small test setup to validate pixel to camera vector transformation
- Validation:
  - SIL environment setup between PyCubed and Simulation
  - First SIL test for detumbling control
  - Gyroscope noise analysis

Blockers

- Computing resources for LD training
  - ECE Community Compute Clusters
  - Pittsburgh Supercomputing Center
  - ROBO Cluster

Weekly Plan

- Vision
  - Continue training experiments with pruning
  - Continue dataset download
  - Tune hyperparameters and look into custom loss function
  - Working LD detector release by end of week
- Estimation
  - Continue development on batch optimiser
- FSW development
  - Work on PyCubed-Jetson communication
  - Finished PyCubed tasks for Alpha version

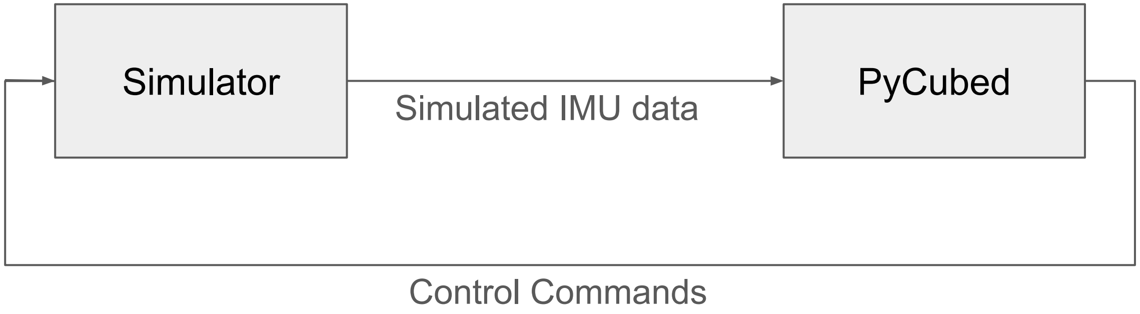
Interface dependencies

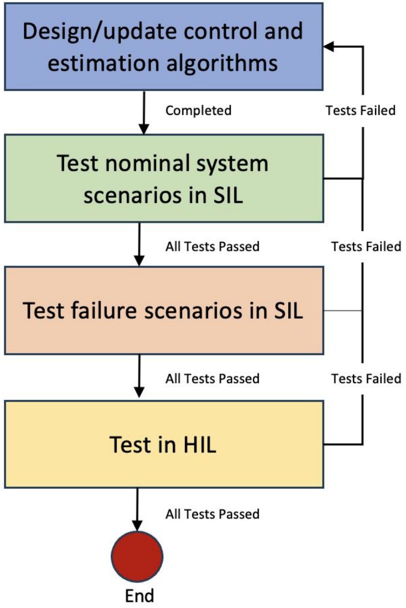
- Final CAD of the CubeSat for test bed design

Integrated testing

- Created small setup to test camera vector generation using image coordinates
- Verified transformation equations using actual x,y and depth information to find corresponding pixel coordinates and vice versa.
- Tested batch optimisation using groundtruth satellite ECEF and landmarks detected.
- Process
    - Get landmark lat long and convert to ECEF, get satellite groundtruth ECEF
    - Subtract the two vectors to get vector pointing from Landmark to satellite, invert to get vector from satellite to landmark
    - Convert this vector into camera frame to get camera vectors
    - Use ECI coordinates and these vectors as inputs to estimator

GNC Software Validation



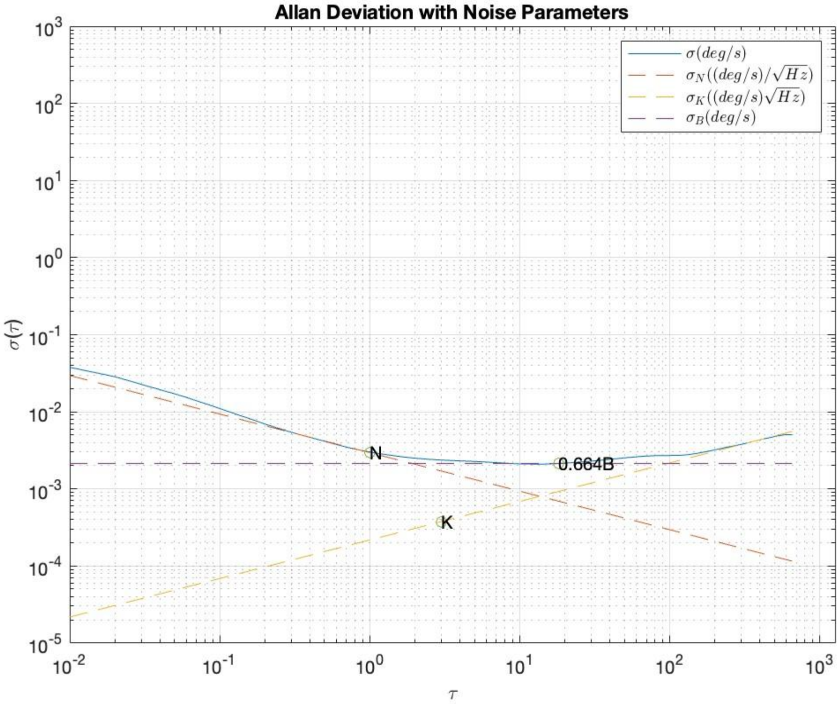


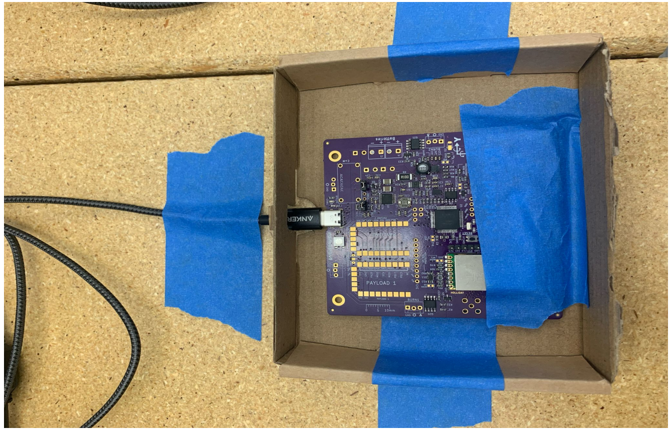
Software in Loop environment setup completed.

- Sensor readings generated in real-time on Simulator
- Readings are sent to PyCubed with Magnetic Control software running via Serial Communication
- Magnetic Control software processes the sensor readings and generates the control commands
- Simulator computes the next state for the satellite using the received control commands

Preliminary testing for Detumbling Control completed.

Gyroscope Noise Analysis





6 hours of datalogging of stationary IMU

- Allan variance for gyroscope noise parameter analysis, Parameters: N (angle random walk), K (rate random walk), B (bias instability)
- Helps us identify noise sources in stationary gyroscope data clusters
- Will be used for modelling the gyroscope accurately