National Aeronautics and Space Administration

Fast Kalman Filtering for Relative Spacecraft Position and Attitude Estimation for the Raven ISS Hosted Payload

Joseph M Galante, John Van Eepoel,
Chris D'Souza, and Bryan Patrick

Spacecraft Servicing

Want to service existing spacecraft:

- Inspect
- Repair
- Refuel
- Relocate

Existing spacecraft present navigation challenges:

- No laser retroreflectors
- No visual fiducials

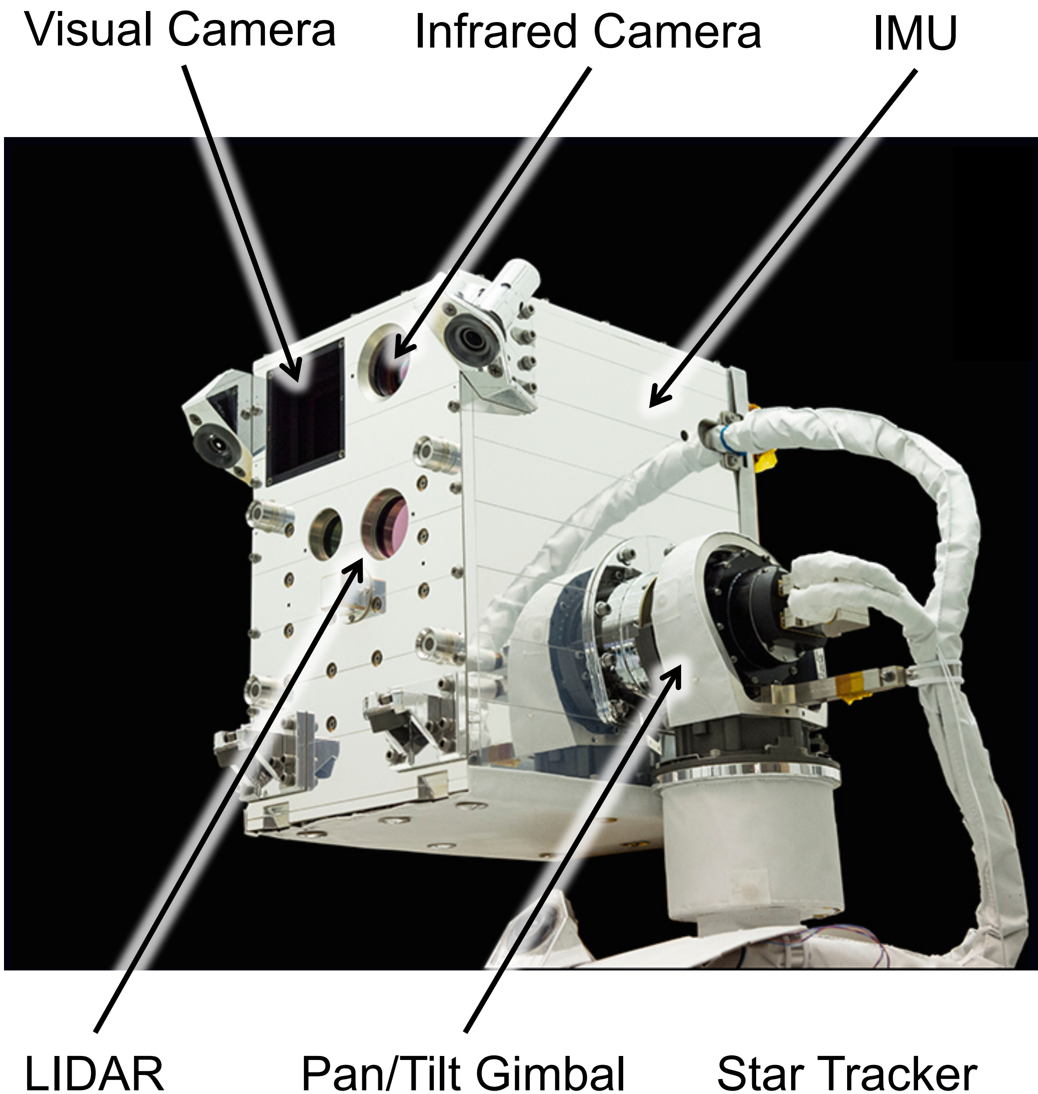
Unmanned servicing spacecraft must perform rendezvous and docking autonomously!

- Communication delays preclude ground control
- Must have accurate navigation solution with sufficient bandwidth for closed loop control

Notional robotic servicing operation rendering

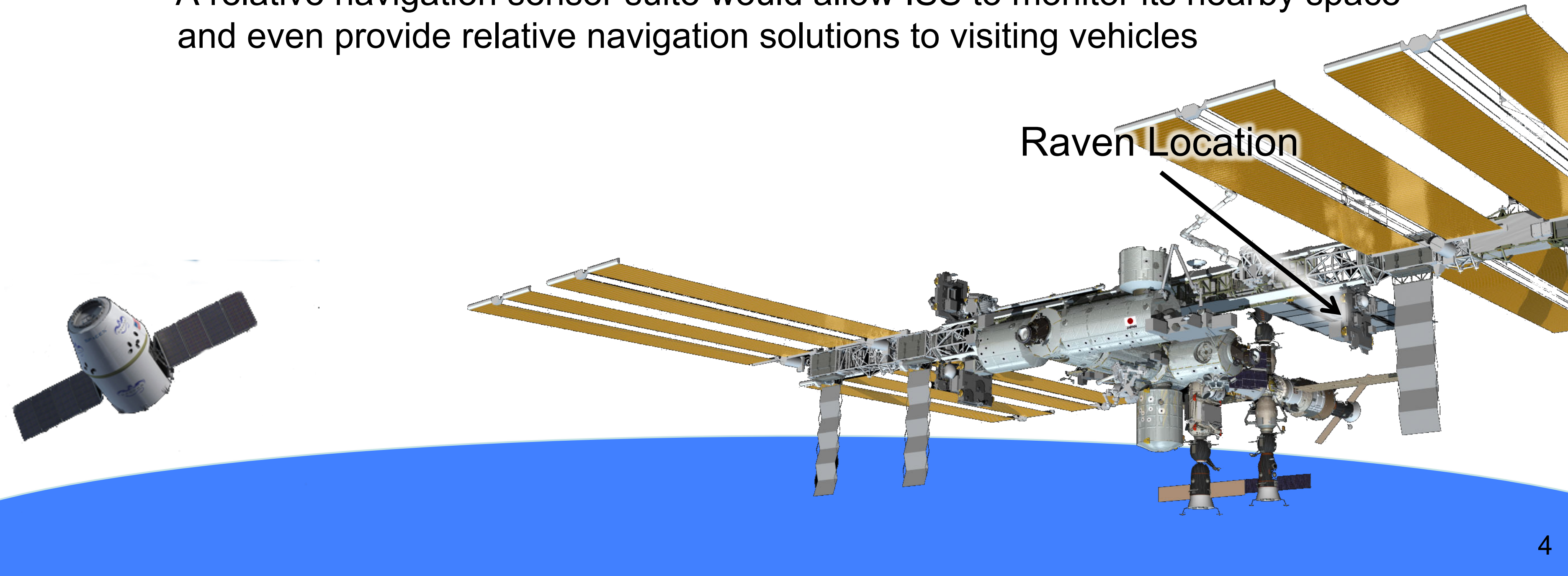
Raven: Relative Navigation Testbed

- ISS hosted payload
  - Anticipating June 2016 launch as part of the DoD Space Technology Program (STP-H5)
  - Mount on port nadir side of ISS
    - Next to solar array rotation joint
  - ISS provides power and comm
- Mission objectives
  - Track ISS resupply vehicles
  - Collect resupply vehicle imagery
    - Visual
    - Infrared
    - LIDAR
- Challenges
  - Command and telemetry outages require autonomous pan/tilt tracking
  - Raven gets no real-time data from ISS
    - No ISS navigation state
    - No GPS measurements
    - We DO get a clock pulse
  - 16 months from project authorization to hardware delivery



ISS Visiting Vehicle Operational Paradigm

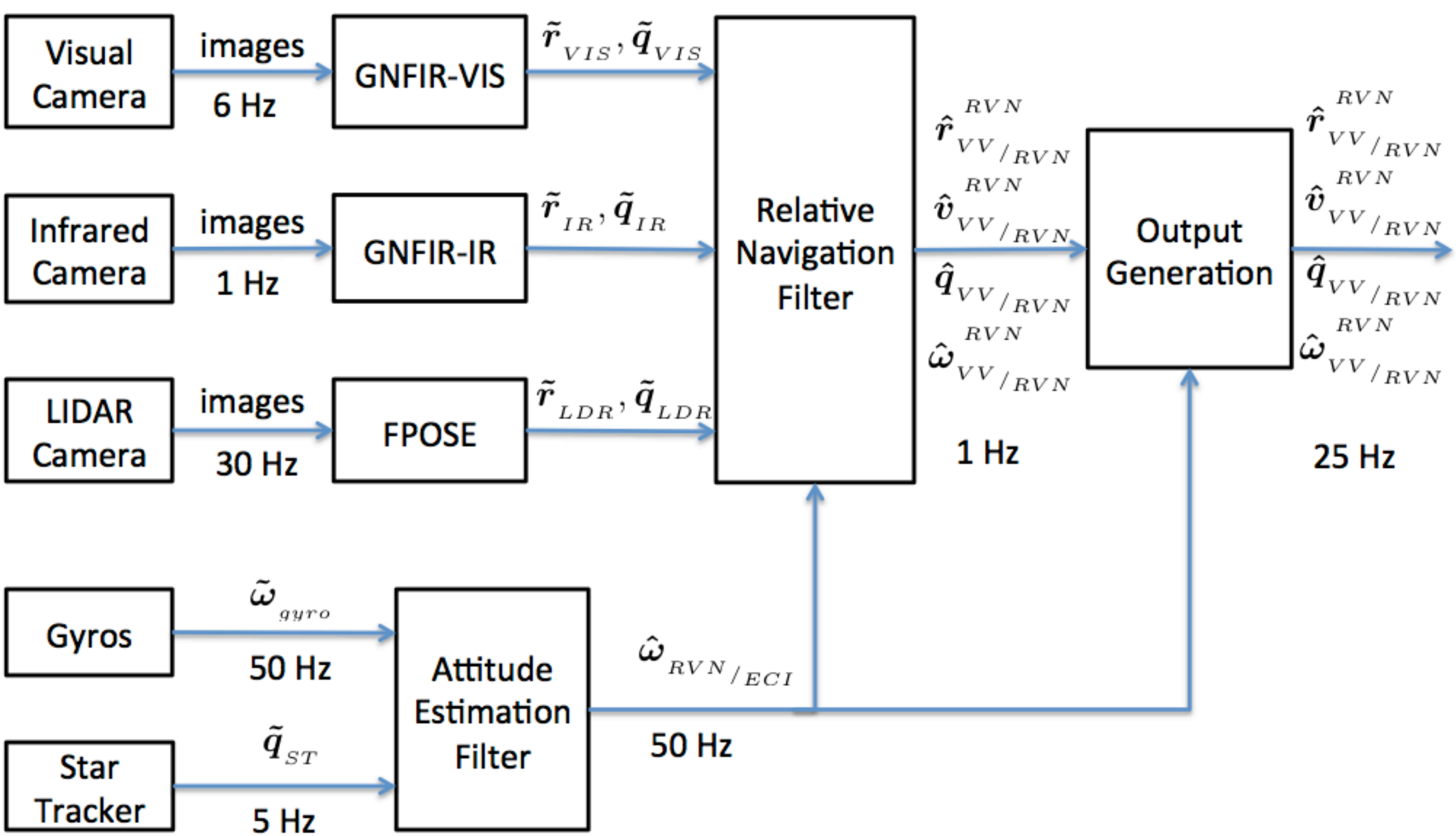
- Resupply vehicles provide relative navigation solution for their prox ops maneuvers, monitored by ISS mission control and ISS crew
- Resupply vehicles must use their own relative navigation sensor suite and associated computation, incurring cost and design complexity
- ISS does not produce its own relative navigation solution
- Raven is a prototype of a new paradigm:
  - Air traffic control uses local radars to monitor airspace
  - A relative navigation sensor suite would allow ISS to monitor its nearby space and even provide relative navigation solutions to visiting vehicles



Relative Navigation Filter (RNF) Overview

- Multiplicative Extended Kalman Filter (MEKF) formulation tracks relative pose = translation and orientation
  - MEKF formulation explicitly maintain quaternion constraints
  - Extension of MEKF to pose is similar to Junkins, Geller, Tweddle
- Raven includes a GSFC SpaceCube 2.0 flight processor
  - fast and powerful multi-core flight computer with FPGA
- Demanding filter rates
  - Pointing controller requires frequent filter estimate updates
  - Pose measurements from computer vision available at high rate
- Information available:
  - Relative pose from optical sensors
  - Inertial attitude and rate from star tracker and gyro
  - NO orbital information in real-time (neither ISS solutions nor raw GPS)
- Focus on what information is available
  - No orbital information precludes a Clohessy/Wiltshire or higher fidelity dynamics model
  - Relative pose measurements are frequent and well modeled
  - Account for camera rotation using star tracker and gyro (separate filter)

RNF Block Diagram



RNF Translation States

Filter State

$$\begin{bmatrix} {}^{RVN}\boldsymbol{r}_{VV/RVN} \\ {}^{RVN}\boldsymbol{v}_{VV/RVN} \\ \boldsymbol{q}_{VV/RVN} \\ {}^{VV}\boldsymbol{\omega}_{VV/ECI} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \\ \boldsymbol{b} \end{bmatrix}$$

Definitions

$$\boldsymbol{v} = \dot{\boldsymbol{r}} = {}^{RVN}\frac{d}{dt}\,{}^{RVN}\boldsymbol{r}_{VV/RVN}$$

$$\dot{\boldsymbol{v}} = \ddot{\boldsymbol{r}} = {}^{RVN}\frac{d^2}{dt^2}\,{}^{RVN}\boldsymbol{r}_{VV/RVN}$$

Dynamics

$$\begin{aligned} {}^{ECI}\frac{d^2}{dt^2}\,{}^{RVN}\boldsymbol{r}_{VV/RVN} &= \frac{1}{m_{VV}}\boldsymbol{F}_{VV} - \frac{1}{m_{ISS}}\boldsymbol{F}_{ISS} \\ &\approx W_{tran}\boldsymbol{w}_{tran} \qquad \sim \mathcal{N}\left(\boldsymbol{0}, W_{tran}W_{tran}^{T}\right) \end{aligned}$$

Kinematics

$$\begin{aligned} {}^{ECI}\frac{d^2}{dt^2}\,{}^{RVN}\boldsymbol{r}_{VV/RVN} &= \dot{\boldsymbol{v}} + \underbrace{\cancel{{}^{RVN}\boldsymbol{\alpha}_{RVN/ECI} \times \boldsymbol{r}}}_{\text{negligible}} + 2\,{}^{RVN}\boldsymbol{\omega}_{RVN/ECI} \times \boldsymbol{v} + \underbrace{\cancel{{}^{RVN}\boldsymbol{\omega}_{RVN/ECI} \times {}^{RVN}\boldsymbol{\omega}_{RVN/ECI} \times \boldsymbol{r}}}_{\text{negligible}} \\ &\approx \dot{\boldsymbol{v}} + 2\,{}^{RVN}\boldsymbol{\omega}_{RVN/ECI} \times \boldsymbol{v} \end{aligned}$$

Combining the above yields:

$$\dot{\boldsymbol{v}} \approx -2\,{}^{RVN}\boldsymbol{\omega}_{RVN/ECI} \times \boldsymbol{v} + W_{tran}\boldsymbol{w}_{tran}$$

RNF Rotation States

Filter State

$$\begin{bmatrix} \boldsymbol{r}^{RVN}_{VV/RVN} \\ \boldsymbol{v}^{RVN}_{VV/RVN} \\ \boldsymbol{q}_{VV/RVN} \\ \boldsymbol{\omega}^{VV}_{VV/ECI} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \\ \boldsymbol{b} \end{bmatrix}$$

Definition

$$\boldsymbol{q}_{VV/RVN} = \boldsymbol{q}_{VV/ECI} \otimes \boldsymbol{q}^{-1}_{RVN/ECI}$$

Kinematics

$$\begin{aligned} \dot{\boldsymbol{q}}_{VV/RVN} &= \frac{1}{2}\boldsymbol{\omega}^{VV}_{VV/ECI} \otimes \boldsymbol{q}_{VV/RVN} - \frac{1}{2}\boldsymbol{q}_{VV/RVN} \otimes \boldsymbol{\omega}^{RVN}_{RVN/ECI} \\ &= \frac{1}{2}\Big(\underbrace{\boldsymbol{\omega}^{VV}_{VV/ECI}}_{\text{filter state}} - R\left(\boldsymbol{q}_{VV/RVN}\right)\underbrace{\boldsymbol{\omega}^{RVN}_{RVN/ECI}}_{\substack{\text{assume known} \\ \text{(from attitude filter)}}}\Big) \otimes \boldsymbol{q}_{VV/RVN} \end{aligned}$$

Dynamics

$$\frac{{}^{ECI}d}{dt}\boldsymbol{\omega}^{VV}_{VV/ECI} = J^{-1}\left(\left(J\boldsymbol{\omega}^{VV}_{VV/ECI}\right) \times \boldsymbol{\omega}^{VV}_{VV/ECI}\right) + W_{rot}\boldsymbol{w}_{rot}$$

$$W_{rot}\boldsymbol{w}_{rot} \sim \mathcal{N}\left(\mathbf{0}, W_{rot}W^{T}_{rot}\right)$$

RNF Measurement Bias States

Filter State

$$\begin{bmatrix} \boldsymbol{r}^{RVN}_{VV/RVN} \\ \boldsymbol{v}^{RVN}_{VV/RVN} \\ \boldsymbol{q}_{VV/RVN} \\ \boldsymbol{\omega}^{VV}_{VV/ECI} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \\ \boldsymbol{b} \end{bmatrix}$$

Filter state is augmented with a bias for each sensor channel

$$\boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_{VIS,tran} \\ \boldsymbol{b}_{VIS,rot} \\ \boldsymbol{b}_{IR,tran} \\ \boldsymbol{b}_{IR,rot} \\ \boldsymbol{b}_{LDR,tran} \\ \boldsymbol{b}_{LDR,rot} \end{bmatrix}$$

Each sensor bias is assumed to be an independent first order Gauss Markov process

$$\dot{b}_j = -\frac{1}{\tau_j} b_j + \sigma_j w_j \qquad \sigma_j w_j \sim \mathcal{N}\left(0, \sigma_j^{\,2}\right)$$

Linearized Error State Dynamics

Filter State

$$\begin{bmatrix} \boldsymbol{r}^{RVN}_{VV/RVN} \\ \boldsymbol{v}^{RVN}_{VV/RVN} \\ \boldsymbol{q}_{VV/RVN} \\ \boldsymbol{\omega}^{VV}_{VV/ECI} \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} \\ \boldsymbol{v} \\ \boldsymbol{q} \\ \boldsymbol{\omega} \\ \boldsymbol{b} \end{bmatrix}$$

Linearized Error State

$$\Delta \boldsymbol{x} = \begin{bmatrix} \Delta \boldsymbol{r} \\ \Delta \boldsymbol{v} \\ \Delta \boldsymbol{q} \\ \Delta \boldsymbol{\omega} \\ \Delta \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{r} - \hat{\boldsymbol{r}} \\ \boldsymbol{v} - \hat{\boldsymbol{v}} \\ \boldsymbol{g}\left(\boldsymbol{q} \otimes \hat{\boldsymbol{q}}^{-1}\right) \\ \boldsymbol{\omega} - \hat{\boldsymbol{\omega}} \\ \boldsymbol{b} - \hat{\boldsymbol{b}} \end{bmatrix}$$

Linearized Error State Dynamics derived in paper (linear time varying system)

$$\Delta \dot{\boldsymbol{x}} = F \Delta \boldsymbol{x} + W \boldsymbol{w}$$

First order approximation used to compute error state transition matrix

$$\begin{aligned} \Phi(\Delta t) &= \mathbb{I} + \Delta t F + \frac{\Delta t^2}{2!} F^2 + \dots \\ &\approx \mathbb{I} + \Delta t F \end{aligned}$$

Process noise matrix preserves kinematic constraints

$$Q(\Delta t) = E\left\{ \left[\int_{t-\Delta t}^{t} \Phi\left(t - \epsilon\right) W \boldsymbol{w}(\epsilon) d\epsilon \right] \left[\int_{t-\Delta t}^{t} \Phi\left(t - \eta\right) W \boldsymbol{w}(\eta) d\eta \right]^T \right\}$$

Translation Measurement Component

Pose measurements from sensor CAM are denoted $\left(\tilde{\boldsymbol{r}}^{VV}_{VV/RVN,CAM}, \tilde{\boldsymbol{q}}_{VV'/RVN,CAM}\right)$

The translation component is modeled as:

$$\tilde{\boldsymbol{r}}^{VV}_{VV/RVN,CAM} = \boldsymbol{r}^{VV}_{VV/RVN} + \boldsymbol{b}_{CAM,tran} + M_{CAM,tran}\boldsymbol{m}_{CAM,tran}$$

measurement true FOGM bias Gaussian white noise

Where the First Order Gauss Markov Bias is as given before:

$$\dot{\boldsymbol{b}}_{CAM,tran} = -\begin{bmatrix} 1/\tau_1 & & \\ & 1/\tau_2 & \\ & & 1/\tau_3 \end{bmatrix} \boldsymbol{b}_{CAM,tran} + \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \sigma_3 \end{bmatrix} \boldsymbol{w}_{b,CAM,tran}$$

Resulting in the translation component innovation:

$$\begin{aligned} \Delta\boldsymbol{r}^{innov}_{CAM} &= \tilde{\boldsymbol{r}}^{VV}_{VV/RVN,CAM} - \hat{\boldsymbol{r}} - \hat{\boldsymbol{b}}_{CAM,tran} \\ &= \Delta\boldsymbol{r} + \Delta\boldsymbol{b}_{CAM,tran} + M_{CAM,tran}\boldsymbol{m}_{CAM,tran} \end{aligned}$$

Rotation Measurement Component

Pose measurements from sensor CAM are denoted $\left(\tilde{\boldsymbol{r}}^{VV}_{VV/_{RVN},CAM}, \tilde{\boldsymbol{q}}_{VV'/_{RVN},CAM}\right)$

The rotation component is modeled as:

$$\tilde{\boldsymbol{q}}_{VV'/_{RVN},CAM} = \boldsymbol{q}\left(\boldsymbol{b}_{CAM,rot} + M_{CAM,rot}\boldsymbol{m}_{CAM,rot}\right) \otimes \boldsymbol{q}_{VV/_{RVN}}$$

measurement FOGM bias Gaussian white noise true

Where the First Order Gauss Markov Bias is as given before:

$$\dot{\boldsymbol{b}}_{CAM,rot} = -\begin{bmatrix} 1/\tau_4 & & \\ & 1/\tau_5 & \\ & & 1/\tau_6 \end{bmatrix} \boldsymbol{b}_{CAM,rot} + \begin{bmatrix} \sigma_4 & & \\ & \sigma_5 & \\ & & \sigma_6 \end{bmatrix} \boldsymbol{w}_{b,CAM,rot}$$

The orientation component innovation is a bit more involved:

$$\begin{aligned} \Delta\boldsymbol{q}^{innov}_{CAM} &= \boldsymbol{q}^{-1}\left(\hat{\boldsymbol{b}}_{CAM,rot}\right) \otimes \tilde{\boldsymbol{q}}_{VV'/_{RVN},CAM} \otimes \hat{\boldsymbol{q}}^{-1}_{VV/_{RVN}} \\ &= \boldsymbol{q}^{-1}\left(\hat{\boldsymbol{b}}_{CAM,rot}\right) \otimes \boldsymbol{q}\left(\boldsymbol{b}_{CAM,rot} + M_{CAM,rot}\boldsymbol{m}_{CAM,rot}\right) \otimes \Delta\boldsymbol{q} \end{aligned}$$

$$\Delta\boldsymbol{g}^{innov}_{CAM} = \boldsymbol{g}\left(\Delta\boldsymbol{q}^{innov}_{CAM}\right) \approx \Delta\boldsymbol{g} - \Delta\boldsymbol{b}_{CAM} + M_{CAM,rot}\boldsymbol{m}_{CAM,rot}$$

EKF Measurement Update Procedure

for k=*1,2,3,...* **do**

> **Propagate State Estimate to Measurement Time**
>
> $\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1}^+ + \int_{t_{k-1}}^{t_k} \boldsymbol{f}(\boldsymbol{x}(\tau), \tau)\, d\tau$
>
> (note $\Delta\hat{\boldsymbol{x}}_{k-1}^+ = \mathbf{0}$, so $\Delta\hat{\boldsymbol{x}}_k^- = \Phi_k \Delta\hat{\boldsymbol{x}}_{k-1}^+ = \mathbf{0}$)
>
> **Propagate State Covariance to Measurement Time**
>
> compute Φ_k and Q_k
>
> $P_k^- = \Phi_k P_{k-1}^+ \Phi_k^T + Q_k$
>
> **Perform Measurement Update**
>
> compute H_k
>
> $K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k \right)^{-1}$
>
> $P_k^+ = \left(I - K_k H_k \right) P_k^- \left(I - K_k H_k \right)^T + K_k R_k K_k^T$
>
> $\Delta\hat{\boldsymbol{x}}_k^+ = \Delta\hat{\boldsymbol{x}}_k^- + K_k \left(y_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-, t_k \right) \right)$
>
> **Transfer Information to State Estimate, Reset Error Estimate**
>
> $\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \Delta\hat{\boldsymbol{x}}_k^+$
>
> $\Delta\hat{\boldsymbol{x}}_k^+ = \mathbf{0}$

end

EKF Measurement Update Procedure

for k=*1,2,3,...* **do**

Propagate State Estimate to Measurement Time

$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k-1}^+ + \int_{t_{k-1}}^{t_k} \boldsymbol{f}(\boldsymbol{x}(\tau),\tau)\,d\tau$

(note $\Delta\hat{\boldsymbol{x}}_{k-1}^+ = \mathbf{0}$, so $\Delta\hat{\boldsymbol{x}}_k^- = \Phi_k \Delta\hat{\boldsymbol{x}}_{k-1}^+ = \mathbf{0}$)

Propagate State Covariance to Measurement Time

compute Φ_k and Q_k

$P_k^- = \Phi_k P_{k-1}^+ \Phi_k^T + Q_k$

Perform Measurement Update

compute H_k

$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + R_k\right)^{-1}$

$P_k^+ = \left(I - K_k H_k\right) P_k^- \left(I - K_k H_k\right)^T + K_k R_k K_k^T$

$\Delta\hat{\boldsymbol{x}}_k^+ = \Delta\hat{\boldsymbol{x}}_k^- + K_k\left(y_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-, t_k\right)\right)$

Transfer Information to State Estimate, Reset Error Estimate

$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + \Delta\hat{\boldsymbol{x}}_k^+$

$\Delta\hat{\boldsymbol{x}}_k^+ = \mathbf{0}$

end

equivalent

$$\hat{\boldsymbol{x}}_k^+ = \hat{\boldsymbol{x}}_k^- + K_k\left(\boldsymbol{y}_k - \boldsymbol{h}\left(\hat{\boldsymbol{x}}_k^-, t_k\right)\right)$$

MEKF Measurement Update Procedure

Compute Kalman Gain

$$K_k = P_k^- H_{VIS}^T \left(H_{vis} P_k^- H_{VIS}^T + R_{VIS} \right)^{-1}$$

Update Covariance

$$P_k^+ = \left(\mathbb{I}_{30\times30} - K_k H_{VIS} \right) P_k^- \left(\mathbb{I}_{30\times30} - K_k H_{VIS} \right)^T + K_k R_{VIS} K_k^T$$

Compute State Estimate Update

$$\Delta \boldsymbol{x}^{update} = \begin{bmatrix} \Delta \boldsymbol{r}^{update} \\ \Delta \boldsymbol{v}^{update} \\ \Delta \boldsymbol{g}^{update} \\ \Delta \boldsymbol{\omega}^{update} \\ \Delta \boldsymbol{b}^{update} \end{bmatrix} = K_k \begin{bmatrix} \Delta \boldsymbol{r}_{VIS}^{innov} \\ \Delta \boldsymbol{g}_{VIS}^{innov} \end{bmatrix}$$

Apply Update to State Estimate

$$\hat{\boldsymbol{r}}_k^+ = \hat{\boldsymbol{r}}_k^- + \Delta \boldsymbol{r}^{update}$$

$$\hat{\boldsymbol{v}}_k^+ = \hat{\boldsymbol{v}}_k^- + \Delta \boldsymbol{v}^{update}$$

$$\hat{\boldsymbol{q}}_k^+ = \boldsymbol{q}\left(\Delta \boldsymbol{g}^{update} \right) \otimes \hat{\boldsymbol{q}}_k^-$$

$$\hat{\boldsymbol{\omega}}_k^+ = \hat{\boldsymbol{\omega}}_k^- + \Delta \boldsymbol{\omega}^{update}$$

$$\hat{\boldsymbol{b}}_k^+ = \hat{\boldsymbol{b}}_k^- + \Delta \boldsymbol{b}^{update}$$

- Same paradigm as the (cumbersome version) of an EKF
- Quaternion update makes this an MEKF a la Lefferts, Markley, and Shuster
- Inclusion of translation states in an MEKF already in the literature
  - Kim, Crassidis, Cheng, Fosbury, Junkins
  - Woffinden, Geller
  - Tweddle, Saenz-Otero
- We augment with biases

Observability Issue

- Can’t instantaneously solve for all sensor biases AND relative pose

Observability Issue

- Can't instantaneously solve for all sensor biases AND relative pose

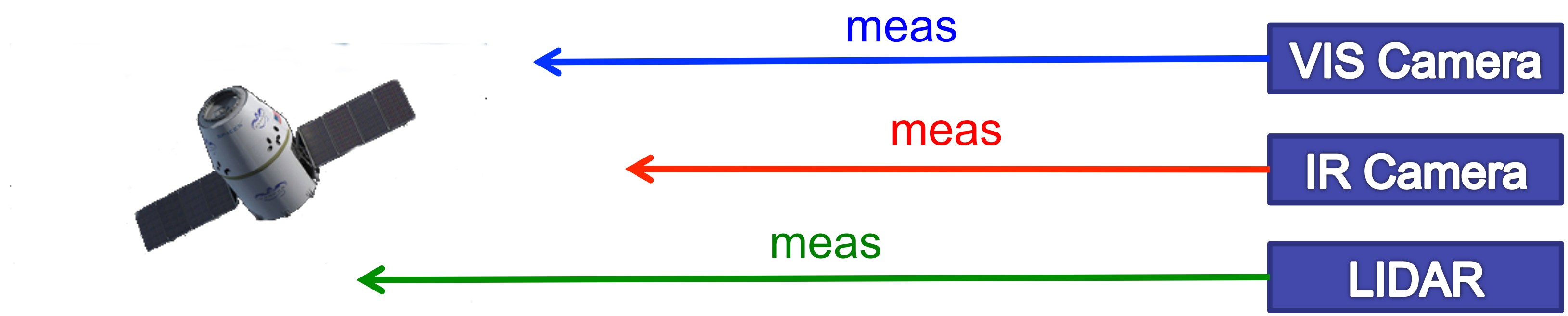
VIS Camera

IR Camera

LIDAR

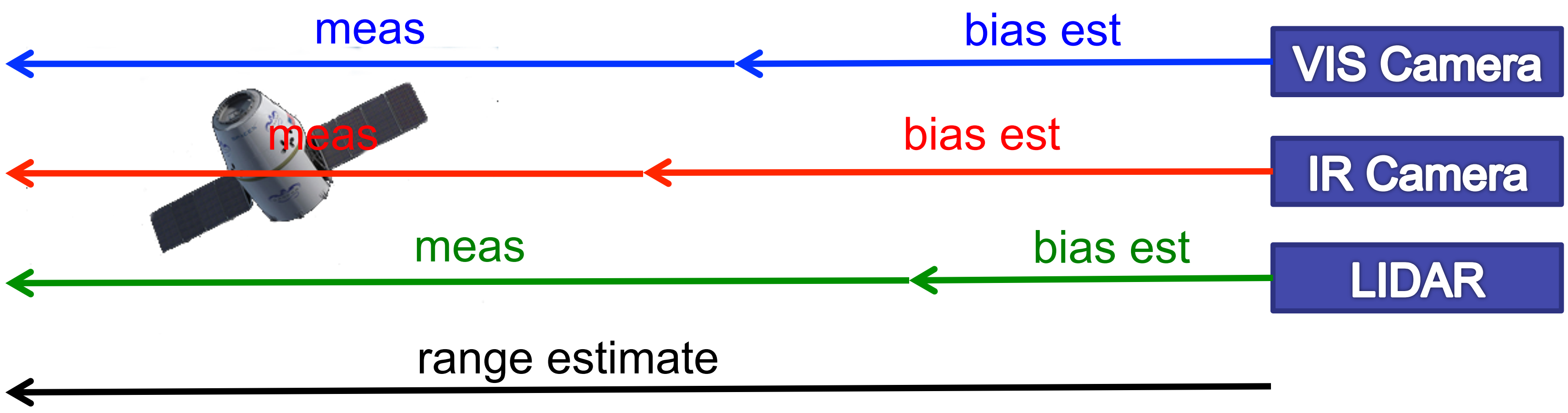
Observability Issue

- Can’t instantaneously solve for all sensor biases AND relative pose



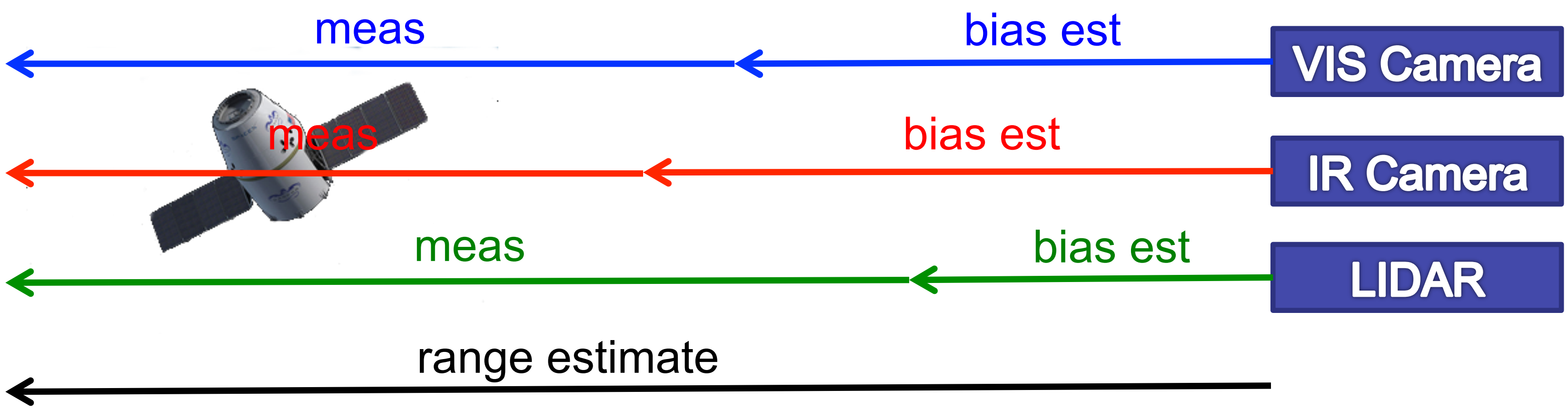
Observability Issue

- Can't instantaneously solve for all sensor biases AND relative pose



Observability Issue

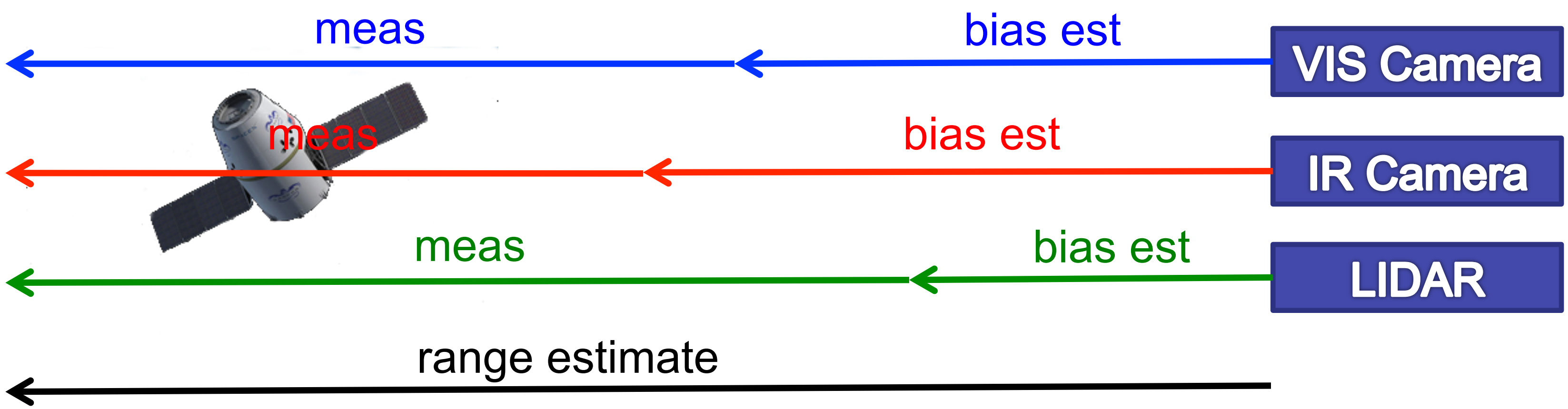
- Can't instantaneously solve for all sensor biases AND relative pose



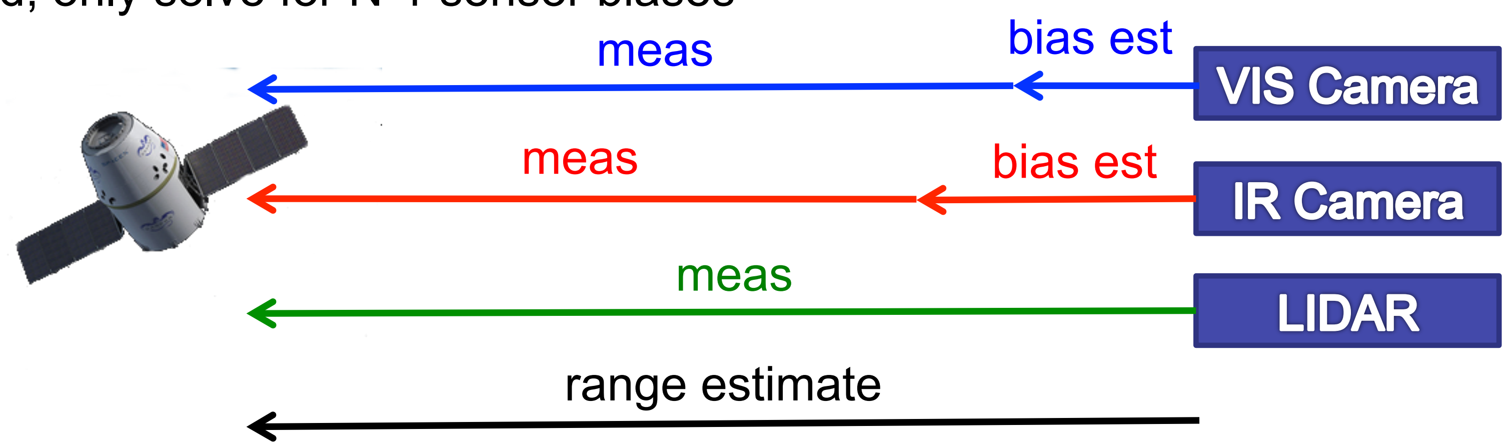
- Relative dynamics aren't "rich enough" to correctly solve over time
- Instead, only solve for N-1 sensor biases

Observability Issue Resolved

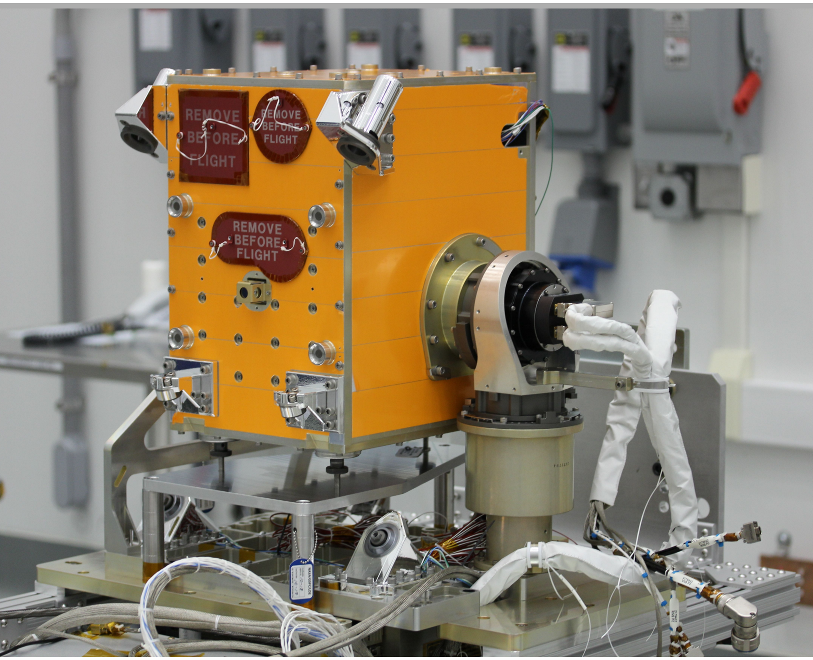
- Can't instantaneously solve for all sensor biases AND relative pose

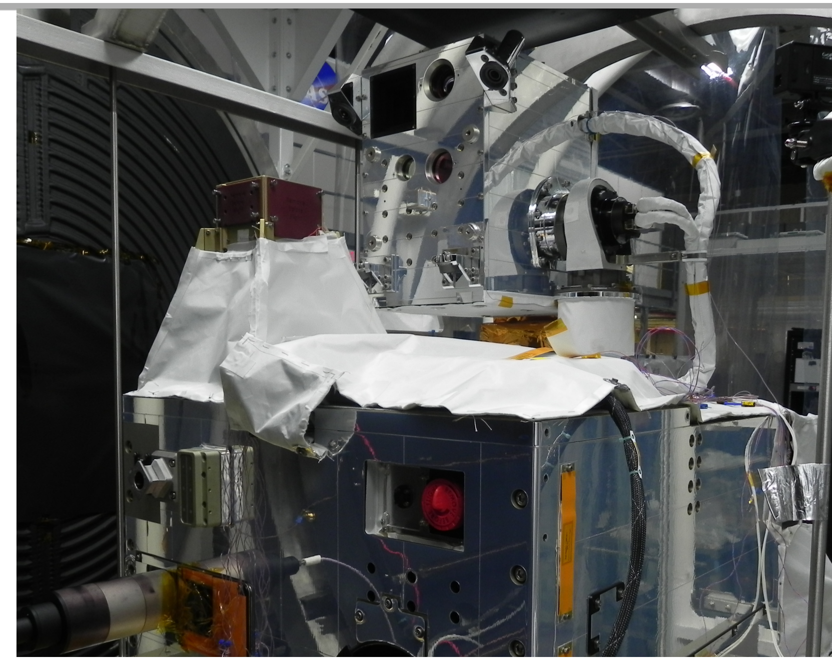


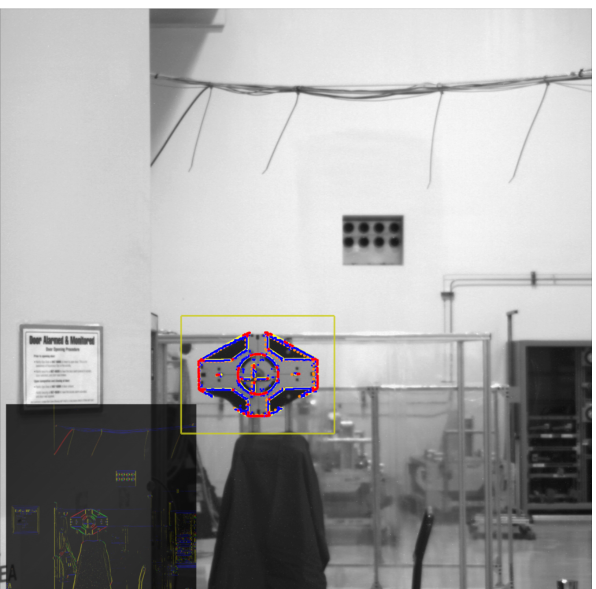
- Relative dynamics aren't "rich enough" to correctly solve over time
- Instead, only solve for N-1 sensor biases



Raven Development Montage







Raven_dragon_rend_gnfirTrack_fov35.mp4 goes here, shows GNFIR tracking synthetic imagery of SpaceX Dragon in complicated lighting

Questions?

- Raven_fsp_CDR_viscam.mp4 goes here
  - Freespace rendering showing third person perspective as well as Raven perspective
  - SpaceX Dragon rendezvous
  - Synthetic imagery shows ISS shadow on Dragon