National Aeronautics and Space Administration

Simultaneous Localization and Mapping for Satellite Rendezvous and Proximity Operations Using Random Finite Sets

Lauren Schlenker & Mark Moretto

David Gaylor, Richard Linares

Code 590

NASA Goddard Space Flight Center

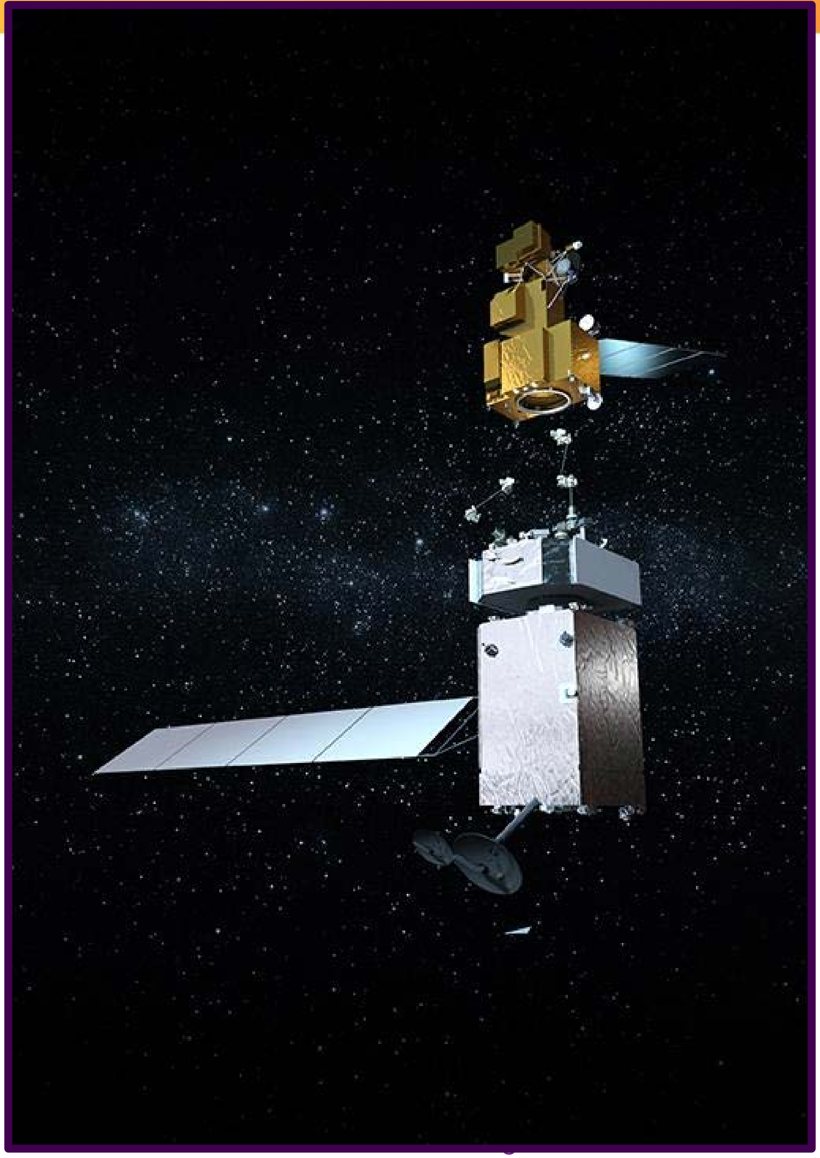
MISSION ENGINEERING AND SYSTEMS ANALYSIS DIVISION

Summary:

- **Autonomous spacecraft relative navigation:** a necessary requirement of future exploration/servicing missions.
- **Challenging environments:** no *a priori* map, need to track multiple features using measurements that can be noisy, have extraneous measurements (clutter), and missed detections.
- **Random Finite Set based filters:** a recent development, specifically formulated for these kinds of problems.
- **Initial simulations:** *first known demonstration* of a Random Finite Set based filter for spacecraft pose estimation and mapping.

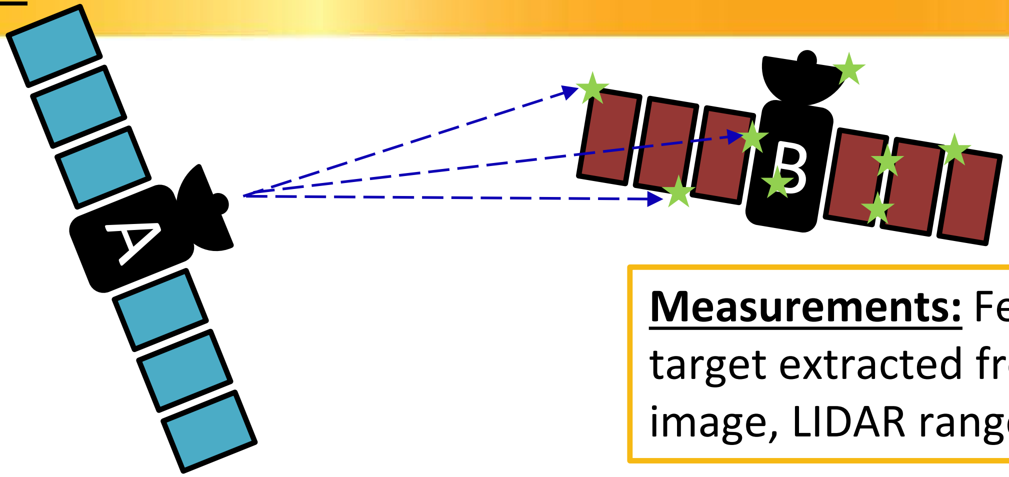
- **<u>Future of space exploration missions</u>:** spacecraft must be able to *autonomously* navigate their environment.
  - Rendezvous with non-cooperative satellite (satellite servicing).
  - Satellite swarms/formation flying.
  - Small body missions (asteroids, comets).

- **<u>This is a hard problem</u>:**
  - Challenging dynamics
  - Multi-Target Tracking: many things to track
  - Measurement Limitations/Issues:
    - Lighting conditions
    - Features enter/exit FOV
    - Extraneous measurements (clutter)
    - Missed detections
  - Lack of *a priori* information
    - No *a priori* map or *a priori* map has significant uncertainty

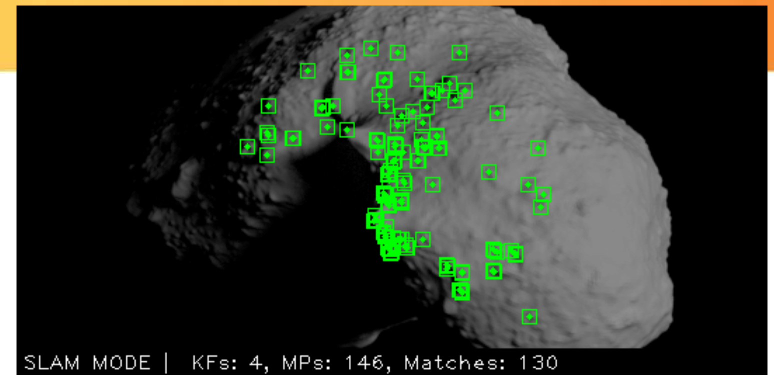


Credit: NASA, Restore-L

- **Problem:** Estimate location and rates of observer relative to client.



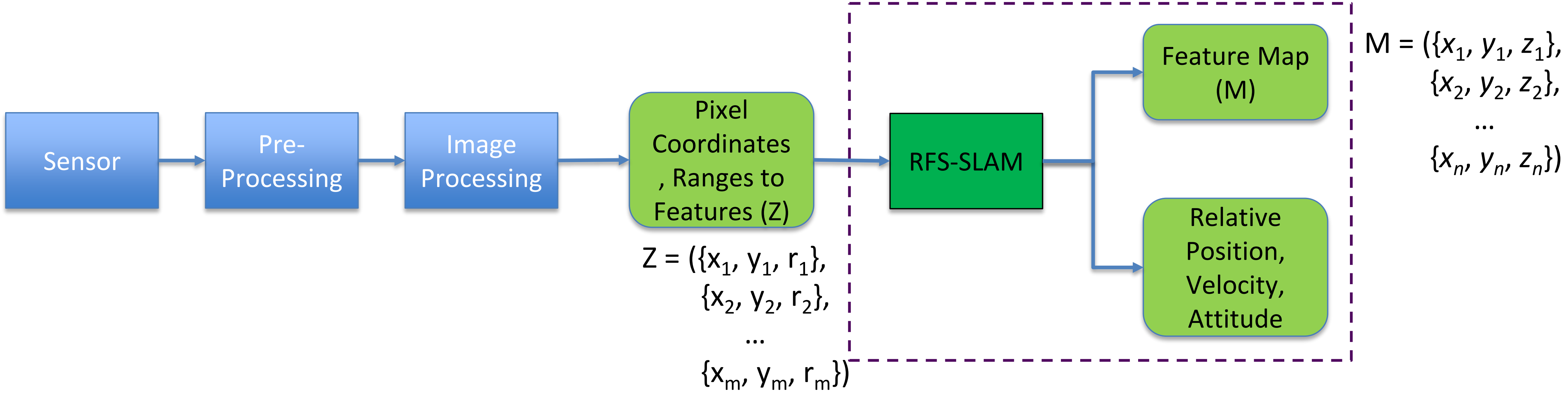
Measurements: Features on target extracted from optical image, LIDAR range to features



Features extracted from an asteroid image. **Credit: Kulumani, 2017**

- **Solution:** Formulate as a Simultaneous Localization and Mapping (SLAM) problem.
  - Localization = estimating observer's pose (position, velocity, attitude relative to client)
  - Mapping = estimating feature map (points or edges on a client – spacecraft or asteroid)
  - Simultaneously: pose depends on the map and vice versa
  - **More Problems:** Traditional SLAM methods often diverge due to issues with data association, high uncertainty.
    - *False sensor returns (clutter.)*
    - *Missed detection of expected features.*
    - *Map features entering/exiting Field of View.*

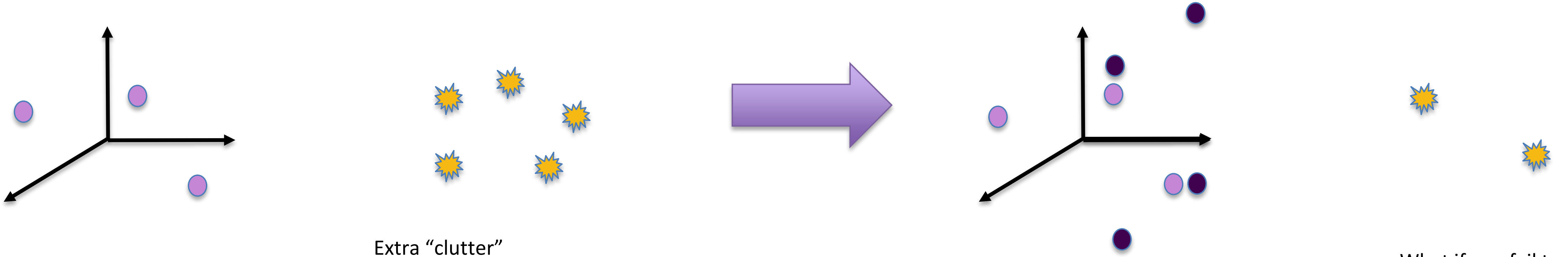
Block Diagram:



Unassociated measurements including clutter

Big Problem: Data association is hard.

States: t = k-1 | **Measurements: t = k-1** | **States: t = k** | **Measurements: t = k**



Extra "clutter" measurements, so how do you match measurements and targets?

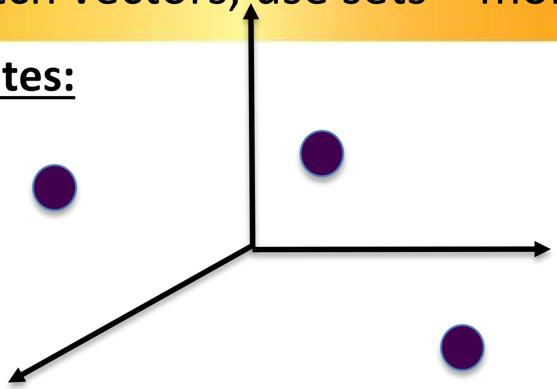
How do you know what moved where? What if something new entered? What if something from before left?

What if you fail to detect some things?

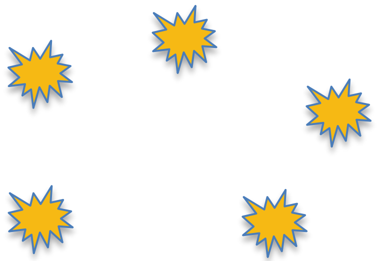
- The Kalman Filter was originally developed for single-target tracking. Must associate measurements with targets outside of the filter.
  - Heuristic methods have been used with the Kalman Filter to handle multi-target tracking.
    - Divergence if association is wrong.
    - Computationally expensive – especially as number of targets/measurements increases.
      - For 12 features and 20 measurements, data association matrix is 43GB, Matlab won't even initialize ☹

Reformulate: Ditch vectors, use sets – more natural, unified framework

True States:



Measurements:



Random Vectors:

$$X_1 = \begin{bmatrix}4\\5\\6\end{bmatrix} \quad X_2 = \begin{bmatrix}1\\3\\2\end{bmatrix} \quad X_3 = \begin{bmatrix}2\\1\\5\end{bmatrix}$$

$$Z_1 = \begin{bmatrix}4\\1\\3\end{bmatrix} \quad Z_2 = \begin{bmatrix}1\\8\\2\end{bmatrix} \quad Z_3 = \begin{bmatrix}7\\4\\6\end{bmatrix} \quad Z_4 = \begin{bmatrix}8\\1\\6\end{bmatrix} \quad Z_5 = \begin{bmatrix}9\\0\\4\end{bmatrix}$$

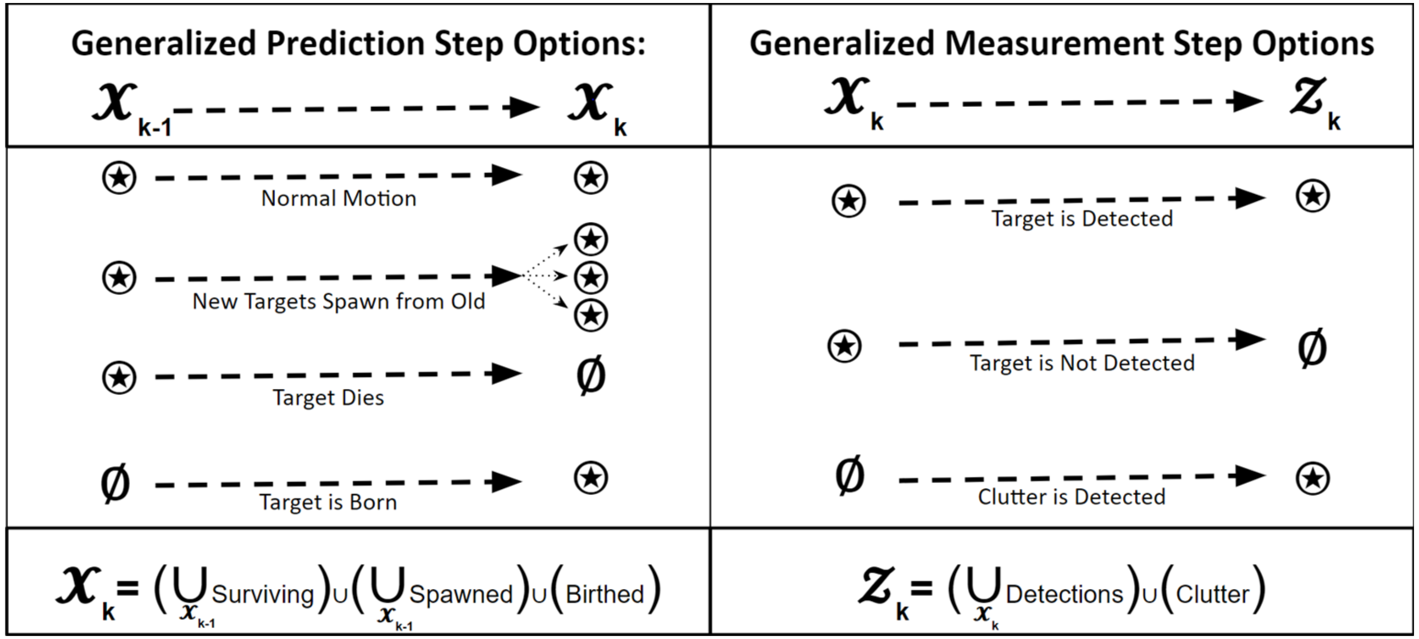
Random Finite Sets:

$$\mathcal{X} = \left\{\begin{bmatrix}4\\5\\6\end{bmatrix}, \begin{bmatrix}1\\3\\2\end{bmatrix}, \begin{bmatrix}2\\1\\5\end{bmatrix}\right\} = \left\{\begin{bmatrix}2\\1\\5\end{bmatrix}, \begin{bmatrix}4\\5\\6\end{bmatrix}, \begin{bmatrix}1\\3\\2\end{bmatrix}\right\} = \left\{\begin{bmatrix}1\\3\\2\end{bmatrix}, \begin{bmatrix}4\\5\\6\end{bmatrix}, \begin{bmatrix}2\\1\\5\end{bmatrix}\right\}$$

$$\mathcal{Z} = \left\{\begin{bmatrix}4\\1\\3\end{bmatrix}, \begin{bmatrix}1\\8\\2\end{bmatrix}, \begin{bmatrix}7\\4\\6\end{bmatrix}, \begin{bmatrix}8\\1\\6\end{bmatrix}, \begin{bmatrix}9\\0\\4\end{bmatrix}\right\}$$

Doesn't matter which order you put elements in, it's still the same set.
This is very powerful, if you have math to manipulate sets (we do.)

- **Random Finite Sets (RFS) are a more natural way of formulating the general SLAM problem.**
  - Vector-based formulation: existing target expected to continue to exist, and expected to be measured.
  - RFS formulation: other general propagation and measurement situations can be handled directly.



These other "options" usually prevent traditional SLAM approaches from converging.

- **No prior knowledge of the environment required!**
  - Ideal for non-cooperative rendezvous scenarios, no map of the target may be available.

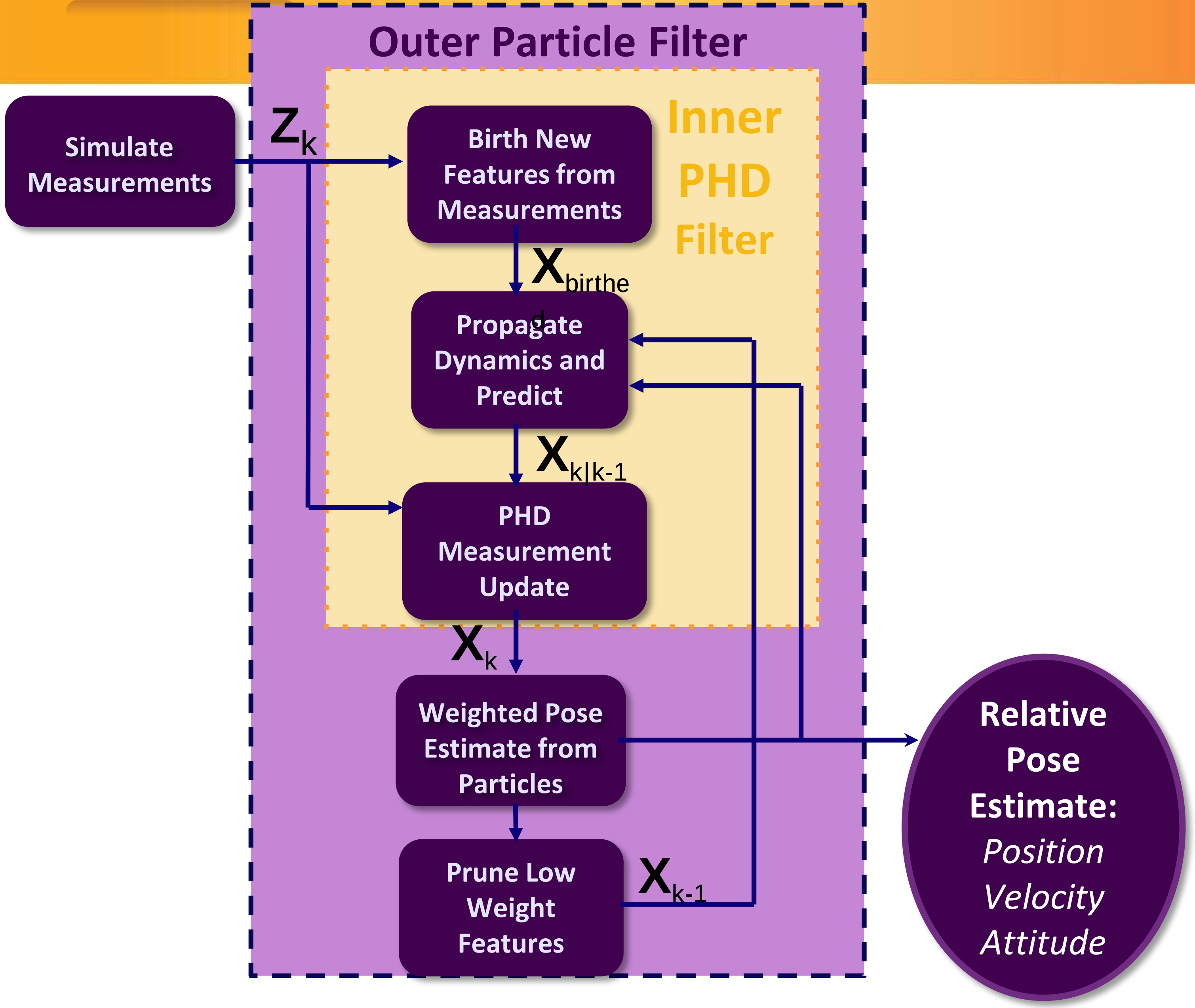
- **No data association required between measurements and targets!**
  - Dramatically reduces computational complexity, no chance of diverging due to incorrect association.

- **<u>Probability Hypothesis Density Filter (PHD Filter):</u>**
  - ***Proposed by Mahler in the 1990's, has been used for ground and naval robotics since the early 2000's***
  - ***Very similar to Kalman Filter:***
    - Optimal Bayes Filter
    - Linear Gaussian dynamics and measurement models
    - Equations are set theoretic analogs to Kalman filter prediction and update steps
  - ***The differences make it more general and flexible for multi-target tracking in realistic environments:***
    - Can model probabilities of detection/survival, clutter, etc. directly in the mathematics
      - Add a set of "clutter" to your measurement set
      - Add a set of "birthed targets" to your state set

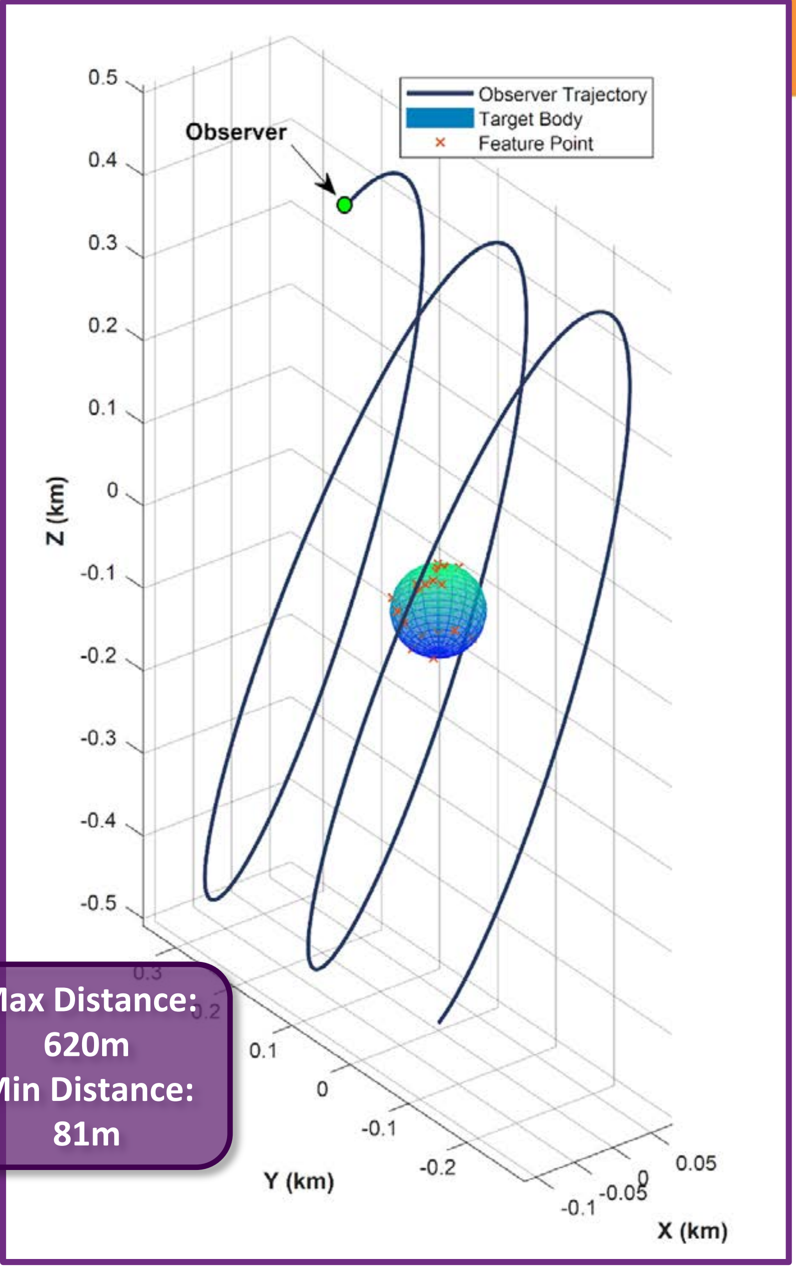
RFS-SLAM Filter Structure:

PHD filter wrapped inside a particle filter

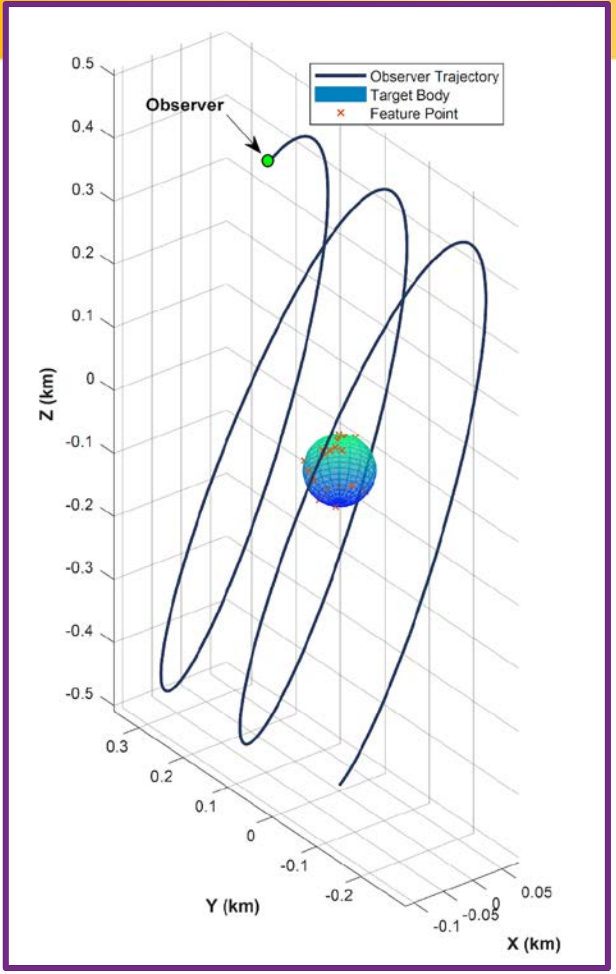
- **Pose is represented by particles**
- ***Mapping:*** **PHD filter determines which cluttered, noisy measurements correspond to actual map features.**
  - Each particle's feature map is conditioned on a different pose.
- ***Localization:*** **Quality of each feature map determines the "weight" of each particle (W_k).**
  - Weighted average of pose can be calculated.
  - Alternatively, maximum *a posteriori* estimate.



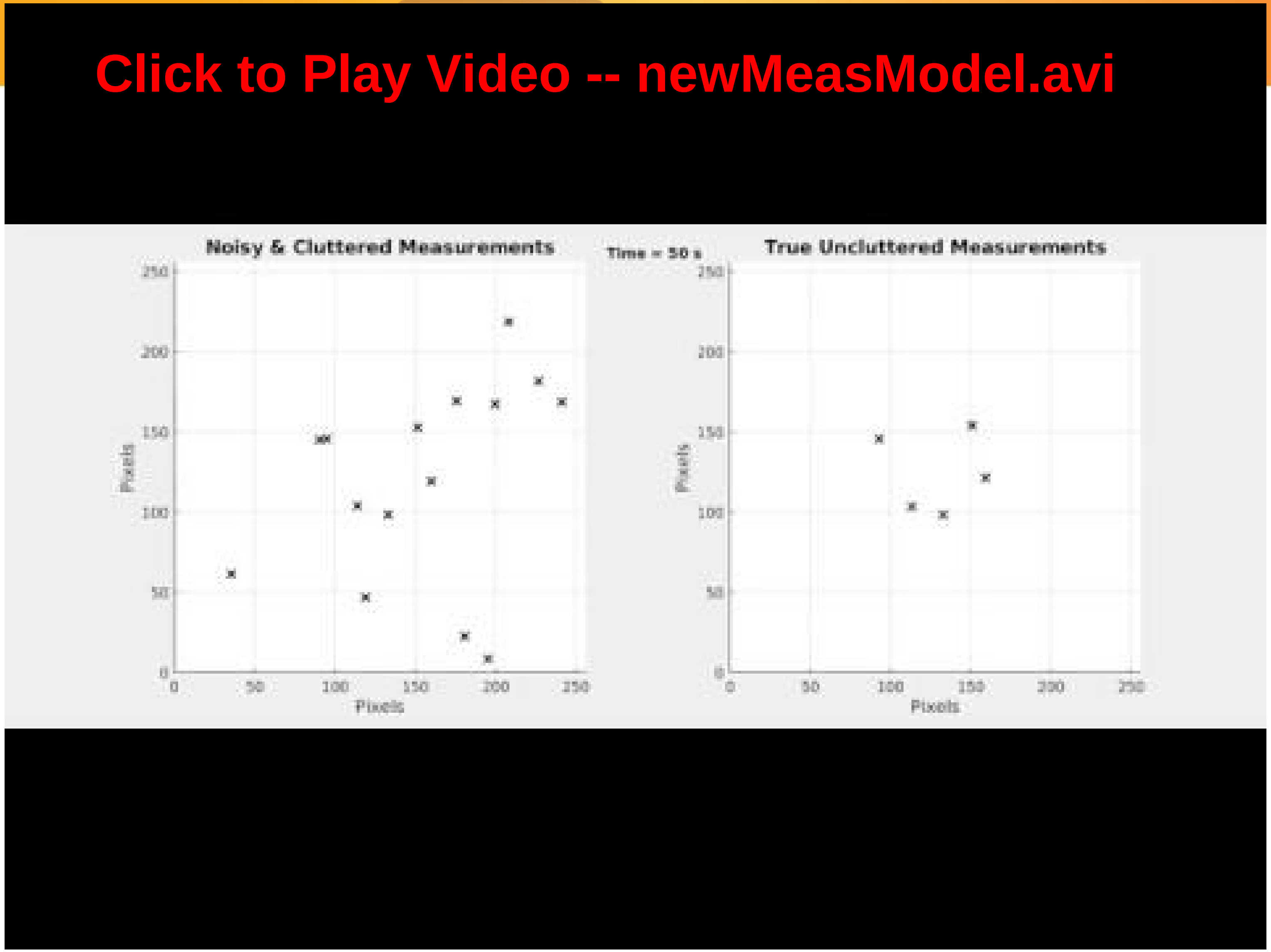
- <u>Target:</u> Sphere, radius 50m, with randomly distributed features
  - *Chosen to simplify simulation of feature occlusion*
    - Simple dot products to determine which simulated measurements are visible
- <u>Observer:</u> Attitude assumed to always point at target
  - Attitude control keeps target centered in sensor field of view (assumed perfect)
- <u>Orbital Dynamics:</u> Clohessy-Wiltshire equations for relative motion
  - Target attitude is constant in the CW frame
- <u>Attitude:</u> Modified Rodrigues Parameters
  - 1 easily avoidable singularity
- <u>Measurements:</u> flash LIDAR (simulated)
  - 256x256 pixel image, 14° Angle of View (similar to Raven)
  - Features = pixel coordinates and range relative to boresight
- <u>Estimation:</u> in target body-fixed frame
  - Map = body-fixed feature locations (static)
  - Pose = observer position, velocity, attitude relative to target



Max Distance: 620m
Min Distance: 81m



Measurement Noise: [1 pixel, 1 pixel, 10 mm]



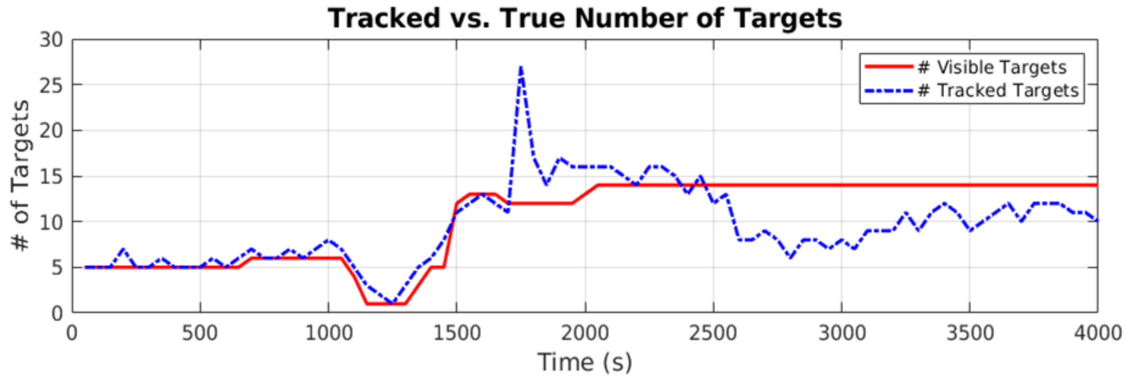
Note: This video shows the image plane; range measurements are also used, but are difficult to visualize.

NASA Goddard Space Flight Center
MISSION ENGINEERING AND SYSTEMS ANALYSIS DIVISION

Case 1: Nominal Orbit

Click to Play Video -- Case1.avi

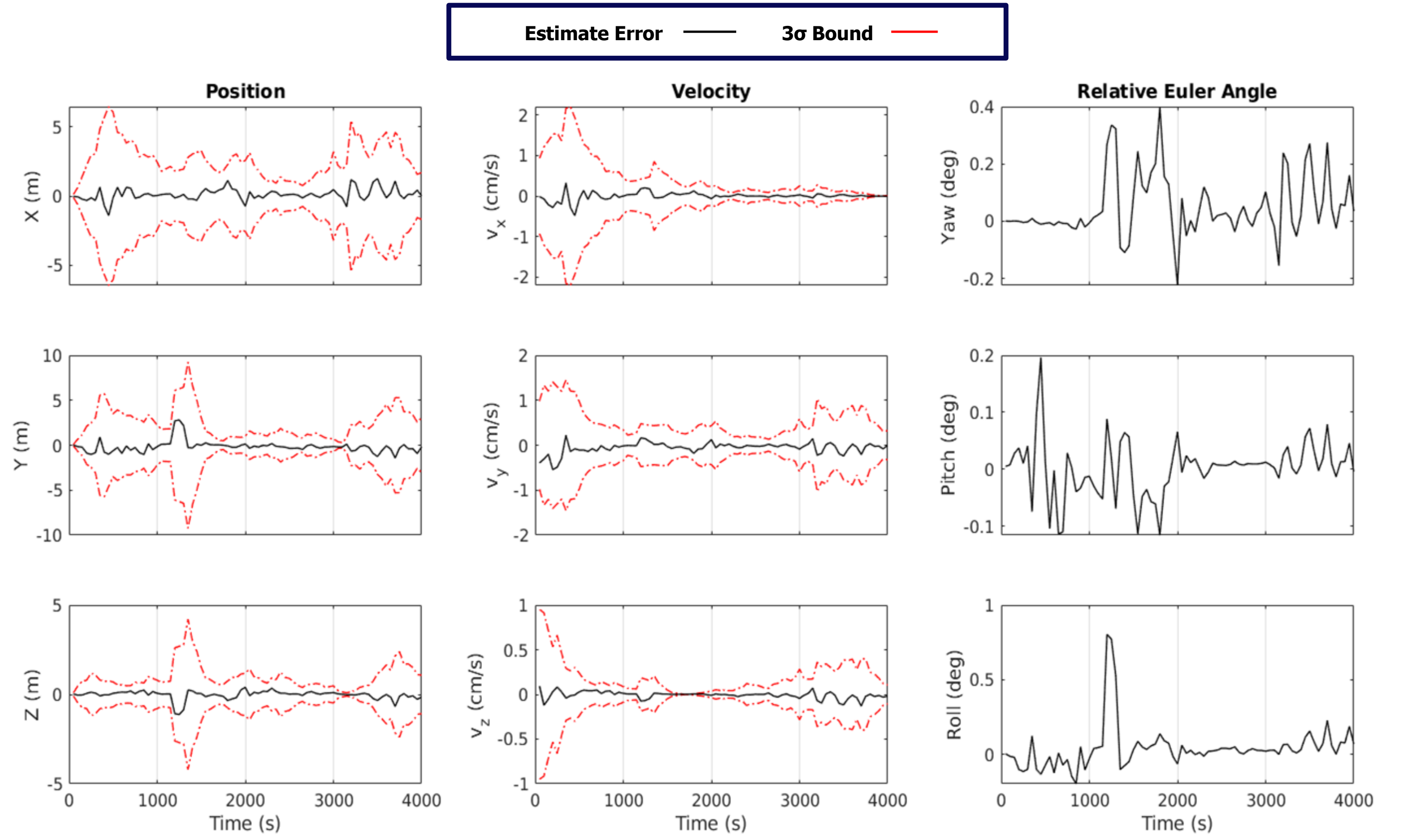
Number of Particles Used: 100

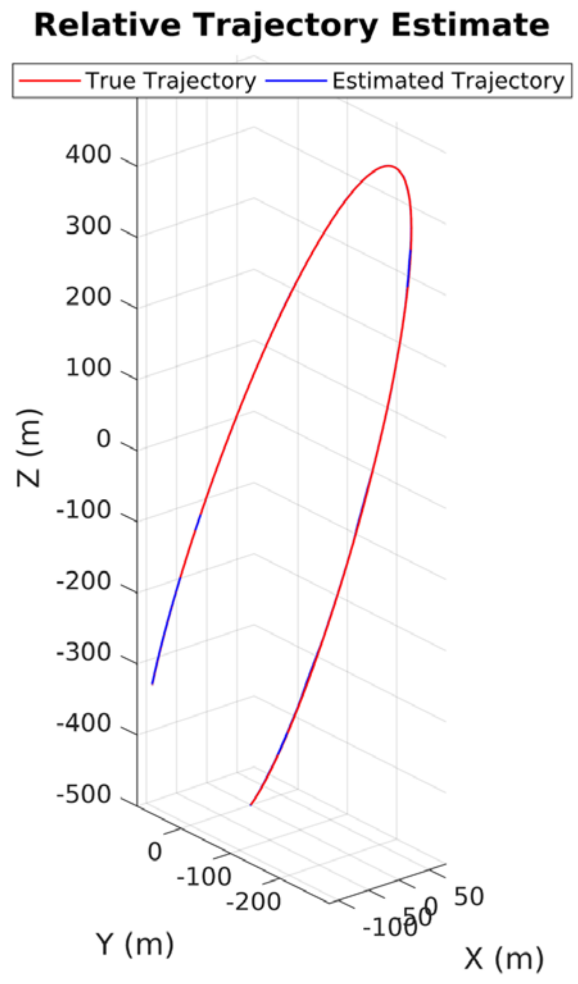


If the filter is well-tuned, the cardinality of the PHD will closely match the number of visible features.

Case 1: Nominal Orbit

Pose Estimate Error vs. Sample Covariance 3σ Bounds

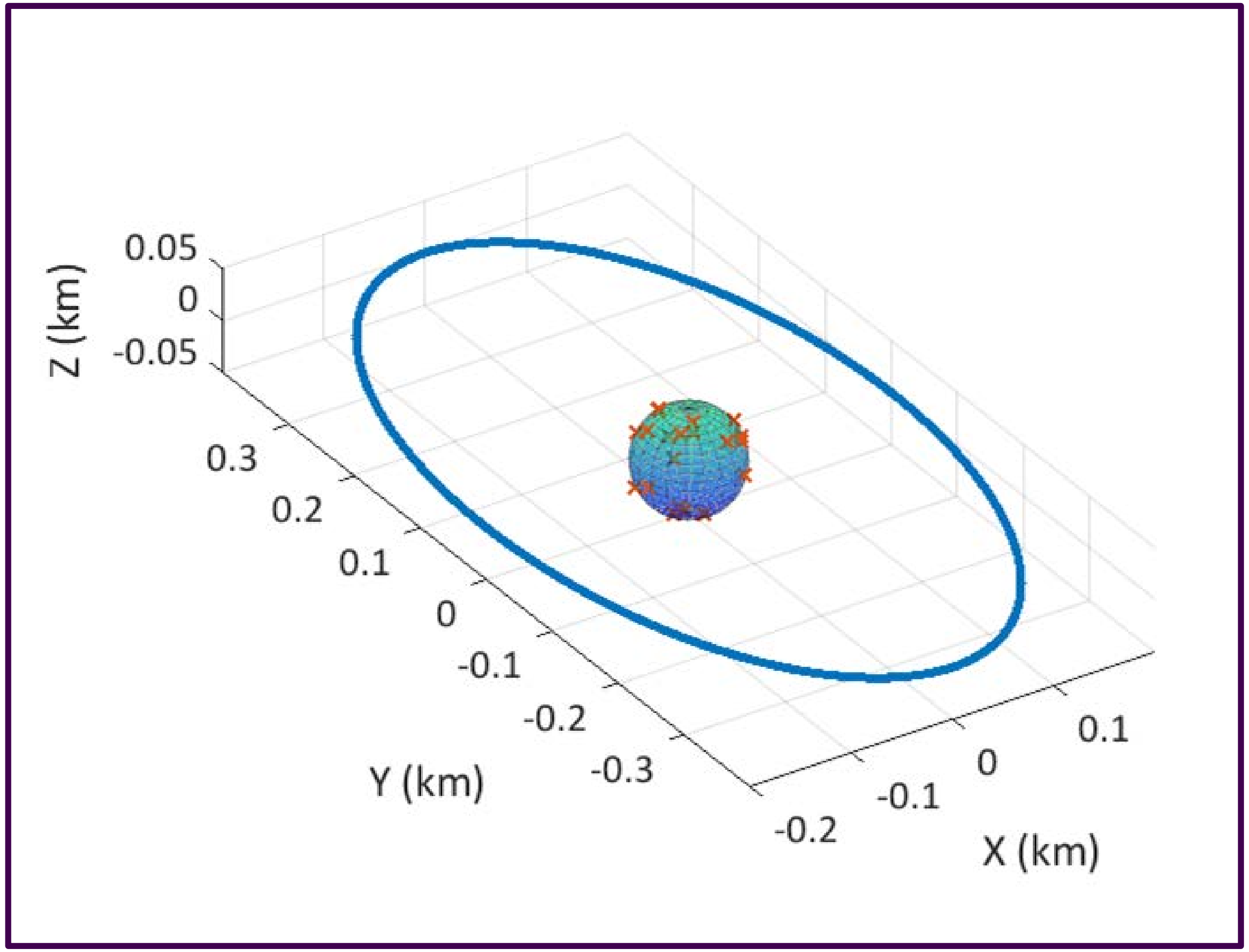




Pose estimate stays well within sample covariance 3σ bounds, and estimated trajectory matches the truth very closely.

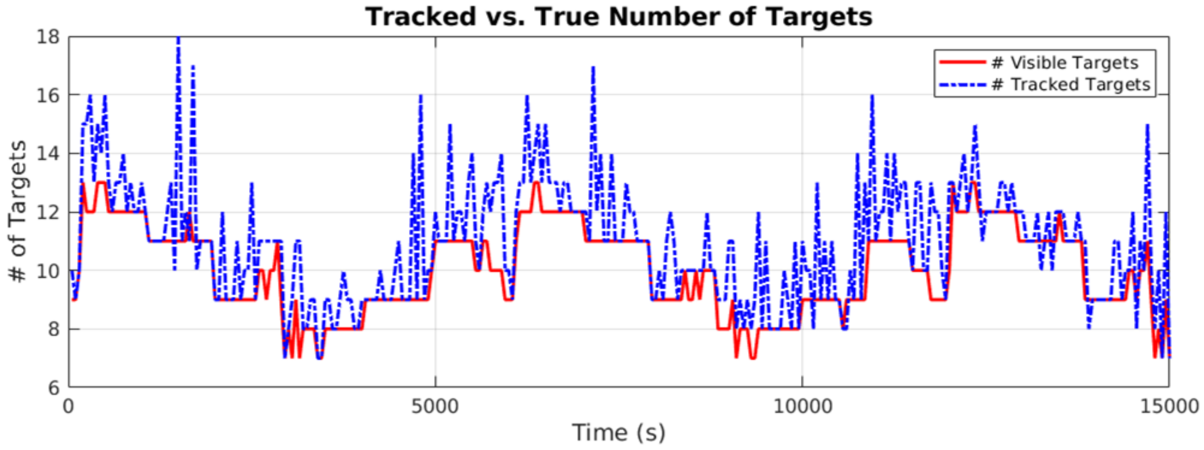
Case 2: Periodic Planar Orbit

Goal: Test filter stability over a longer period of time.



Case 2: Periodic Planar Orbit

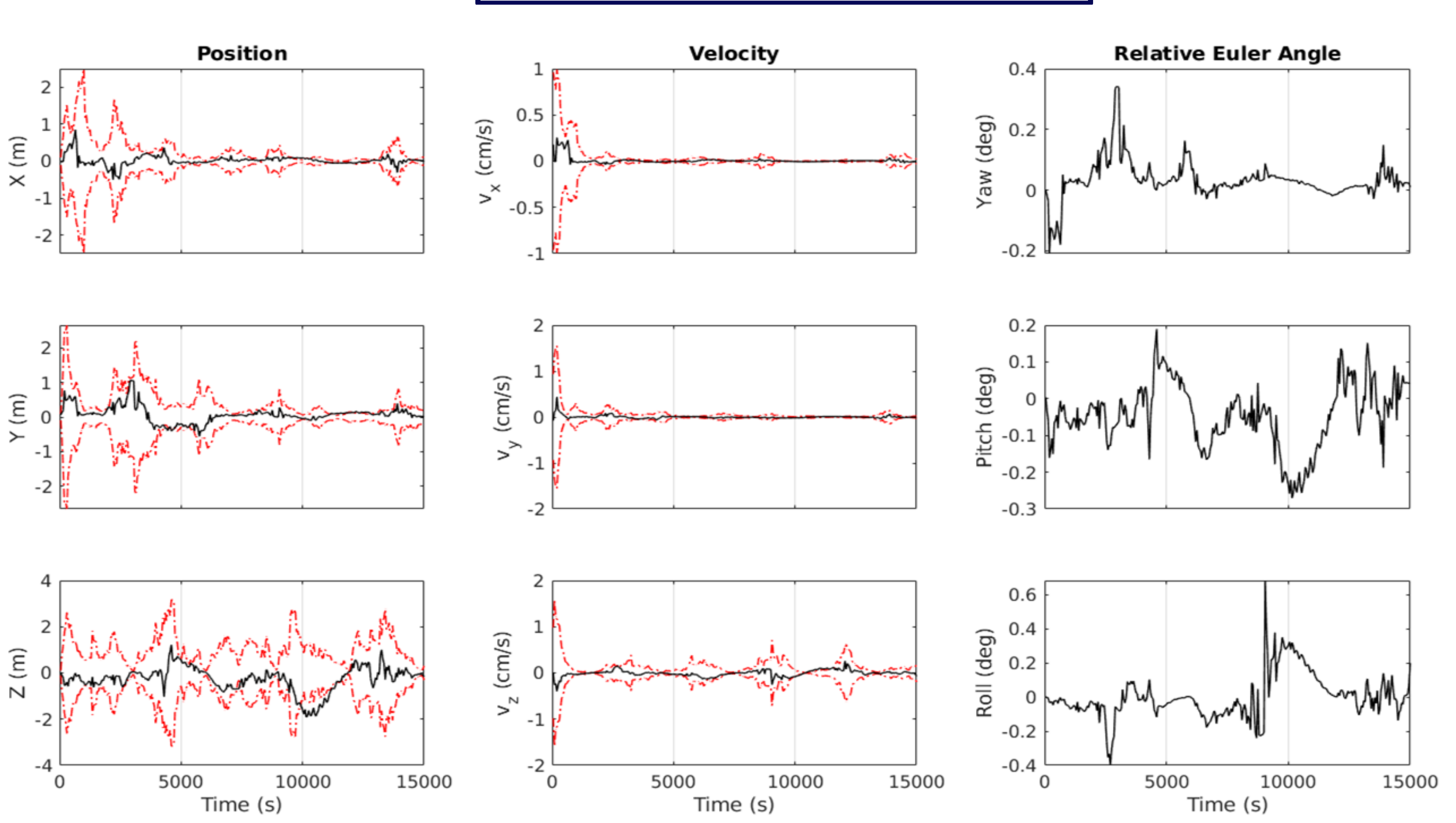
Number of Particles Used: 100

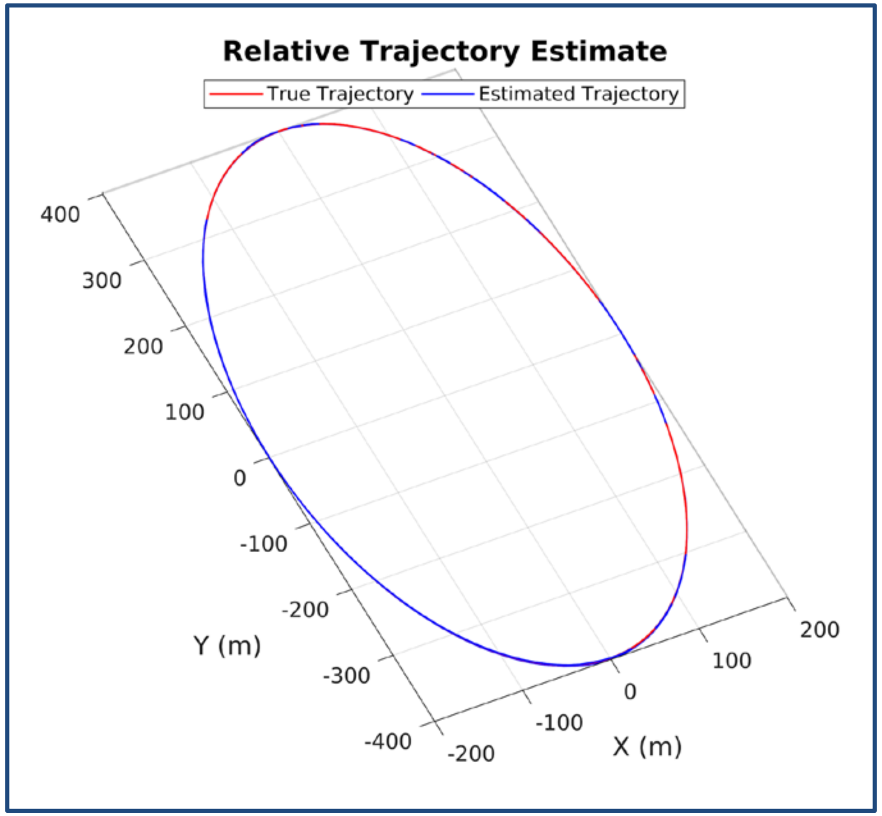


Difference between # of tracked and true targets is periodic with the orbit; filter has been tuned to slightly overestimate # of targets for stability.

Case 2: Periodic Planar Orbit

Pose Estimate Error vs. Sample Covariance 3σ Bounds





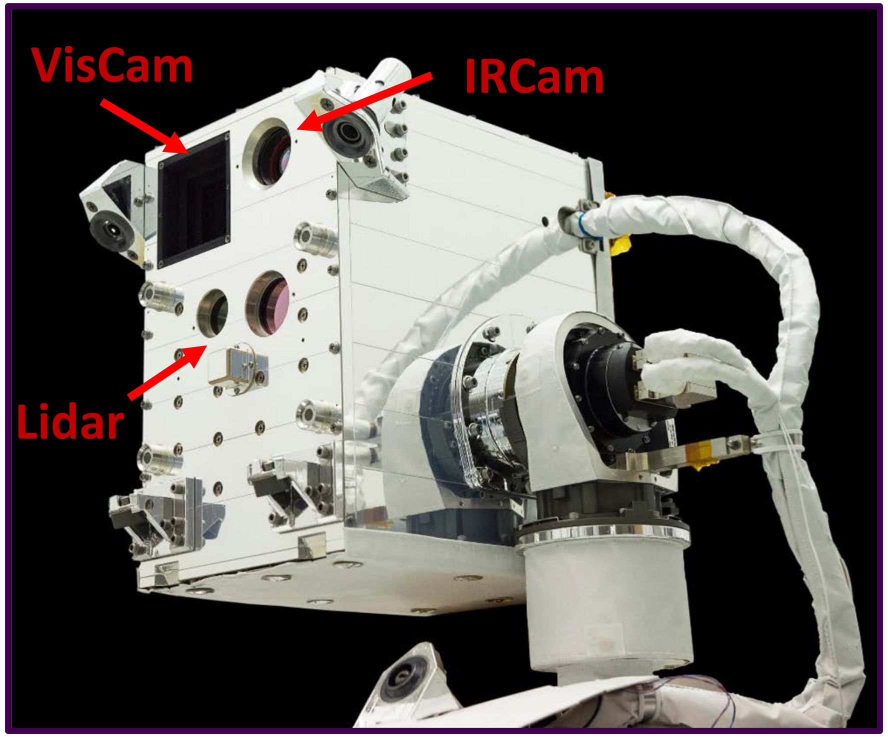
Pose estimate stay within sample covariance 3σ bounds, and estimated trajectory matches the truth very closely over a long period of time.

- We have shown using a particle filter that **a Probability Hypothesis Density RFS filter can feasibly perform SLAM in a realistic noisy, non-cooperative space environment.**
  - ✓ *First known demonstration of RFS SLAM methods for spacecraft pose estimation and mapping.*
  - ✓ Mapping is successful (features are tracked.); Localization is successful (pose is estimated.)
  - ✓ Very little/no *a priori* knowledge of the map given to the filter.
  - ✓ No data association required between states and measurements.
    - ✓ Very little post-processing for feature management.
  - ✓ Heavily cluttered, highly noisy measurements used.
  - ✓ Features move in and out of the sensor field of view.
- **Pose estimate quality is dependent on the number of particles used in the outer particle filter.**
  - More particles is better, but computationally costly.

Moving Forward:

- **Investigate computationally cheaper ways of extracting a pose estimate**
  - e.g. EKF localization from the PHD map, Multi-Bernoulli RFS filter variants.
    - More feasible for onboard computers.
- **More Complicated Dynamics**
  - Higher fidelity dynamics, no constraint on pointing.

- ***Real Life Data***
  - Use optical images and LIDAR measurements from existing missions to perform SLAM. (e.g. Raven data from ISS rendezvous)
    - These missions use *a priori* maps, **demonstrate that RFS performance with and without.**



Credit: NASA, Raven

Summary:

- **Autonomous spacecraft navigation:** a necessary requirement of future exploration/servicing missions.
- **Current methods:** ill-suited for challenging measurement environment during rendezvous/prox ops.
- **Random Finite Set based filters:** a more natural choice for these kinds of problems.
- **Initial simulations:** *first known demonstration* of RFS SLAM methods for spacecraft pose estimation and mapping.
- **Moving forward:**
  - Investigate different variants of filter formulations for robustness, real-time computing.
  - Assess performance using flight data.

Acknowledgements:

- **Funding from NASA GSFC IRAD Program**
- **Dr. Richard Linares, Massachusetts Institute of Technology**
- **Dr. Martin Adams, Universidad de Chile**
  - **Felipe Inostroza**
- **Eugene Skelton and Anthony Yu, NASA GSFC**

Thanks for listening!

Questions?