The Multi-State Constraint Kalman Filter

Or, how we learned to stop worrying and love the null space

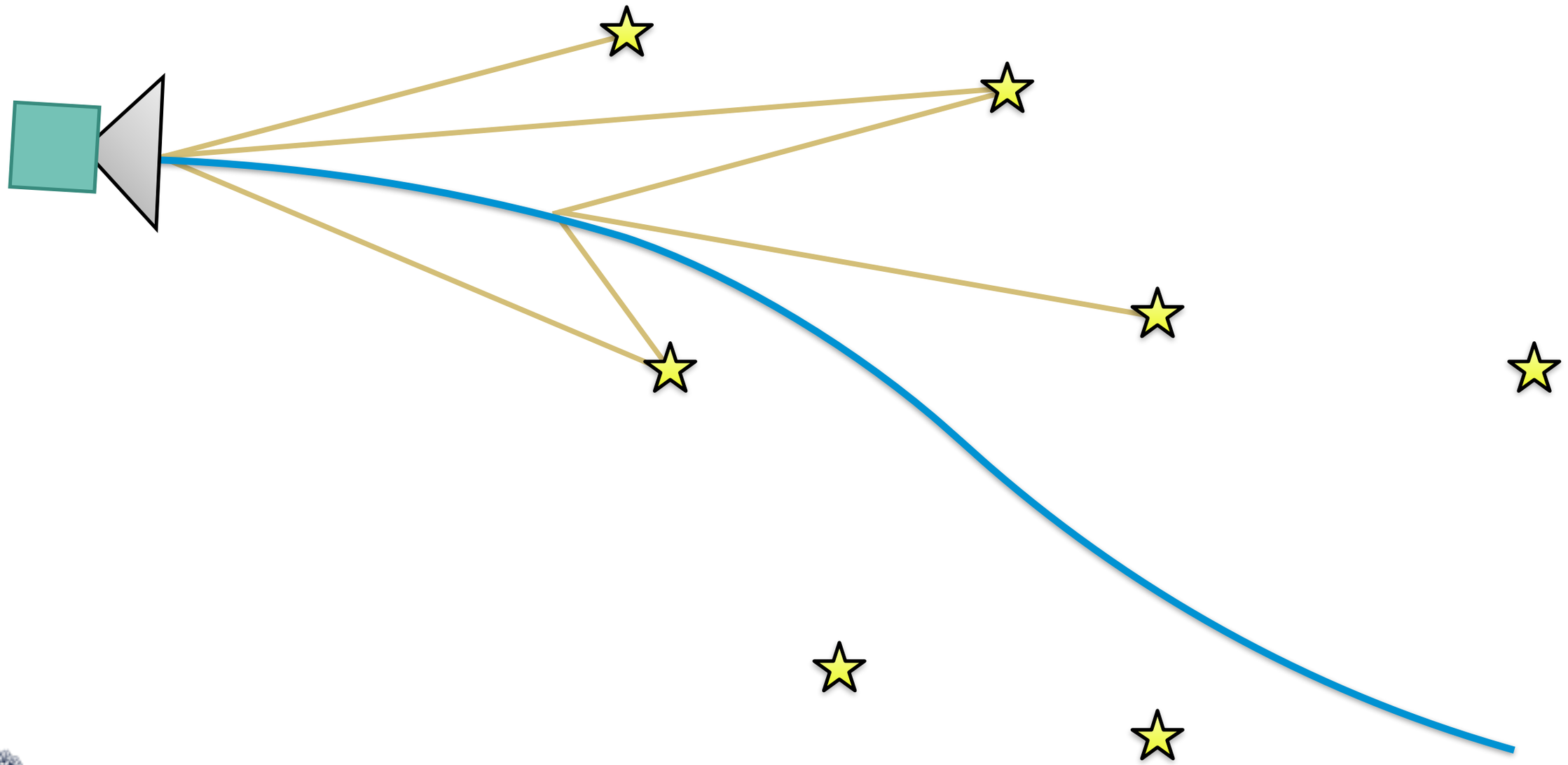
Valentin Peretroukhin and Lee Clement

AER1513 Course Project

Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Problem: Vision-aided Inertial Navigation

Goal: Use an IMU with a monocular camera to estimate motion *without a map*.



Algorithm: MSCKF

Idea: Traditional pairwise landmark triangulation ignores correlations with observations at other states, so use a hybrid batch/recursive filter

Batch component: Track each feature until it goes out of view, then compute its position from *all available measurements*.

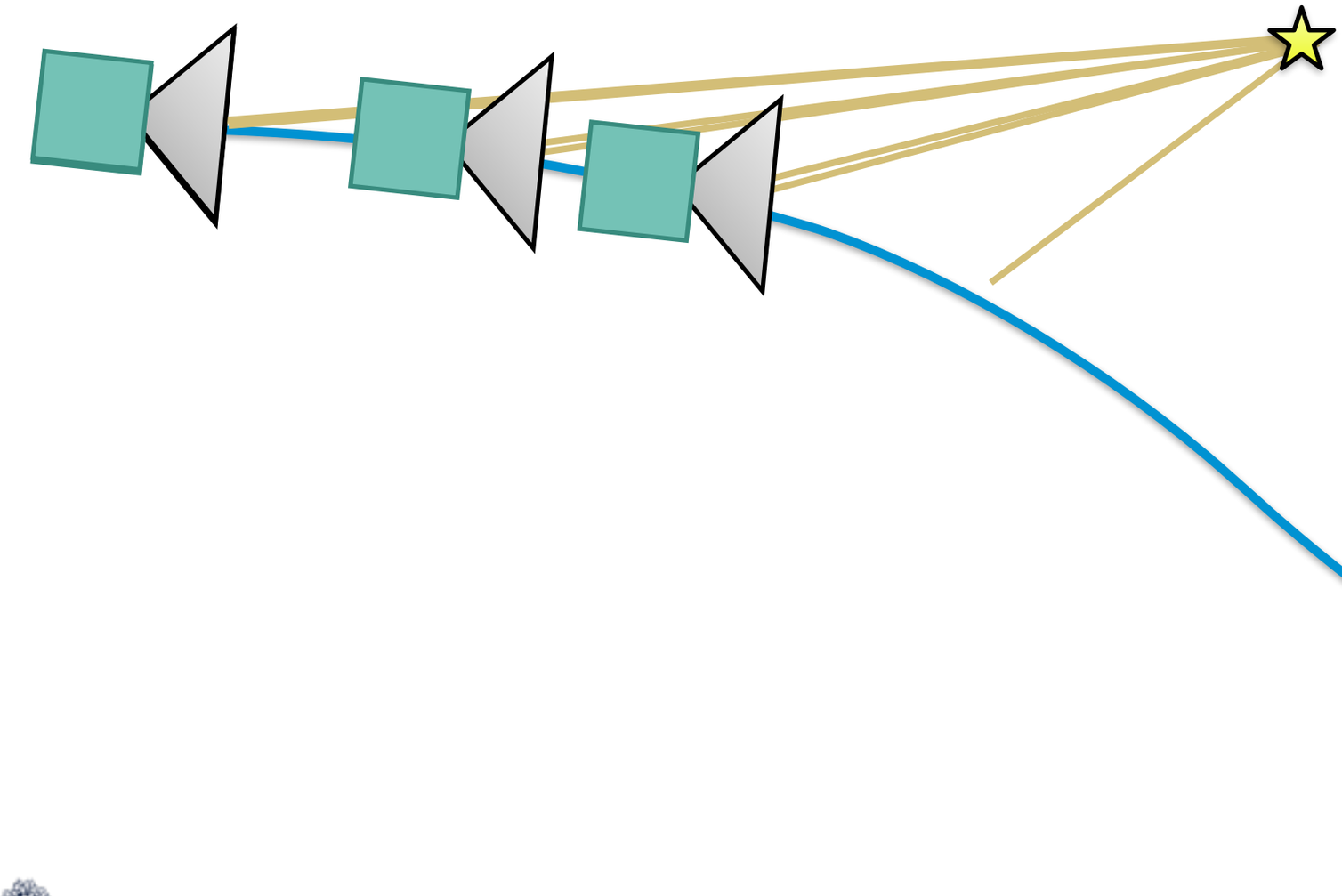
Recursive component: Use landmark position and all of its measurements (with null space trick) *to constrain motion*.

Advantages over plain EKF:

- **Sliding window of poses** allows each constraint to affect multiples states
- **Computational complexity is linear** in number of landmarks instead of cubic for plain EKF SLAM.

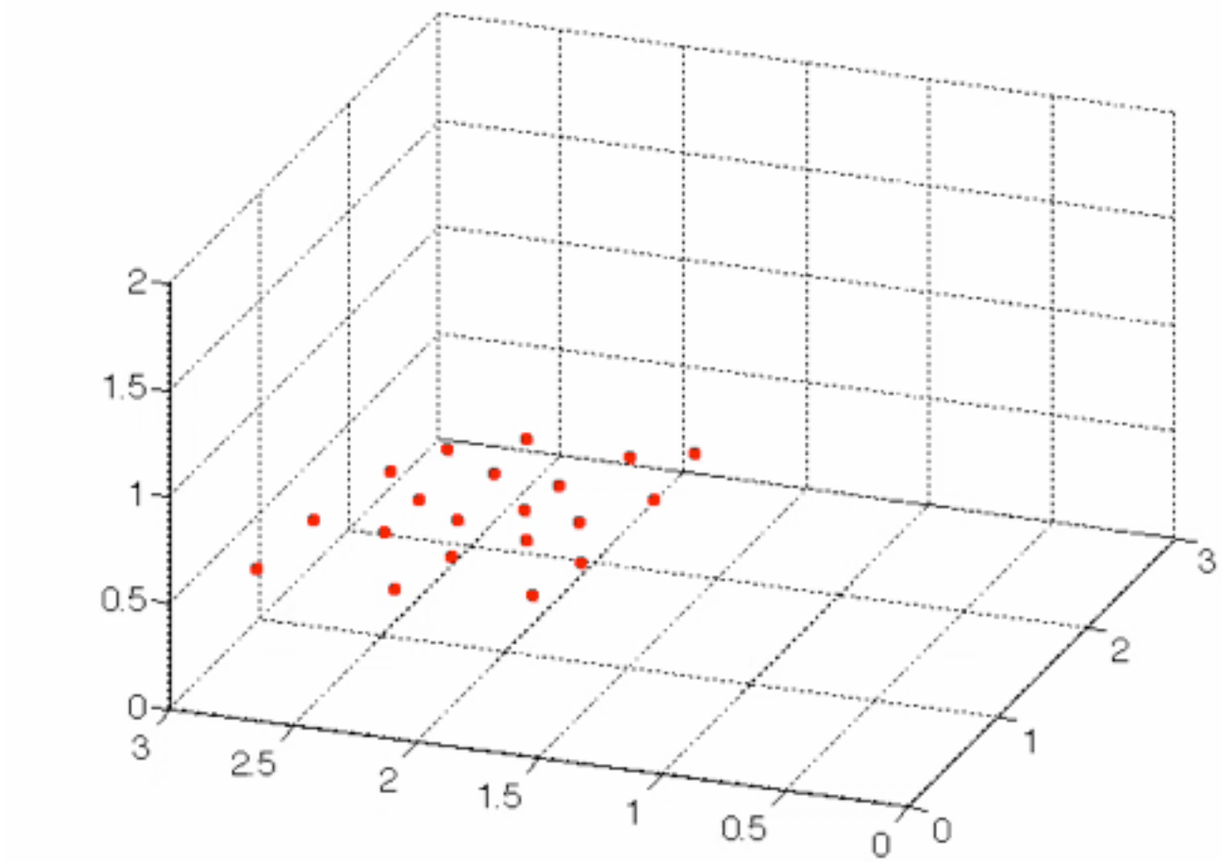
Algorithm: MSCKF

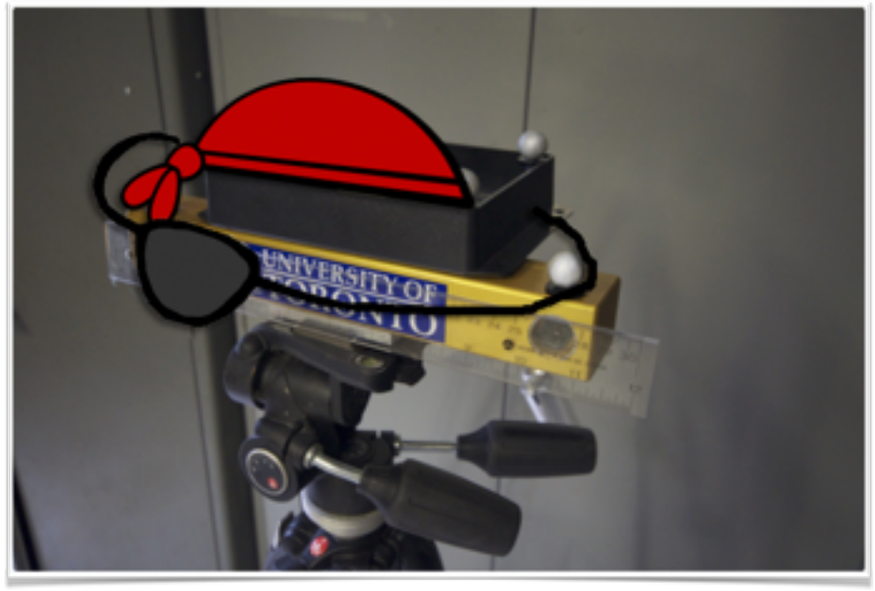
Each set of observations for a given feature (i.e. a ‘feature track’) adds a motion constraint.



1. Estimate feature position
2. Combine observations into exteroceptive update
3. Repeat for all feature tracks

Dataset: Starry Night (Assignment 3)





- ☑ Perfect data association
- ☑ Ground truth for landmark positions
- ☑ Pre-integrated IMU measurements

Planned Analysis: MSCKF

We will **investigate** MSCKF parameters:

1. Feature track length
2. Maximum window size

We will **compare**:

MSCKF *vs.* Sliding Window Batch Estimation

State Estimation Street Fight

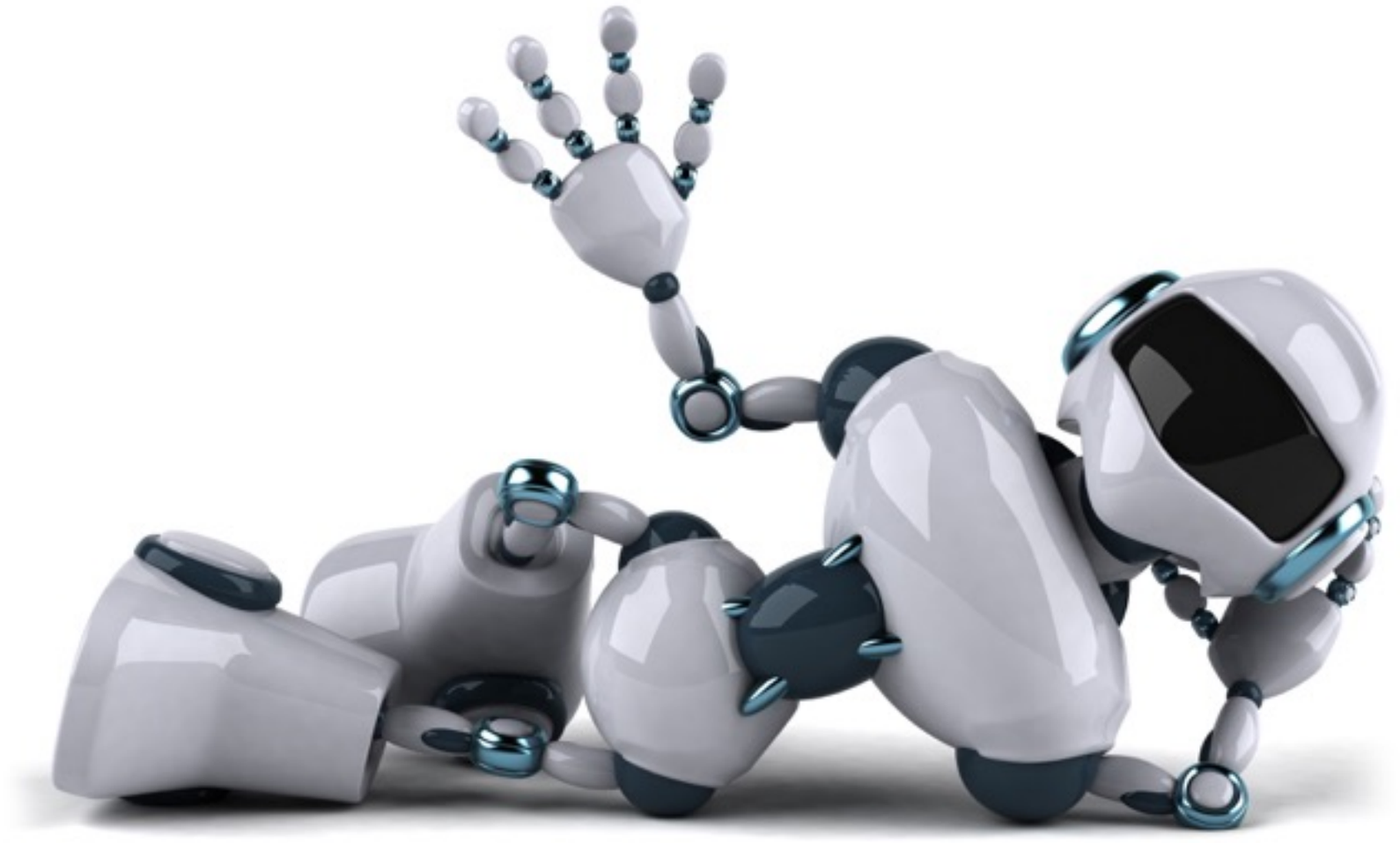
Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Proof of Progress

MSCKF.m
propagateImuState.m
propagateMsckfCova...
augmentState.m
updateState.m
calcF.m
calcG.m
calcJ.m
calcHoj.m
calcTH.m
calcGNPosEst.m
calcResidual.m
getGoodGrade.m
+

```
74
75      %==========================STATE PROPAGATION=======================%
76
77      %Propagate state and covariance
78 -    msckfState = propagateMsckfCovar(msckfState, measurements{state_k}, noiseParams);
79
80      %Add camera pose to msckfState
81 -    msckfState = augmentState(msckfState, camera);
82
83
84      %==========================FEATURE TRACKING========================%
85      % Add observations to the feature tracks, or initialize a new one
86      % If an observation is -1, add the track to featureTracksToResidualize
87 -    featureTracksToResidualize = {};
88 -    for featureId = 1:20
89 -        meas_k = measurements{state_k}.y(:, featureId);
90 -        if ismember(featureId, trackedFeatureIds)
91 -            if meas_k(1,1) == -1
92                  %Add to residualize queue and remove from the original
93                  %struct
94 -                featureTracksToResidualize{end+1} = featureTracks{trackedFeatureIds == featureId};
95 -                featureTracks = featureTracks(trackedFeatureIds ~= featureId);
96 -                trackedFeatureIds(trackedFeatureIds == featureId) = [];
97 -            else
98                  %Append observation and increase k2
99 -                featureTracks{trackedFeatureIds == featureId}.observations(:, end+1) = meas_k;
100 -                featureTracks{trackedFeatureIds == featureId}.k2 = featureTracks{trackedFeatureIds == featureId}.k2 + 1;
101 -            end
102 -        else
103              %Track new feature
104 -            track.featureId = featureId;
105 -            track.observations = meas_k;
106 -            track.k1 = state_k;
107 -            track.k2 = state_k;
108 -            featureTracks{end+1} = track;
109 -            trackedFeatureIds(end+1) = featureId;
110 -        end
111 -    end
112
113     %==========================FEATURE RESIDUAL CORRECTIONS=======================%
114
115 -    featuresToResidualize = []; %1xN matrix of feature ids (this is just the column of y_k_j)
```

Thanks!



Institute for Aerospace Studies
UNIVERSITY OF TORONTO

Image credit: galleryhip.com