



Nautilus Automates, You Analyze

Reproducible Analysis of Sensor Data in R

Dr. Benjamin Ashwell
Dr. Kevin Kirshenbaum

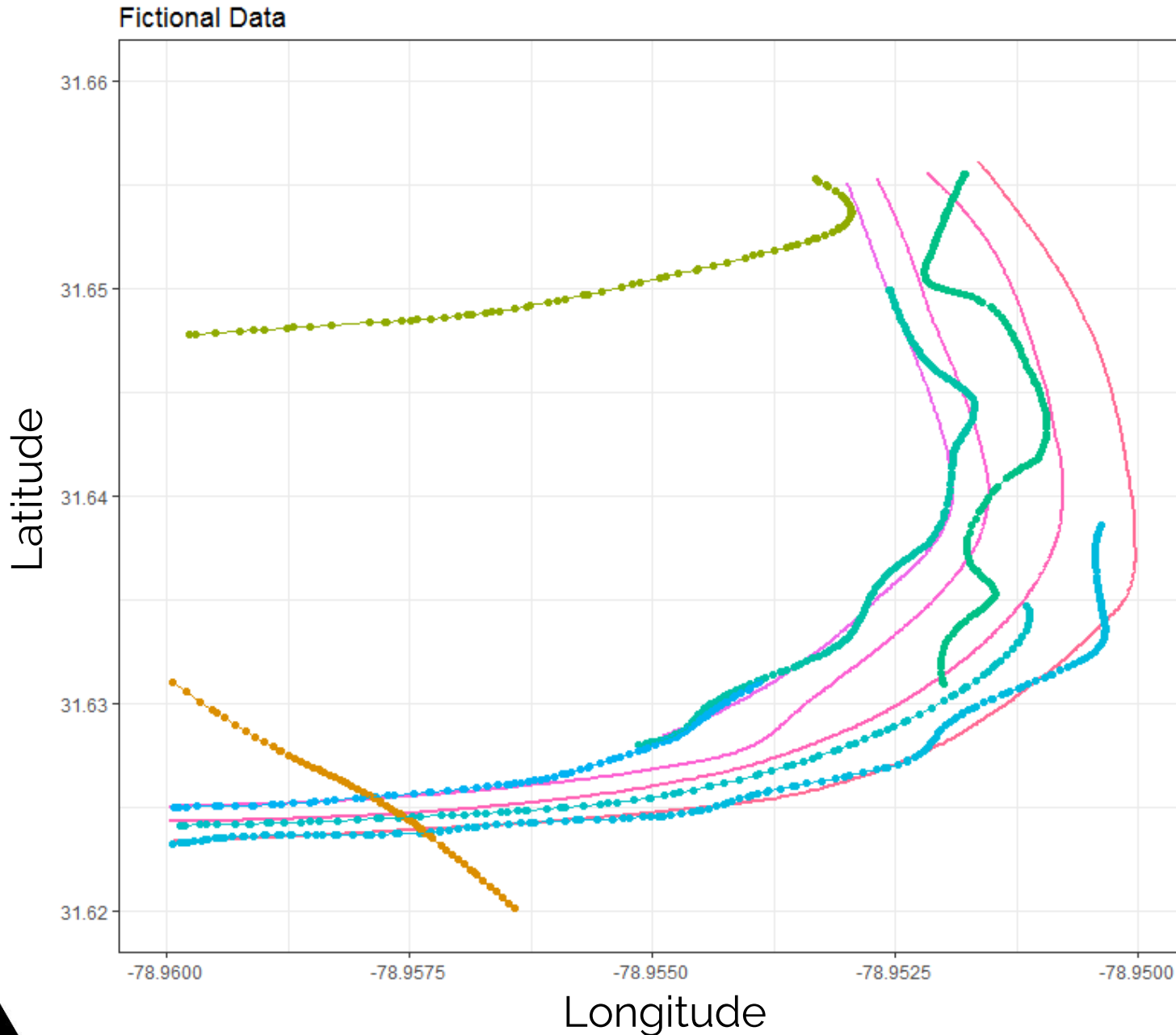
21 March, 2018

We need to know how well sensors that provide positional data work

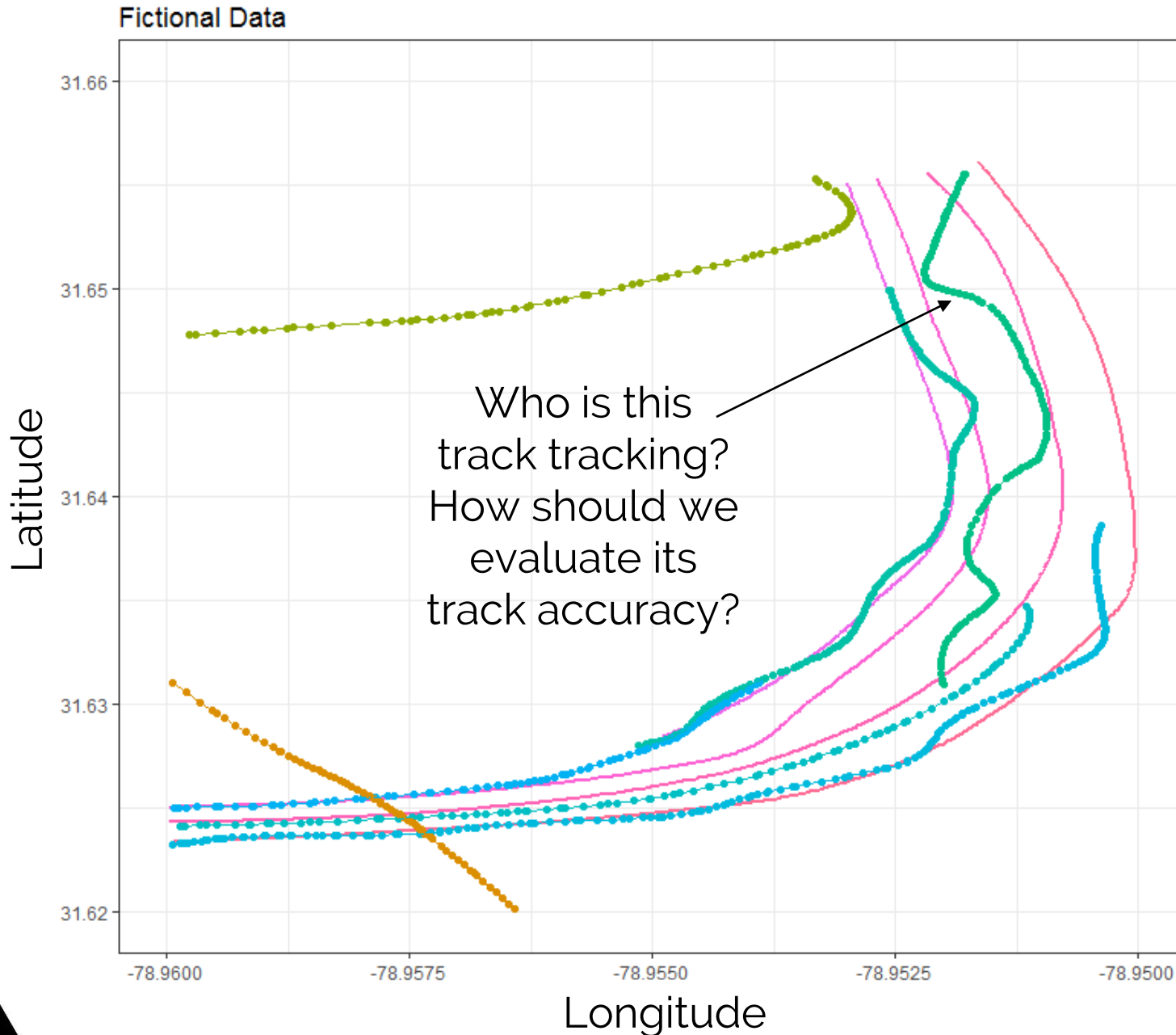
We need to know how well sensors that provide positional data work

This apparently simple goal is actually very complicated

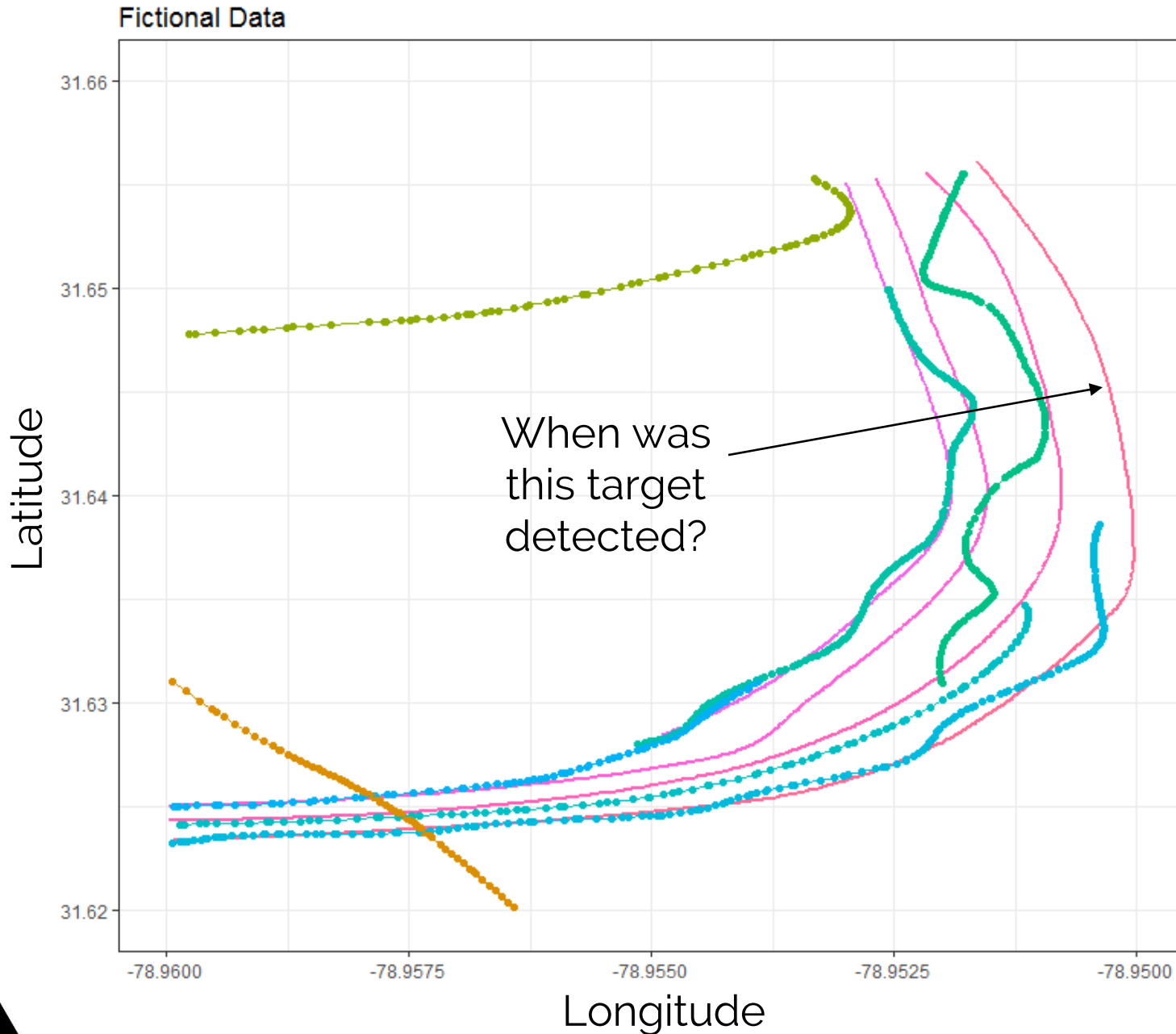
Track-to-truth assignment is often tedious and ambiguous



Track-to-truth assignment is often tedious and ambiguous



Track-to-truth assignment is often tedious and ambiguous



The things that matter for sensor performance change with the mission

Surveillance radars



Fire control radars



Nautilus takes tracks and truth...

Target A



Track 1

Target B



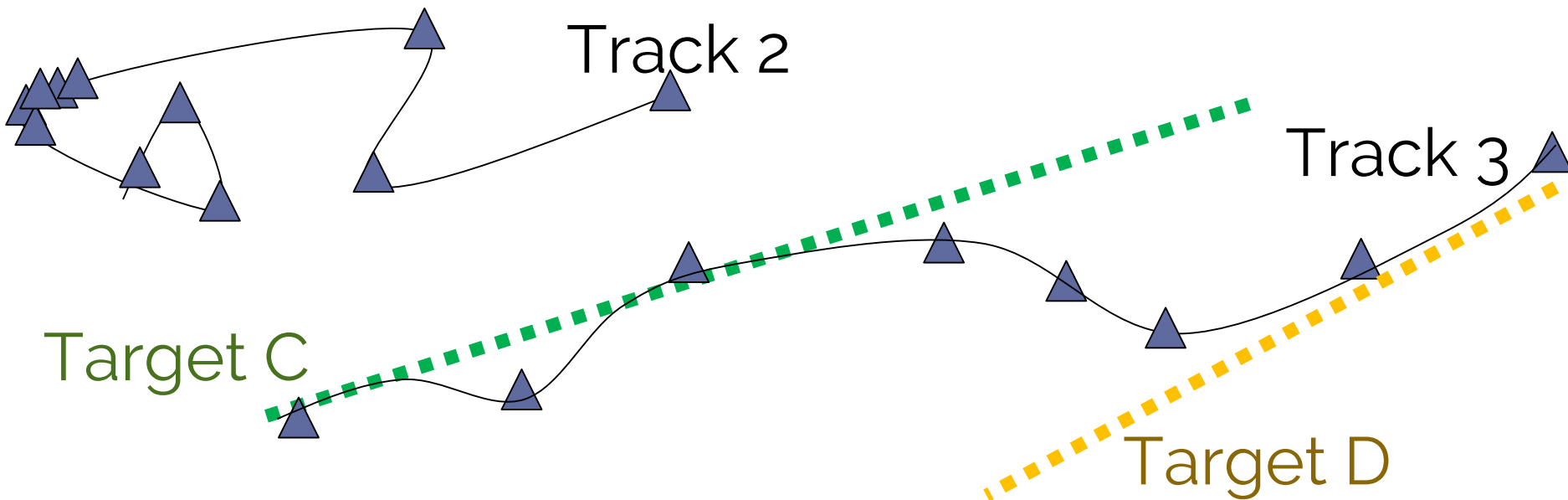
Track 2

Target C



Track 3

Target D

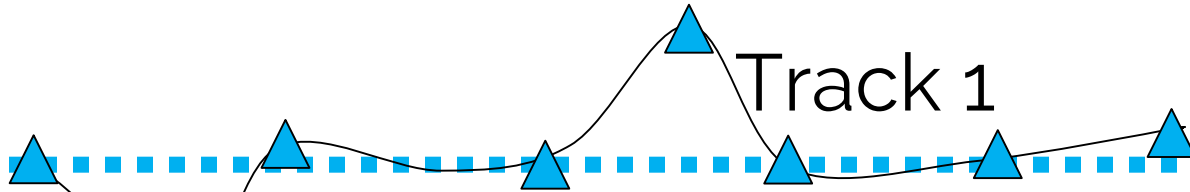


... and assigns tracks to the most likely targets

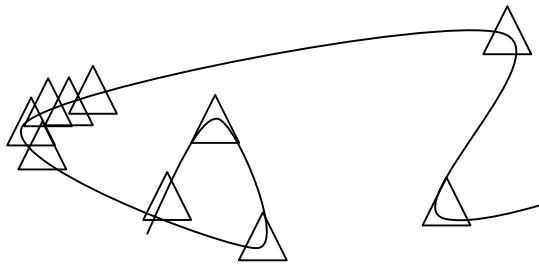
Target A



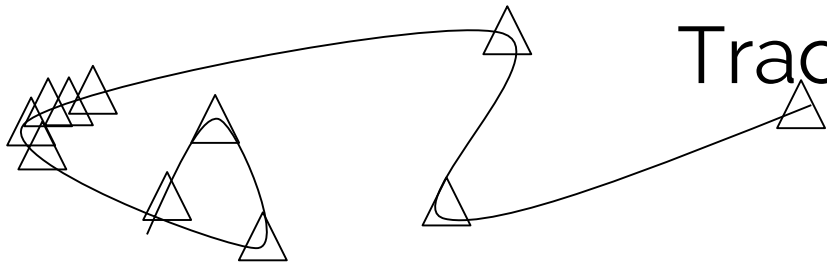
Track 1



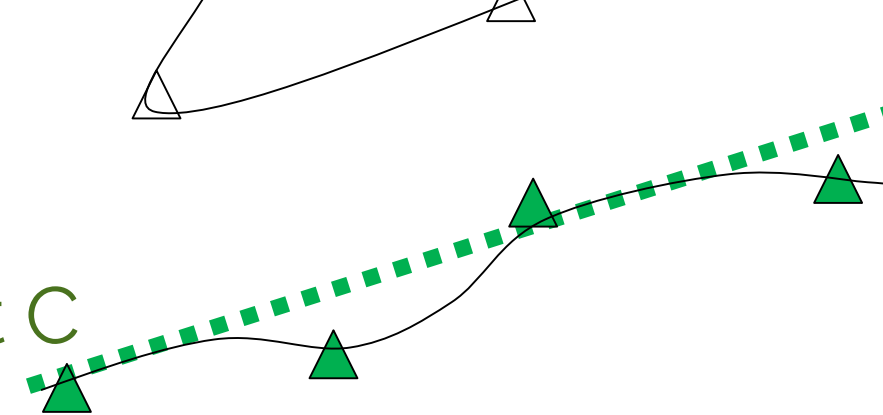
Target B



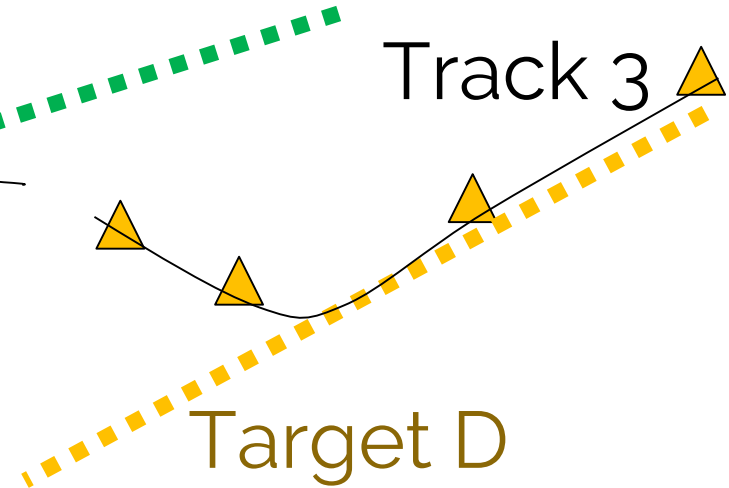
Track 2



Target C



Track 3

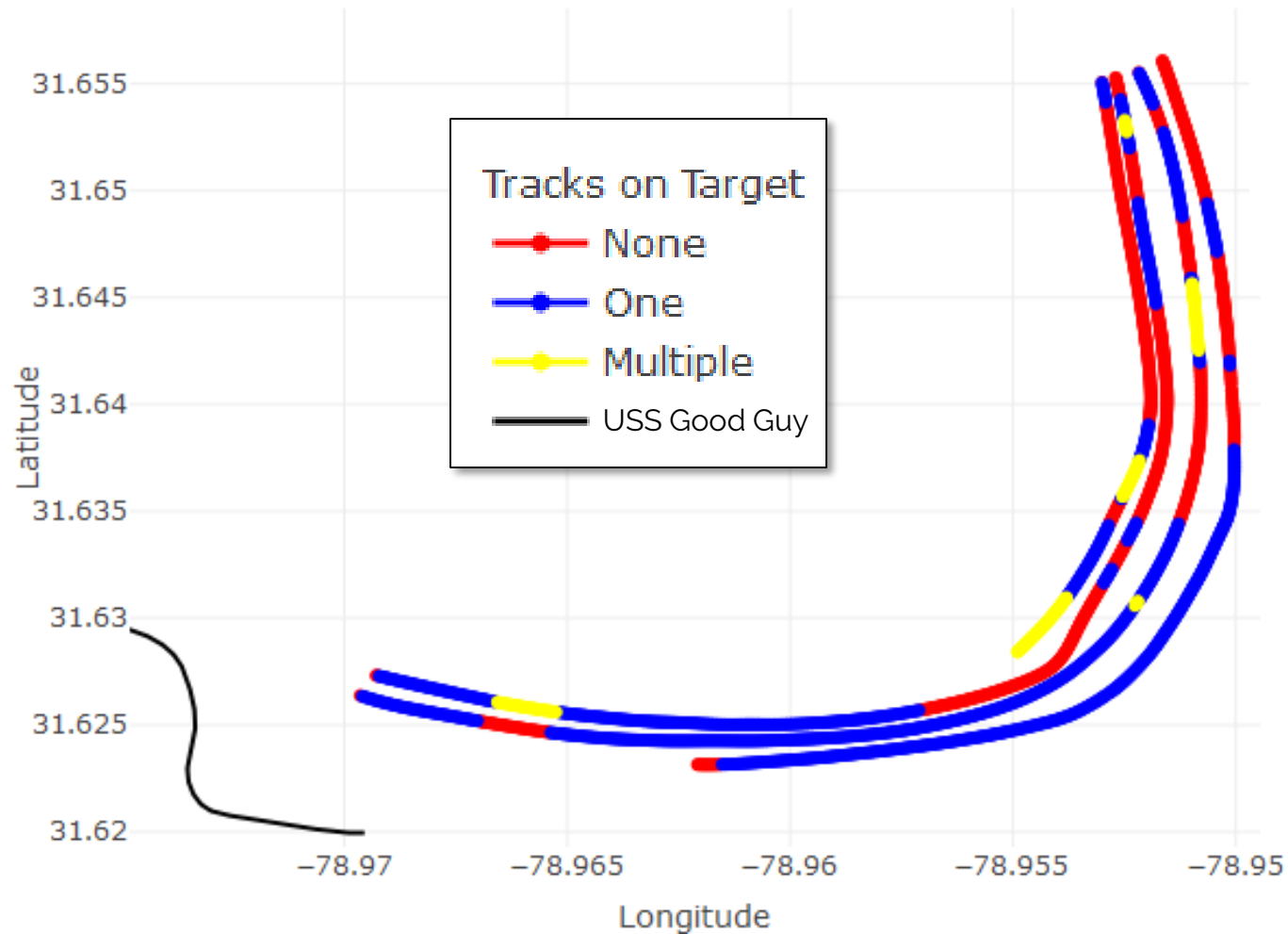


Target D

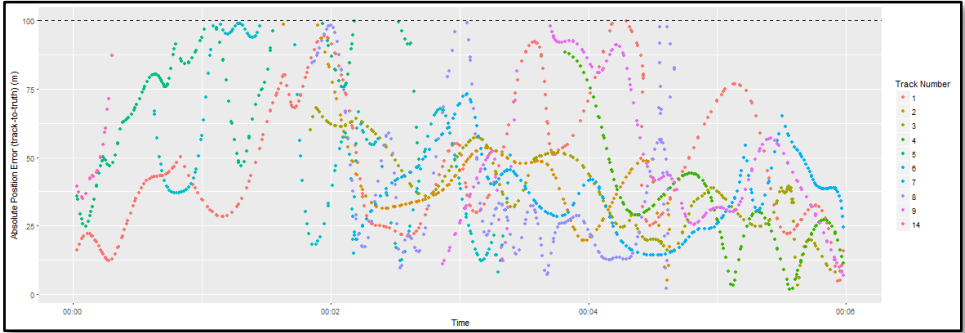
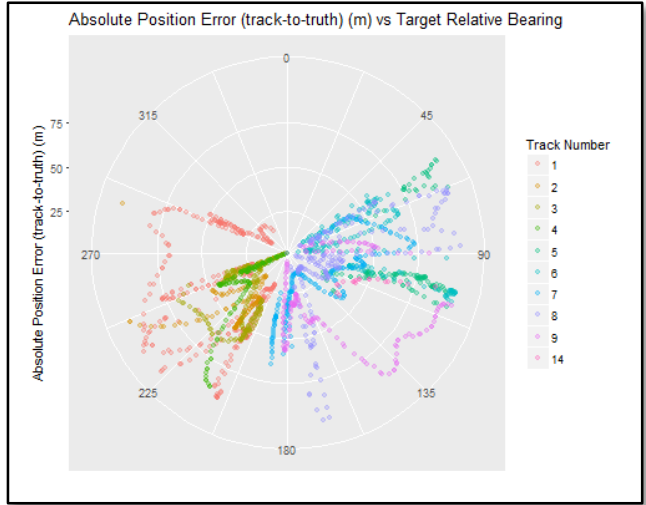
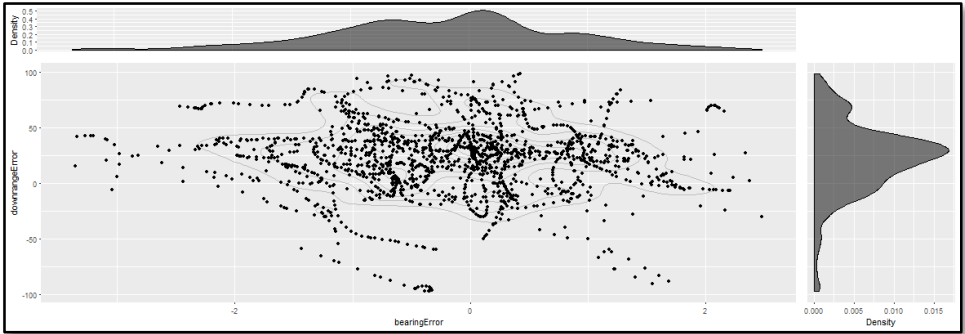
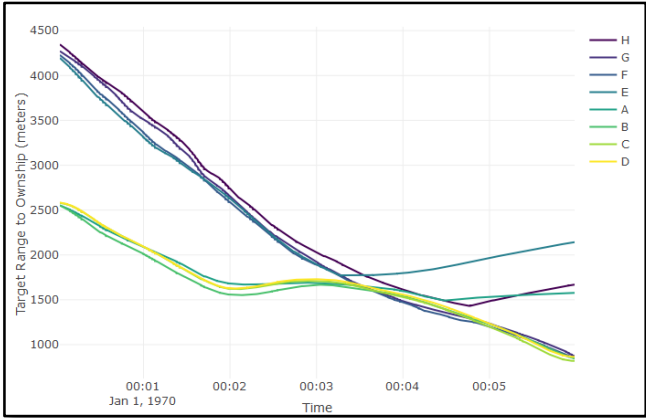
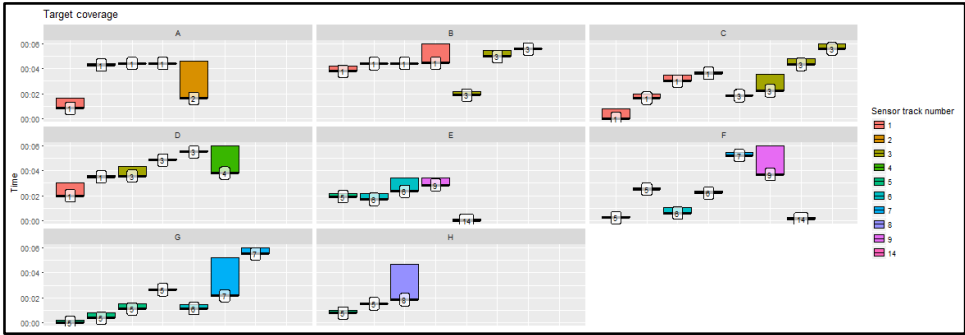
With track-to-truth assignments, we can start answering questions

Target Name	Detection Range (km)	Detection Time	Avg Position Error (m)
F-18	2.16	1/1/2015 0:15	46.36
EA-18G	1.58	1/1/2015 0:22	45.6
P-8A	2.58	1/1/2015 0:07	40.54
LCS 4	4.2	1/1/2015 1:13	31.46

With track-to-truth assignments, we can start answering questions



Nautilus produces a number of different plots



Our approach, Nautilus, compares sensor reports to reality

Nautilus is designed for **operational testing**, so we're interested in the **correctness of actionable information** provided by the sensor

Nautilus is system agnostic

What you need to use Nautilus

- A basic understanding of R
- Truth data for ownship (the sensor) and targets: time, latitude, longitude, altitude
- The sensor tracks: time, latitude, longitude, altitude, track ID
- Only 2 (or 3) user inputs to assign tracks to targets
- Patience and a willingness to experiment

Which target is each track following?

Target A

Target B

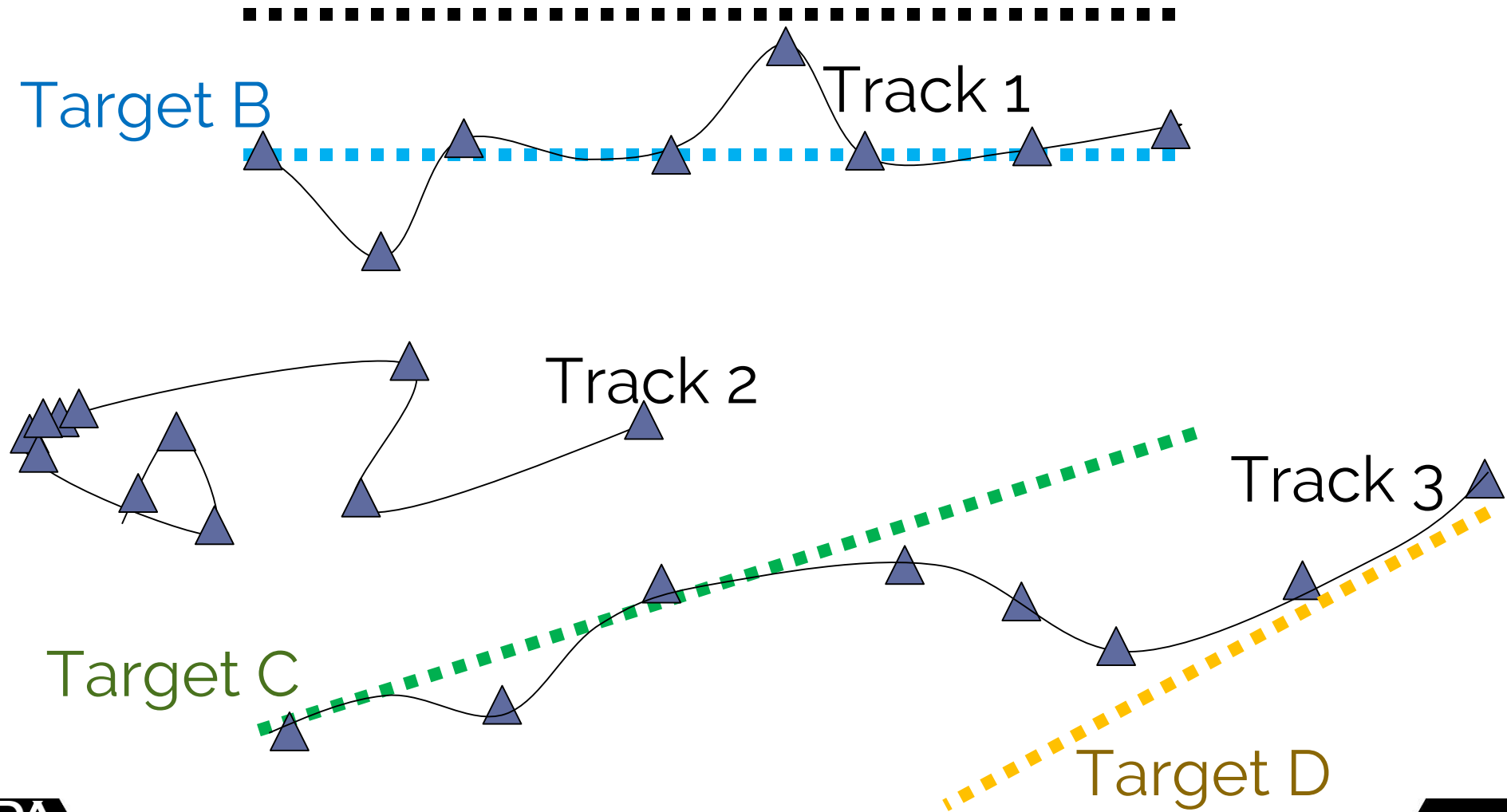
Track 1

Track 2

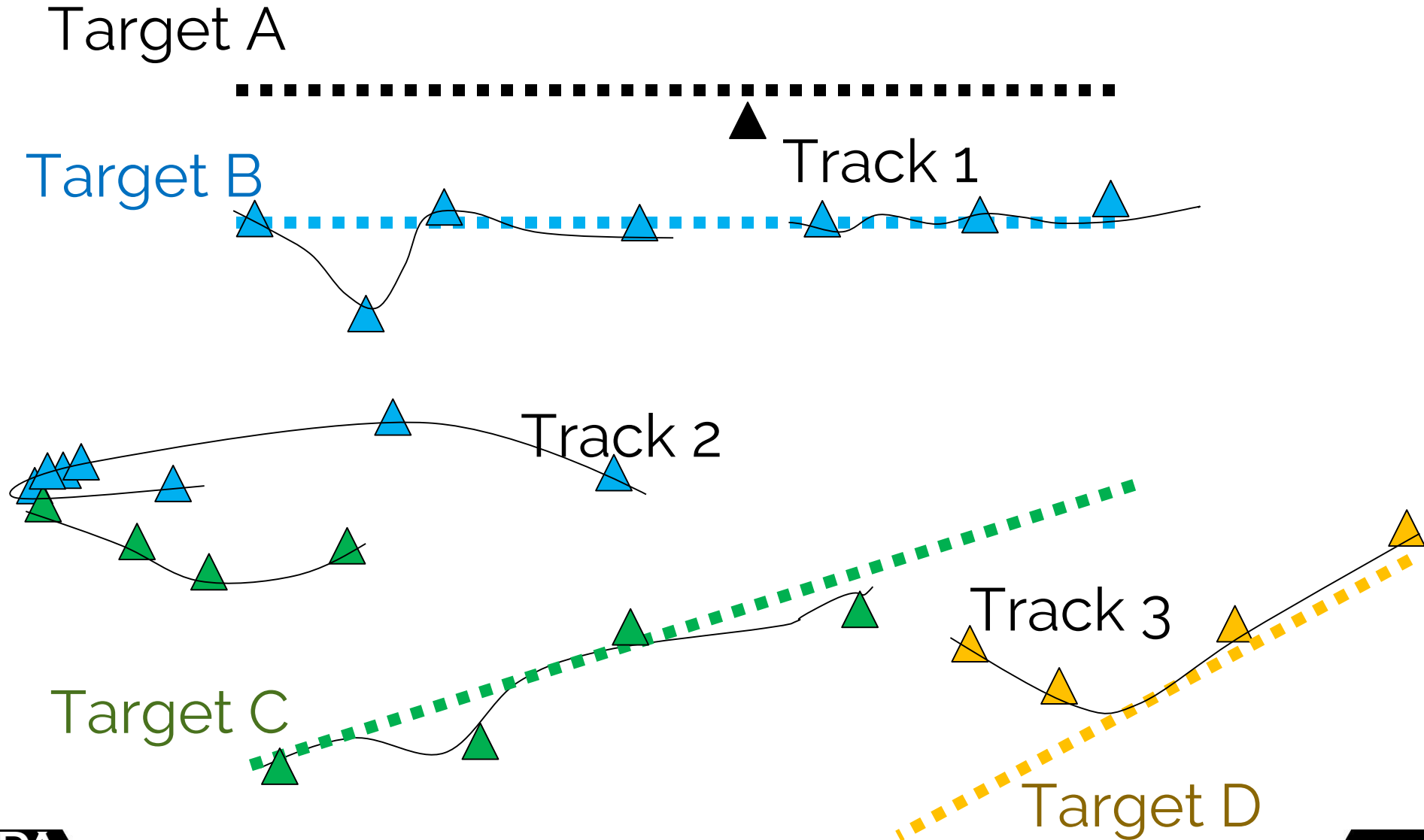
Track 3

Target C

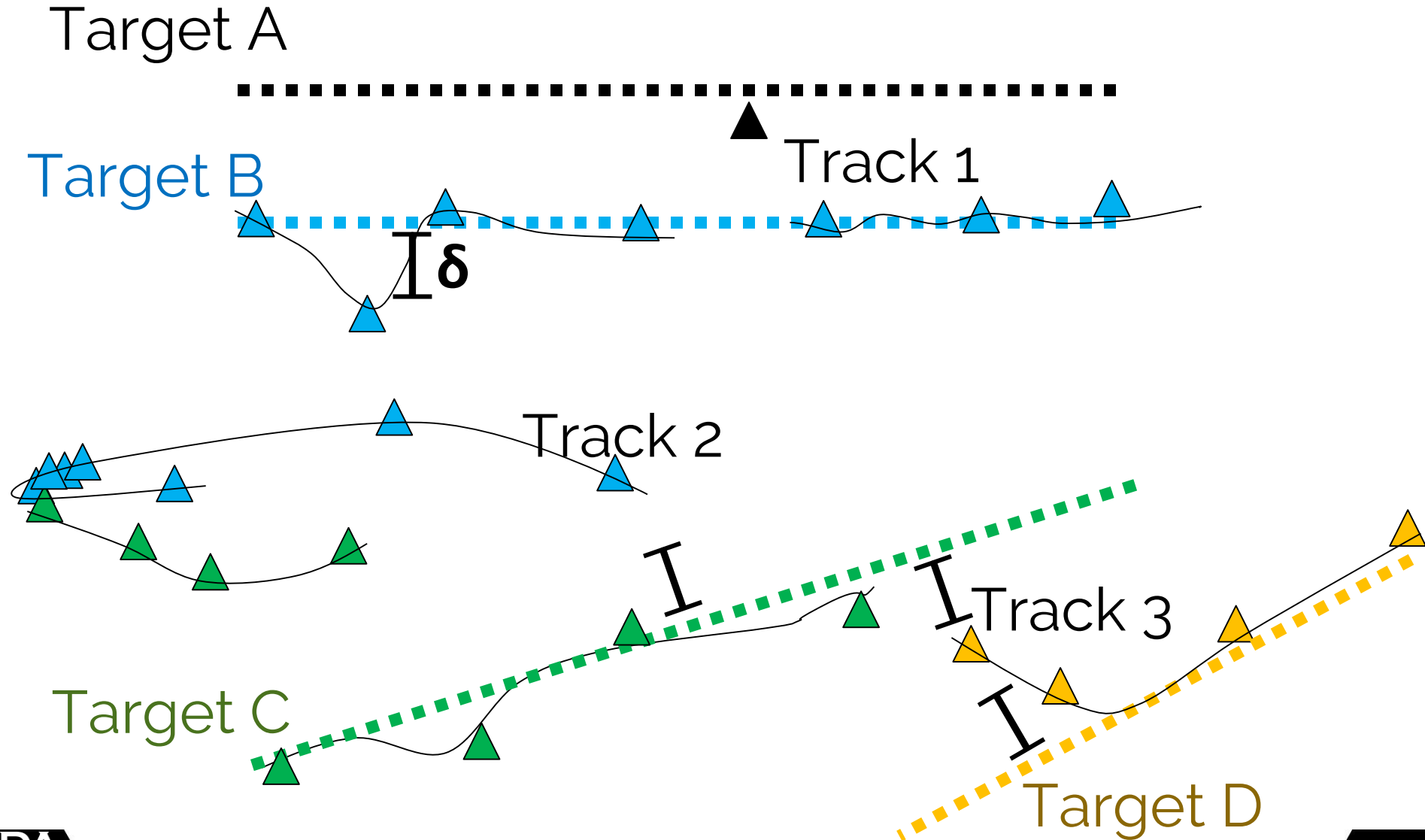
Target D



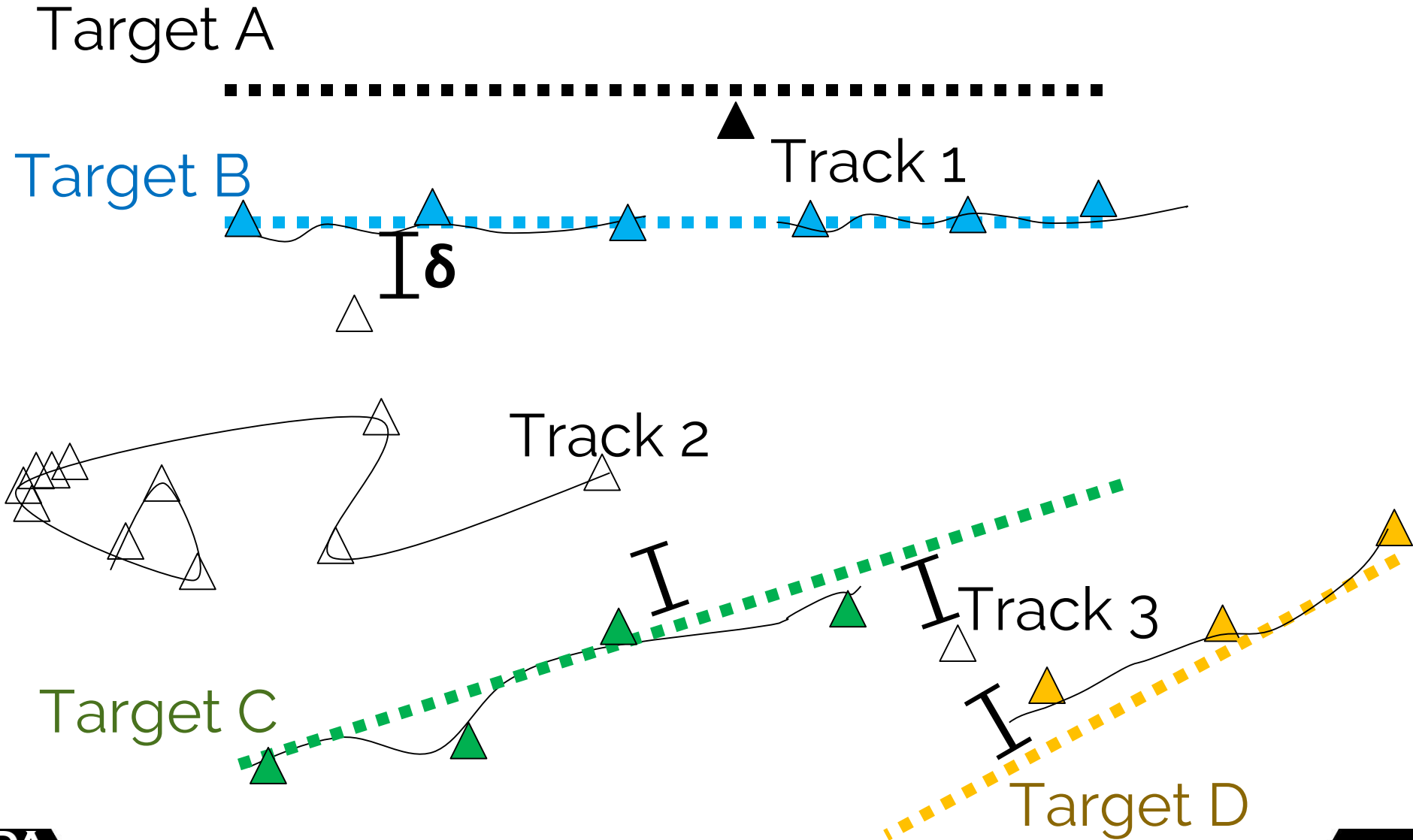
Step 1: Nautilus assigns each sensor point to the closest target



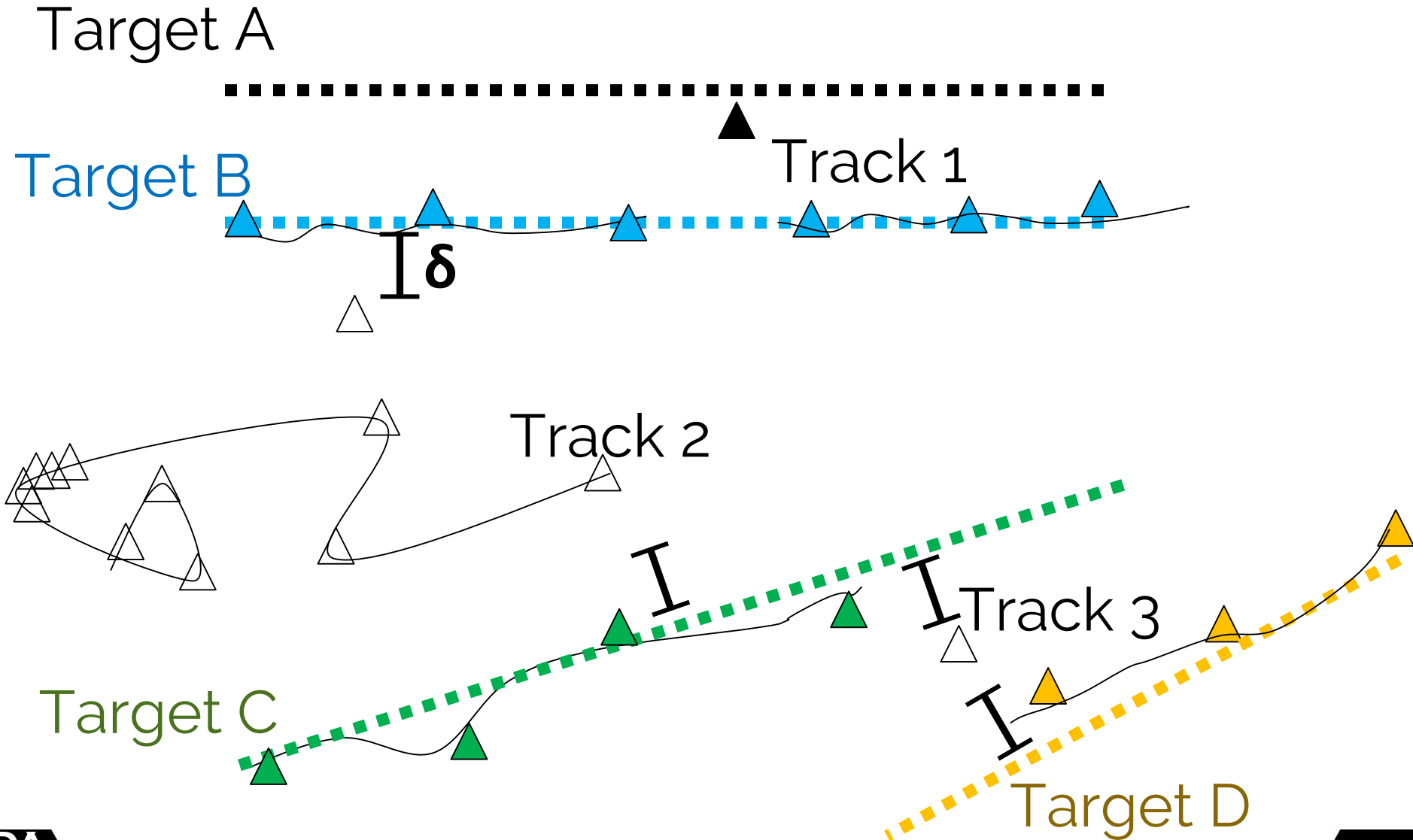
Applies a cutoff distance: any sensor point 'wronger' than δ is a false track



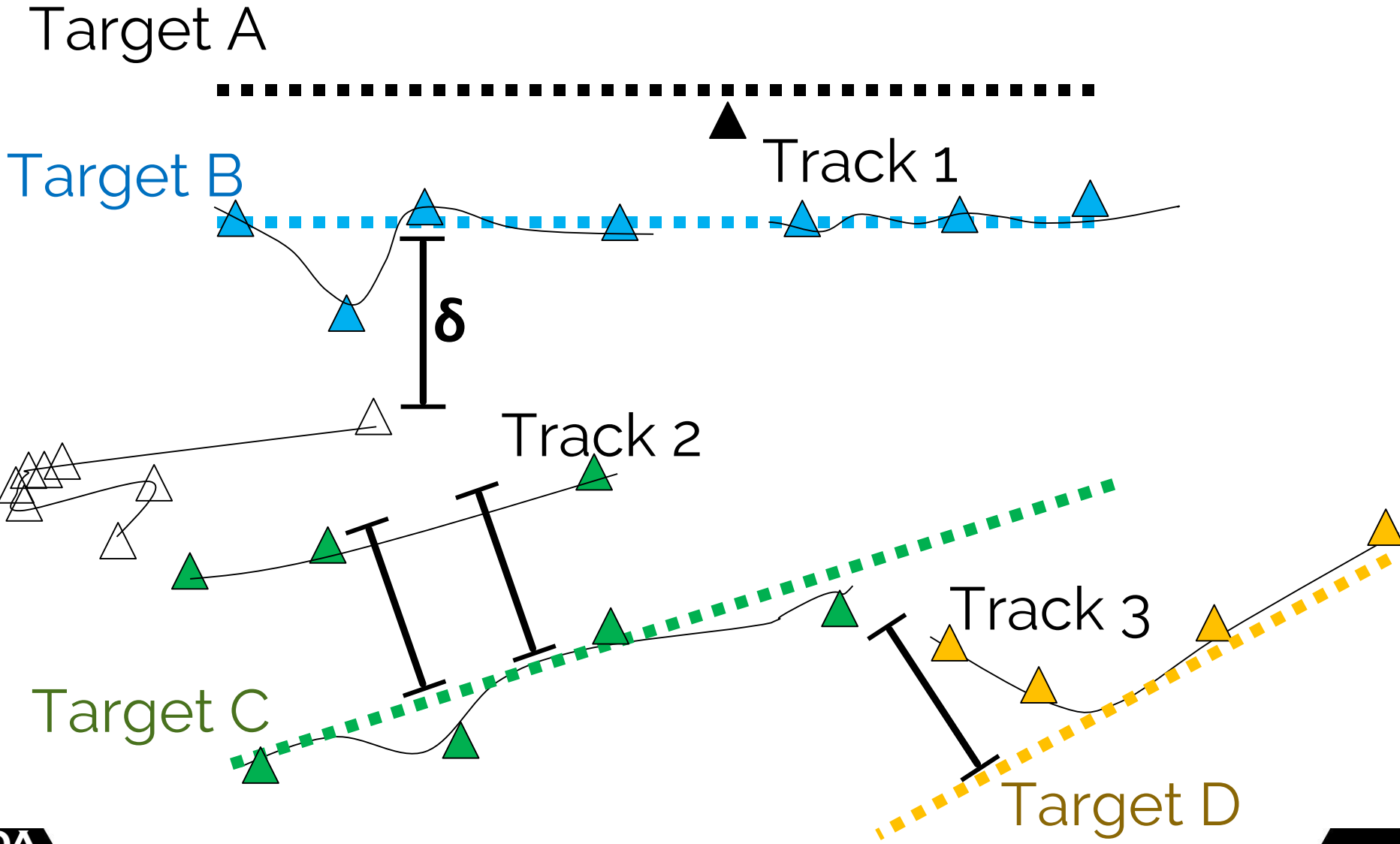
Step 2: Removes points for which we have no nearby truth data



A small cutoff may unfairly reduce error estimates...



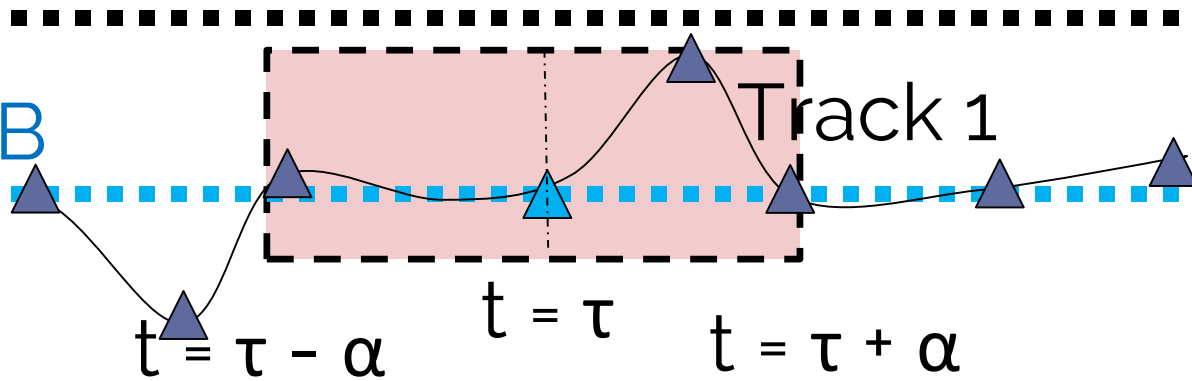
But a large cutoff may include additional false track points



Changing the method to a window method will smooth the target assignment

Target A

Target B



Track 2

Track 3

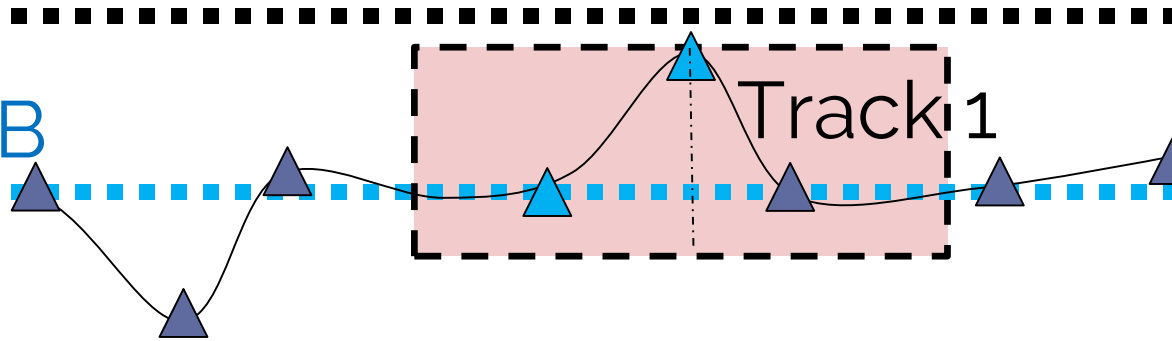
Target C

Target D

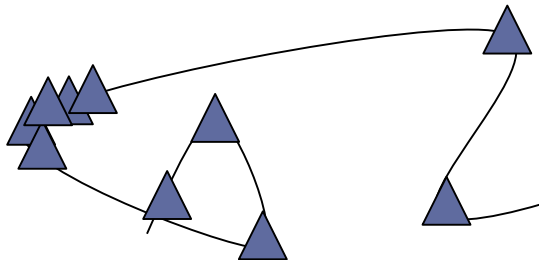
Changing the method to a window method will smooth the target assignment

Target A

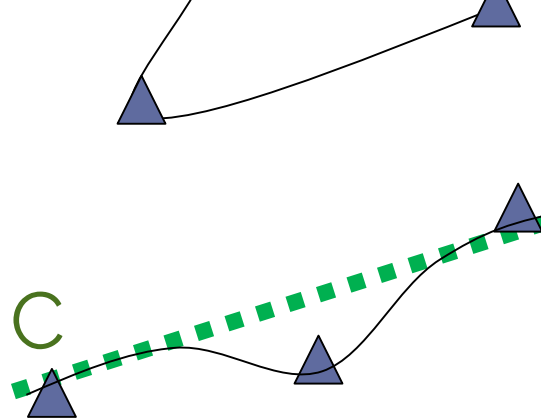
Target B



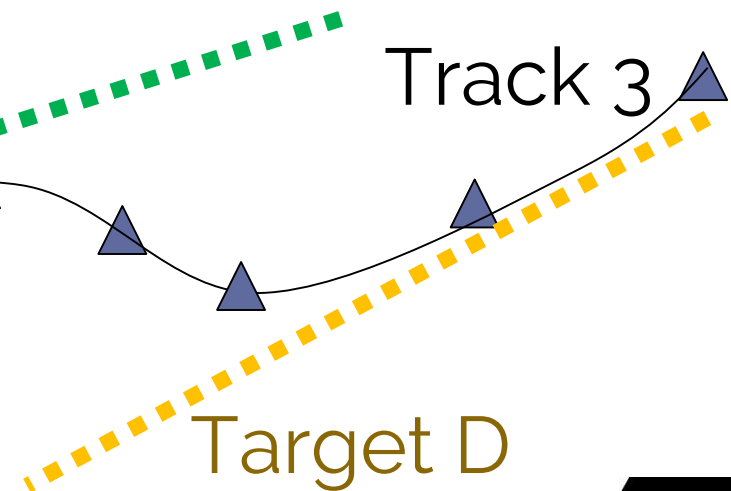
Track 2



Target C



Track 3

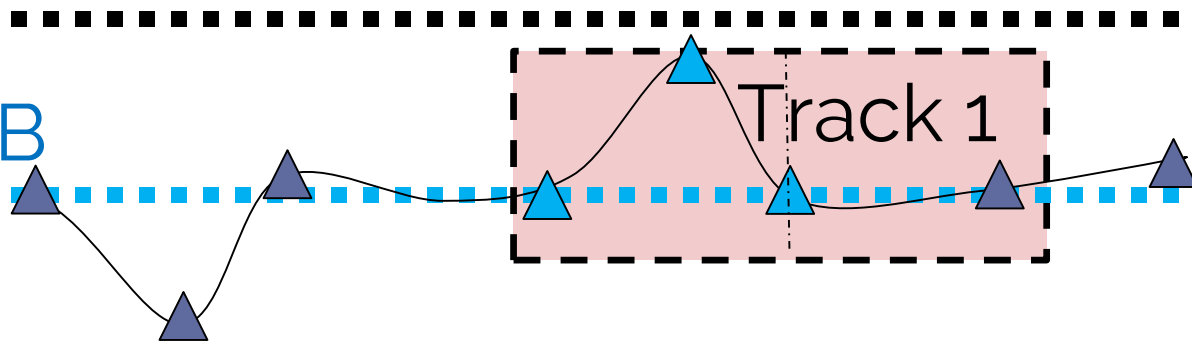


Target D

Changing the method to a window method will smooth the target assignment

Target A

Target B

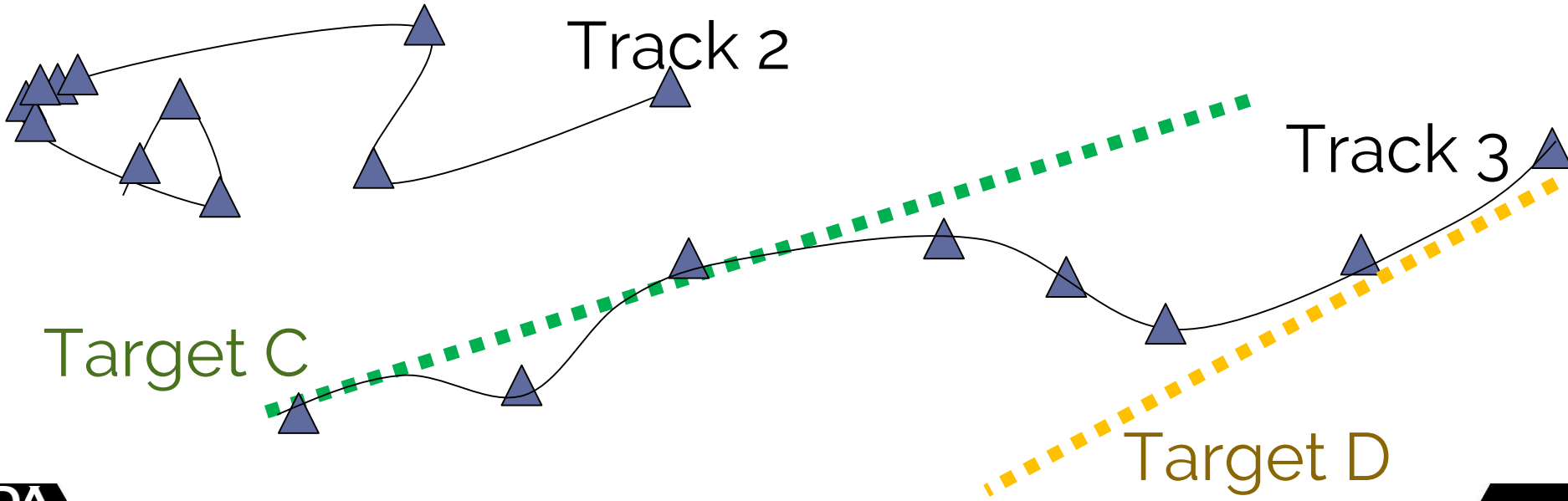


Track 2

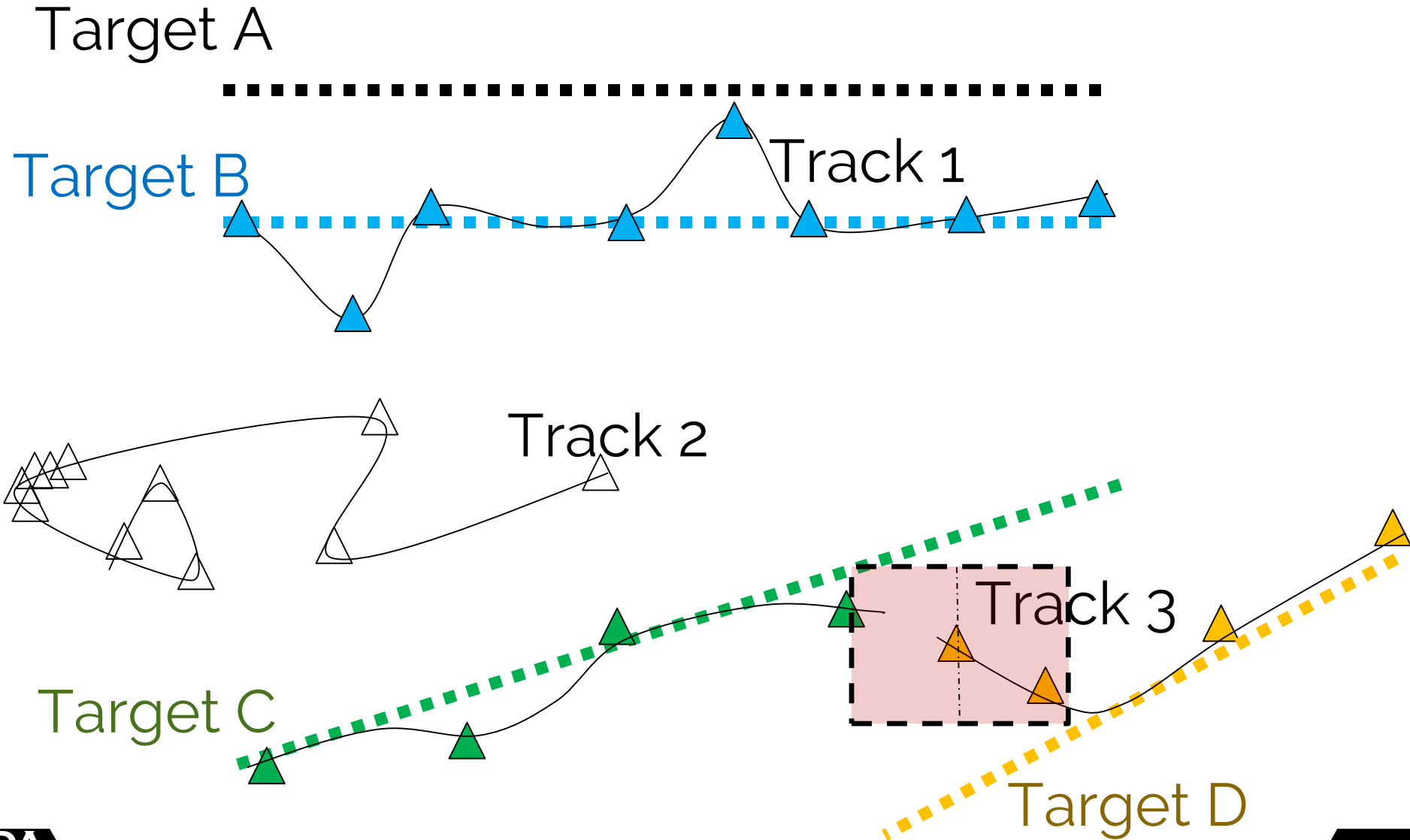
Track 3

Target C

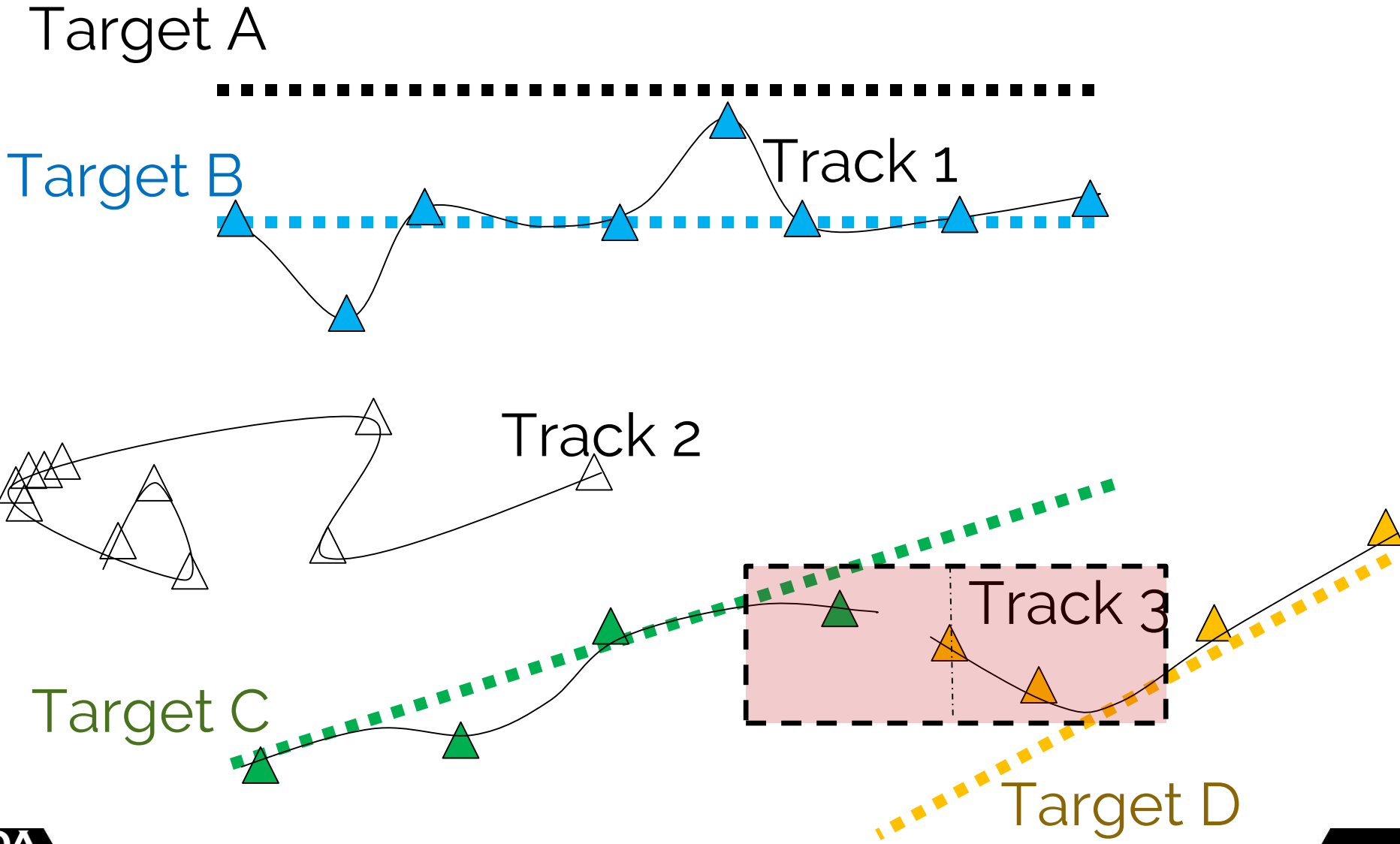
Target D



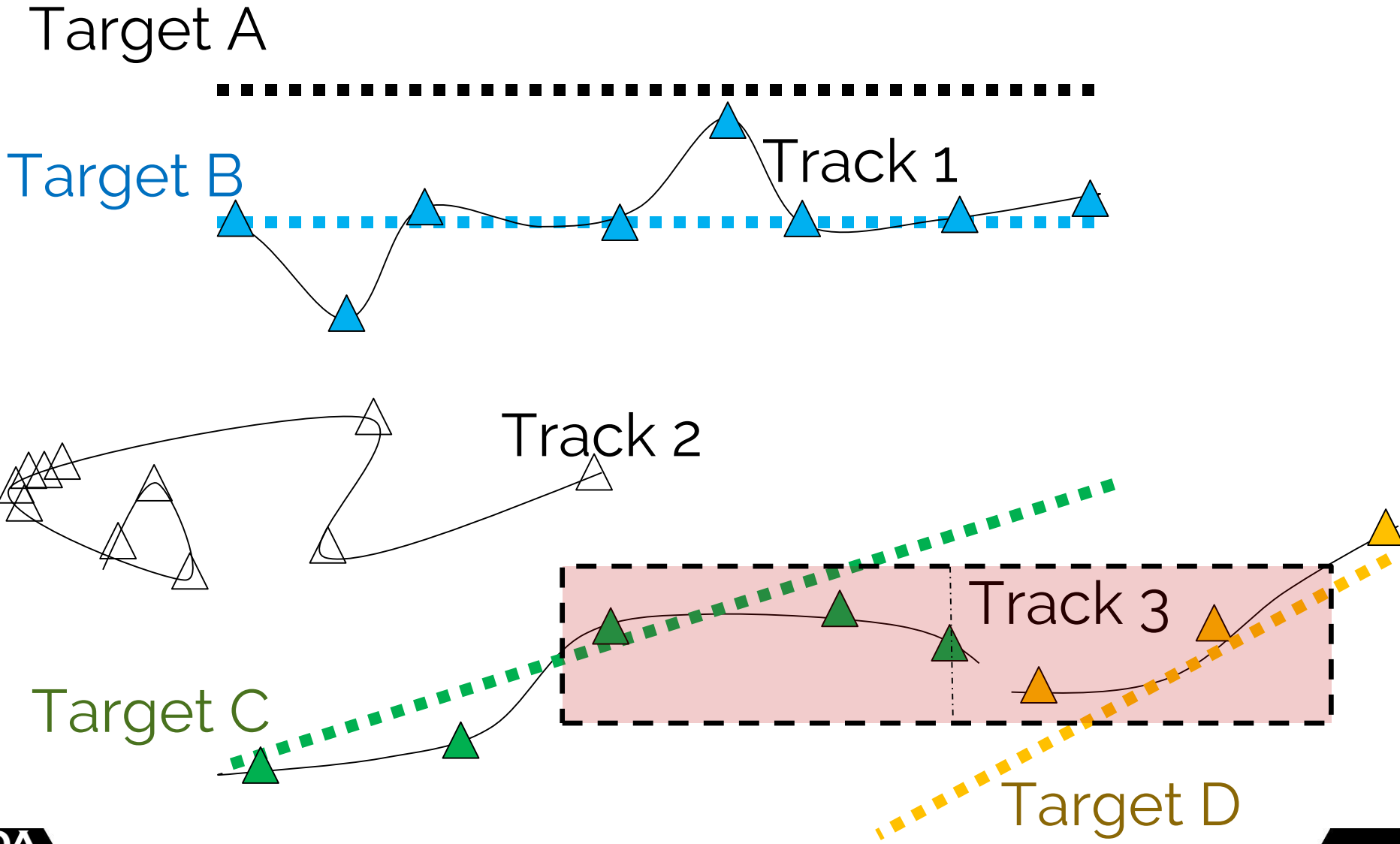
Adjusting window size may change results



Adjusting window size may change results



Adjusting window size may change results



How do we know our assignments are right?

Nautilus automates, you analyze

Because it is sensor system agnostic, Nautilus cannot judge the reasonability of its answers



R Demonstration

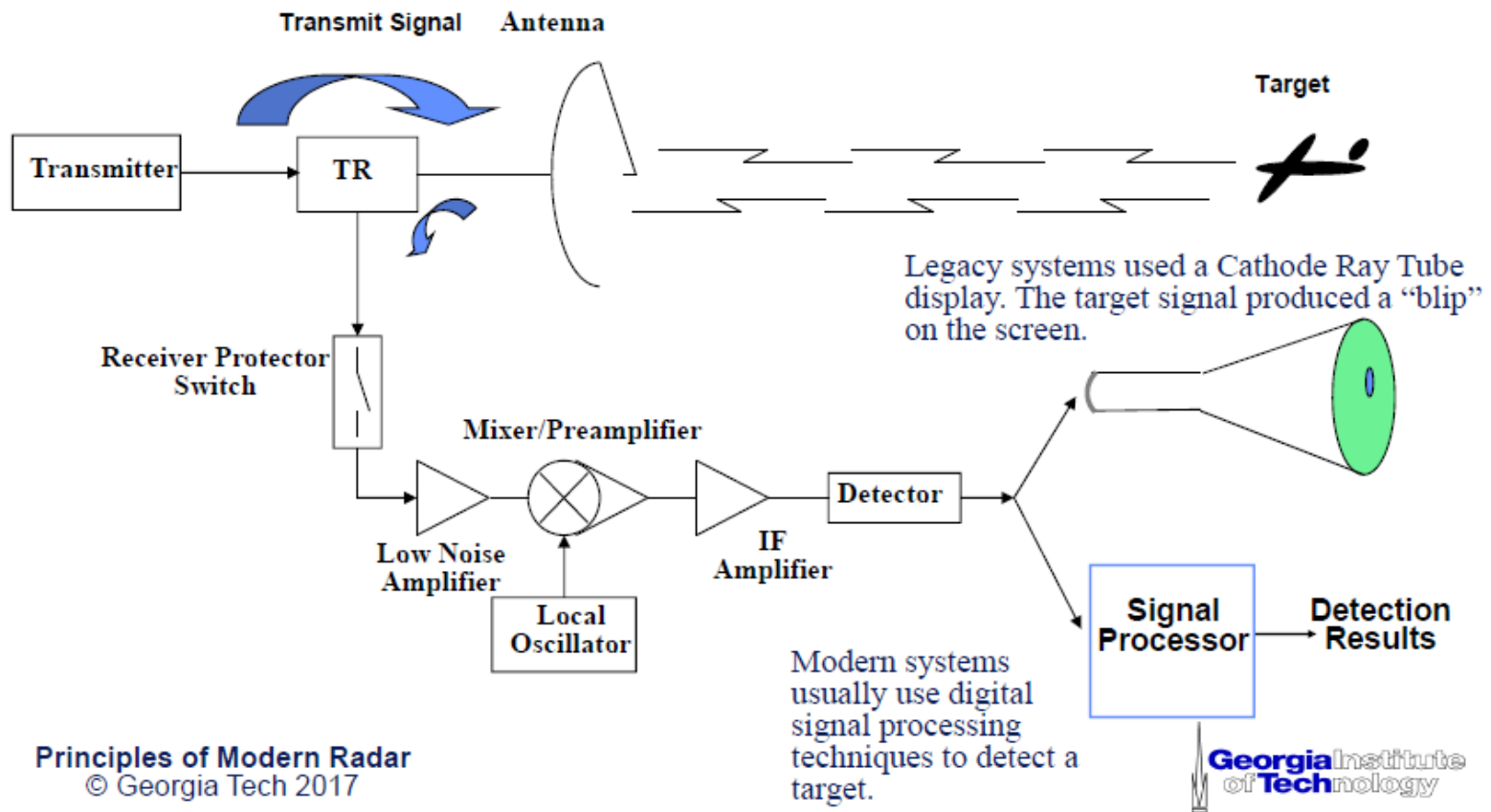
Our code is modular – use the pieces that you find useful

- Automatically calculate values like relative bearing, target aspect, and slant range between targets
- Useful interactive figures for event reconstruction
- Import and export data from/to SIMDIS

Backup

What do we even mean by performance?

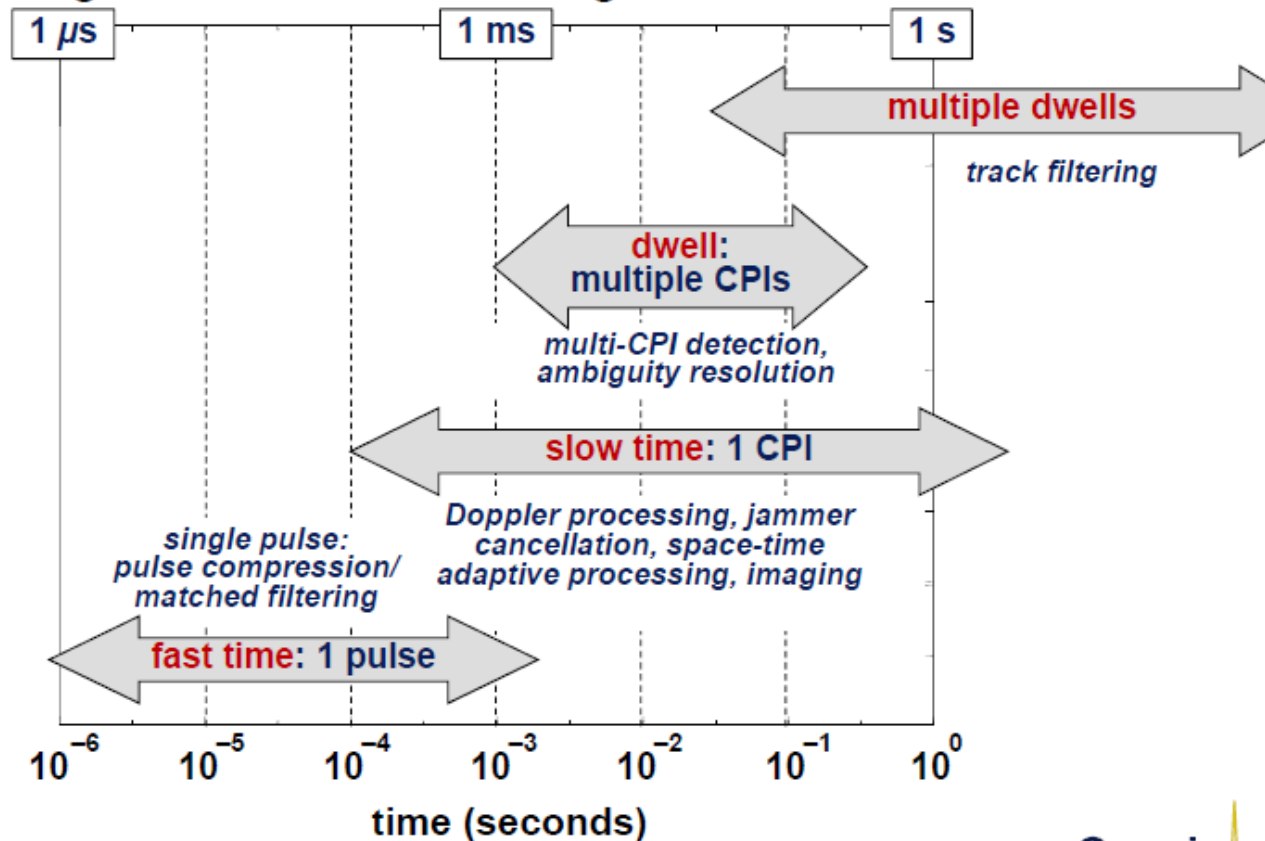
Simplified Radar Block Diagram



What do we even mean by performance?

Radar Data Time Scales

- Radars collect and combine data over time scales covering several orders of magnitude



Nautilus tries to do what your eyes do automatically

1. Calculate the distance from each track to each target
(User specifies **assignment method** and **cutoff distance**)
2. Assign track to closest target (caution: reasonable assumption)
3. Flag false tracks (distance $> \delta$)
4. Generate figures and diagnostics