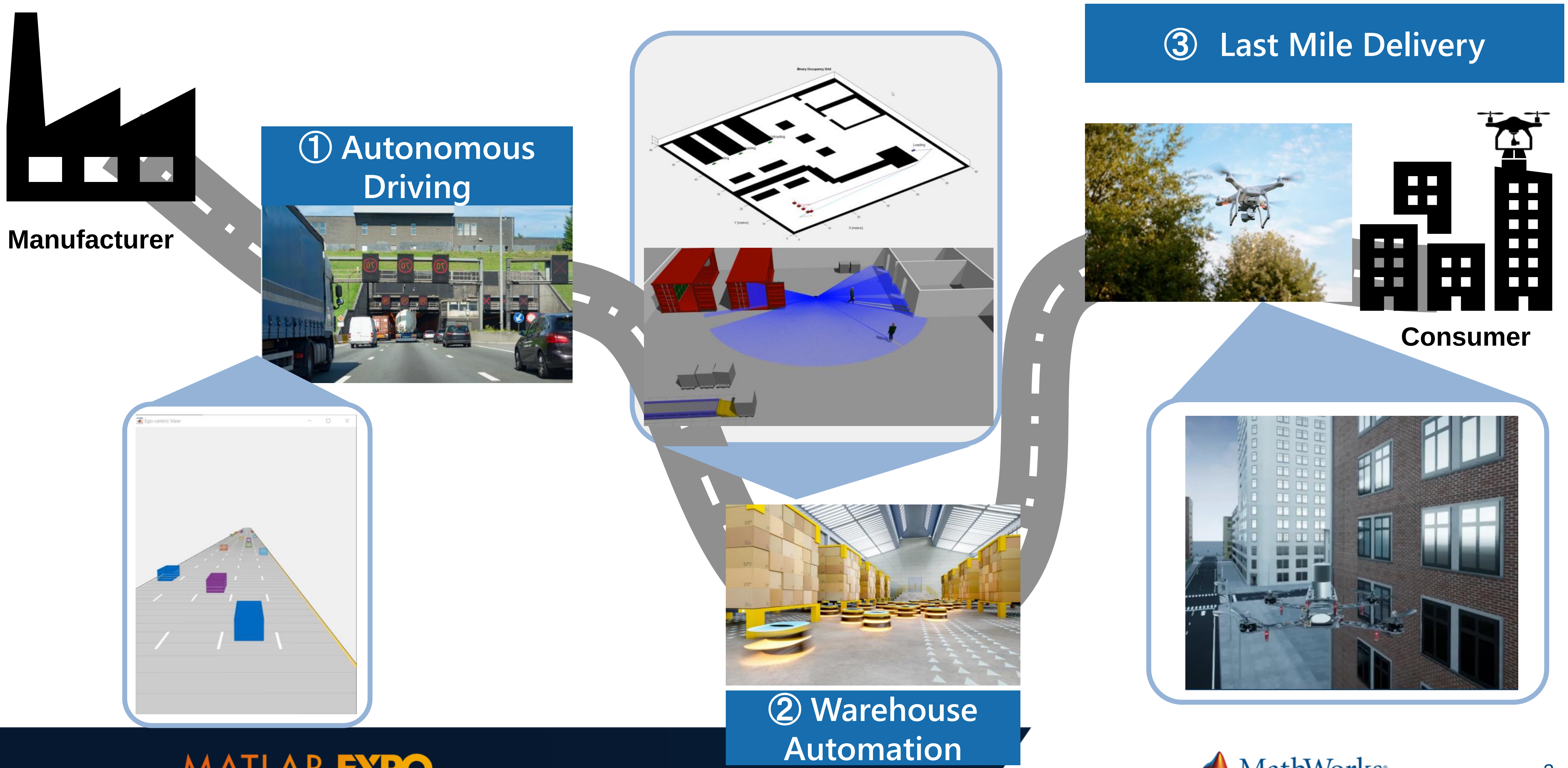
Sensor Fusion and Navigation for Autonomous Systems Using MATLAB & Simulink

Abhishek Tiwari
Application Engineering

MATLAB EXPO

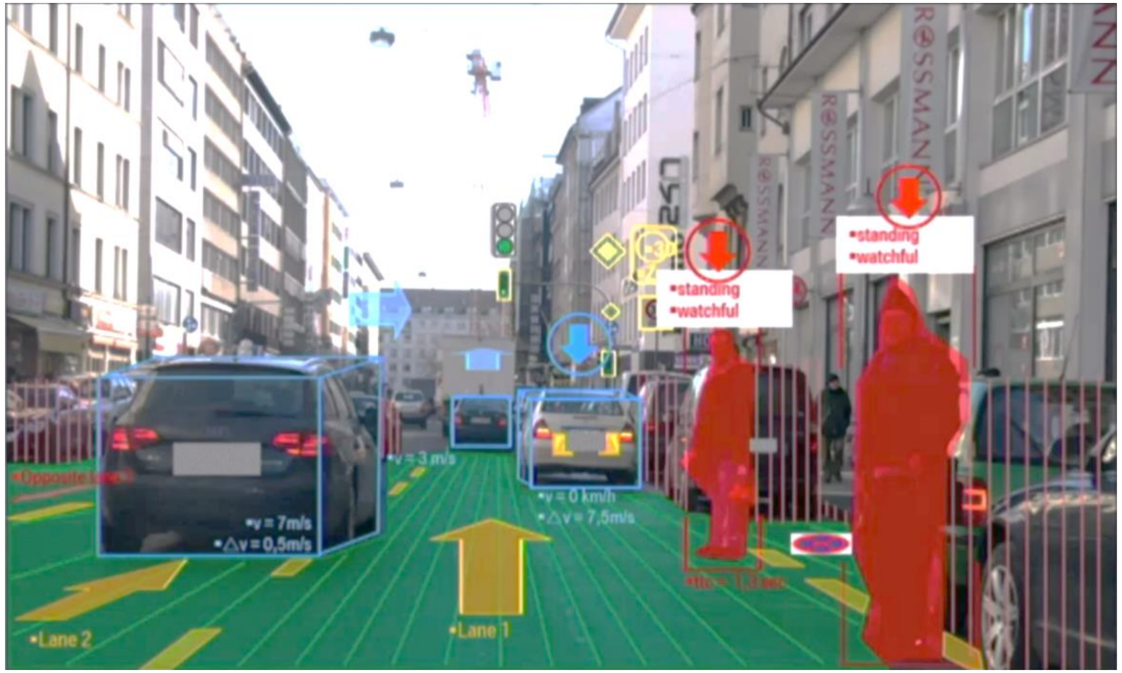
MathWorks®

Smart autonomous package delivery



Capabilities of an Autonomous System

Perception



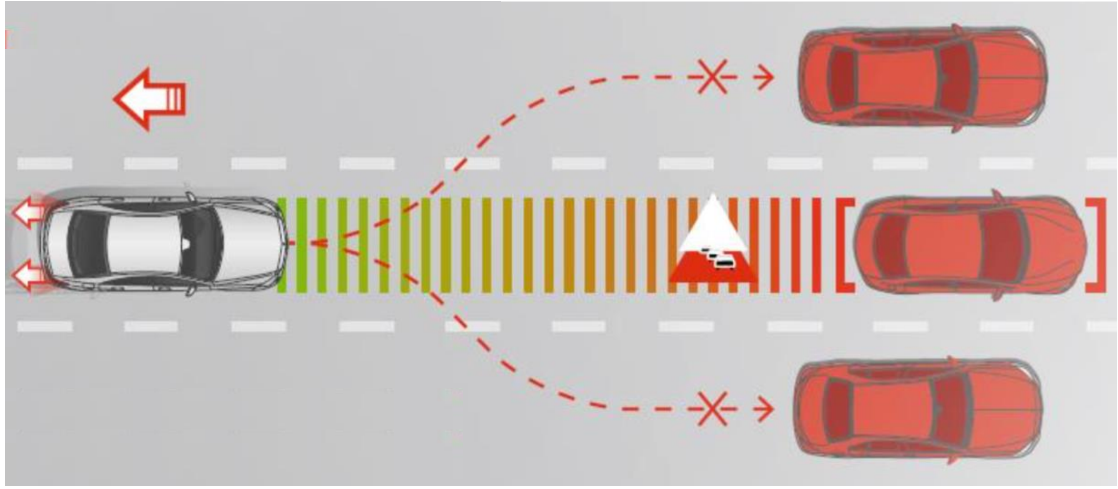
Some common Perception tasks

- Design localization algorithms
- Design environment mapping algorithms
- Design SLAM algorithms
- Design fusion and tracking algorithms
- Label sensor data
- Design deep learning networks
- Design radar algorithms
- Design vision algorithms
- Design lidar algorithms
- Generate C/C++ code

Capabilities of an Autonomous System

Perception

Planning



Some common Planning tasks

- Visualize street maps
- Connect to HERE HD Live Map
- Design local and global path planners
- Design vehicle motion behavior planners
- Design trajectory generation algorithms
- Generate C/C++ code

Capabilities of an Autonomous System

Perception

Planning

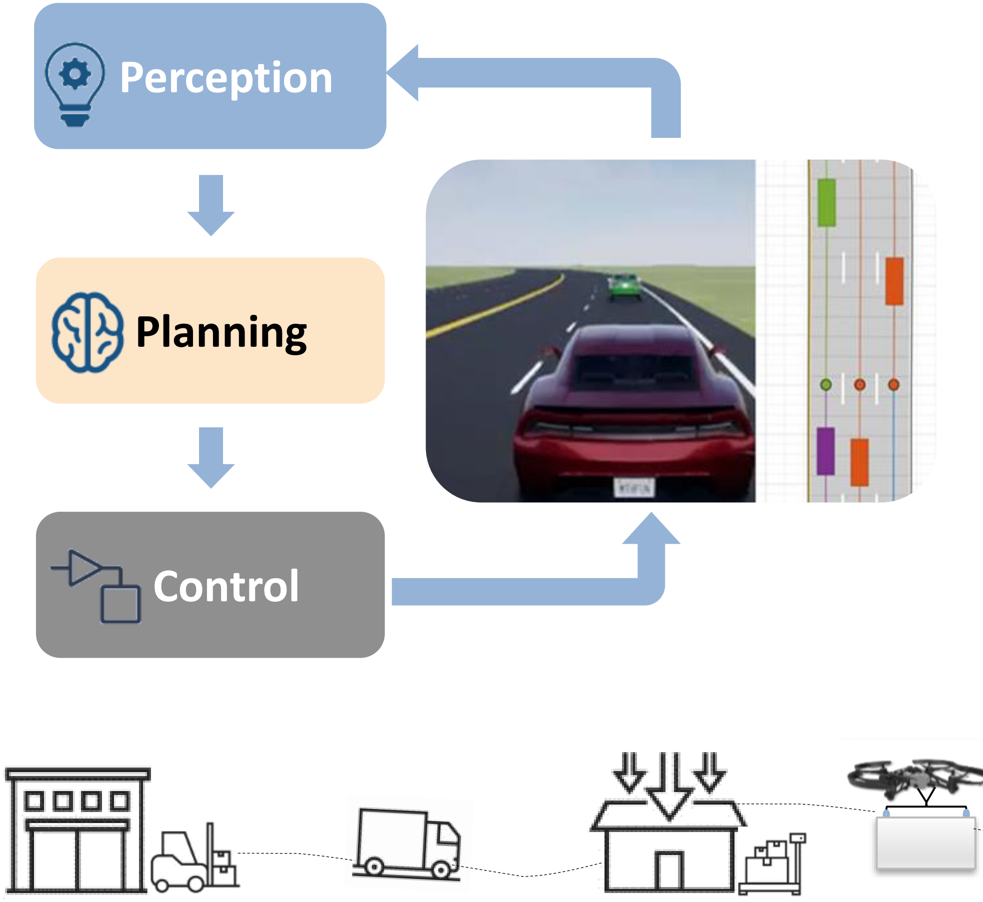
Control

Some common Control **tasks**

- Connect to recorded and live CAN data
- Design reinforcement learning networks
- Model vehicle dynamics
- Automate regression testing
- Prototype on real-time hardware
- Design path tracking controllers
- Design model-predictive controllers
- Generate production C/C++ code
- Generate AUTOSAR code
- Certify for ISO26262

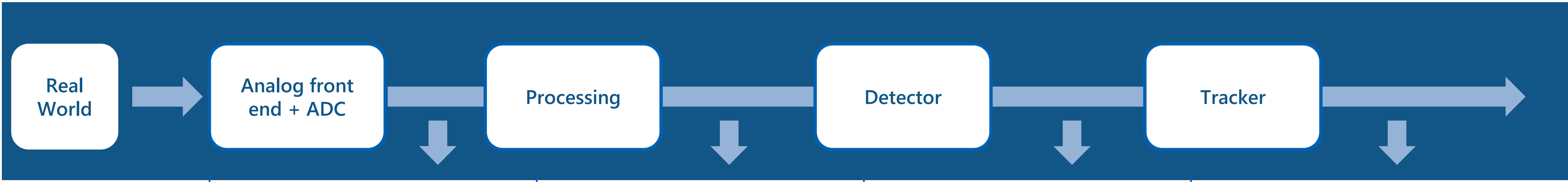
In this talk, you will learn

- Reference workflow for autonomous navigation systems development
- MATLAB and Simulink capabilities to design, simulate, test, deploy algorithms for sensor fusion and navigation
  - Perception algorithm design
  - Fuse sensor data to maintain situational awareness
  - Mapping and Localization
  - Path planning and path following control

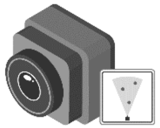
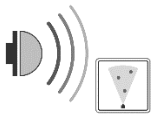
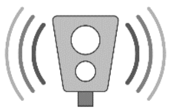


Sensor outputs vary along the processing chain

Sensor outputs vary along the processing chain

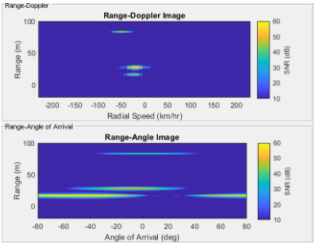


Virtual Scenario

	Signals/Pixels	Processed Signals	Object Detections	Tracks
 Camera				
 Radar				
 Lidar				

Sensor outputs vary along the processing chain

Real World → Analog front end + ADC → Processing → Detector → Tracker →

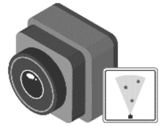
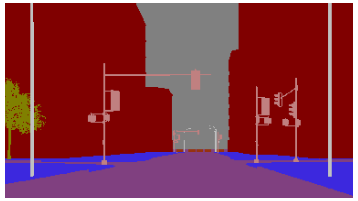
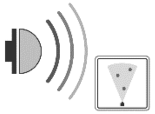
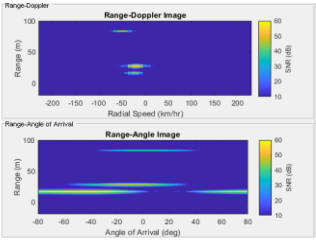
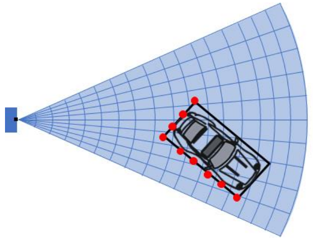
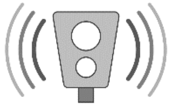
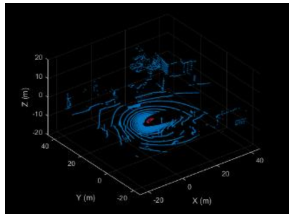
	Signals/Pixels	Processed Signals	Object Detections	Tracks
Camera				
Radar				
Lidar				

Virtual Scenario

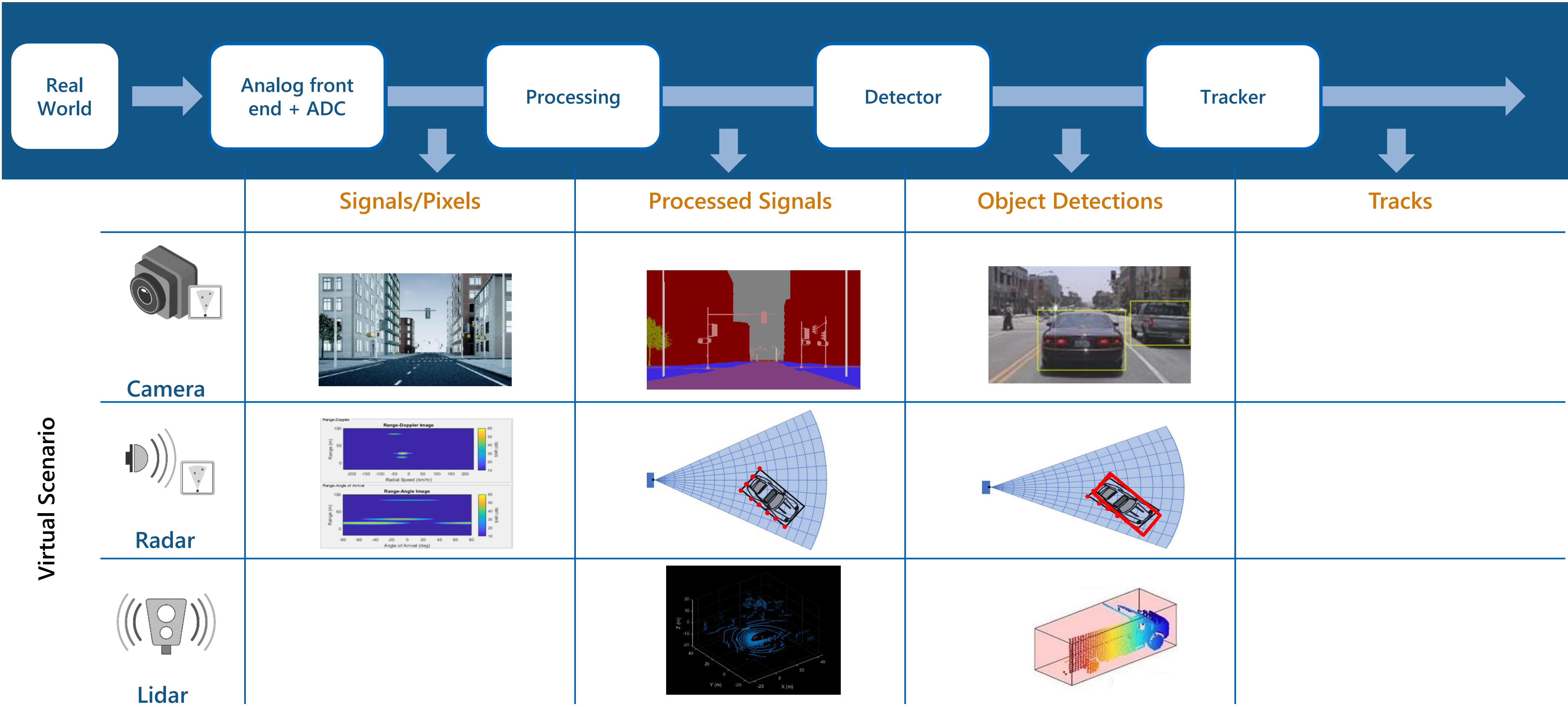
Sensor outputs vary along the processing chain

Real World → Analog front end + ADC → Processing → Detector → Tracker →

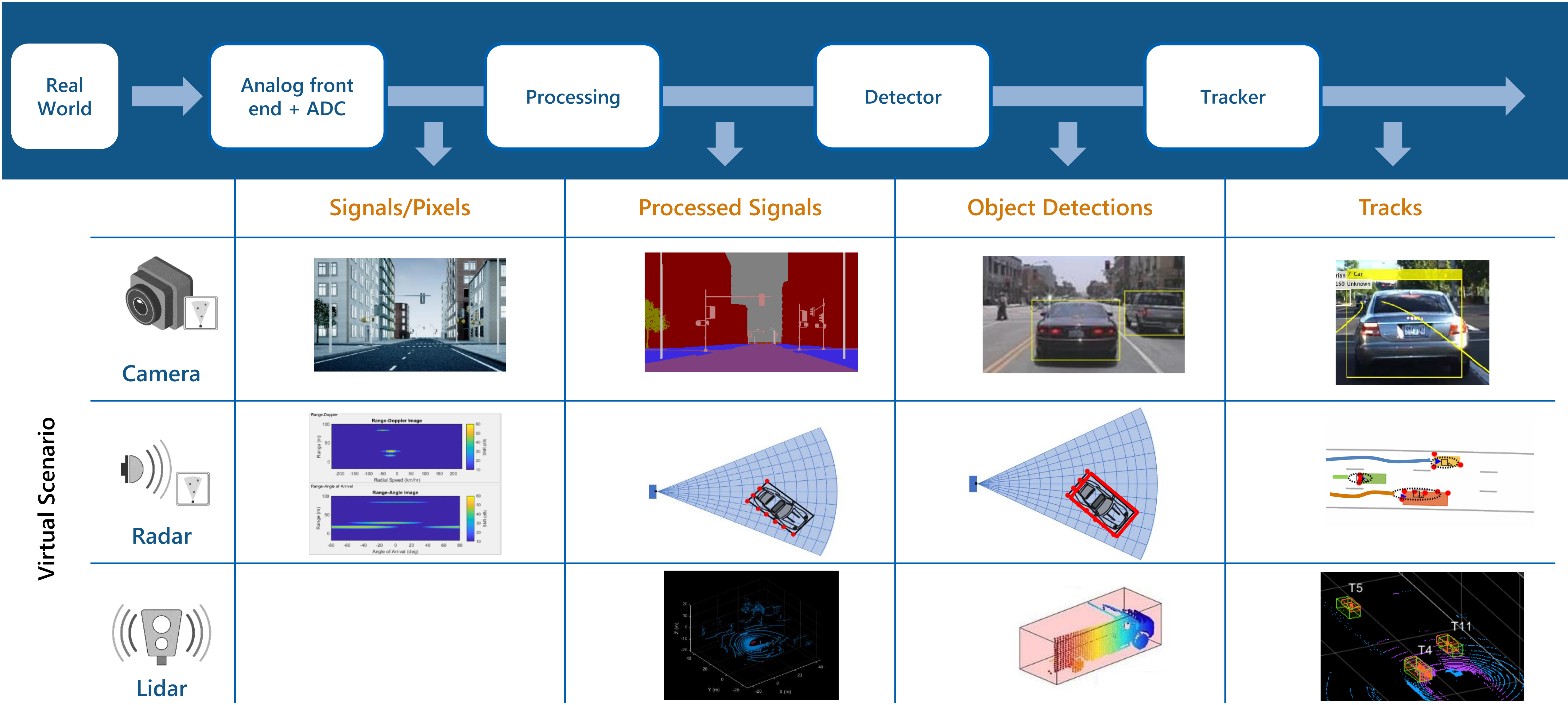
Virtual Scenario

	Signals/Pixels	Processed Signals	Object Detections	Tracks
 Camera				
 Radar				
 Lidar				

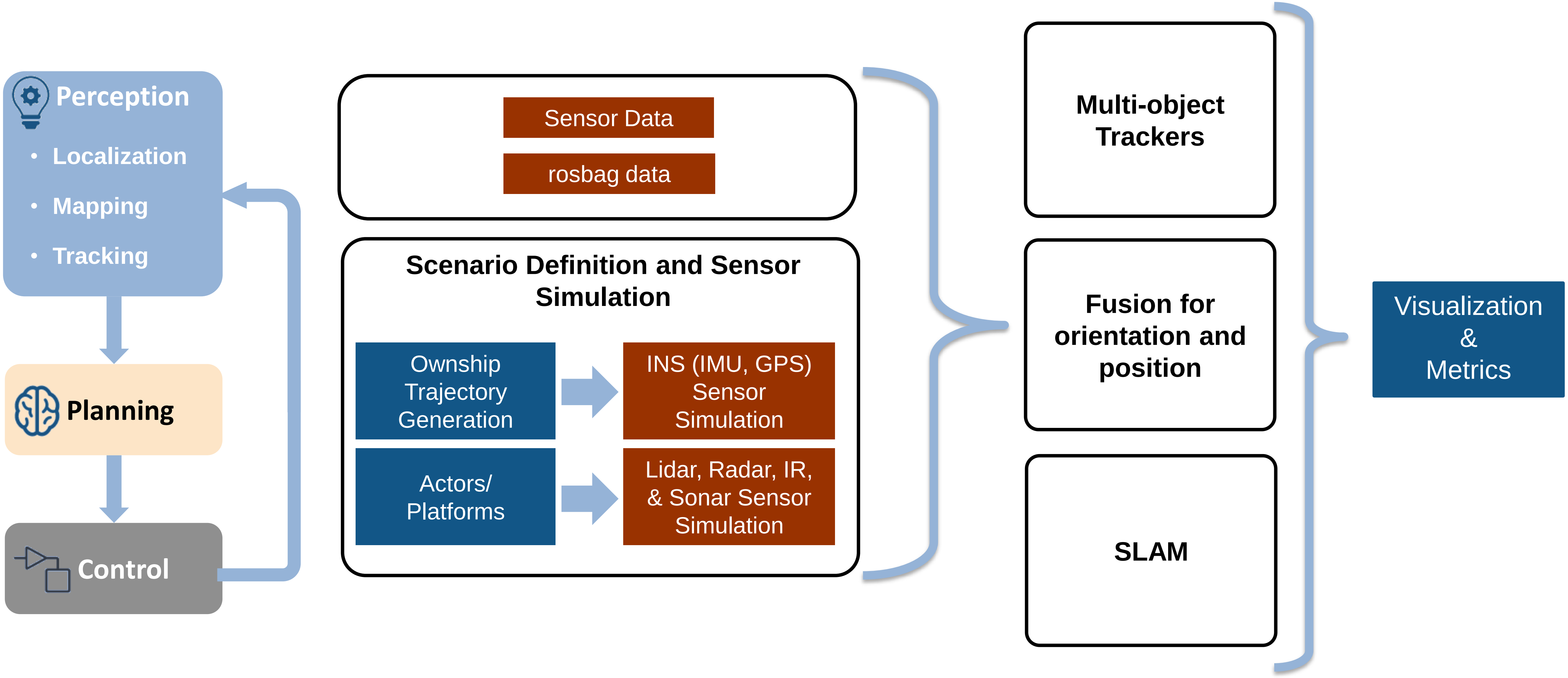
Sensor outputs vary along the processing chain



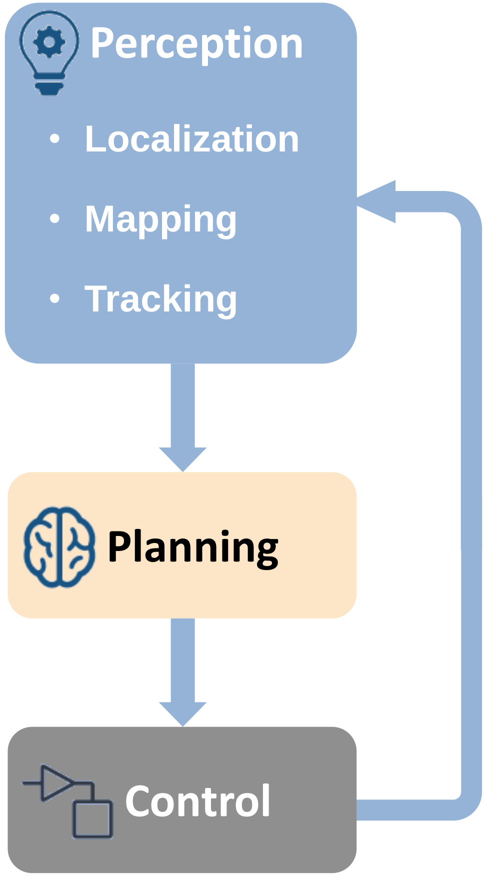
Sensor outputs vary along the processing chain



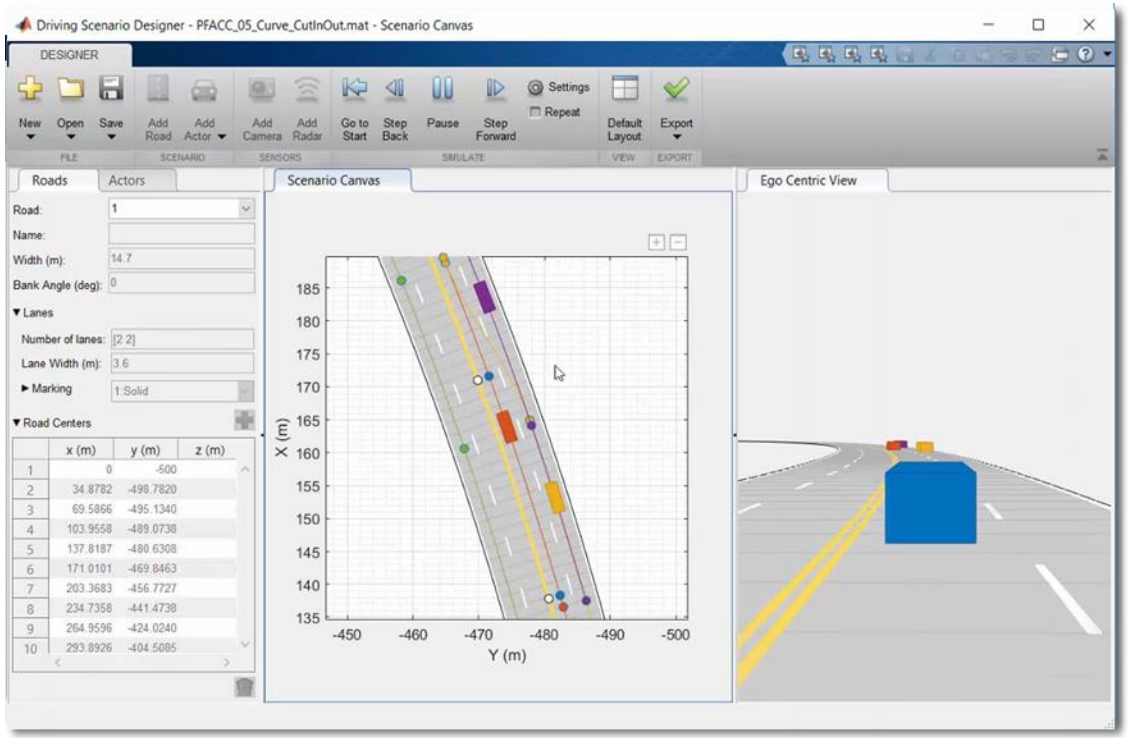
Many options to bring sensor data to perception algorithms



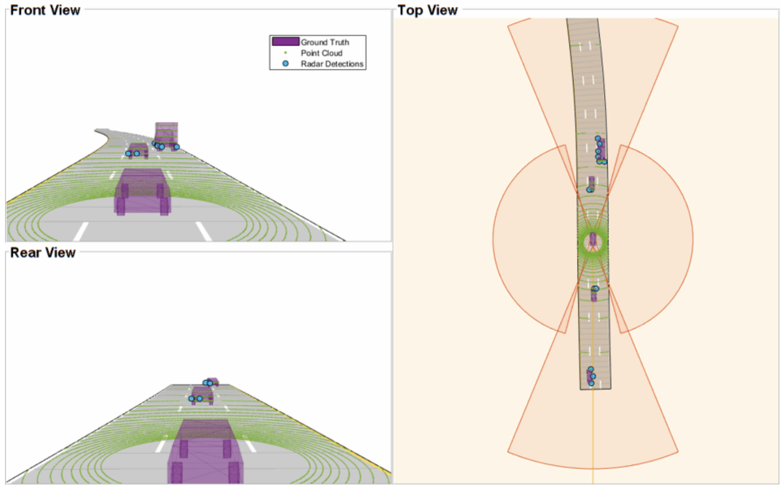
Live data can be augmented for a more robust testbench



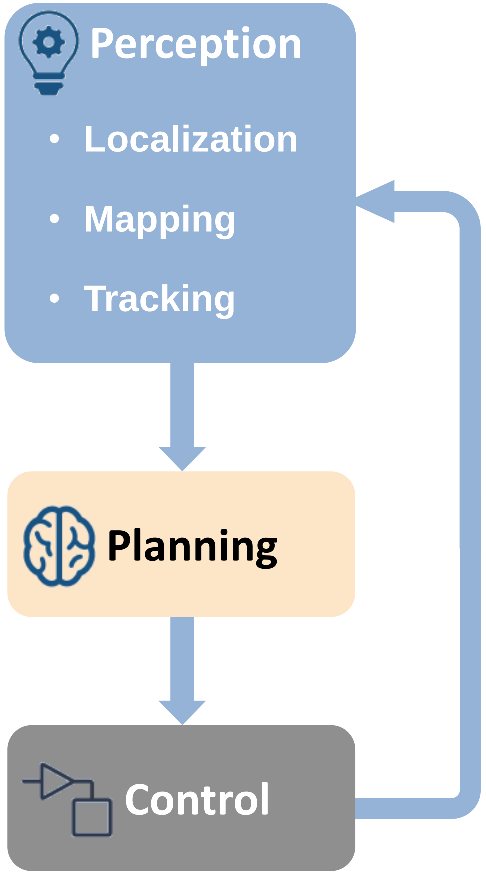
Define scenarios



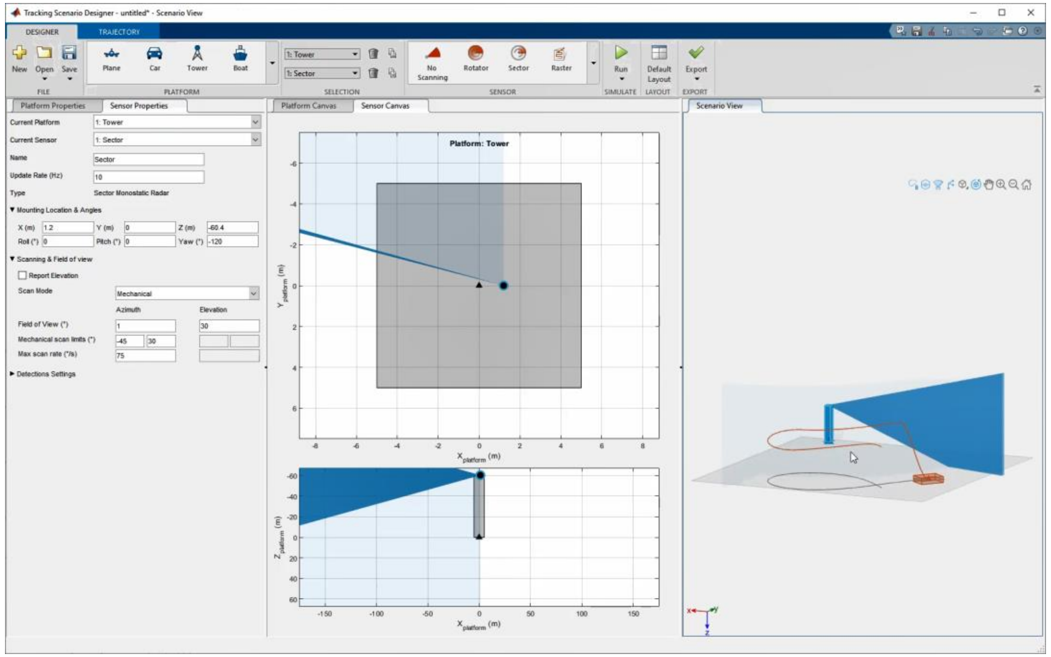
Simulate sensors



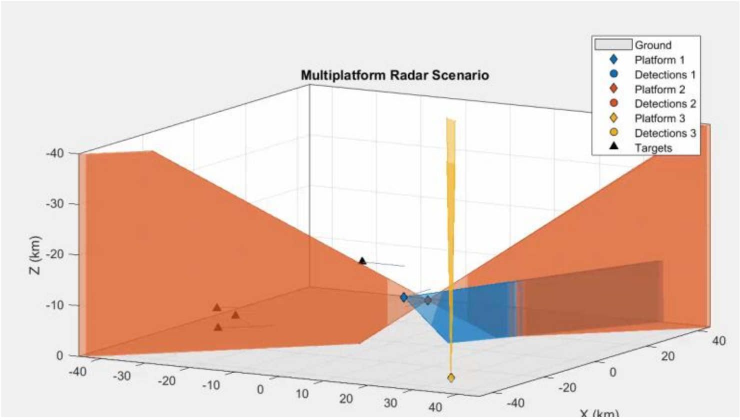
Live data can be augmented for a more robust testbench



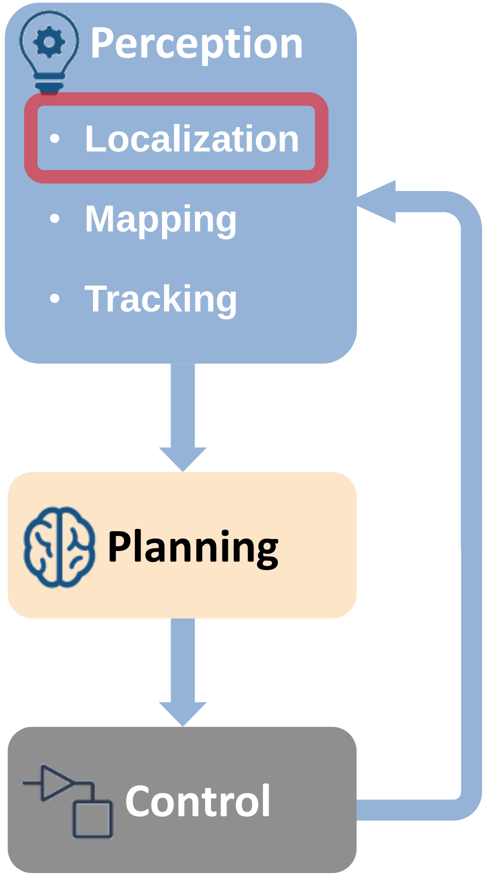
Define scenarios

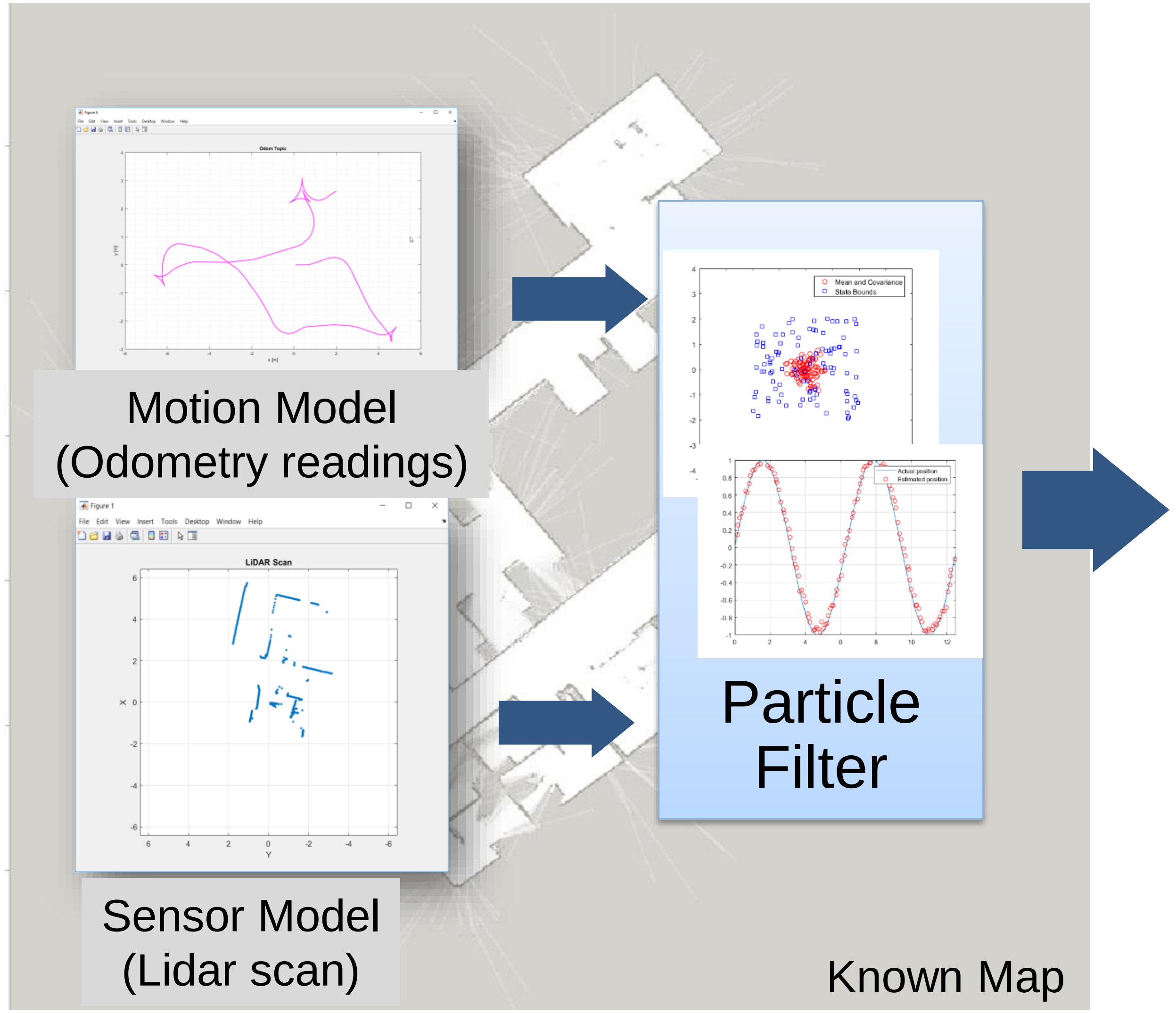


Simulate sensors



Estimate the pose using Monte Carlo Localization

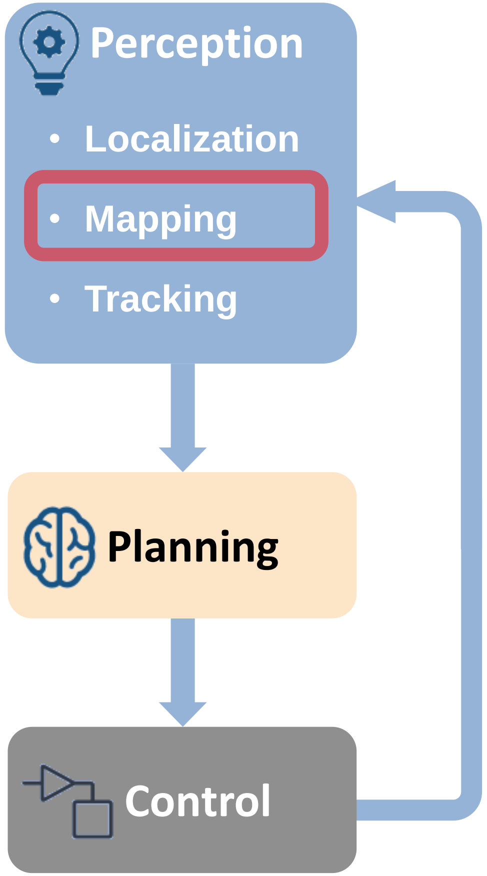




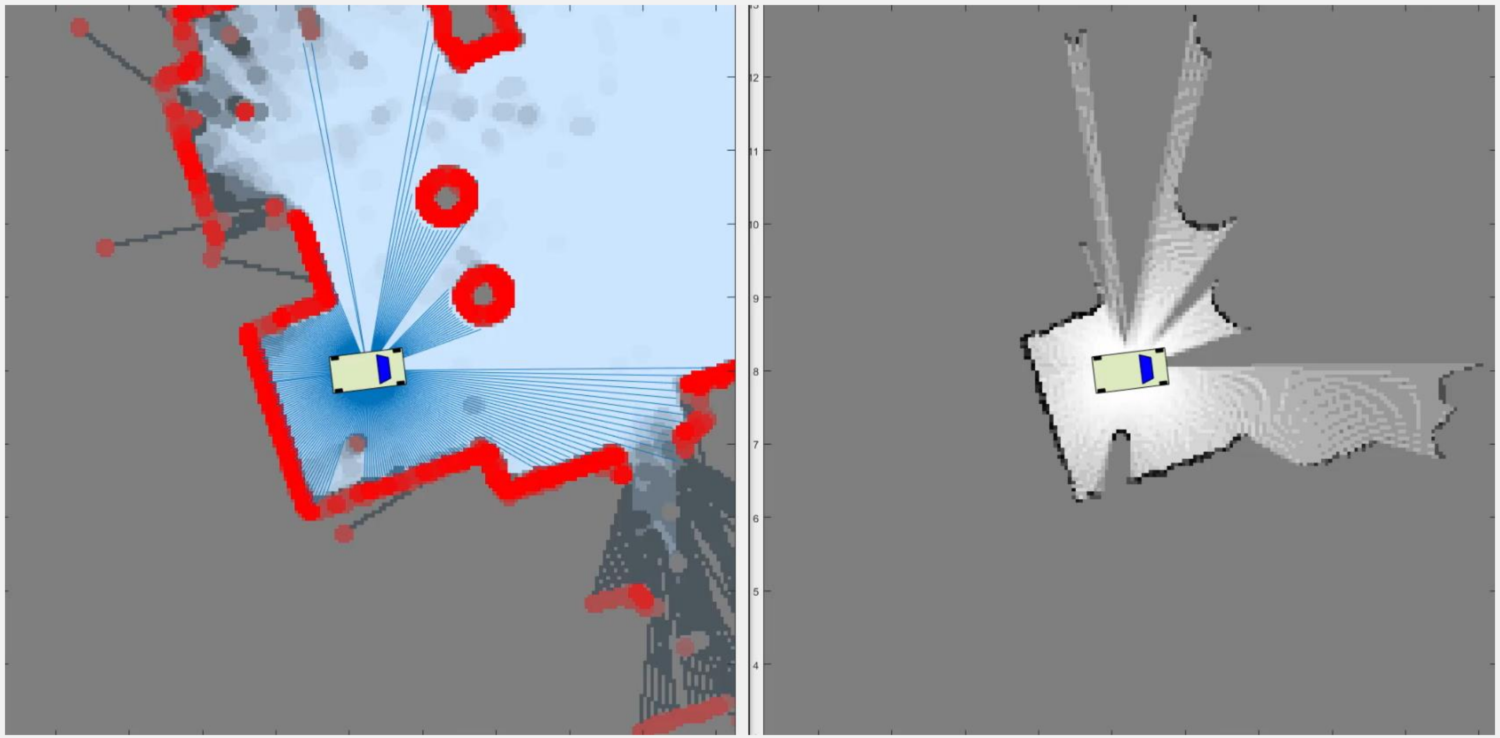
What is the world around me?

Egocentric occupancy maps

Dynamic Environment

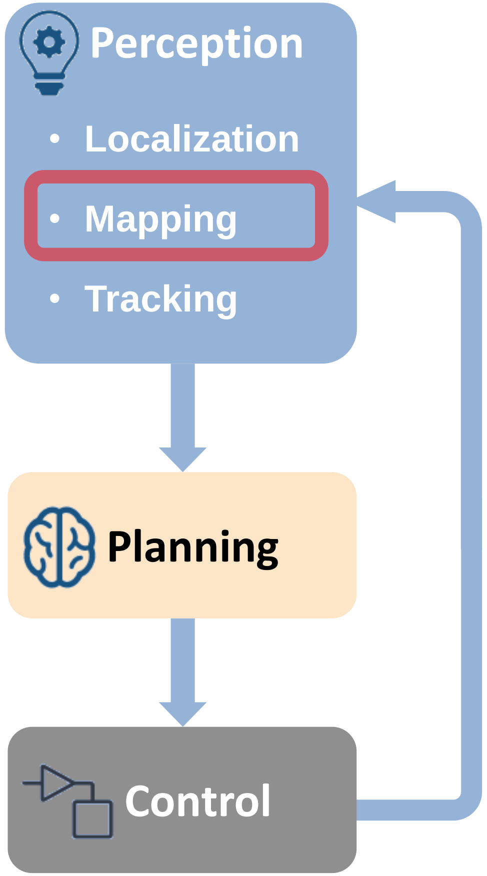


- Support dynamic environment changes
- Synchronization between global and local maps

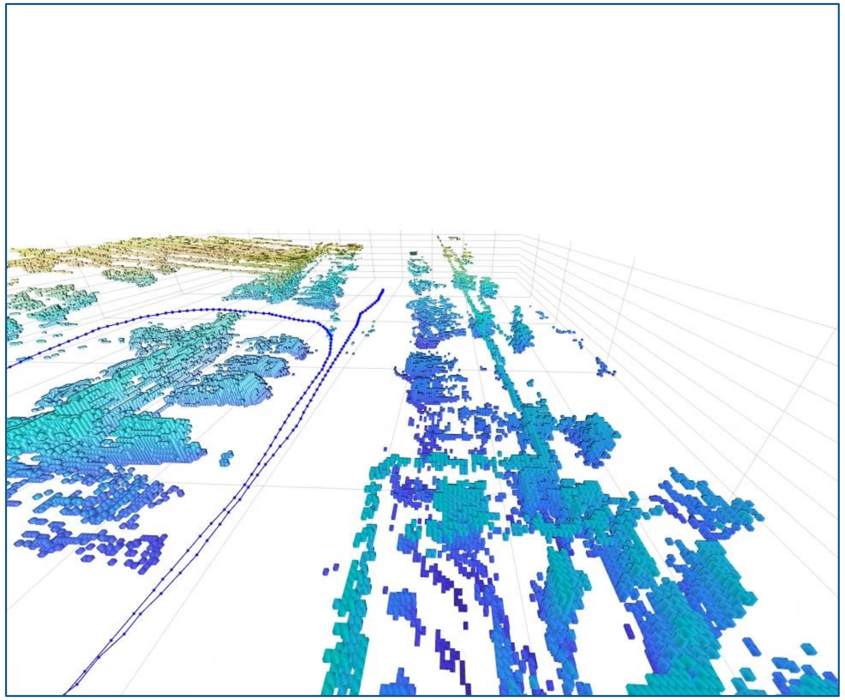


What is the world around me?

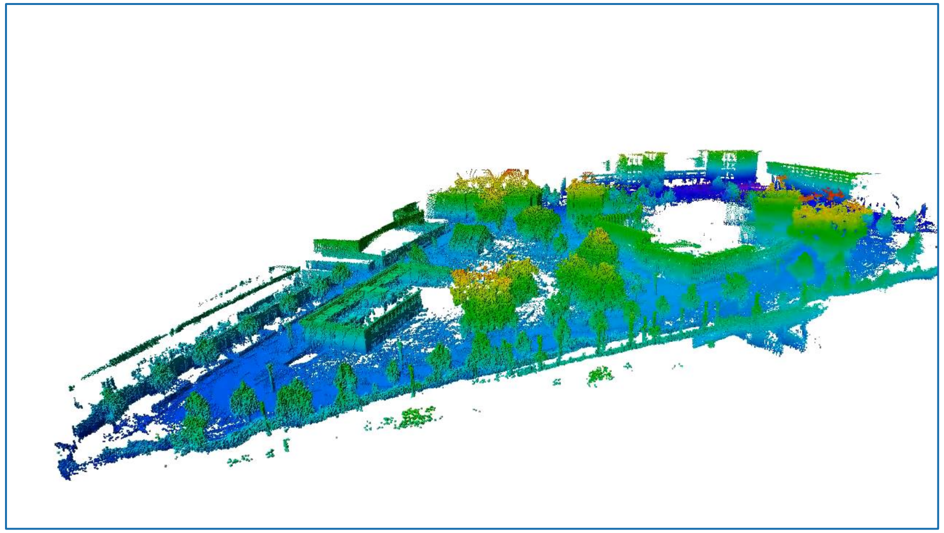
3D Occupancy Map



3D map for Autonomous Driving

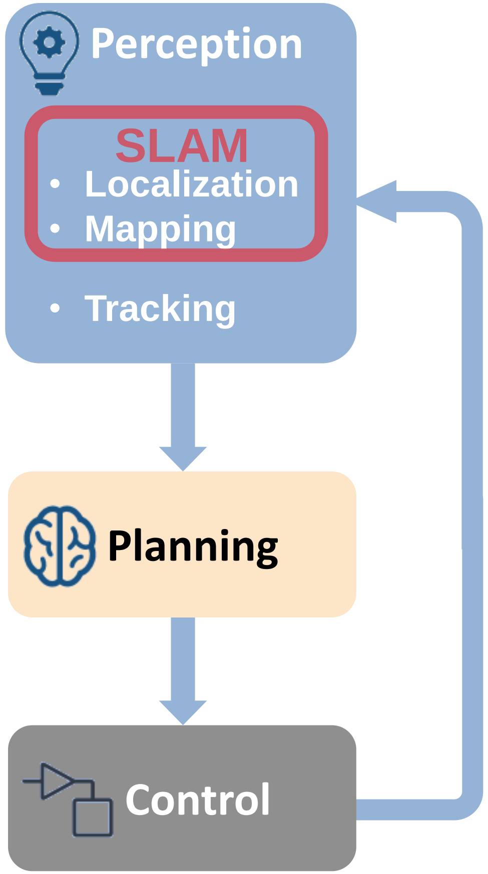


3D map for UAV motion planning

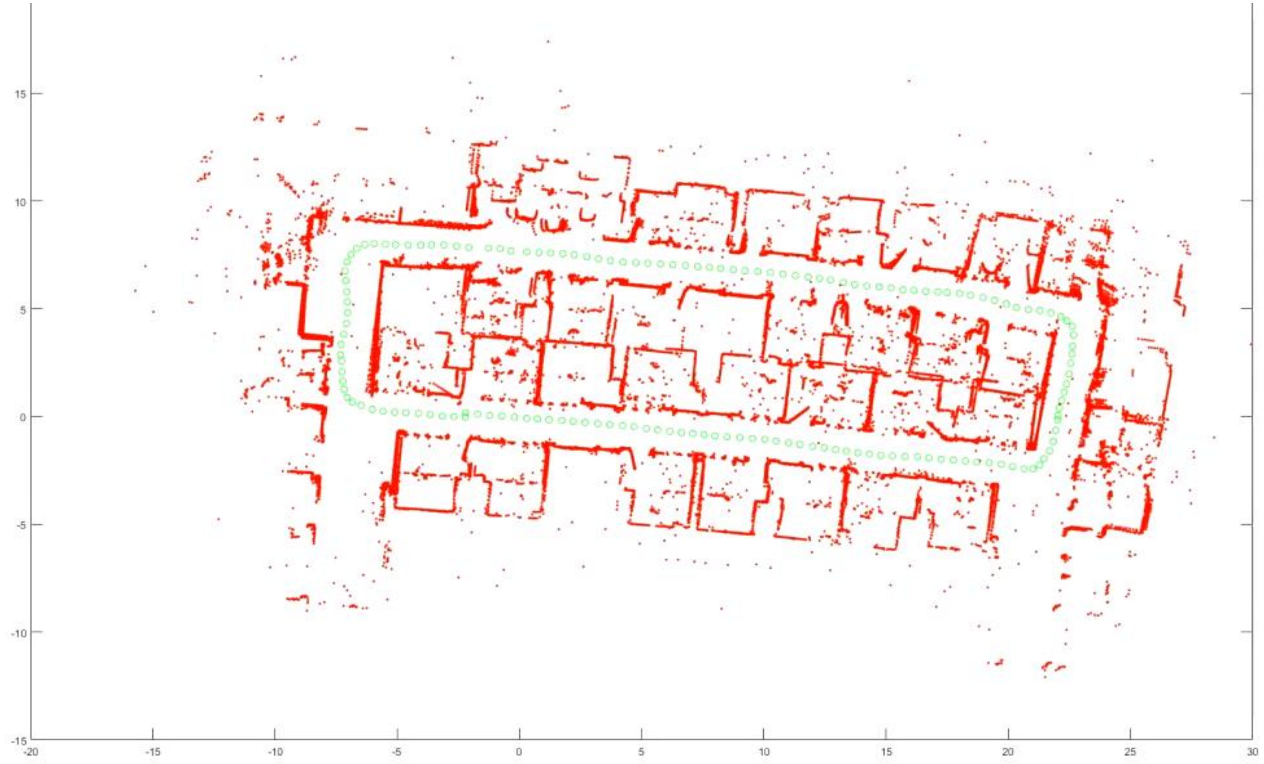


Where am I in the unknown environment?

Simultaneous Localization and Mapping (SLAM)



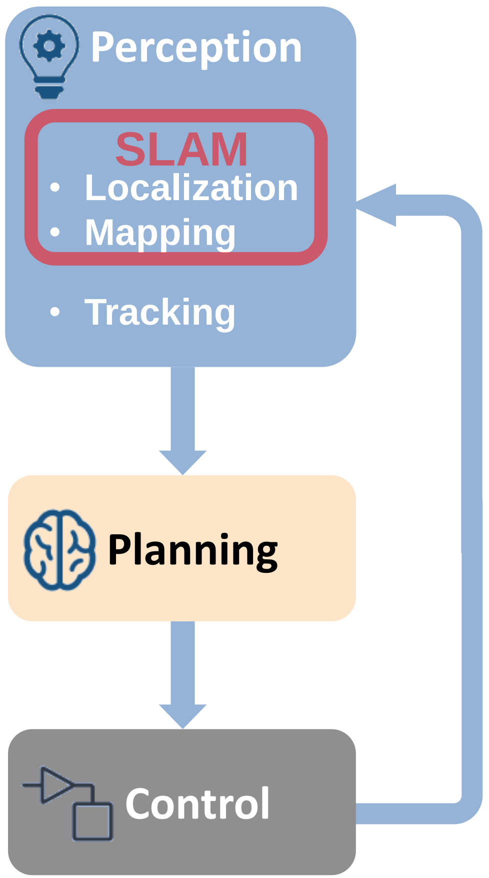
2D Lidar SLAM

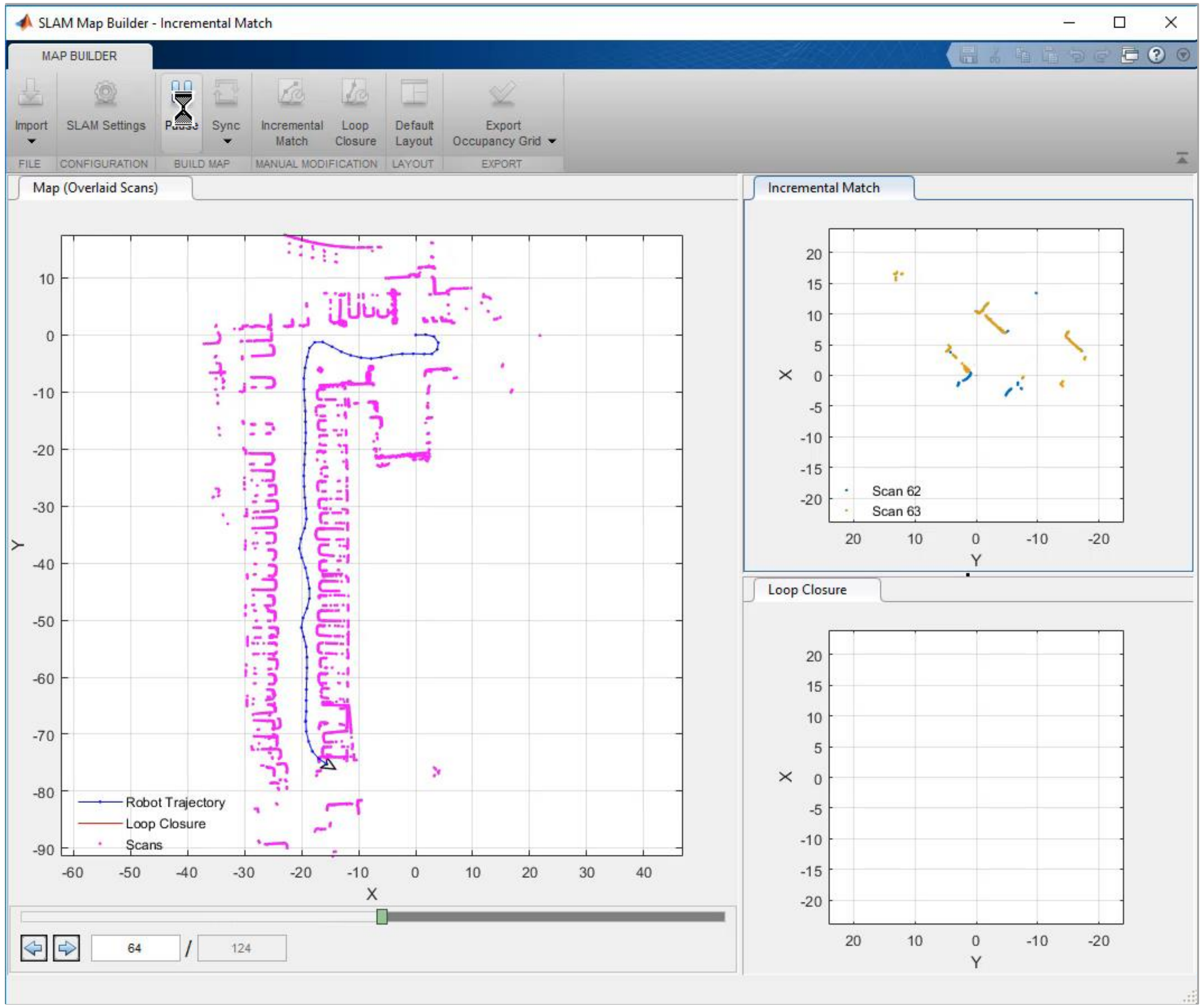


Build a map of an unknown environment while simultaneously keeping track of robot’s pose.

Simultaneous Localization and Mapping

SLAM Map Builder App (2D only)





App enables more interactive and user-friendly workflow

Simultaneous Localization and Mapping

3D Lidar SLAM

Perception

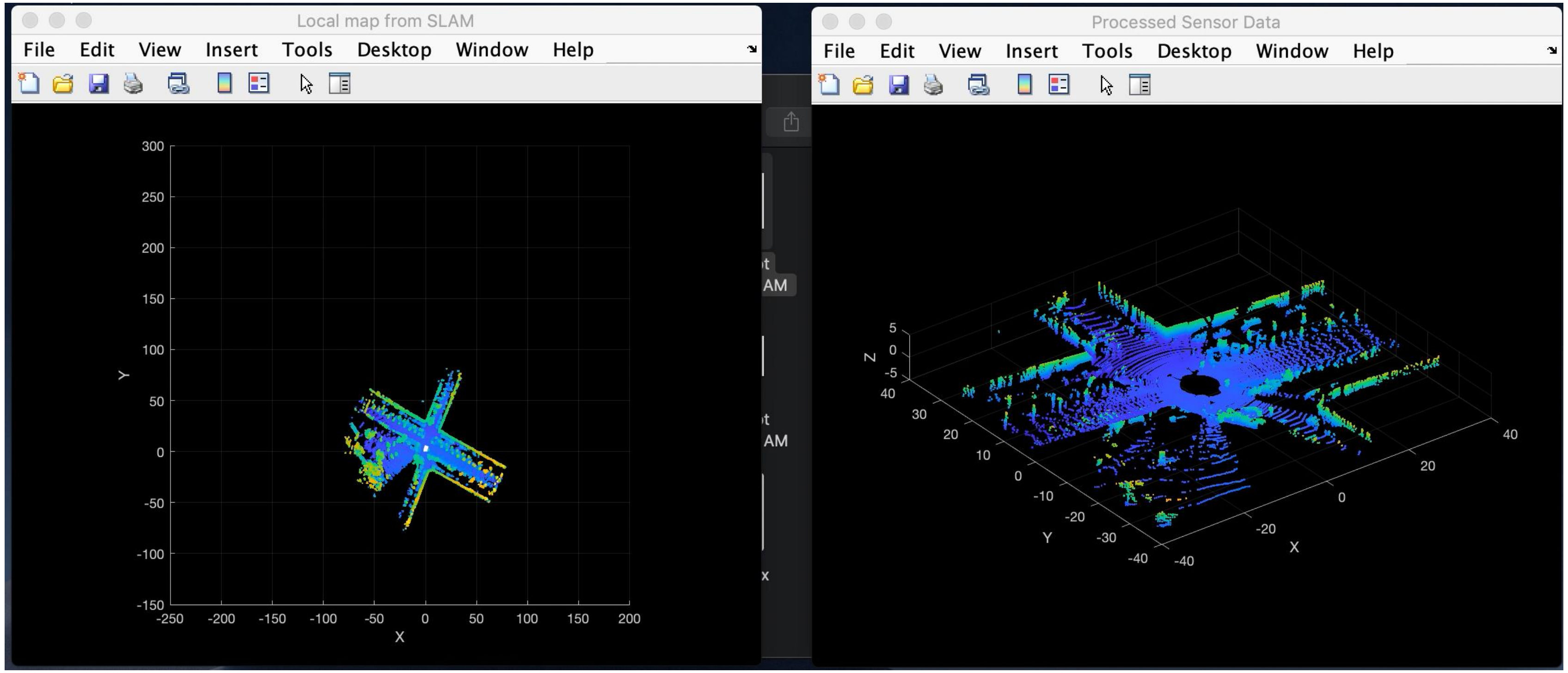
SLAM
- **Localization**
- **Mapping**
- **Tracking**

Planning

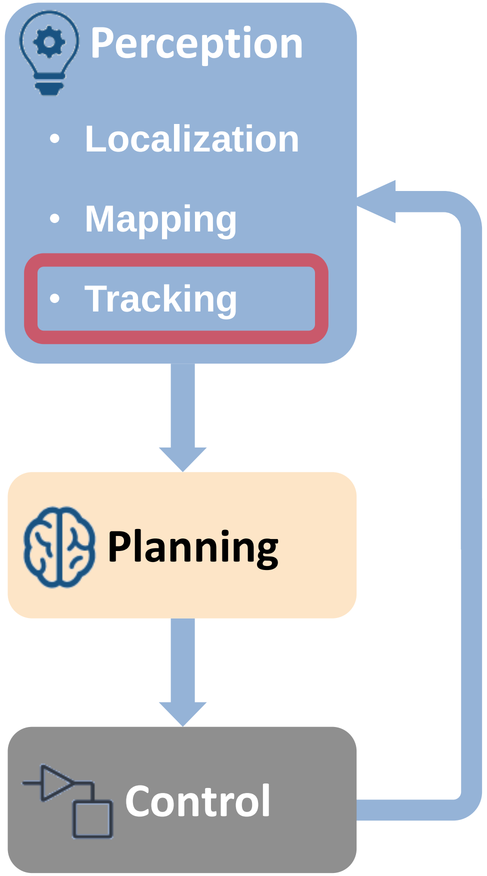
Control

Point Cloud Processing → Local Mapping → Loop Closure Detection → Pose Graph Optimization → Map representation

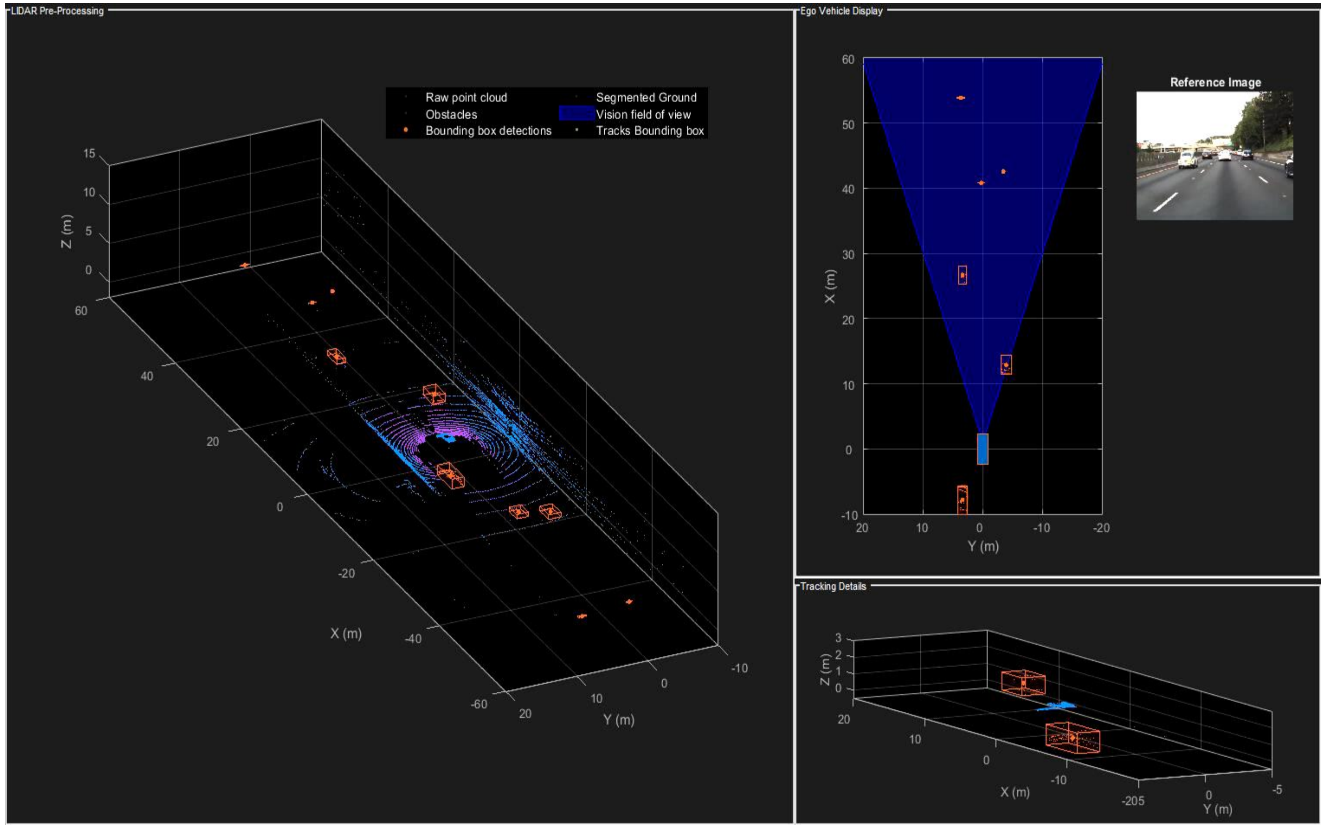
Computer Vision	Navigation



Autonomous systems can track objects from Lidar point clouds

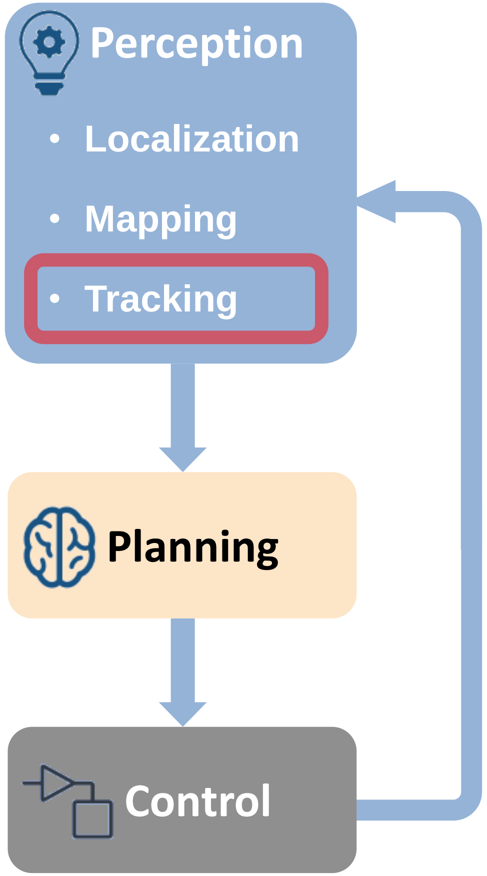


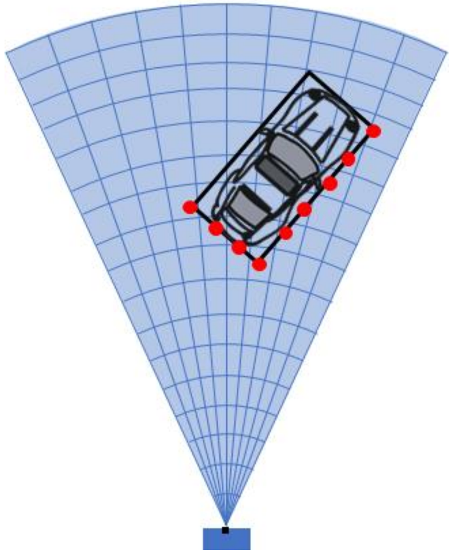
Track Objects Using Lidar: From Point Cloud to Track List

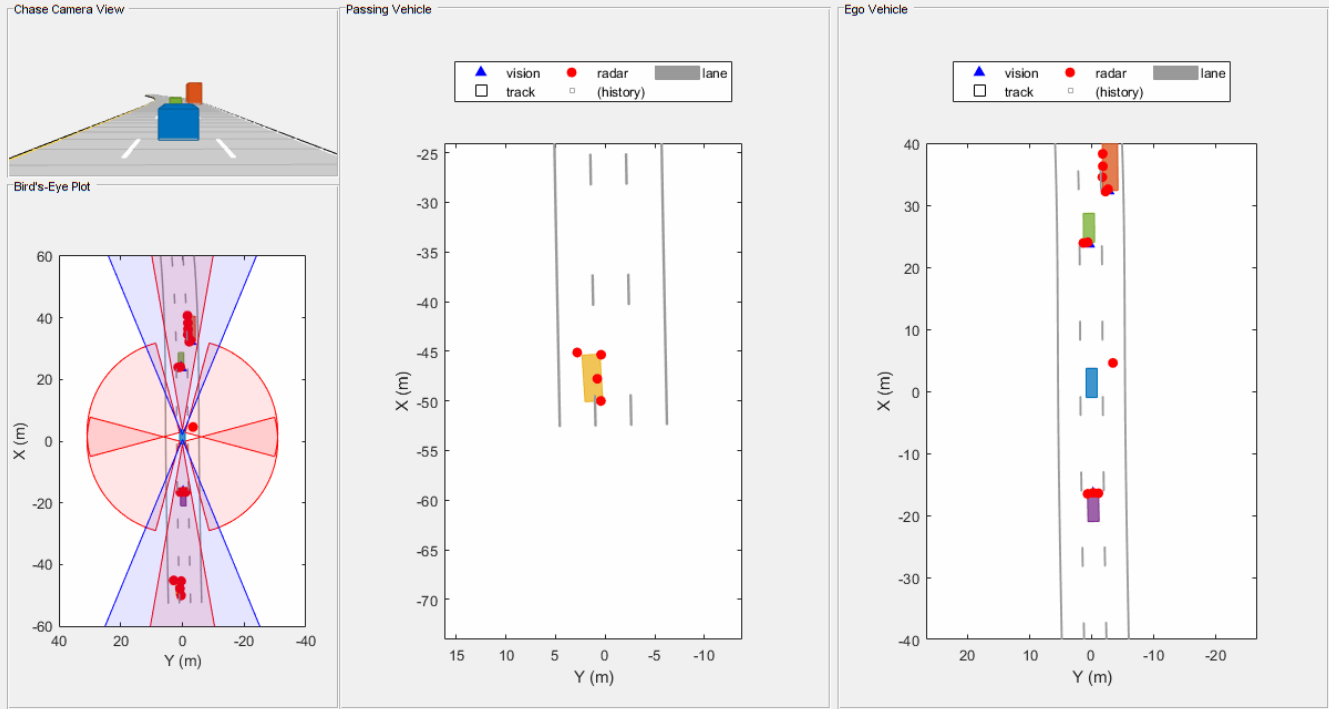


Track surrounding objects during automated lane change

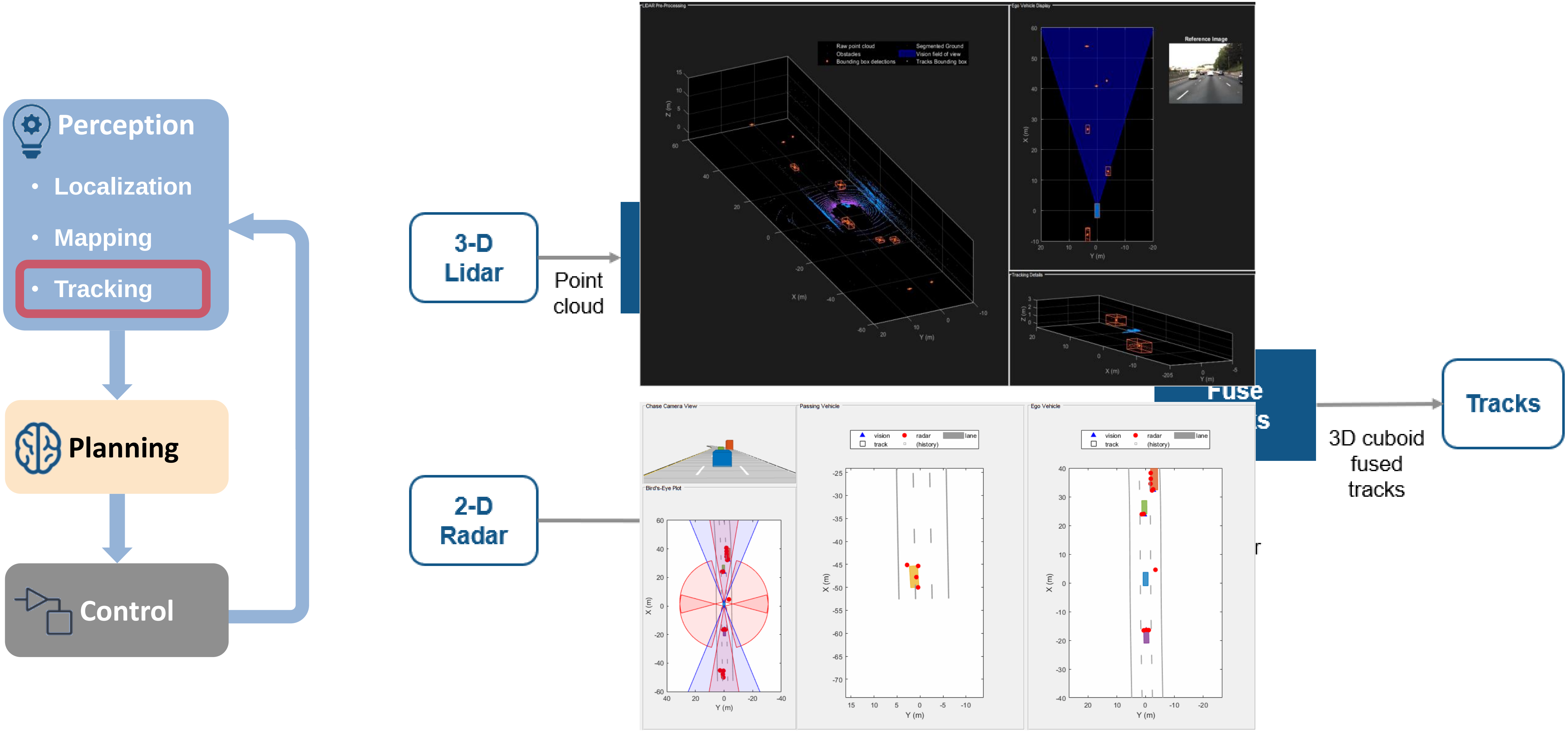
2D radar can be used to track position, size, and orientation



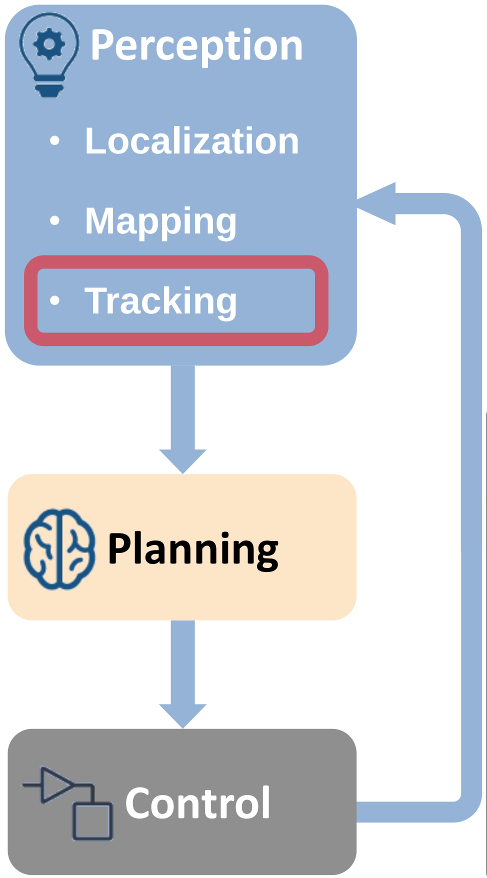


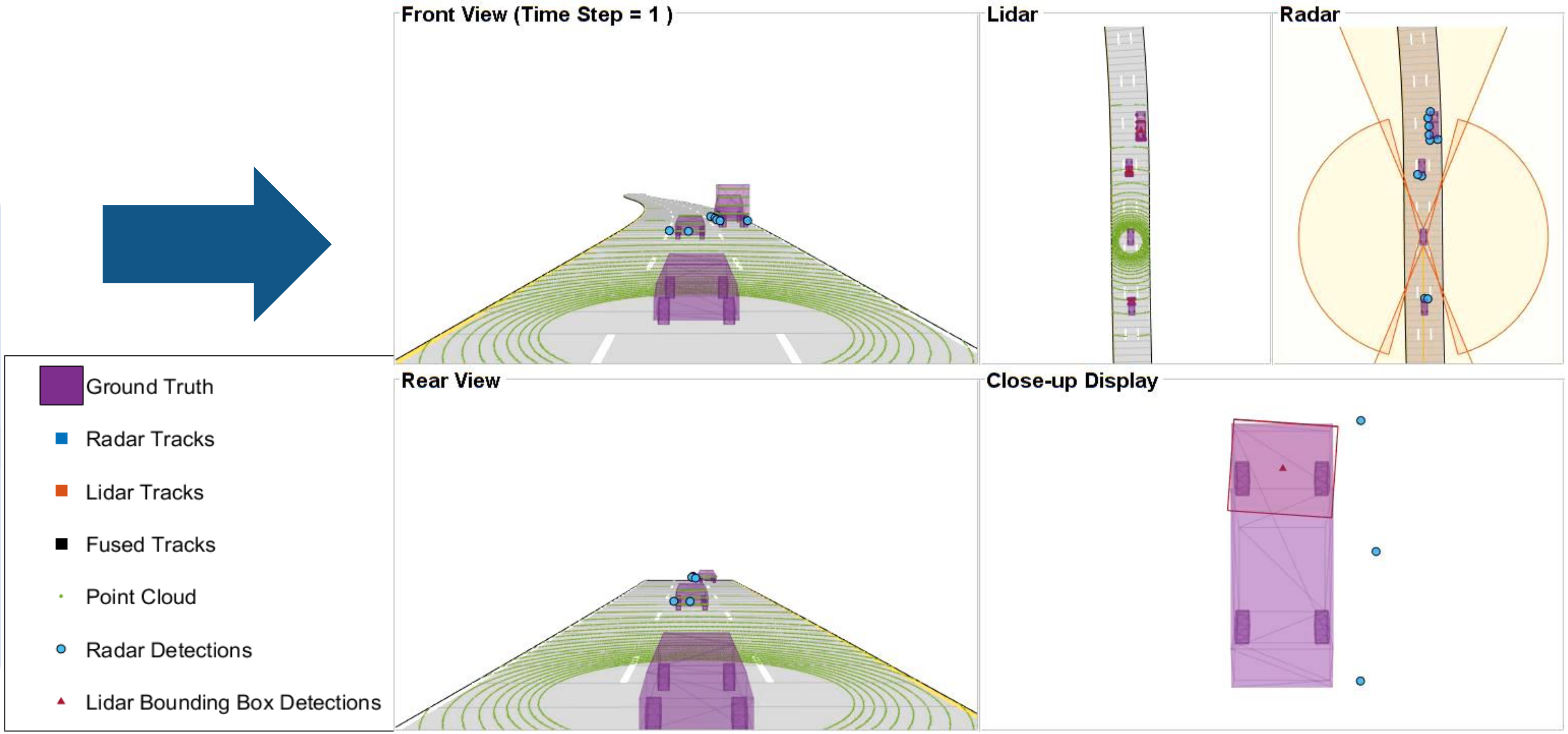


Fusing multiple sensor modalities provides a better result



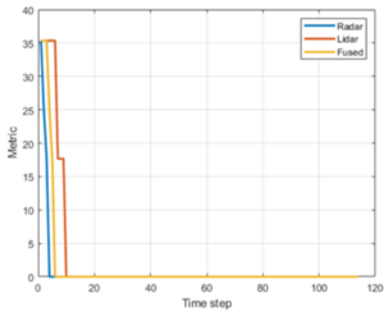
Radar and Lidar fusion can increase tracking performance



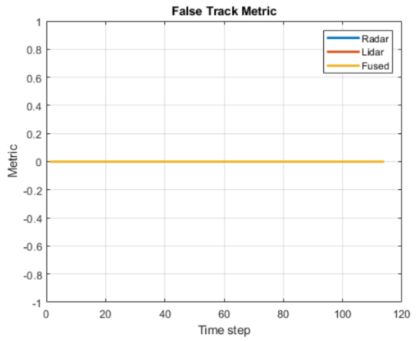


Fuse lidar point cloud with radar detections

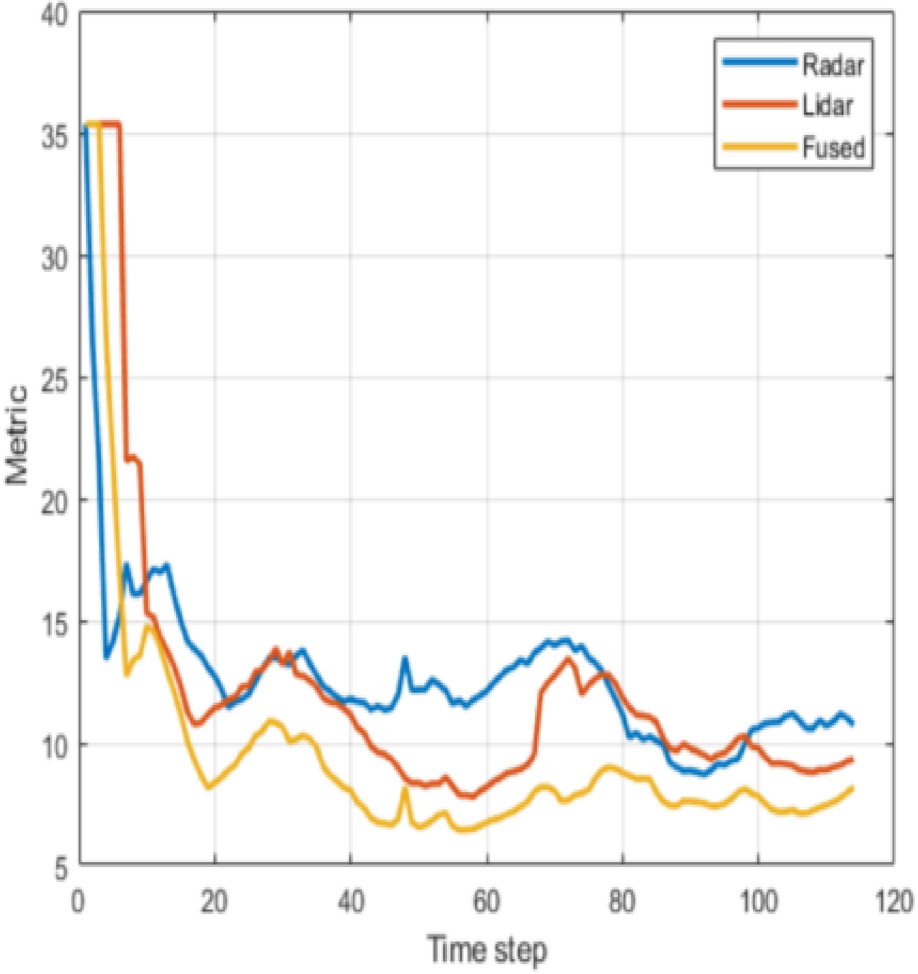
Missed Targets



False Tracks

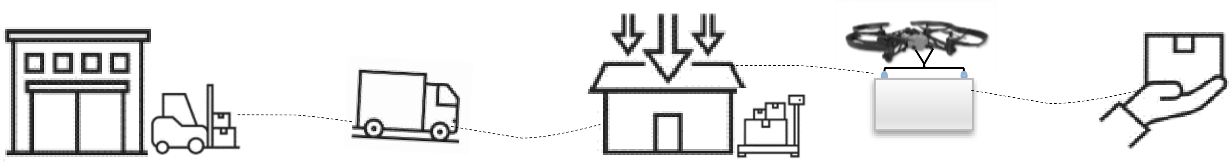


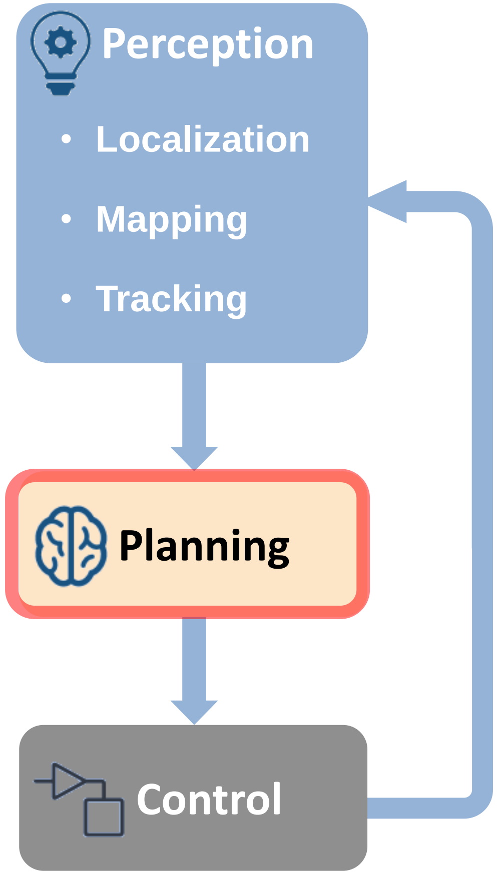
GOSPA



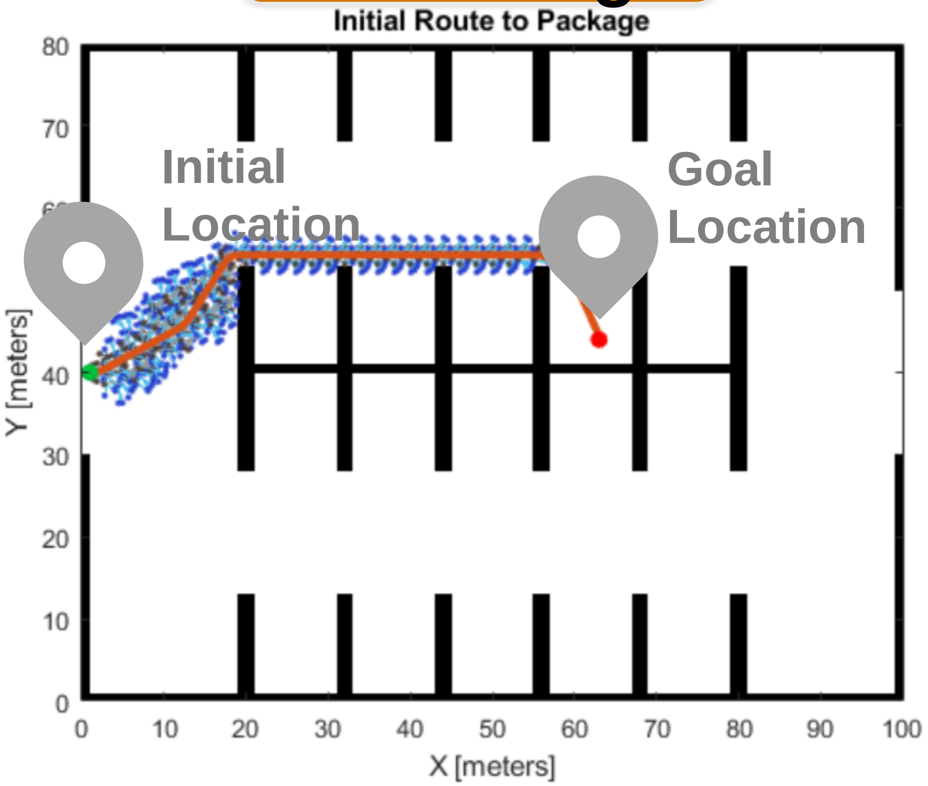
- Assess missed tracks
- Assess false tracks
- Assess Generalized Optimal Sub Pattern Assignment Metric (GOSPA)

Plan a path from start to destination

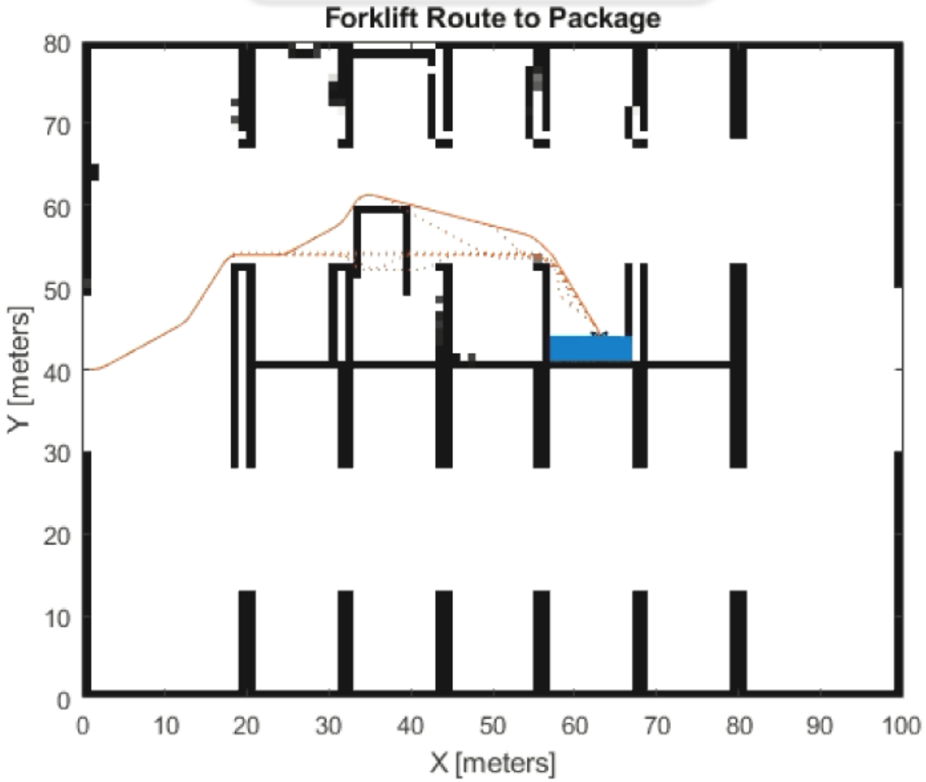




Global Planning



Local Re-planning



Plan a path from start to destination

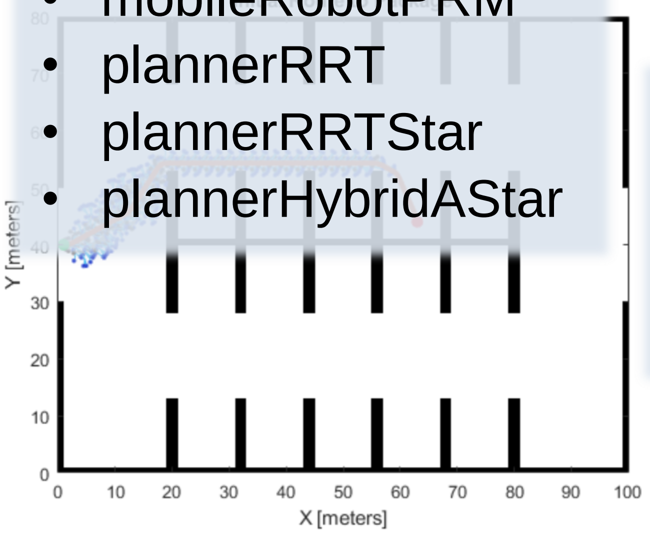
Perception
- Localization
- Mapping
- Tracking

Planning

Control

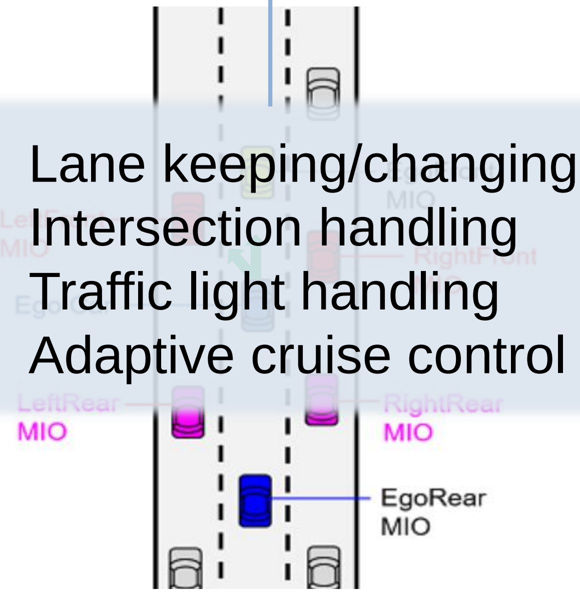
Global Planning
- Path planning algorithms

- mobileRobotPRM
- plannerRRT
- plannerRRTStar
- plannerHybridAStar



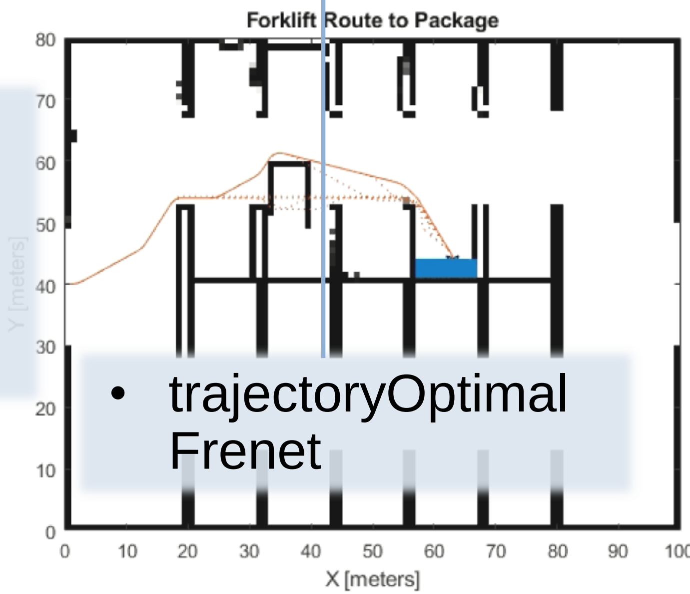
Behavior Planning
- High-level decision making

- Lane keeping/changing
- Intersection handling
- Traffic light handling
- Adaptive cruise control



Local Re-planning
- Trajectory generation

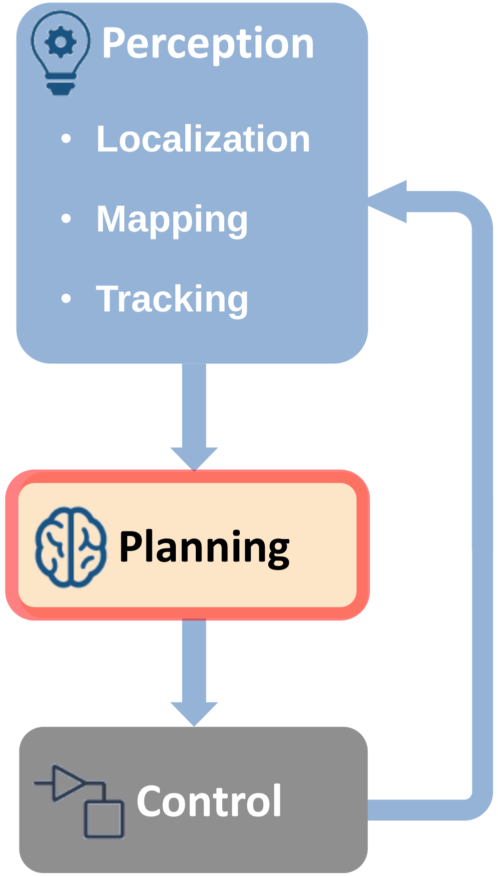
- trajectoryOptimalFrenet

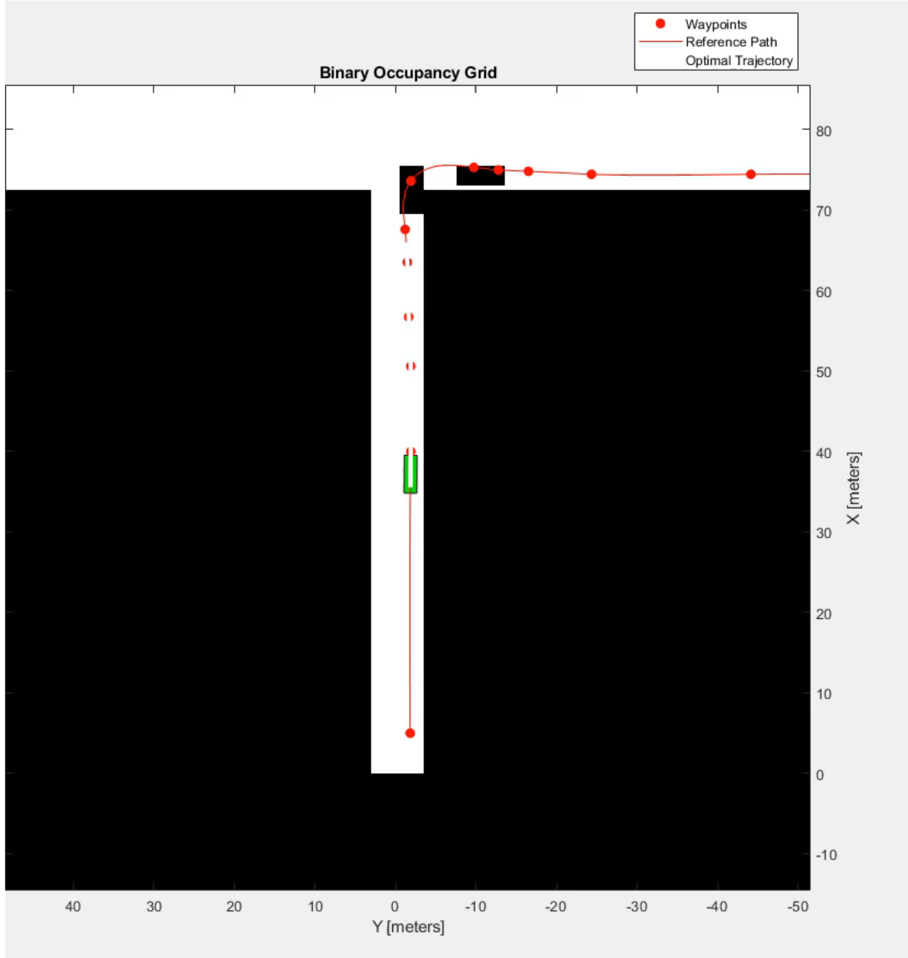


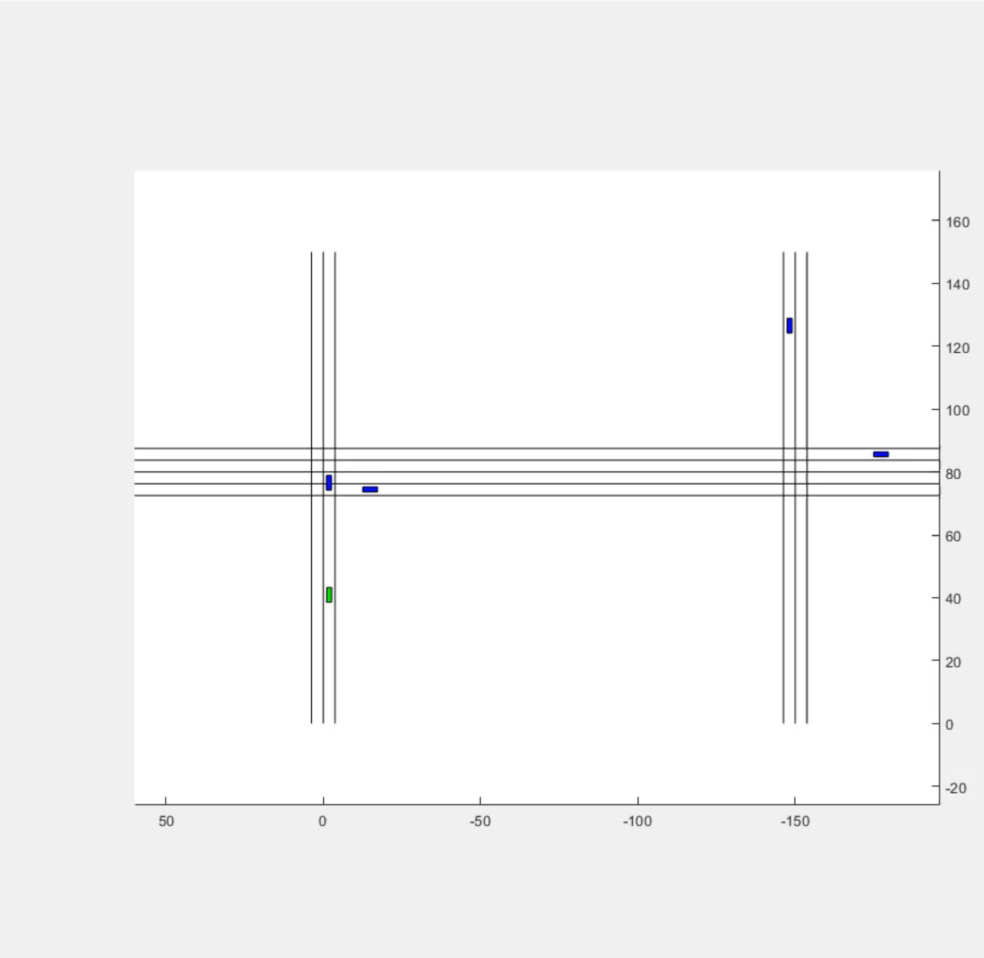
Urban driving needs planning on multiple levels

Global, behavior, and local planners

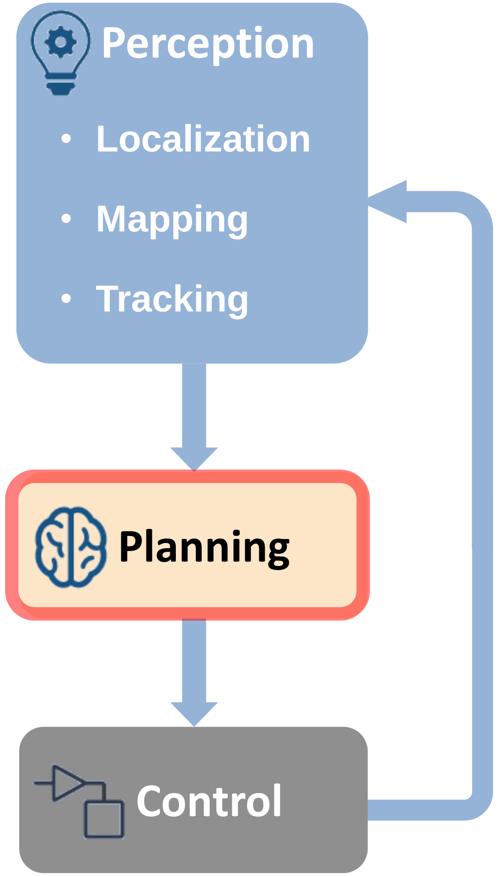
Generate optimal trajectories for local re-planning and merge back with the global plan



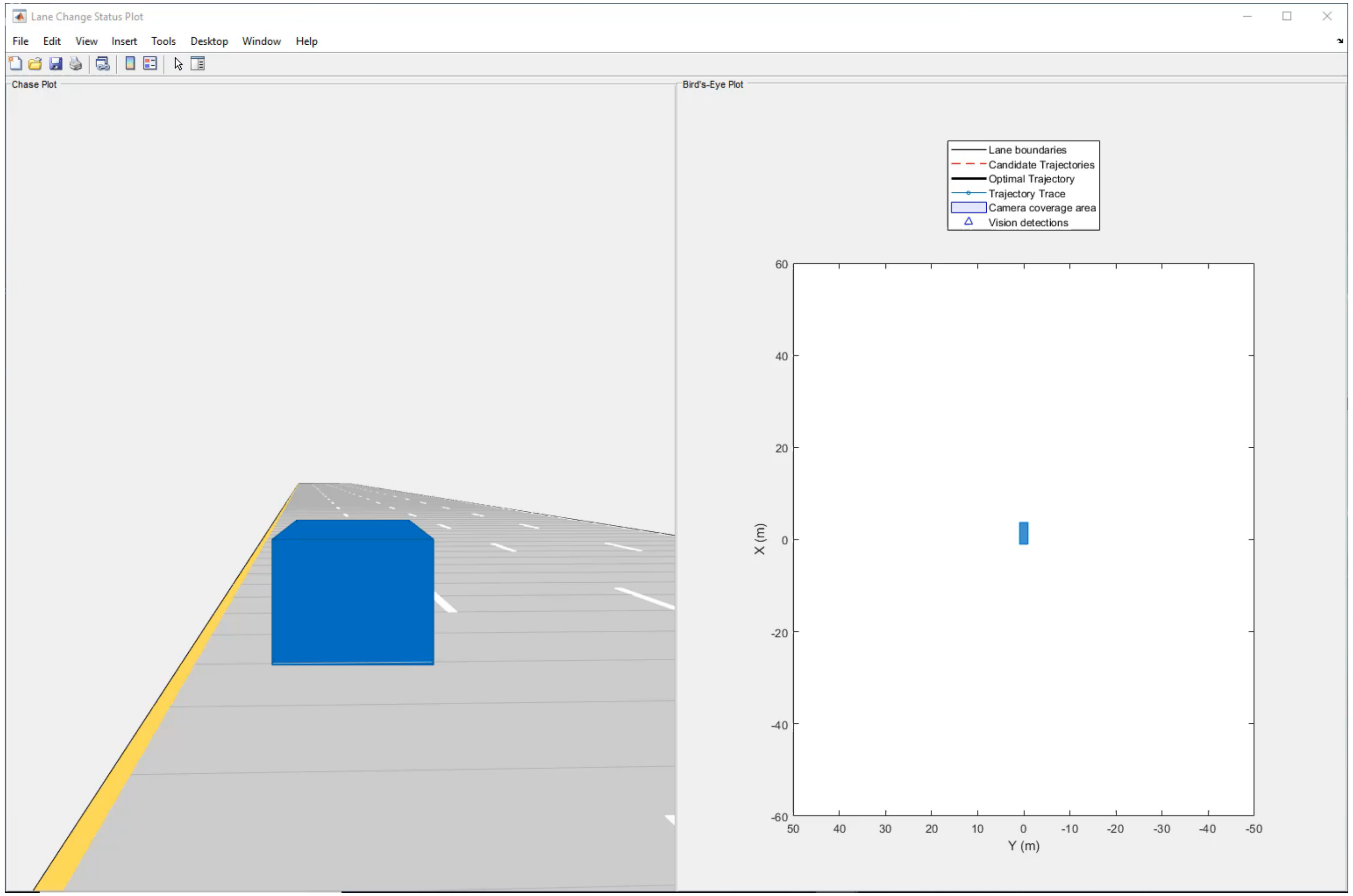




Simulate shortest path to change lanes on a highway



Simulate trajectory generation and the lane change maneuver

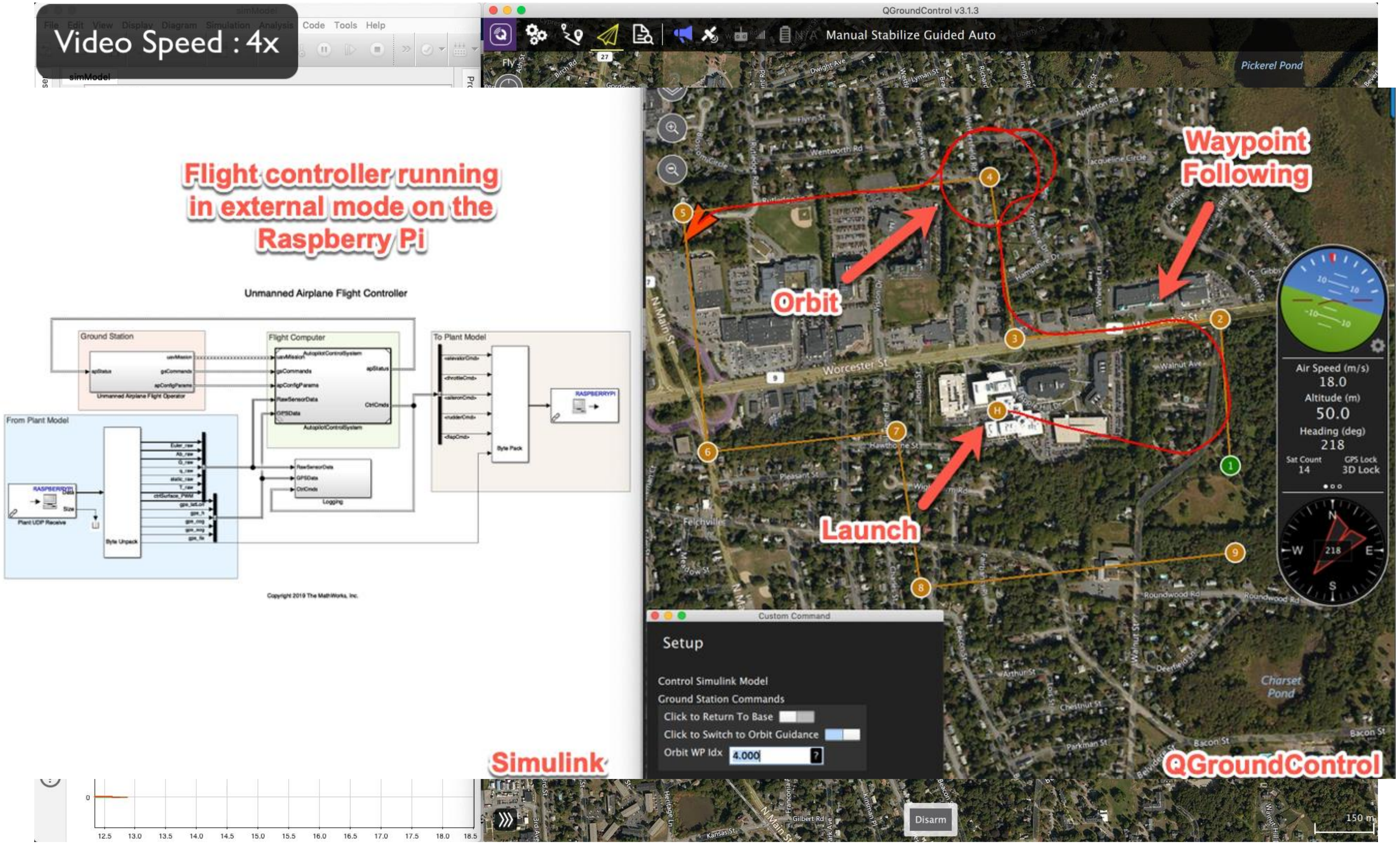


Mission planning for UAV leads to last mile delivery

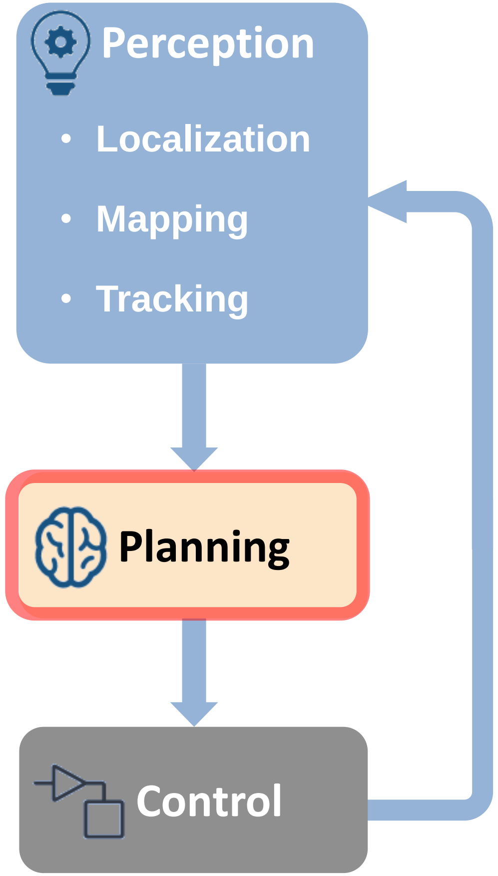
Perception
- Localization
- Mapping
- Tracking

Planning

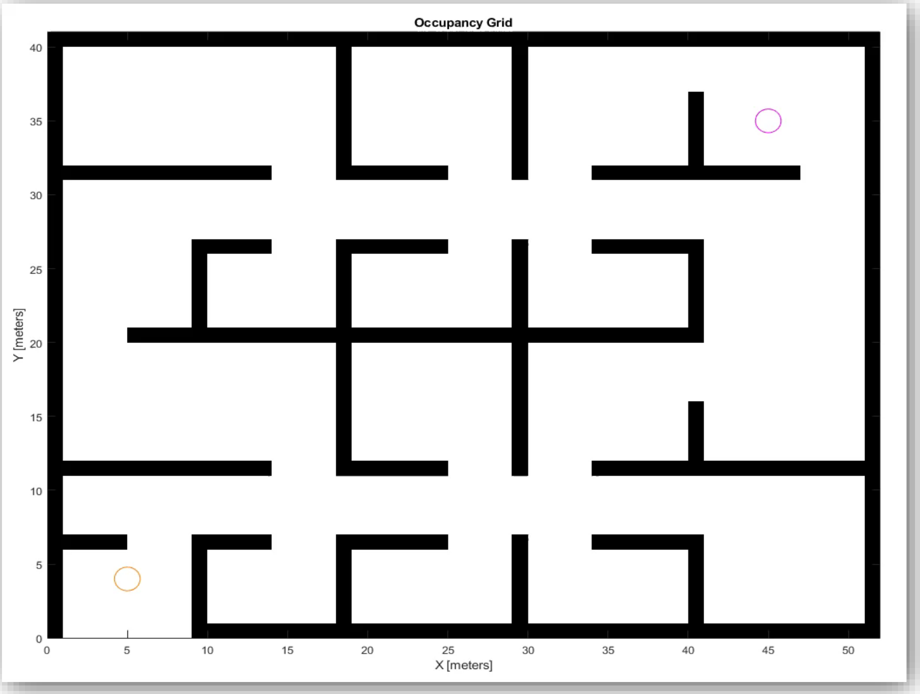
Control



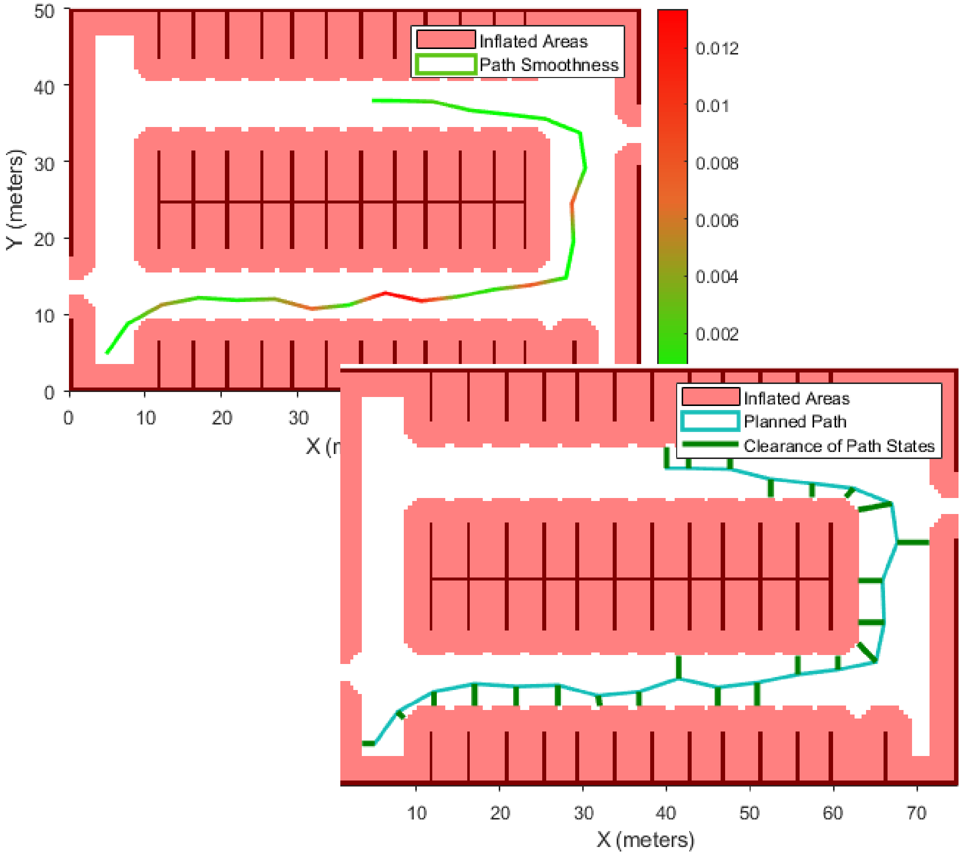
Choose a path planner based on your application



Sampling-based planners such as RRT*

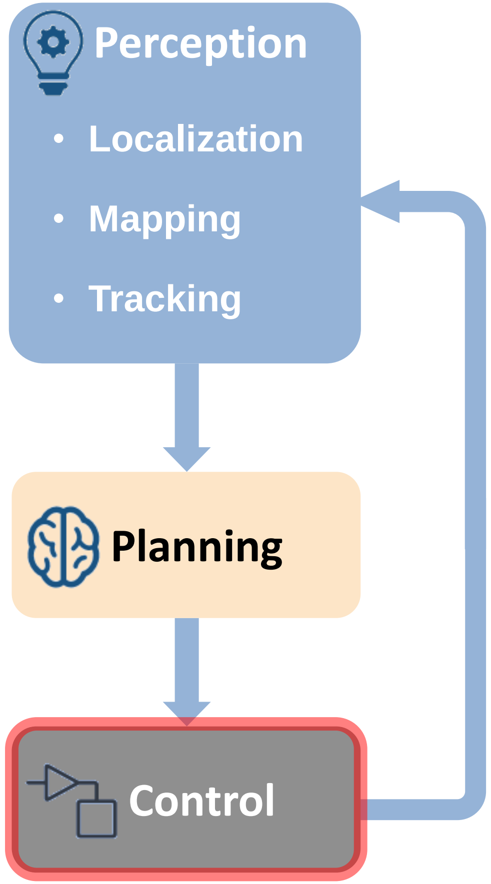


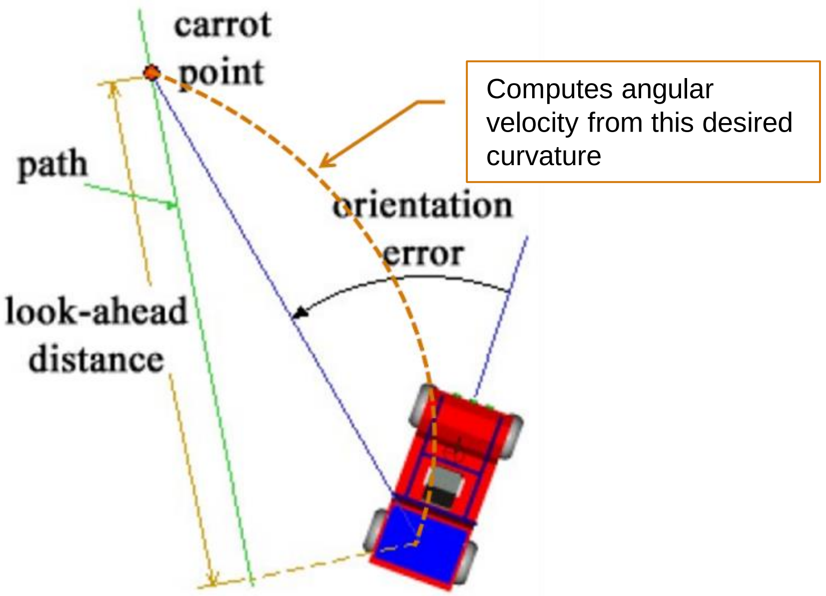
Use path metrics to compare different paths

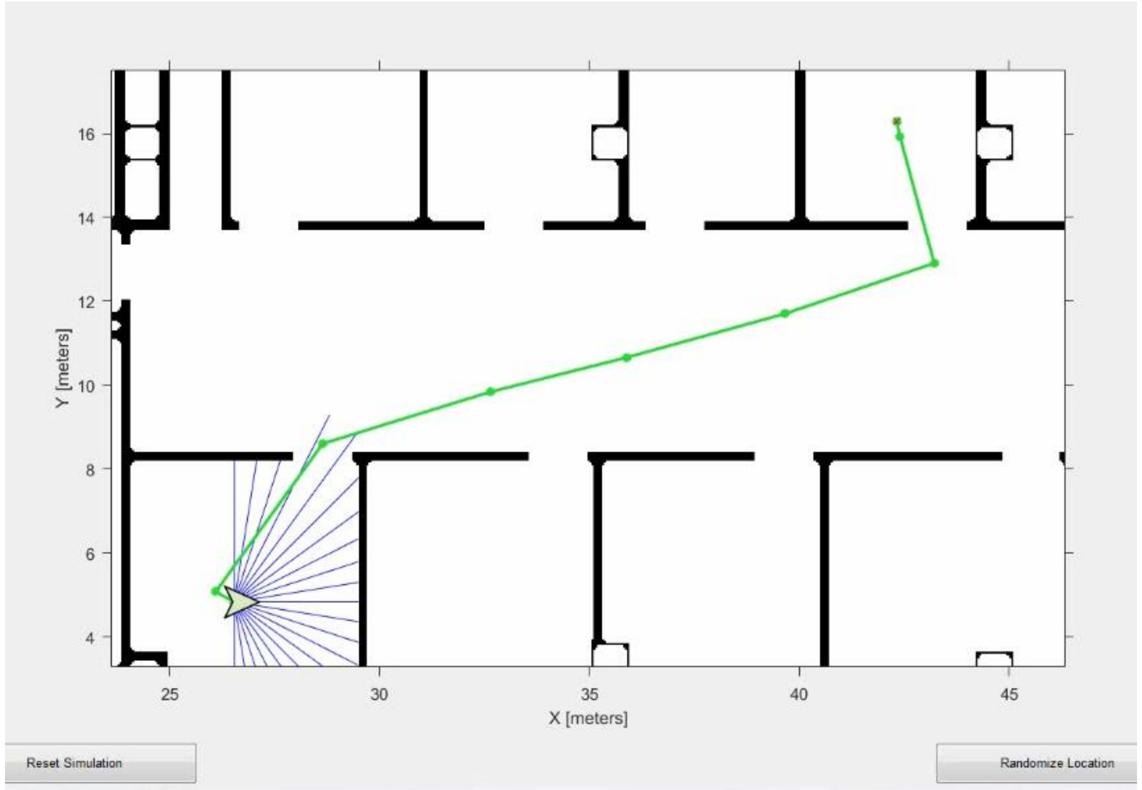


Compute control commands for ground vehicles

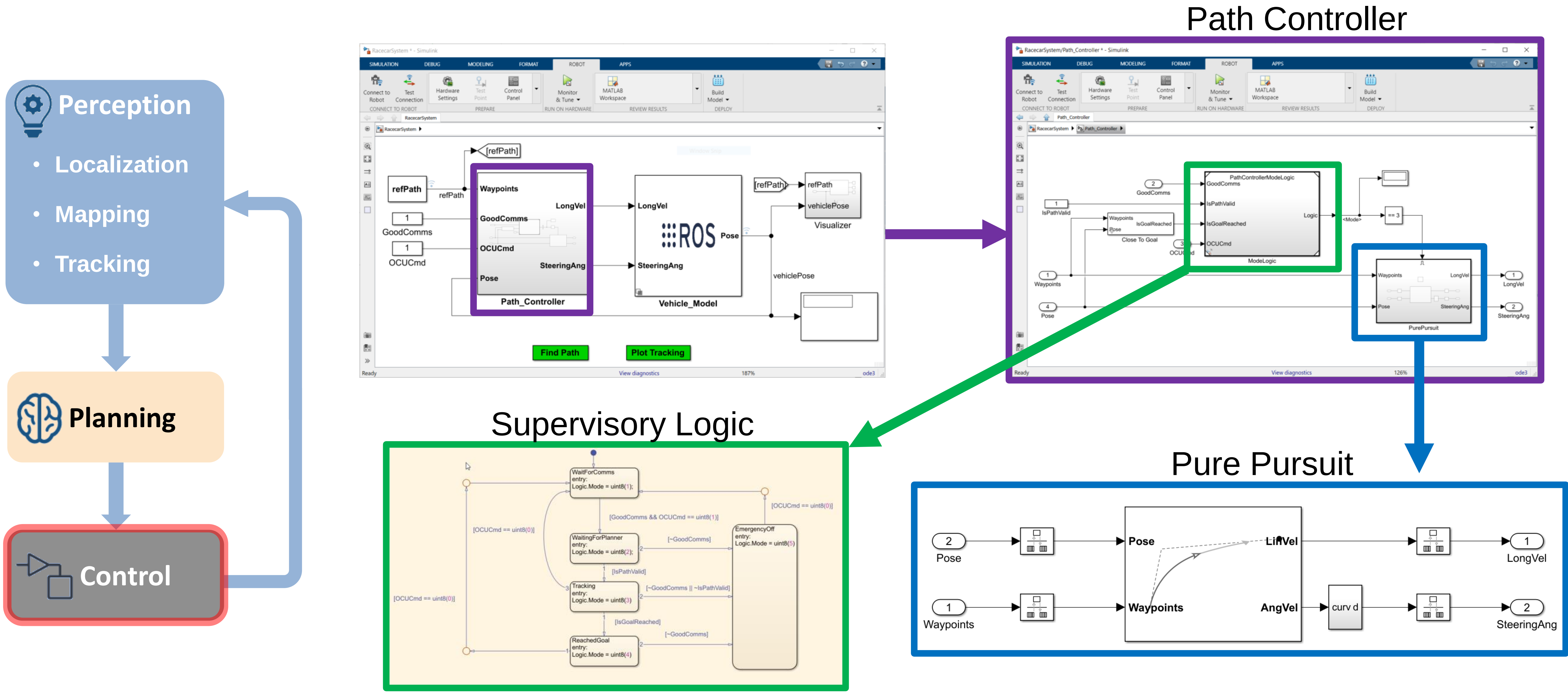
Compute linear and angular velocity commands for a mobile robot



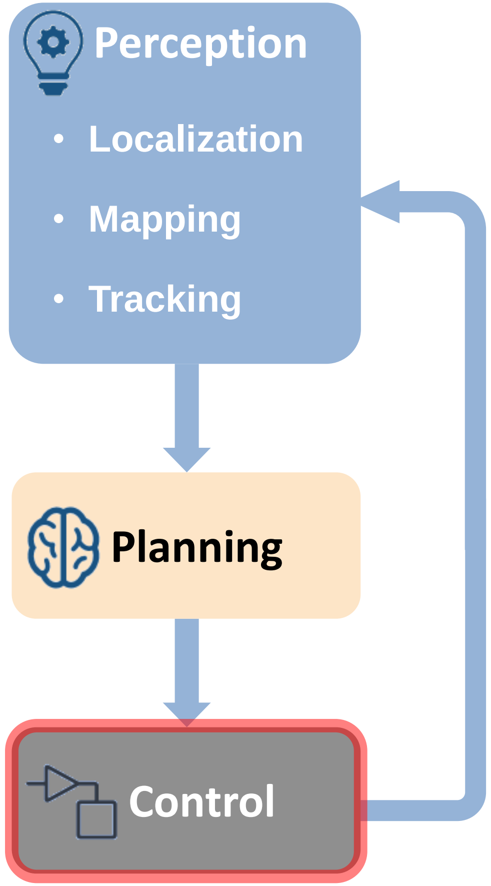




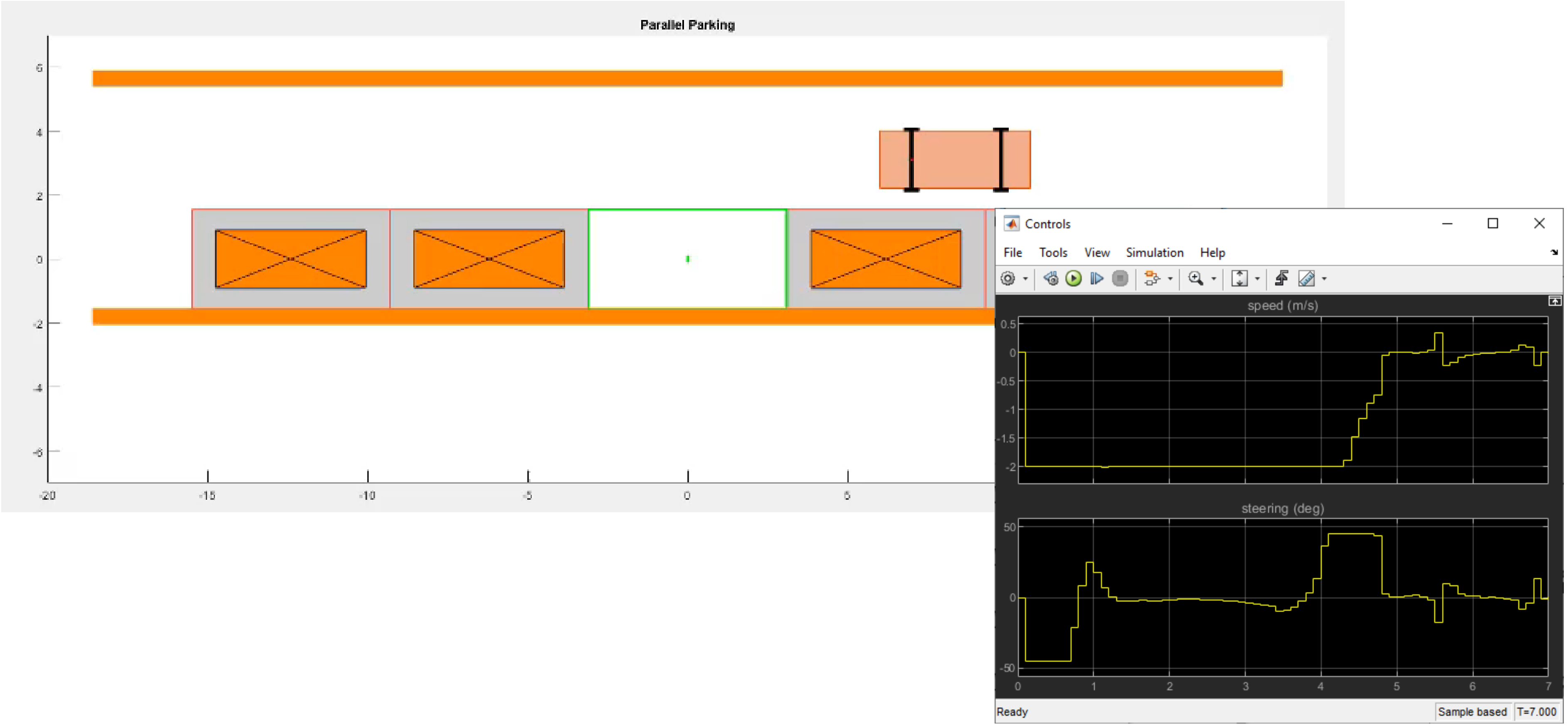
Use Pure Pursuit controller with supervisory logic



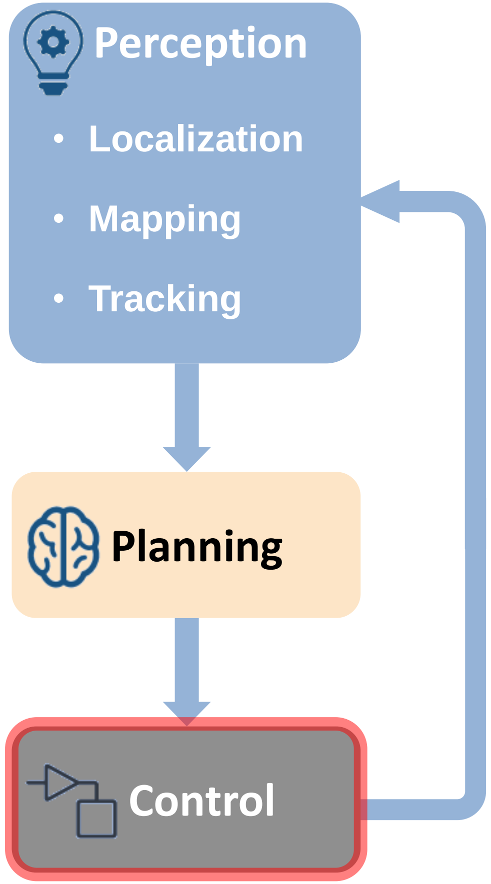
Send control commands to the vehicle to follow the planned path



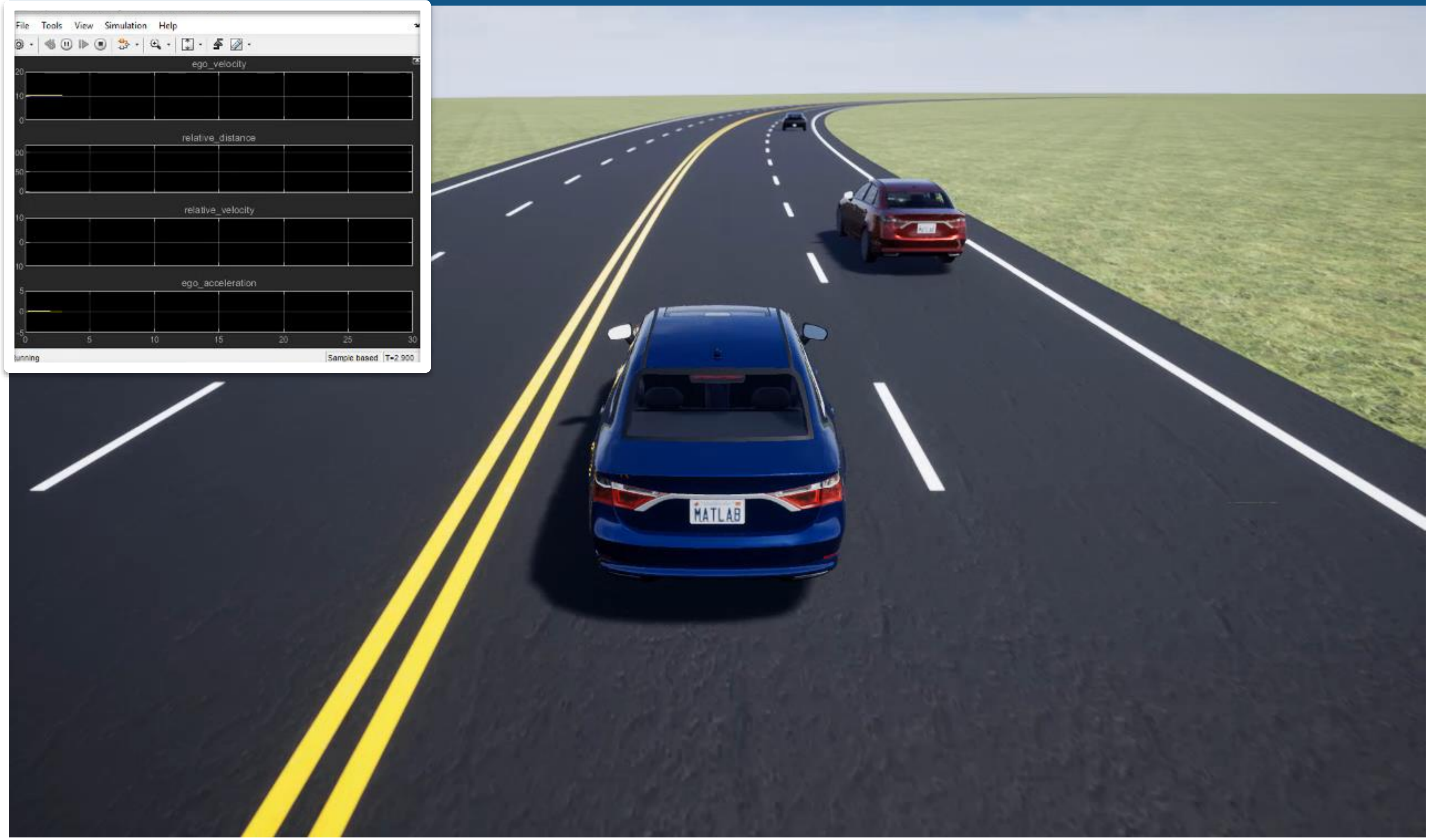
Calculate the steering angle and vehicle velocities to track the trajectories



Control lane change maneuver for highway driving

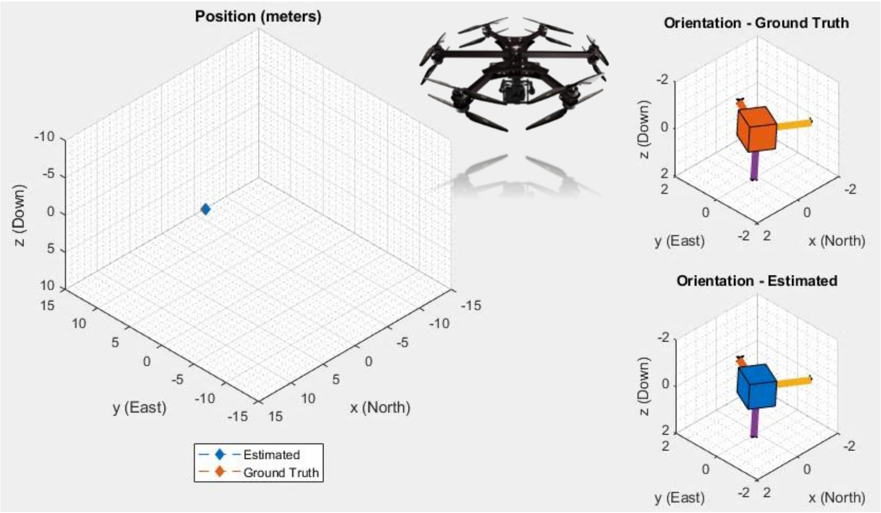


Longitudinal and Lateral Controllers to adjust the acceleration and steering

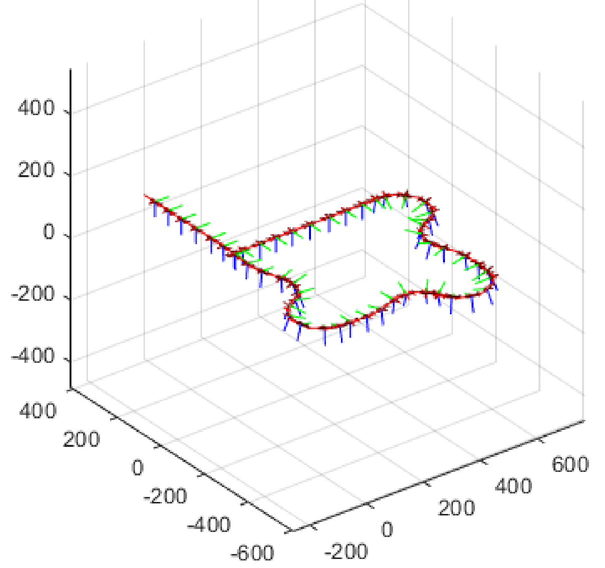


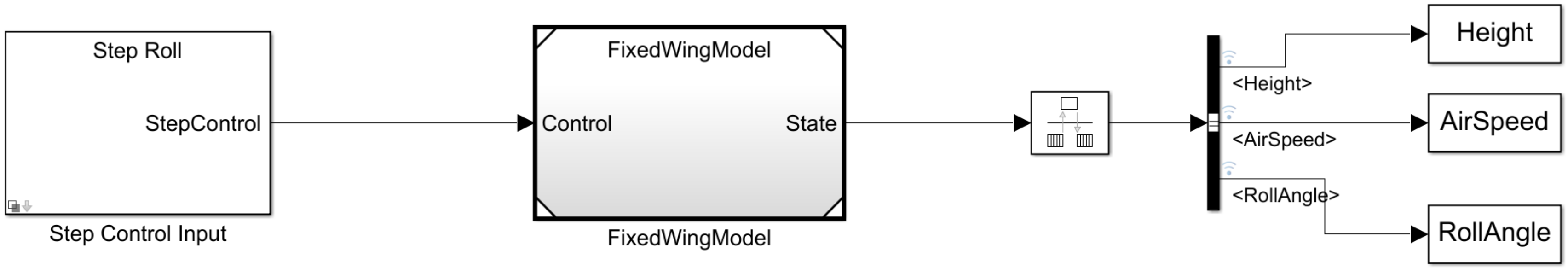
Simulate high-fidelity UAV model with waypoint following

Simulate GPS and IMU sensor models

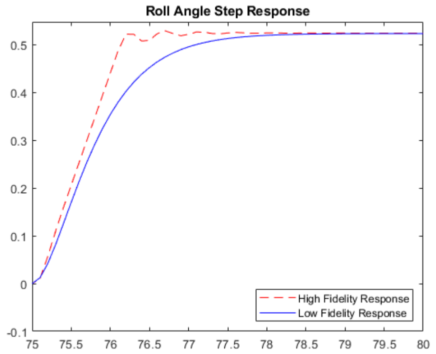


Waypoint following controller

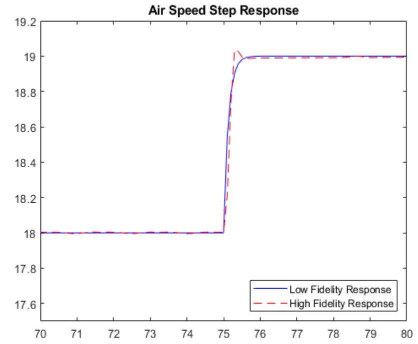




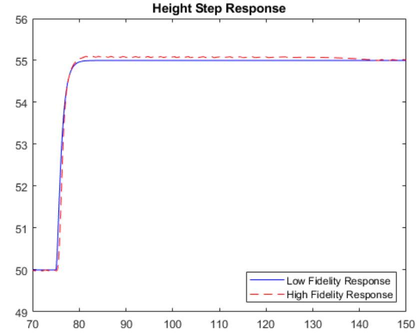
Approximate High-Fidelity Model with Low-Fidelity Model



Roll Angle

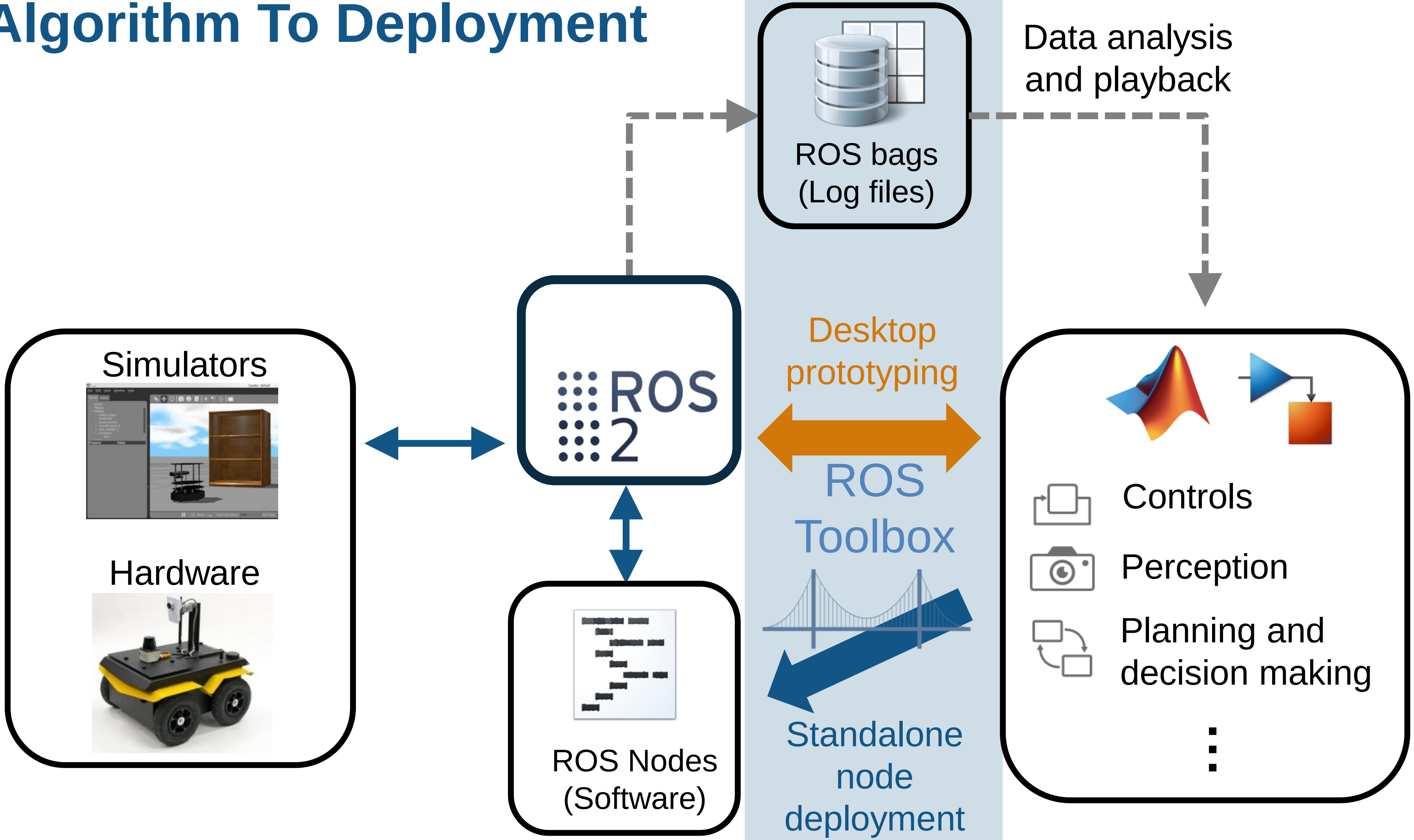


Air Speed

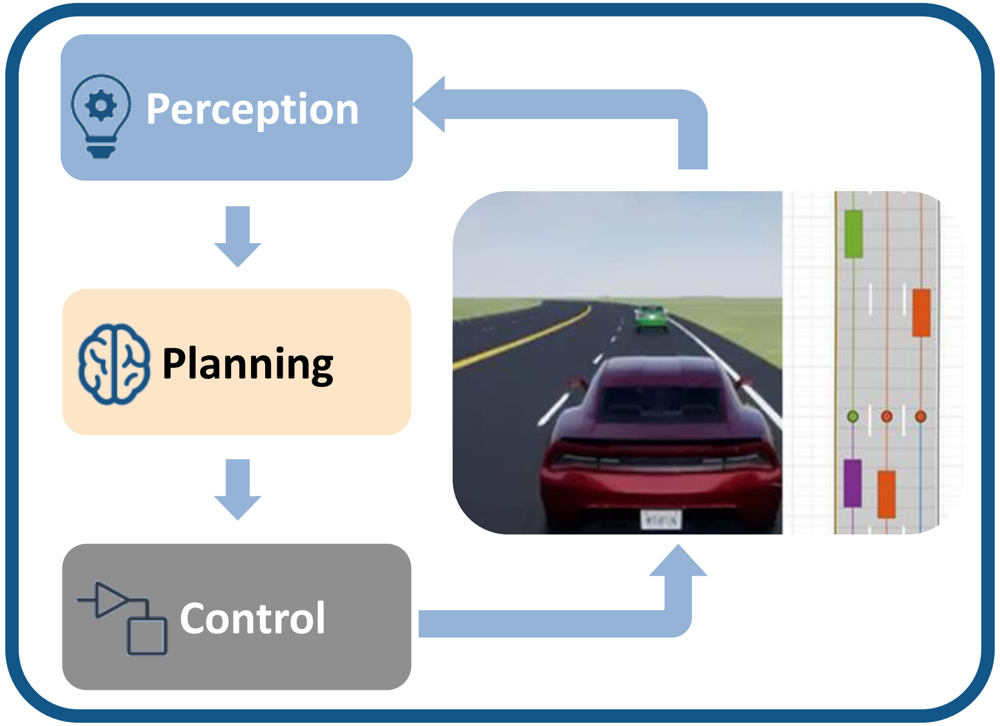


Height

Bridging ROS with MATLAB and Simulink: From Algorithm To Deployment



Deploy and test sensor fusion and navigation algorithms on hardware

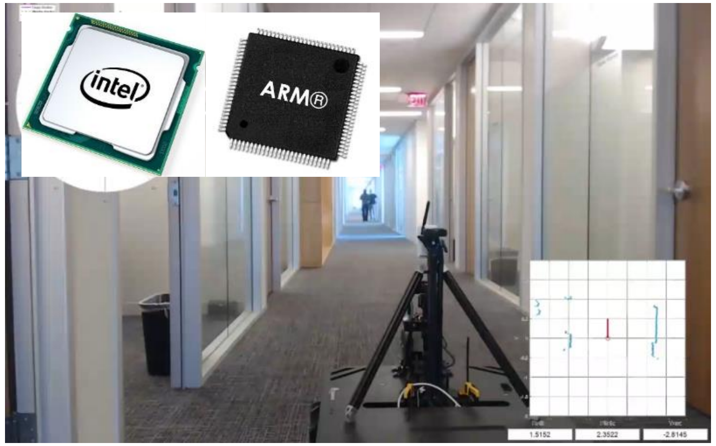


C/C++ Code

HDL Code

GPU Code

Processors



FPGAs

GPUs

Full Model Based Design Workflow for Autonomous Systems

Verification & Validation

Connect / Deploy

Code Generation

ROS Toolbox

AUTOSAR Blockset

Autonomous Algorithms

Sensor Fusion and Tracking Toolbox

Perceive

Plan & Decide

Navigation Toolbox

Robotics System Toolbox

Automated Driving Toolbox

Stateflow

Reinforcement Learning Toolbox

Model Predictive Control Toolbox

Control

Sense

Computer Vision Toolbox

Image Acquisition Toolbox

Platform

MATLAB

Simulink

You can lower risk in your autonomous navigation development

Localization

Mapping

Tracking

Perception

Motion Planning

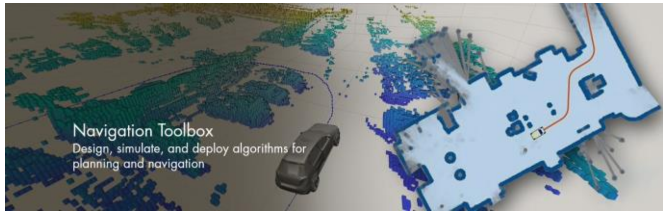
Path Metrics

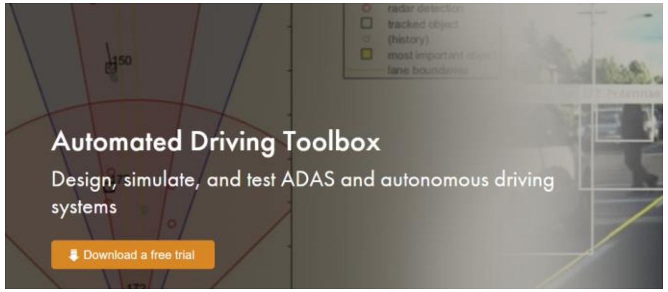
Planning

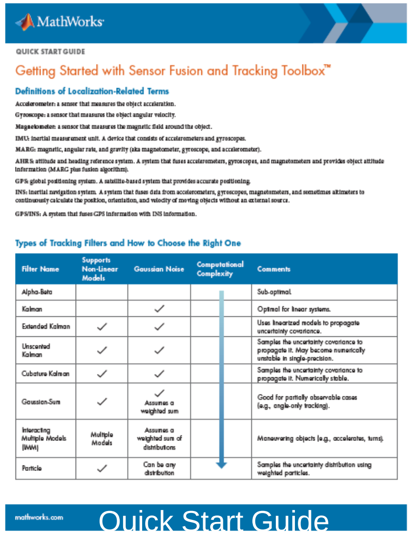
Path/Waypoint Following

Control

There are many resources to get started with







MathWorks

QUICK START GUIDE

Getting Started with Sensor Fusion and Tracking Toolbox™

Definitions of Localization-Related Terms

Accelerometer: a sensor that measures the object acceleration.

Gyroscope: a sensor that measures the object angular velocity.

Magnetometer: a sensor that measures the magnetic field around the object.

IMU: inertial measurement unit. A device that consists of accelerometers and gyroscopes.

MARG: magnetic, angular rate, and gravity (aka magnetometer, gyroscope, and accelerometer).

AHRS: attitude and heading reference system. A system that fuses accelerometers, gyroscopes, and magnetometers and provides object attitude information (MARG plus fusion algorithm).

GPS: global positioning system. A satellite-based system that provides accurate positioning.

INS: inertial navigation system. A system that fuses data from accelerometers, gyroscopes, magnetometers, and sometimes altimeters to continuously calculate the position, orientation, and velocity of moving objects without an external source.

GPS/INS: A system that fuses GPS information with INS information.

Types of Tracking Filters and How to Choose the Right One

Filter Name	Supports Non-Linear Models	Gaussian Noise	Computational Complexity	Comments
Alpha-Beta				Sub-optimal.
Kalman		✓		Optimal for linear systems.
Extended Kalman	✓	✓		Uses linearized models to propagate uncertainty covariance.
Unscented Kalman	✓	✓		Samples the uncertainty covariance to propagate it. May become numerically unstable in single-precision.
Cubature Kalman	✓	✓		Samples the uncertainty covariance to propagate it. Numerically stable.
Gaussian-Sum	✓	✓ Assumes a weighted sum		Good for partially observable cases (e.g., angle-only tracking).
Interacting Multiple Models (IMM)	Multiple Models	Assumes a weighted sum of distributions		Maneuvering objects (e.g., accelerates, turns).
Particle	✓	Can be any distribution		Samples the uncertainty distribution using weighted particles.

mathworks.com

Quick Start Guide

Tech Talks

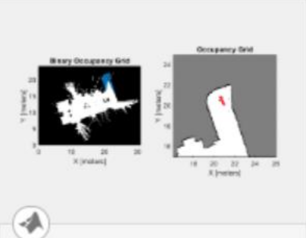
Series: Understanding Sensor Fusion and Tracking

Part 1: What is Sensor Fusion?

This video provides an overview of what sensor fusion is and how it helps in the design of autonomous systems. It also covers a few scenarios that illustrate the various ways in which sensor fusion can be implemented.

Part 2: Fusing a Mag, Accel, and Gyro to Estimate Orientation

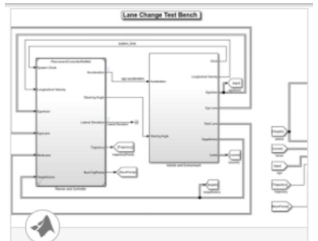
This video describes how we can use a magnetometer, accelerometer, and a gyro to estimate an object's orientation. The goal is to show how these sensors contribute to the solution, and to explain a few things to watch out for along the way.



Create Egocentric Occupancy Maps using Range Sensors

Create an egocentric occupancy map by using ray-tracing with our rangeSensor sensor model.

Open Live Script



Lane Change for Highway Driving

Simulate an automated lane change maneuver system for highway driving scenario.

Open Live Script

Contact for Training & Consulting Services

Exploit the full potential of MathWorks products

Thank you!

abhishet@mathworks.com