MATLAB EXPO 2018

Automated Driving Development
with MATLAB® and Simulink®

Marc Willerton
Application Engineering, MathWorks
marc.willerton@mathworks.co.uk

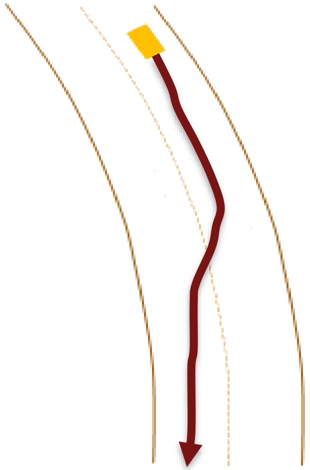
Challenges in Automated Driving

Understanding the area around me

Should I stop?

Perception

Understanding the best way to get there



How should I get there?

Planning

Understanding how to move

How should I navigate this path?

Control

Designing a forward collision warning system

Deep Learning

Certification

GPU

Labelling

Data generation

Surface mapping

Tracking

Mapping

Detection

Radar

Vehicle Dynamics

Computer Vision

Perception

Control

CAN

SLAM

IMU

Scenario Testing

GPS

Sensor Fusion

Model Predictive Control

Planning

Data logging

Weather effects

FPGA

ROS

Lidar

V2V

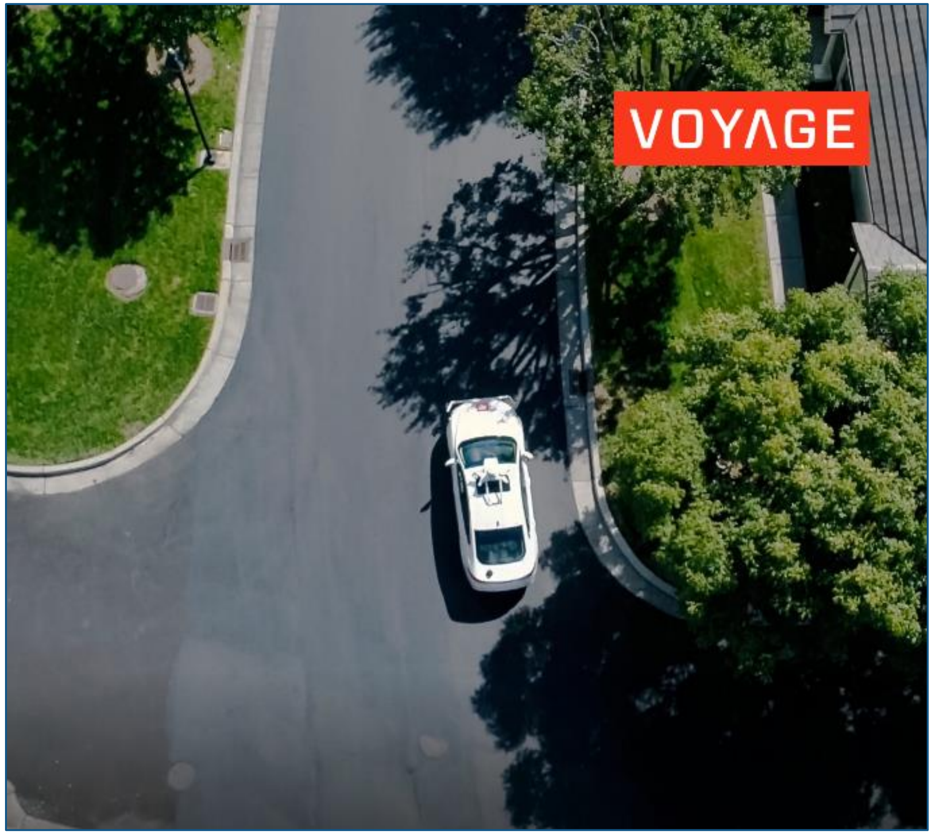
Wheel Encoder

Depth Perception

Voyage Develops longitudinal controls for self-driving taxis

Designed longitudinal model predictive controller, generated ROS node, and integrated with their open source software framework for perception and planning

> *"We were searching for a prototyping solution that was fast for development and robust for production. We decided to go with Simulink for controller development and code generation, while using MATLAB to automate development tasks."*
>
> *- Alan Mond, Voyage*



Voyage's self driving car in San Jose, California.

Voyage user story, 2017

Scania Re-simulation workflow to develop AEB sensor fusion algorithm

[Scania presentation recording]() from 2015 MathWorks Automotive Conference, Germany

Source: Model-Based Design Approach to Resource Efficient Object Fusion for an Autonomous Braking System, Jonny Andersson, Scania

How can I make a forward collision warning system?

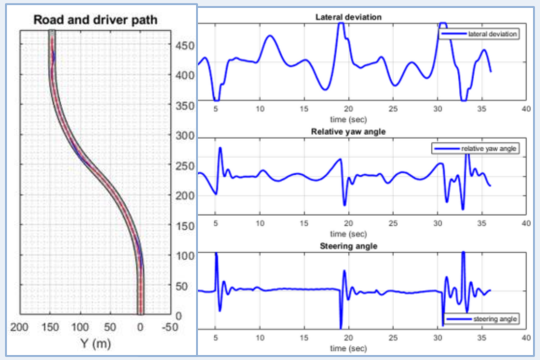
Classification (Deep learning)

Sensor detection visualization

Perception

Control

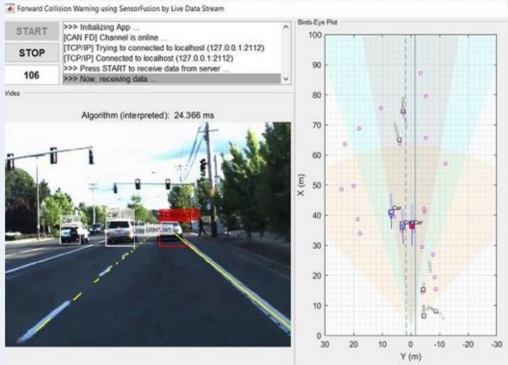
Control and Vehicle Dynamics



Sensor development

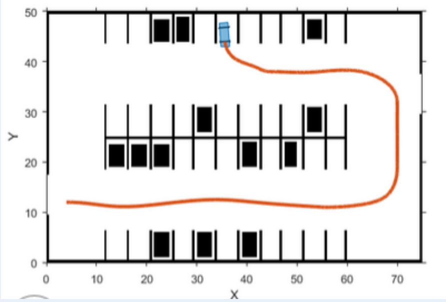
Scenario testing

Sensor fusion and tracking



Planning

Path planning



SLAM

ROS

How can I make a forward collision warning system?

Classification
(Deep learning)

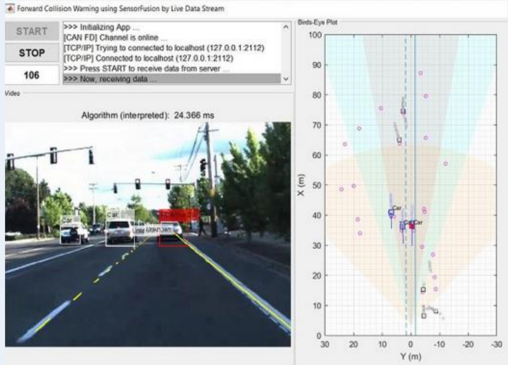
Sensor detection visualization

Perception

Sensor development

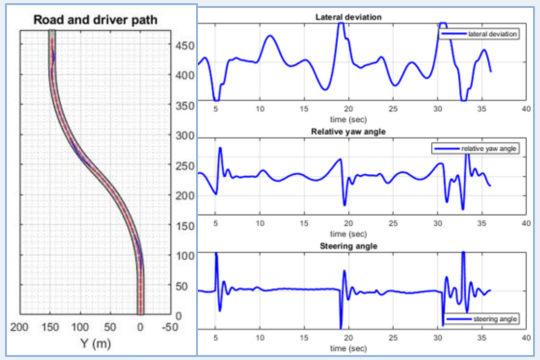
Scenario testing

Sensor fusion and tracking



Control

Control and Vehicle Dynamics

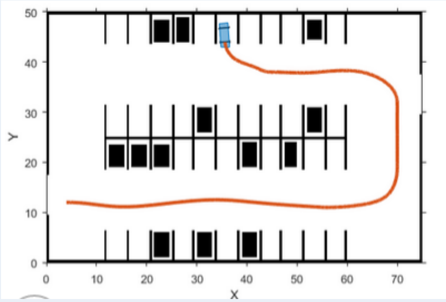


Planning

SLAM

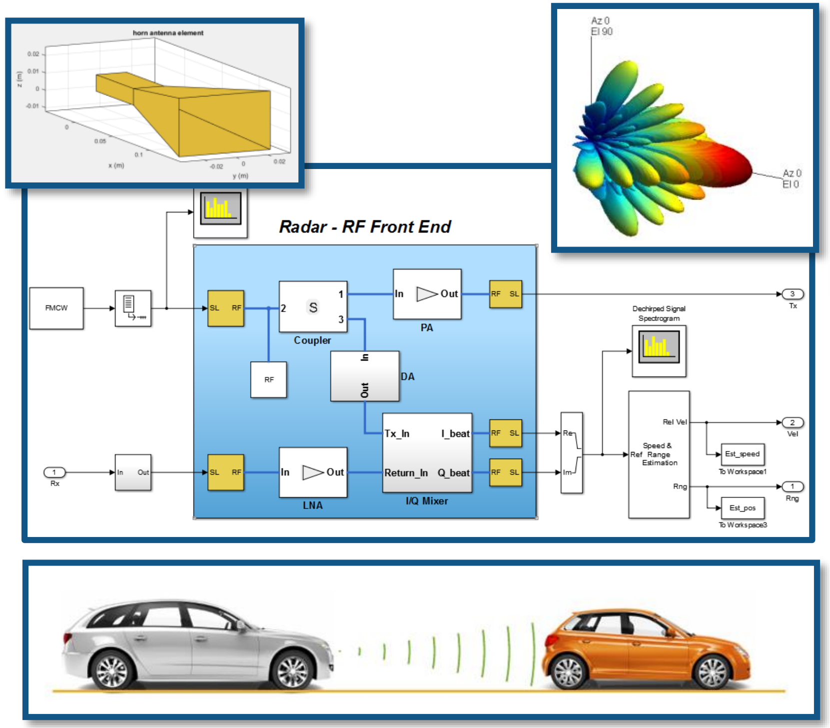
ROS

Path planning



Sensor Development in MATLAB and Simulink

Radar

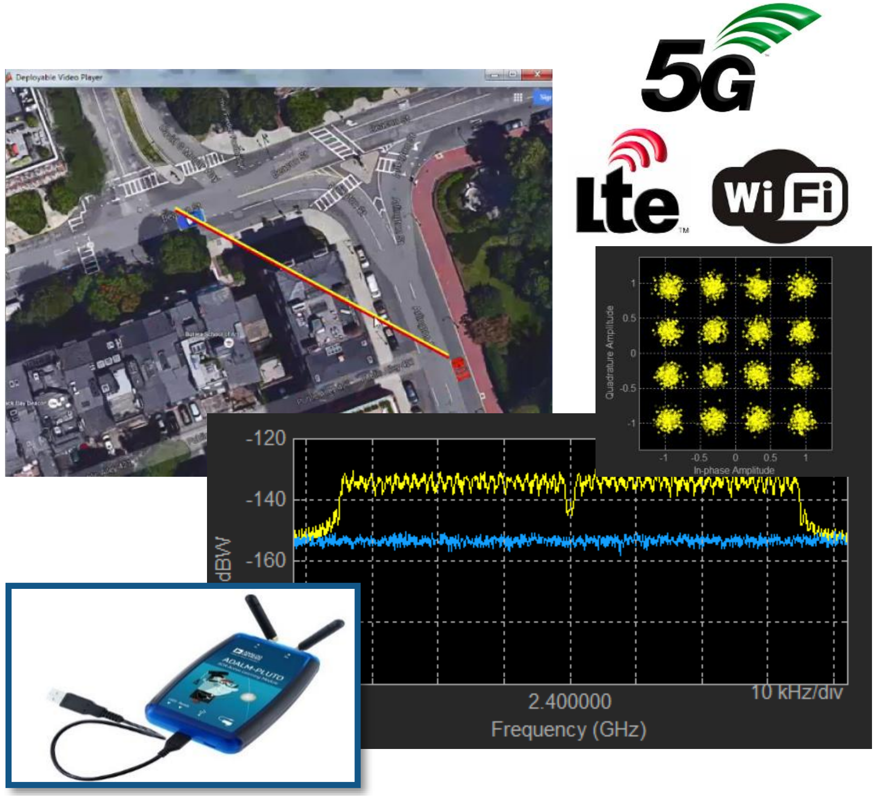


- Antenna Modelling
- Antenna Array Design
- Beamforming and Direction Finding Algorithms
- Detect and Track objects
- Develop RF component architectures and model imperfections
- Radar control

Vision

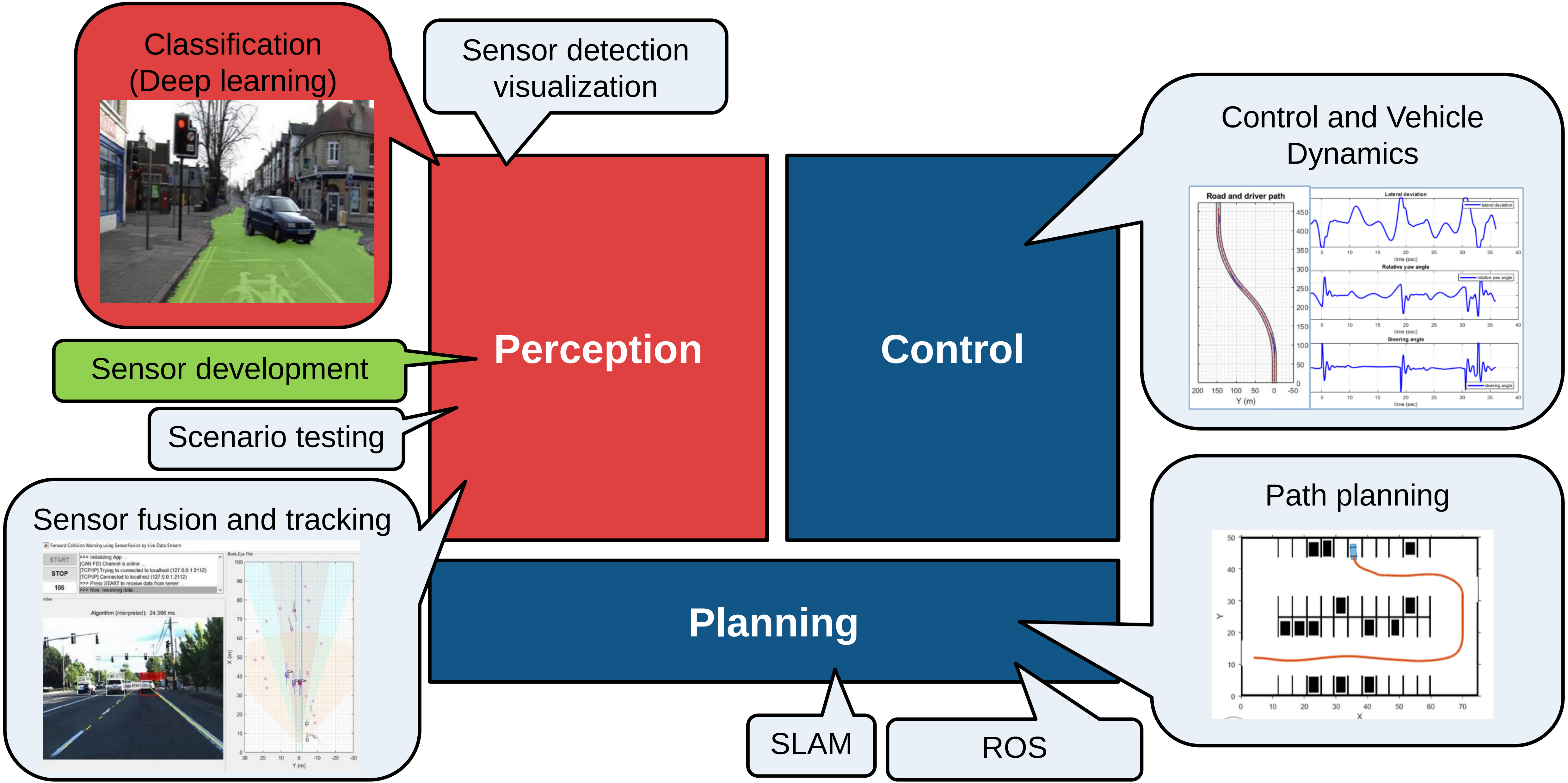
- Feature detection, extraction and matching
- Image segmentation, registration, enhancement
- Detect and Track objects
- Image overlay
- Feed to deep learning algorithms
- Deploy to processor / GPU / FPGA

Wireless Communications



- Extract DSRC + C-V2X performance metrics
- PHY Communications Modelling
- Pre-built standards based functions
- RF propagation models and visualisations
- Prototype with Software Defined Radio

How can I make a forward collision warning system?



Classification (Deep Learning)

Deep Learning approach

End-to-end learning

Feature learning + Classification

Car ✓

Truck

:

Bicycle

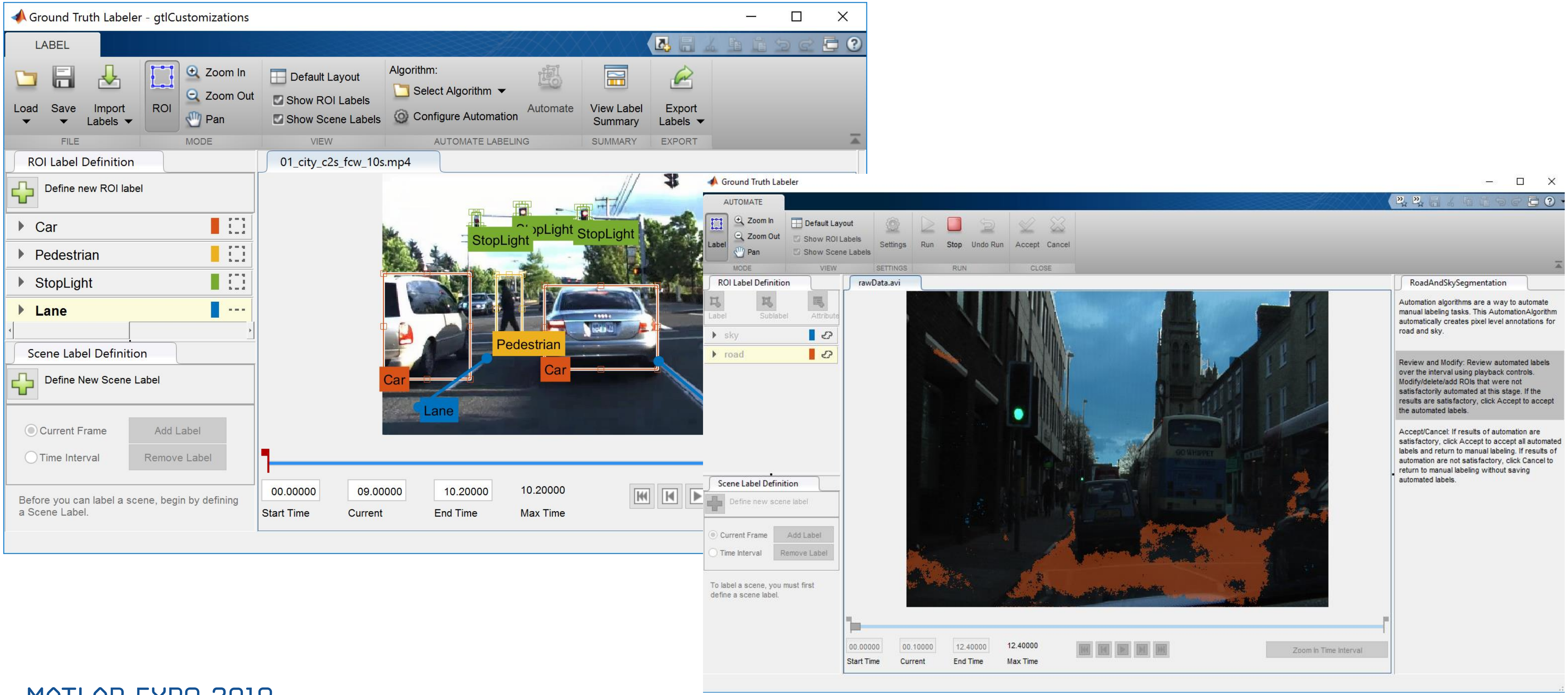
Deep Learning Made Easy in MATLAB

Semantic Segmentation Using Deep Learning

Train free space detection network using deep learning

Computer Vision System Toolbox™

Ground Truth Labeling App for labelling Training Data automatically



Generate CUDA Code for Embedded Deployment

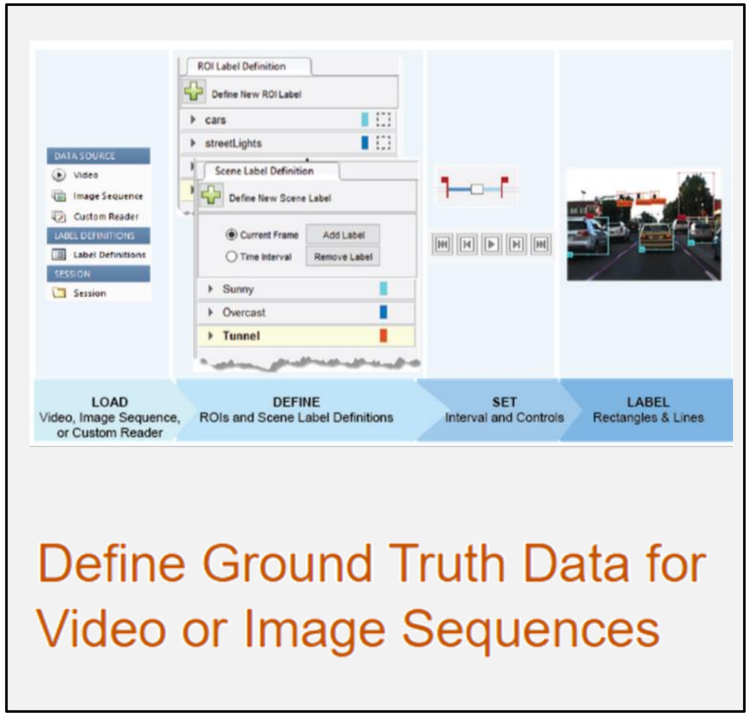
```
% Save network to MAT file
save('SegNet.mat', 'net’)
```

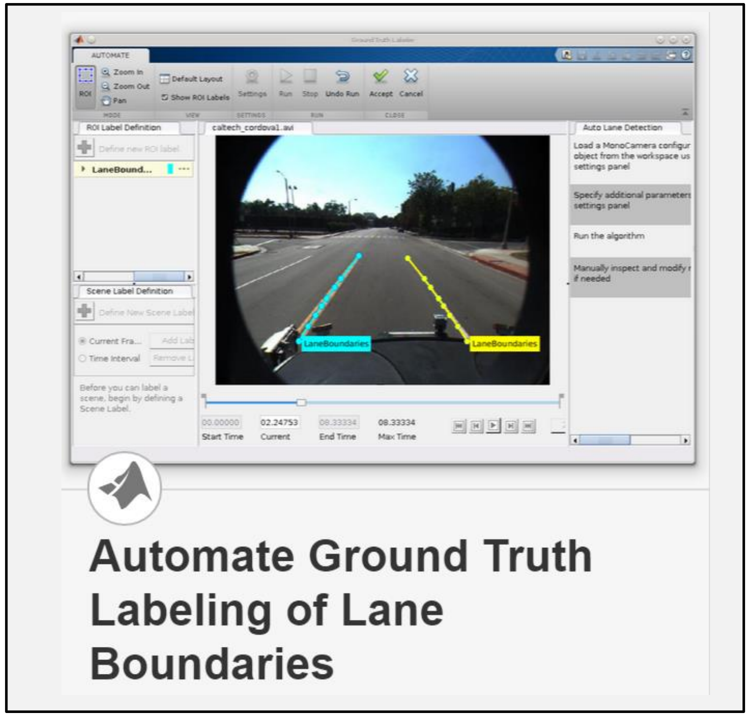
```
function out = segnet_predict(in) %#codegen
persistent mynet;
if isempty(mynet)
    mynet = coder.loadDeepLearningNetwork('SegNet.mat');
end
out = predict(mynet,in);
```

```
% Generate CUDA code
cfg = coder.config('lib');
cfg.TargetLang = 'C++';
codegen -config cfg segnet_predict -args
{ones(360,480,3,'uint8')} -report
```

Automated Driving System Toolbox introduced examples to:

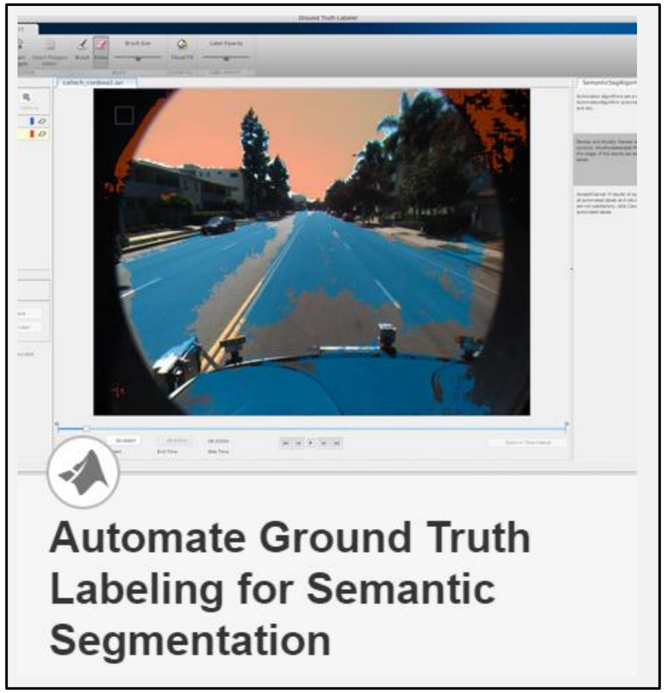
Accelerate the process of Ground Truth Labeling





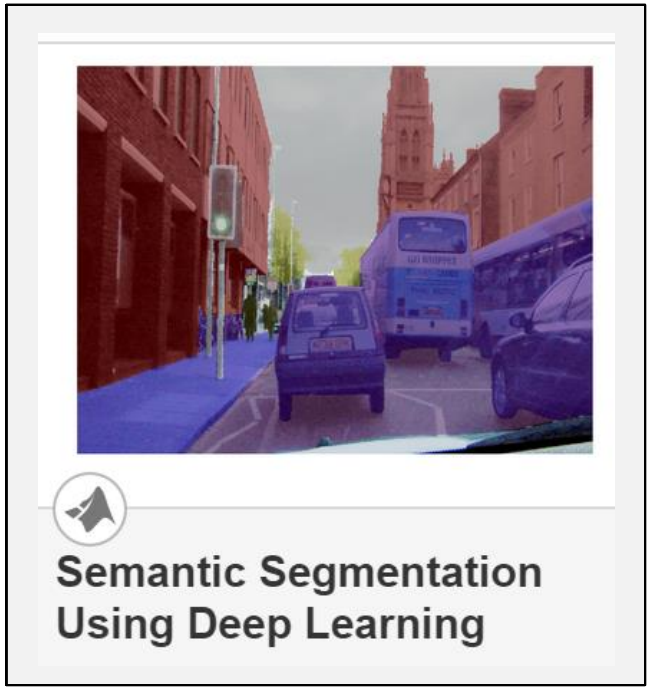
- **Label detections** with Ground Truth Labeler App
- **Add your own automation algorithm** to Ground Truth Labeler App
- **Extend connectivity** of Ground Truth Labeler App to visualize lidar data

Learn more about developing deep learning perception algorithms with these examples



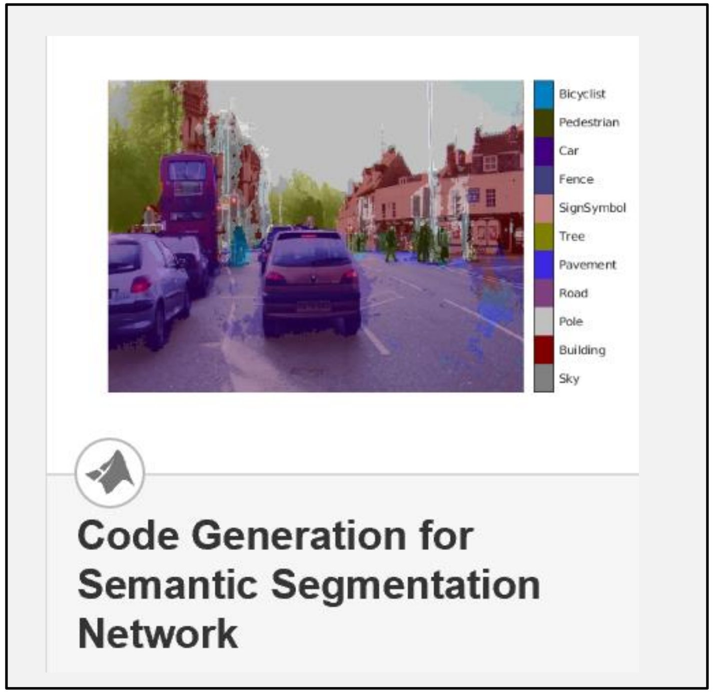
- **Add semantic segmentation** automation algorithm to Ground Truth Labeler App

  Automated Driving System Toolbox™



- **Train free space detection network** using deep learning

  Computer Vision System Toolbox™



- **Generate CUDA® code** to execute directed acyclic graph network on an NVIDIA GPU

  GPU Coder™

How can I make a forward collision warning system?

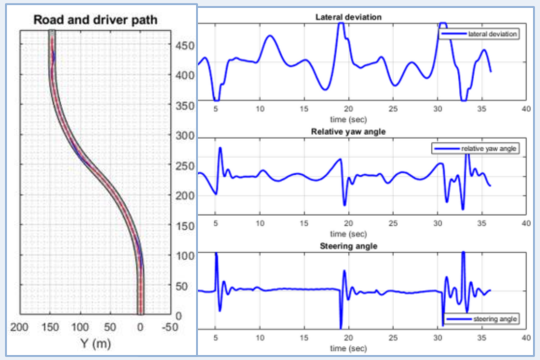
Classification (Deep learning)

Sensor detection visualization

Perception

Control

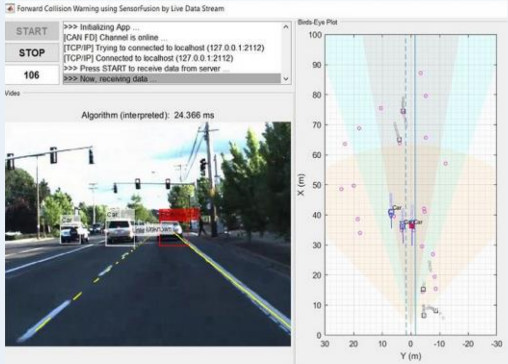
Control and Vehicle Dynamics



Sensor development

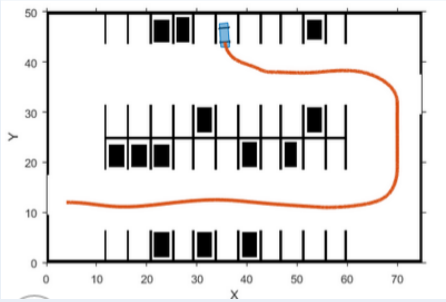
Scenario testing

Sensor fusion and tracking



Planning

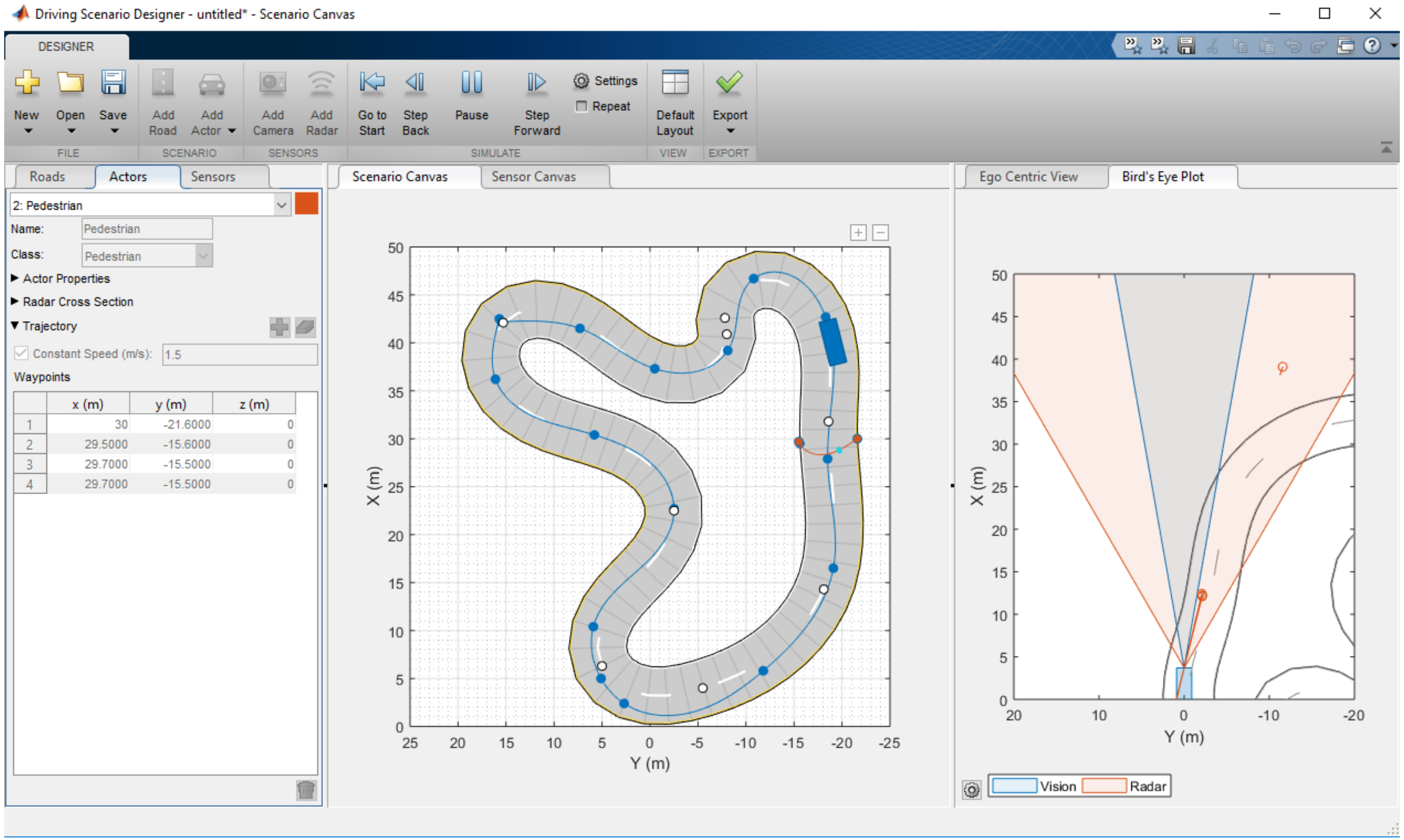
Path planning



SLAM

ROS

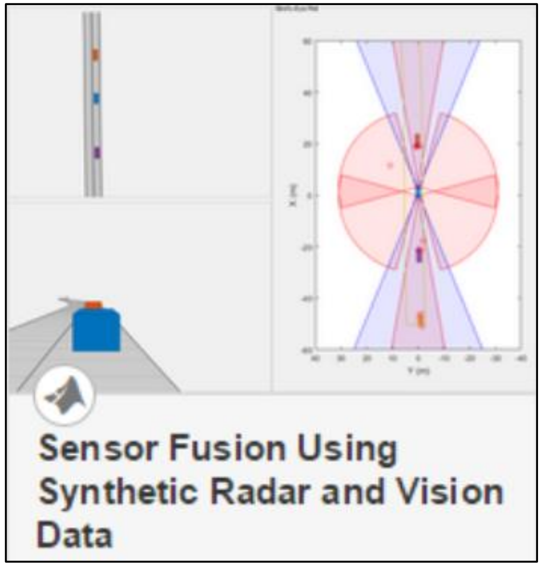
Example: Synthesizing scenarios to test sensor fusion algorithms



Build scenarios programmatically or interactively

Load prebuilt scenarios (e.g. NCAP)

Export to MATLAB / Simulink



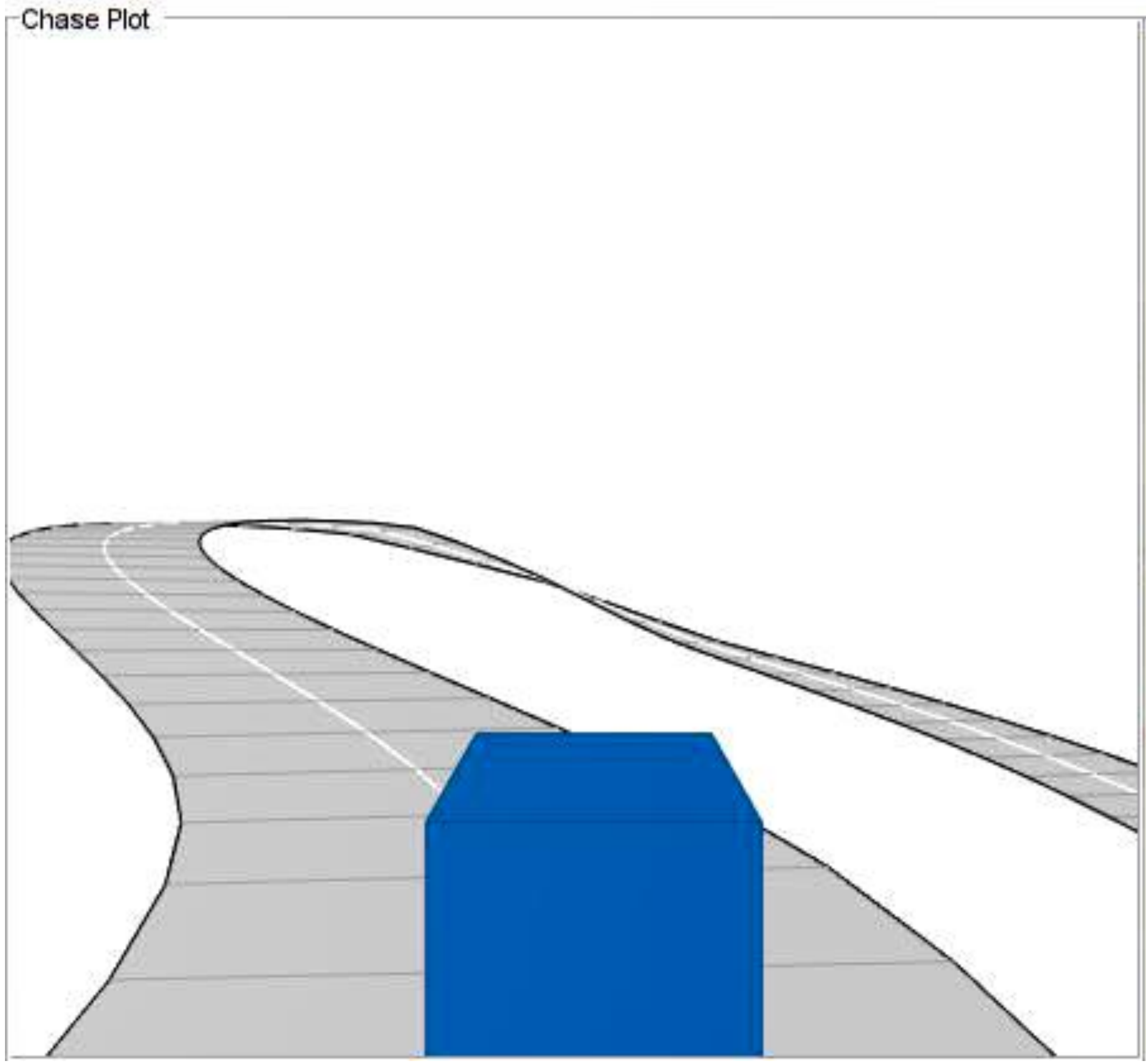
Synthesize driving scenario to test multi-object tracker

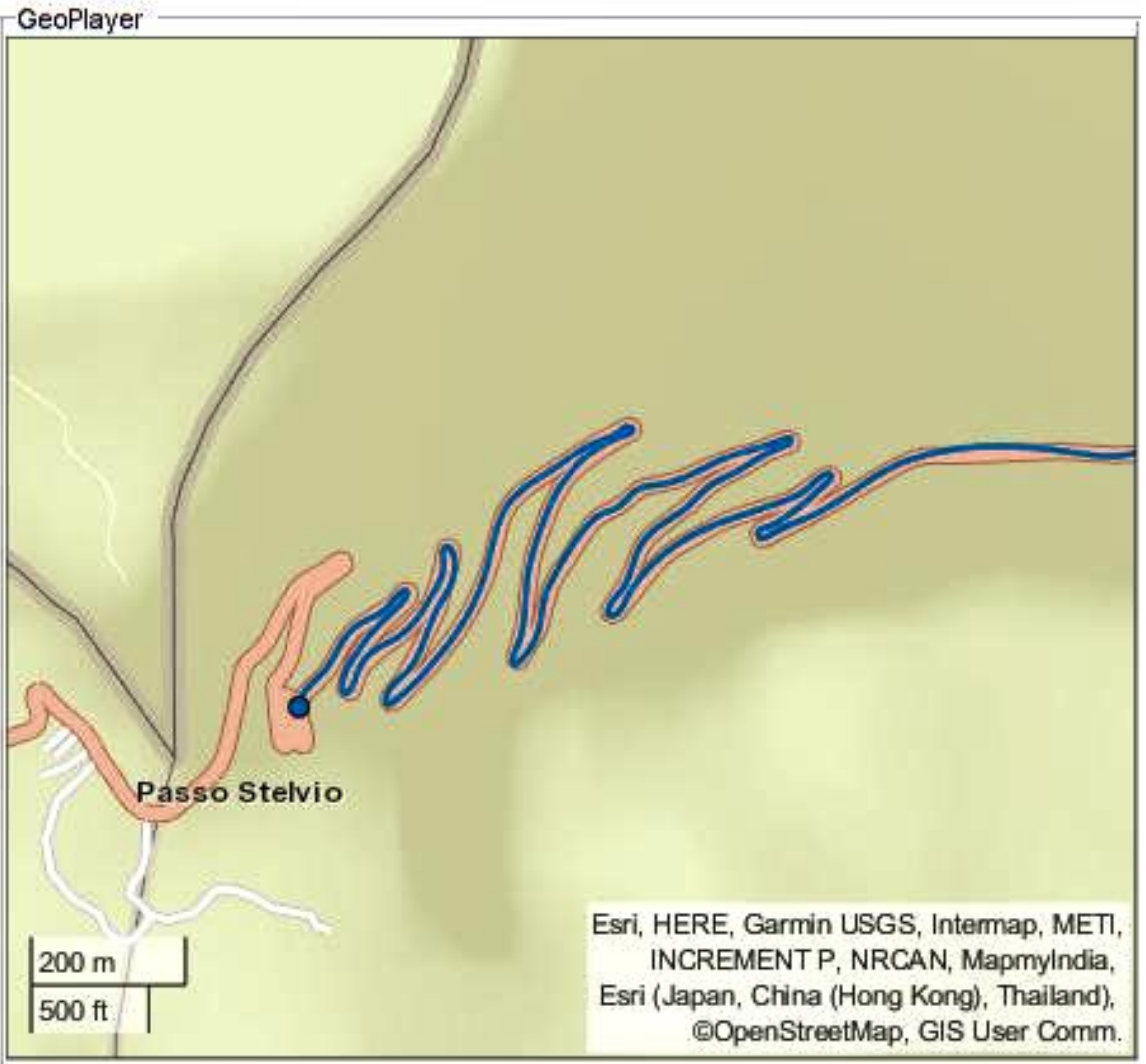
Example: Stelvio Pass GPX Data Import

Complex paths overlaid on maps can be imported from external logs and then simulated with models of cars and sensors

```
p = gpxread('StelvioPass.gpx');

flatearth_pos = lla2flat([ p.Longitude' p.Latitude' p.Elevation'],...
    [p.Longitude(1),p.Latitude(1)],0,p.Elevation(1));

s=drivingScenario;
roadWidth = 4;
road(s,flatearth_pos,roadWidth);

v1 = vehicle(s,'Position',flatearth_pos(1,:),'Velocity',[0 0 0],
'Yaw', 140);

speed = 50;
path(v1,flatearth_pos,speed);

chasePlot(v1,'Centerline','on');
player = geoplayer(p.Longitude(1), p.Latitude(1));

restart(s)
while advance(s)
   t = targetPoses(v1);
   a = flat2lla(s.Actors.Position,roadCentres(1,1:2),...
         0,roadCentres(1,3));
   plotPosition(player,a(2),a(1));
end
```

Example: Stelvio Pass GPX Data Import





How can I make a forward collision warning system?

Classification (Deep learning)

Sensor detection visualization

Perception

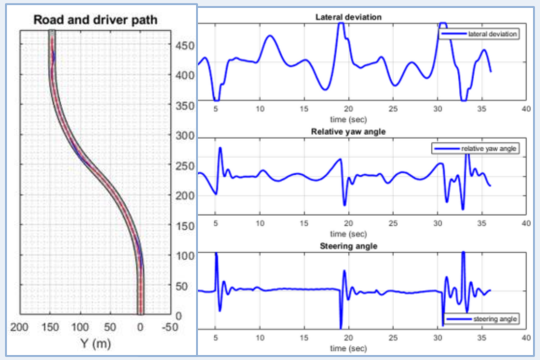
Sensor development

Scenario testing

Sensor fusion and tracking

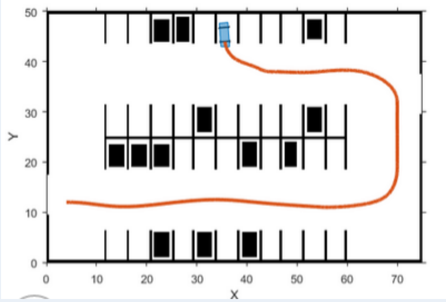
Control

Control and Vehicle Dynamics



Planning

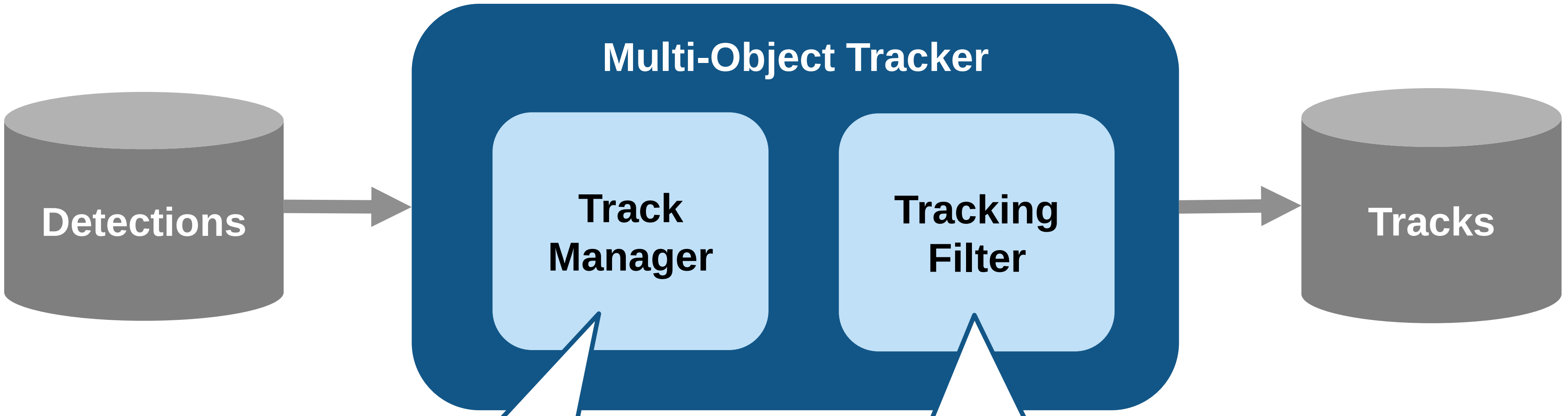
Path planning



SLAM

ROS

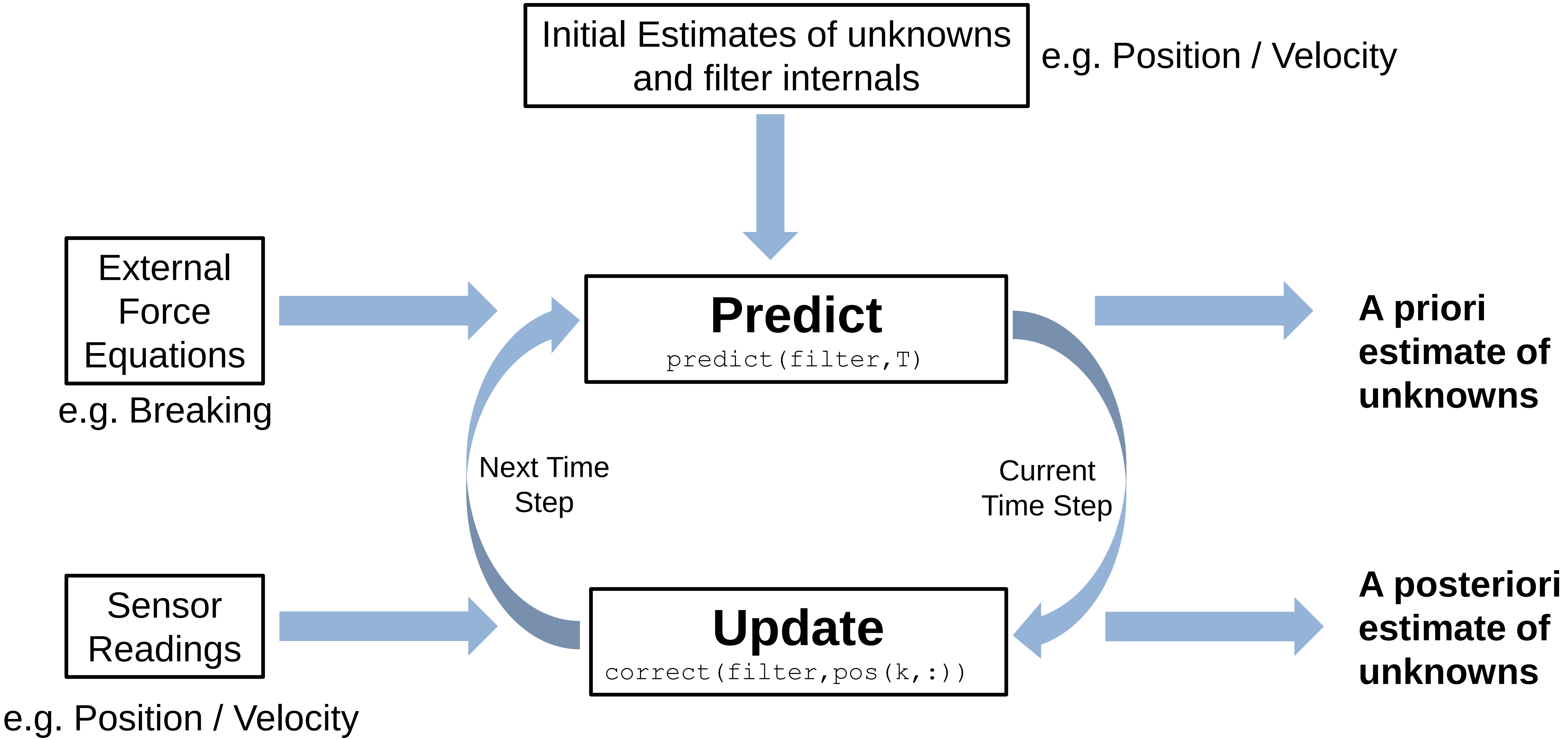
Multi-object tracker to develop sensor fusion algorithms



- Assigns detections to tracks
- Creates new tracks
- Updates existing tracks
- Removes old tracks

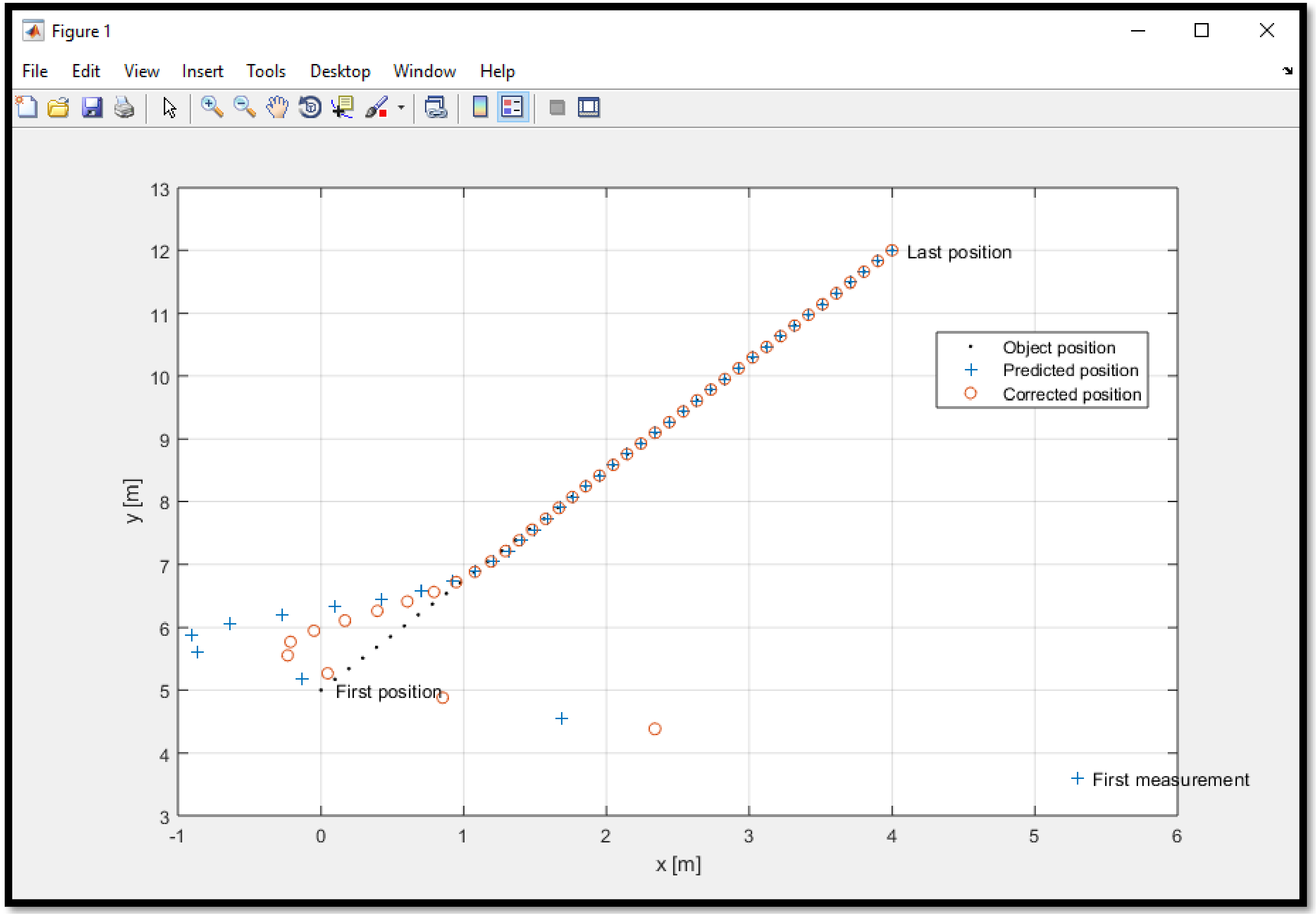
- Predicts and updates state of track
- Supports linear, extended, and unscented Kalman filters

Tracking Filter: Kalman Filter Basics



Constant Velocity Kalman Filter

```
initialState = [5.3;0;3.6;0];
filter = trackingKF('MotionModel','2D Constant…
Velocity','State',initialState);

vx = 0.2;
vy = 0.1;
T  = 0.5;
pos = [linspace(0,4,42); linspace(5,12,42)].';

for k = 1:size(pos,1)
    pstates(k,:) = predict(filter,T);
    cstates(k,:) = correct(filter,pos(k,:));
end

plotResults
```



Fuse and track multiple detections from different sensors

Radar Detections

Time
Position
Velocity

Vision Detections

Time
Position
Velocity

Detection Packer

Multi-Object Tracker

Track Manager

Kalman Filter

Tracks

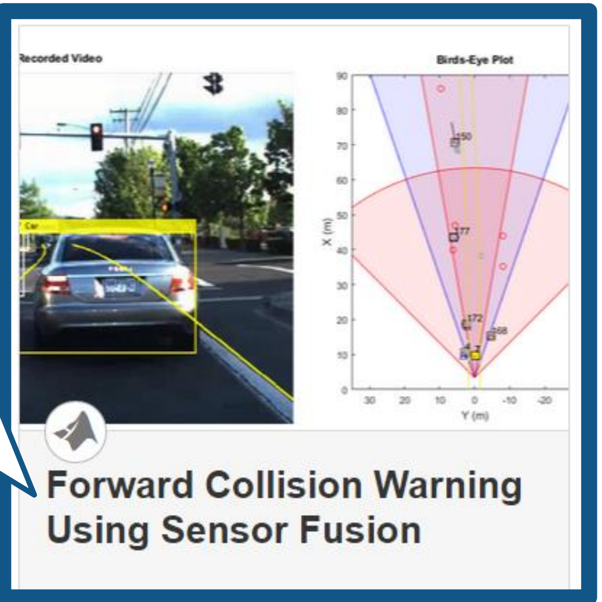
Time
State
Covariance
Track ID
Age
Is Confirmed
Is Coasted

Detections

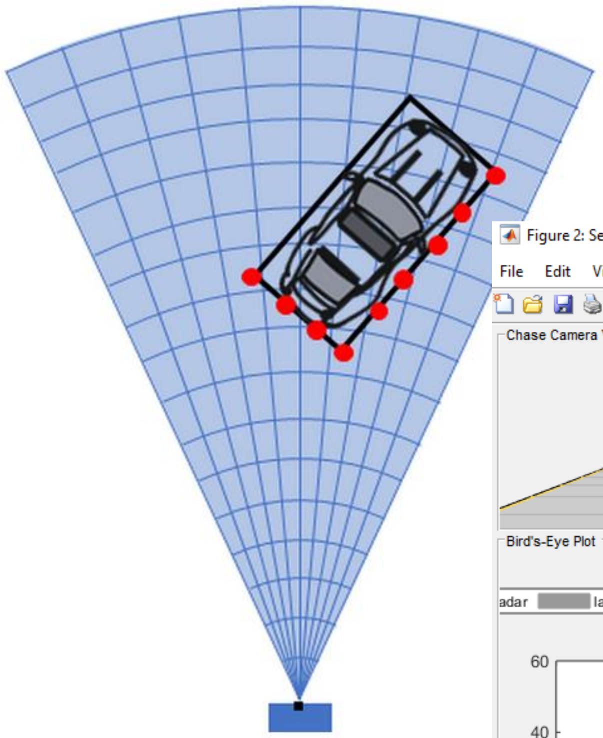
Time
Measurement
Measurement Noise

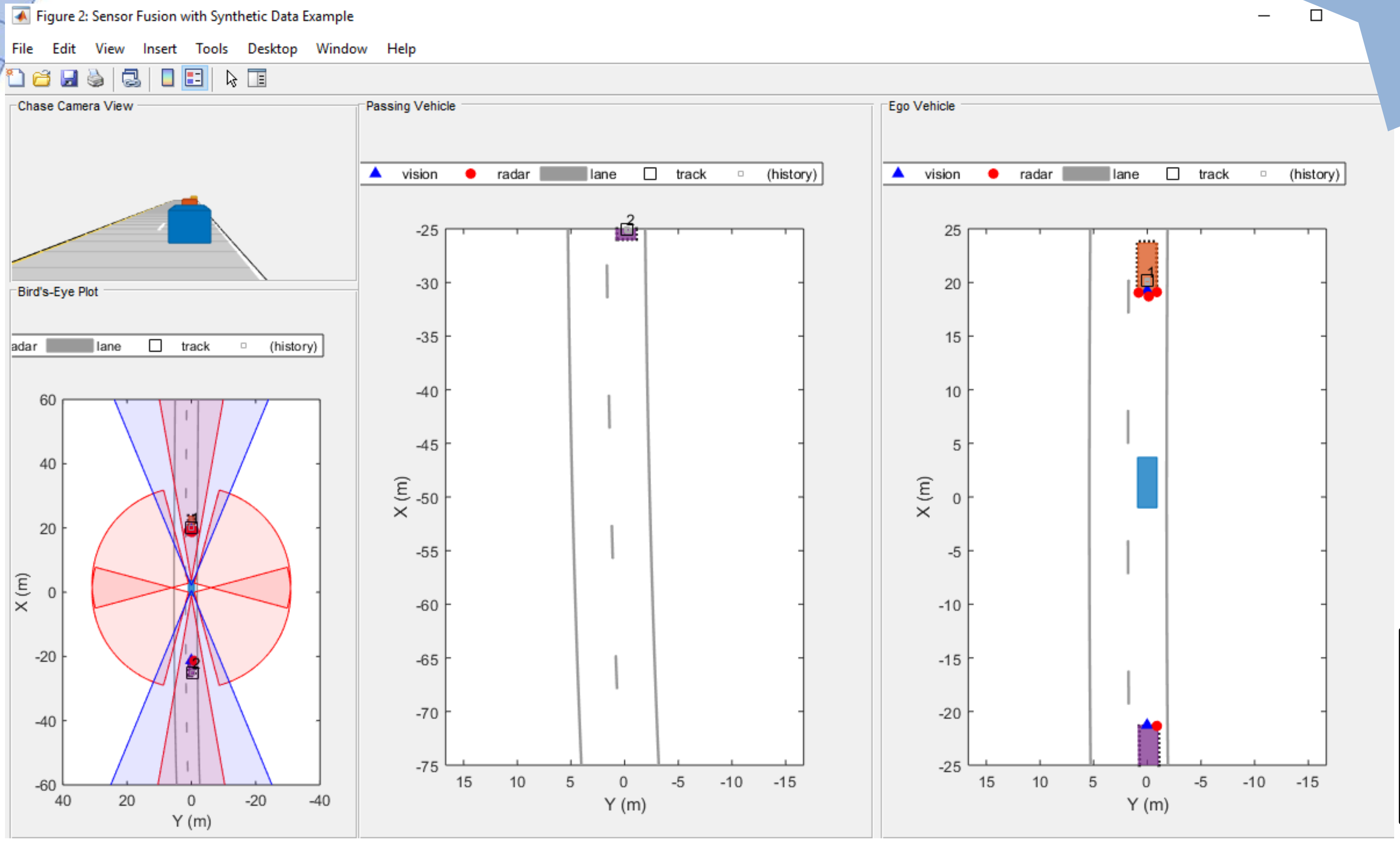
Forward Collision Warning Using Sensor Fusion

- Packing sensor data into object detections
- Initializing Kalman filter
- Configuring multi-object tracker



More complex tracking: Extended Object Tracking





New in R2018b: Sensor Fusion and Tracking Toolbox!

Enhanced Trackers and sensor models

How can I make a forward collision warning system?

Classification
(Deep learning)

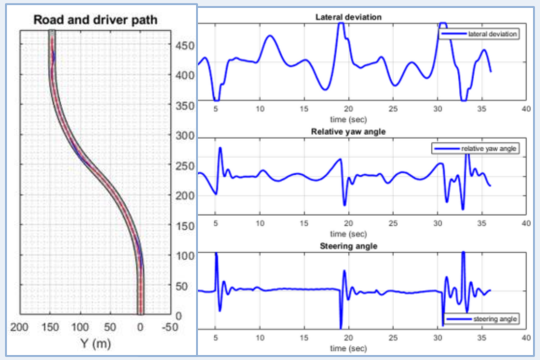
Sensor detection visualization

Perception

Control

Planning

Control and Vehicle Dynamics

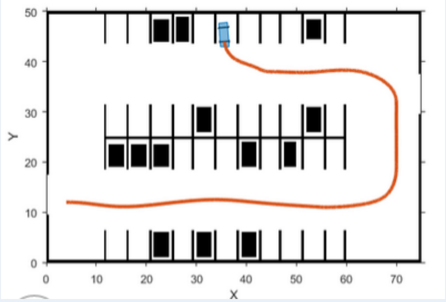


Sensor development

Scenario testing

Sensor fusion and tracking

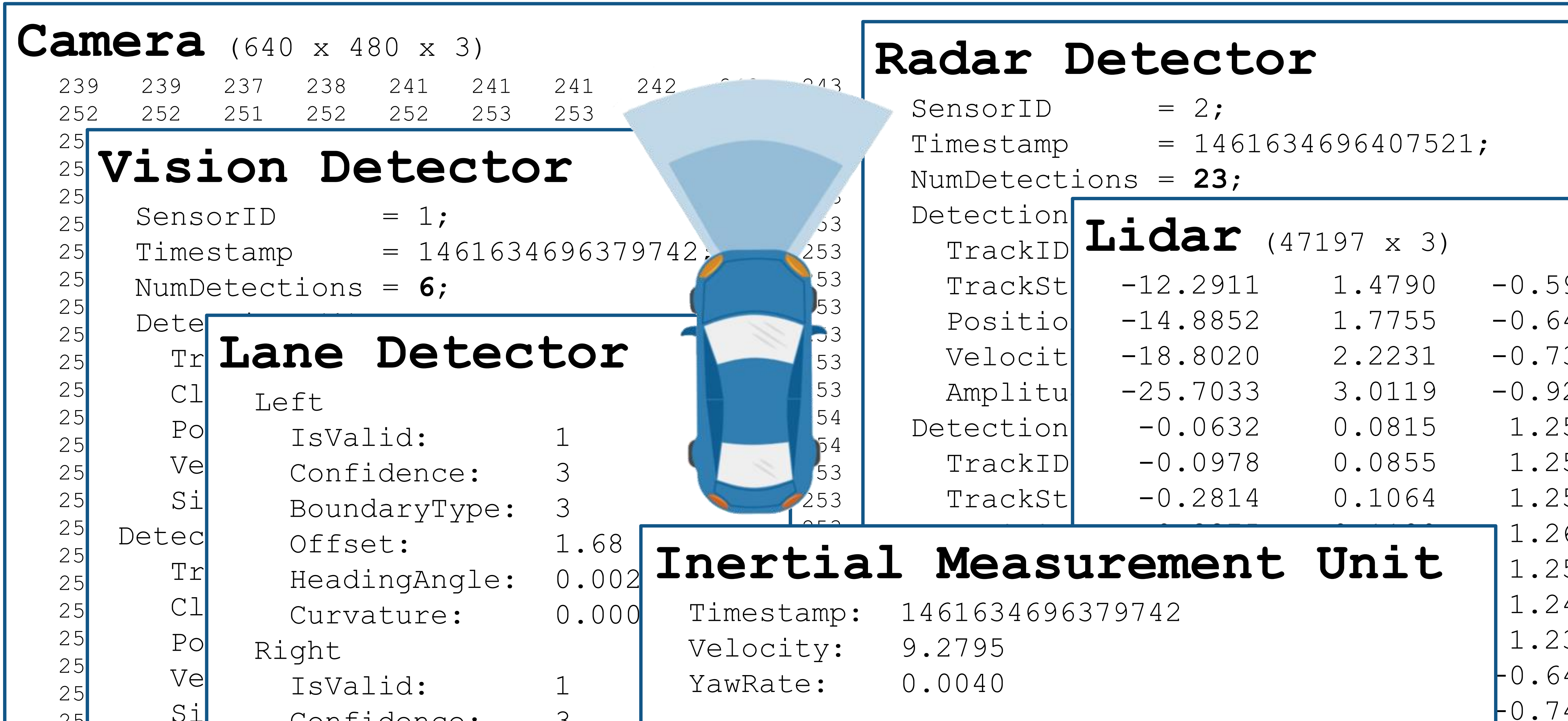
Path planning



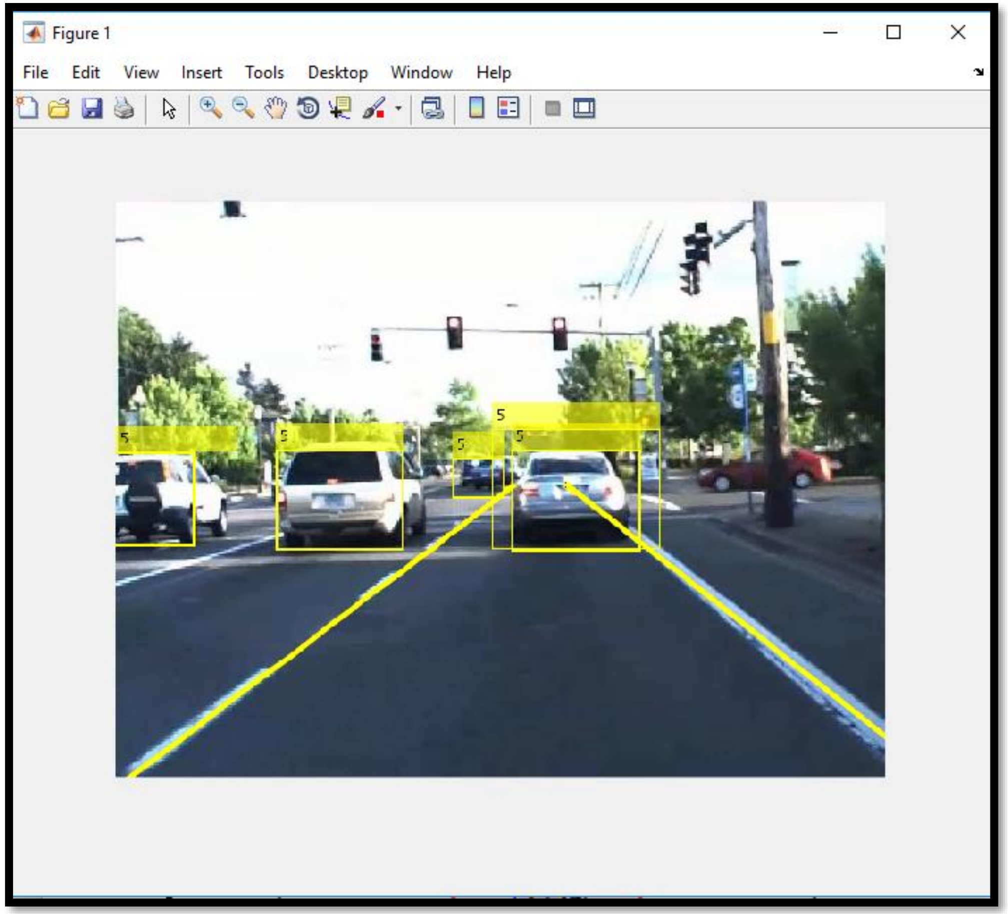
SLAM

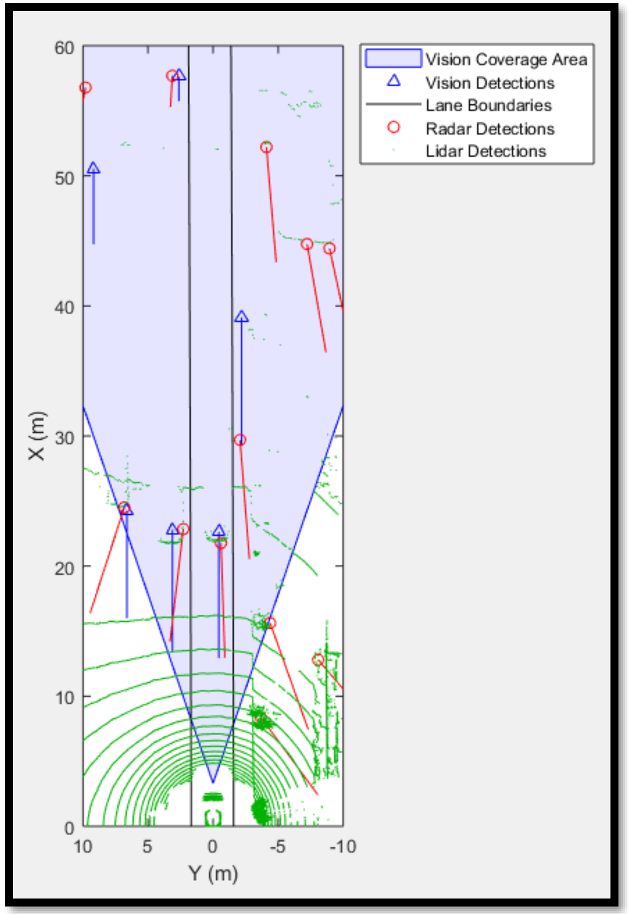
ROS

Sensor detection Visualisation



Example: Explore Logged Vehicle Data

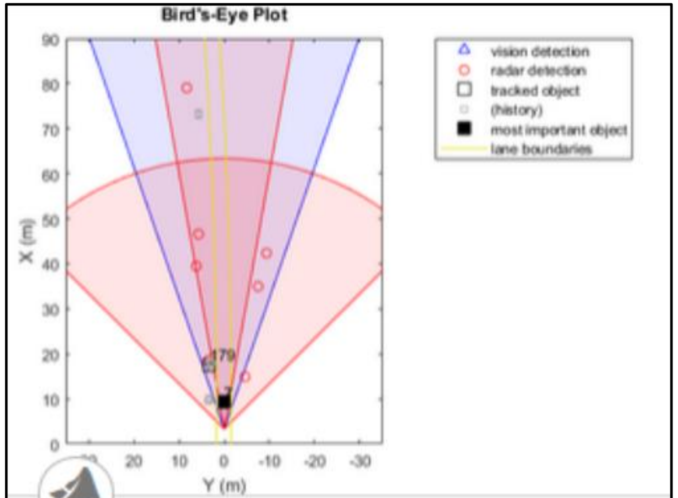




Bring in sensor detection data and build up a visualisation using this example:
web(fullfile(docroot, 'driving/examples/visualize-sensor-coverage-detections-and-tracks.html'))

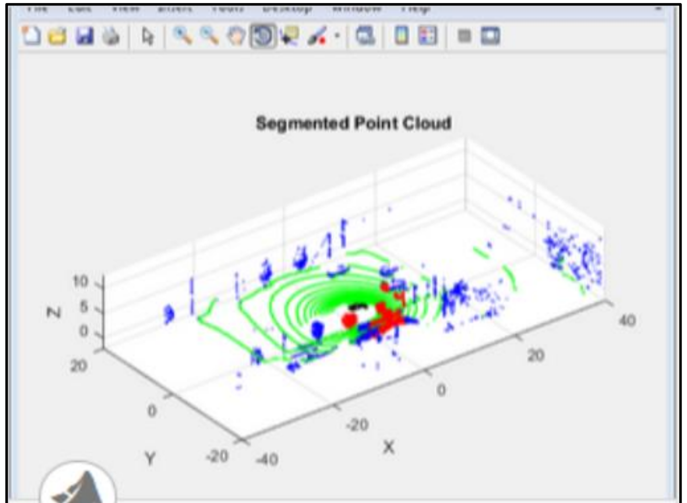
Learn more about visualizing vehicle data

by exploring examples in the Automated Driving System Toolbox



Visualize Sensor Coverage, Detections, and Tracks

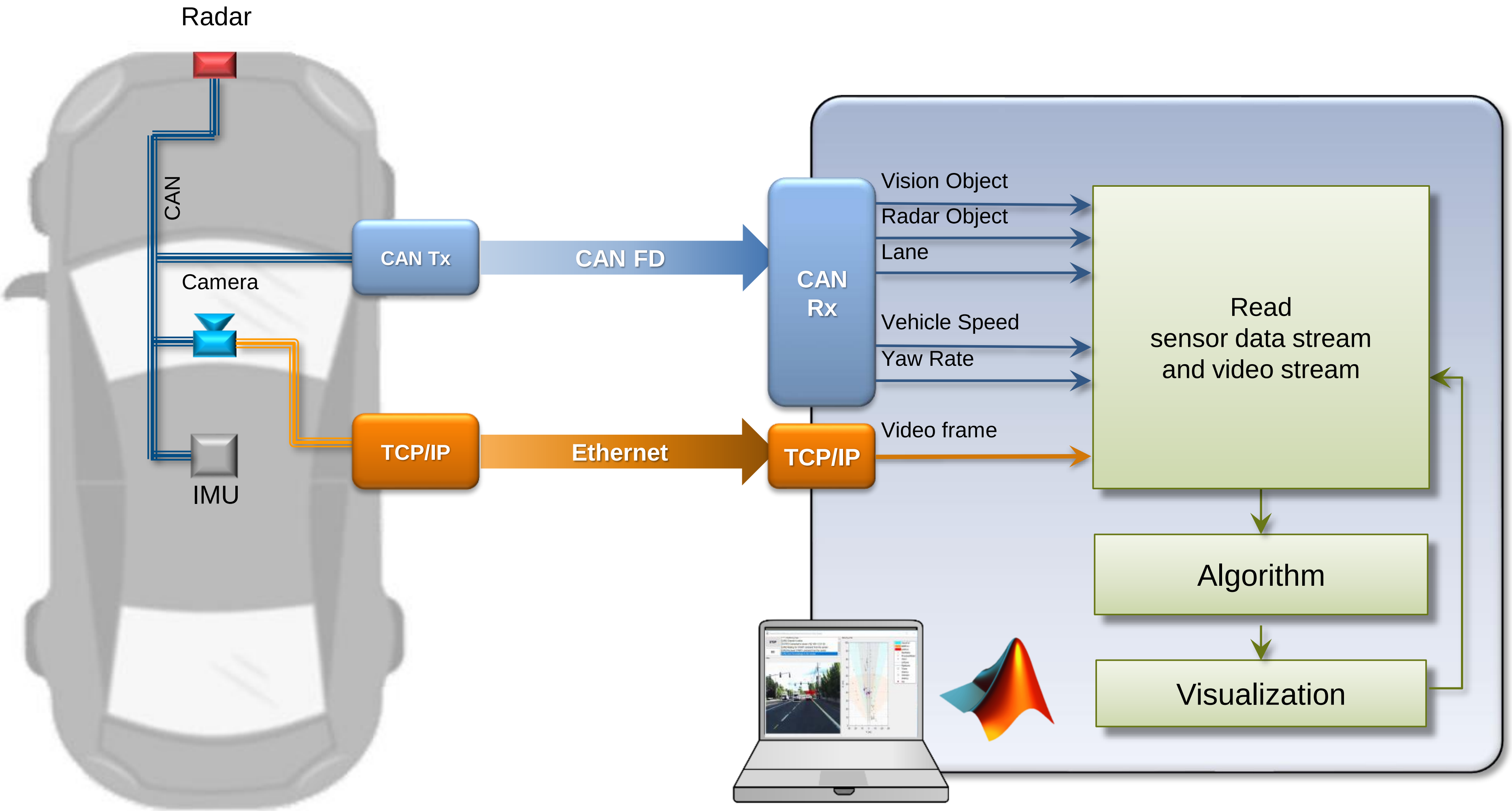
Annotate Video Using Detections in Vehicle Coordinates



Ground Plane and Obstacle Detection Using Lidar

- **Plot object detectors in vehicle coordinates**
  - Vision & radar detector
  - Lane detectors
  - Detector coverage areas
- **Transform between vehicle and image coordinates**
- **Plot lidar point cloud**

Testing perception algorithms with live data



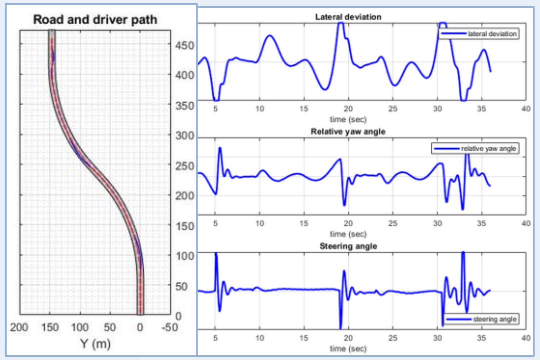
https://uk.mathworks.com/help/vnt/examples/forward-collision-warning-can-fd.html

How can I make a forward collision warning system?

Classification (Deep learning)

Sensor detection visualization

Control and Vehicle Dynamics



Perception

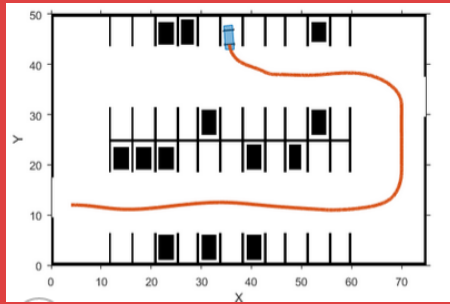
Control

Sensor development

Scenario testing

Sensor fusion

Path planning

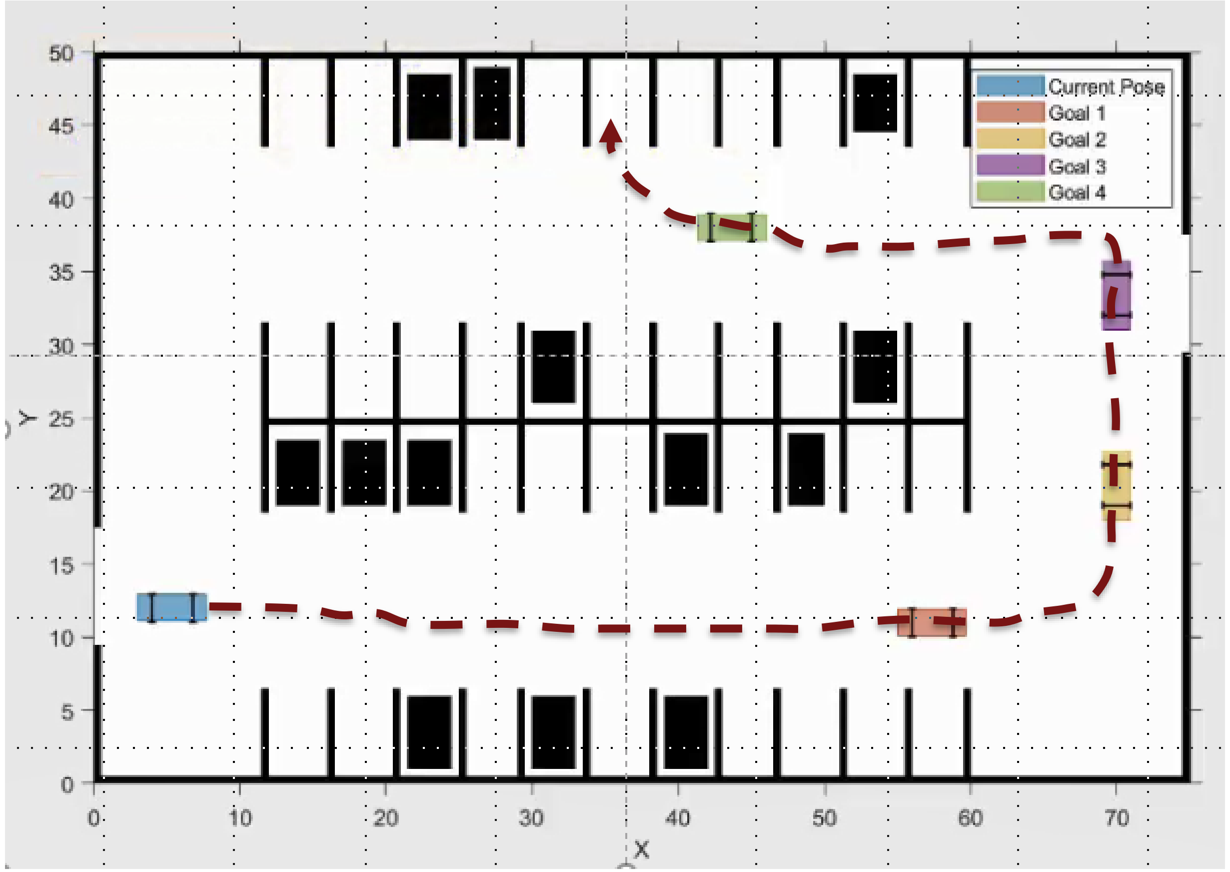


Planning

SLAM

ROS

What is Path Planning?



https://uk.mathworks.com/help/releases/R2018a/driving/examples/automated-parking-valet.html

Parking Valet Example

- Grid-based search
  - Split the area in to a grid
  - Move between adjacent grid squares by solving an optimization problem

- **Motion planning** – use Rapidly exploring Random Tree (RRT)
- Path smoothing – so a controller can execute
- Trajectory generation – create speed profile
- Vehicle control – get steering angle and generate acceleration commands
- **Goal checking**

- May also need to derive map
- May also be dynamic obstacles

How can I make a forward collision warning system?

Classification (Deep learning)

Sensor detection visualization

Sensor development

Scenario testing

Sensor fusion

Perception

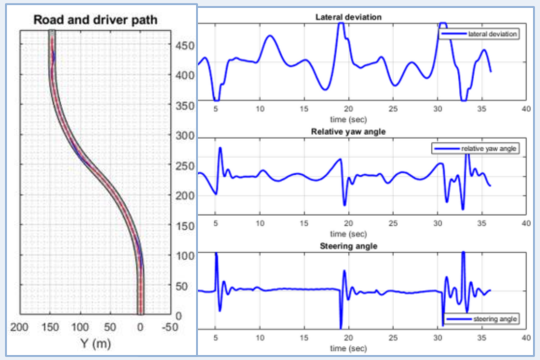
Control

Planning

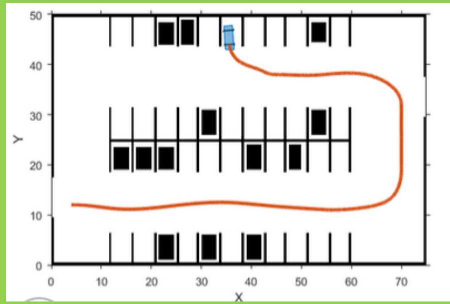
SLAM

ROS

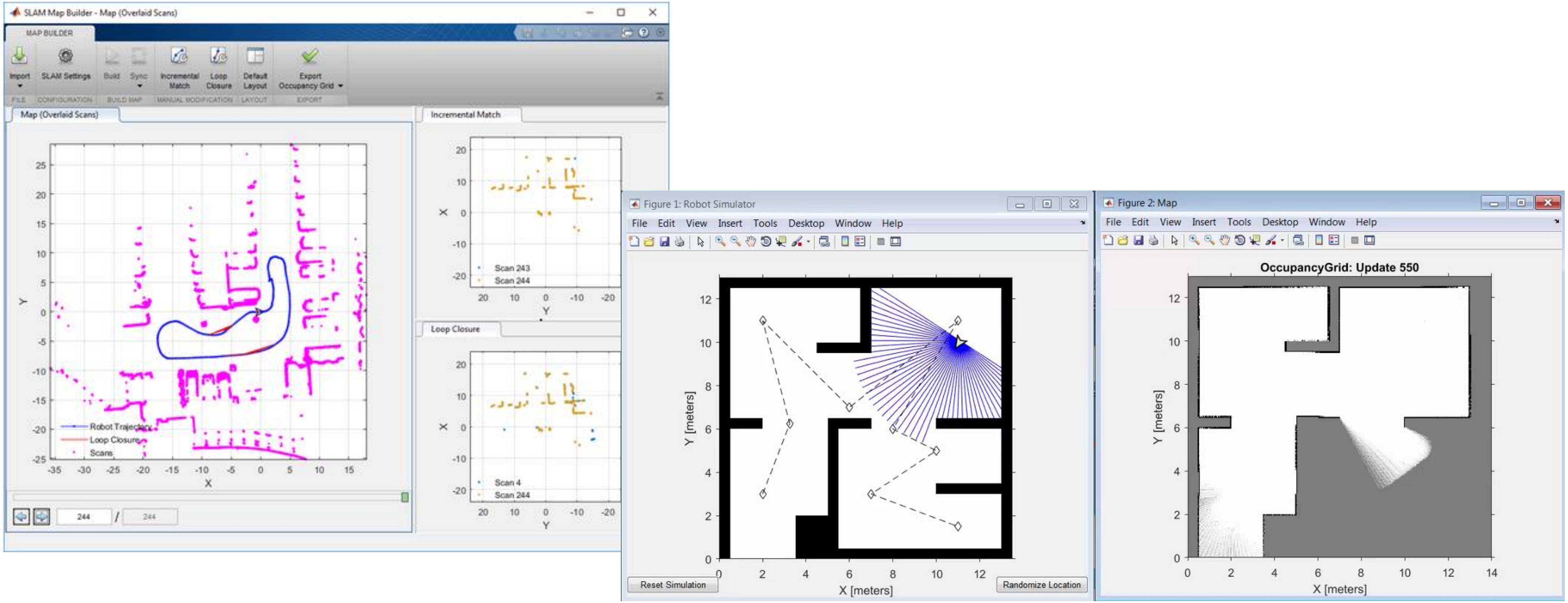
Control and Vehicle Dynamics



Path planning

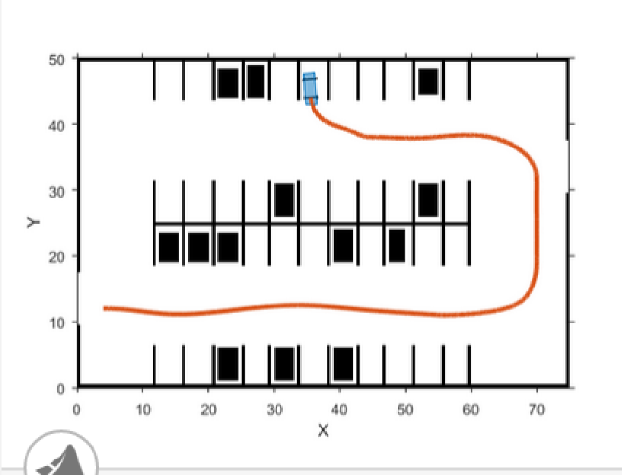


What about when there is no map?

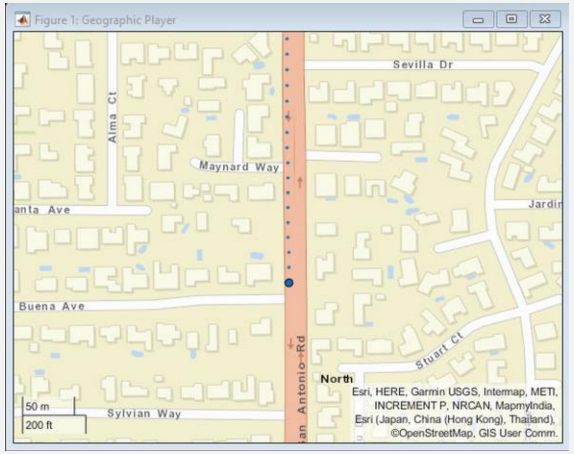


MATLAB has SLAM algorithms which can be used to derive maps during the localisation process

Learn about developing path planning algorithms with these examples



Automated Parking Valet



Animate Sequence of Latitude and Longitude Coordinates

- **Plan path** for automobile given pre-defined map

  Automated Driving

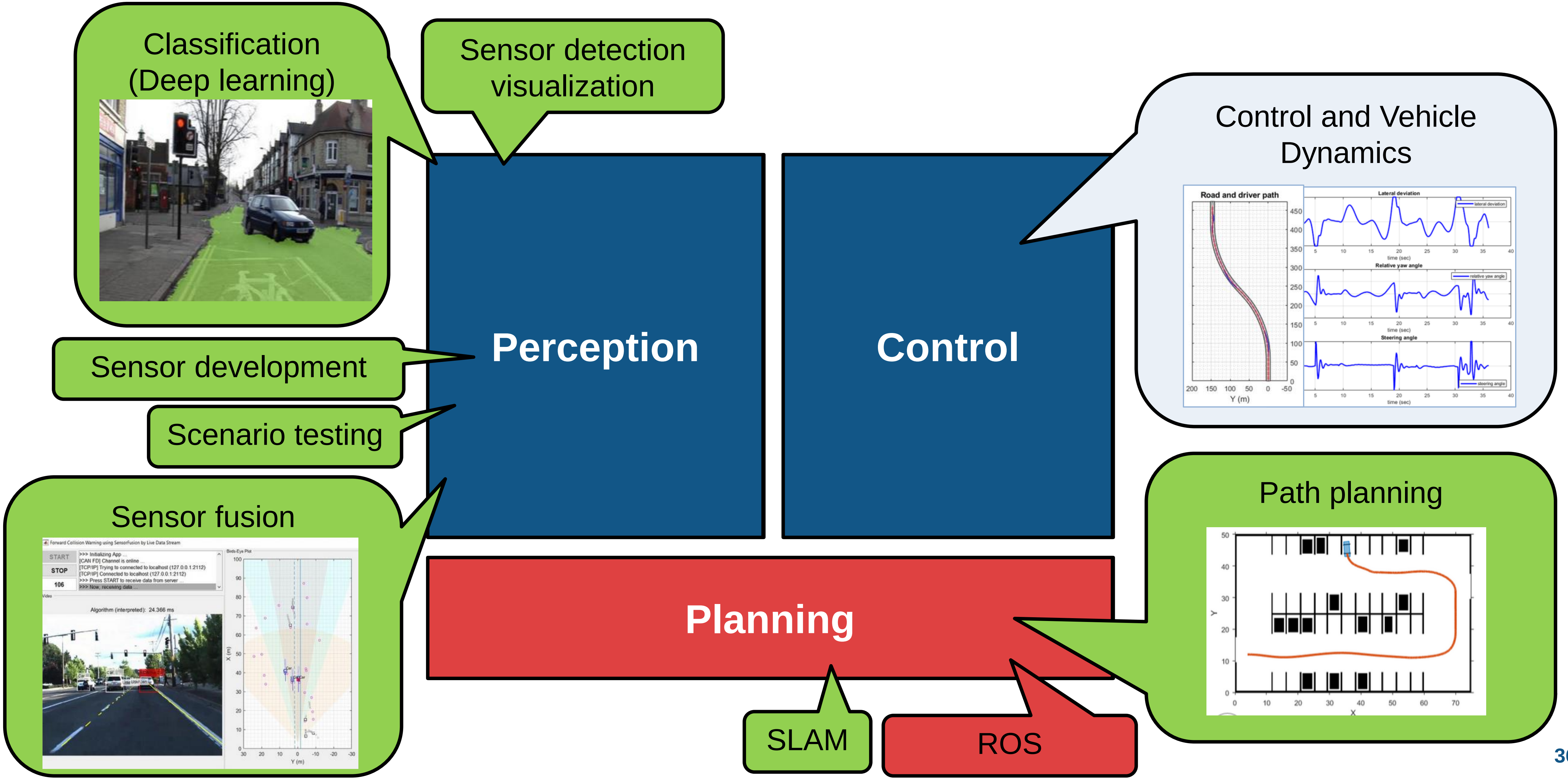
  System Toolbox™

- **Plot map tiles** using World Street Map (Esri)

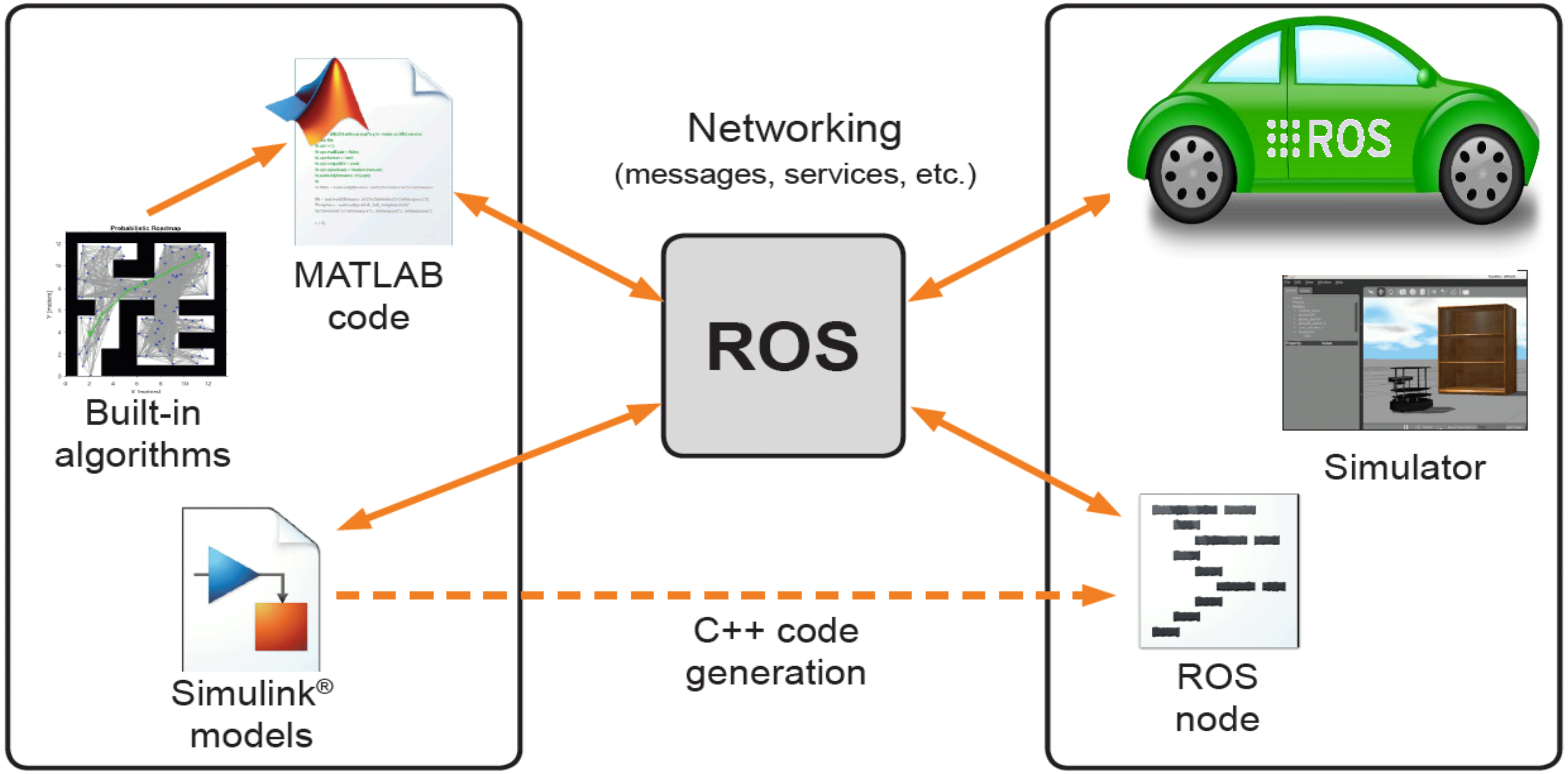
  Automated Driving

  System Toolbox™

How can I make a forward collision warning system?



ROS is important in Automated Driving



How can I make a forward collision warning system?

Classification (Deep learning)

Sensor detection visualization

Sensor development

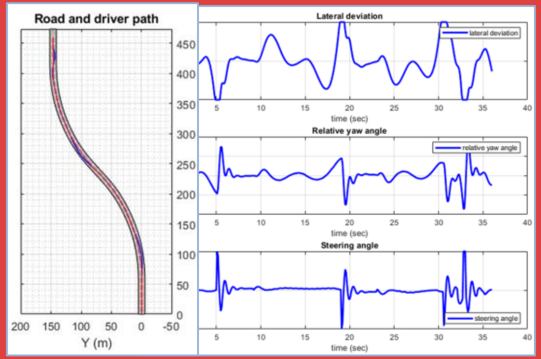
Scenario testing

Sensor fusion

Perception

Control

Control and Vehicle Dynamics

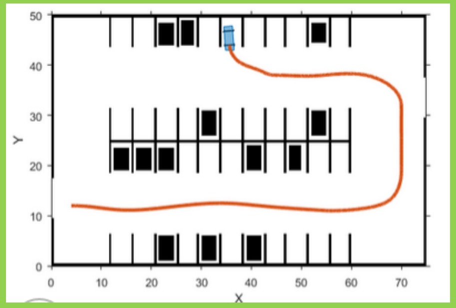


Planning

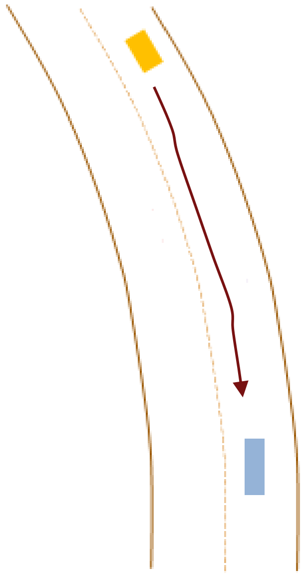
SLAM

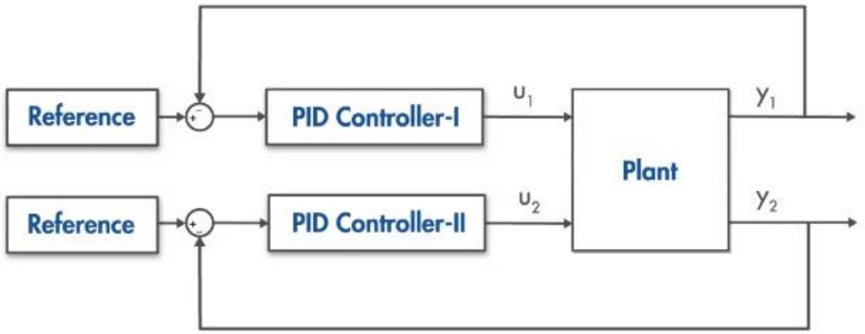
ROS

Path planning

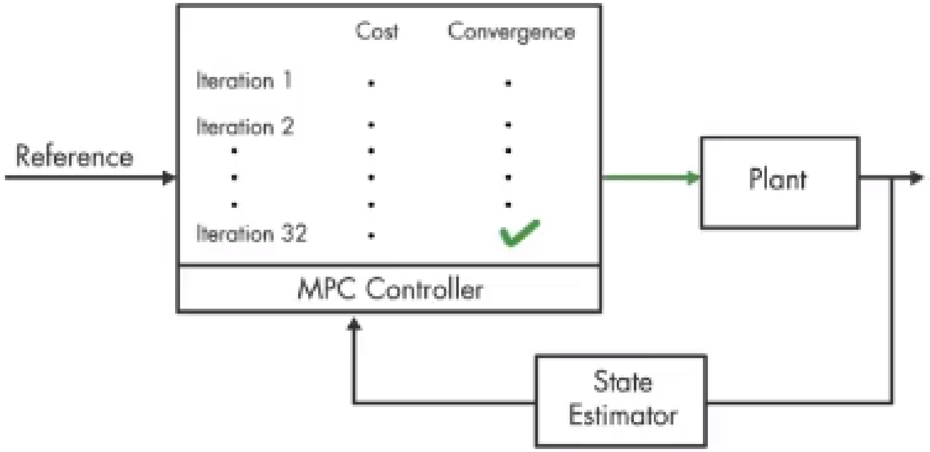


Developing Forward Collision Warning using Model Predictive Control





Traditional Control

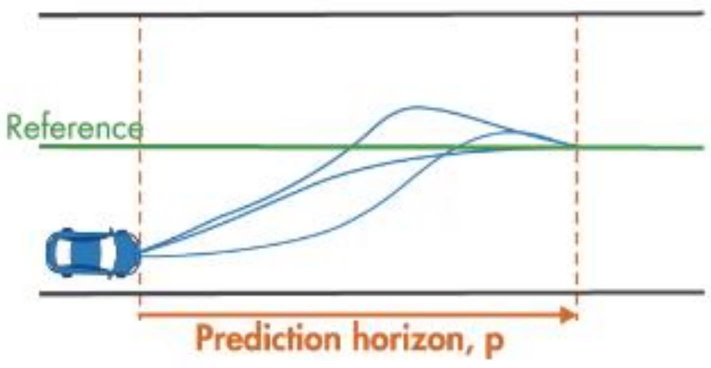


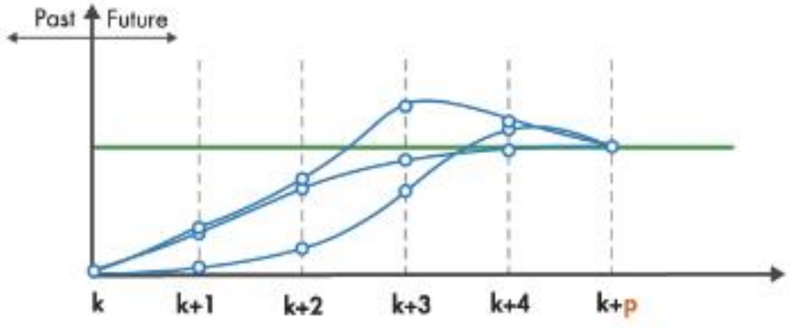
Model Predictive Control

MPC is good for:

- **Complex Nonlinear Systems (e.g. MIMO) –** outputs and non-linearities are all considered together and there are fewer parameters
- **Constraints –** changing hard and soft constraints (e.g. speed) can be considered within the optimisation
- **Preview –** it is forward looking

Model Predictive Control – the basics



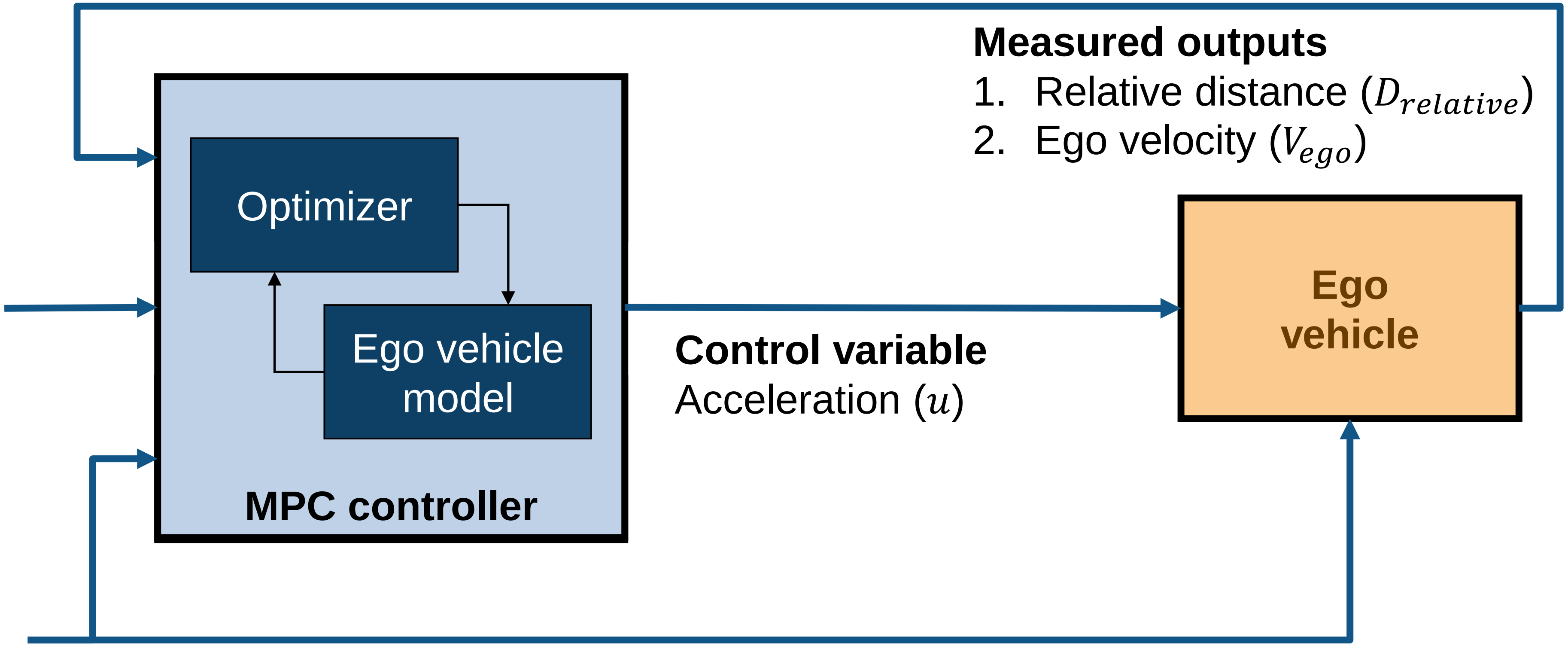


Parameters:
- Period
- Prediction horizon
- Control horizon
- Constraints

Main Goal: Try iterations to get to the final position with the minimum cost

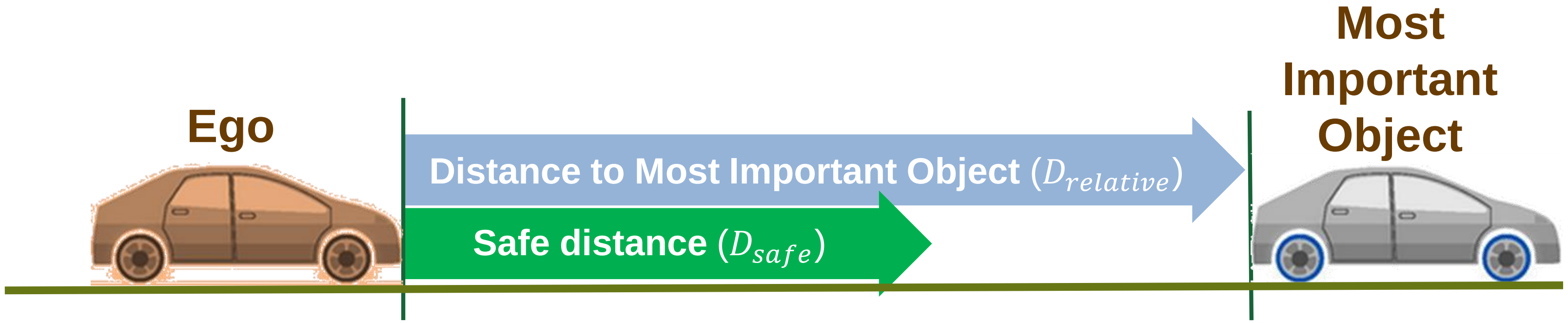
Example: Developing a Cruise Control Algorithm

References
1. Safe distance (D_{safe})
2. Ego velocity set point (V_{set})

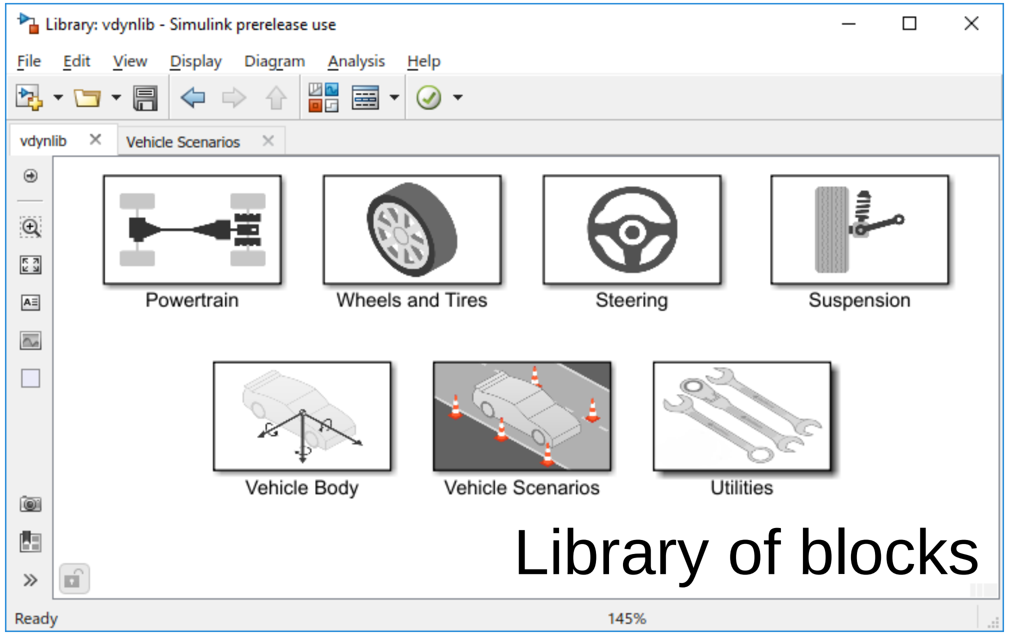


Disturbances
Most Important Object velocity

$$
\begin{aligned}
\text{minimize:} \quad & \left|V_{ego} - V_{set}\right|^2 \\
\text{subject to:} \quad & D_{relative} \geq D_{safe} \\
& u_{min} \leq u \leq u_{max}
\end{aligned}
$$



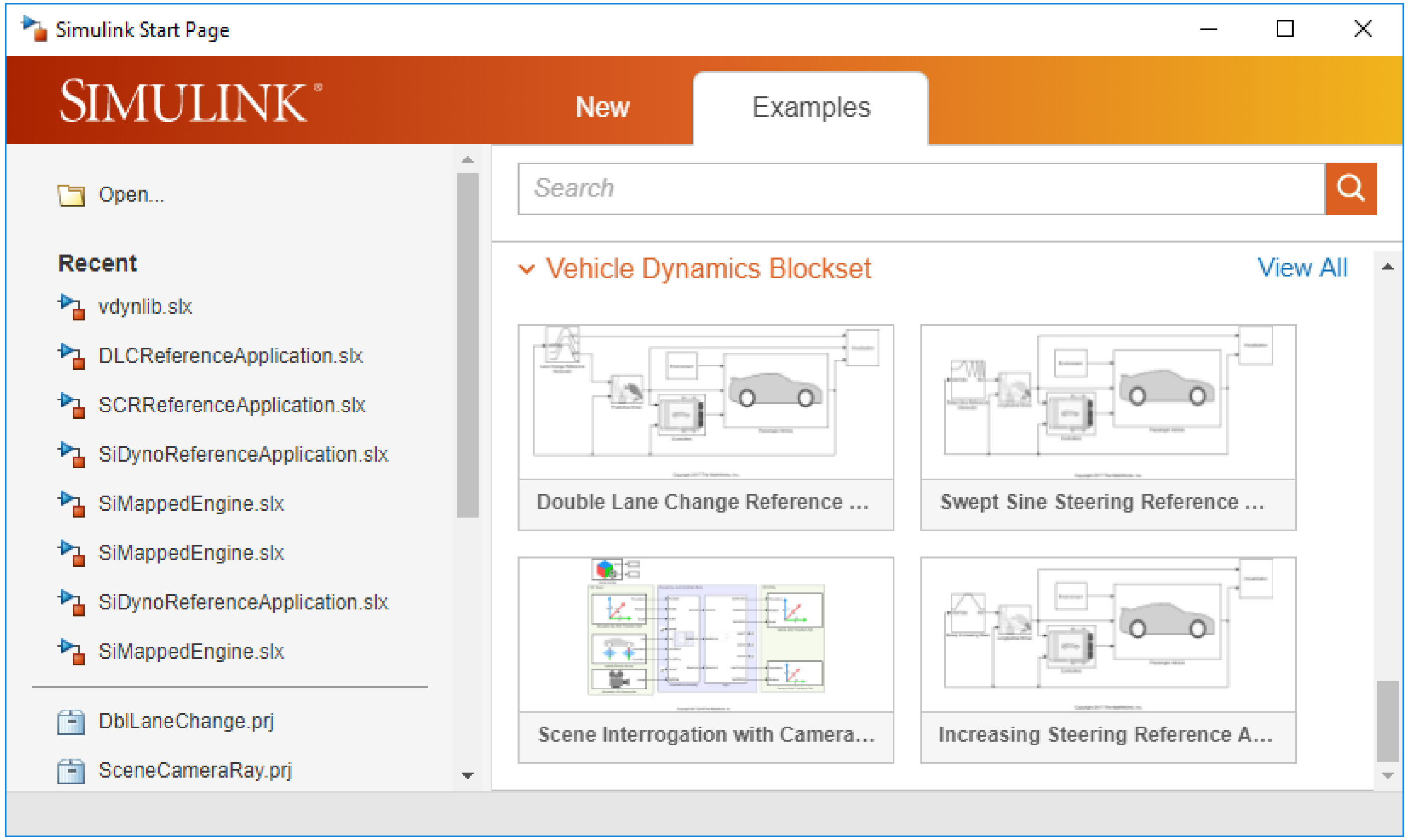
Model Vehicle Dynamics using Vehicle Dynamics Blockset



Library of blocks

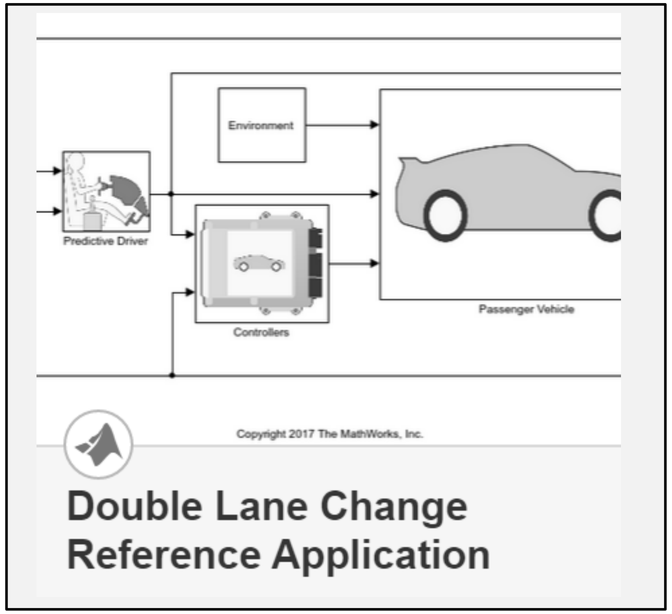
Game engine

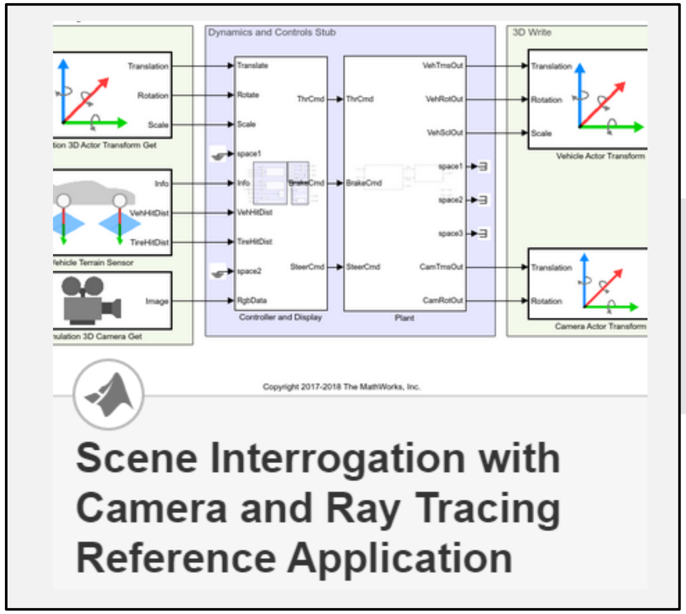
Pre-built reference applications

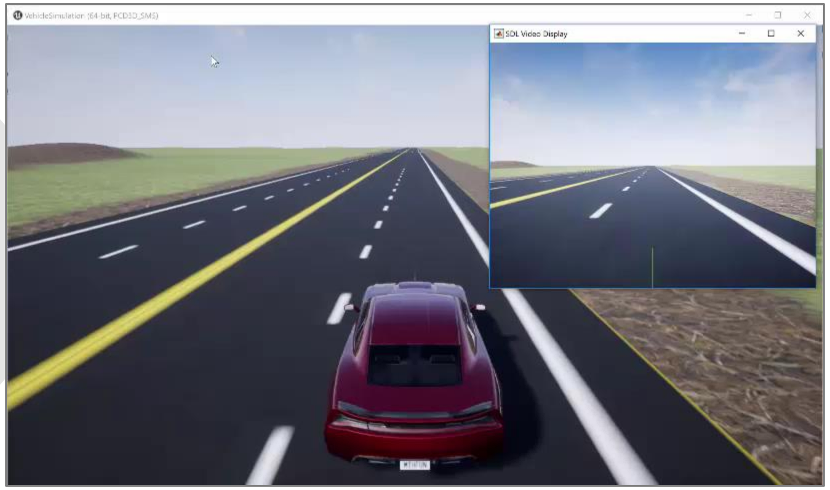


Chassis Controls Study: Braking Test with ABS

Model Vehicle Dynamics using Vehicle Dynamics Blockset







- **Simulate vehicle dynamics** for closed loop design

  Vehicle Dynamics Blockset™

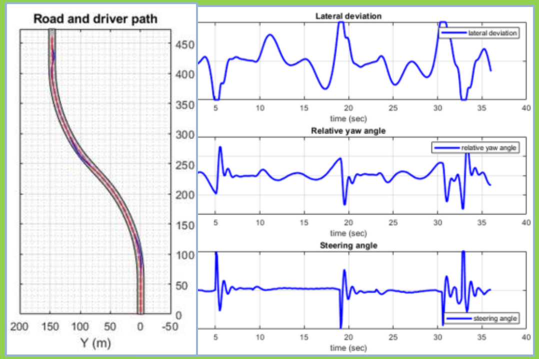
- **Co-simulate with Unreal Engine** to set actor positions and get camera image

  Vehicle Dynamics Blockset™

Classification (Deep learning)

Sensor detection visualization

Control and Vehicle Dynamics



Perception

Control

Sensor development

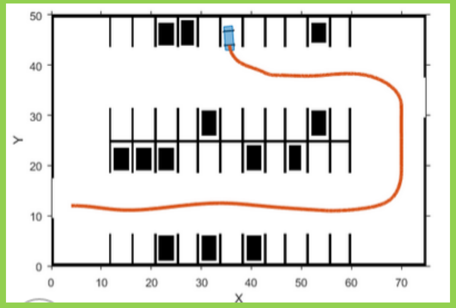
Scenario testing

Questions?

marc.willerton@mathworks.co.uk

Sensor fusion

Path planning



Planning

SLAM

ROS