

# Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic



**Gabriel Choi**, General Motors LLC  
**Seo-Wook Park**, MathWorks



# Contents

- Super Cruise – History and Future
- Lane Centering with Super Cruise
- Motivation
- Create virtual driving scenario from recorded data
- Simulate closed-loop model for lane centering system
- Conclusion
- Q & A

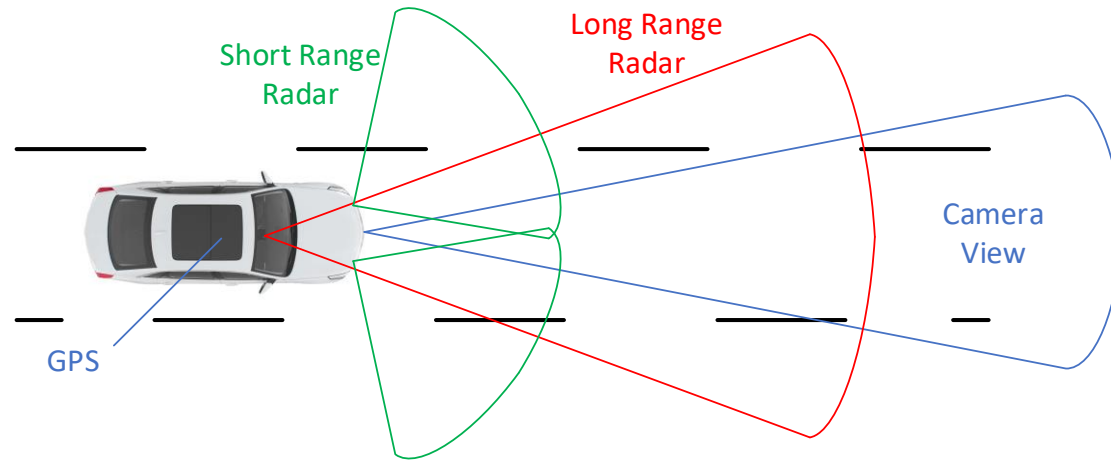
# Super Cruise – History and Future

- Debuted in 2017 with CT6 Sedan
  - Lane Centering in addition to Full Speed Range Adaptive Cruise Control
  - Uses High-Definition Map and Front Camera to detect Lane Marks
- Automated Lane Change for 2021 Cadillac CT4 / CT5 / Escalade
  - Lane Change following Driver Request
  - Able to accelerate and decelerate slightly to search gap to change in
- Eventually will be expanded to many name plates





# Lane Centering with Super Cruise on Cadillac CT6

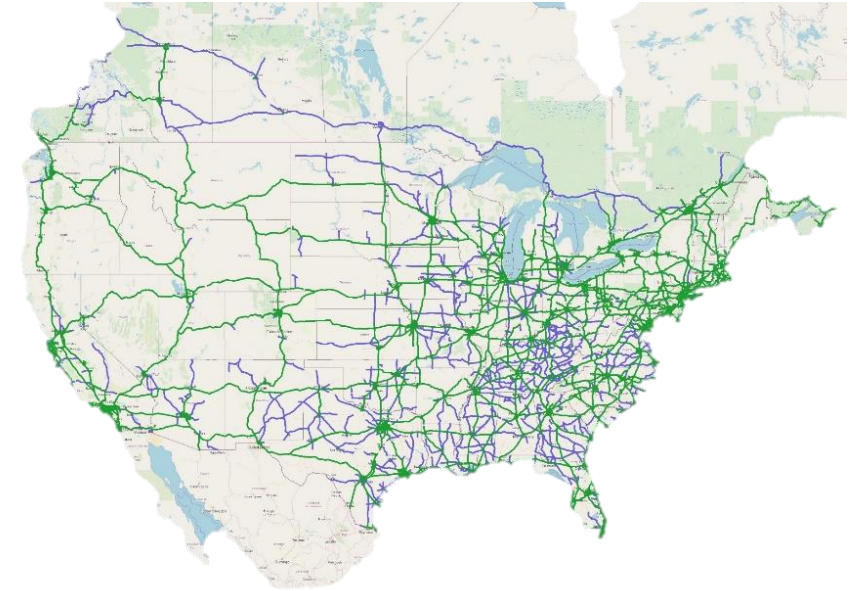


- **Sensors**
  - Pre-Scanned High Definition Map
  - Map matching with GPS
  - Camera
  - Long Range Radar
  - Short Range Radars
- **Actuation**
  - Electric Power Steering
- **Driver Monitoring System for Safety**
  - Infra-red Face Recognition
  - Steering Wheel Touch Sensor
  - Chime and Vibration Seat



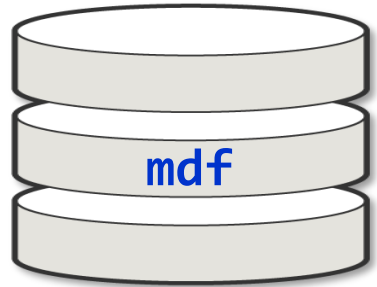
# System validation for driving automation system

- Pains
  - Big data size from “tens of thousands of miles” test drive
    - Time consuming for data analysis
  - Not easy to reproduce a real-world traffic situation with closed-loop simulation
    - Hazardous test scenarios
    - Unwanted system behavior
- Virtual driving scenario from recorded data
  - Reduce development time
  - Enable closed-loop simulation to identify the root causes for unwanted system behavior



→ How to create driving scenario from recorded data?

# Create virtual driving scenario from recorded data



MDF(Measurement Data Format)



```

% Create an object to access the MF4 file.
mdfObj = mdf('RecordedData.mf4');
  
```

ego						sensor				
3905x5 timetable						1953x3 timetable				
	Time	1 Latitude	2 Longitude	3 Vx	4 Vy		Time	1 lane	2 vision	3 LRR
1	1.7871 sec	42.3775	-83.0382	27.4446	0	1	1.7871 sec	1x1 struct	1x1 struct	1x1 struct
2	1.8071 sec	42.3775	-83.0382	27.4498	0	2	1.8271 sec	1x1 struct	1x1 struct	1x1 struct
3	1.8271 sec	42.3775	-83.0382	27.4571	0	3	1.8671 sec	1x1 struct	1x1 struct	1x1 struct
4	1.8471 sec	42.3775	-83.0382	27.4678	0	4	1.9071 sec	1x1 struct	1x1 struct	1x1 struct
5	1.8671 sec	42.3775	-83.0382	27.4745	0	5	1.9471 sec	1x1 struct	1x1 struct	1x1 struct
6	1.8871 sec	42.3775	-83.0382	27.4800	0	6	1.9871 sec	1x1 struct	1x1 struct	1x1 struct
7	1.9071 sec	42.3775	-83.0382	27.4897	0	7	2.0271 sec	1x1 struct	1x1 struct	1x1 struct
8	1.9271 sec	42.3775	-83.0382	27.5004	0	8	2.0671 sec	1x1 struct	1x1 struct	1x1 struct
9	1.9471 sec	42.3775	-83.0382	27.5070	0	9	2.1071 sec	1x1 struct	1x1 struct	1x1 struct
						10	2.1471 sec	1x1 struct	1x1 struct	1x1 struct

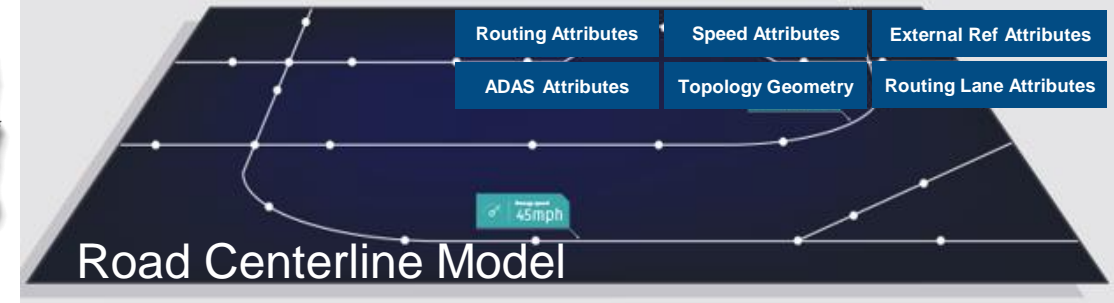
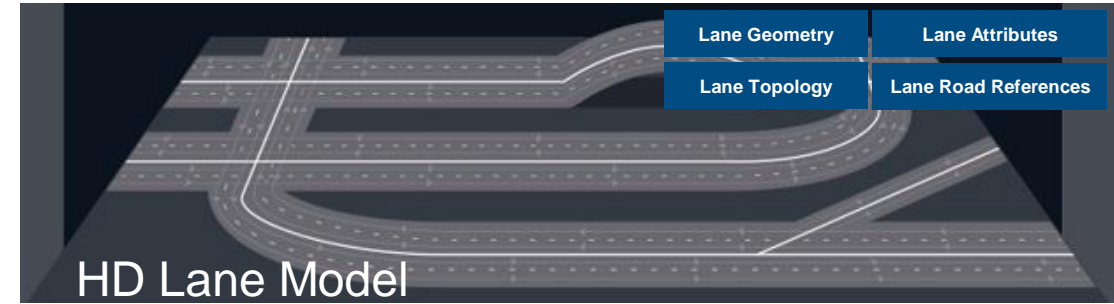
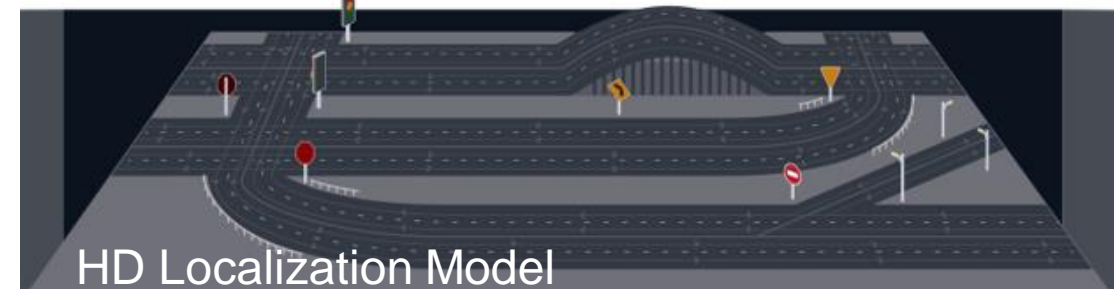
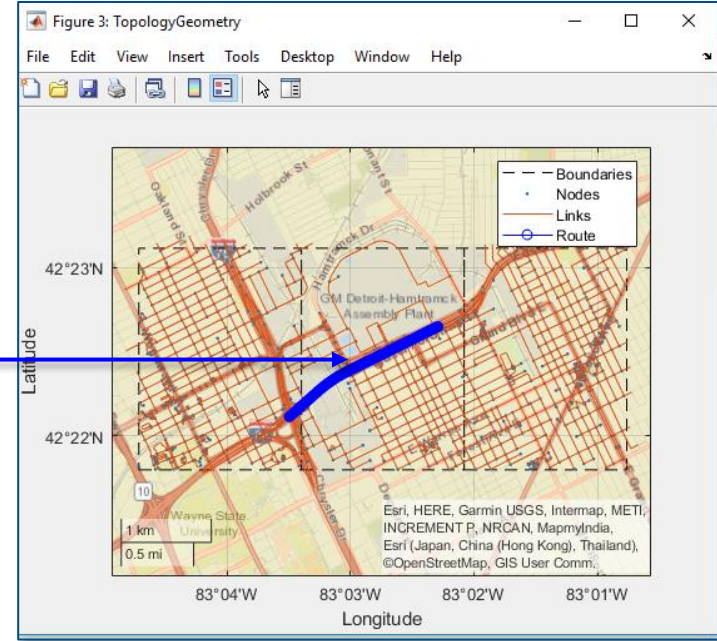
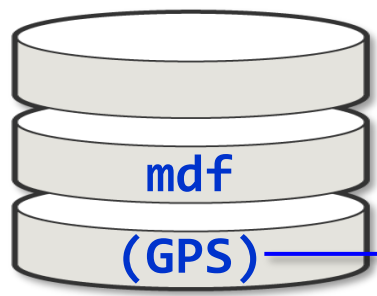
GPS

Vehicle Data

On-board sensors



# Create virtual driving scenario from recorded data



```

HDLMreader = hereHDLMreader(latitude, longitude);
topologyGeometry = HDLMreader.read("TopologyGeometry");
plot(topologyGeometry);
    
```

High-Definition Map with Tiled Layers

# Create virtual driving scenario from recorded data

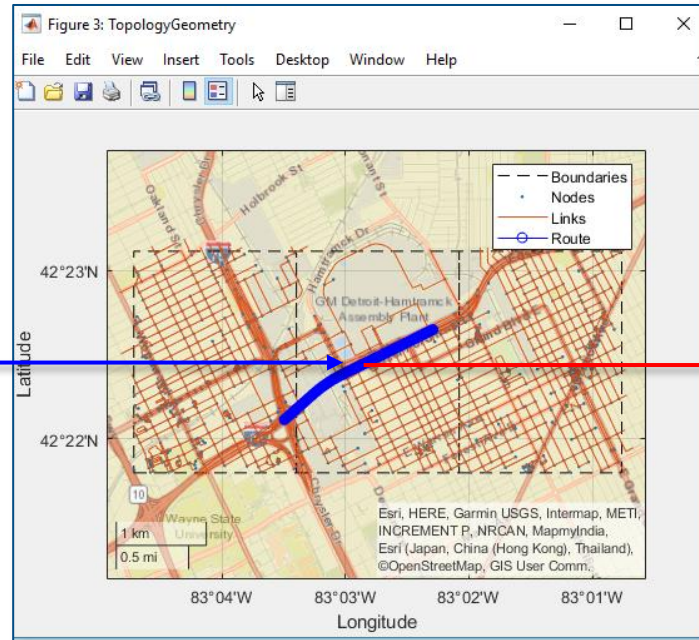
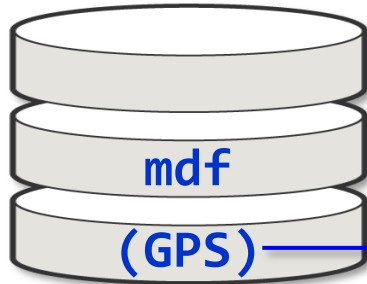
Record and  
select data

Reconstruct  
road network

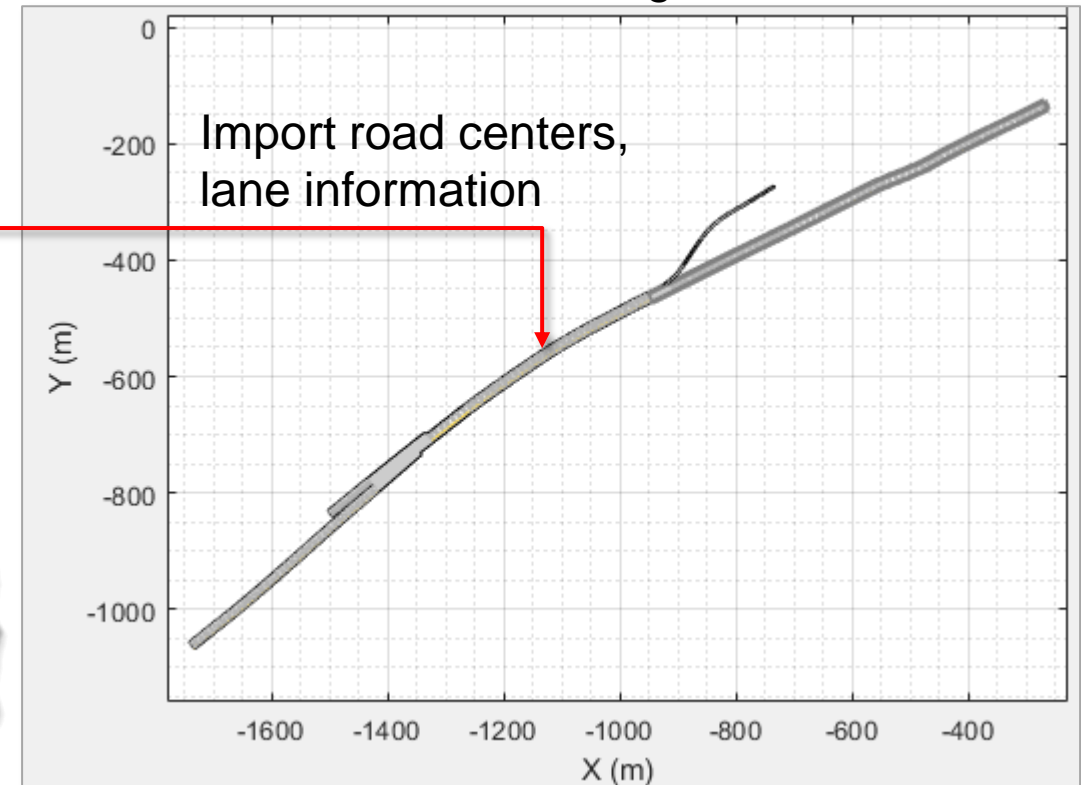
Localize ego  
trajectory

Reconstruct  
target vehicles

Compare with  
recorded video



Create driving scenario



```
HDLMreader = hereHDLMreader(latitude,longitude);
topologyGeometry = HDLMreader.read("TopologyGeometry");
plot(topologyGeometry);
```



# Create virtual driving scenario from recorded data

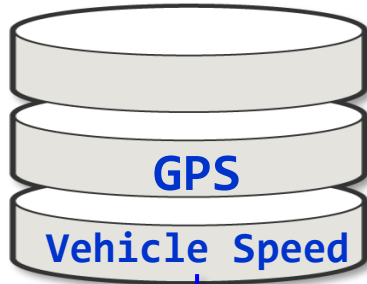
Record and  
select data

Reconstruct  
road network

Localize ego  
trajectory

Reconstruct  
target vehicles

Compare with  
recorded video



Driving Scenario Designer - GM\_session\_original - Actors

DESIGNER

FILE SCENARIO SENSORS SIMULATE VIEW EXPORT

Roads Actors Scenario Canvas Ego-Centric View

1: original (ego vehicle)

Name: original Set As Ego

Class: Car

3D Display Type: Sedan

Actor Properties

Radar Cross Section

Trajectory

Wait Time at Waypoints

Constant Speed (m/s):

Waypoints

	x (m)	y (m)
1	-1.8205	0.6406
2	-3.0239	3.1114
3	-4.2581	5.5821
4	-5.3380	8.0529
5	-6.5414	10.5237
6	-7.7448	12.9945
7	-8.9790	15.4881
8	-10.2132	17.9589

trajectory(egoCar, waypoints, speeds)

X (m)

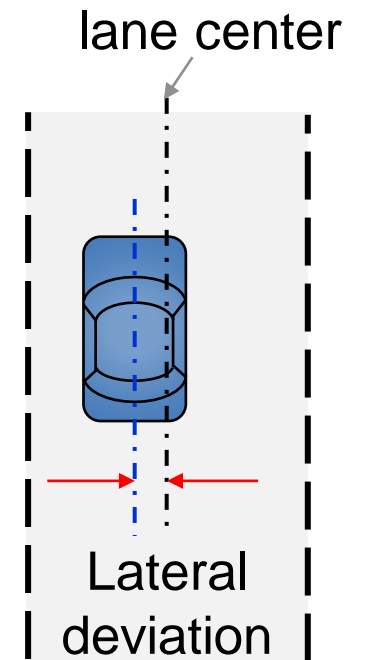
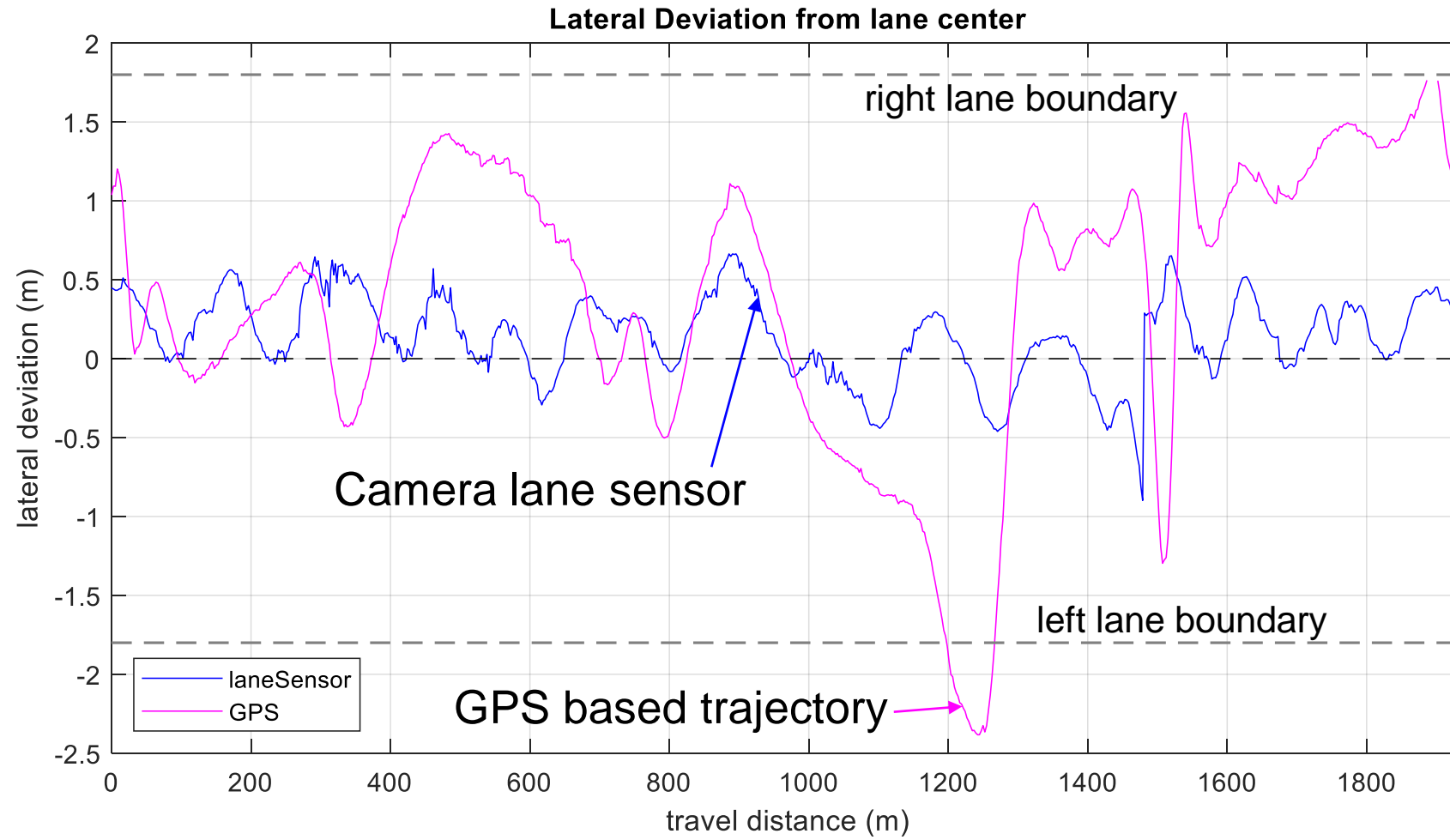
Y (m)

T=0s

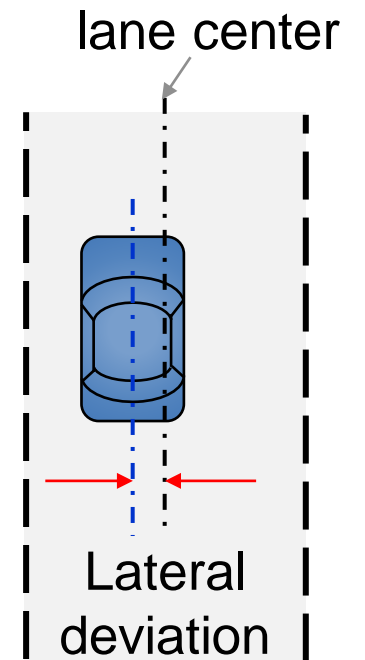
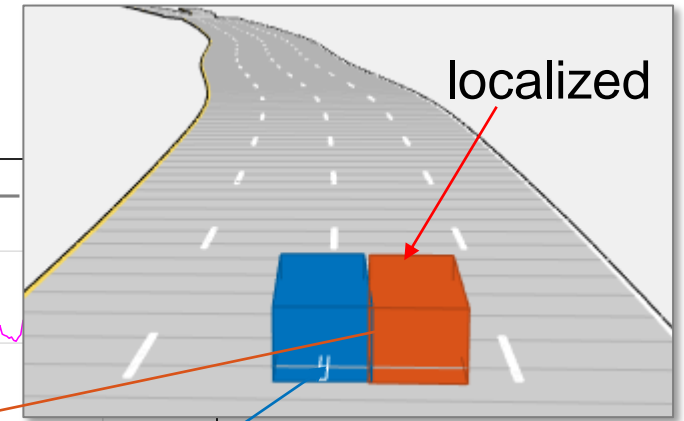
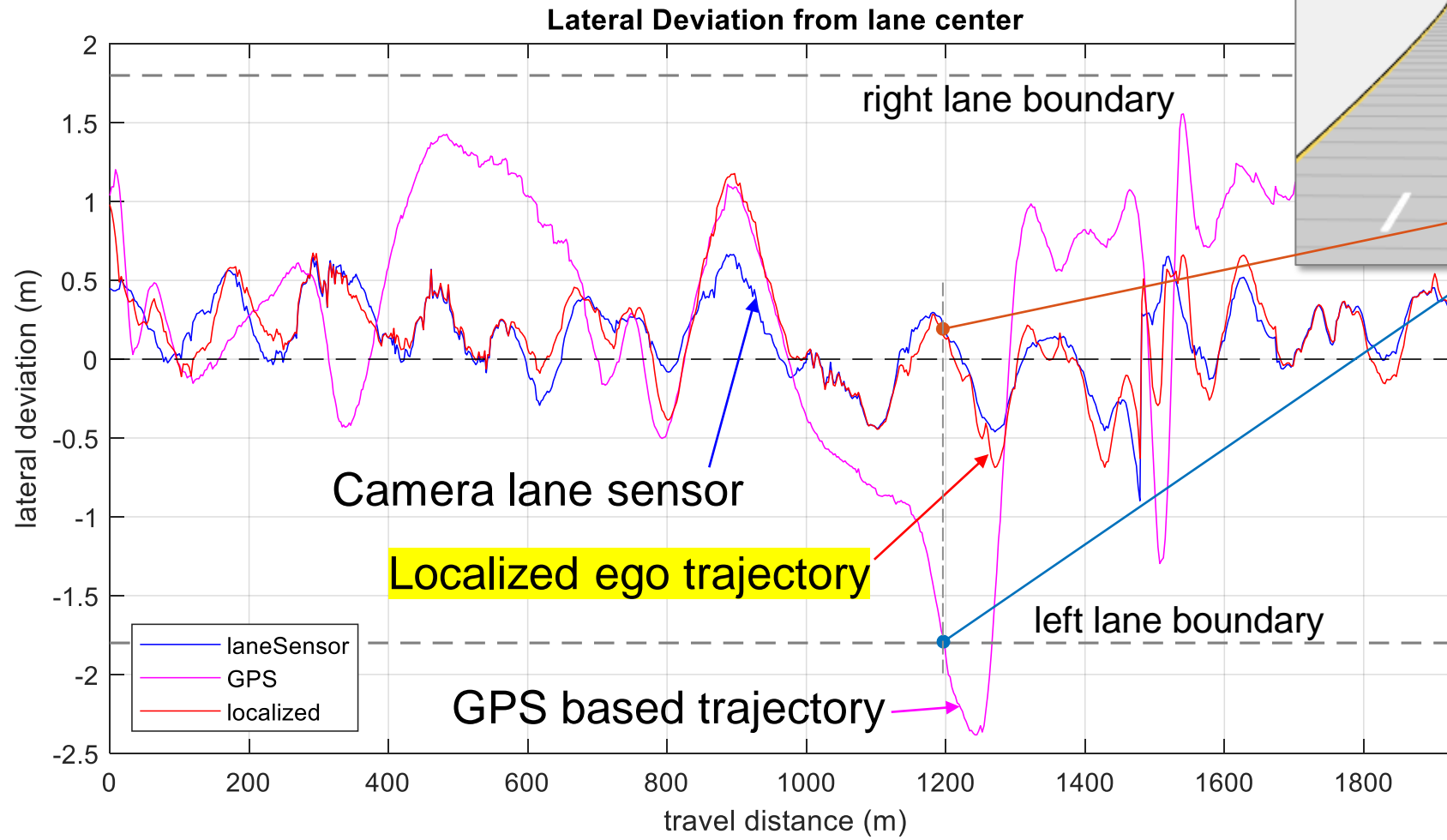
Road interactions disabled

Ego car

# Lateral deviation of ego vehicle from lane center



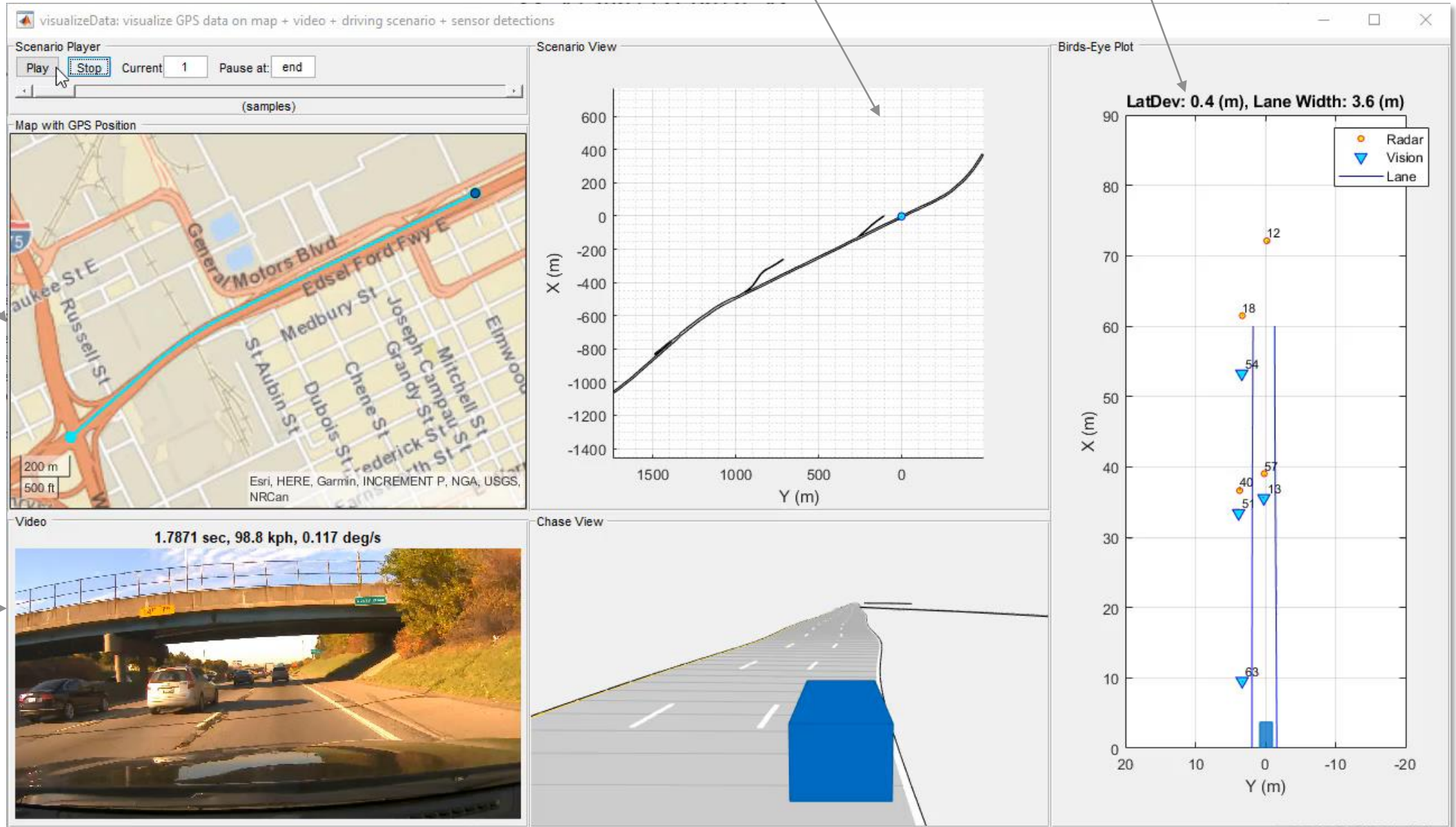
# Localized ego trajectory by lane sensor data



# Visualize data

Driving scenario view  
**drivingScenario**

Bird's-eye plot with sensor detections  
**birdsEyePlot**



Map with GPS position  
**geoplayer**

Video with time stamp, ego speed & yaw rate  
**VideoReader**  
**imshow**



# Create virtual driving scenario from recorded data

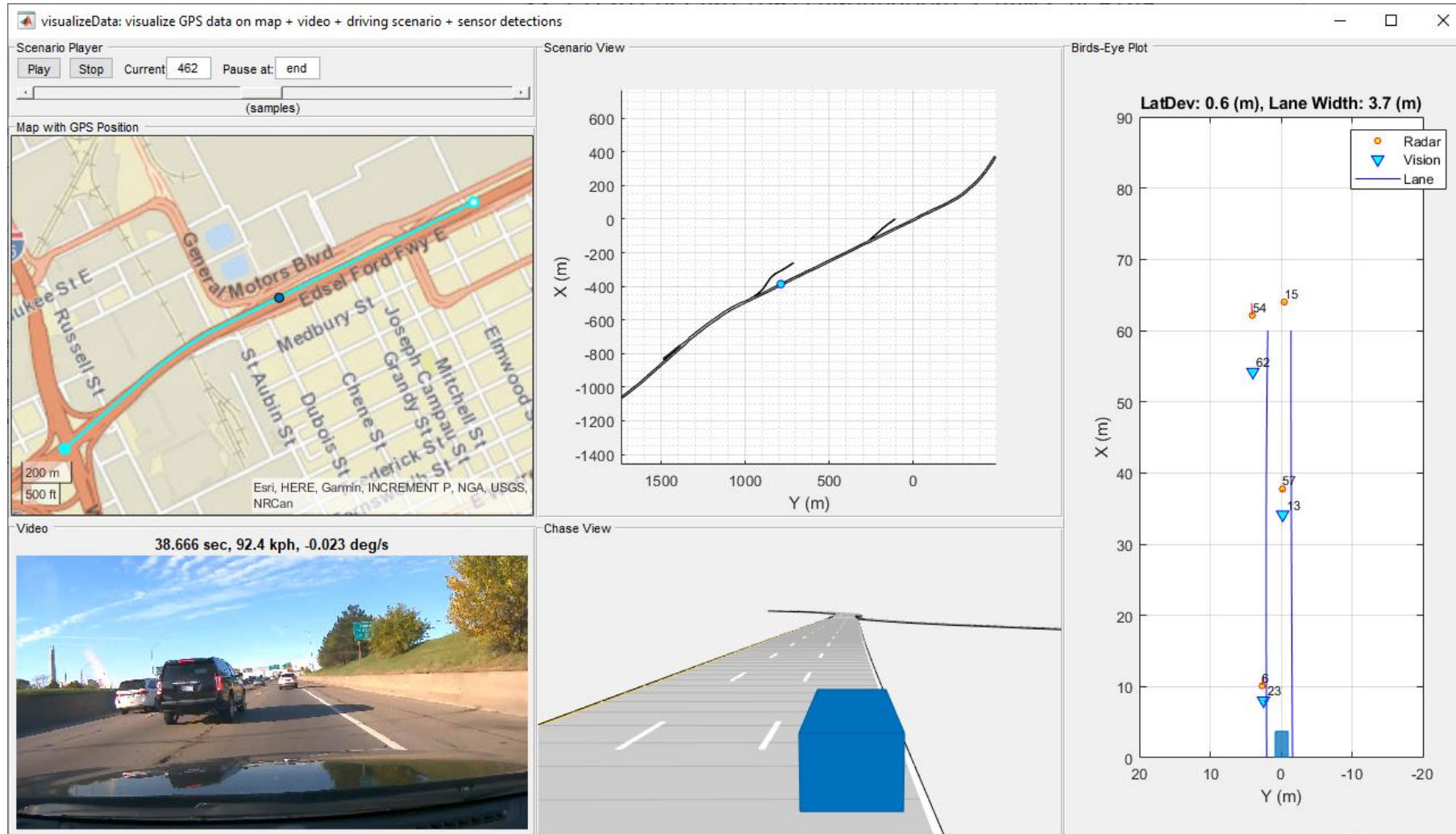
Record and  
select data

Reconstruct  
road network

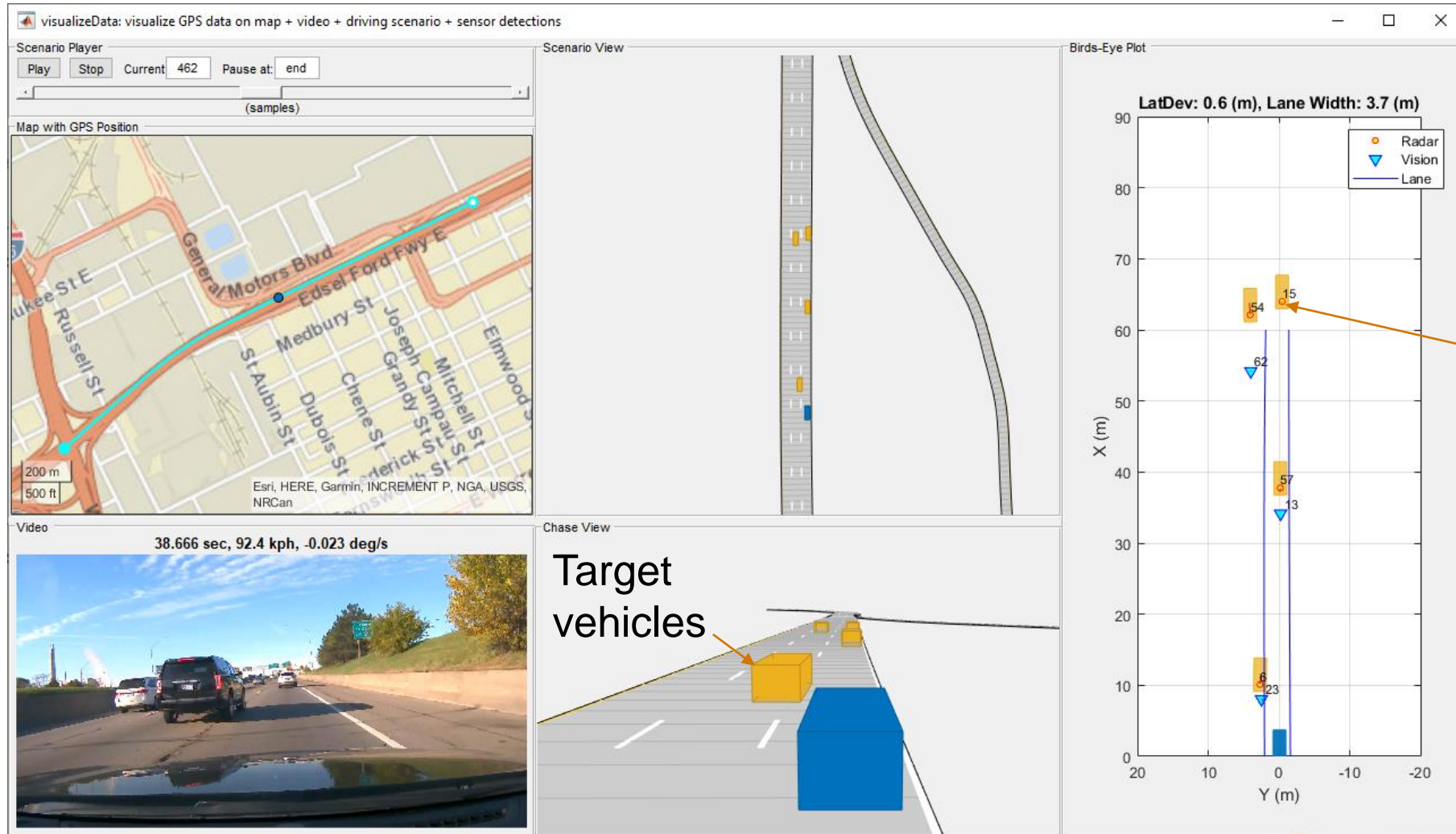
Localize ego  
trajectory

Reconstruct  
target vehicles

Compare with  
recorded video



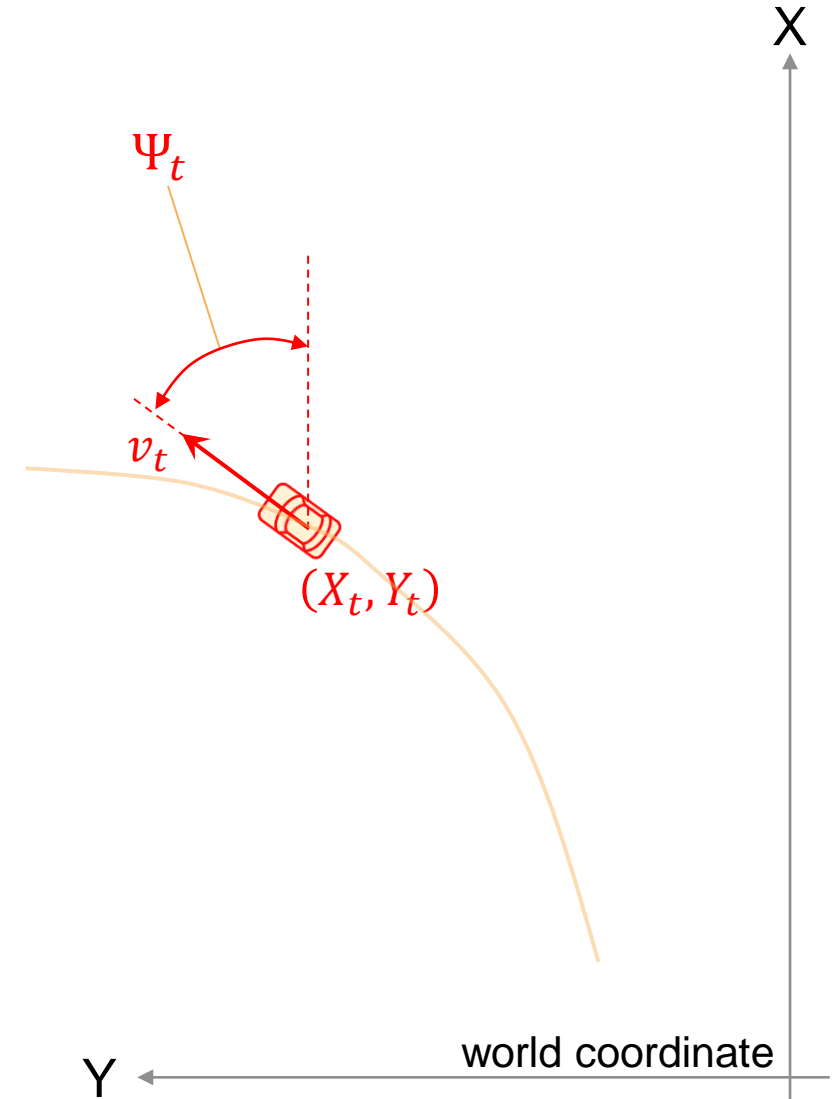
# Create virtual driving scenario from recorded data



Radar  
 Detections  
 Report  
 targets only  
 in ego and  
 neighbor lanes

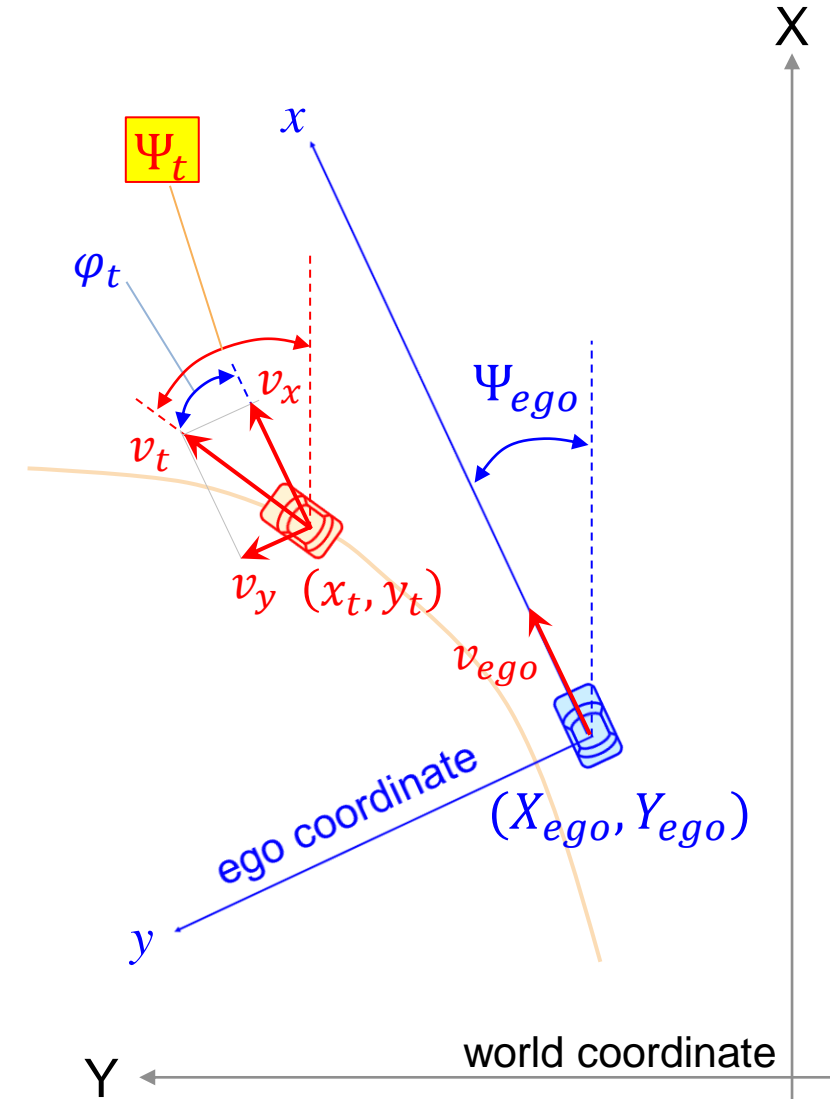
# Target vehicle trajectory

- Target vehicle trajectory is defined by a series of actor poses
- Actor poses consist of
  - Position  $(X_t, Y_t)$
  - Velocity  $v_t$
  - Orientation  $\Psi_t$in world coordinate



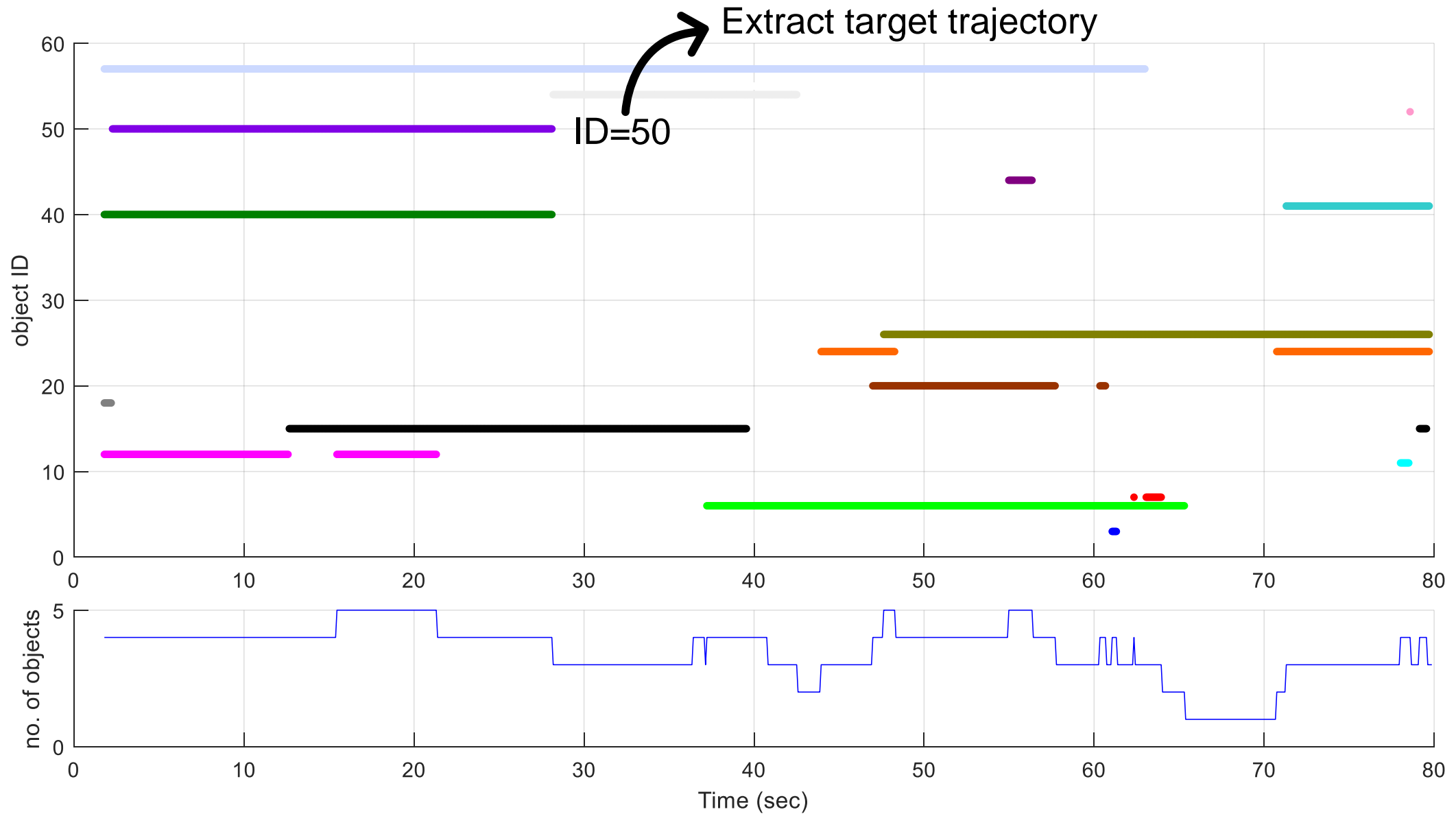
# Estimate heading angle of target vehicle

- Radar detections provides target position and velocity in ego coordinate
  - $x_t, y_t, v_x, v_y, (\varphi_t?) \leftarrow \text{radar detections}$
- Estimate heading angle of target vehicle
  - Heading angle in ego coordinate
    - $\varphi_t = \tan^{-1} \left( \frac{v_y}{v_x} \right)$
  - Heading angle in world coordinate
    - $\Psi_{target} = \varphi_t + \Psi_{ego}$
- Target position in world coordinate
  - $(X_t, Y_t) = (X_{ego}, Y_{ego}) + \mathbb{R}(\Psi_{ego}) \cdot (x_t, y_t)^T$

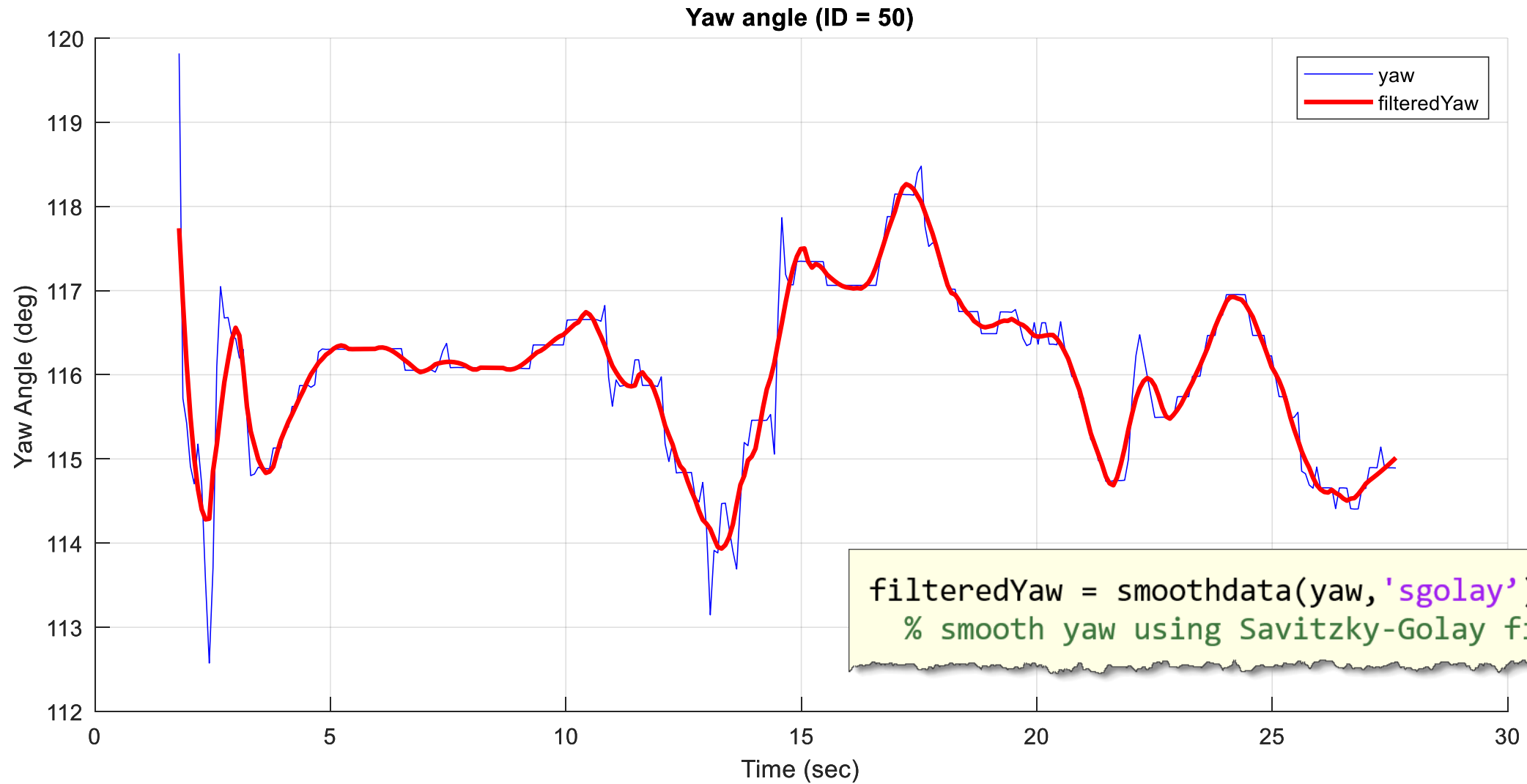




# Extract target trajectory from radar detections



# Example of estimated yaw angle



# Create virtual driving scenario from recorded data

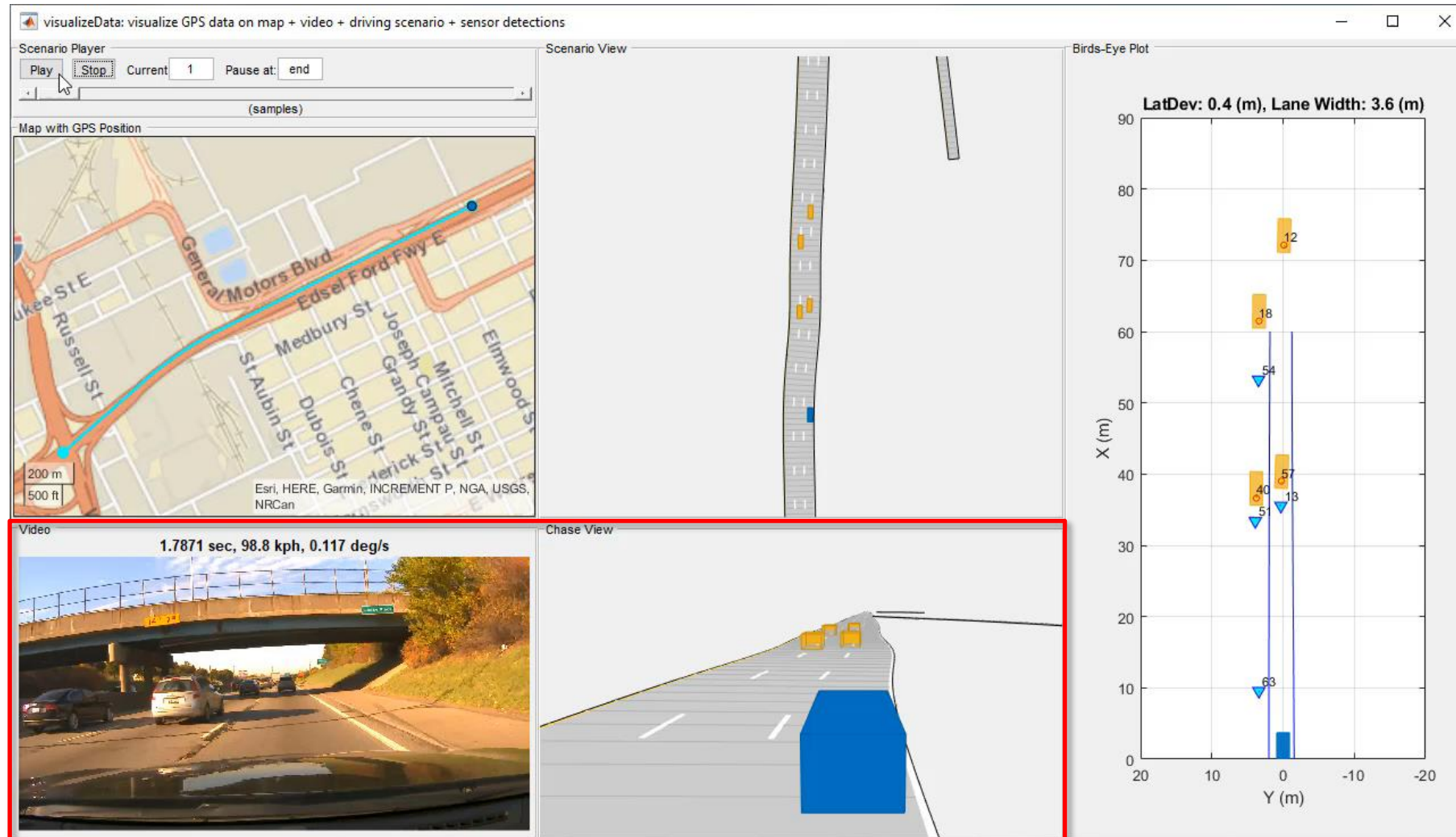
Record and  
select data

Reconstruct  
road network

Localize ego  
trajectory

Reconstruct  
target vehicles

Compare with  
recorded video



Report radar  
detections  
in ego and  
neighbor lanes

# Synthesize sensors with virtual driving scenario

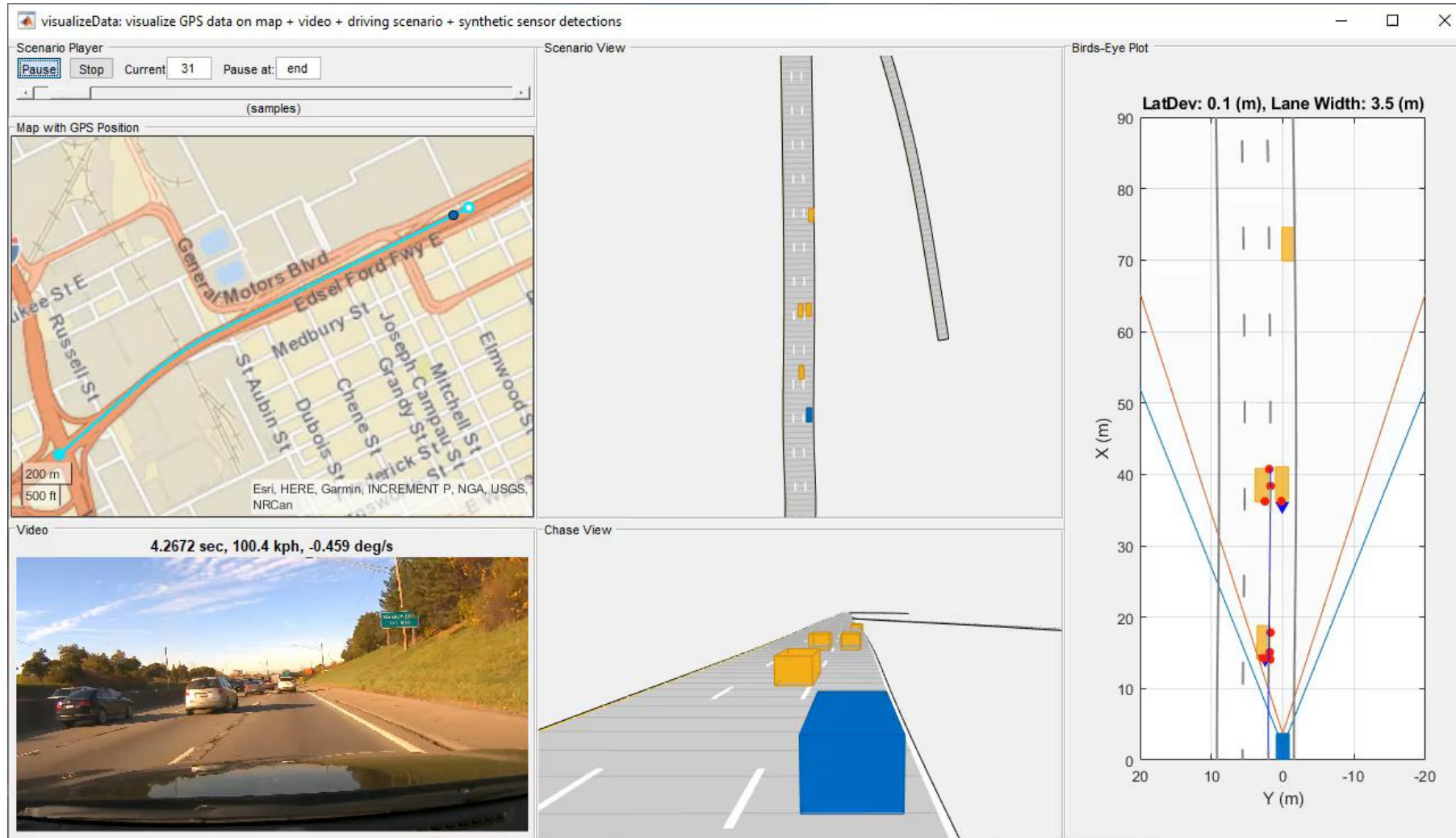
Record and  
select data

Reconstruct  
road network

Localize ego  
trajectory

Reconstruct  
target vehicles

Compare with  
recorded video



Synthesize  
sensors

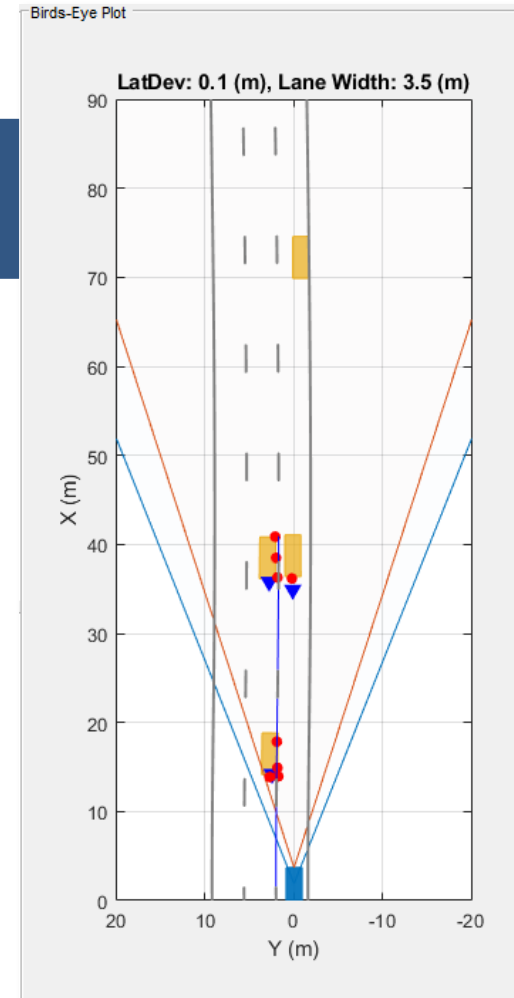
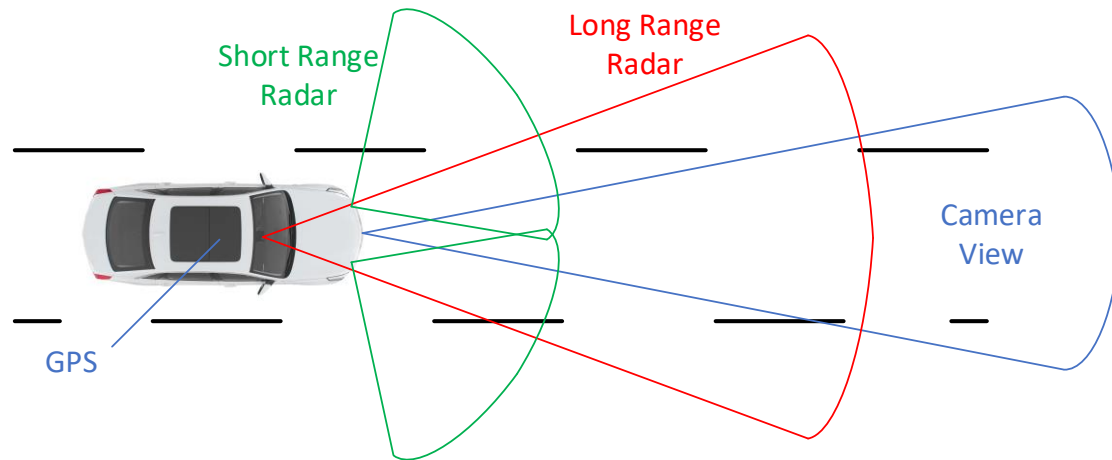


# Integrate driving scenario + sensors with a close-loop system

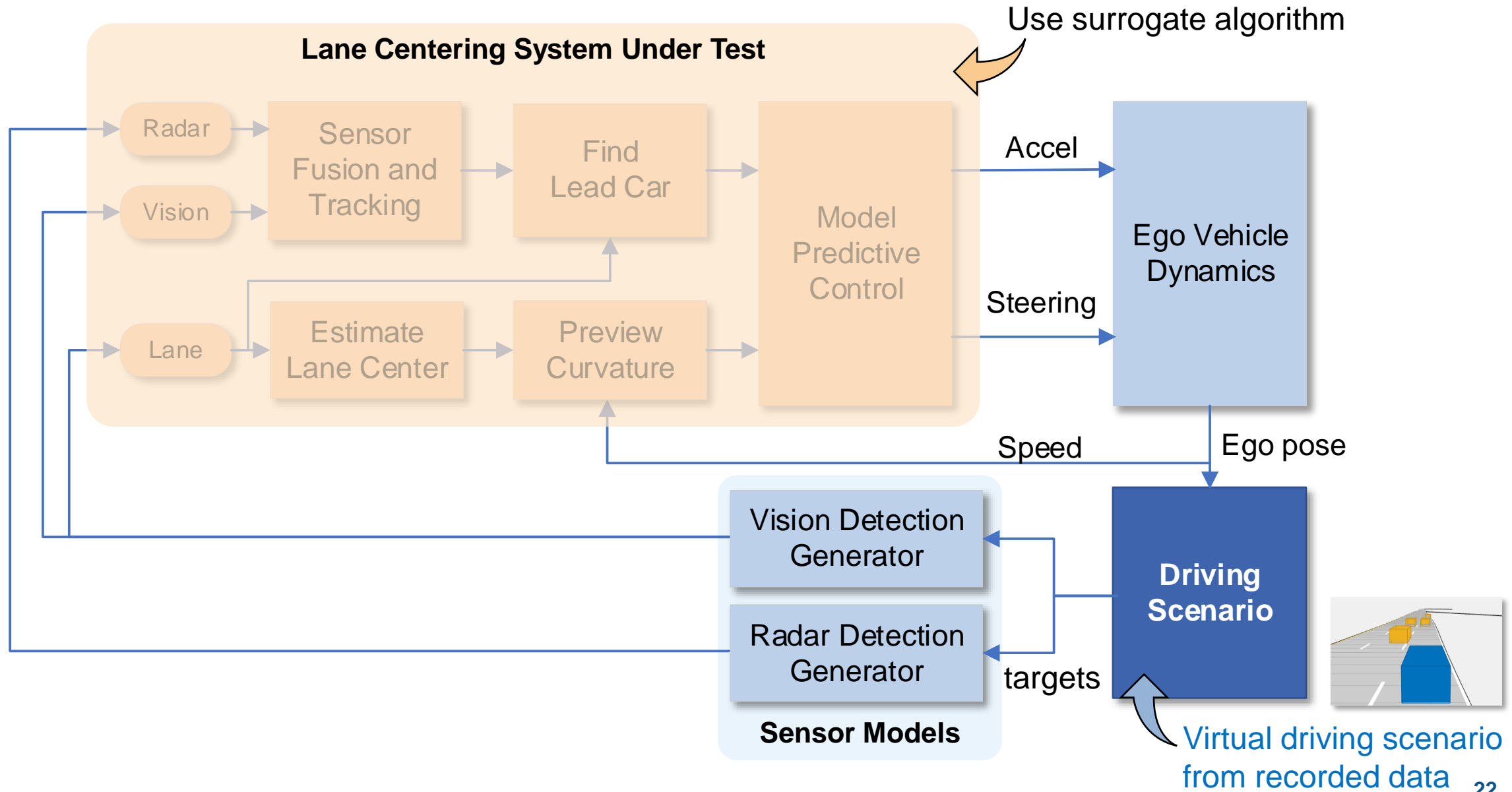


Integrate **driving scenario + sensor models** with a close-loop system

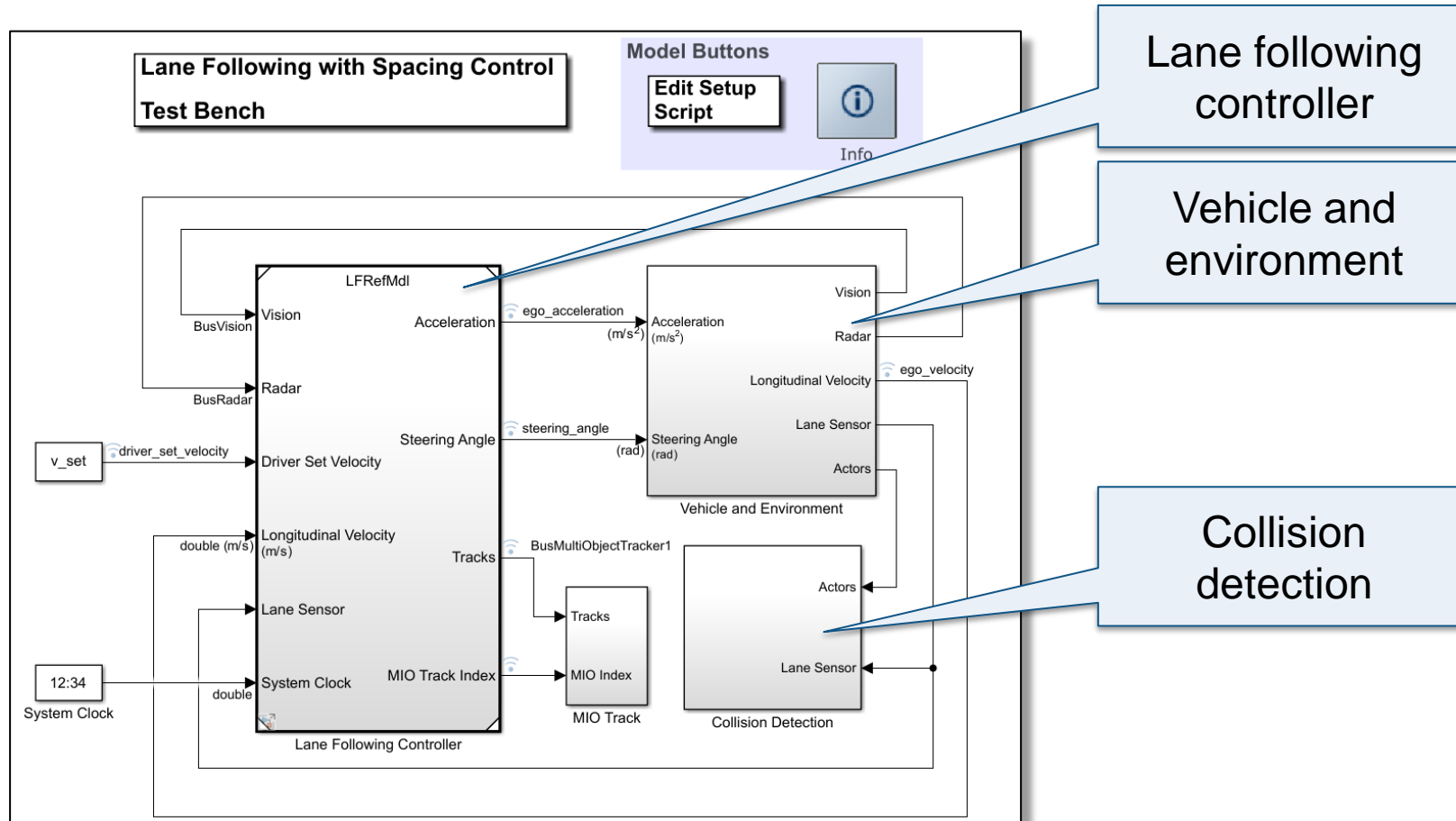
Synthesize sensors



# Closed-loop system for lane centering



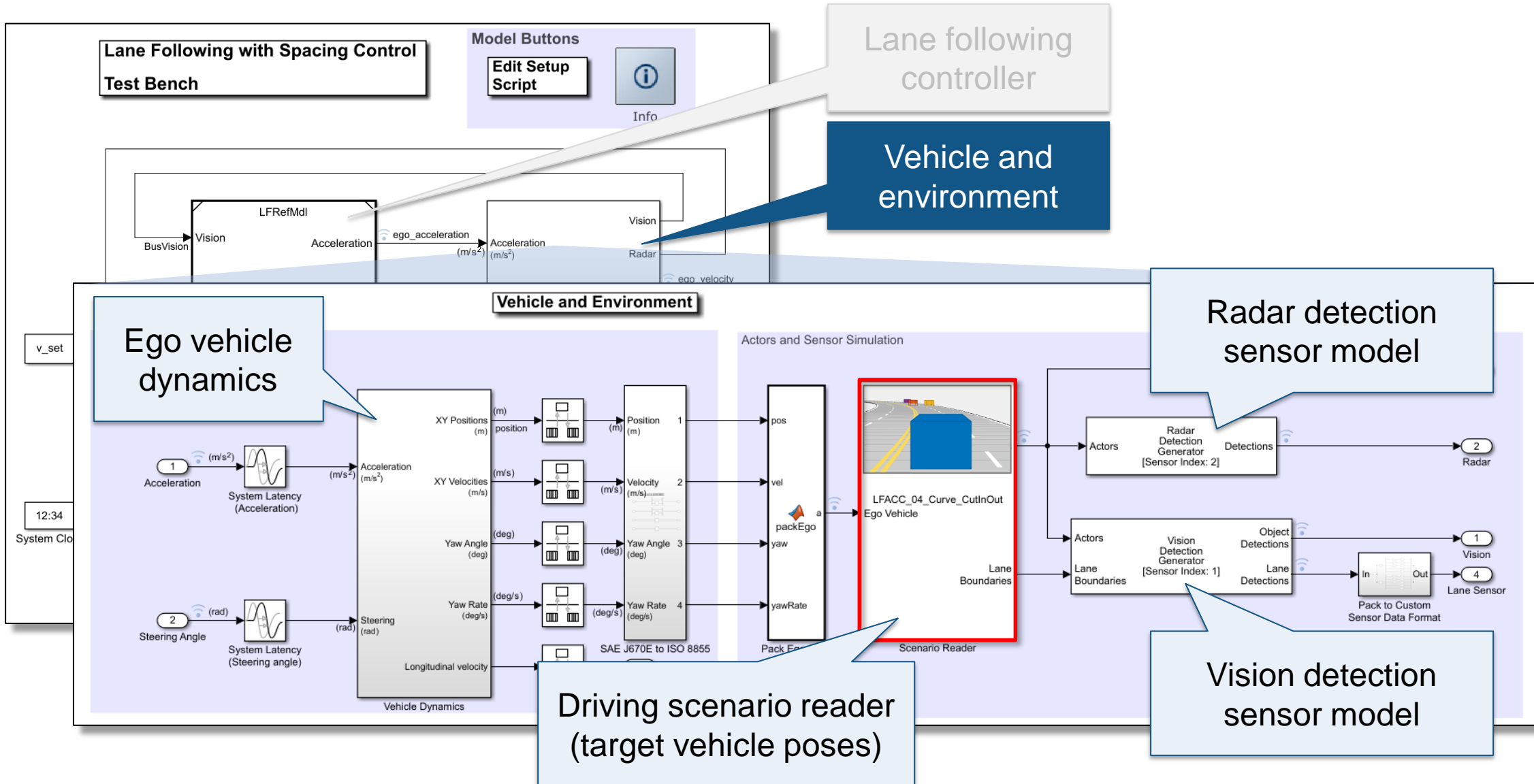
# Surrogate closed-loop system for lane centering



[Lane Following Control with Sensor Fusion and Lane Detection](#)

Automated Driving Toolbox™

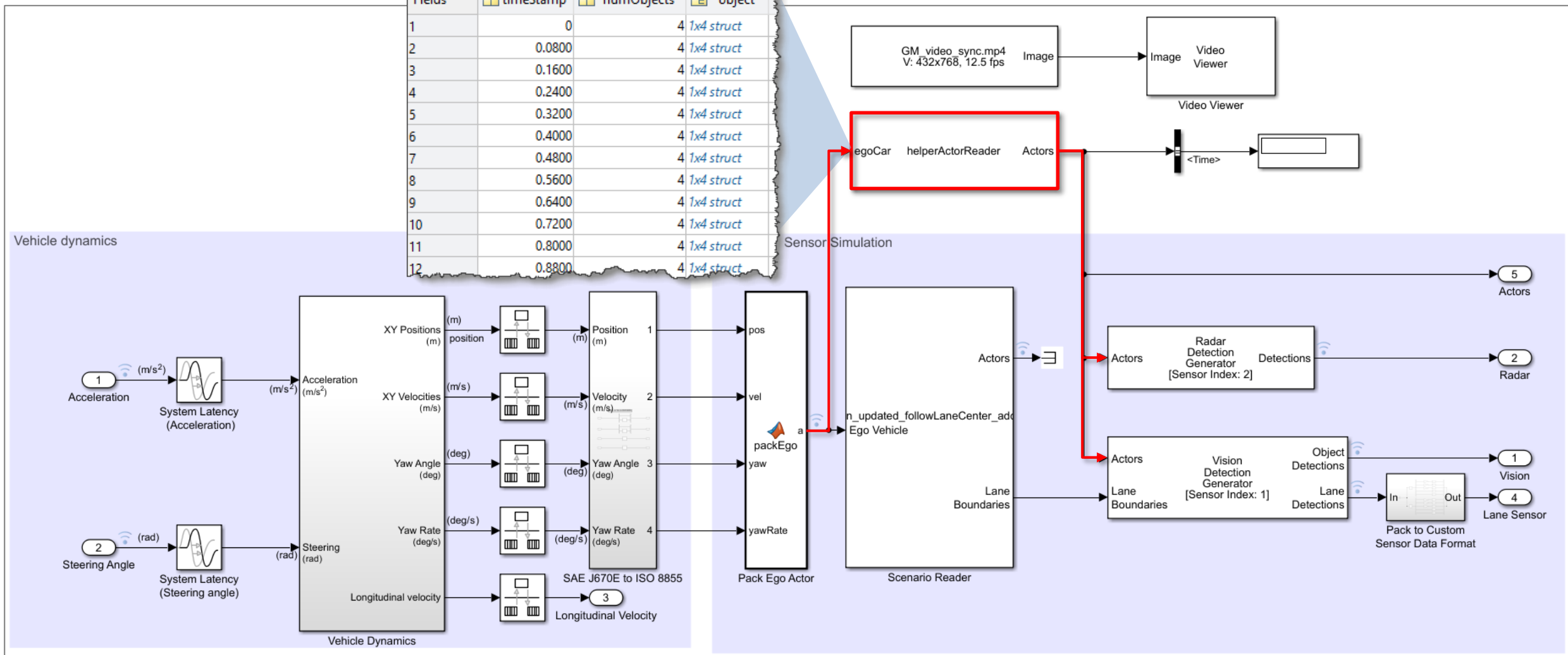
# Surrogate closed-loop system for lane centering



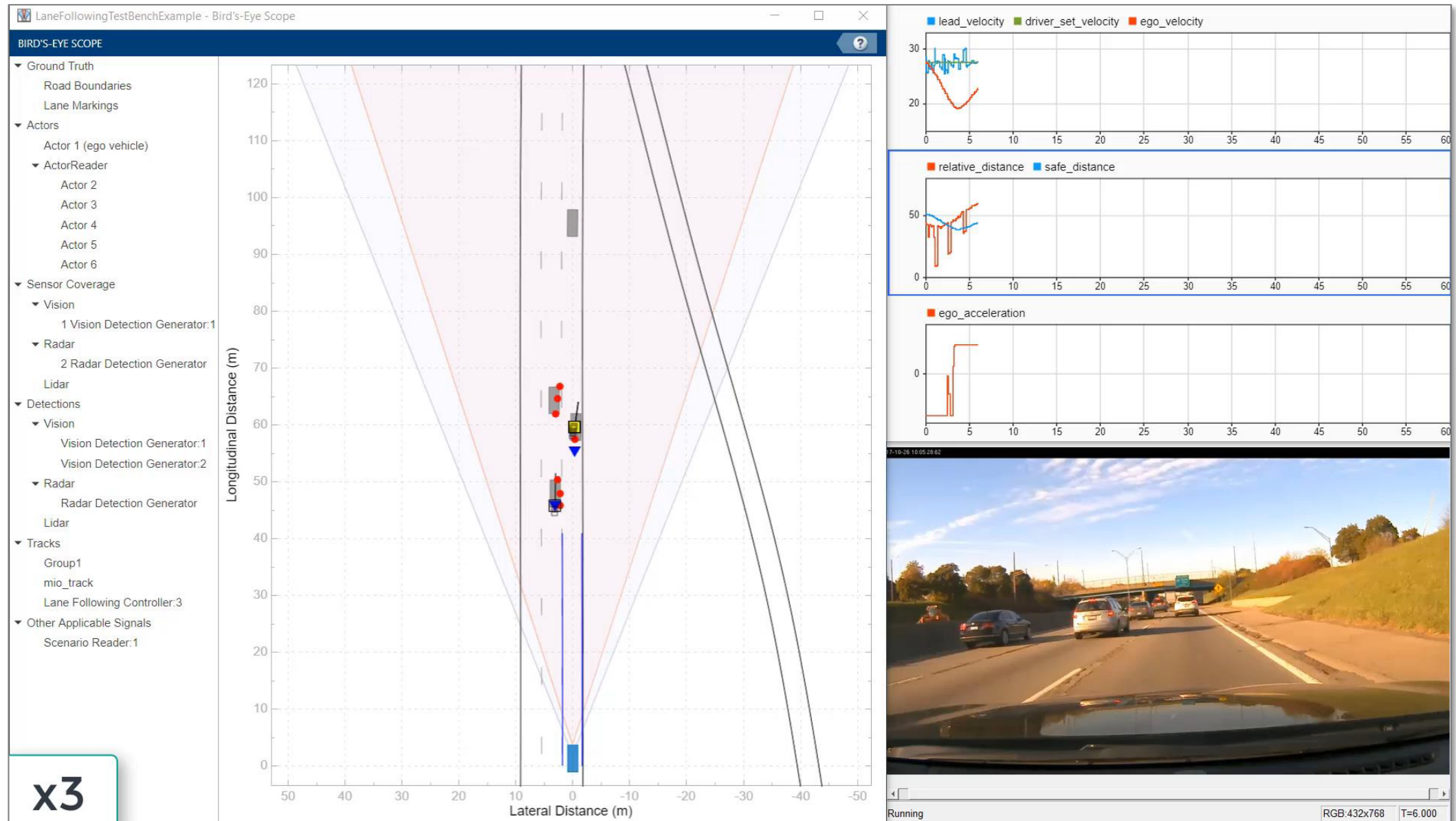


# Lane centering test bench with recorded target vehicles

Fields	timeStamp	numObjects	object
1	0	4	1x4 struct
2	0.0800	4	1x4 struct
3	0.1600	4	1x4 struct
4	0.2400	4	1x4 struct
5	0.3200	4	1x4 struct
6	0.4000	4	1x4 struct
7	0.4800	4	1x4 struct
8	0.5600	4	1x4 struct
9	0.6400	4	1x4 struct
10	0.7200	4	1x4 struct
11	0.8000	4	1x4 struct
12	0.8800	4	1x4 struct

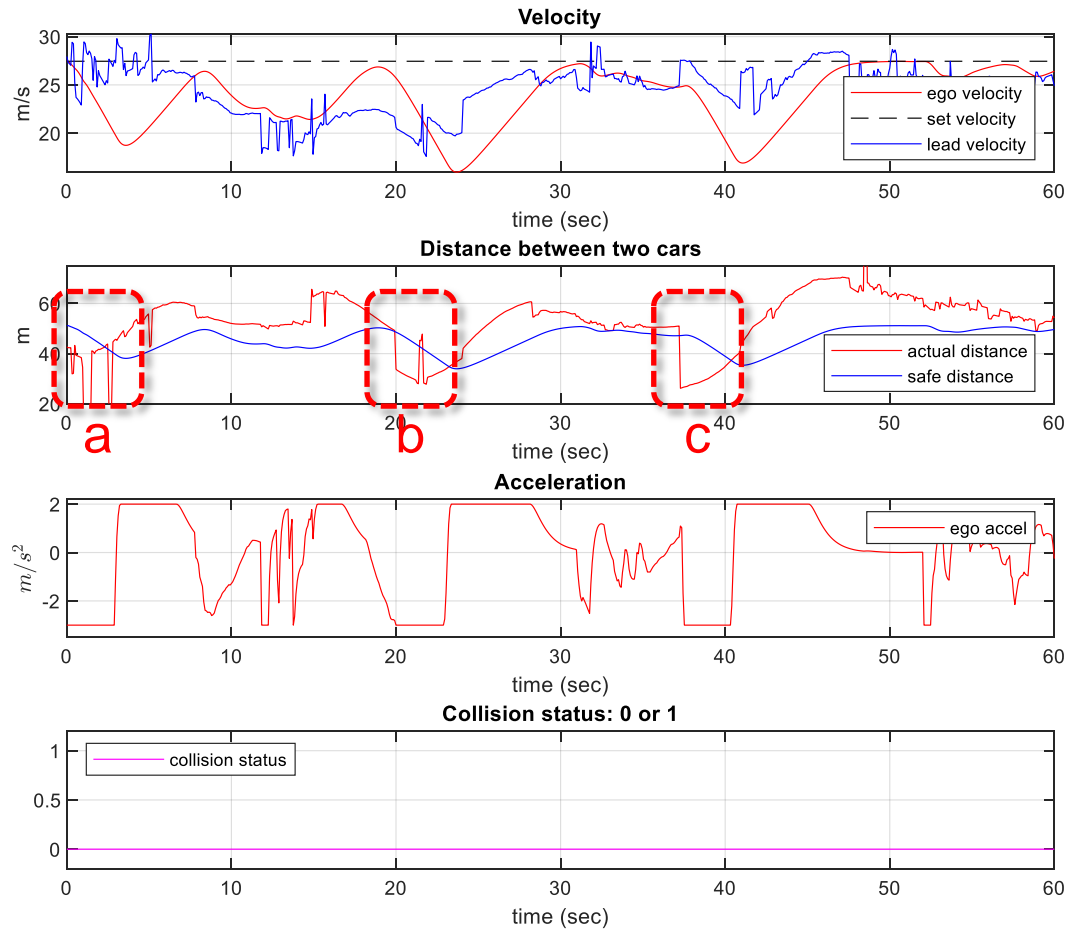


# Closed-loop simulation using reconstructed virtual driving scenario

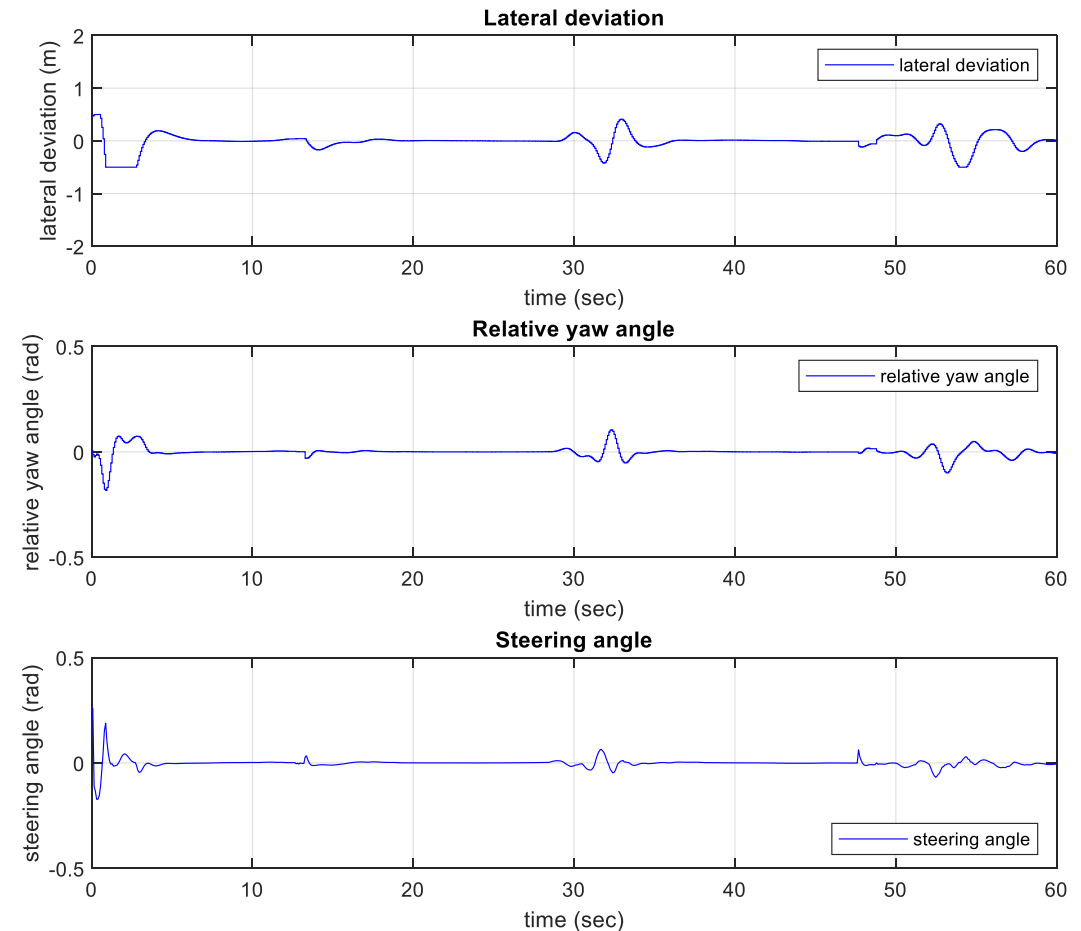


# Simulation result (longitudinal & lateral control performance)

## longitudinal control performance



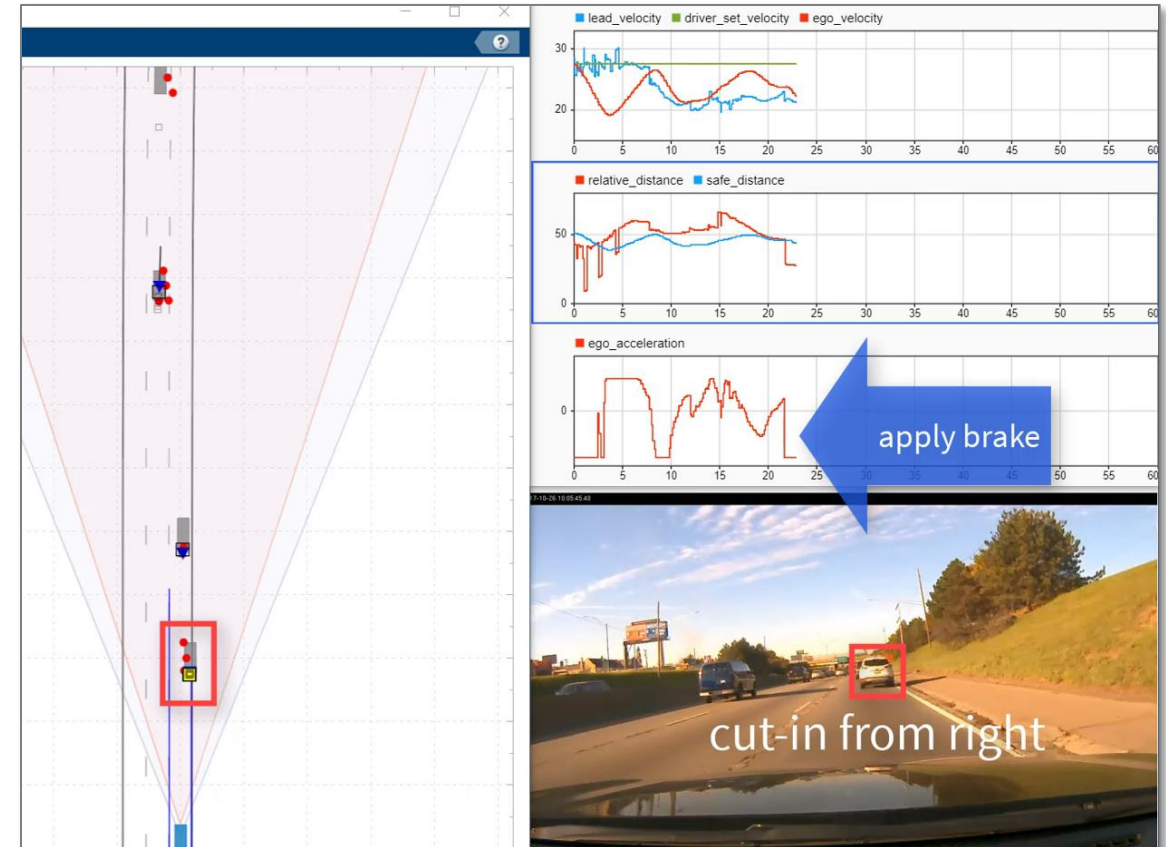
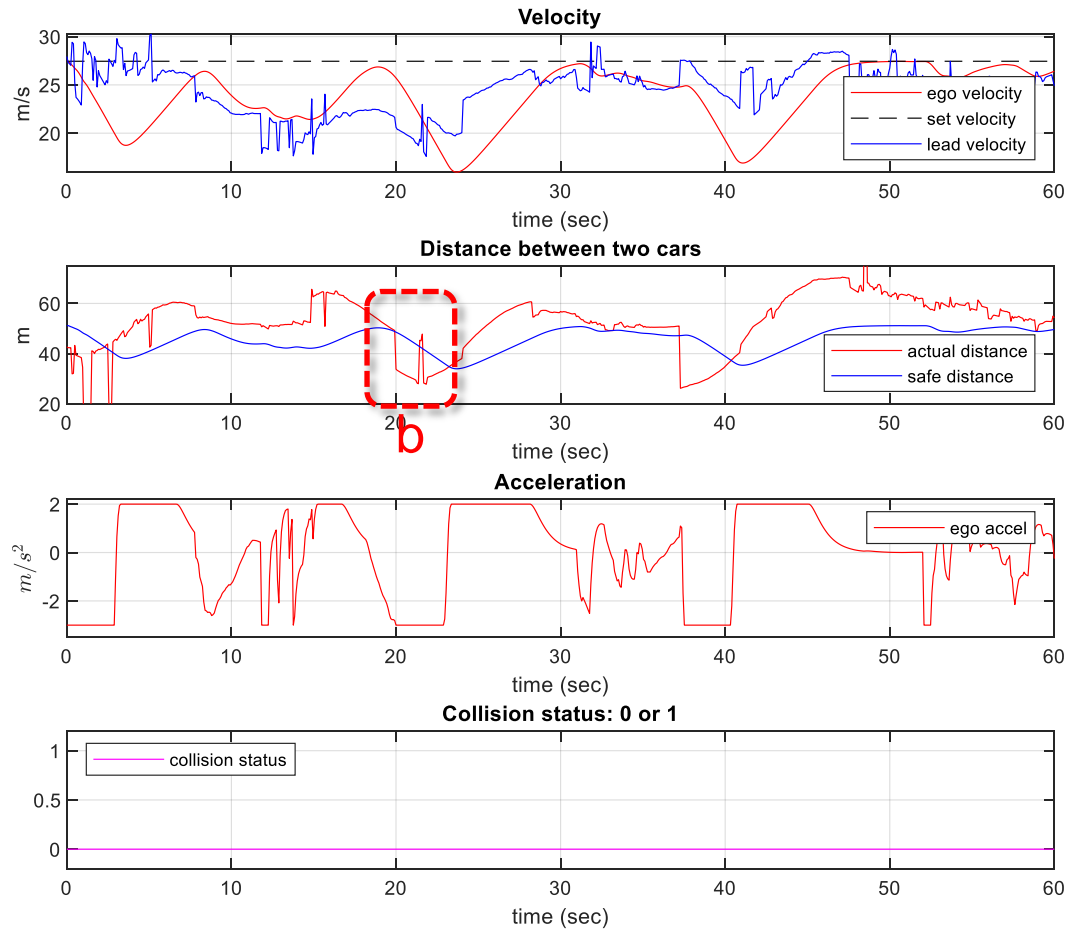
## lateral control performance



a,b,c : problem cases where headway distance drops below the safe distance.

# Driving case (b): cut-in vehicle at low speed

## longitudinal control performance

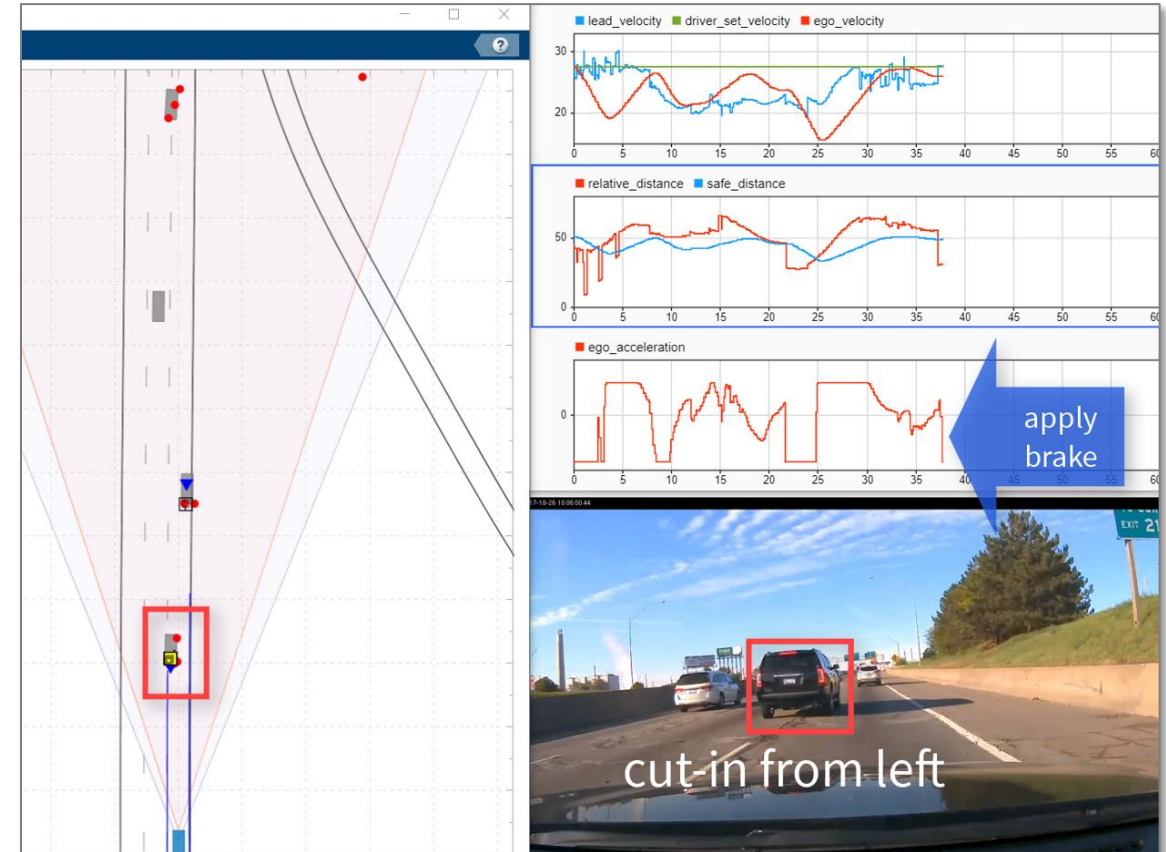
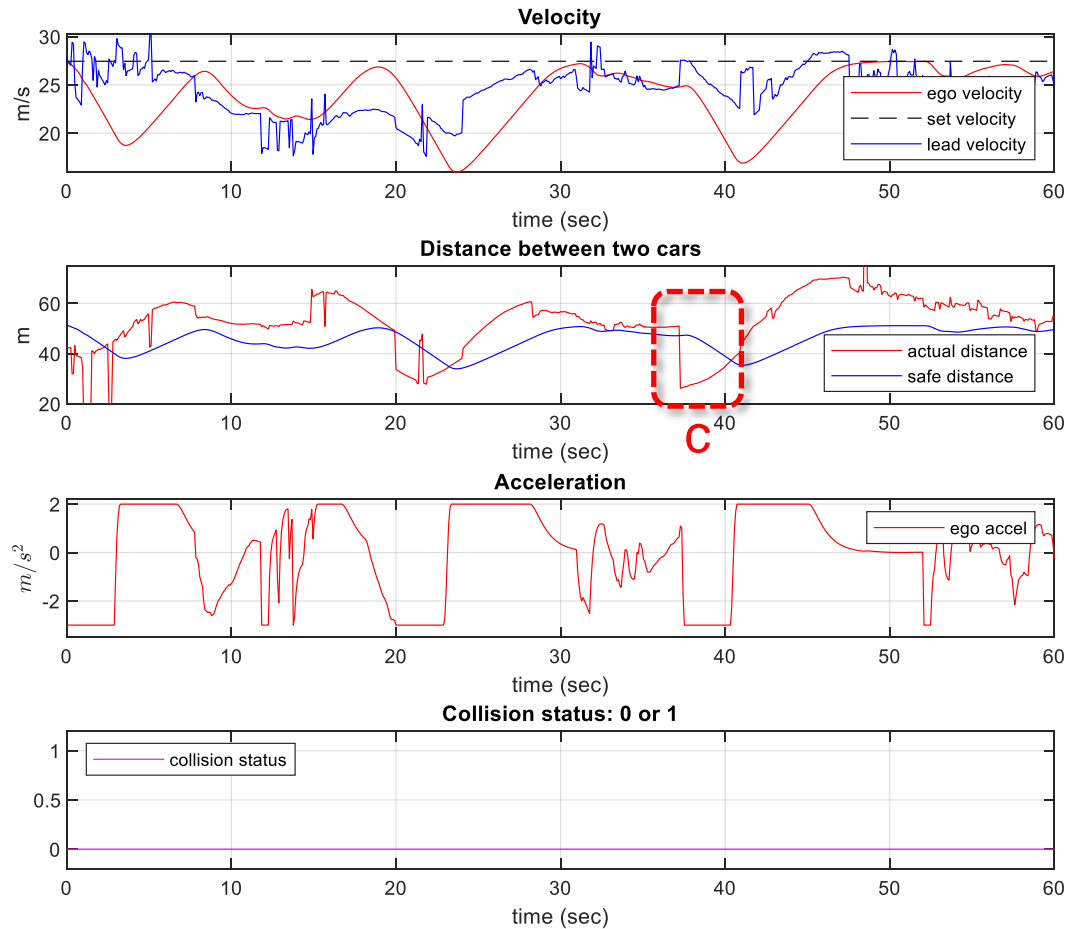


a,b,c : problem cases where headway distance drops below the safe distance.



# Driving case (c): cut-in vehicle with too close distance

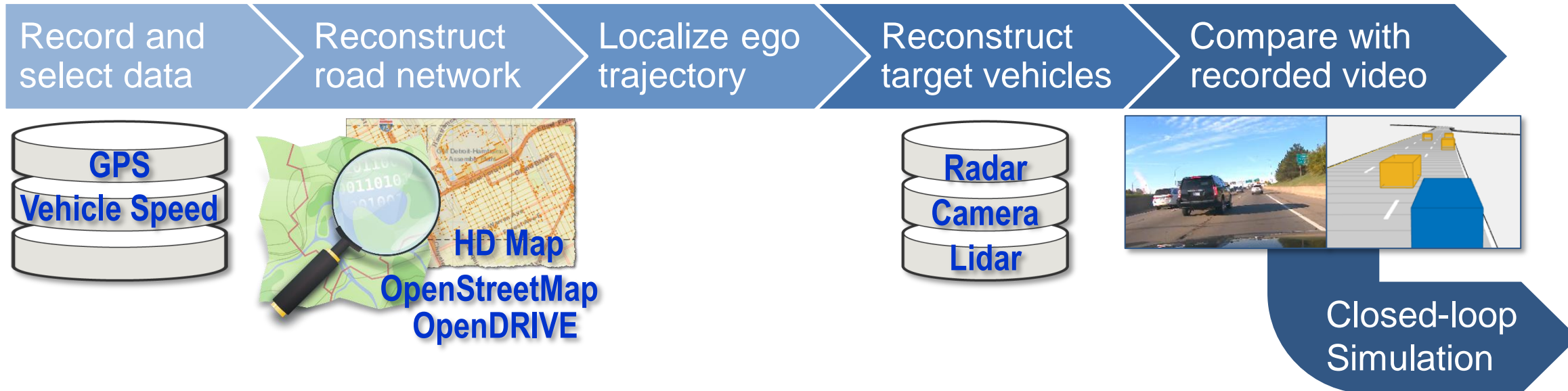
## longitudinal control performance



a,b,c : problem cases where headway distance drops below the safe distance.

# Conclusion


- Created virtual driving scenario from recorded data



- Reproduced real-world driving scenario in the virtual simulation environment
  - Assess functional behavior and identify root cause for problem cases
  - Reduce development time with limited resources
  - Enable repetitive tests for hazardous scenarios

## Remark

- Collaborative effort between GM and MathWorks.
- This study has been published in the SAE paper.

2020-01-0718    Published 14 Apr 2020	
	<h3 style="color: #008080;">Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic</h3> <p><b>Seo-Wook Park, Kunal Patil, Will Wilson, and Mark Corless</b> The MathWorks, Inc.</p> <p><b>Gabriel Choi and Paul Adam</b> General Motors LLC</p> <p><i>Citation:</i> Park, S.-W, Patil, K., Wilson, W., Corless, M. et al., "Creating Driving Scenarios from Recorded Vehicle Data for Validating Lane Centering System in Highway Traffic," SAE Technical Paper 2020-01-0718, 2020, doi:10.4271/2020-01-0718.</p>

# Presenter contact info and poll questions

Please contact us with questions

- Gabriel Choi, General Motors LLC ([gabriel.choi@gm.com](mailto:gabriel.choi@gm.com))
  - Seo-Wook Park, MathWorks ([spark@mathworks.com](mailto:spark@mathworks.com))
- 
- Poll questions : I found this technique the most interesting
    1. Access mdf data
    2. Road network creation from HD map
    3. Ego vehicle localization
    4. Reconstruct target vehicles
    5. Data visualization
    6. Close-loop system integration for lane centering with Simulink
  - If you would like to an individual follow-up, please provide your name and email address in the WebEx poll area.