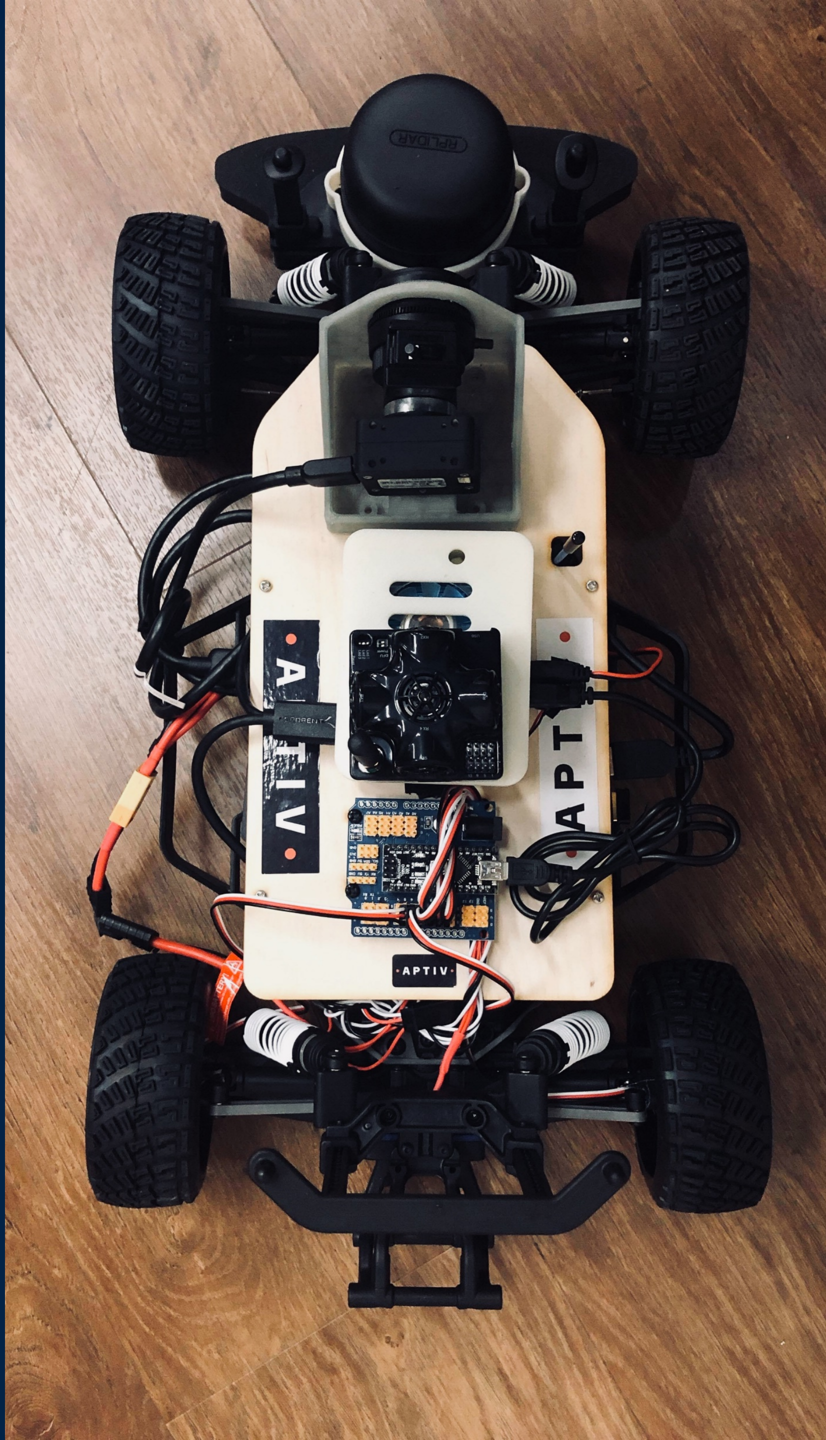
GOLDENEYE

• APTIV •



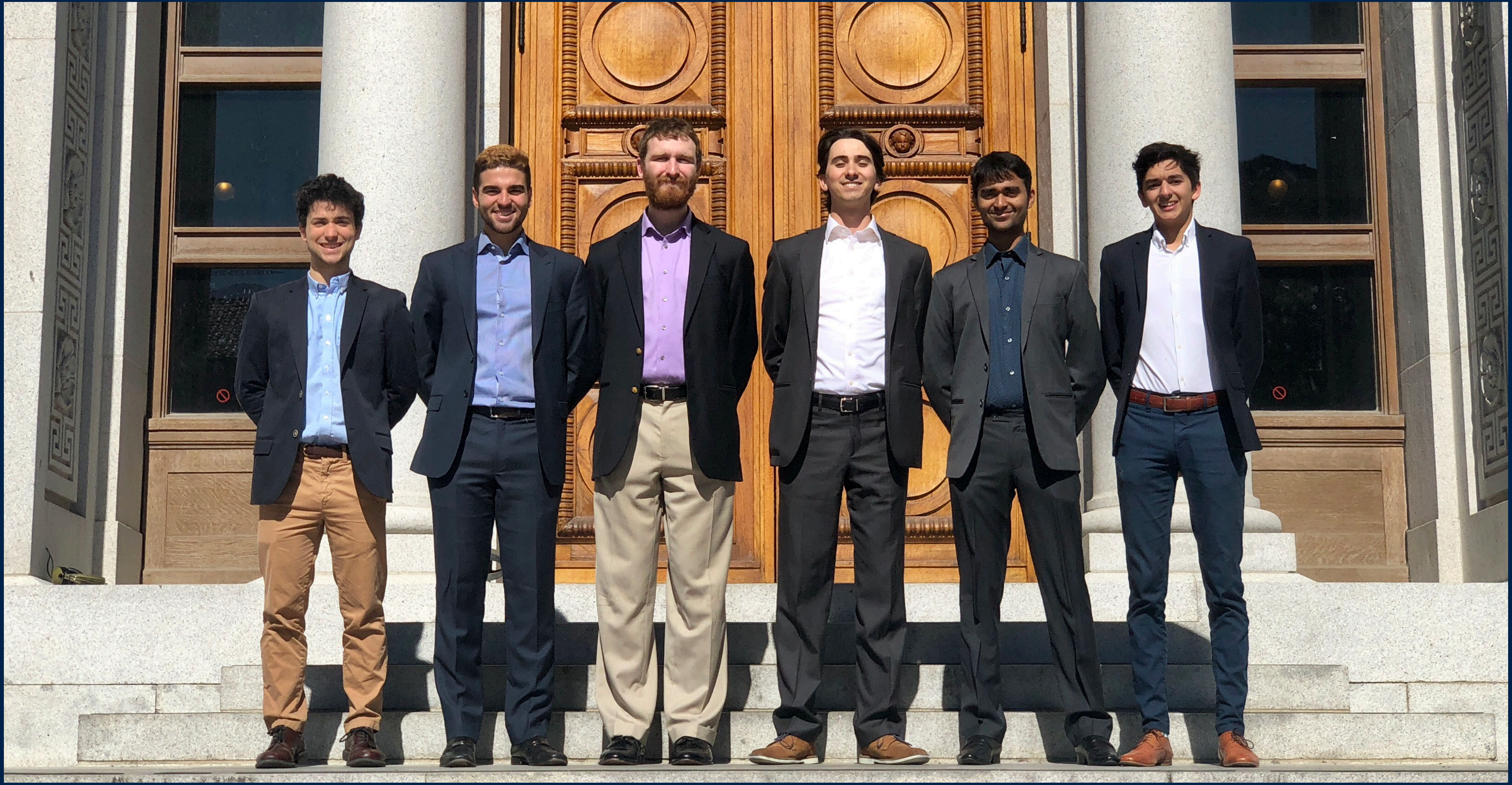
APTIV PLC

Mountain View, CA

April 20, 2018

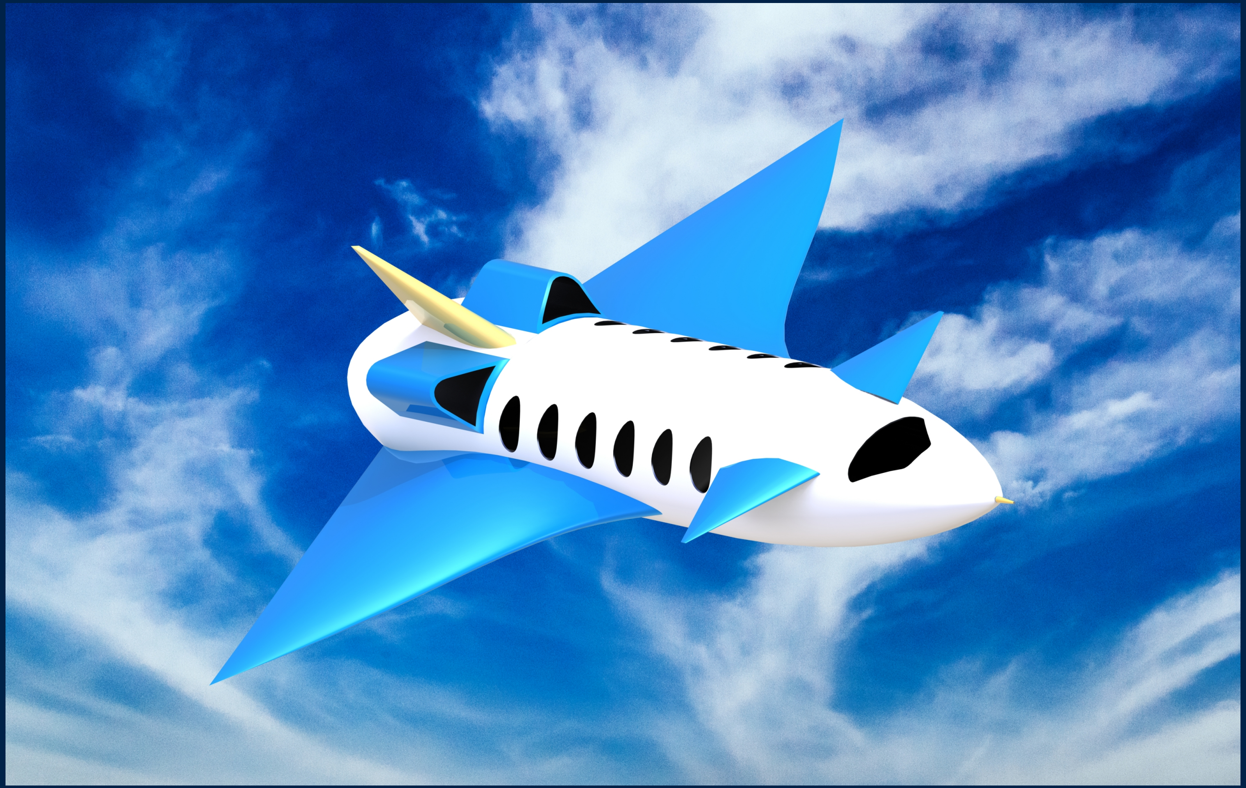
CONTENTS

TEAM



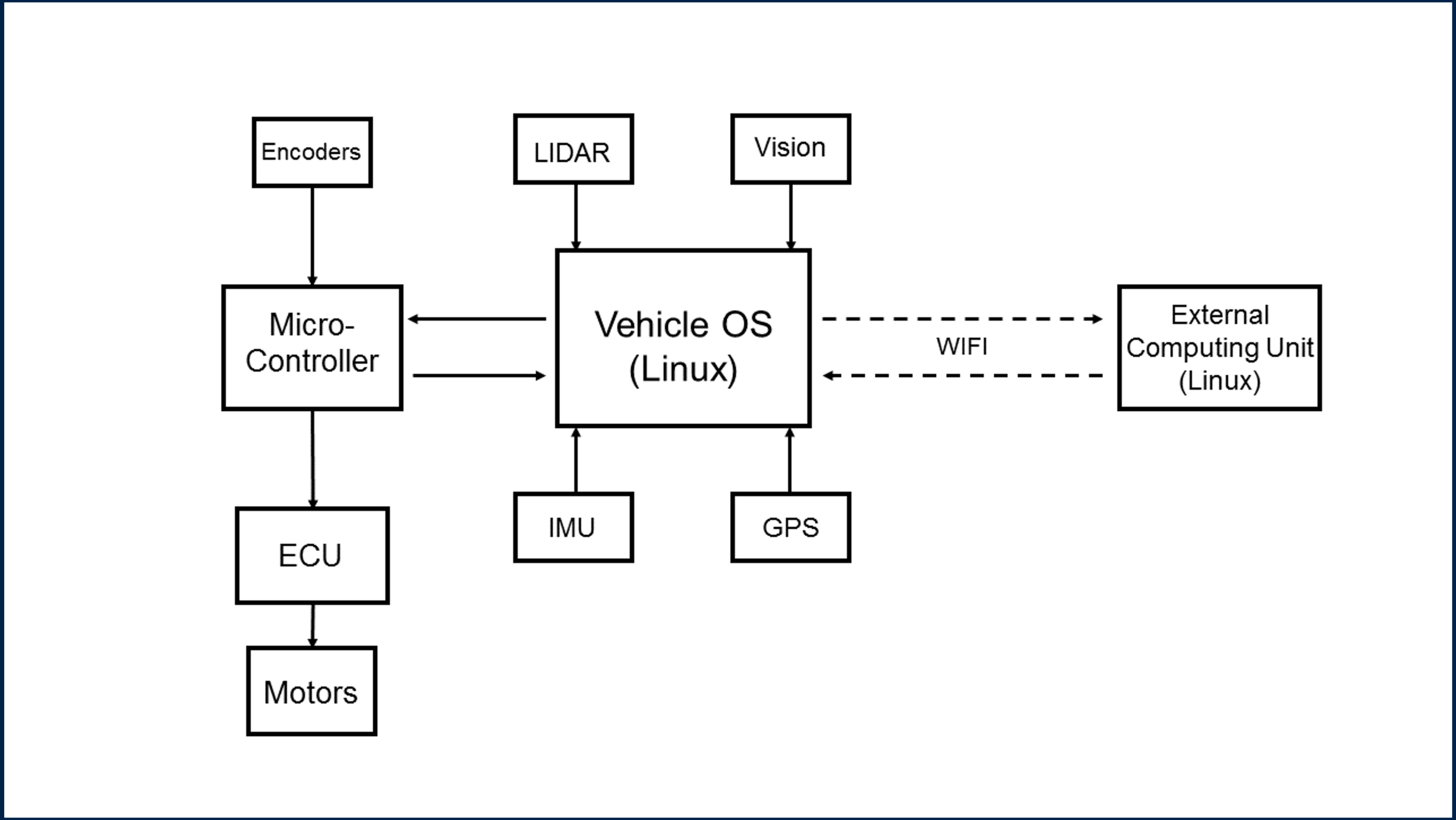
TEAM

- Berkeley RSO
- Won prize in NASA's 2016-17 Aeronautics Design Challenge for a supersonic business jet concept
- With help from Ashish Krupadanam, a new project focused on autonomous vehicles has started
- Working with Prof. Francesco Borrelli's MPC Lab, 2 1/10 scale RC cars have been built
- Goal: develop applications in predictive driving



Supersonic Business Jet Design

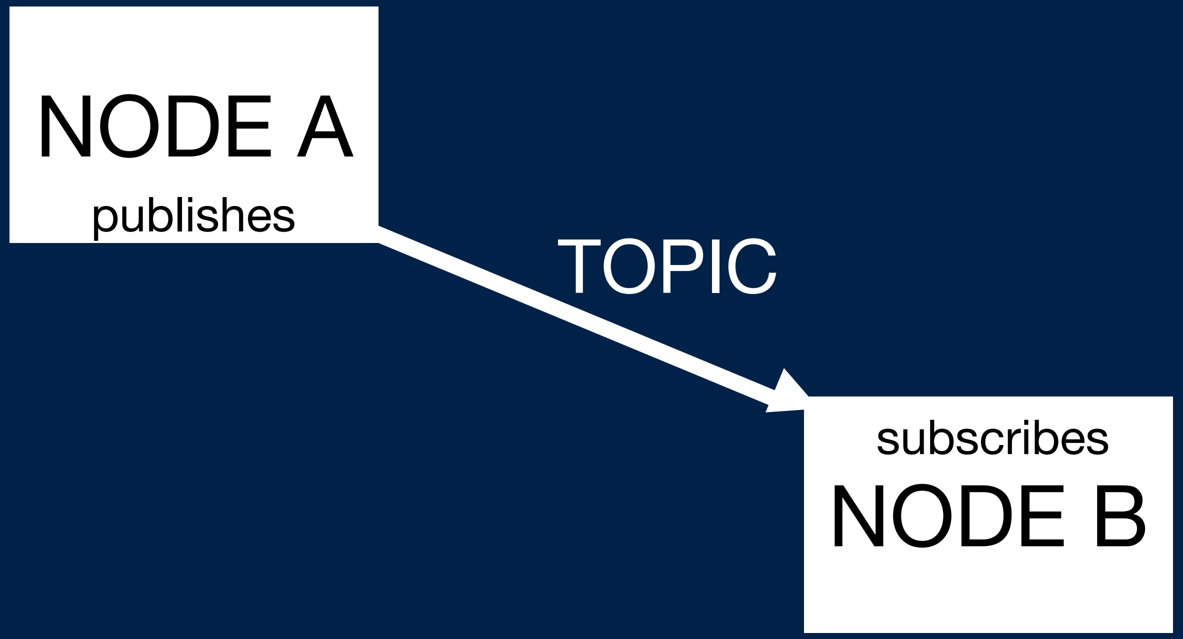
HARDWARE ARCHITECTURE



SOFTWARE ARCHITECTURE

Use Robotic Operating System (ROS)

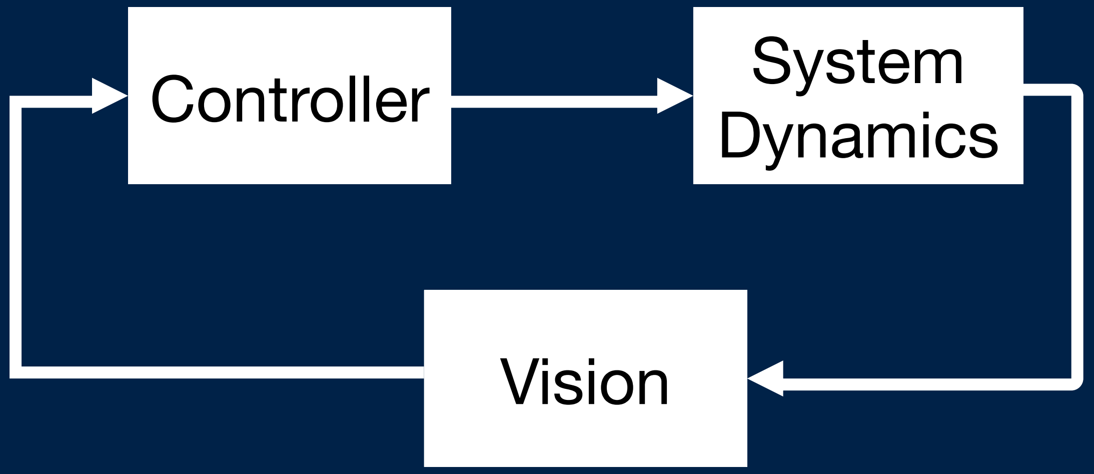
- Flexible, Robust development framework
- Provides pipeline for communication between programs running in parallel
- Instantiate Nodes, representing processes in system
- Nodes communicate over Topics, through pre-specified datatypes (messages)
- Framework allows for computation on multiple machines



LANE KEEPING

SISO Controller

- PI controller maps the error to an actuation command
- Feedback control makes the system relatively robust to imperfect plant and control constants
- Controller takes into account delay to compensate for high lookahead of camera



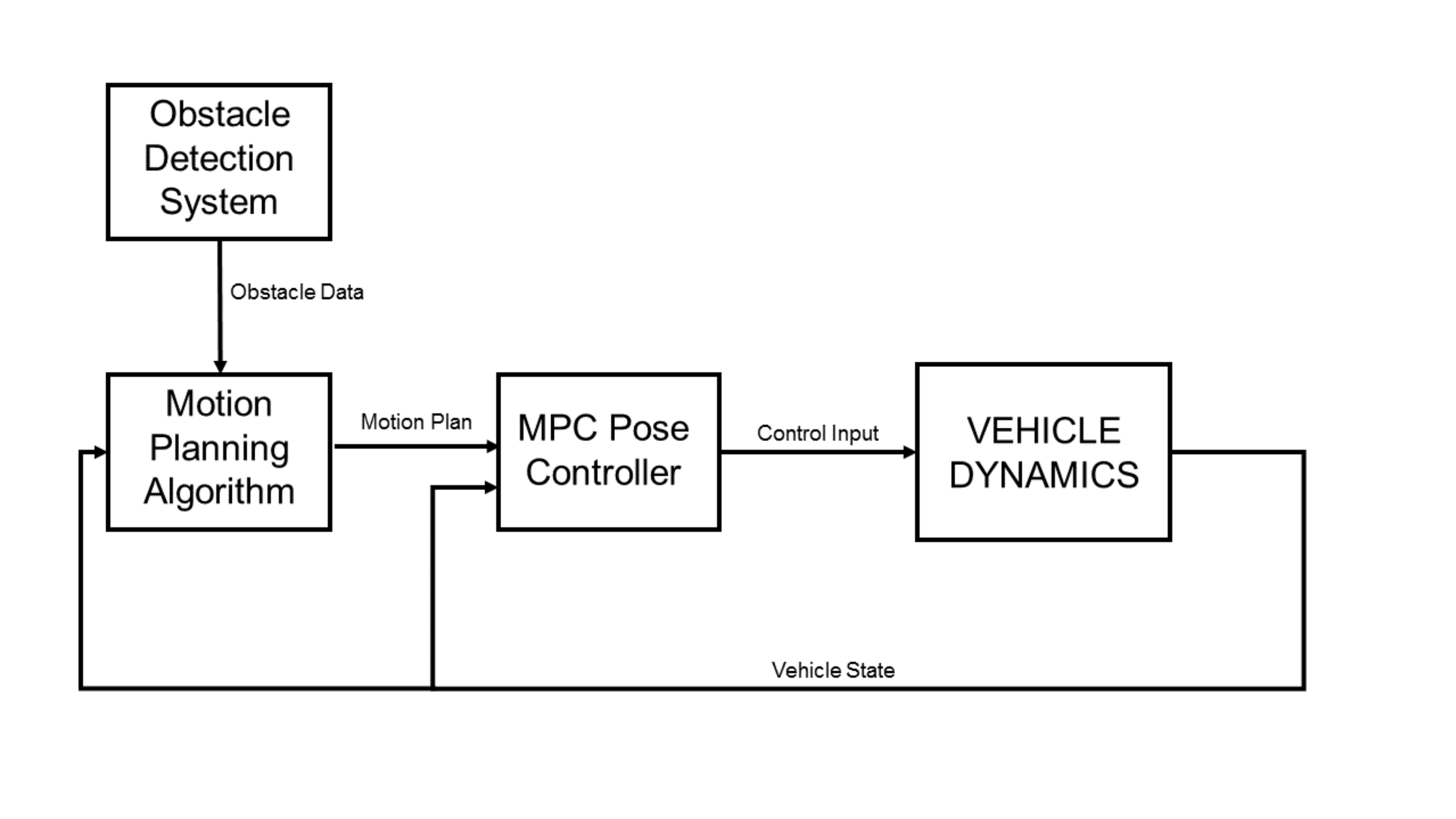
LANE KEEPING

Vision Algorithm

- Thresholding followed by detection of y-coordinate of maximum contrast
- Prioritizing computational efficiency on the Odroid for real time image processing means we can only look at 10 rows of the image to determine error

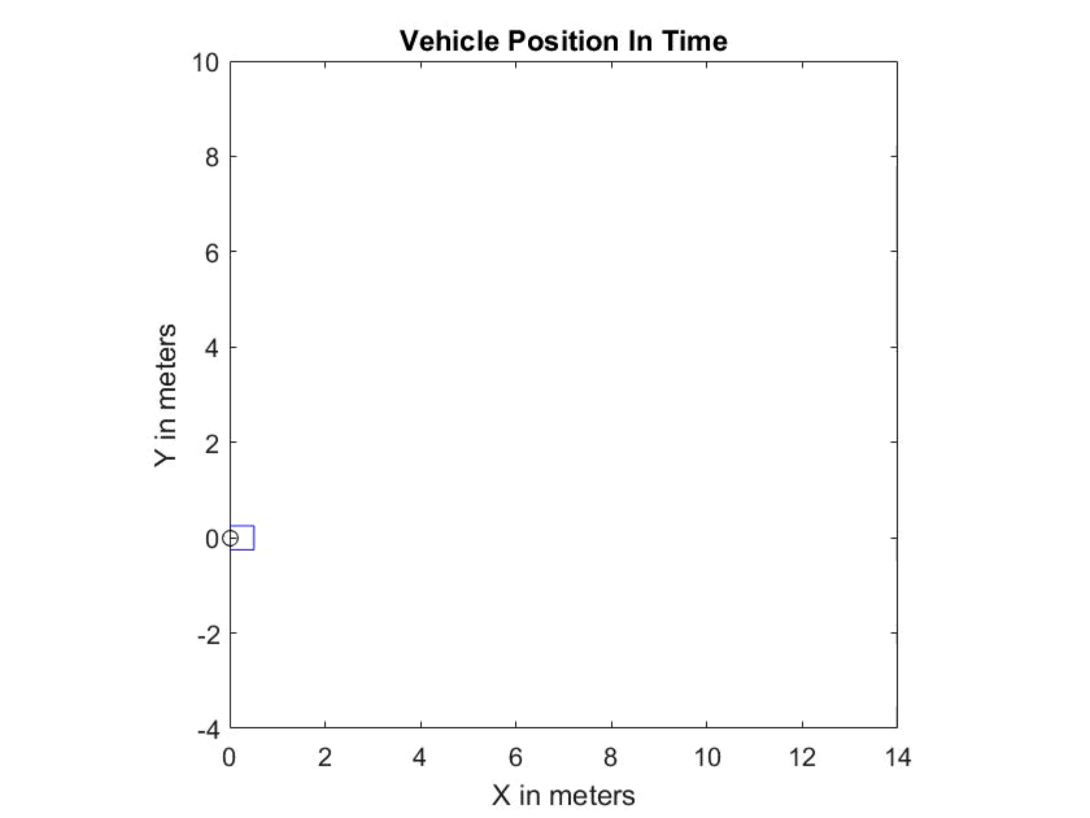

DVS

POINT TO POINT NAVIGATION



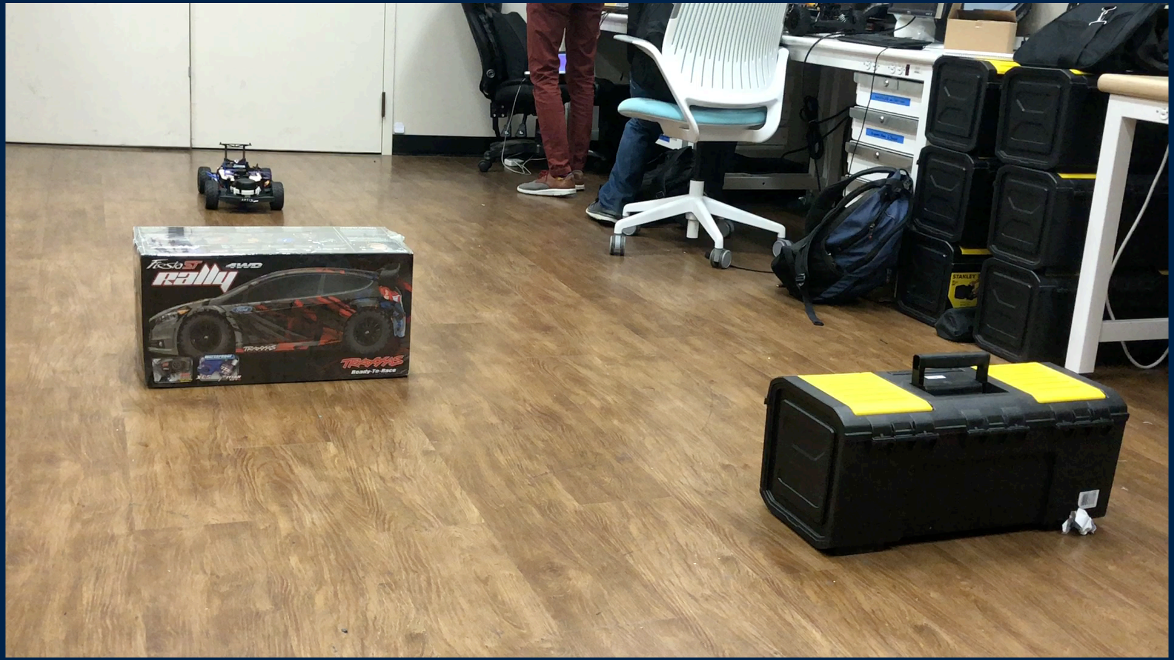
MPC (Model Predictive Controller)

- System dynamics are known
- In discrete time, state vector $\vec{x}_{[k+n]}$ depends on $\vec{x}_{[k]}$ and control input $\vec{u}_{[k]}$, $\vec{u}_{[k+1]}, \ldots, \vec{u}_{[k+n-1]}$
- Formulate and solve optimization problem at each timestep



MPC (Model Predictive Controller)

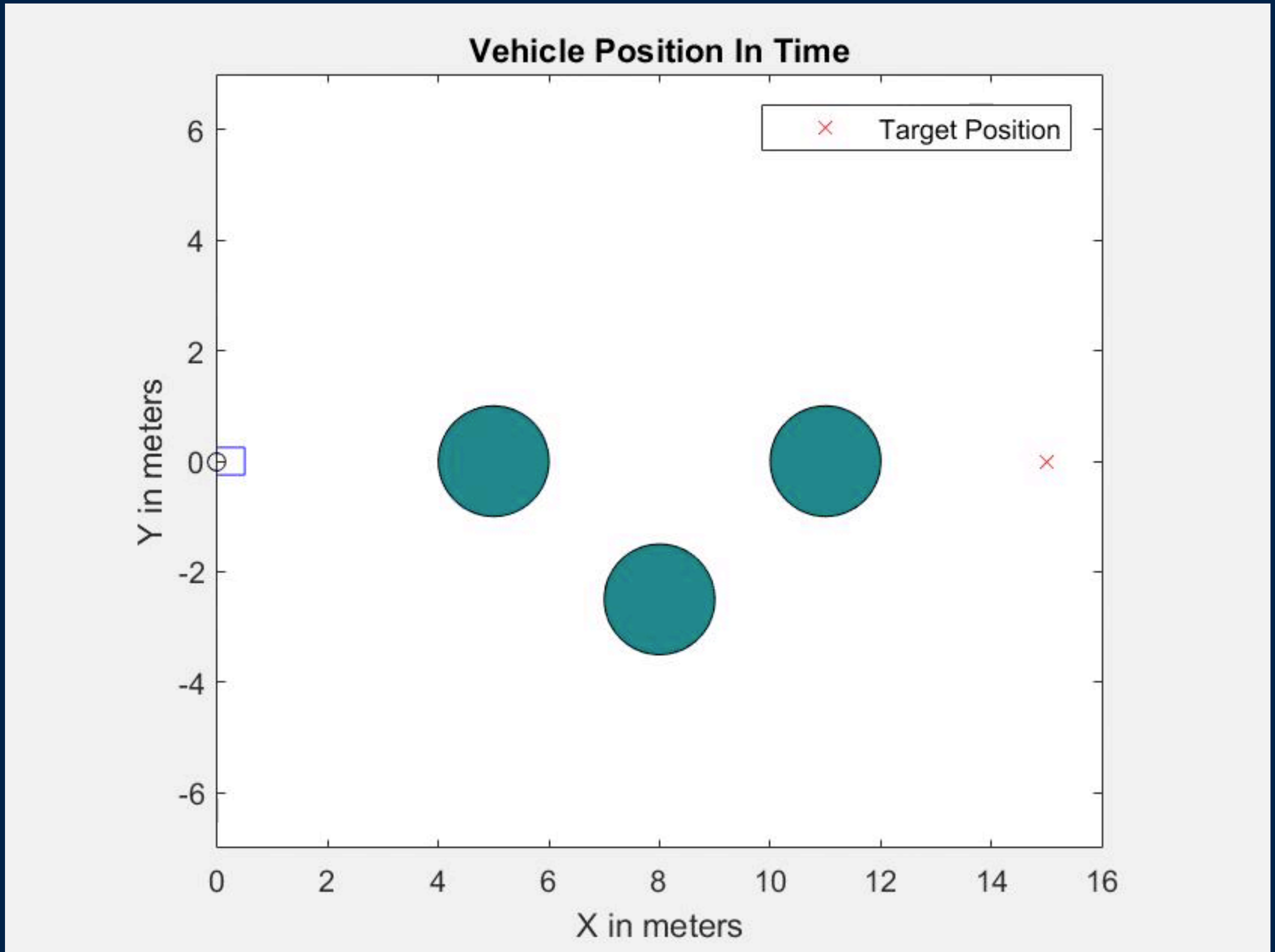
- Single target point
- Brute force trajectory generation over finite receding horizon
- Evaluate cost function over all trajectories, execute optimal
- Penalizes heavy actuation to achieve smoother trajectories



Euclidean Distance Cost Function
→ Aggressive Actuation

MOTION PLANNING

- Motion planner plans path before initiating motion
- Path represented by set of points $[x_i, y_i]$
- Planner regulates waypoint sent to MPC controller
- Approach guarantees feasible solution in real time, low computational cost.

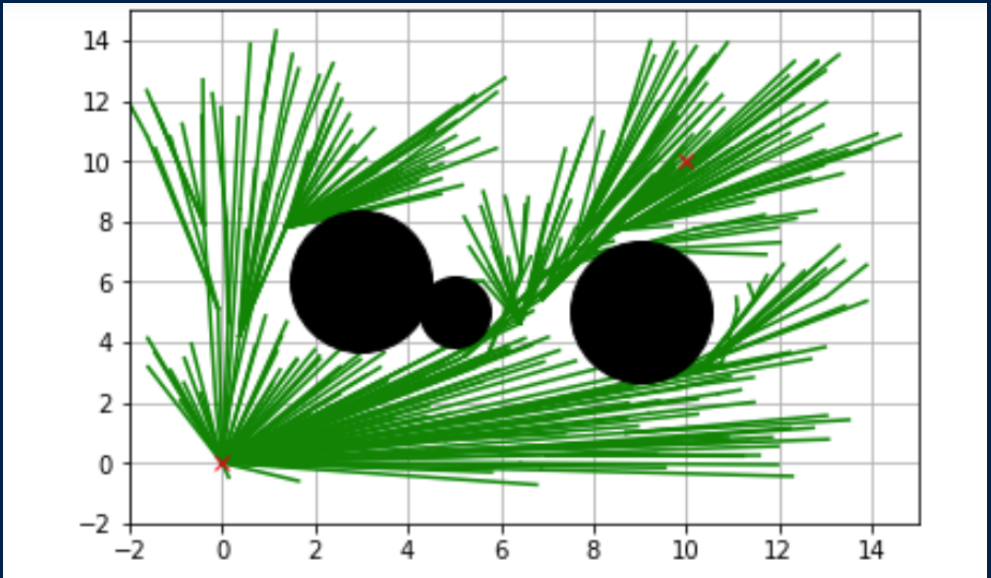


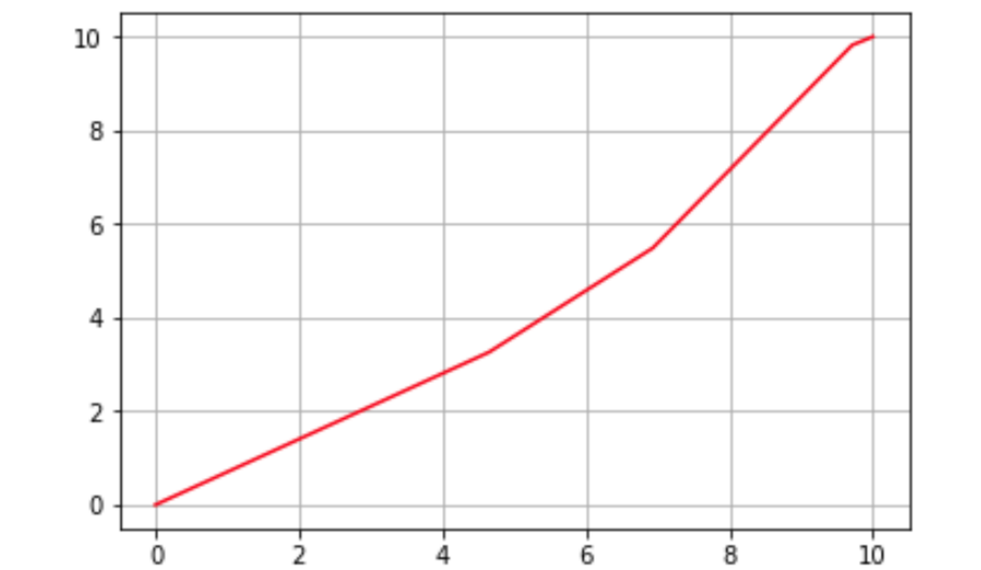
MOTION PLANNING

RRT* Algorithm

- Key Observation: Creating a feasible trajectory from scratch is hard, checking whether a given trajectory is feasible is easy.

- Developed by Prof. Emilio Frazzoli et al. at MIT, current CTO at Nutonomy (Aptiv acquisition)





MOTION PLANNING

Geometric Algorithm

- Works for navigation through field of obstacles.
- Generates sparse set of waypoints, relies on MPC controller to navigate in a smooth manner.
- Capitalizes on knowledge of turning radius

Greedy Search

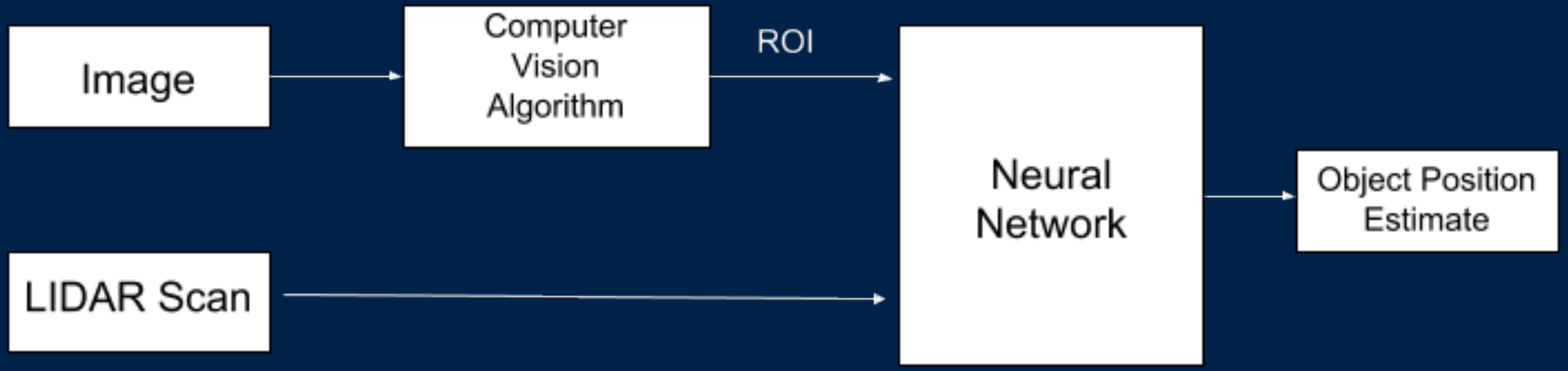
1. current location = start point
2. while(goal != reached):
   a. obs = find nearest obstacle within range of motion
   b. on circle between current location and obs : sample feasible points
   c. find optimal point
   d. append optimal point to path and set current location to optimal point

SENSOR FUSION

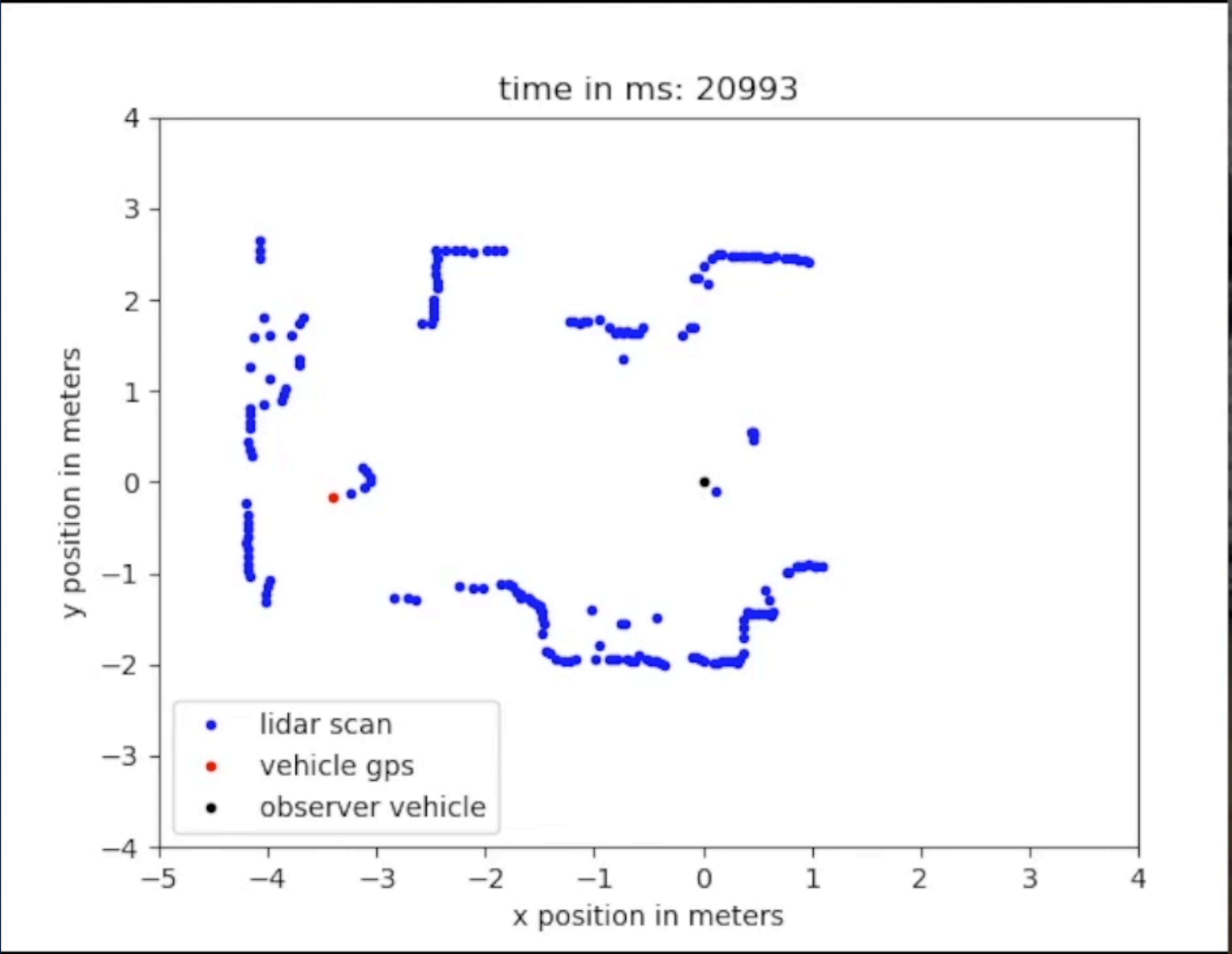
- Camera provides high resolution in detecting detail, though fails to perceive depth
- LIDAR data generates depth information through 2D-point cloud, but lacks detail needed to distinguish objects
- **Goal:** Capitalize on strengths of respective sensor feeds and generate accurate estimates of object location

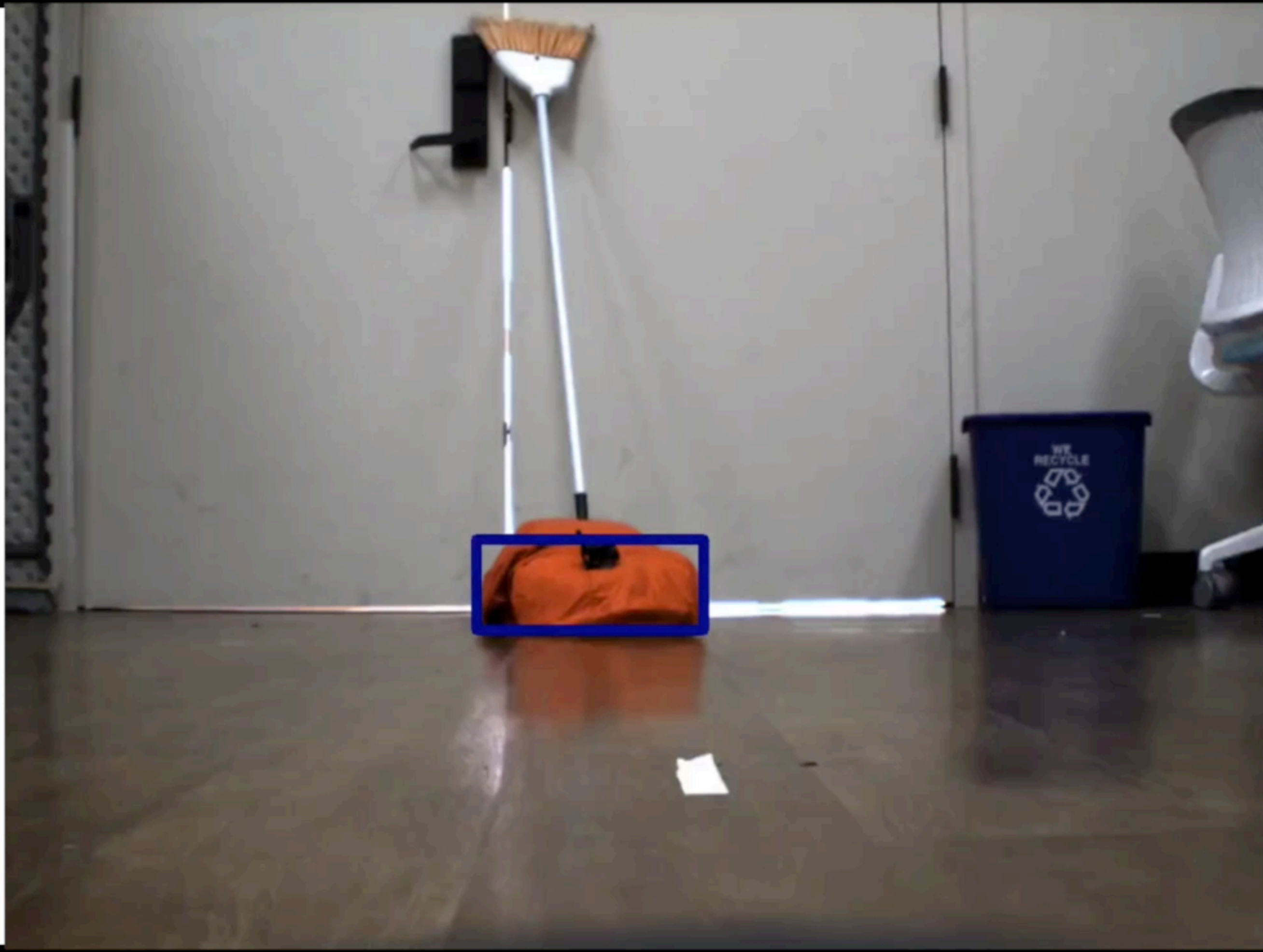
SENSOR FUSION

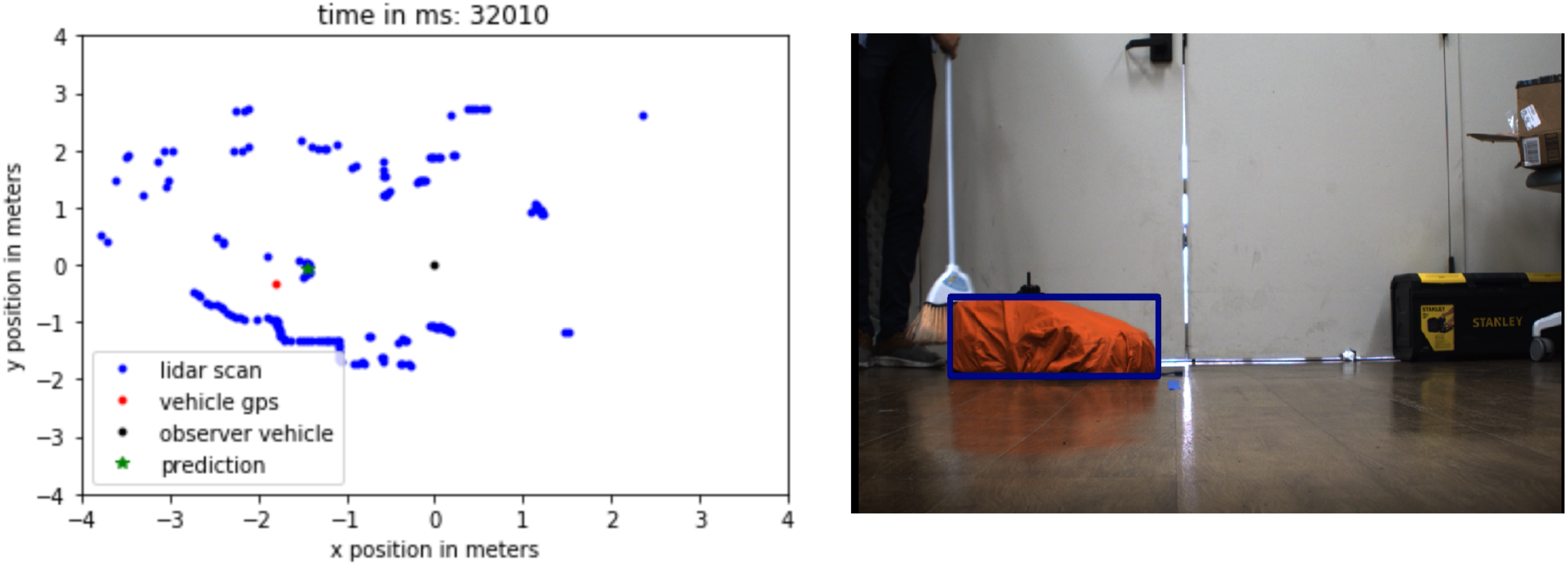
Given a region of interest in an image, how can we fuse the corresponding LIDAR points for depth perception?



- LIDAR, camera, and GPS data collected during training to associate vehicle position in space relative to another object
- This is done by driving two cars within each other's field of view—GPS data is referenced to generate ground truth relative coordinates


time in ms: 20993
4
3
2
1
0
−1
−2
−3
−4
y position in meters
−5
−4
−3
−2
−1
0
1
2
3
4
x position in meters
lidar scan
vehicle gps
observer vehicle





time in ms: 32010
y position in meters
x position in meters
4
3
2
1
0
−1
−2
−3
−4
−4
−3
−2
−1
0
1
2
3
4
lidar scan
vehicle gps
observer vehicle
prediction
STANLEY

CURRENT WORK

Motion Planning

- Combining methods like RRT* for global path planning while using local optimizers like Trajopt for smoother short-distance paths
- Defining allowed and disallowed regions of sampling as continuous blocks
- Using longer horizon MPC controllers to penalize extreme actuation

SLAM

- Using in-build ROS SLAM to localize the car indoor and detect allowed regions using LIDAR
- Using EKF to track other vehicles using LIDAR and arriving at probabilistically accurate estimates of position, velocity, acceleration and predicted paths

FUTURE WORK

Racing Inference and Planning

- Using both cars together around a track to infer racing strategies of observed vehicles around a racetrack
- Adapting to opponents strategy using an MPC controller further optimized with Q-Learning

Simulator

- A reasonably accurate simulator of the car is essential to making deep learning or Q-Learning based approaches feasible
- Will facilitate exploration of these techniques given sparse data collected in the real world, can explore meta-learning and fine-tuning of simulator trained models