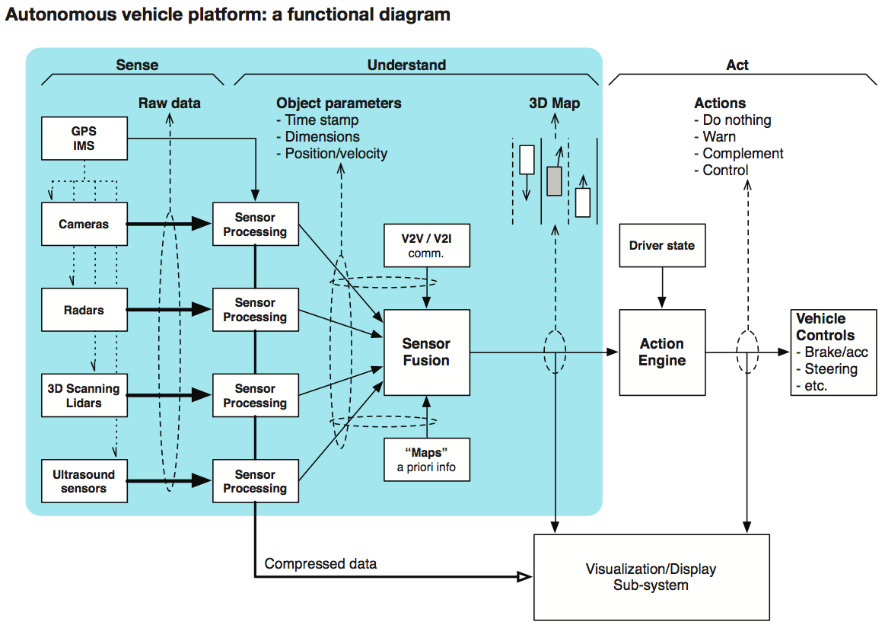
Introduction

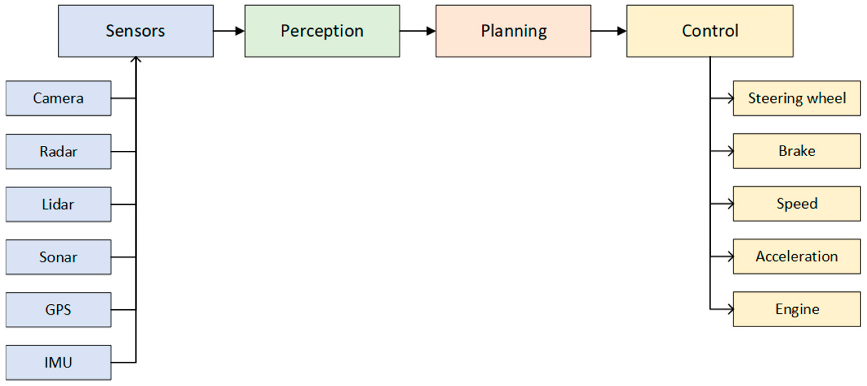
We have started sensor project in this semester. Then, we chose to be in hardware team. We started learning matlab toolboxes and watching sample videos. There are number of sensors in Fusion Sensor project. We have taken the most significant ones: camera, lidar and radar sensor in Matlab and we have used camera sensor for object detection in Python.

Background information

By talking about camera sensor, Camera recordings show a **visual representation of the world in color.** Apart from the colors, they can also provide texture and contrast data. This information can be evaluated with the aid of software, for example in order to reliably identify a road marking or a traffic sign. Both static and moving objects can be precisely detected and identified. Longer version of LiDAR is Light Detection and Ranging which plays a huge part in the self-driving part of the car. The job of a LiDAR sensor is to 'map' an area surrounding it by measuring distances of objects via reflecting light rays. Moreover, Radar (radio detection and ranging) sensors work similarly as LiDAR, but transmit electromagnetic waves to sense the environment. The Radar outputs can be organized in three different levels: raw data in the form of time-frequency spectrograms, clusters from applying on raw data, and tracks from performing object tracking on the clusters.

Block Diagram





Kalman Filter

We can create a multi-object tracker to fuse information from radar and video camera sensors. The tracker uses Kalman filters that let you estimate the state of motion of a detected object. I used the sensor measurements made on a detected object to continuously solve for the position and velocity of that object. To track moving objects, I used constant-velocity or constant-acceleration motion models. I have researched kalman filter not for implement but for learning.

Steps for:

- To predict position and velocity with some uncertainty.

- Then measure the experimental position and velocity with some uncertainty.

- Finally, increase the certainty of our prediction by combining our prediction with the measurement information.

Implementation

First of all, these two figures were for us to learn how to implement by drivingScenarioDesigner in the next slide.

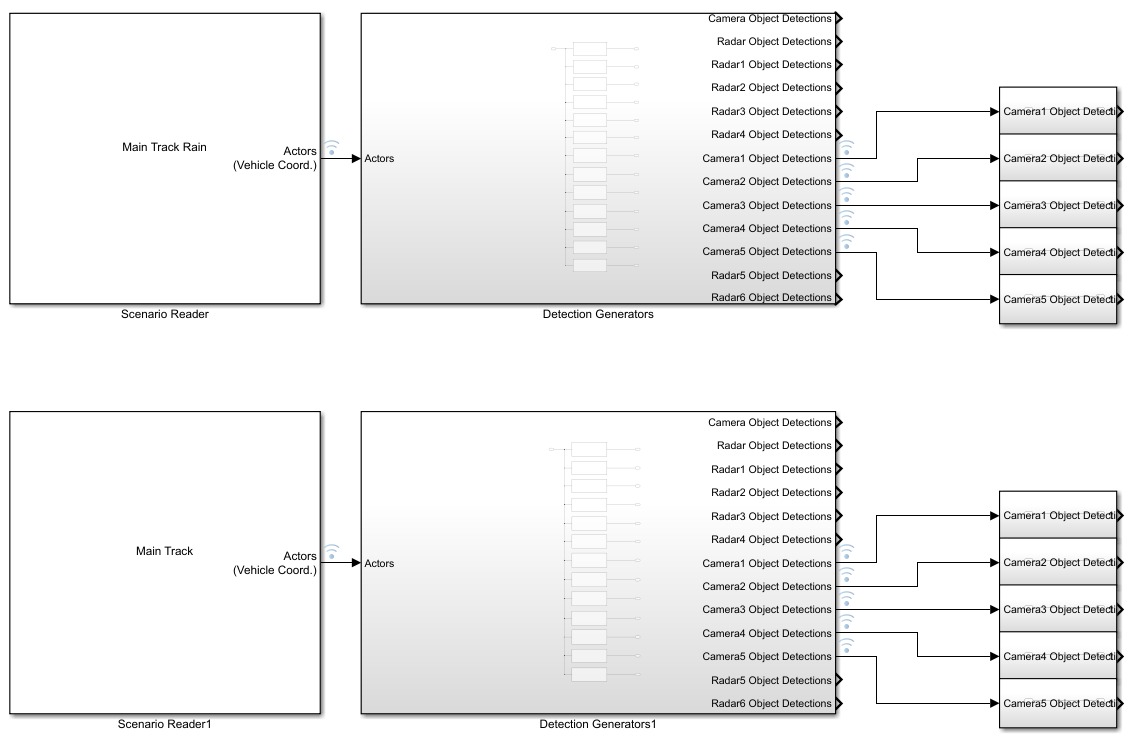
We have started in Driving Scenario Designer app in Matlab. We did synthetic data via drivingScenarioDesigner. We have created our scenario and put sensors, actors and roads. I have my ego vehicle(blue car) which has sensors on it and it is the control mechanism of the view. We have detected the actors thanks to the sensors of the ego vehicle. From data inspector, I have extracted the plotting which gives the number of detections according camera 2, camera 4. I logged signals , enables logging data for specified signals to the MATLAB workspace.

We can do these by DrivingScenarioDesigner

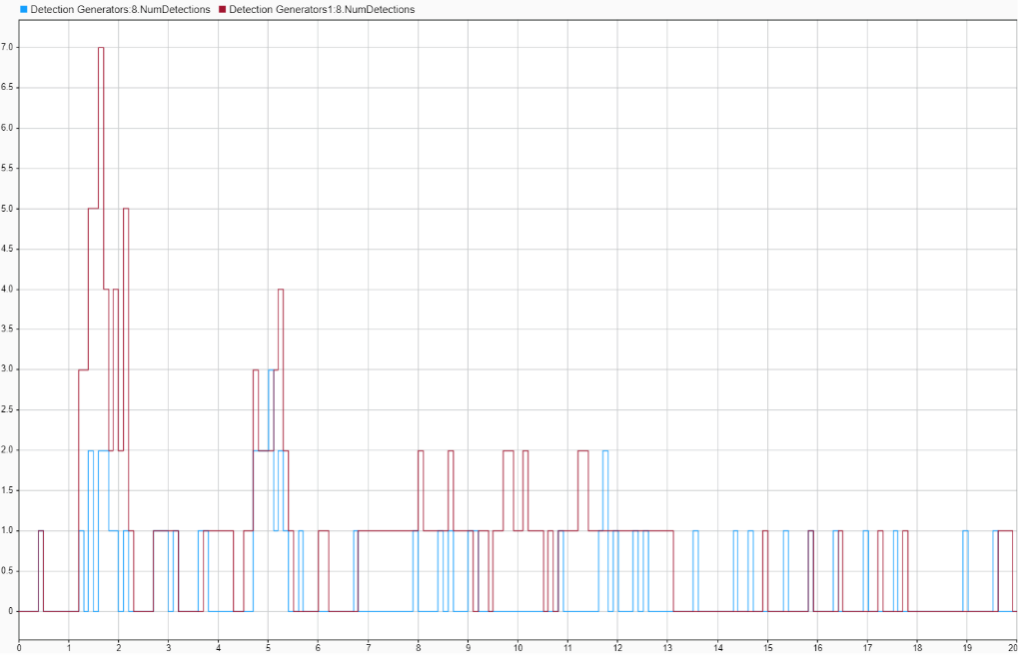
Create and define scenarioFuse and track detections from different sensorsSynthesize scenarios to test

This is our designed road that we created

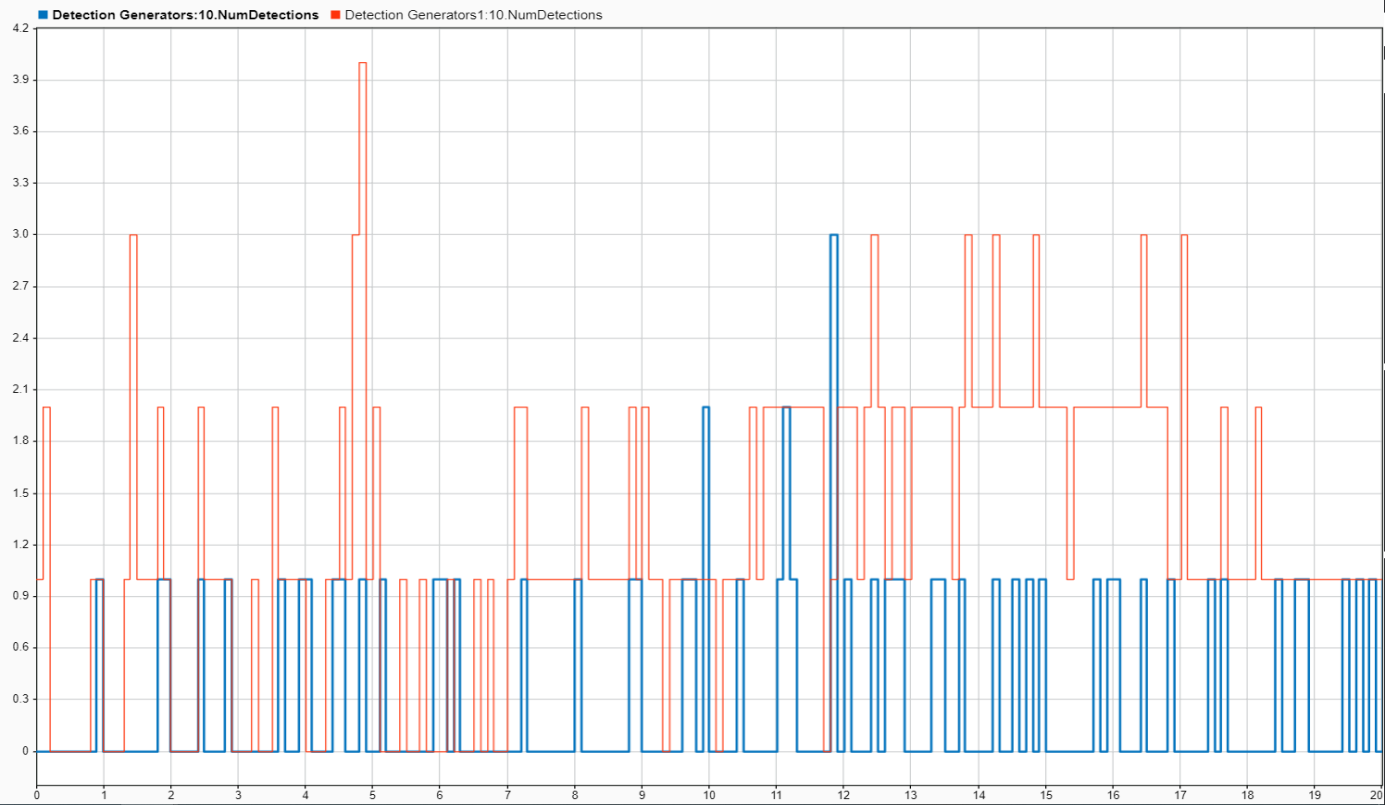
This figure shows the sensors placement



This is a simulink page, opening from matlab. In our scenario, signals starts from reader goes to actors which are camera and radar. But we dealed with only camera sensors in here by logging signals because in rainy weather, radar sensor has still good performance but not the camera. So, we compared camera sensors. After logging signals, we have taken the ouput from the Simulation Data Inspector.



From this figure, for Camera 2 model,blue box represent number of detections when the weather is sunny and the red box represents number of detections when weather is rainy.



This figure is similar to Camera 2, but it is Camera 4 model, this time orange box represent number of detections when the weather is sunny and the blue box represents number of detections when weather is rainy.

Briefly, we have created 2 conditions, which are low number object detection and high number object detection scenarios for camera and radar sensor.

In real life, we would have used 2 megapixel MegaPlus Interline Charge Coupled Device(CCD) Camera, High Frame Rate, 1920 x 1080 Pixel from eurocity dataset.

After much work on Matlab and Simulink, we have decided to work on python with Berkan (project member in a software team) and he leads us to camera/vision sensor. We collected the images of the EuroCity Persons dataset from a moving vehicle in 31 cities of 12 European countries. Recordings were made with a state-of-the-art automotive-grade two-megapixel camera (1920 x 1024) with rolling shutter at a frame rate of 20 Hz. The camera, mounted behind the windshield, originally yielded 16-bit color images; this high dynamic-range was important for capturing scenes with strong illumination variation (e.g. night-time, low-standing sun shining directly into the camera). EuroCity Persons benchmark, and for allowing comparisons with existing datasets, the original 16-bit color images were converted to 8-bit by means of a logarithmic compression curve with a parameter setting different for day and night.

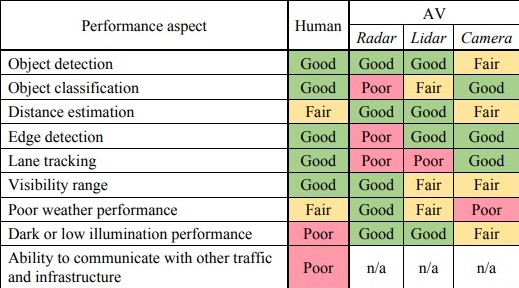
Results and Discussion

Our result indicates object detections in different weathers depends on quality and model of sensor. Camera sensor performs poorly in bad weather conditions or limited visibility. The photos demonstrate rainy weather and sunny weather comparison in different cities. In the results, we have figured out why and when camera sensor is poor. Because car wiper and rain drop bubble blocks and decrease number of the detections in rain. Other reasons are noise, degree of sensors, Detection Probability, False Positives Per Image

But for the noise, we didn’t change, kept as fixed)

Conclusion

We did research about different sensor models such as camera, radar and lidar in this sensor fusion Project. We finished the object detection from one sensor module mergeing data and tools that we have created and took. We have learned different performance aspects of the sensors from researchers. In rainy, snowy and foggy weather camera sensors camera sensor is not enough to detect objects



CDD:A charge-coupled device (**CCD**) is an integrated circuit containing an array of linked, or coupled, capacitors.

Rolling shutter: It is a method of image capture in which a still picture