

ILR - 05

Perception System Using Stereo Vision
and Radar

Team A - Amit Agarwal

Harry Golash, Yihao Qian, Menghan Zhang, Zihao (Theo) Zhang
Sponsored by: Delphi Automotive

November 23, 2016

Table of Contents

- 1. Individual Progress 1
 - 1.1 Delphi ESR 2.5 1
- 2. Challenges 4
 - 2.1 Data Understanding 4
- 3. Teamwork 4
- 4. Future Plans 5
- 5. References 5

1. Individual Progress

Since the previous progress review, I was tasked with working on the Radar and getting a more meaningful data visualization and gain further understanding of the messages from the Radar. For this I acquired PolySync software and worked with Zihao on getting a visualization on his computer. In addition, I did a detailed study of the literature about the Delphi ESR 2.5 provided by PolySync.

1.1 Delphi ESR 2.5

Over the past couple of weeks, Zihao and I had been working on getting sensible data from the Delphi ESR 2.5 Radar. After multiple challenges to obtain data from the Radar, PolySync provided a very reliable and simple method of visualizing and obtaining data. PolySync is an advanced middleware platform designed to help engineers build, test and deploy safe driverless vehicle applications quickly. A visual stream of real time data was obtained through the software. After some initial testing was conducted in basement of the Newell Simon Hall, the Delphi ESR 2.5 was mounted on the test car for further testing on the road. The test car used was a 2004 Volvo S60 2.5T AWD acquired by Harry. The Delphi ESR 2.5 was mounted in front of the grill of the test car. This was done using multiple zip-ties. The zip-ties were perfect for an initial round of testing as they were easy to fix to the curved and ribbed structure of the front grill without any surface study needed. The radar was inspected for how it responds to the forces experienced due to being at the front of the test car. It was determined to be quite robust to such forces. Once the Delphi ESR 2.5 was mounted, multiple routes of wiring were tested for the CAN and power connections with the Radar. Finally, wires were routed through the grill, under the hood, through the front passenger window into the car. The voltage generator was connected to the cigarette lighter socket using a power inverter with an AC output. The AC power was routed into a multi socket surge protector. Since the testing conducted was for the whole team, multiple power outlets were used to power the cameras, the radar, and the laptops connected to them. DC power was taken from the inverter using another adapter which provided 24 V as required for the Radar. The CAN outlet was connected to the laptop using a Kvaser Leaf Light v2. The first round of testing was performed overnight in Pittsburgh on roads near the campus for over 30 miles for at speeds from 20-30 miles per hour. Data was visualized on the PolySync Viewer using the PolySync Dynamic Driver. In addition, a simultaneous video of the road was taken so that visual inspection of the data can be performed later in the lab. A detailed inspection of the data while driving and after driving made way for understanding the tracking points that were obtained from the radar in the visualization. Another round of testing was then conducted to save the data obtained for this iteration. In this iteration, the car was run around 20 miles at speeds from 15-30 miles per hour around the campus, and the data was recorded twice for five minutes each. This data was brought back to the lab for further evaluation.

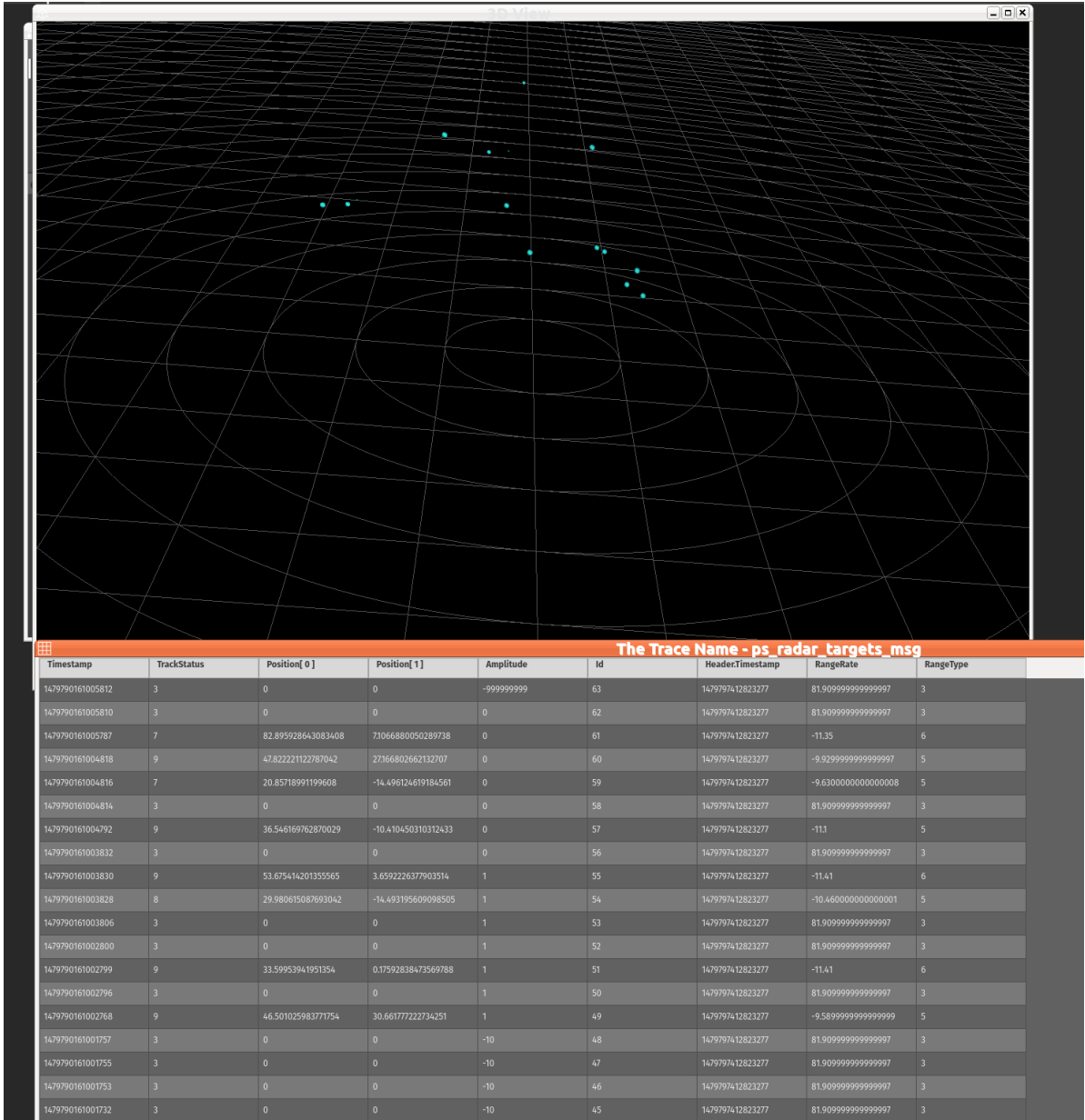


Figure 1. Radar Visualization Interface

The visualization was played and data values were inspected through the logging feature of PolySync. A variety of different data is provided in addition to just the location of the visualization for a running stream of data. The tables were inspected and even though it was easy to understand a couple of the data fields, a lot of data fields which changed in value over time contained important information. I conducted a literature study of the documentation available through PolySync. After reading multiple pages on the Publisher/Subscriber architecture, the Radar CAN documentation, I found the API documentation for PolySync. The documentation contained the description of the C++ class of the radar messages received by the name of *ps_radar_track_stream_msg* that was obtained from the radar. This contained a description of the fields inside the message class,

specifically the *position[0]*, *position[1]*, *rangeType*, *rangeRate* and *trackStatus* which helped describe the visualized points much easier. Figure 1 shows an example of the screen visualization obtained using PolySync with the data for that instant displayed in the table below the visualization. The data was better understood using the table like the one shown below in Table 1, obtained from the API documentation from PolySync.

TrackStatus		
0	TRACK_STATUS_INVALID	Track status information is invalid.
1	TRACK_STATUS_UNKNOWN	Track status information is unknown.
2	TRACK_STATUS_NOT_AVAILABLE	Status information on the track is not available.
3	TRACK_STATUS_NO_TRACK	Track is not valid and should not be considered usable.
4	TRACK_STATUS_RAW_TARGET	Track is a raw target measurement and is not tracked.
5	TRACK_STATUS_ACTIVE	Track is active/valid.
6	TRACK_STATUS_CIPV	Track is flagged as the closest in path vehicle.
7	TRACK_STATUS_NEW	Track is new to the tracker, ie the start of a track.
8	TRACK_STATUS_NEW_UPDATED	Track is new and was updated.
9	TRACK_STATUS_UPDATED	Track was updated.
10	TRACK_STATUS_COASTED	Track was coasted.
11	TRACK_STATUS_MERGED	Track was merged.
12	TRACK_STATUS_INVALID_COASTED	Track is invalid and was coasted.
13	TRACK_STATUS_NEW_COASTED	Track is new and was coasted.
14	TRACK_STATUS_KIND_COUNT	Number of ps track status kind values.

Table 1. TrackStatus details

After understanding the data well, a proper understanding of the axes on which the data was based was understood. The axes are shown with respect to the car in Figure 2 below assuming the Radar is mounted on the front of the car with the right side of the Radar mounted facing up. Using this the data was understood further and inferences were made about the tracks obtained from the Radar.



Figure 2. Data axes

2. Challenges

The major challenges faced were due to the message data provided by the Delphi ESR 2.5 being hard to parse and hard to understand. In addition, the API documentation was hard to locate.

2.1 Data Understanding

Obtaining the data apart from the visualization was not hard but learning how to do that was a little challenging due to the platform being new to me. Once the data was recorded and saved, the names of the fields in the Message class were not enough to properly describe the meaning of the data. There is no direct link to the API documentation of the *ps_radar_track_stream_msg* class. This was found after sending PolySync an email which gave a link to a page of the PolySync community forum answer, which had a link to the documentation. Apart from this, intuitively understanding the data was hard due to noisy nature of the data obtained from the Delphi ESR 2.5.

3. Teamwork

The work this week was performed well by all 5 team members. Zihao and I worked on the Radar and getting sensible data from it which can be visualized. After obtaining this data, we worked on obtaining and understanding the data provided by the Radar in quantifiable numbers. I was majorly responsible for finding and comprehending the required the API documentation, resources and instruction manuals while Zihao ran the Linux system on his laptop and was working with the command line interface. Yihao and Menghan worked on the Object Detection algorithms and literature study. In addition to that, they implemented and tested the stereo vision algorithms for a baseline of 1 m. Harry handled the prototyping of the mounting structure of the Radar and

obtaining a GUI for visualizing the stereo image data. Harry and Zihao together worked on developing the Radar mounting CAD for 3D printing. All members worked well and in a timely manner.

4. Future Plans

In the future, we plan to fabricate the final mounting structure for the Radar, and calibrate the baseline for 60 m target distance for stereo vision with the updated mounting structure for the cameras. I will be working on setting up the Radar to collect the data in sync with the Cameras and prepare the team for the FVE presentation.

5. References

[1] "Delphi ESR 2.5". PolySync Support Center. N.p., 2016. Web. 12 Nov. 2016.