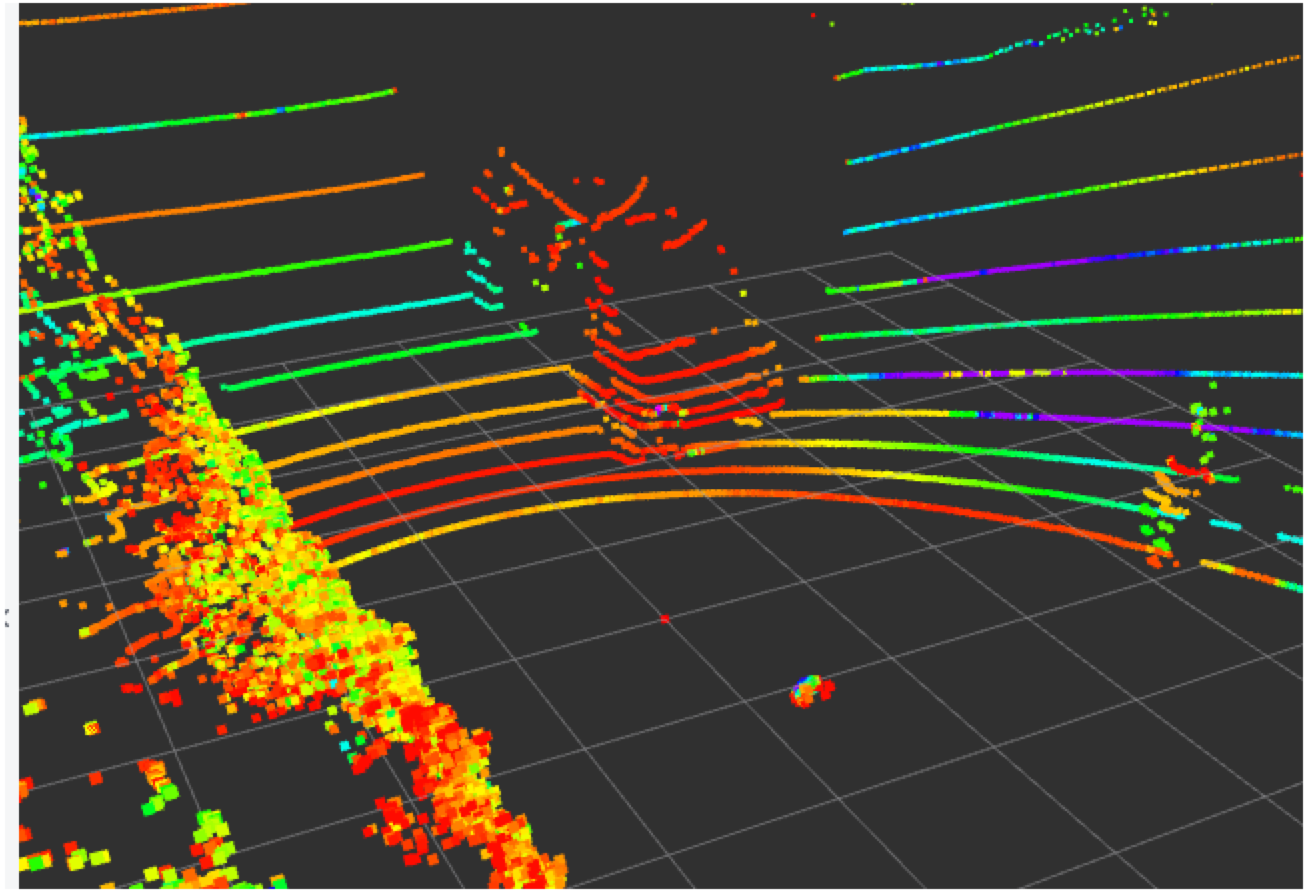
Sensor fusion – GPS & LIDAR

Setup

Aim: Get best possible estimate of distance b/w vehicles

Setup



Sensor information and variables

GPS

$f(x) = \text{Distance}(A_{GPS}, B_{GPS})$

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma_f} e^{-\frac{(x-\mu_f)^2}{2\sigma_f^2}}$$

LIDAR

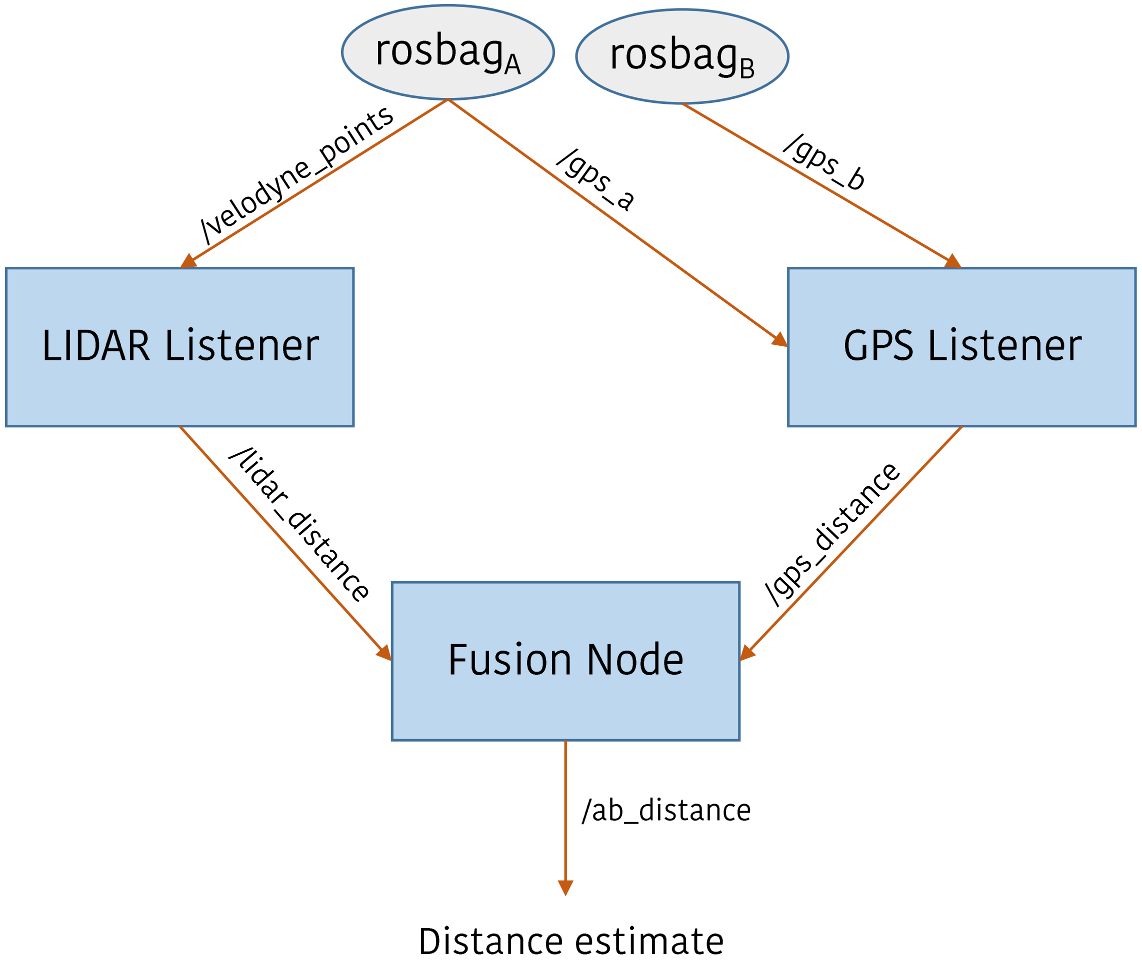
$g(x) = A_{LIDAR\_READING\_OF\_B}$

$$g(x) = \frac{1}{\sqrt{2\pi}\sigma_g} e^{-\frac{(x-\mu_g)^2}{2\sigma_g^2}}$$

Best optimal estimate is a scaled Gaussian PDF obtained from the product of f(x) and g(x) with standard deviation and mean:

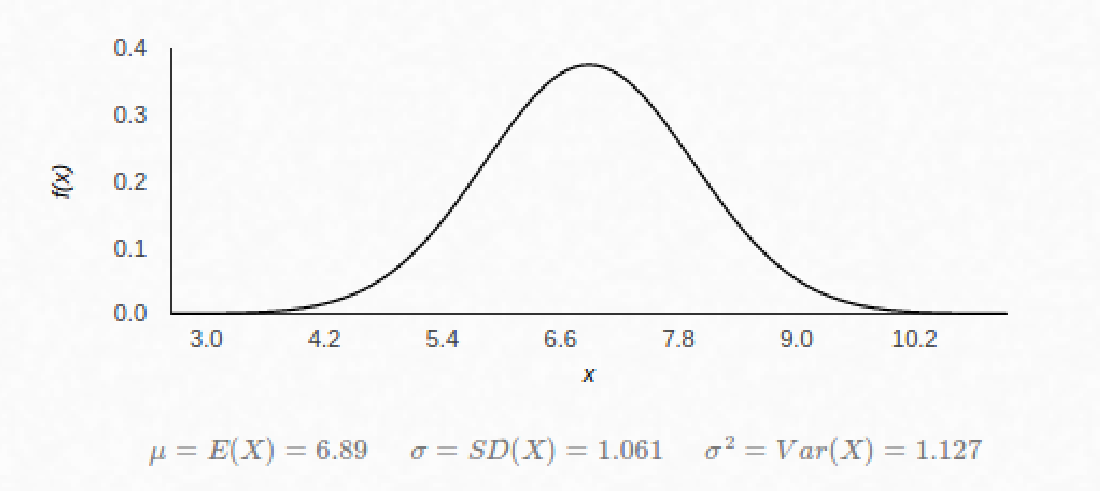
$$\sigma_{fg} = \sqrt{\frac{\sigma_f^2 \sigma_g^2}{\sigma_f^2 + \sigma_g^2}} \quad \text{and} \quad \mu_{fg} = \frac{\mu_f \sigma_g^2 + \mu_g \sigma_f^2}{\sigma_f^2 + \sigma_g^2}$$

ROS Package



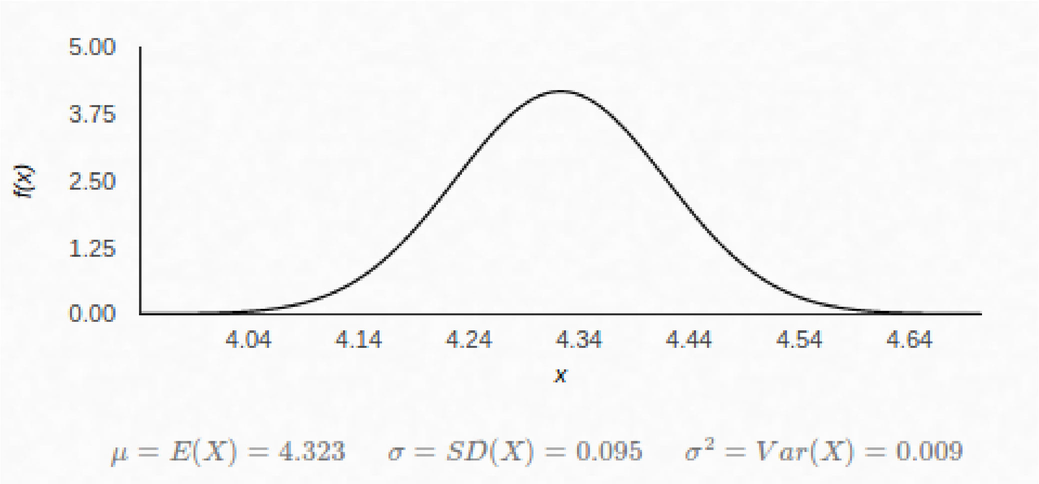
rosbag$_A$
rosbag$_B$
/velodyne_points
/gps_a
/gps_b
LIDAR Listener
GPS Listener
/lidar_distance
/gps_distance
Fusion Node
/ab_distance
Distance estimate

Results - GPS



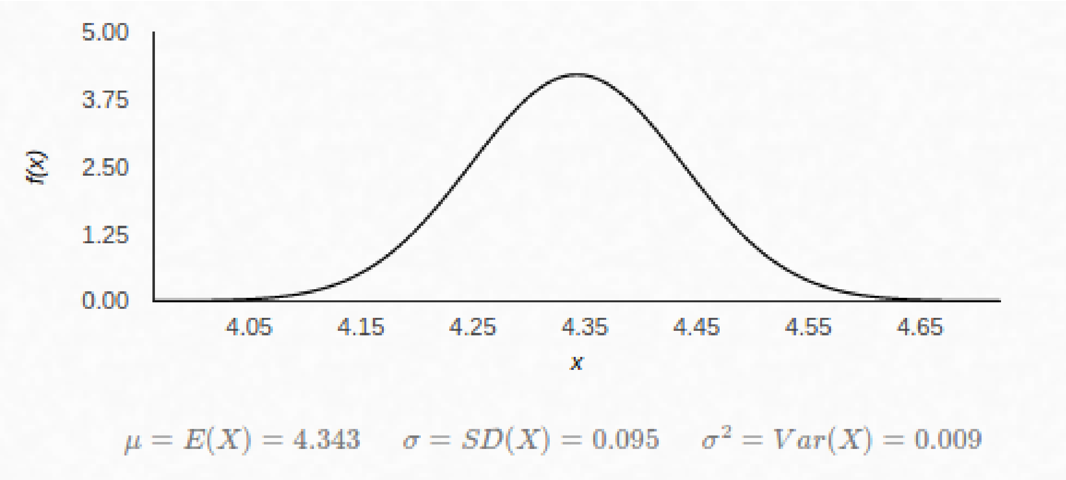
Mean = 6.89
Standard deviation = 1.061

Results - LIDAR



Mean = 4.323
Standard deviation = 0.095

Results – Fused information



Mean = 4.343

Standard deviation = 0.0946

Code and presentation

All code has been open-sourced at Github.

This presentation can be found online here.

References

[1] Computing the distance between two locations on Earth from coordinates, John D Cook
[2] Products and Convolutions of Gaussian Probability Density Functions, P.A. Bromiley
[3] How a Kalman filter works, Tim Babb
[4] Understanding the Basis of the Kalman Filter via a Simple and Intuitive Derivation, Ramsey Faragher