**שיעורי בית – תרגיל ב – EKF**

בתרגיל זה ננסה לעקוב אחרי תנועתו של הולך רגל ההולך בתאוצה לא ידועה. אנחנו מקבלים מדידות מהולך הרגל משני חיישנים שונים – LIDAR המספק לנו (דרך ענן נקודות) את מיקום הולך הרגל בקורדינטות X-Y ודרך RADAR המספק לנו את המרחק של הולך הרגל מאיתנו, הזווית בינו לביננו והמהירות שלו בכיוון הוקטור הזה.

נראה את החשיבות של הכנסת התאוצה כמודל במשוואות ואת החשיבות של הוספת שני חיישנים לקבלת תוצאה טובה יותר.

למידע המסופק בקובץ, נוסף גם מיקומי האמת של הולך הרגל על מנת שאפשר יהיה לחשב את השגיאה דרך RMSE. אני מזכיר שוקטור המצב שלנו מכיל את מיקום הולך הרגל בשני הצירים ואת מהירותו בשני הצירים.

שאלות:

1. מלאו את המטריצות במחברת הפייתון על מנת לאפשר וקטור מצב של 4 – מיקום בציר X-Y ומהירות בציר X-Y כמו שלמדנו בכיתה.
2. עכשיו, השתמשו בקובץ המצורף והתייחסו לחיישן הלידר בלבד. רק דרך חיישן ה-LIDAR בלבד ותוך הנחת מודל מהירות קבועה (ללא תאוצה), חשב את מיקומו של הולך הרגל דרך משוואות קלמן. מה הטעות RMSE?

matrix([[7.41826686, 8.67626569, 4.05611556, 3.85419828]])

1. כעת, הוסף מודל תאוצה לא ידועה (מטריצת Q). עבור הרעש בתאוצות בציר הX והY הצב 9. חשב את הטעות RMSE.

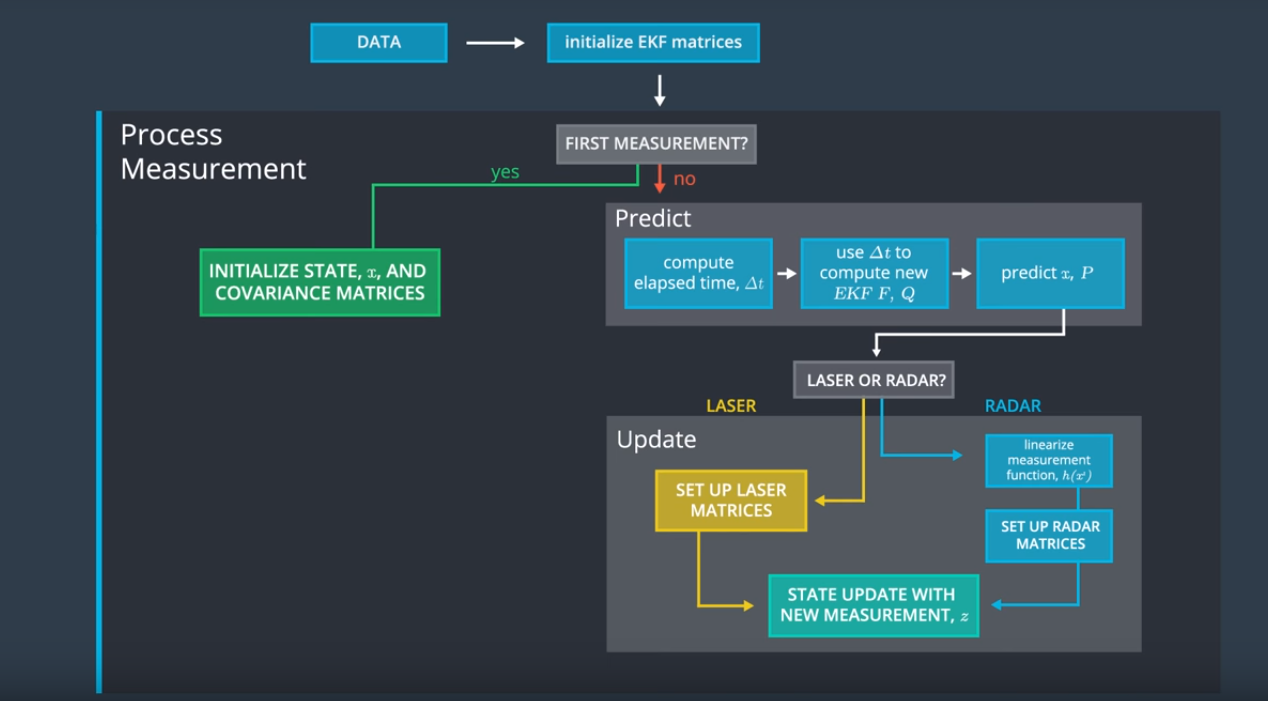
matrix([[0.12219136, 0.09837984, 0.58251275, 0.45669849]])

1. כעת הוסף את מדידות הRADAR. במה השתפרה הטעות RMSE.?

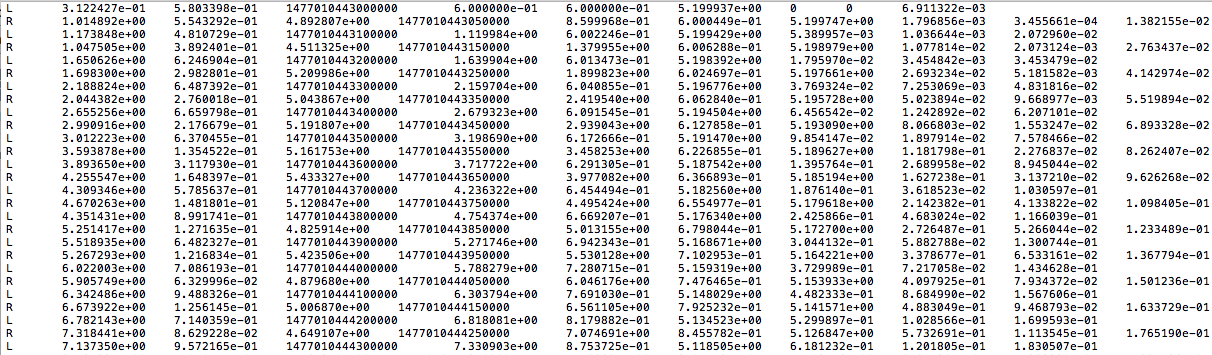
matrix([[0.09722562, 0.08537612, 0.45085468, 0.43958819]])

Plan for solution:

* **initializing** Kalman filter variables
* **predicting** where our object is going to be after a time step Δ*t*
* **updating** where our object is based on sensor measurements



Here is a screenshot of the first data file:

**[[](https://classroom.udacity.com/nanodegrees/nd013/parts/40f38239-66b6-46ec-ae68-03afd8a601c8/modules/0949fca6-b379-42af-a919-ee50aa304e6a/lessons/f758c44c-5e40-4e01-93b5-1a82aa4e044f/concepts/45d7787c-0b2b-455d-836e-e53b22972586)](https://classroom.udacity.com/nanodegrees/nd013/parts/40f38239-66b6-46ec-ae68-03afd8a601c8/modules/0949fca6-b379-42af-a919-ee50aa304e6a/lessons/f758c44c-5e40-4e01-93b5-1a82aa4e044f/concepts/45d7787c-0b2b-455d-836e-e53b22972586)**

**[Screenshot of Data File](https://classroom.udacity.com/nanodegrees/nd013/parts/40f38239-66b6-46ec-ae68-03afd8a601c8/modules/0949fca6-b379-42af-a919-ee50aa304e6a/lessons/f758c44c-5e40-4e01-93b5-1a82aa4e044f/concepts/45d7787c-0b2b-455d-836e-e53b22972586)**

Each row represents a sensor measurement where the first column tells you if the measurement comes from radar (R) or lidar (L).

For a row containing radar data, the columns are: **sensor\_type, rho\_measured, phi\_measured, rhodot\_measured, timestamp, x\_groundtruth, y\_groundtruth, vx\_groundtruth, vy\_groundtruth, yaw\_groundtruth, yawrate\_groundtruth**.

For a row containing lidar data, the columns are: **sensor\_type, x\_measured, y\_measured, timestamp, x\_groundtruth, y\_groundtruth, vx\_groundtruth, vy\_groundtruth, yaw\_groundtruth, yawrate\_groundtruth**.

Whereas radar has three measurements (rho, phi, rhodot), lidar has two measurements (x, y).

You will use the measurement values and timestamp in your Kalman filter algorithm. Groundtruth, which represents the actual path the bicycle took, is for calculating **root mean squared error**.

Yaw and yaw rate will be introduced in the UKF lecture (you will use the same data set for the UKF project). You do not need to worry about yaw and yaw rate ground truth values.

The radar Cov matrix is:

Radar<< 0.09, 0 , 0

0, 0.0009, 0

0, 0, 0.09

The lIdar cov matrix is:

Lidar<< 0.0225, 0

0, 0225