PROJECT #2

Parts A and B – Sensor Data Fusion / EKF Localization
Part C – System identification via optimization

Introduction

Relevant part of Project 2 is focused on implementing Sensor Data Fusion, in a Bayesian fashion, via applying the EKF approach. For that purpose, we continue working in our case of study: “localizing a platform”. We have seen, in Project 1, that there are diverse sources of information, e.g. provided by sensors' measurements and from a process model such as the kinematic model of the platform. Now, we want to combine those sources of information in a consistent way, in which qualities of those sources of information are considered, for generating estimates of the variables of interest, i.e. the state of our system, the vehicle's pose. Those estimates are required to be permanently provided (i.e. frequently and at any time when required). The estimation process will generate optimal estimates in the form of an expected value and a covariance matrix. The estimation process will operate in real-time, and it will require low processing cost; all those are characteristics which are relevant in industrial and critical applications.

A second part of the project focusses on exploiting optimizers for tuning model parameters, which is a relevant problem in many applications in which we know the structure of the plant's nominal model, but we do not accurately know certain model parameters. That task involves defining a proper cost function and then exploiting it, by using an optimizer, for estimating the unknown parameters.

Part A requires implementing an EKF based localizer based on kinematic model and LiDAR measurements. For solving this part, you will modify your solution implemented for Project1 (or create a new one, from scratch if you prefer).

Your program will maintain estimates of the vehicle pose, i.e. an expected value and an associated covariance matrix.

You will perform the prediction step in the same way you used to run your kinematic model, in Project1. In that way you will always maintain expected value and covariance matrix, statistically describing the vehicle pose, at any time.

In addition, you will perform update steps each time at which observations are available, i.e. when you process a LiDAR measurement, in which you detect OOIs from which you are able to associate some of them with map landmarks. The update will exploit measurements of range (and of bearing if

required). Those range and bearing measurements, are the same you already exploited in Project1.PartD.
The data association (DA) component will run in the same way you used it in in Project 1; however, in this case, the DA will be based on the prior expected value, before the actual update is applied.

The datasets to be used are not “friendly” such those datasets used in Project 1. The datasets used in Project 2 do contain measurements which are polluted by noises, which are initially assumed to be WGN (although some of them are not), and whose variances are known. LIDAR’s measurements are polluted with noise, in addition to the quantization error and limited angular resolution. Speed measurements and gyroscope measurements are also polluted by noise, in addition to the small quantization errors which were present in the datasets used in P1.

Part A1. Implement the EKF localizer, exploiting only the range observations. Verify that the estimation process is able to estimate all the vehicle states (pose), including the heading.
Comment: this case is similar to that in which we do not have a LiDAR but we have a sonar sensor, which is able to measure distances to OOI, but no angles.

Part A2. Modify your solution in Part A1, to implement a localizer which only process bearing observations (similar to a camera based one), i.e. not processing range observations.

Part A3. From the implementations in A1 and A2, implement a localizer able to process all available observations (range and bearing measurements, from multiple OOIs.)

Use datasets named Data19b.mat”, Data20b.mat”, Data21b.mat”.
Assume the following characteristics for the noise which polluted the sensors’ measurements

Speed measurements:	standard deviation: 5cm/second;
Gyroscope measurements:	standard deviation: 1 degree/second
OOIs range observations: standard deviation: 10cm	
OOIs bearing observations: standard deviation: 2degrees	

Part B requires implementing an EKF based localizer, based on kinematic model and LiDAR. In addition to estimate the system states (vehicle’s pose), the process will estimate the bias, “**b**”,’ which pollutes the gyroscope measurements.

This part of the project can be implemented by modifying part A.

In addition to the assumptions considered in part A, in part B you will assume that the gyroscope's bias is constant (but unknown); we also know that the bias value is limited to be in the range from -1. to +1 degrees/second. You are not intended to apply any off-line calibration approach for solving this part. We assume the vehicle never stops for allowing that type of calibration to be applied.

Your program must provide visual indication of the value being estimated. That visualization can be given during the operation of the program (in "real-time"), or at the end of the program (plotting the evolution of the bias expected value against the time).

Part C Propose an off-line approach, based on optimization, for jointly estimating two unknown system's parameters: "Lx" (LiDAR longitudinal displacement), and the gyroscope bias, "b". You can use the optimizer you prefer, from the suite of optimizers offered by Matlab. The approach must be able to work with any of the provided datasets, individually.

You will assume that you know that parameter Lx is in the range of values, from Lx=0 to 1.2m.

In your solution, you will use the information provided by the sensors' measurements (speed sensor, gyroscope and LiDAR) and by the map of landmarks. You will not use the provided ground truth for that purpose (because in real-life applications you may not have that information).

Hints about how to solve this problem: You need to define an adequate cost function, ***C(b , Lx)***, to be minimized by the optimizer. A possible basic cost function may be as follows:

For each hypothetical set of parameters, predict the vehicle's pose at certain time t0+T (being the length of T of few seconds) and exploit the discrepancy of the predicted global position of the detected OOIs and the position of their associated landmarks.

You are not intended to do the tuning manually, you are intended to solve it via optimization, as the objective of part C is to get experience using optimizers for solving more challenging problems.

Part C requires submitting a brief report (one to three pages), for explaining your approach

Additional resources for solving Project 2:

If your modules for OOIs detection and for Data Association (produced for solving Project 1), do not perform satisfactorily, you may use the solutions we provide (an API, in binary format, p-code), which you can call from your program. We expect that some other necessary components, such us the implementation of the kinematic model, must be solved by the students (you solved it in Project1.PartA). If you consider that your implementation of that component has not been successful, contact the lecturer.

Questions about this project: ask the lecturer via Moodle or by email (j.guivant@unsw.edu.au)

The lecturer will show a possible solution working, during lecture time, on week 8 (for parts A and B) and week 9 (for part C).

Part A, B and C are worth 40%, 30% and 30%, respectively, of Project 2 overall mark.

Details about submission of the project will be given in lecture 9.
Submission will be open from the end of week 9 till the end of week 10 (Week 10, Friday, 23:55).