MTRN4010.2023 PROJECT #2

Sensor Data Fusion / EKF Localization
System identification via optimization

Introduction

A substantial 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., those 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 is focused on proposing and implementing an EKF based localizer, which exploits the platform's kinematic model, the LiDAR measurements and an a priori map of surveyed landmarks (the same sources of information you used in project 1).

For solving the implementation component of Part-A you may modify your solution 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 addition, you will perform update steps each time at which observations are available, i.e., when a LiDAR measurement is processed, in which OOIs are detected. Some of those OOIs will be associated with map landmarks. The update step will exploit measurements of range to OOIs. Those range measurements are the same you may have already obtained and exploited in Project1/Part F.

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 used in Project 2 do contain measurements which are polluted by noises, which are assumed to be WGN (although some of them are not, strictly), and whose variances are known. LIDAR's measurements are polluted with noise, in addition to the quantization error and the 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 A is composed by 3 subparts, which can be solved independently.

Subpart A1. Implement the EKF localizer, exploiting only the OOIs' range observations. Verify that the estimation process is able to estimate all the vehicle states (pose), including the heading.
(Note: 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 OOIs, but not their angular coordinates.)
Test your solution using the full map (defined in 'data.Context.Landmarks')
In addition, test your solution using the sparse maps (defined in 'data.Context.Landmarks2' and data.Context.Landmarks4'). Those sparse maps correspond to cases in which not all the landmarks have been surveyed, so that those maps only provide information about subsets of the deployed landmarks. Based on those sparse maps you will test your solution in more challenging conditions.
Your solution must be able to localize the platform, achieving a performance well better than that of a pure kinematic based one.
Use the datasets named "**Data06k.mat", "Data07k.mat"** and "**Data08k.mat"**. Students who have interest may request additional datasets after trying the initially released datasets.

Assume the following characteristics for the noise which polluted the sensors' measurements:

Speed measurements: standard deviation: 10cm/second.
Gyroscope measurements: standard deviation: 4 degree/second.
OOIs range observations: standard deviation: 25 cm.
OOIs bearing observations: standard deviation: 3 degrees.

Part A2. It requires the proposal (but not implementing it in a program) of an EKF based localizer, based on kinematic model and LiDAR. However, in this case we exploit the full capabilities of the LiDAR (i.e., it is not operating as sonar but providing full 2D information). Your proposal must detail the necessary equations for implementing all the parts of the EKF update steps.

Your proposal must the assume the following:

1) Measured output variables (used in the EKF update step): 2D position of OOIs (OOIx, OOIy) in LiDAR's cartesian coordinate frame (LiCF). It is specified, in this part, that **we are not allowed** to convert those variables to other representations (e.g., polar or to cartesian in other CFs), when we use them in the EKF update. They must be processed in their original representation, consequently the observation models (aka output equations) must correspond to those physical variables.
2) We assume that the measurements of OOIx and OOIy are polluted by uncertainty (noises) which behave as GWN, with standard deviation of 20cm for both measurements. The uncertainty that affects OOIx is independent of that of OOIy (we assume both noises are WGN.)
   Those assumptions are adequate for covering the uncertainties introduced by the LiDAR and by your feature extraction process (from Project 1).
3) States being estimated: 2D pose of the platform in the GCF.

Additional assumptions:

a) The platform's initial pose is accurately known.
b) Process model inputs are polluted as described in part A1.
c) There are no other relevant uncertainties affecting the process model.

Part A3. It requires the proposal (but not implementing it in a program) of an extension of your EKF defined in part A1 (or that of part A2 if you prefer). The required modification has the purpose of estimating, in real-time, certain parameters of the system, which are constant but initially not accurately known to us. The parameters are the LiDAR displacement (Lx,Ly) (the same you had considered in Project 1 and in parts A1 and A2 of this project. In this case as the parameters are not well known, we need to "fine tune" them. The parameters are assumed to have these characteristics: Lx: nominal value =30cm, but we know it may be a constant value between 15 and 45 cm.

Ly: nominal value =10cm, but we know it may be a constant value between 0 and 20 cm.

You are required to propose adequate adaptations for making the localizer able to estimate the platform's pose and the parameters Lx and Ly, simultaneously, in real-time.
We may assume that the platform's initial pose is accurately known.
We assume that the overall estimation process will be observable, so that we propose the adaptations not being concerned about the system's observability.

Relevance of subparts in part A: A1:40%, A2=30%, A3:30% (of Part A)

Marking criterion for Part A1:

Implementation of necessary equations: 20% of Part A1
Implementation Working (close to accuracy and consistency specs): 20% of Part A1
Accuracy 30% of Part A1
Consistency 30% of Part A1

Consistency: In all the cases (datasets and map densities)
When analyzing each marginal PDF, your consistency plots must indicate that the discrepancy is lower than twice the estimated standard deviation, during at least 80% of the trip.

Accuracy:
The maximum discrepancy (|actual value – expected value|) must satisfy that:

For full map (for all the datasets and for the full duration of the trip.)
For X and Y, the maximum error must be <0.4m.
For heading the maximum error must be <3 degrees.

For 50% density map (for all the datasets and for the full duration of the trip.)
For X and Y, the maximum error must be <0.45m.
For heading the maximum error must be <3.2 degrees.

For 25% density map (for all the datasets and for the full duration of the trip.)
For X and Y, the maximum error must be <0.55m.
For heading the maximum error must be <4 degrees.

Consistency and accuracy will be evaluated only if “consistency plots” are produced. Students may use the provided API for that purposed.
If no consistency plots are produced, no marks will be given in these marking items.

Definition:
The “duration of trip”: the interval of time starting at the time at which the platform begins to move, i.e., excluding any initial period during which the platform was not moving; however, any subsequent stopping periods are not excluded.

“density map”: Available landmarks, in which 100,50 and 25 of the original landmarks are available.
“25%” means that ¼ of the originally deployed landmarks are included in the given map (name of map: Landmarks4)
50%” means that ½ of the originally deployed landmarks are included in the given map (name of map: Landmarks2)
“100%” means that all the originally deployed landmarks are included in the given map (name of map: Landmarks)

Consistency plots: Refers to those of the marginal PDFs, and not to the joint PDF. Consistency plots are to described in lecture 8, when the lecturer shows his results for Part A.
“marginal PDFs”: Although there can be many other subsets of variables, in our “consistency plots” we focus on the marginal PDFs about the individual scalar components of the state vector.

Marking criteria for Part A2.

1) Prediction step 0% (necessary, it must be properly defined, -20% if not properly solved)
2) Correct observation function 45%
3) Propose correct H matrix 45%
4) Correct R matrix 20%

Marking criteria for Part A3.

1) Correct state vector 0% (An incorrect vector will invalidate the rest of the items)
2) Complete Prediction step 10% (of full mark of A3)

3) Initialization of expected value 10% (-20% if not solved)
4) Initialization of P matrix 10% (-20% if not solved)
5) Observation function 25%
6) H matrix 25%
7) R matrix 5%

Notice: A3 can be solved based on the approach used in A2 or on that of A1 (which is the same one given in class). This imply that not having solved A1 and A2 is no reason for not solving A3.

Part B. You are required to propose and to implement an off-line approach, based on optimization (i.e., not based on EKF or other similar online approach), for estimating an unknown parameter: The gyroscope's bias. For that purpose, you can exploit any data component offered in our dataset files, including the ground truth of the platform's poses, LiDAR scans, initial pose, etc. You decide what data components are useful, and how to use them in your solution. This optimization process will perform automatically what you have done manually/visually in Project 1, part E1.

Subpart B1: Proposal of the approach. Details about your approach will be given in a report (in a section in the Project main report). Your ideas must be explained via text and via mathematical expressions, i.e., all necessary equations and models which would be used in an actual implementation, independently if you implemented it or not, in subpart B2).

Subpart B2: Requires you to implement the approach you proposed in B1. For verifying the performance of your approach, you will use the dataset you used in Part E of Project 1.

Your program must run in the following way:

Program input: name of the dataset file to be processed.

Program outputs, after completing the parameter estimation (all the described outputs must be produced by your program)

1) Your program will print the estimated value of the bias.
2) In a figure, it will plot:
    a. The position component of the predicted poses, assuming bias=0, in green color.
    b. The position component of the predicted poses using the estimated bias, in red color.
    c. The position component of the ground truth poses, in blue.
    d. The walls of the context infrastructure (as you did in Project 1)

Additional requirements: The tuning process must take less than 5 minutes to complete the optimization.

Assumptions: The bias is constant. Possible values of the bias are in the range from -2 to +2 degrees/second.

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

Marking criteria for Part B.

For part B1. Approach and report. 50% of total mark of part B

The report must explain your approach, using text and properly expressing concepts in mathematical terms (equations), in relevant parts. Although it is not unique, a correct cost function must be proposed. An incorrect cost function will result in part B1 receiving 0 marks.

For part B2. Implementation and accuracy: 50% of total mark of part B

Your implementation must follow the proposal presented in part A1. If not, this part will receive no marks.

When being initialized, assuming the nominal bias =0, the estimation process must take less than 3 minutes to converge (in our lab computers). The estimated parameter must be not more than 0.1 degrees/second away from the real one (you can test it, with the dataset used in Project 1/PartE1)

Visual outputs: Printing the estimated parameters, and plots at the end of estimation: Those do not provide marks but are necessary to evaluate the effectivity of your approach. If those are not provided, the process will be considered not working (thus receiving 0 marks).

Suggestion: you may test your approach as many times as you want, just using other datasets that are free of bias, by adding a fictitious bias and trying your optimizer to infer it.

Additional resources for solving Part A1 of 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 should have solved it in Project1.PartA and in related tutorial problem; if you did not, you will need to solve it for being able to try project 2/Part A1.)

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.

Details about submission of the project will be given in lecture 8.

Submission will be open from the end of week 9 till the end of week 10 (Week 10, Friday, time 23:55).

Late submissions: Work submitted late without an approved extension by the course coordinator is subject to a late penalty of **15%** (of the maximum mark possible for this assessment) per calendar day. The late penalty is applied per calendar day (including weekends and public holidays) that the assessment is overdue. There is no pro-rata of the late penalty for submissions made part way through a day. Work submitted after five days (120 hours) will not be accepted and a mark of zero will be awarded for that assessment item.

(End of document)