ENGO 585: WIRELESS LOCATION

GEOMATICS ENGINEERING

SCHULICH School of Engineering

LAB #2: ESTIMATION

OBJECTIVES: To review non-linear least-squares estimation and implement sequential least-squares and a very simple Kalman filter.

DATE: Friday, January 29, 2021

DATE DUE: Thursday, February 11, 2021

This lab may be completed individually or in groups of two

INTRODUCTION

In this lab you will use very simple range observations to compute a position using Least-squares and a very simple Kalman filter. The purpose of this lab is to gain some insight into the procedure and challenges of land-based wireless location using TOA/TDOA methods. You should keep in mind that this is a simplified exercise. For example, atmospheric and multipath effects have been ignored.

TASKS

Task 1: Parametric Least Squares

The file ***Lab2data.txt,*** on D2L, contains the 150 epochs or range observations observed by a user to four targets with (x,y) coordinates of (0,0), (100,0), (100,100) and (0,100). Target coordinates and ranges are in metres. The format of each row of the file is time (s) , range 1, range 2, range 3, range 4 (in metres).

From the data, it should be pretty obvious where the user is located and also if the user is static or moving. Determine (from looking at the data) if the user is stationary for any of the epochs in the data set. (Hint, the user is stationary for some then moving for others).

You will recall from ENGO 361 and 419 that a non-linear least-squares problem such as this requires an initial position estimate and the solution must be iterated by re-evaluating the misclosure vector and design matrix as your position estimate converges. You may assume any initial position as your point of expansion. Assume the ranges are uncorrelated with each other and have equal, but unknown, measurement standard deviation.

1a. Compute the 2-D position solution for each epoch (ie for each row of the file). Plot the results.

1b. For the epochs where the user is stationary, compute the batch parametric least squares position solution (ie use a bunch of measurements together to get a more precise estimate).

In order to compute the covariance matrix of the position solution, you will need to determine the *a posteriori* variance factor of the batch solution. From this value, determine an estimate of the measurement standard deviation then plot the error ellipse of the batch solution (using the information in the cofactor matrix scaled by the variance factor).

1c. Plot the residuals of one of the range measurements. Discuss. Do they look like residuals should, particularly given your estimate of the a posteriori variance factor?

1d. Plot the 2D error ellipse for any single epoch solution from the part of 1a on top of the 2D position solution from part 1a (during the period where the user is stationary). How many of the single epoch position solutions fall within this error ellipse? Does this make sense?

Task 2: Summation-of-Normals and Sequential LS.

2a Repeat Task 1b using summation of normal instead of batch parametric least squares.
2b Repeat Task 1b using sequential least squares.

Discuss the results. Do they match the batch result as expected?

Task 3: Kalman filtering

3a. Test the code you have already developed for 2b Sequential Least Squares on the entire data set (ie the static part and the kinematic part). What happens? Does it work? If not, why not? Plot the results.

3b. Convert your Sequential LS solution into the simplest possible Kalman filter. Do this by adding process noise, Q, to the covariance of the estimate after each epoch. This is called a "Random Walk" dynamics model and will be discussed in class. Experiment with different values of Q varying from zero to large. What happens to your solution in each case you test? Plot the results for at least 3 different values of Q and discuss.

3c. Add two velocity states to your filter. To do this you will have to add two more columns to any design matrices that should be columns of zeros (because the ranges don't observe the velocities). You will have to make the covariance matrix of the states larger, and the Kalman gain matrix K will also change dimensions. Implement a constant velocity model transition matrix. This will be a 4 by 4 matrix analogous to the 2 by 2 example given in the notes. The measurements don't change, you still only have ranges, but it should be possible to estimate the velocity states over time. At first they will be near zero but when the user starts moving, the velocity states should estimate the velocity components. You may set the initial values of the two velocity components to zero and their initial variances to some large number indicating that you don't know a priori how fast the user is moving. Compare the estimated velocity states to the slopes of the position estimates as a function of time obtained using single epochs of data and least squares (from Part 1a.)

MATERIAL TO BE SUBMITTED

Lab materials should be submitted to the D2L dropbox for the lab.

Your report should include: Position solutions, a posteriori variance factor, measurement standard deviation estimate, All least-squares and Kalman filtering source code in C++ or Matlab (plotting should be done using Matlab).

This is not just a numerical exercise. Methodology, discussion and analysis are required in each section. Your lab report as a whole should be easy to understand, well written, and professionally presented (ie. sections should have heading, figures should have captions etc.) and should enable someone else to reproduce you results.