Location Determination

Shubham Kumar

1. Problem Description 3

1.1. Constraints 3

2. High Level Design 3

2.1. TOA 3

2.1.1. Solving the TOA equations 4

3. Test Case analysis 4

3.1. Car A 4

3.2. Car B 5

4. Filtering 6

4.1. Measurement Filtering 6

4.2. Position Filtering 6

# Problem Description

The problem statement requires locating user position (in a car) based on radio signals emitted by the user and received by base stations located around the user position.

## Constraints

The following constraints are specified as part of the problem

* The base stations are synchronized
* User dynamics are constrained to a car
* The user position is in a plane
* Base stations locations are fixed and provided
* No other information is available (frequency, Doppler, sensors)

# High Level Design

## TOA based system

In the system, there are 3 variables – User Position (x,y) and user bias (t) since the receiver clock is asynchronous to base station clock. Given that there 4 base stations are monitoring the transmitted signal, there will be up to 4 time-of-arrival measurements available in an instant. Each of these TOA measurements correspond to a pseudo-range

T1

T2

T3

T4

## Solving the TOA equations

This system of 4 psuedo-ranges and 3 unknowns (x,y,t) can be solved using a least-squares estimation or as time difference of arrival. The TDOA approach has the advantage of eliminating user time bias but should be equivalent in terms of performance.

Let Tt be the time of transmission from the user and Ta be the time of reception at station 1. We can also assume Tt to be roughly synchronized to the station time with an offset

The pseudo-range can now be specified as:

Here, the first term is the range with respect to station 1.

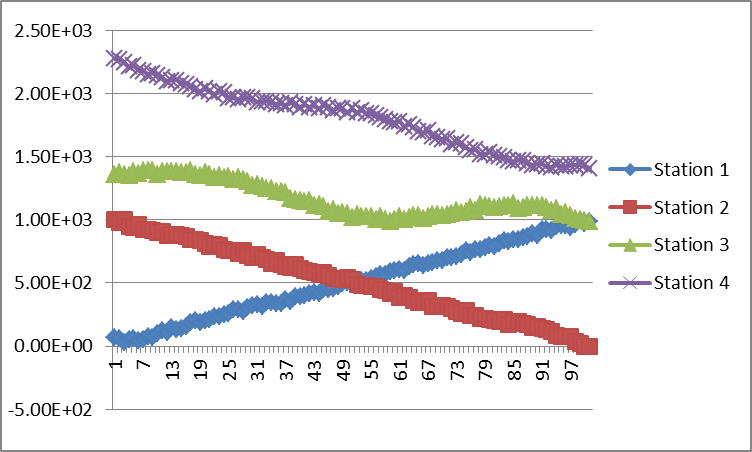
With 4 measurements, we have an overdetermined system with 3 variables. The LS approach is taken here to solve for the user position

# Test Case analysis

There are two test cases given here. A simple TOA analysis of the two test cases is done to obtain relevant information about the system

## Car A

A time plot of the pseudo ranges of each of the stations shows a clean data without any anomalies



## Car B

Time plot of Car B data shows interesting characteristics. It can be seen that the pseudo-range for each of the stations has unusual jumps

* Big jumps of 6000m
* Small jumps of 1000m
* Missing data for certain epochs i.e not every epoch has 4 measurements

Given that we have established the user dynamics to be constrained to a car, jumps of 5KM+ are unlikely. For a car going at 100Mph, the expected change of pseudo-ranges is likely to be in the range of 50m only. Given this, the likely explanation is errors in recording TOA at the base stations. The smaller jumps while cannot arise from user dynamics as well but could arise from multi-path reflection.

# Filtering

Given the data in the previous section, it is advisable to implement some filtering on the input measurements and also on the output position reports for a smoother user experience.

## Measurement Filtering

One simple measurement filtering is to monitor instantaneous changes in ranges and reject large jumps. Care should be taken that this is not a common jump across all the stations – in which it is likely a user clock jump. However, sample data does not present such a scenario.

The key assumption for implementing this is that the user starts in a clean environment and the first fix is free from errors.

## Position Filtering

There are many possible approaches for filtering the position output. A popular approach is to use a Kalman filter, which in our case would require us to implement as a 6-state filter [x,y,tb, Vx, Vy, td] – in addition to user position and bias, we would also include user velocity and time drift. However, in the absence of any frequency information, we have to either assume a constant acceleration model or fall back to a 3-state vector with constant velocity model. Since we do not know have this information, Kalman filter approach is not suitable.

Instead, since we have chosen the LS method for computation, a simple Sum-squared-error (SSE) filter is implemented here.

Where K is derived from SSE. A high SSE indicates poor convergence and the new position should be de-weighted whereas a small SSE indicates good convergence and a higher weightage.