# Experiment 1:

1. There are 5 input files:
   1. The first file contains 20 epochs, is part of the easy suite.
   2. The other four contain 900 epochs (15 mins) each, downloaded from <ftp://cddis.nasa.gov/gnss/data/highrate>.
2. For each file, I randomly select N\_Trials=10 of the following errors:
   1. Assistance error in position: uniform azimuth in 0-360 Deg, uniform distance in 0-3 Km.
   2. Assistance error in time: uniform integer number of milliseconds 0-2000
   3. Random common bias added to all pseudo-ranges.
3. For each epoch in each file I calculate a Coarse-Time-Nav fix (from scratch), one for each set of errors described in ‎2. Repeat this section once with Diggelen’s original algorithm, and once with my fixed algorithm.
4. I calculate the distribution of position estimation error, and the distribution of error in assistance-time error estimation.

Plots:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# Experiment 2:

Now assume we use Coarse Time Navigation to get a state vector that minimizes the residuals of the problem in the least-squares sense.

In our interpretation of the problem, is the relevant transmission time of the nearest satellite, and we estimate by . From the nature of the least squares solution, in the general case **will not be an integral millisecond**, although it is known (up to some issues concerning clock corrections) that it should, i.e. the satellites transmit at round milliseconds.

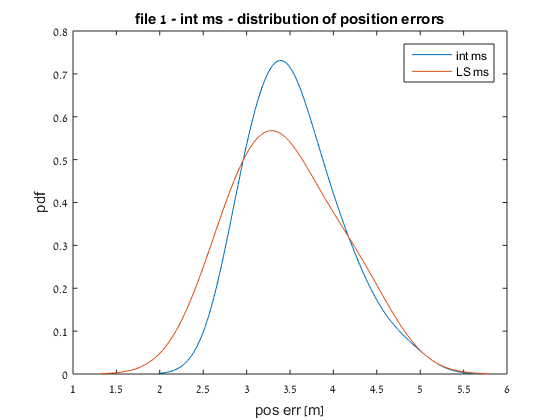
In this experiment we get a state vector

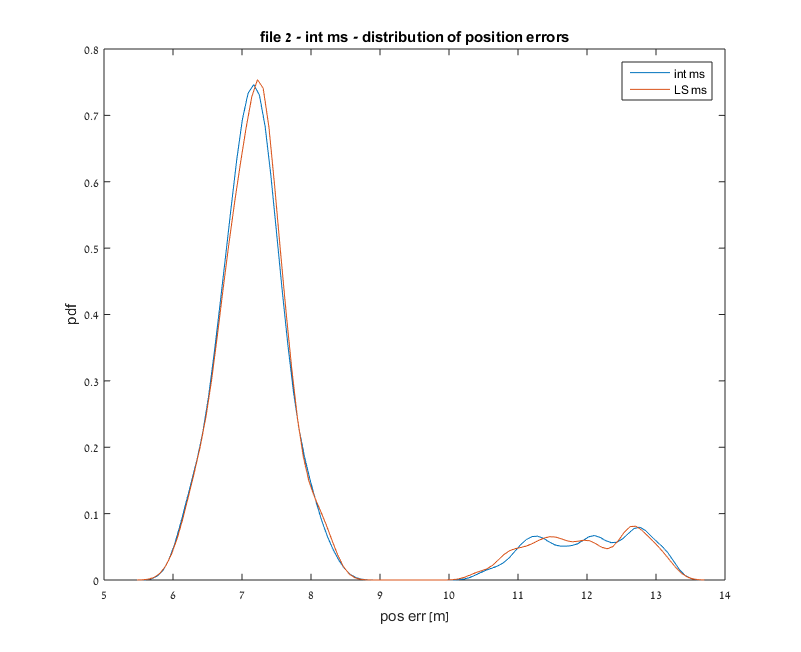
Then we construct a vector of integral milliseconds hypotheses :

Then we run the regular navigation trilateration algorithm:

Then we sort the solutions according to their residuals and choose the least residual one.

Results show that this method doesn’t have a strong clear advantage over the regular least-squares CTN.





# Talk about normalization issue in the observation matrix