Final steps

Assuming one has been successful in generating satisfactory position data with both the tapad and kalman approach there remain two further steps whose main purpose is to ensure that any remaining problems are not overlooked.

1. Velocity repair

This is now something of a misnomer, since the repair function should only be used very rarely. This repair feature is outlined below.

Currently, the more relevant aspect of this processing step is that it filters the position data again with the filters that were used at the start of processing to filter the raw amplitudes.

Wrapper script: do_velocityrepair_base.m

The data generated by this step (stored in files with 'velrep' in the path name) represents the final stage of the basic position calculation.

2. Euclidean distance between solutions

A good quality-control procedure is to store the difference between alternative solutions with the position data. The usual choice is to compare the tapad solution (at the output of the velocity repair procedure) with the kalman solution (an alternative would simply be to compare the input and the output of the velocity repair).

Wrapper script: do_eucdist2pos_base.m

Further notes

(1)

velocity repair (even just filtering) is probably superfluous for kalman

(2)

Sometimes the tapad solution may be basically OK, but the kalman solution is much better just for one individual sensor.

The following wrapper script can be used to replace the data from the 'main' solution with data from an alternative solution:

do_movesensor_base.m