

Localisation example. Encoder/Laser, using EKF.

Documentation

The following Matlab code implements an Encoder/Laser localisation, with an extended Kalman filter. The code is very easy to understand. It is basically an extended Kalman filter using a real data set taken with the ute in the car park.

To start the filter, run the script file *localiser*.

Function Definitions:

localiser:

Function to run the code

FindT:

Function to find the index of the time sensor vector, for the time we choose.

[GPSTIME, LONG, LAT]=ReadGpsData(file):

Function to read the gps data, and transform it to a local navigation frame. This is valid only in a "local" frame, because we're doing a linearization at a reference point.

[Time, STEERING, SPEED1]=ReadUteData(file):

Similar to the *ReadGpsData* function: read the encoder data

[xpred, Ppred]=pred(xest, Pest, dt, u):

Function to make the prediction stage of the EKF. The inputs are the estimated state vector, the estimates covariance matrix, and the encoder parameters (velocity and steering).

[beacon]=getpos:

Function to get the position of the artificial beacons, from a gps data set.

laserview(RR, a, xp, hhh3, hhh4, hhh2, LASERr, LASERo):

Function to plot the laser scan. It plots all the points with a colour and the high intensity points with a different colour.

[LASERr, LASERo, RR, a]=getdata(laser):

Function to get the range and bearing to the beacons. Gets the high intensity point and evaluate the centre of the beacon.

[xest, Pest, innov, S, index]=update_gps(xpred, Ppred, zgps):

Implement the update stage of the EKF with a gps observation. The inputs are the predicted state vector, the predicted covariance matrix, and the gps data (longitude and latitude). The GPS is only used for initialisation purposes.

[xest, Pest, innov, S]=update_laser(xpred, Ppred, zlaser, beacons):

Implement the update stage of the EKF with a laser observation. The inputs are the predicted state vector, the predicted covariance

matrix, the laser frame, and the beacons position (remember that is localisation).

[meanq,q,chib_up,chib_low,timeinn]=inn_analyse(inn,S):

Function to analyse innovation sequences. Is used only to plot the Normalised Innovations and Innovation Confidence Bounds.

Rxx=auto(x):

Computes autocorrelation of input data set. N is the number of data points, x is a column matrix holding the input data set. Uses fft method as advertised in Maybeck p193.

Plots:

Function to do the off-line plots when the filter finish.