**Position Data Toolboxes:**

**MQL Toolbox:** Interpolates position data and allows construction queries to extract timestamps from specific trajectories. Run with Graphical User Interface (GUI)  or individual functions. Type ‘loadScreen’ into command line to run GUI. More information about GUI and tutorial with screenshots: <http://www.cs.bris.ac.uk/Research/MachineLearning/mql/>. Information about individual functions and their parameters in function help sections.

***Interpolation:***Uses two interpolation methods to clean up signal loss, validates interpolation informing the user of its reliability.

[pos\_interp] = interpolation (data, mazeBox, timeout, maxDist, boxI, distI, invalidD, invalidT)

***plotRuns:***Used by MQL to visualise trajectories between two timestamps

plotRuns(startpoints, endpoints, interp)

***run\_query:*** queries position data to return trajectories specified by query line coordinates

[timestamps, valid] = run\_query(interp\_data, query, avoid)

**Bayesian Reconstruction Toolbox:** After training on spiking and position data, allows the reconstruction of positions from spiking data, many functions for visualisation of data and reconstruction. Detailed instructions for all functions in ReadmePDF.

**Position Functions:**

**build\_lin:** Given coordinates of points on a single trajectory through a maze, build\_lin fills in the coordinates which make the lines that connect the supplied coordinates.

idline = build\_lin(corner\_points)

**lin\_maze:** Flattens x,y trajectories to one dimension by fitting them to an ideal trajectory through the maze and returning the indexs of the ideal line that the trajectory coordinates fit best.

[linearisedPos, idealLinPos] = lin\_maze(idline,traj)

**centralarmtimes:** Extracts x, y position and timestamps from Nlx data and finds timestamps corresponding to entry and exit from central arm in both directions.

[ts,xpos,ypos,choicetimesin,choicetimesout,forcedtimesin,forcedtimesout] = centralarmtimes(nvt\_fn,mode, selection)

**checkpos:** No help in function

[xc,yc,ts]=checkpos(p,t\_choice,t\_force,sec)

**remposnoise:** Remove sudden jumps. Function is commented out.

pos=remposnoise(pos)

**smoothpos2:** No help in function

[sfpos,ind1]=smoothpos2(x,y,x0,y0,ts,view)

**smoothpos\_22:** No help in function

[sfpos,ind1]=smoothpos(xn)

**spectral\_map:** No help in function, no comments in code

[smp]=spectral\_map(p,csc,cts,cFs,frng)

**xtract\_pos:** extracts positions and trials from .nvt file. Uses smoothpos and fillmiss

[trial,p] = xtrct\_pos(pat,nvt\_fn,mode,selection,maze,view)

**fillmiss:** Interpolates values in a matrix, called by xtract\_pos

Mf = fillmiss(M)

**smoothpos:** No help in function, called by xtract\_pos

[out,ind1]=smoothpos(pos,p0,view)