AN EASY-GOING INSTRUCTION

Main contact person: Wenbo Wang wenbo.wang@student.tut.fi
Additional contact person related to 'functions' and 'classes' folders: Pedro Silva pedro.silva@tut.fi

Main files related to the mobility models:

- **Human Mobility Models and related files:** each can be run separately
  - **random_walk_mobility_model.m**
  - **random_waypoint_mobility_model.m**
  - **boundless_simulation_area_mobility_model.m**

    (In the above three models, basically any modification of parameters cause no errors.)
  - **random_direction_mobility_model.m**
  - **hybrid_mobility_model.m**

  (In the above two models, **setVariable.m** and **thirdDi.m** are used to generate the variables set and the height dimension variables respectively.)
  - **boundless_simulation_model.m** (In this file, it gives an observation from 3-D view of what the boundless simulation area is like.)

Additional files:

  - apDesign.m
  - fpDesign.m
  - estimation0.m

    (The above three *.m file are function file.)
  - mapping.m (In this file, it calculates the power of each track points heard from different APs.)

- **Analysis files**
  - ap_plot.m
  - fp_plot.m
  - ap_statistics.m
  - fp_statistics.m
  - exampleoffp.m
  - downsampling_plot.m
  - error_probability.m
  - RMSEvsNoise_plot.m
- **Figure_plot**
- **Real_data**
  - reformFP.m
  - reformTRK.m
  - AverageRMSE1.m
  - AverageRMSE2.m
- **Velocity_and_angles**
  - fp_v.m
  - velocity.m

- angle.m

Detailed description: Human Mobility Models and related files

Random Walk Mobility Model

Because the usage of Figure_plot class, make sure that you set the path correctly before run this code. This rule is valid wherever the Figure_plot class is applied.
You are free to change the parameters inside the random walk mobility model. But keep this in mind that the axis scale may be not enough to show your figure. Therefore, when you modify the parameters, the axis scale should be changed accordingly.

Random Waypoint Mobility Model

Same as the random walk mobility model.

Boundless Simulation Area Mobility Model

Note: this is not a flawless version, thus there are some points needed to pay attention.
1. Try to keep the velocity small, and meantime keep the steps relatively large;
2. Try to keep the direction range reasonable, otherwise the track looks funny.

Together with this model, *boundless_simulation_model.m* presents a 3-D doughnut-shaped simulation area, which is the look of boundless area in 3-D view.

In the *boundless_simulation_model.m*, the *x_max* and *y_max* decide the volume of doughnut. You can modify these parameters accordingly.

setVariable

function variable = setVariable(min,max,distribution,n)
min and *max* are the lower/upper bound of the variables set, *distribution* should be string class with only two options – 'uniform' and 'normal', *n* is the amount of the variables. More details you can see inside the code.

thirdDi

function z = thirdDi(amount,timeEachStep,z0,ground,sky)

amount is the amount of the height dimension variables, *timeEachStep* is the resolved time, *z0* is the first variable of the height dimension, *ground* and *sky* is the lower/upper bound of the height dimension variables.

Random Direction Mobility Model

Fs is the sampling frequency, when you increase this value the high accuracy will be achieved. However, the computation becomes complex. Therefore, the suggestion is that to keep this value 10.

We applied ***setVariable*** function here, you can freely change the distribution, from 'uniform' to 'normal' or vice versa.

Note: when you change any parameters, you should pay attention to the parameters inside ***thirdDi***. The amount of the height dimension should be as the same as the other two dimensions.

FYI, the ***random_direction_mobility_model_2Dplot.m*** is only for plot usage, nothing different.

Hybrid Mobility Model

Mostly you can operate as the same as the random direction mobility model.

Suggestion for the further modification:

1. Change the distribution velocity, angles, moving time, pause time and *dist*;
2. Pay attention to the parameter *dist*, modify it so that the user may move around any center point, and the center point could be different from floor to floor, the center point could also be multiple;

apDesign

function AP = apDesign(xmax,ymax,noOfFloor,floorHeight)

You can freely modify any parameters inside this function file.

The structure of data is shown here:

$$\begin{Bmatrix} row1\colon\ index \\ row2\colon\ power \\ row3\colon\ position \end{Bmatrix}$$

fpDeisgn

function [n_ap,fingerPrint,AP] = fpDesign(xmax,ymax,noOfFloor,floorHeight,tol,n_var)
n_var is the variance of noise variables. This parameter is usually read from pre-stored AP data.
The structure of data is shown here:

$$\begin{Bmatrix} row1\colon\ positions \\ row2\colon \begin{bmatrix} sub-row1\colon\ index\ of\ ap \\ sub-row2\colon\ heard\ power \end{bmatrix} \end{Bmatrix}$$

The data is stored in **cell** form.

estimation0

function [B,I,est_pos,trk_pos,error] = estimation0(var,fingerPrint,track,n,algorithm)
var is the variance of noise variables.
algorithm, you can choose 1 – 3 algorithm. (you can add more algorithms to this function file)
Here the error is given in the term of **RMSE**.
About the algorithm 1 – 3 , you can find detailed explanation in my thesis, this is done by checking on the tut dpub system.

mapping

The file is used to form track database. Because in my thesis phase, I used a 3 floor, 50 m*50 m simulation area, the floor loss is only calculated in 3 floors case, you need to adjust this part to adapt your model(s).
*Most of the parameters should be loaded from predefined AP database.

Analysis File

All these files are highly dependent on my thesis. You can read these files while reading my thesis. The details will not be narrated here.

Figure_plot

This is an axe class written by Figueiredo e Silva Pedro, and it gives splendid looking of figures. For further usage of this class, you may contact him.

Velocity and Angle

This part is related to my thesis chapter 7. For more details you can read the code itself, it is understandable.