

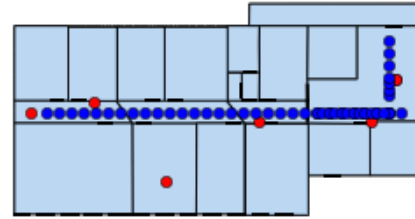
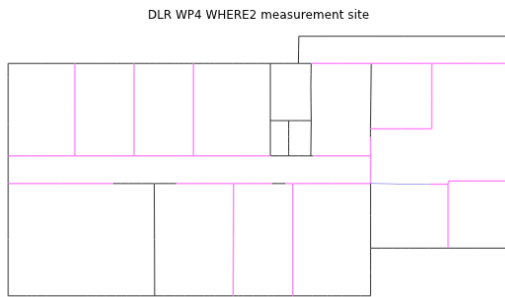
DEV-W2-DLR

Bernard Uguen

I. SIMULATION CREATION

```
<matplotlib.figure.Figure at 0x7f04b0732510>
```

Create the simulation object with defstr layout



```
array([[ 0.   , 19.52 , 30.574, 28.606,  0.85 , 6
        30.574],
       [ 0.   , -0.69 ,  2.8   , -0.74 ,  0.   , 0
        2.8   ],
       [ 0.   ,  1.446,  1.291,  1.467,  1.18 , 1
        1.291]])
```

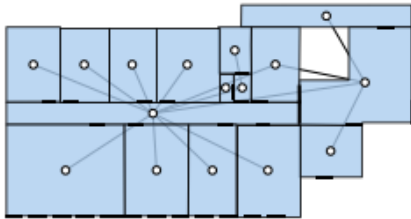
Select Tx and Rx positions

Warning : no furniture file loaded

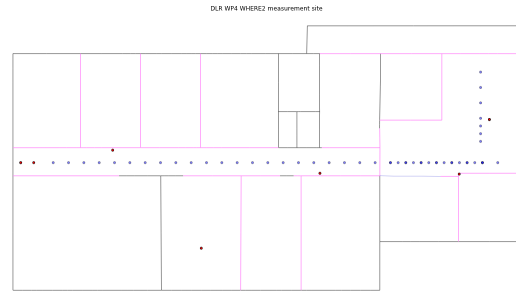
A. Measures from WHERE2 WP4 measurement campaign @ DLR

B. Loading the Layout

$(-5.0, 35.0, -10.0, 10.0)$



```
(<matplotlib.figure.Figure at 0x2a12610>,
 <matplotlib.axes.AxesSubplot at 0x3e74750>)
```



transmitter : [0.85 0. 1.18] is in room 1

mobile node : [29.12 0. 1.275] is in room 7

Next actions :

- + Integrate showr2d in Layout class or Signature (TBDefined)
- + Construct Rays3D from ht and hr Signatures :
- + Integrate reflexion on ceil and Floor -----
- + Calculate geometrical informatio out of rays²⁴⁸
- + Convert in .tud format (Pyray compatible files) from 1 to 8
- + Use multiwall on all those paths

Signatures :

248
from : 1 to 8
3 : 24
4 : 236
5 : 1554

37 [1, 18]
82 [15, 18]
96 [13, 15]
99 [8, 9]
114 [11, 13]
118 [9, 11]

Run raytracing determination between itx and irx

Load Tud object from generated file. A Tud object is gathering all the necessary information for evaluating the field along the rays of the link (itx,irx)

II. HOW TO USE THE GRRAYTUD OBJECT?

A. *Gt* is an object from class *GrRayTud* (Group of rays in TUD format)

1) *Gt.eval*: This method evaluates the field along the rays.

2) *Gt.info(r)*: returns the information associated to a given ray number *r*

3) *Gt.ray(r)*: return the index of the interactions of ray *r* into the *Gt.II* matrix

4) *Gt.typ(r)*: return the type of the interactions of ray *r*

B. *Gt* Attributes

1) *Gt.I*: *Gt* has an attribute *I*, which is a class Interactions.

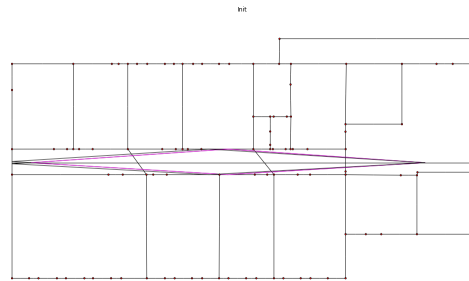
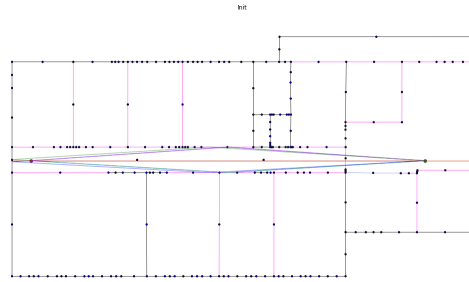
This class Interactions gather all basis interactions : B / T / R / L

All basis interactions B / T / R / L have the same attributes:

- *idx* : the index of the interaction
- *data* : which is a `np.shape(self.data) = len(self.idx), 3`
- `data[:,0]` = theta
- `data[:,1]` = si
- `data[:,2]` = sr or st (named sout in the following)

T and R basis interactions have also an extra attribute:

- *dusl* : a dictionary of used slab (the number is the **position** into the *self.idx*, not directly the index)



2) *Gt.dli*: the *Gt* object has an attribute : **dli** => **dictionary of length of interaction**

The key of this dictionary is the number of interaction. Thus, Lets see what contains *Gt.dli* for 3 interactions. This is a new dictionary which gives:

- 'nbrays' : The number of ray which have 3 interactions (here only 1)
- 'rayidx' : The index of the ray (here only the ray number 0)
- 'rays' : a np array which contains the index of the interaction matrix