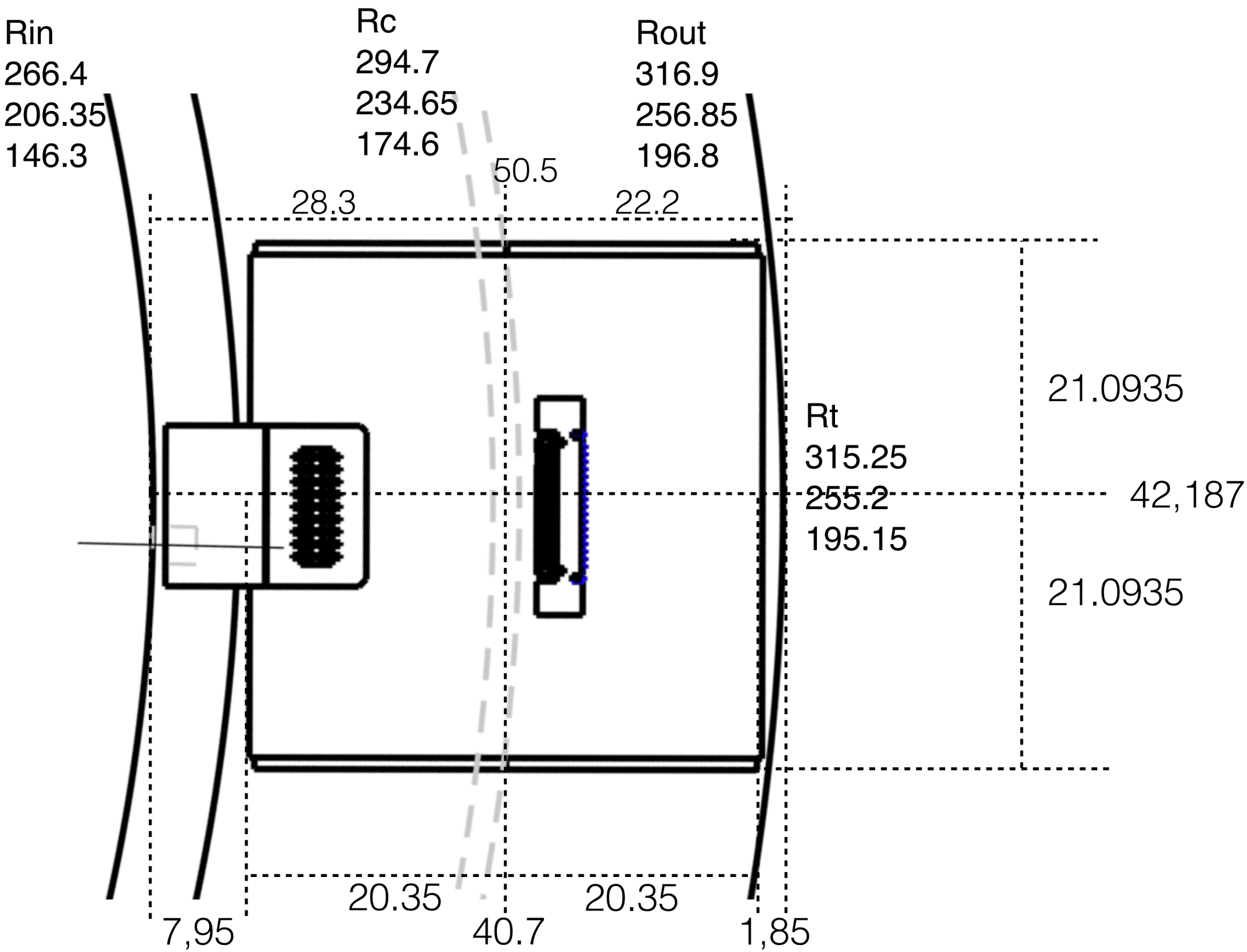
ITK geometry

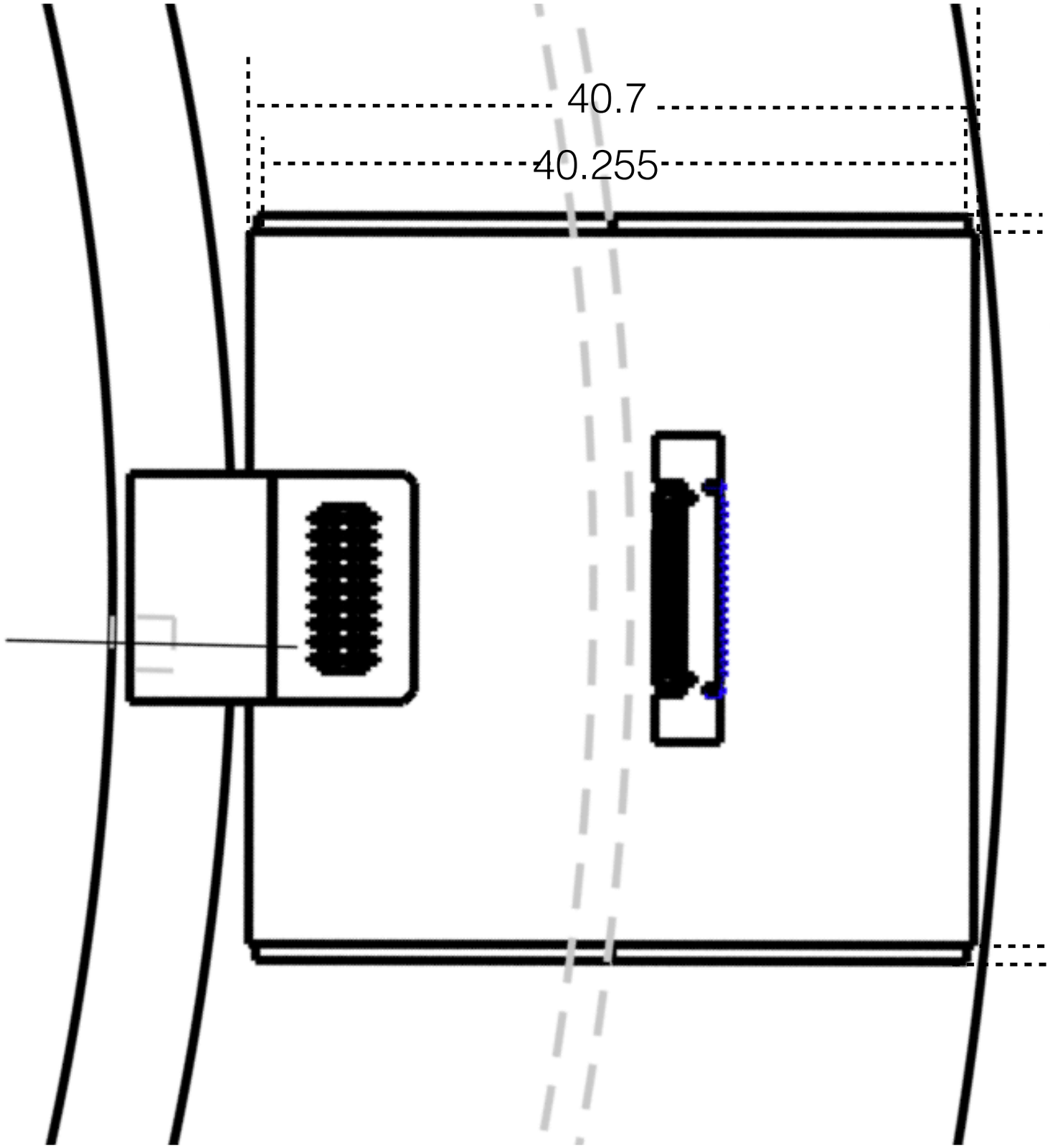
HR and module

15-Giugno-2022

HR and module radial and azimuthal extension

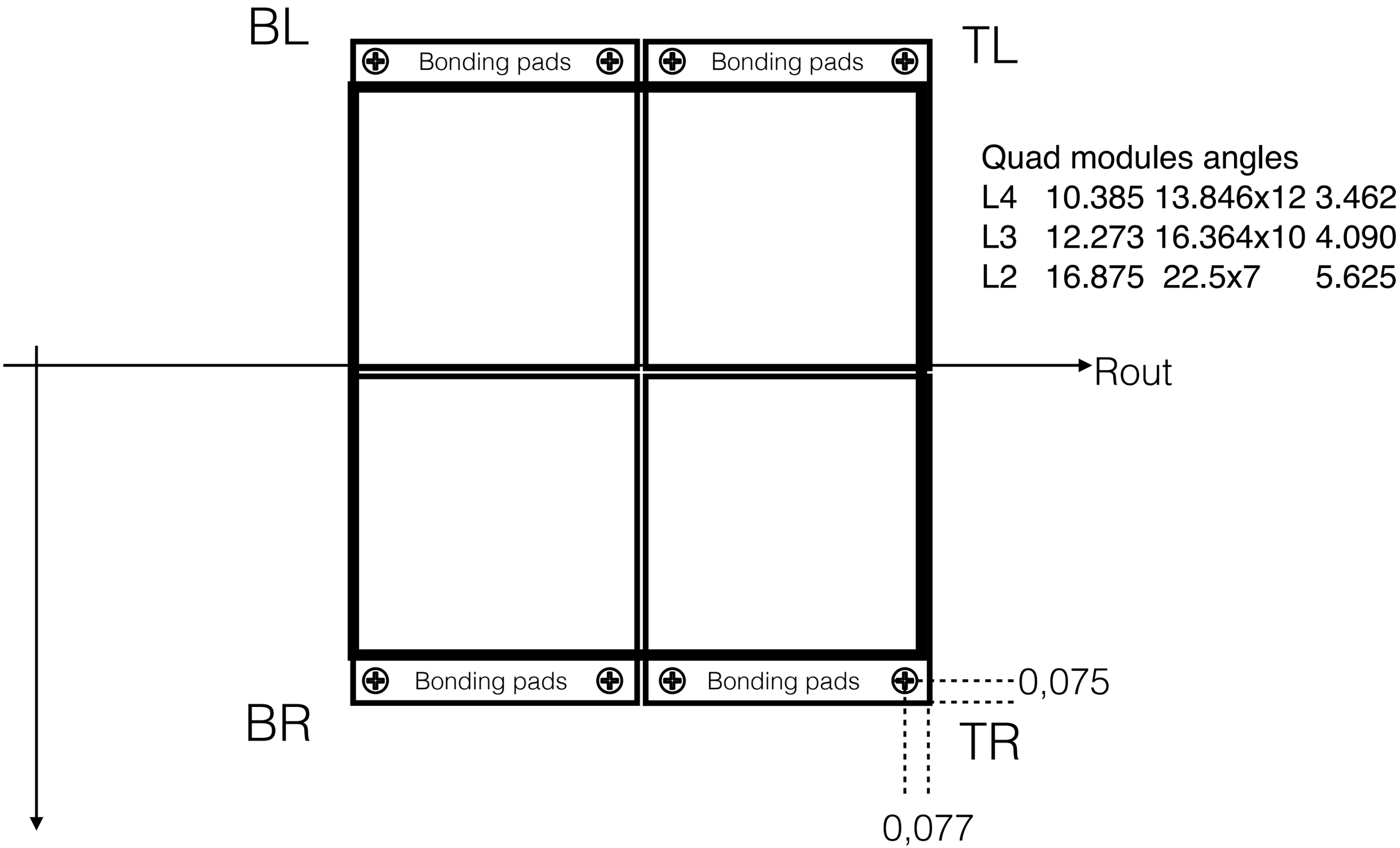


Module top corners



	L4	L3.	L2
Rout	316.9	256.85	196.8
Rc	294.7	234.65	174.6
Sensore			
Rt0	315.05	255.0	194.95
RtL=RtR	315.656	255.748	195.928
Rout-RtL	1.244	1.102	0.872
Chip			
Rt0	314.828	254.777	194.727
RtL=RtR	315.534	255.649	195.866
Rout-RtL	1.366	1.201	0.934

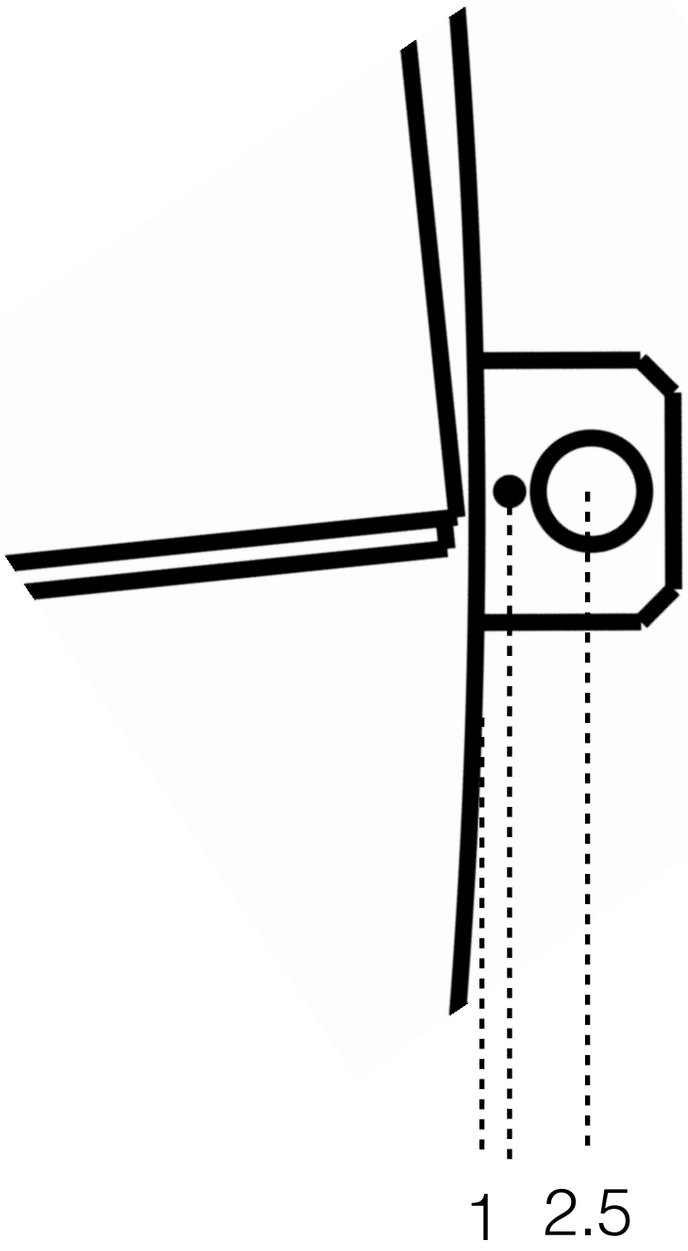
Module fiducial points



Quad modules angles

L4	10.385	13.846x12	3.462
L3	12.273	16.364x10	4.090
L2	16.875	22.5x7	5.625

HR fiducial points



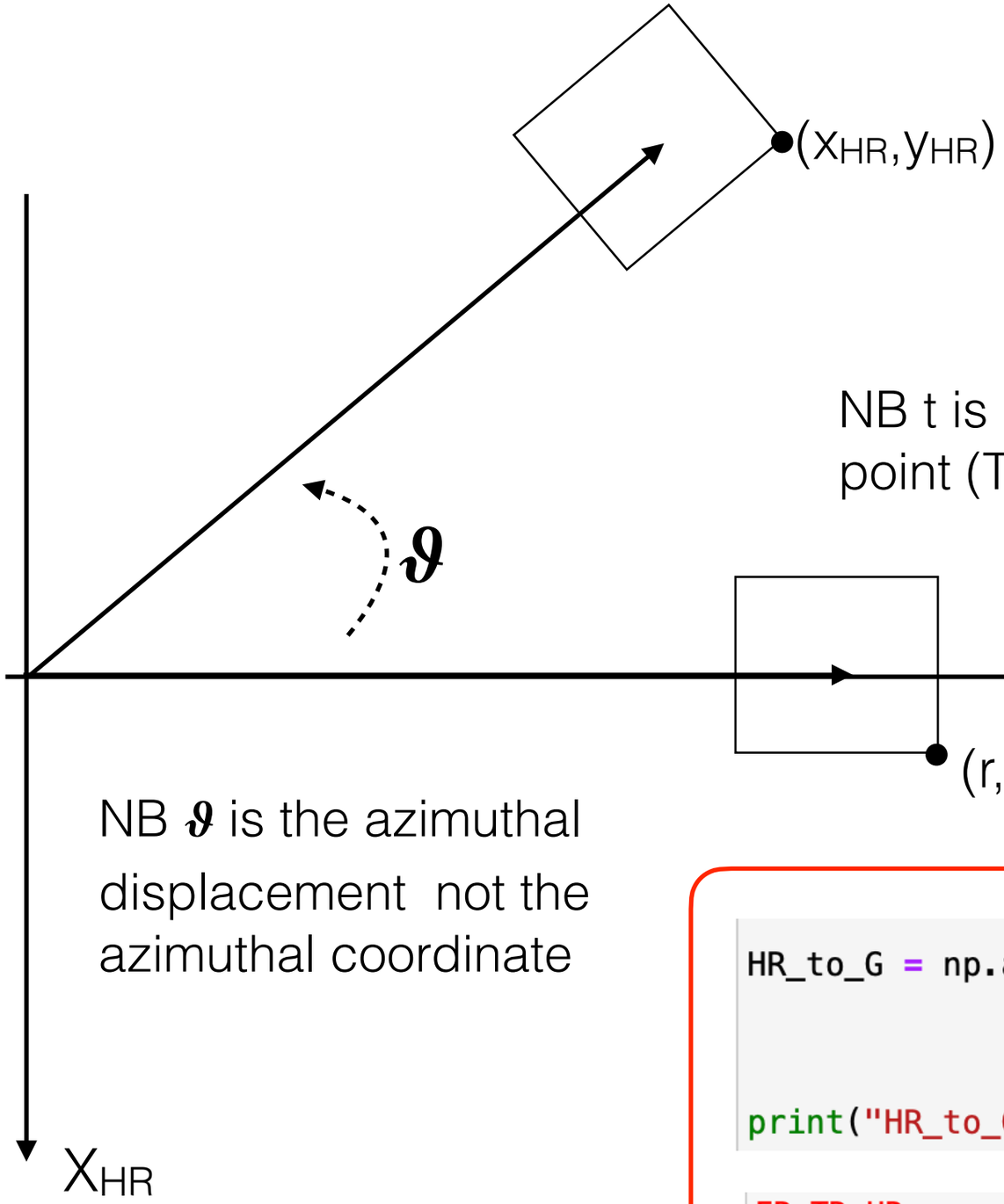
	L4	L3.	L2
Rin	266.4	206.35	146.3
Rout	316.9	256.85	196.8
RFP=Rout+1	317.9	257.85.	197.8
Rlug=Rout+3.5	320,4	260.35	200.3

Lugs angles

L4	2.0	34.5	34.5	38.0	34.5	34.5
	2.0	36.5	71.0	109.0	143.5	178.0
L3.	2.2	44.8	43.0	43.0	44.8	
	2.2	47.0	90.0	133.0	177.8	
L2	2.5	53.7	67.6	53.7		
	2.5	56.2	123.8	177.5		

HR coordinate system

HR coordinate system



$y_{HR} = r\cos(\vartheta) + t\sin(\vartheta)$

$x_{HR} = -r\sin(\vartheta) + t\cos(\vartheta)$

NB t is positive for the point (Top Right TR)

Radial coordinate system

NB ϑ is the azimuthal displacement not the azimuthal coordinate

Easy to think first in the radial coordinate system

```
HR_to_G = np.array([
                      [ math.cos(phi_HR),-math.sin(phi_HR)],
                      [ math.sin(phi_HR), math.cos(phi_HR)]
                    ])
print("HR_to_G=\n", HR_to_G)
```

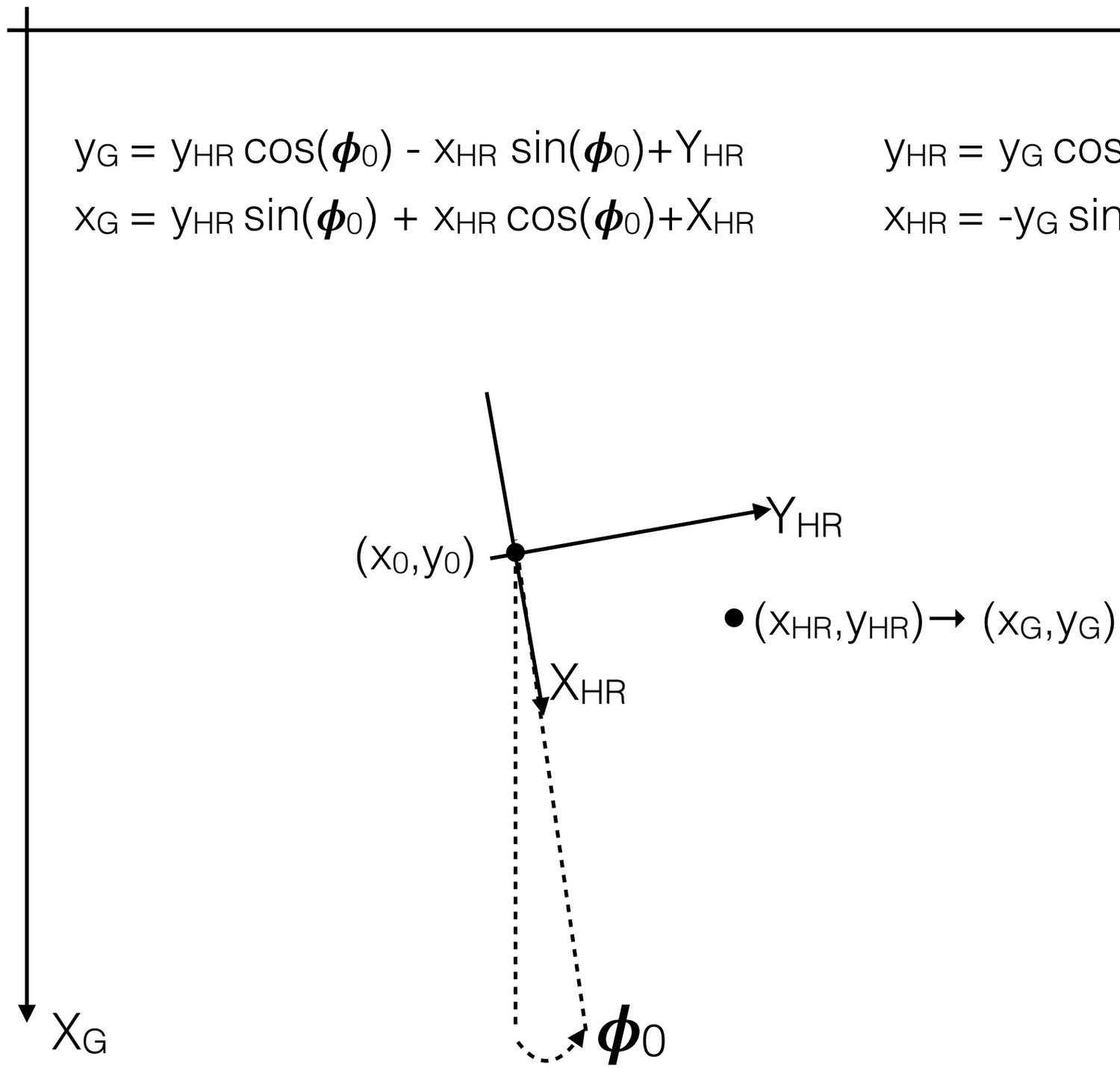
```
FP_TR_HR  = np.dot(FP_TR,Rad_to_HR)
FP_TR_G   = np.dot(FP_TR_HR,HR_to_G)+xy_HR0
print("L2 FP_TR_G=",FP_TR_G)
```

EXAMPLE IN SWAN

Gantry coordinate system

$$y_G = y_{HR}\cos(\phi_0) - x_{HR}\sin(\phi_0) + Y_{HR}$$
$$x_G = y_{HR}\sin(\phi_0) + x_{HR}\cos(\phi_0) + X_{HR}$$

$$y_{HR} = y_G\cos(\phi_0) + x_G\sin(\phi_0) - Y_{HR}$$
$$x_{HR} = -y_G\sin(\phi_0) + x_G\cos(\phi_0) - X_{HR}$$



Python software on swan

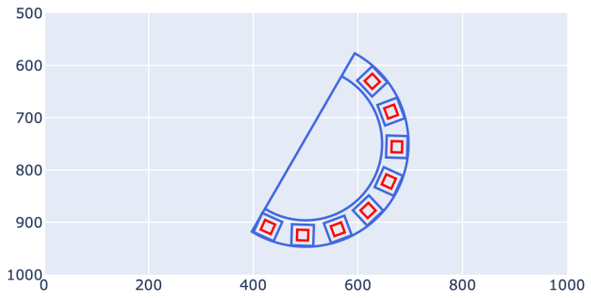
SWAN › CERNBox › SWAN_projects › Geometry

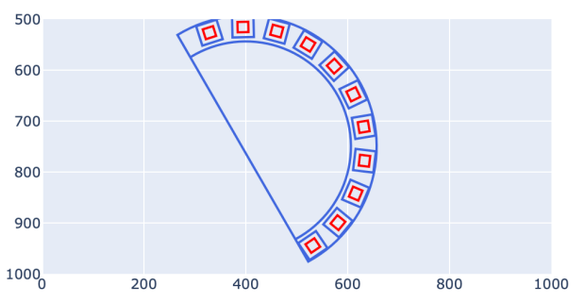
Geometry

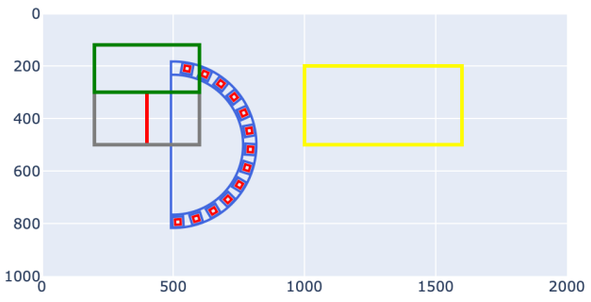
NAME
draw.ipynb
DrawL2.ipynb
DrawL3.ipynb
DrawL4.ipynb
HRgeometry.ipynb
RadialRefSys.ipynb
Square.ipynb

In [1]:
```
import math
import numpy as np
import plotly.graph_objects as go

phi_HR      = math.radians(30)
phi_HR_start=-np.pi/2-math.radians(1.5)+phi_HR
phi_HR_stop = np.pi/2+math.radians(1.5)+phi_HR
xy_HR0 = np.array([750, 500])
HR_y0  = xy_HR0[1]
HR_x0  = xy_HR0[0]

print("---HR phi offset and xy offset on Gantry---")
print(phi_HR , xy_HR0)

HR_to_G = np.array([
                      [ math.cos(phi_HR),-math.sin(phi_HR)],
                      [ math.sin(phi_HR), math.cos(phi_HR)]
                    ])
print("HR_to_G=\n", HR_to_G)
```

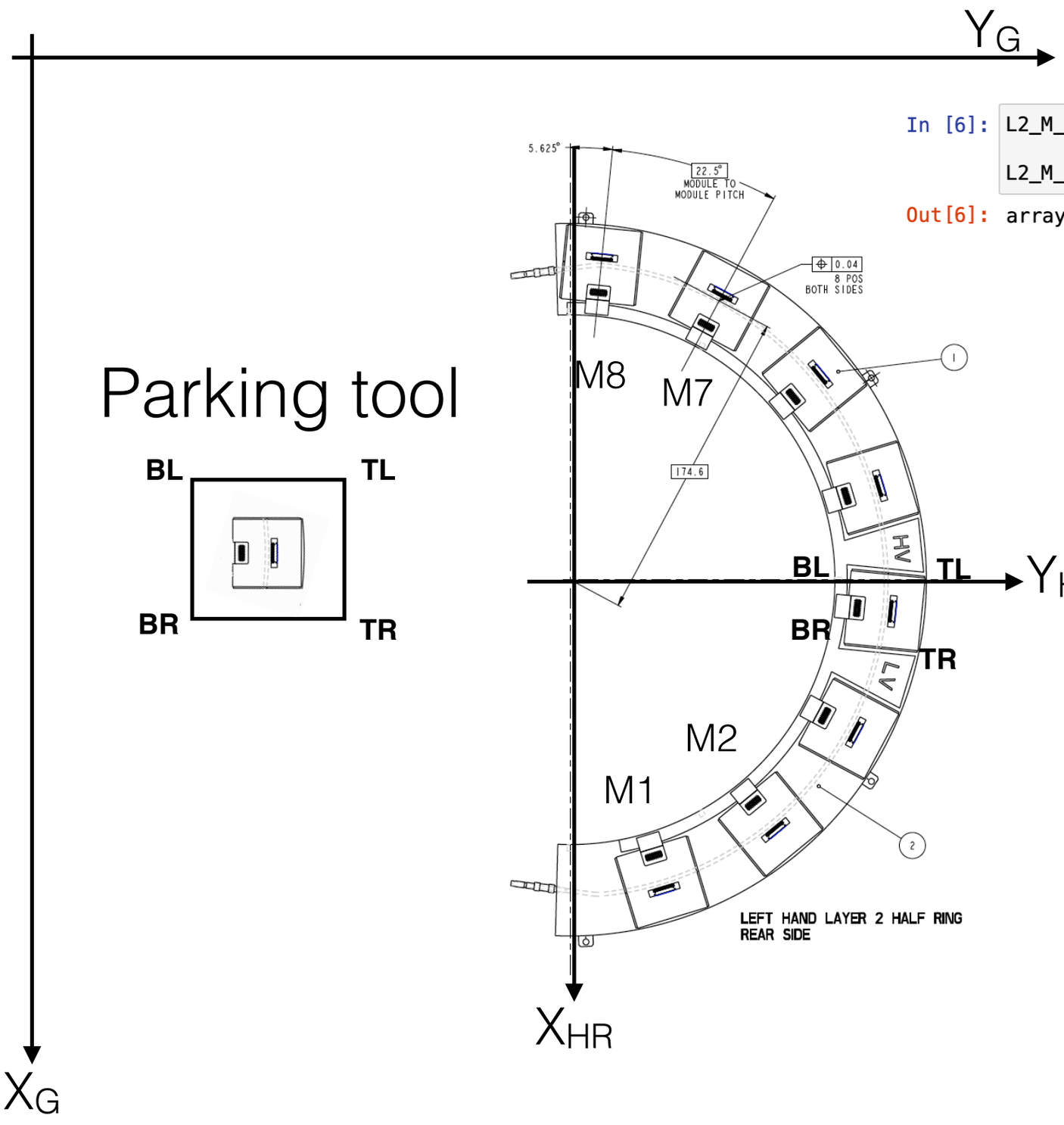
```
---HR phi offset and xy offset on Gantry---
0.5235987755982988 [750 500]
HR_to_G=
[[ 0.8660254 -0.5      ]
 [ 0.5        0.8660254]]
```







HR vs Gantry reference frame




```
In [6]: L2_M_theta=np.array([22.5*0,22.5*1,22.5*2,22.5*3,22.5*4,22.5*5,
                             22.5*6,22.5*7])+16.875-90
        L2_M_theta

Out[6]: array([-73.125, -50.625, -28.125,  -5.625,  16.875,  39.375,  61.875,
               84.375])
```

Right now:

- module angle shifted of 90 degrees in labview code
- Module long side rotated of 90 degrees