Fast transfer Objects

0.) Both sensor Quality → 0 - Neither; 1 - Wii Can see; 2 - Lidar Can See; 3 - both can see; 4 - Lidar sees Multiple objects, Wii sees none; 5 - Lidar Sees Multiple Obj, Wii sees different Obj.

1.) Lidar Angle (-1 if no object, multiple objects)

2.) Lidar distance (" ")

3.) Lidar X (" ")

4.) Lidar Y (" ")

5.) Lidar Qual (" ")

6.) Wii Camera Qual → 0, Can't see; 1, Left camera sees 1; 2, Right Camera sees 1; 3, Left Camera sees 2; 4, Right Camera Sees 2; 5, Left sees 1, Right sees Other; 6. Both see 1; 7. Both see 2

7.) Wii 0 angle (allways reflects Wii angle)

8.) Wii 1 angle (" ")

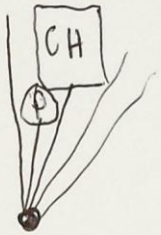
9.) Wii X

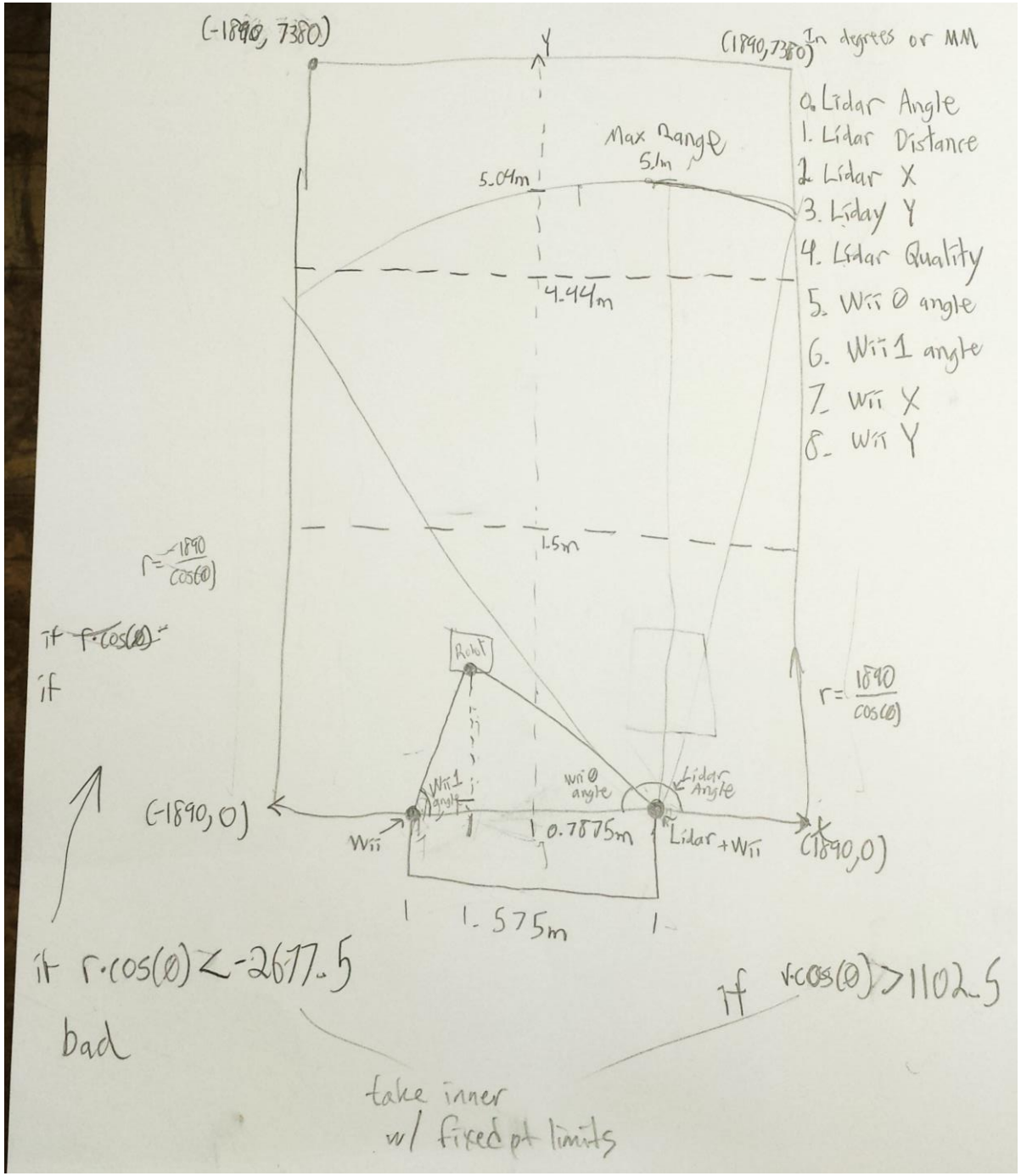
10.) Wii Y

11.) Averaged X (-1 if both don't agree)

12.) Averaged Y (" ")

```
for (i=0; i<180 ; i++)
{
 if (abs(Distance[i] - Distance[i+1])> CUTOFF)
 { start of Object=i;
  for (j=i ; j< 180; j++)
  {
   if (abs(Distance[j] - Distance[j+1]) >cutoff)
   { end of Objet=j;
     Distance to Objet= Distance[(i+j)/2];
     Quality of object = Quality[(i+j)/2];
     Angle of Objet =(i+j)/2;
     X of objet  = PolartoX(Distance[a object, i+j/2). //returns x with offset
     y of object = PolartoY(" ").//returns Y (no offset needed?)
     break;
   }
  }//end loop to find end of object
 }
}//end find start of object
//end function.
```




(-1890, 7380)
(1890, 7380)
In degrees or MM
0. Lidar Angle
1. Lidar Distance
2. Lidar X
3. Lidar Y
4. Lidar Quality
5. Wii 0 angle
6. Wii 1 angle
7. Wii X
8. Wii Y
Max Range 5.1m
5.04m
4.44m
1.5m
r = 1890 / cos(θ)
if r·cos(θ)
if
Robot
r = 1890 / cos(θ)
Wii 1 angle
Wii 0 angle
Lidar Angle
(-1890, 0)
Wii
0.7875m
Lidar + Wii
(1890, 0)
1.575m
if r·cos(θ) < -2677.5
bad
if r·cos(θ) > 1102.5
take inner
w/ fixed pt limits