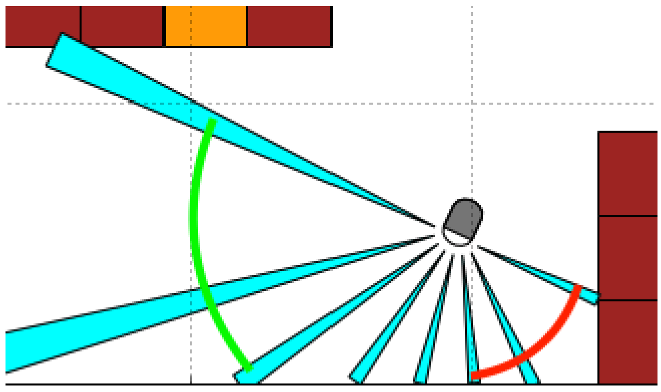
Searching for Free Space

After an obstacle is detected, the robot must turn either left or right in search for free space, and move forward again.

Here is one possible solution:



- `Find the minimum of the left sensors (0, 1, 2)`
- `Find the minimum of the right sensors (5, 6, 7)`
- `If the left minimum is bigger than the right minimum`
  - `Turn left`
- `Else`
  - `Turn right`
- `In any case, keep turning until both front sensors (3, 4) are bigger than the chosen minimum`

In the sample figure, the robot would turn right, since the minimum of the right side sensors (green arc) is bigger than the one of the left side (red arc).

In [3]:

```
import packages.initialization
import pioneer3dx as p3dx
p3dx.init()
```

Minimum and maximum of an array

For finding the minimum and maximum of an array, you can use the Python built-in functions min and max (https://docs.python.org/2/library/functions.html#max).

In [4]:

```
p3dx.distance
```

Out[4]:

```
[0.822314977645874,
 1.107301115989685,
 1.1228526830673218,
 0.9759001135826111,
 0.9870963096618652,
 1.1230719089508057,
 1.5802762508392334,
 3.8271193504333496]
```

In [5]:

```
min(p3dx.distance)
```

Out[5]:

```
0.8090829849243164
```

In [6]:

```
max(p3dx.distance)
```

Out[6]:

```
3.745323657989502
```

You can use only some values of the array, with the Python slice notation for lists (http://stackoverflow.com/questions/509211/explain-pythons-slice-notation):

In [7]:

```
# left sensors (0,1,2)
p3dx.distance[0:3]
```

Out[7]:

```
[0.8257567286491394, 1.1148502826690674, 1.1353843212127686]
```

In [8]:

```
# front sensors (3,4)
p3dx.distance[3:5]
```

Out[8]:

```
[0.9862480759620667, 0.9806881546974182]
```

In [9]:

```
# right sensors (5,6,7)
p3dx.distance[5:]
```

Out[9]:

```
[1.1138113737106323, 1.6000301837921143, 3.793931007385254]
```

Exercise

Implement the above-mentioned algorithm for turning towards free space:

- Find the minimum of the left sensors (0, 1, 2)
- Find the minimum of the right sensors (5, 6, 7)
- If the left minimum is bigger than the right minimum
  - Turn left
- Else
  - Turn right
- In any case, keep turning until both front sensors (3, 4) are bigger than the chosen minimum

```
In [10]:
```

```python
threshold = 1 # in meters
min_left_dist  = p3dx.distance[0:3]
min_right_dist = p3dx.distance[5:]
if min_left_dist > min_right_dist:
    wl = -1
    wr = 1
else:
    wl = 1
    wr = -1
while p3dx.distance[3] < threshold or p3dx.distance[4] < threshold:
    p3dx.move(wl, wr)
p3dx.stop()
```

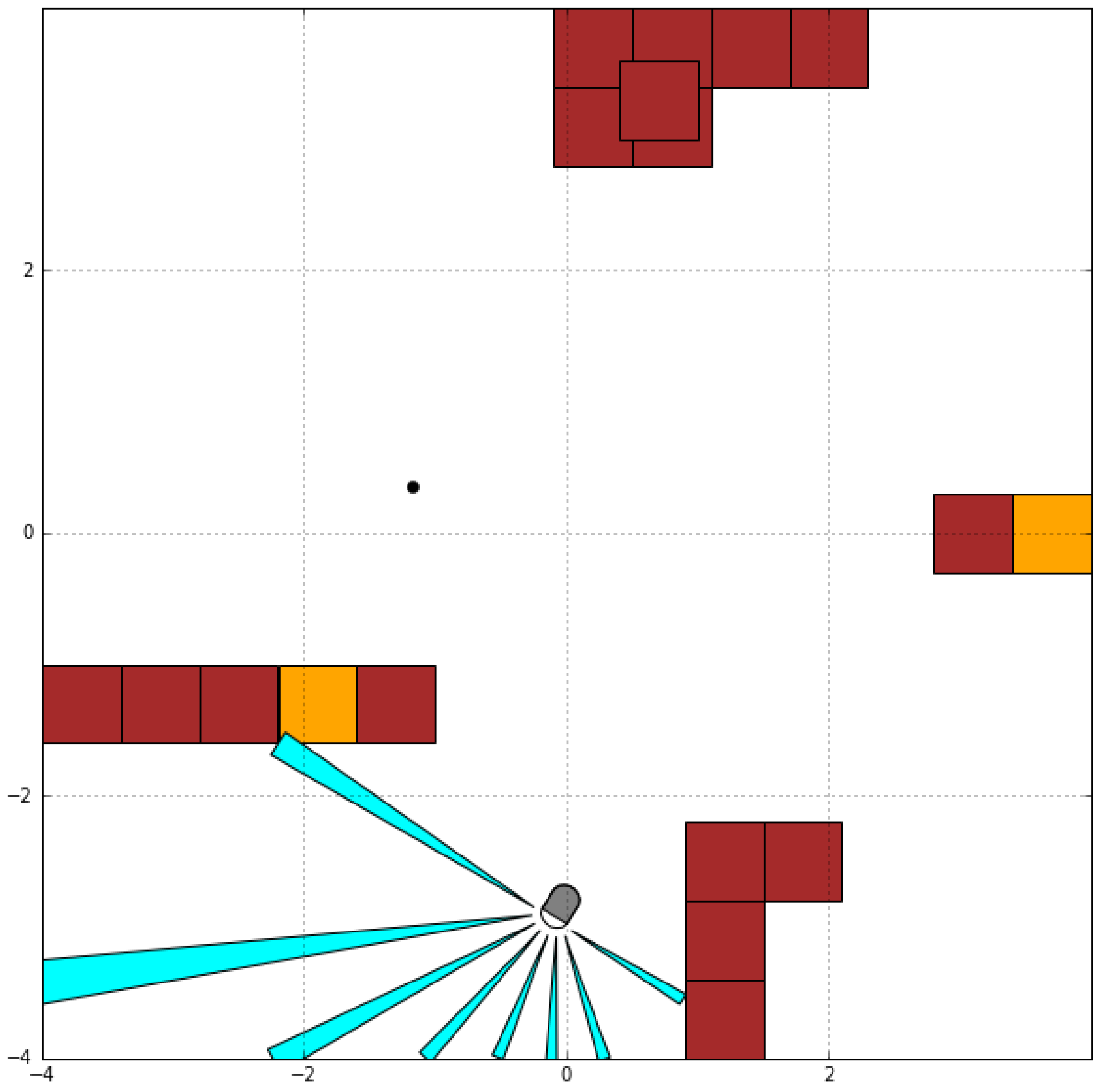
You can plot a diagram with the resulting position of the robot in the environment, and the measurements of the ultrasonic sensors.

```
In [11]:
```

```python
%matplotlib inline
import ultrasonic
```

In [12]:

```
ultrasonic.plot()
```



Let's put together the last two exercises in a simple application: wandering (Wandering.ipynb).