Lab 6: Localization in a Known Map

16-311: Introduction to Robotics

Gillian Rosen: gtr@andrew.cmu.edu
Rauhit Ashar: rashar@andrew.cmu.edu

Models & Software We used the Discrete Bayes Filter outlined in lecture. The observation and transition models that we wrote are described below. Initially our probability distribution is uniform for all occupiable spaces.

Observation Model To build our observation model, we use the following algorithm:

1. For each tennis ball T with position (tx, ty), range r, and bearing b
2. For every angle $\psi \in [0, 2 * \pi]$ representing all possible orientations of the tennis ball
3. Compute a candidate point $p = (cx, cy)$ which has Euclidean distance r away from T
4. If, when you trace a ray from T to p, you do not encounter an obstacle, increase the probability at the cell (cx, cy, ψ) in the probability distribution.

We compute all of our candidate points

$$\forall \psi \in [0, 2\pi],\ p = \begin{bmatrix} cx \\ cy \end{bmatrix} = \begin{bmatrix} tx + \cos(\psi) * r \\ ty + \sin(\psi) * r \end{bmatrix} \tag{1}$$

We trace a ray from the robot to the ball using the definition of visibility given in lecture.

$$(x, y) | \lambda x + (1 - \lambda) y \in FS \tag{2}$$

If there are no obstacles between the tennis ball and the robot in our map, we declare that we can see the tennis ball.

Next, compute our orientation:

$$\theta = \pi - atan2(cy - ty, cx - tx) - b \tag{3}$$

Finally, smooth the resulting probabilities using a 3-dimensional Gaussian blur.

Transition Model To update our position, we use the following algorithm:

1. Given $(\delta x, \delta y, \delta\theta)$ which is the displacement of the robot in world frame coordinates
2. Iterate over all possible (θ, x, y) and shift each probability over by $(\delta x, \delta y, \delta\theta)$

Finally as in our observation model, we smooth the resulting probabilities using a 3-dimensional Gaussian blur.

Discussion of Process For our transition model, our approach was straightforward. However we initially thought that the change in position was given in robot frame coordinates. This meant that we were doing a lot of unnecessary and incorrect translation. Once we eliminated the rotation, the model updated correctly.

For our observation model, we first attempted to compute probabilities for every possible robot position. However, this was too computationally intense and we instead opted to use the tennis balls to isolate the positions that the robot could be in.

A possible bug in our code is that we did not account for noise in our transition model. That is, we were discretely shifting all of our probabilities. This is a problem because our $(\delta x, \delta y)$ could have noise. Furthermore, we weighted our transition model lower than our observation model.

Discussion of Results Overall, our simulation is successfull. The transition model correctly shifts the proposed robot model with the same direction and magnitude as the actual robot's movements. Furthermore, the observation model identifies the possible positions without excluding legitimate possibilities or introducing too much uncertainty. While there is some error in the final predicted position of the robot, this error is only about 10 cm which is an acceptable margin.

A useful technique that we could implement in the future is to incorporate lack of information as a meaningful measurement. Currently, the only information source for our observation model is the presence of one or more tennis balls and their positions relative to the robot. If we made decisions based on observing no tennis balls, then we could increase the speed and accuracy of the observation model.

Pictures below!

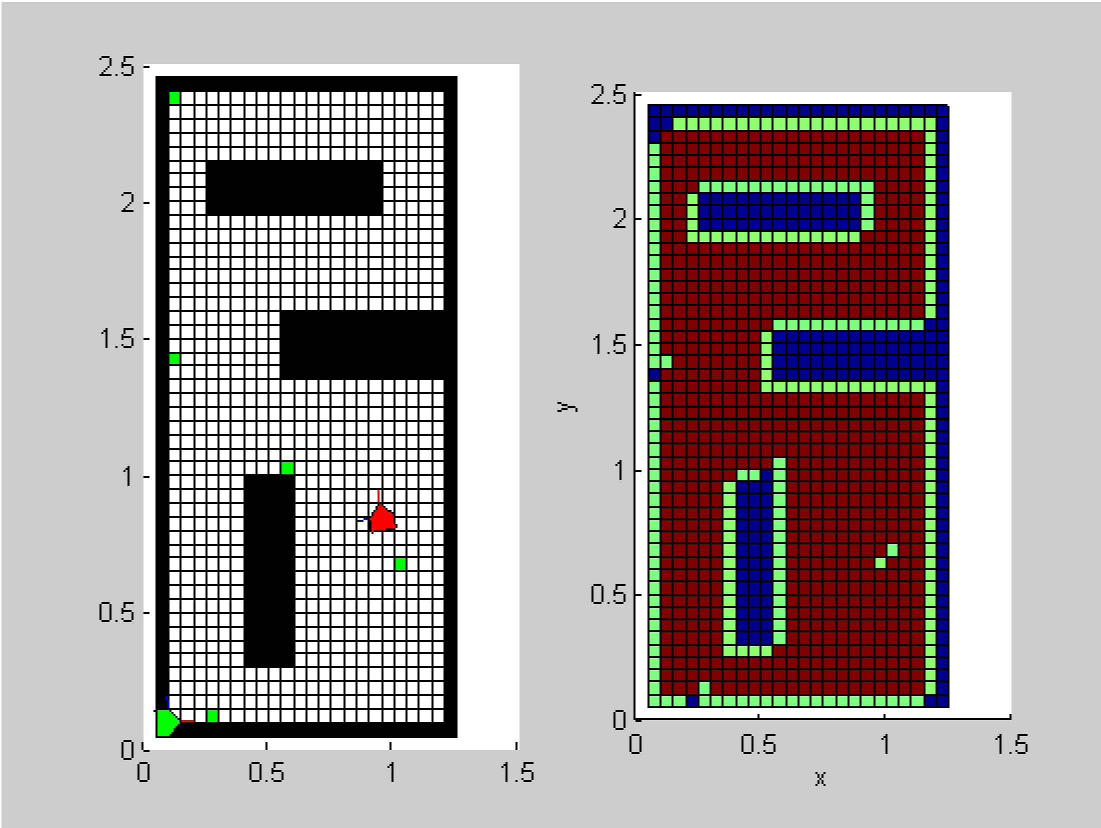
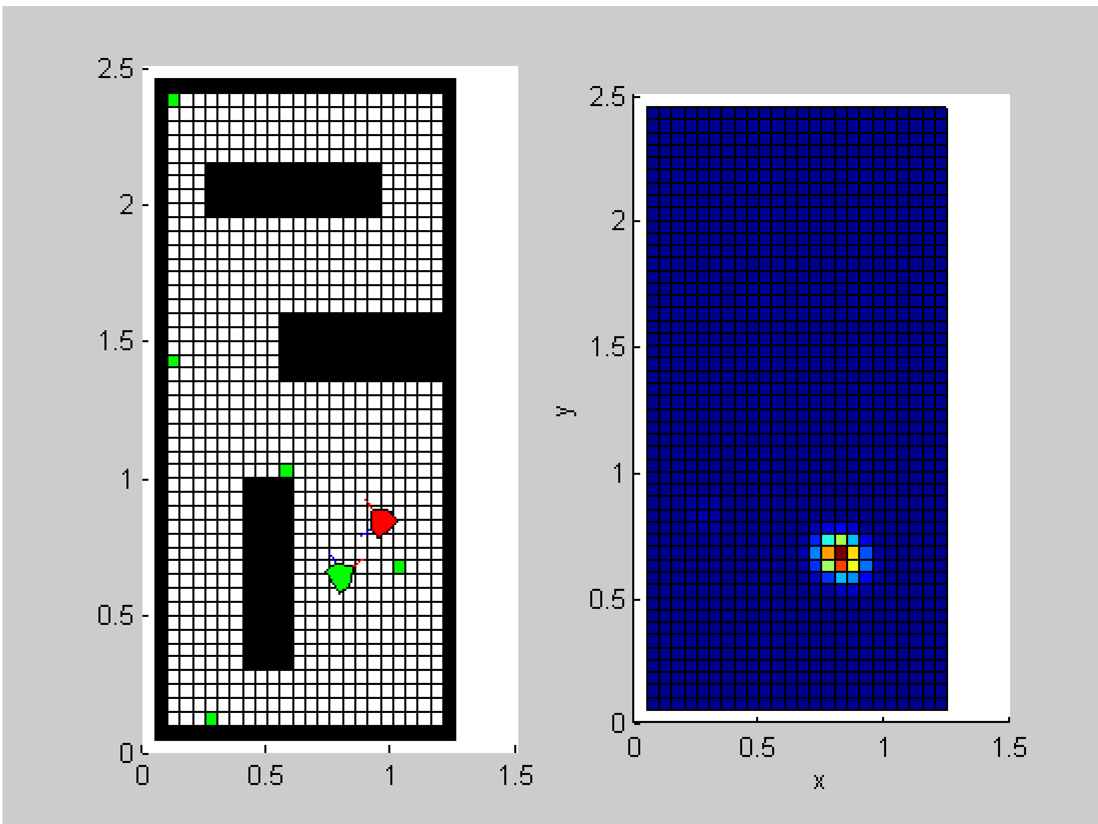


Figure 1: The robot begins in the standard start position.



2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
y
x

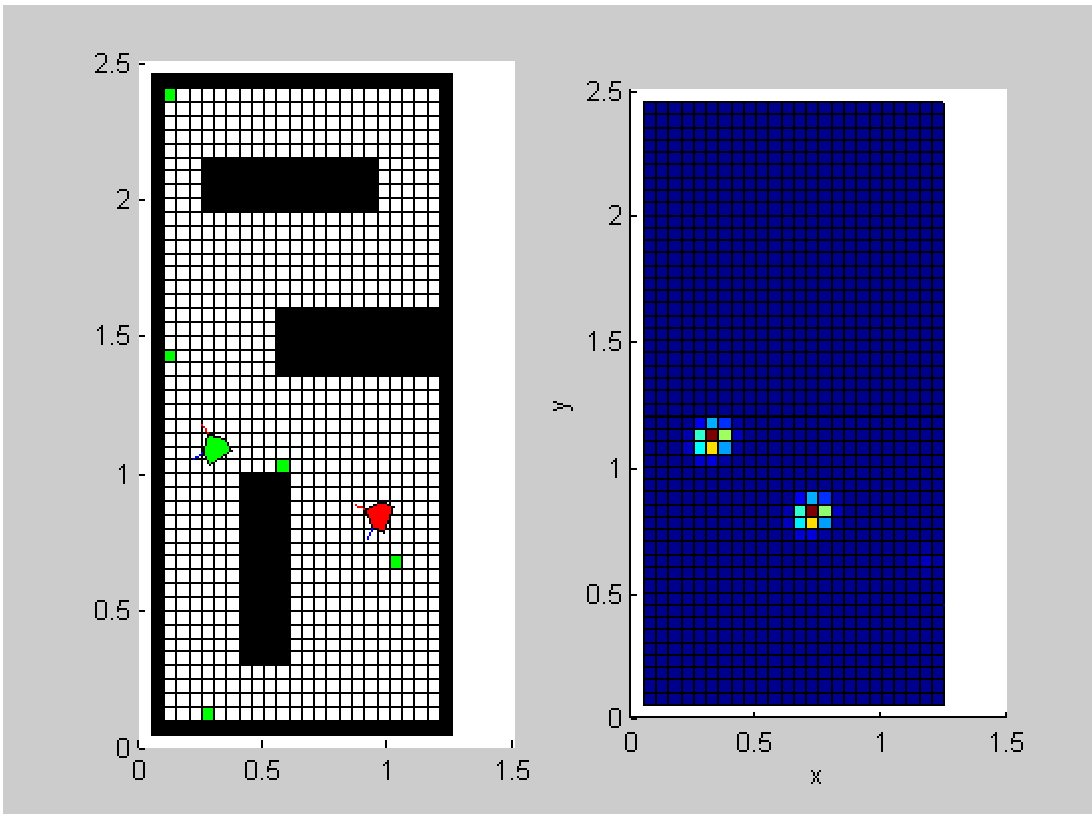
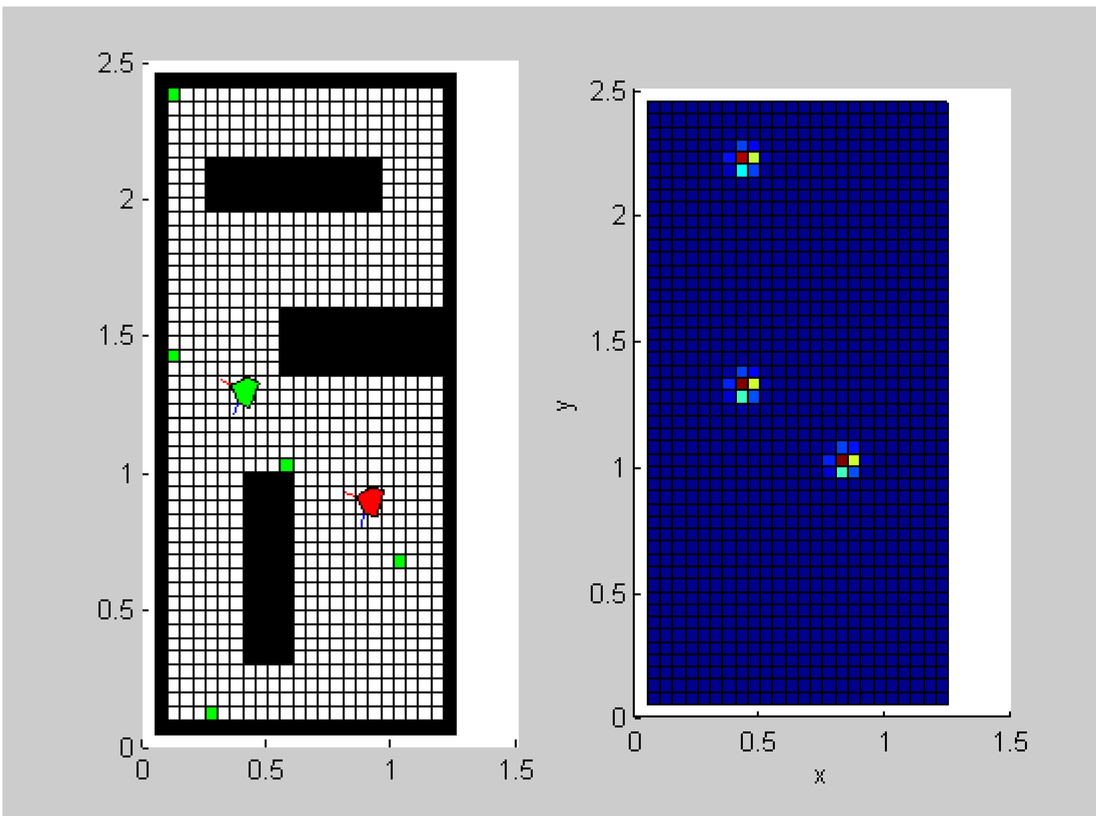


Figure 2: When it rotates, it sees a tennis ball in front of it. Given the bearing and range to that ball, it identifies the positions that it is most probably facing.



2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
y
x

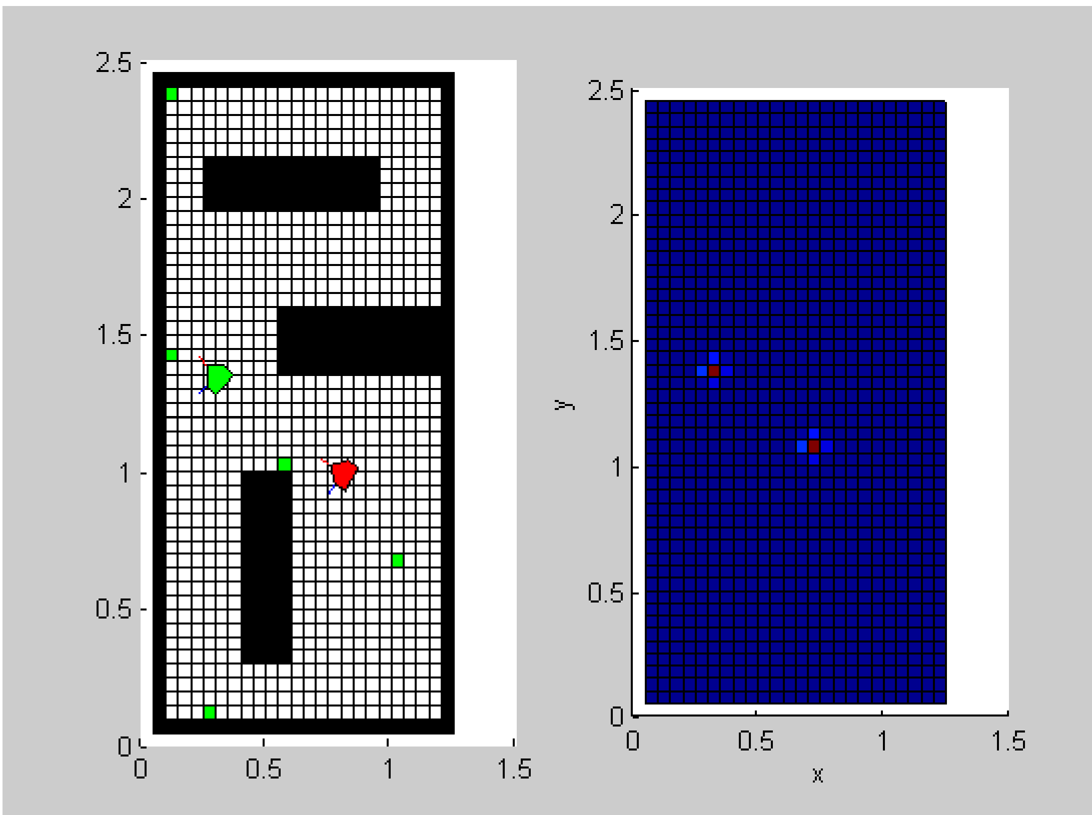


Figure 3: As it approaches the ball it hones in on the two most probable positions.

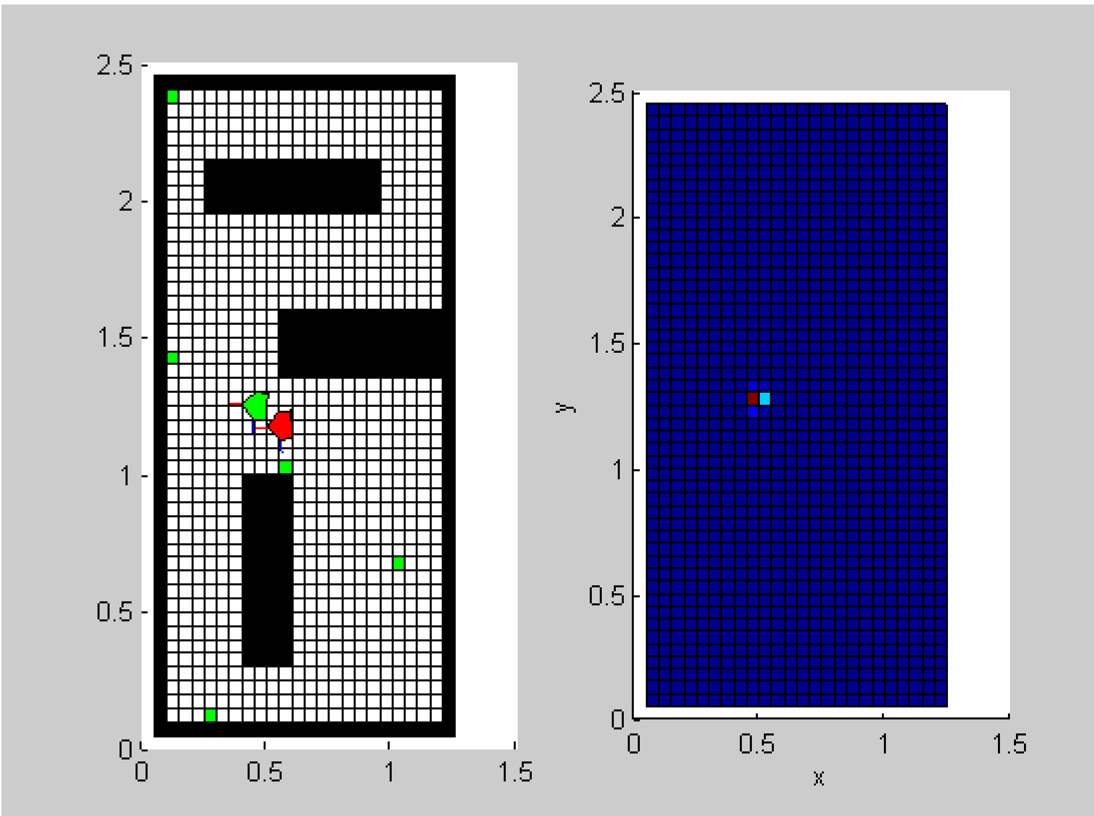


Figure 4: After observing the central tennis ball at its left the robot chooses the middle position as the most probable position.

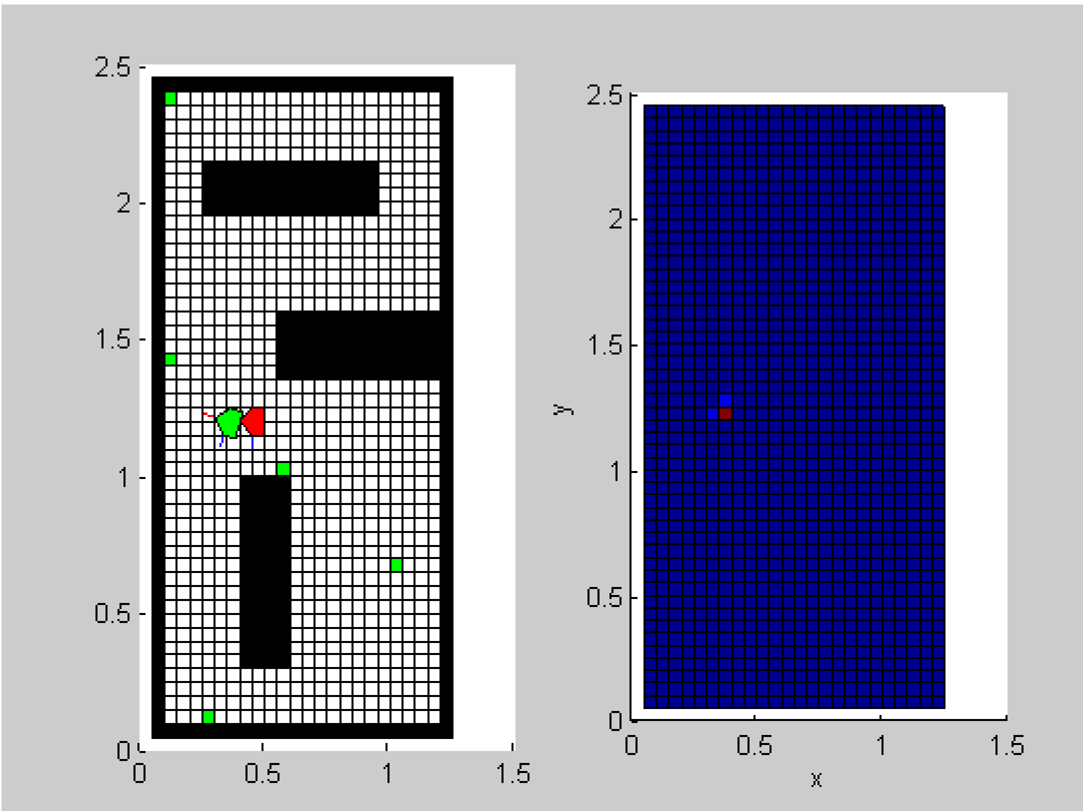
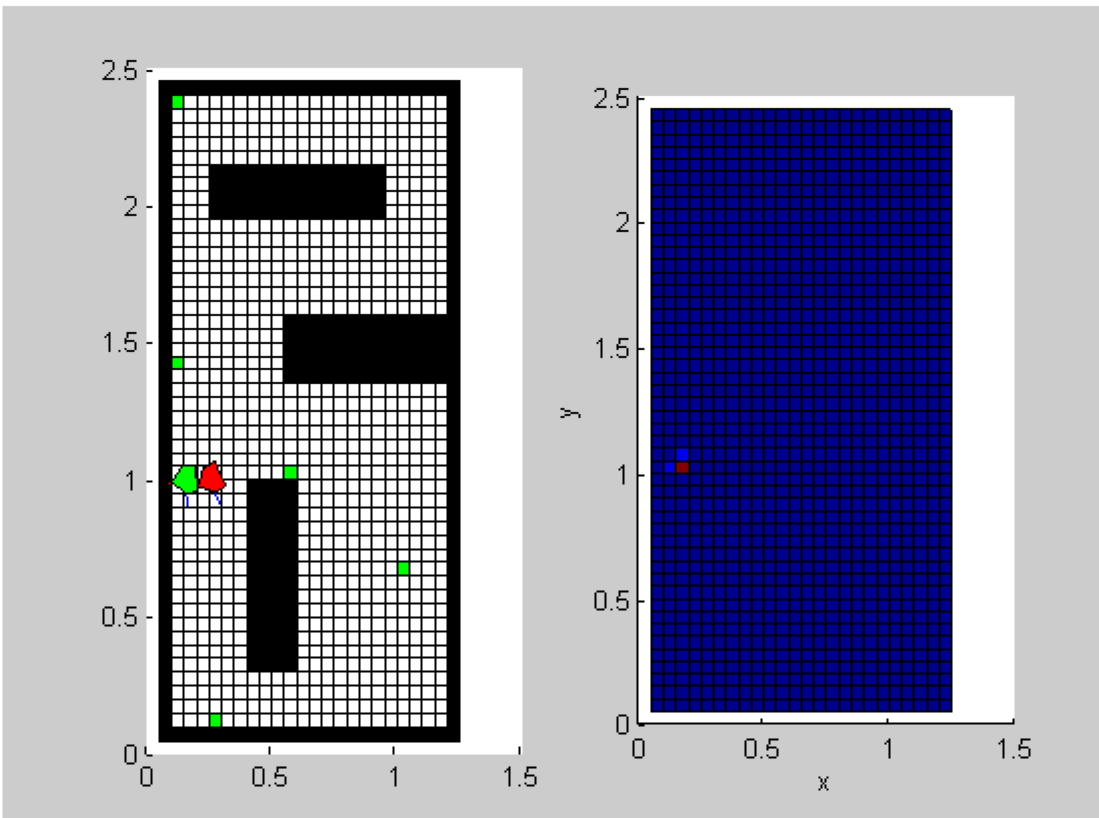


Figure 5: From this point the robot moves toward the goal. Since it no longer observes any tennis balls it maintains a small constant amount of error.



2.5
2
1.5
1
0.5
0
0
0.5
1
1.5
2.5
2
1.5
1
0.5
0
y
0
0.5
1
1.5
x