# Tutorial 5 Strategy for Project 3

Two key issues need to be addressed:

1. How are rewards modelled?
2. Transitions between states and assignment of probabilities to these transitions

## Modelling rewards

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| O1 |  |  | D | O2 |
|  |  |  |  |  |
|  |  | HH |  |  |
|  | S |  |  |  |
| O3 |  |  |  | O4 |

Hazard area H with negative reward

5

4

3

2

1



5

4

3

2

1



An example starting position for the robot

1 2 3 4 5

**Figure 1**

Rewards are easily mapped as only two exit reward positions exist at destination D (4,5) (indexing first by column and then by row) and at H (3,3).

Modelling transitions

First, we will define a blocked position as one of:

1. An obstacle.
2. A position that is reached by a move opposite (at 180 degrees) to the intended movement. For example, if the intended movement is “left” from P, then Q is a blocked position where Q is right of P.

TransProb(s,a)

{

s’ = T(s,a) // T is a function that returns the state s’ when action a is taken from s.

// For example, if the robot is currently in (2,2) and intends to move right

// then s’ is (3,2) with indexing by column and then row

if s’ is the intended state and s’ is not a blocked state

{

p = 0.9 // agent moves to the intended state 0.9 of the time.

}

else if s’ is a blocked position

{

p = 0.0 // agent remains at its current position without moving.

}

else if s’ is not a blocked position and s’ is not the intended state

{

p = (1-0.9)/(n\_dest(s)-1)

// ndest\_(s) is a function that returns the number of (non-blocked)

// destinations from state s

}

}

TransProb=p

return

Once the rewards and probabilities are available from any given state with every possible action then we can apply the Value Iteration function discussed in the lectures

ValueIteration(

{

max\_iterations=500 // set value as needed, may need to decrease to 100

U0(s)=0 for all states s

set rewards as needed

k=0

repeat

{

k=k+1

for state s

{

+

// r(s)=-0.5 or r=-0.1, the live-in reward as the case may be

} until for all states s