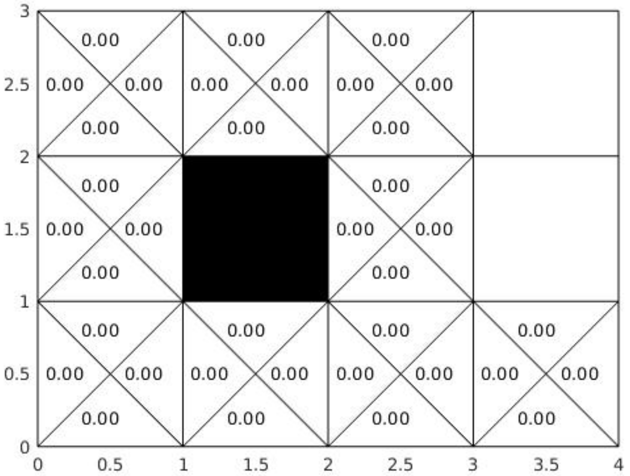
Interactive Session on Reinforcement Learning

Intelligent Control



Problem 1: Q-learning

In this interactive session you are going to use Q-learning to determine a policy for navigating the grid above. This grid is similar to the one we used in class. The square on the upper right and middle right are states for which the game ends if you are in one of those states. To help you I have attached the following files:

- "gridworld.m": This function plots the gridworld map with Q-values for each state-action pair. It takes a matrix of Q-values and the current state as an input.
- "reward.m": This function returns the reward associated with landing in a state.
- "transition.m": This function returns the final state given the initial state and action taken.

Each state is coded by a row vector given the horizontal and vertical indices for the state. The state on the bottom left is [1 1], and the state on the top right is [4 3]. The four actions that you can take are 'east', 'west', 'north', and 'south'.

(a) Program episodes. Write a function that completes one episode. The function should either have a policy hardcoded or made so that the policy can be an input to the function. Have the episode start at the lower left square and move from square to square based on the policy until it reaches one of the terminal sqaures.

(b) Value iteration. Run a bunch of episodes using a policy that randomly selects one of the four actions for each state and after each episode update the Q-values for each state/action pair after each episode.

(c) Epsilon greeder Q-learning. Have your RL agent forget what it learned in part (b) and start over using Q-learning with $\epsilon = 0.2$. This means use the currect policy to decide on a policy 80% of the time and take a random action 20% of the time. Repeat this expercise for various values of the learning rate α.