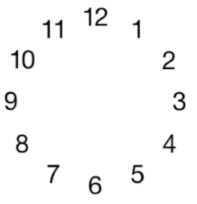
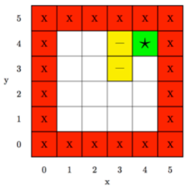
EE183DA Lab 3 – Markov Decision Processes

Problem Description





MDP

$$V^{\pi}(s) = E\Big[\sum_{i=1}^{T} \gamma^{i-1} r_i\Big] \forall s \in \mathbb{S} \tag{1}$$

$$V^{*}(S) = \max_{a} \Big[R(s,a) + \gamma \sum_{s' \in \mathbb{S}} p(s'|s,a) V^{*}(s')\Big] \tag{2}$$

$$V^{*}(s) = \max_{\pi} V^{\pi}(s) \ \forall s \in \mathbb{S} \tag{3}$$

$$\pi^{*} = \underset{\pi}{argmax}\ V^{\pi}(s) \ \forall s \in \mathbb{S} \tag{4}$$

Bellman Equation: Find Optimal Value. Eq. (2)

Value Function: Expected total reward. Eq. (1)

Optimal Policy: Policy for optimal value function Eq. (4)

Optimal Value Function: Eq. (3)

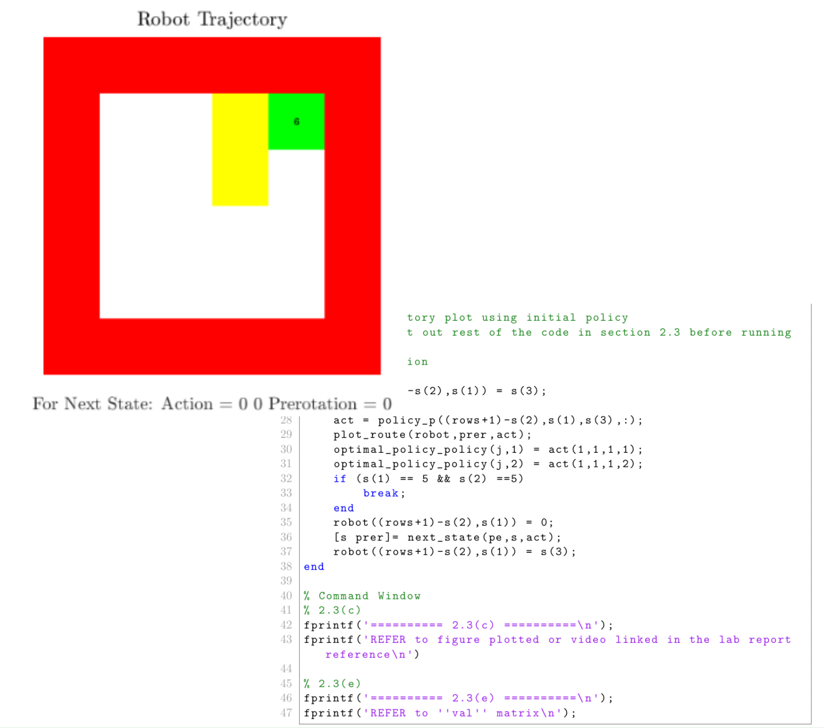
Value Iteration

```
Assign arbitrary values to V(s)
Rerun
    For all states in the state space
        For all actions in the action space
            Q(s,a), where Q(s,a) = $E[r|s,a] + \gamma \sum_{s' \in S} P(s'|s,a)V(s')$
        V(s), where V(s) is max$_a$Q(s,a)
Continue until V(s) converges
```

Policy Iteration

```
Start with an arbitrary policy $\pi_0$
Rerun
    $\pi$ corresponds to $\pi_0$
    Compute values using $\pi$ by solving the following equation:
        $V^{\pi}(S) = E[r|s,\pi(s)] + \gamma \sum_{s' \in S} P(s'|s,\pi(s))V^{\pi}(s')$
    At each state, improve policy
        $\pi'(s) = argmax_a(E[r|s,a] + \gamma \sum_{s' \in S} P(s'|s,a)V^{\pi}(s'))$
Continue until $\pi = \pi'$
```

Matlab Implementation



For Next State: Action = 0 0 Prerotation = 0

```
tory plot using initial policy
t out rest of the code in section 2.3 before running

ion

-s(2),s(1)) = s(3);
    act = policy_p((rows+1)-s(2),s(1),s(3),:);
    plot_route(robot,prer,act);
    optimal_policy_policy(j,1) = act(1,1,1,1);
    optimal_policy_policy(j,2) = act(1,1,1,2);
    if (s(1) == 5 && s(2) ==5)
        break;
    end
    robot((rows+1)-s(2),s(1)) = 0;
    [s prer]= next_state(pe,s,act);
    robot((rows+1)-s(2),s(1)) = s(3);
end

% Command Window
% 2.3(c)
fprintf('========== 2.3(c) ==========\n');
fprintf('REFER to figure plotted or video linked in the lab report
  reference\n')

% 2.3(e)
fprintf('========== 2.3(e) ==========\n');
fprintf('REFER to ''val'' matrix\n');
```