Lab2: Simulation With a Pre-Trained Agent in MATLAB

Abhishek M J - CS21B2018

31-01-2024

Load the environment and agent

```
rng(123)              % Set random seed for reproducibility
load robotmodel.mat   % Load the robot model
agent                 % Display the agent
env                   % Display the environment
```

```
>> rng(123)
>> load robotmodel.mat
>> agent

agent =

  rlDDPGAgent with properties:

       ExperienceBuffer: [1×1 rl.replay.rlReplayMemory]
           AgentOptions: [1×1 rl.option.rlDDPGAgentOptions]
    UseExplorationPolicy: 0
         ObservationInfo: [1×1 rl.util.rlNumericSpec]
              ActionInfo: [1×1 rl.util.rlNumericSpec]
              SampleTime: 0.2500

>> env

env =

SimulinkEnvWithAgent with properties:

           Model : whrobot
      AgentBlock : whrobot/controller
        ResetFcn : []
  UseFastRestart : on
```

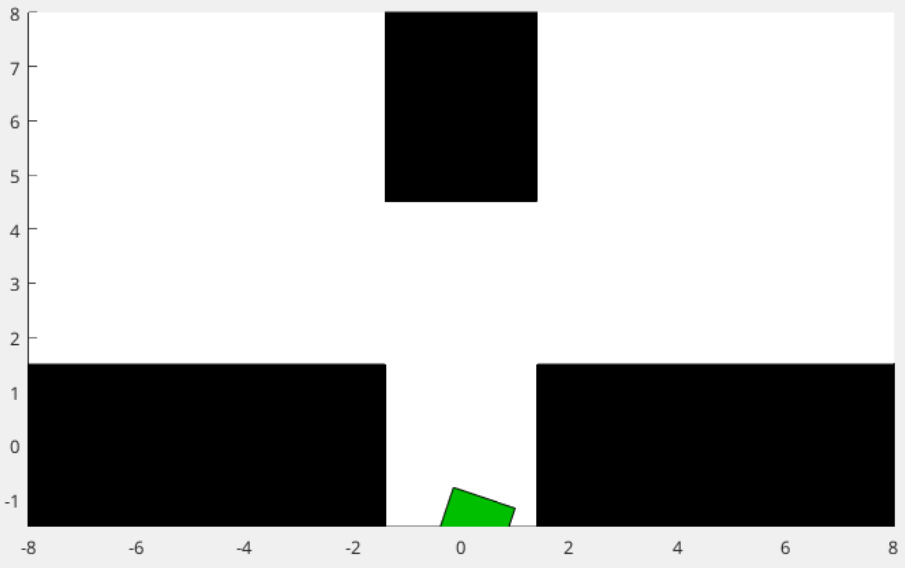
Simulate the environment

```
simout = sim(agent, env)    % Simulate the environment
```

```
simout =

  struct with fields:

      Observation: [1×1 struct]
           Action: [1×1 struct]
           Reward: [1×1 timeseries]
           IsDone: [1×1 timeseries]
   SimulationInfo: [1×1 Simulink.SimulationOutput]

fx >>
```

Zoom: 100% | UTF-8 | CRLF | plotRobot | Ln 10 | Col 4

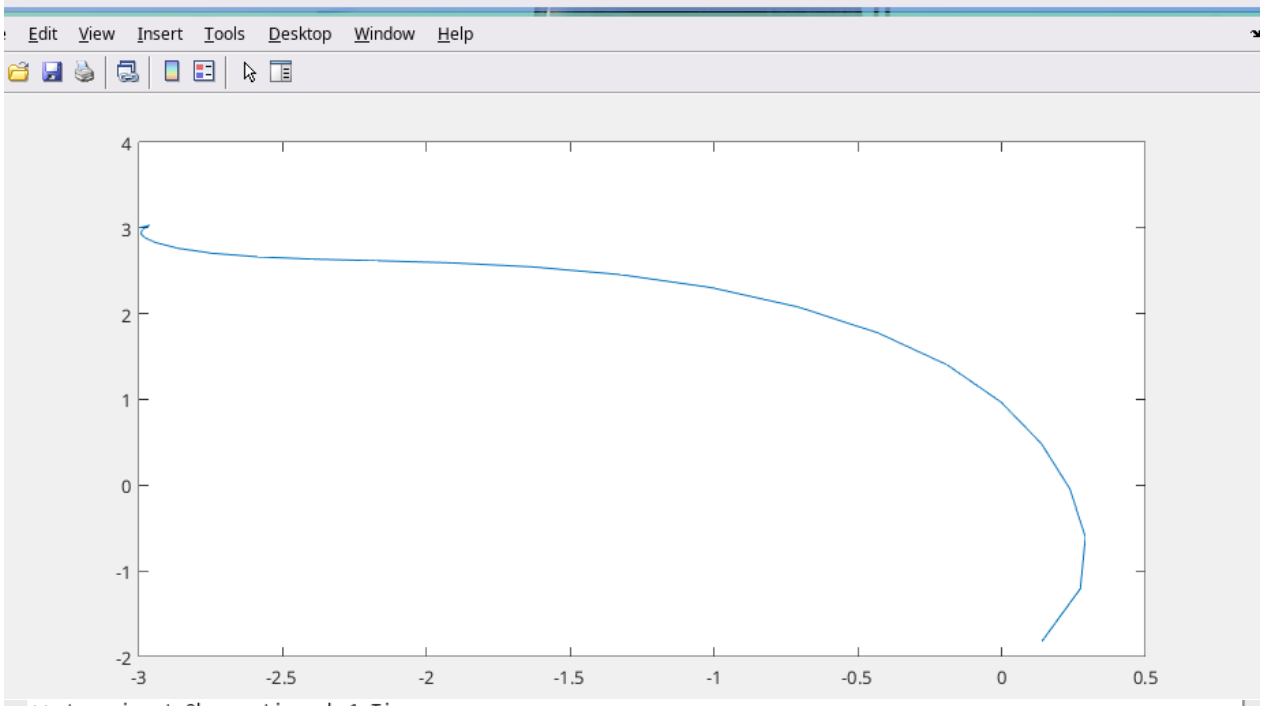
Edit View Insert Tools Desktop Window Help



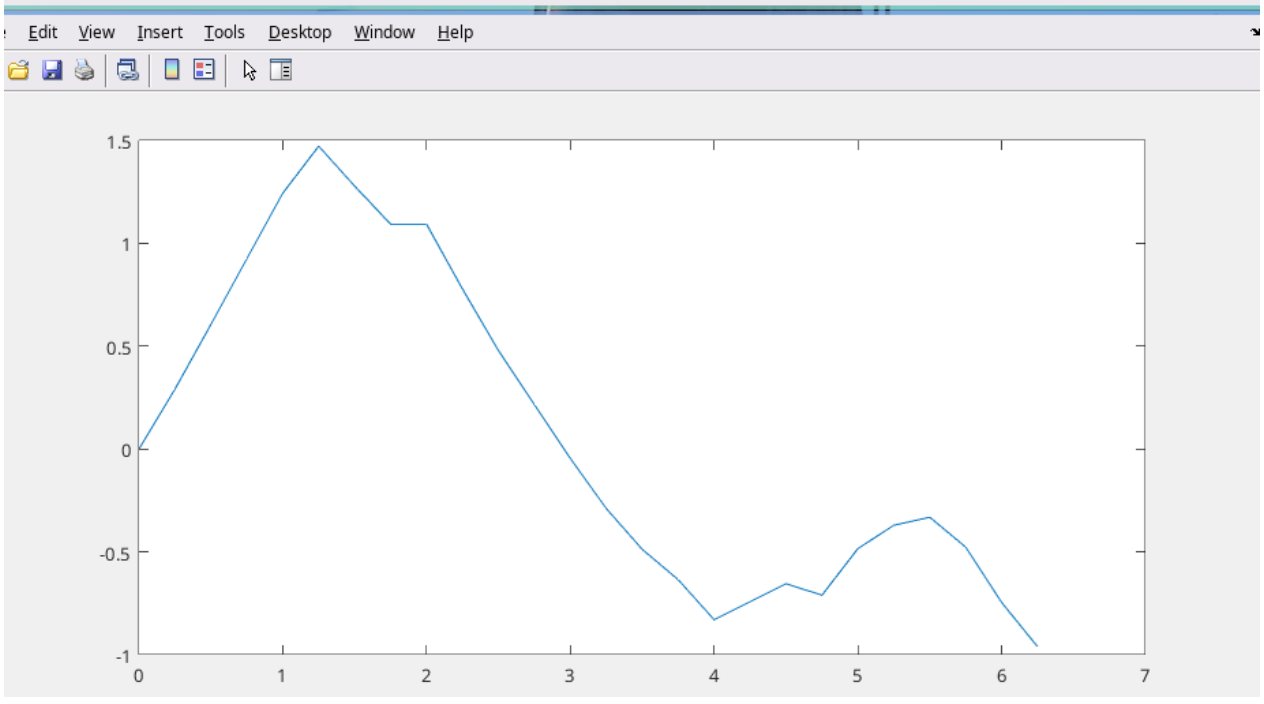
Visualize the simulation results

```
obs = simout.Observation.obs1.Data    % Get the observation data
obsmat = squeeze(obs)                 % Convert the observation data to a matrix
x = obsmat(1,:);                      % Extract the x-coordinate
y = obsmat(2,:);                      % Extract the y-coordinate
plot(x,y)                             % Plot the trajectory

imout.Observation.obs1.Time;     % Get the time data
plot(t,obsmat(6,:))                   % Plot the observation data against time
```

```
>> x = obsmat(1,:);
>> y = obsmat(2,:);
>> plot(x,y)
>>
```



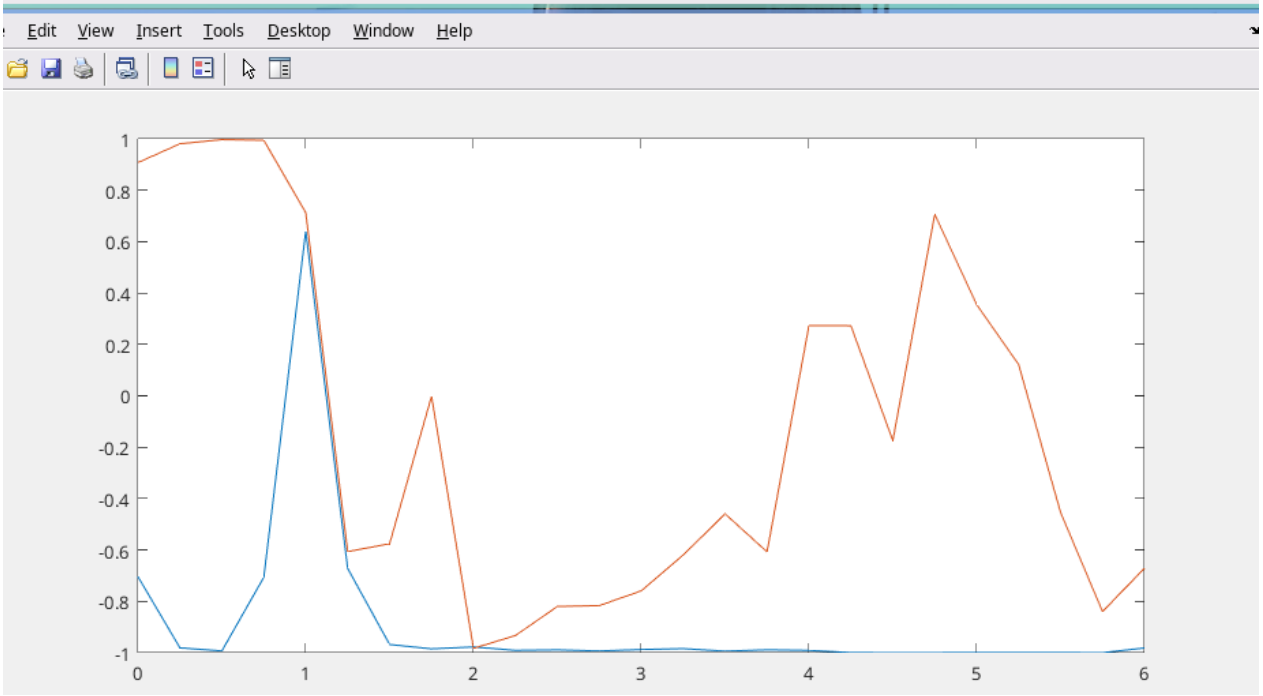

```
>> t = simout.Observation.obs1.Time;
>> plot(t,obsmat(6,:))
>>
```



Visualize the action

```
act = squeeze(simout.Action.act1.Data);      % Get the action data
t = simout.Action.act1.Time;                 % Get the time data
Ftrans = act(1,:);                           % Extract the translational force
Frot = act(2,:);                             % Extract the rotational force
plot(t, Ftrans, t, Frot)                     % Plot the action data against time
```

```
>> act = squeeze(simout.Action.act1.Data);
>> t = simout.Action.act1.Time;
>> Ftrans = act(1,:);
>> Frot = act(2,:);
>> plot(t, Ftrans, t, Frot)
>>
```



Visualize the reward

```
% Change the SimulationOptions to get the reward data
opts = rlSimulationOptions("MaxSteps", 100, "NumSimulations", 5);

simout = sim(agent, env, opts)          % Simulate the environment
s = simout(1)                           % Get the first simulation
t = s.Reward.Time                       % Get the time data
r = s.Reward.Data                       % Get the reward data
plot(t,r)                               % Plot the reward data against time

% Plot the reward data for all simulations
figure
hold on
for k = 1:5
    s = simout(k);
    t = s.Reward.Time;
    r = s.Reward.Data;
```

```
    plot(t,r)
end
hold off
```

```
>> plot(t,r)
>>
```