

---

# random\_agent\_matlab.m

## Table of Contents

Contents .....	1
Code .....	1

**Summary:** Random policy agent used to collect initial data. Contains the functions needed to communicate with the RL-Glue Core.

## Contents

1. helicopter\_agent: set useful paths and function handles
2. helicopter\_agent\_init: initialize data structure
3. helicopter\_agent\_start: take the first step
4. helicopter\_agent\_step: take a step
5. helicopter\_agent\_message: communicate between trainer and agent
6. helicopter\_agent\_end: finish episode
7. helicopter\_agent\_cleanup: save data structure

## Code

```
function theAgent=random_agent_matlab(MDP, codec_base, agent_base)
% Add paths and fill agent structure

    addpath([codec_base 'agent'], [codec_base 'glue'], codec_base);

    theAgent.agent_init=@helicopter_agent_init;
    theAgent.agent_start=@helicopter_agent_start;
    theAgent.agent_step=@helicopter_agent_step;
    theAgent.agent_end=@helicopter_agent_end;
    theAgent.agent_cleanup=@helicopter_agent_cleanup;
    theAgent.agent_message=@helicopter_agent_message;

    global whichMDP;
    whichMDP = MDP;
    global agentDir;
    agentDir = agent_base;

end

function helicopter_agent_init(taskSpec)
% Initialize agent
% Note that helicopter_agent_struct is created and saved in |start| and
% |end|, different from what is done in |GPAgentMatlab|
```

```
end
```

```
function theAction=helicopter_agent_start(theObservation)
```

```
% Take the first step of the agent as the episode starts, and store data
```

```
    global helicopter_agent_struct;  
    global timeStep;  
    helicopter_agent_struct = zeros(2,17);  
    timeStep = 1;
```

```
    theAction = org.rlcommunity.rlg glue.codec.types.Action();  
    theAction.doubleArray = rand(4,1)-1;
```

```
    helicopter_agent_struct(timeStep, 1:12) = theObservation.doubleArray;  
    helicopter_agent_struct(timeStep, 13:16) = theAction.doubleArray;  
    helicopter_agent_struct(timeStep, 17) = 0;
```

```
end
```

```
function theAction=helicopter_agent_step(theReward, theObservation)
```

```
% Take a step and store data
```

```
    global helicopter_agent_struct;  
    global timeStep;
```

```
    theAction = org.rlcommunity.rlg glue.codec.types.Action();  
    theAction.doubleArray = 2*rand(4,1)-1;
```

```
    timeStep = timeStep + 1;  
    helicopter_agent_struct(timeStep, 1:12) = theObservation.doubleArray;  
    helicopter_agent_struct(timeStep, 13:16) = theAction.doubleArray;  
    helicopter_agent_struct(timeStep, 17) = theReward;
```

```
end
```

```
function helicopter_agent_end(theReward)
```

```
% An episode ends and save the data
```

```
    global helicopter_agent_struct;  
    global timeStep;  
    global whichMDP;  
    global agentDir;
```

```
    basename = [agentDir 'randomAgentMatlab/'];  
    filename = sprintf('randomDataMDP%i.mat', whichMDP);  
    fullname = strcat(basename, filename);  
    if exist(fullname, 'file')
```

```
        load(fullname);  
        random_data.x = [random_data.x; helicopter_agent_struct(1:end-1,1:16)];  
        random_data.y = [random_data.y; helicopter_agent_struct(2:end,1:12)];  
        random_data.r_x = [random_data.r_x; helicopter_agent_struct(2:end,1:12)];  
        random_data.r_y = [random_data.r_y; helicopter_agent_struct(2:end,17)];  
        random_data.all = [random_data.all; helicopter_agent_struct];
```

```
    else
```

```
        random_data.x = helicopter_agent_struct(1:end-1,1:16);
        random_data.y = helicopter_agent_struct(2:end,1:12);
        random_data.r_x = helicopter_agent_struct(2:end,1:12);
        random_data.r_y = helicopter_agent_struct(2:end, 17);
        random_data.all = helicopter_agent_struct;
    end
    if timeStep > 1
        save(fullname, 'random_data');
    end
end

function returnMessage=helicopter_agent_message(theMessageJavaObject)
% Custom function for trainer-agent communication

    inMessage=char(theMessageJavaObject);
    global whichMDP;
    if strcmp(inMessage,'What is your name?')==1
        returnMessage='My name is random_agent_matlab';
    elseif strcmp(inMessage,'Which MDP?')==1
        returnMessage=sprintf('I am doing MDP = %i', whichMDP);
    else
        returnMessage='I don\'t know how to respond to your message';
    end
end

function helicopter_agent_cleanup()
% Agent cleanup

end
```

*Published with MATLAB® R2014a*