1 Introduction and Motivation

In This Study, cooperative heterogeneous agents for underwater environment will be developed. For this, cooperative reinforcement learning algorithms will be used. The idea of this learning is that the agents at the beginning don't have any idea about the environment. The environment is resembled by randomly auto-generated maps with randomly allocated targets (static and dynamic). In every new episode, a new map with different allocated targets will be generated. In this way, the agents will acquire solid learning experience and should handle any new situation in any new map.

Exploring hard-to-reach areas, like deep water ship and aircraft wrecks and collecting samples for study, was always a dream for scientists.

Since this kind of environments is unknown, far to reach or to control, large to cover with only one agent, and sometimes dangerous, then a group of autonomous robots should be used to explore these environments and do the tasks.

In this study, the goal is to setup a simulation environment, in which cooperative reinforcement learning algorithms can be evaluated. The environment will be setup in context of underwater robots and use a heterogeneous set of agents.

2 Implementation of the Study

In this study, the learning algorithm will be tested on different maps (world). The number of agents is 4 and they all have different actions. Each of the agents has targets which should be processed. The agents should coopearte by exchanging information about the positions of the targets in different maps.

2.1 The World

The world in this study is assumed to be discrete (Grid World: 15x15 Cells). The movement of the agents is also assumed to be discrete. The dimension of the map is constant for all iterations, however for every new iteration, the location of the targets will be randomly changed. The dimension of the map can be also changed to be bigger or smaller but this hasn't been done in this study. There are two fish in this study. The fish move randomly.

The world as in Figure 2.1) contains the dynamic and the static targets. The world is also resembled in the map as in Figure 2.1).

Figure 2.1: Gridworld as a simulation environment

2.2 Targets

There are dynamic and, static targets in this study. The static targets are the Ships, Plants, and Trash Bins. The dynamic targets are the 2 Fish. as in Figure 2.1). The targets are located in the world and can be seen by the agent as states.

2.3 Kinematics of the Movement

In this study, since the maps are discrete, the agents and the fish are supposed to be moving also in discrete steps as in Figure 2.2).

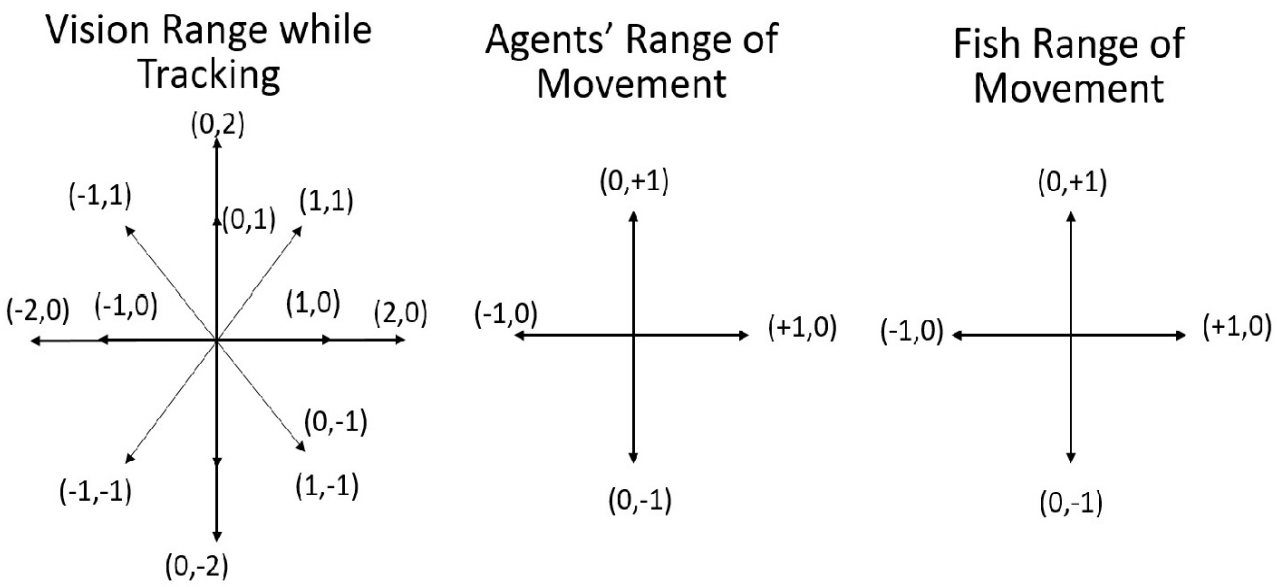

Figure 2.2: Kinematics of Agents, Fish, and Agents' Tracking Range

2.4 Agents

This study is designed for 4 heterogeneous agents and can be extended easily to have more agents with same or with different capabilities. By heterogeneous, it means that the agents have different capabilities (actions). There are 4 agents:

- Recording agent: This agent is responsible for recording the large ship, and also recording the 2 fish.
- Samples-collecting agent: This agent is responsible for collecting samples from the plants.
- Trash-collecting agent: This agent is responsible for collecting the trash in the sea.
- Small-places recording agent: There are small places in small ships, where the normal recording agent cannot get into the ship. This small-places recording agent is supposed to be small to record small ships.

2.5 Actions (Capabilities)

There are shared and unique capabilities. Shared capabilities are: exploration, moving, and tracking. Unique capabilities are: recording, samples collecting, trash collecting, and small places recording.

2.5.1 Exploration Action

All the agents have this capability. The main idea of this action is simply to find new unexplored cells in the map and by exploring new cells, the agent will get a reward. The kinematics for an agent to explore are discrete as in Figure 2.2).

Exploration Action Examples: According to Figure 2.3), there are 4 different exploration cases:

The first case for Agent(2): In this case, all the neighbor cells are explored except one. Then the agent chooses to go in the direction of the unexplored cell as in Figure 2.3).

The second case for Agent(3): In this case, the robot has two possibilities. It can explore in the direction left or down as in Figure 2.3). So it choose one of them randomly.

The third case for Agent(4): In this case, the robot has two (after two steps) unexplored cells. Both of them are in the left direction as in Figure 2.3). That's why it moves in the left direction and then explores again in the next iteration right up or down.

The fourth case for Agent(1): In this case, the robot has two far away unexplored cells. So it goes to the nearest one.

Figure 2.3: Exploration Function examples for 4 different cases

2.5.2 Moving Action

All the agents also have this capability. In order to move, a target needs to be available, otherwise the agent gets a negative reward. However targets are not the same for all agents. The kinematics for an agent to move are the same as the kinematics for the exploration action as in Figure 2.2).

Moving Action Examples: According to Figure 2.4), there are 4 different kinds of movings actions:

The first case for Agent(4): In this case, the agent has one target. So it calculates the distance needed (x+y), and goes to it in steps in every iteration.

The second case for Agent(2): In this case, the agent has two targets, so it calculates the distance (x+y) to them. It finds out that both distances are the same. So it choses to go to one of them randomly.

The third case for Agent(3): In this case, the agent has three targets, but it's already meeting the position of one of them. so If the learning algorithm chooses to move, the agent should standstill.

The fourth case for Agent(1): In this case, the robot has many targets including the dynamic target (fish) which has been seen by the agent(3). So it should move to the nearest target which is the fish in this case.

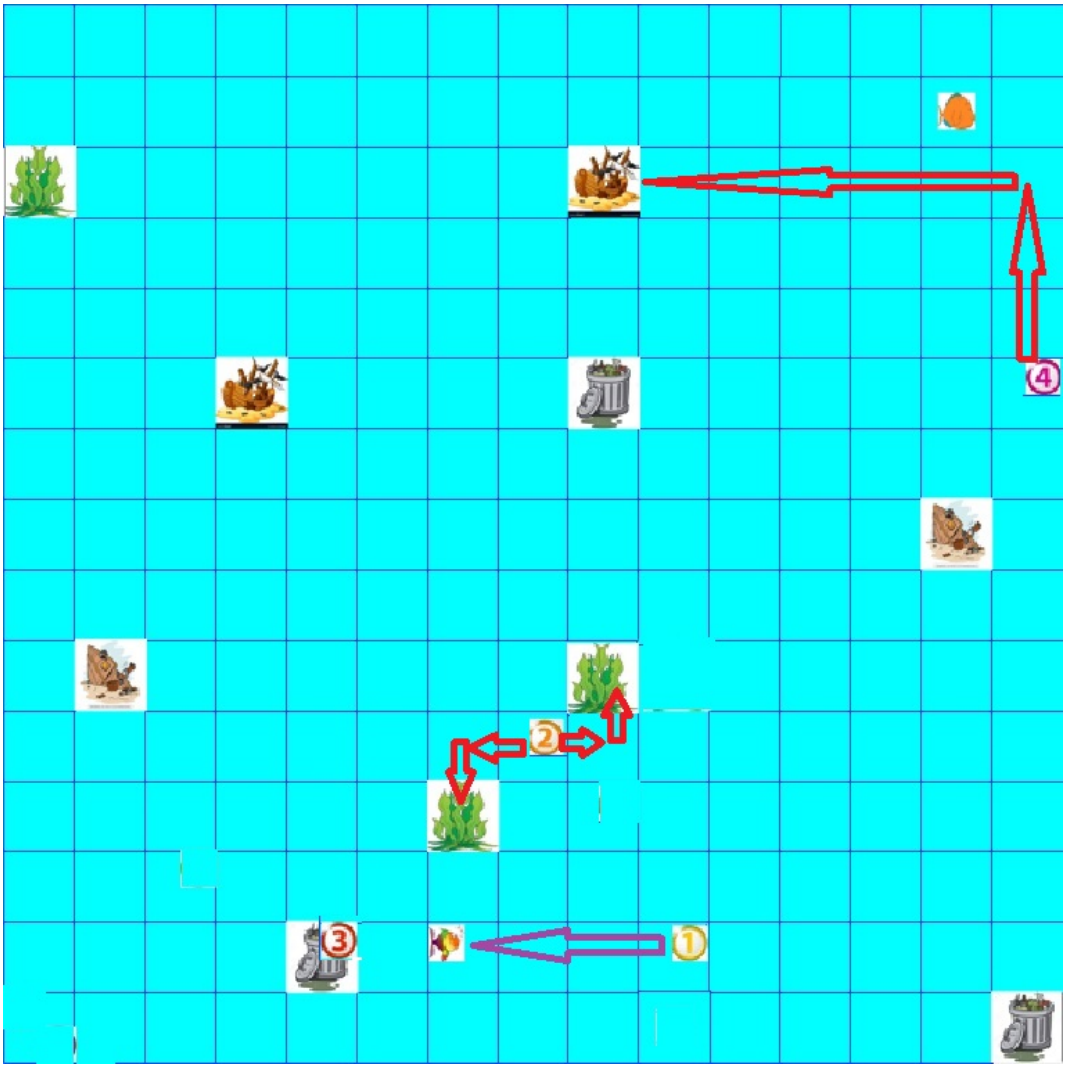

Figure 2.4: Move Function examples for 4 different cases

2.5.3 Tracking action

All the agents have this capability. Tracking is a special technique for all the agents for dynamic targets like the fish. In order to track, then the agent needs to have a Vision Range as in Figure 2.2). All the agents have this vision range and it's dedicated only

for the dynamic targets like the fish.

Fish Movement: In this study, it's assumed that the Fish moves randomly in 4 Directions. Since the world is 2D, then the possible movements of the fish are: Up, Down, Right, and Left as shown in Figure 2.2).

2.5.4 Scanning Action of 4 Agents

Scanning Function is dedicated to find dynamic targets (Fish) as in Figure 2.5). During the development and the learning process, it has been found that the fish might be located in an area which is far away from the agents. To solve this, a new action has been implemented, where all the agents are located in the columns with a displacement between them in the y-direction as in Figure 2.5). As soon as they are located in the map, they start scanning upwards and downwards while keeping the y-displacement betweeen them in order to catch the fish.

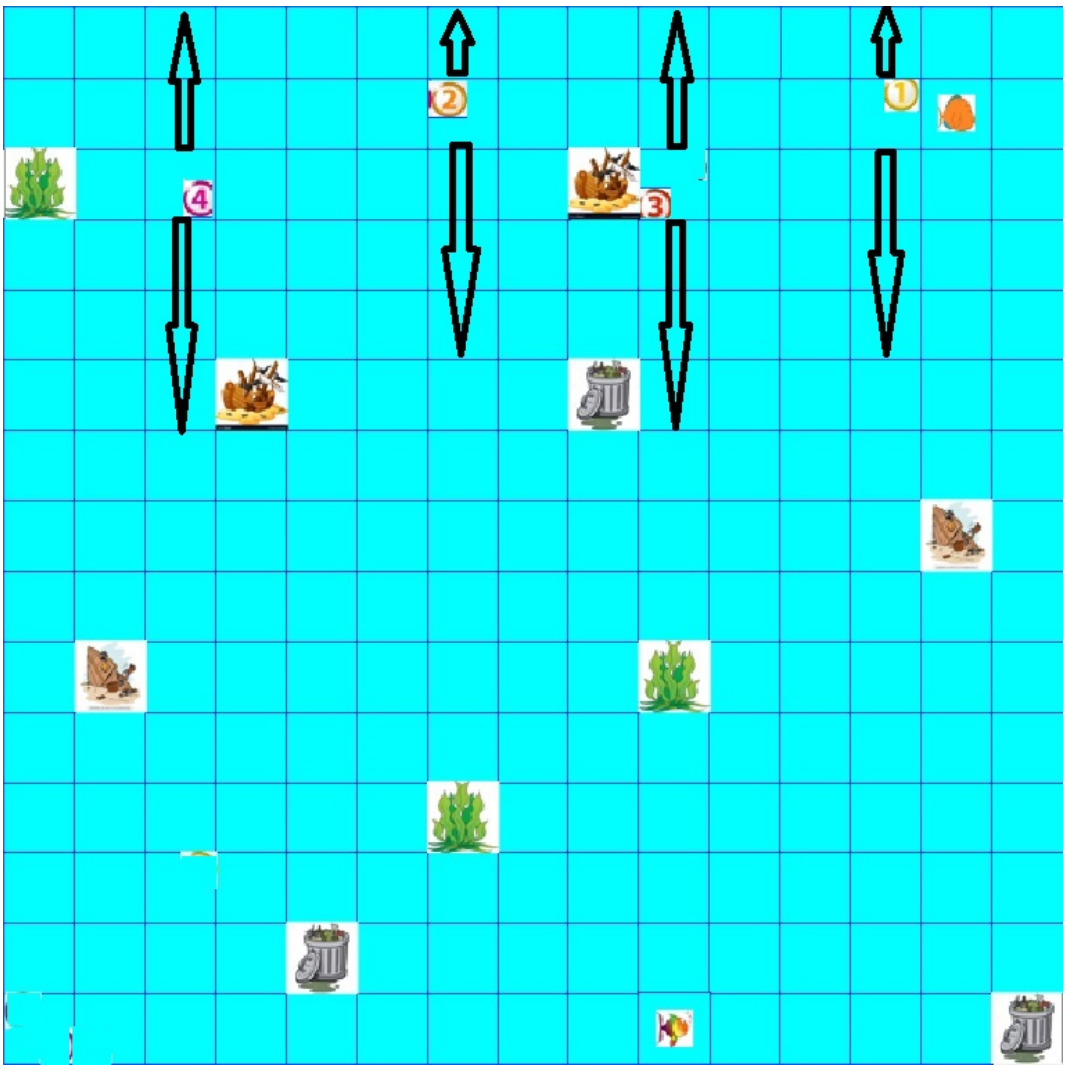

Figure 2.5: Scanning Function to find the Fish

2.6 Mapping of (States, Actions) between the Agents and the World

(Mapping of (states, actions) between world and learner in Figure 2.6)

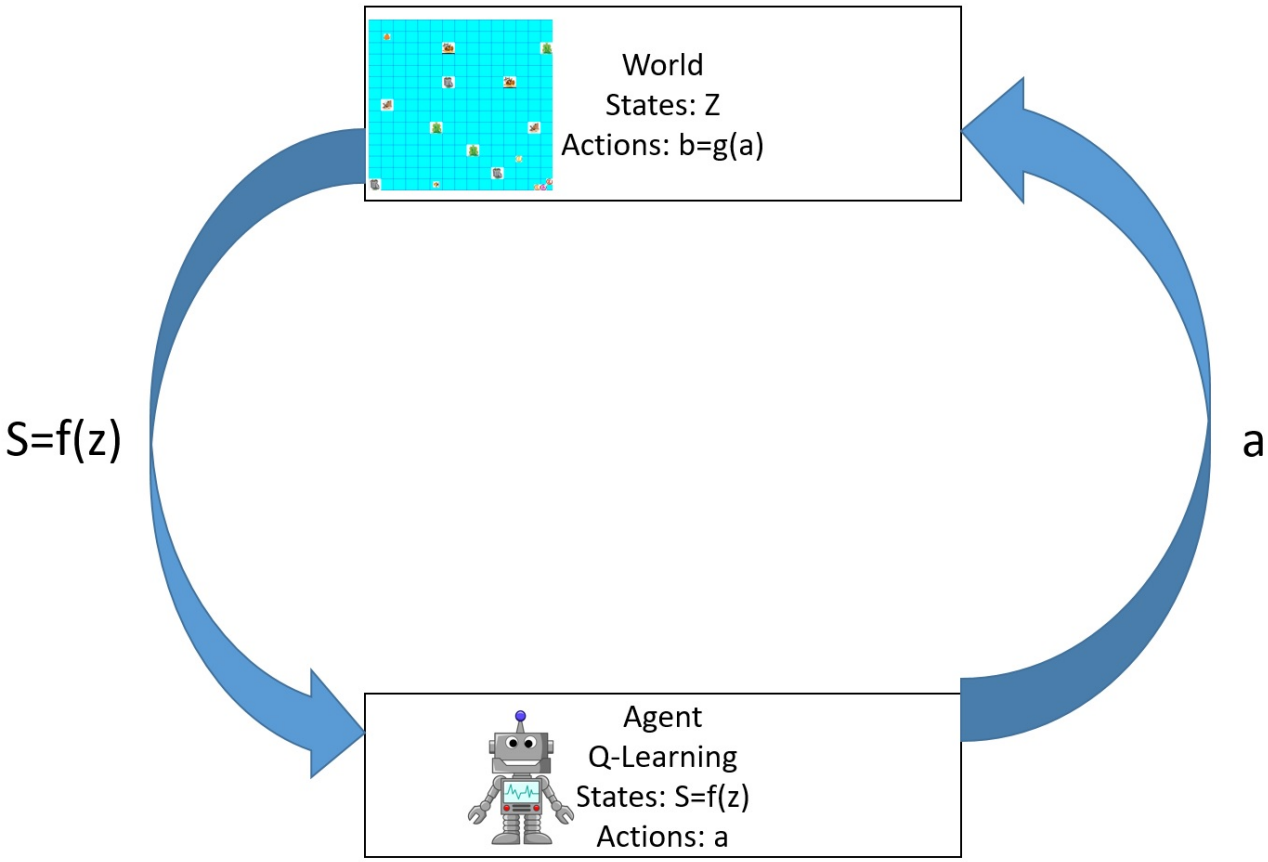

Figure 2.6: Mapping of (states, actions) between world and Learner

In this study, the states and actions of both the world and the learning algorithm are different. As in Figure 2.6 the agent's states (s) are dependent on the world's states (z), whereas the action in the world (b) is dependent on the agent's action (a).

The agent gets its state from the world state. If the world state is for example a fish, then the state of the agent turns into a fish state.

If the learning algorithm chooses for example an exploration action, then according to the exploration function, the agent should move to the next unexplored area.

At first, the agent gets its state from the world state, then the agent takes an action. This action is implemented in the world. The world state changes. The agent gets its new state from the world state and so on.

2.7 Markov Decision Processes of the Agents and the Meaning of the States and Actions

Following below are the MDPs of the agents. The MDP for every agent consists of series of states and actions.

2.7.1 First Agent (Recording Agent)

This agent has targets (ships, plants, and fish). These targets are at the same time states. The following states are found for the recording agent:

- Exploration State
- Ship State
- Plant State
- Fish State
- Near-Fish State
- Fish-and-Ship State
- Fish-and-Plant State
- Fish-near-Ship State
- Fish-near-Plant State

As shown in Figure 2.7), At first the Recording agent is in Exploration State (Null State). He starts exploring and stays in (Null State) until he or another agent finds a target, which is of importance for the Recording agent. Then the agent starts moving to this target and when he reaches this target, then he starts recording this target. Recording Action lasts for 5 steps (iterations). When this target is recorded then this target turns again into exploration state for the agent. It means that the agent doesn't care about this target and sees it as Null.

Another State is the Fish State. Since the Fish is a dynamic target, then it should be tracked by the agent. The agent does tracking action when the fish is in his tracking range as shown in Figure 2.2. In this case the state of the agent is: Near-fish-state. When the recording agent meets the position of the fish then his state turns into fish state. When the agent is in fish state then he should do recording action.

Another possible state is: Fish and ship state. In this case the agent should choose whether to record the fish or the ship. In this case it's better to record the fish since it's dynamic and harder to get than static targets. The same applies for fish and plant state.

Another possible state is: Fish near ship state. In this case, the agent should decide whether to record the ship or track the fish. This also applies for the fish near plant state.

2.7.2 Second Agent (Samples Collecting Agent)

The Second Agent (Samples Collecting Agent) has the following States:

- Exploration State

- Plant-Sample State
- Fish State
- Near-Fish State
- Fish-and-Plant Sample State
- Fish-near-Plant Sammple State

The Second Agent (Samples-Collecting Agent) has the following Actions:

- Exploring Action
- Moving Action
- Samples-Collecting Action
- Tracking Action
- Scanning Action

As shown in Figure 2.8, at first the samples-collecting agent is in exploration state (null state). He starts exploring and stays in (null state) until he or another agent finds a target, which is of importance for the samples-collecting agent. Then the agent starts moving to this target and when he reaches this target, then he starts collecting this target. Samples-collecting action lasts also for 5 steps (iterations). When this target is collected then this target turns again into null state for the agent. It means that the agent doesn't care about this target and sees it as null.

Another state is the fish state. Since the fish is a dynamic target, then it should be tracked by the agent. The agent does tracking action when the fish is in his tracking range as shown in Figure 2.2. In this case, the state of the agent is: near-fish-state. When the samples-collecting agent meets the position of the fish, then his state turns into: fish state. When the agent is in fish state then he can also take tracking action. This is due to the fact, that the fish is moving and the agent should track it.

Another possible state is: Fish and plant-sample state. In this case, the agent is located in the same cell where the plant and the fish are located and he should choose whether to track the fish or collect the plant-sample.

Also another possible state is: Fish near plant-sample state. In this case, the agent should decide whether to collect the plant-sample or track the fish.

2.7.3 Third Agent (Trash Collecting Agent)

The Third Agent (Trash Collecting Agent) has the following States:

- Exploration state
- Trash collecting state

- Fish state
- Near-fish state
- Fish-and-trash state
- Fish-near-trash state

The third agent (Trash collecting Agent) has the following Actions:

- Exploring Action
- Moving Action
- Trash-Collecting Action
- Tracking Action
- Scanning Action

As shown in Figure 2.9, At first the trash-collecting agent is in exploration state (null state). He starts exploring and stays in (null state) until he or another agent finds a target, which is of importance for the trash-collecting agent. Then the agent starts moving to this target and when he reaches this target, then he starts collecting this target. Trash-collecting action lasts for 5 steps (iterations). When this target is collected, then this target turns again into exploration state for the agent. It means that the agent doesn't care about this target and sees it as null.

Another state is the fish state. Since the fish is a dynamic target, then it should be tracked by the agent. The agent does tracking action when the fish is in his tracking range as shown in Figure 2.2. In this case the state of the agent is: Near-fish-state. When the trash-collecting agent meets the position of the fish then his state turns into: Fish state. When the agent is in fish state then he can also take tracking action.

Another possible state is: Fish and trash state. In this case the agent should choose whether to track the fish or collect the trash.

Another possible state is: Fish near trash state. In this case the agent should decide whether to collect the trash or track the fish.

2.7.4 Fourth Agent (Small Recording Agent)

The fourth agent (Small Recording Agent) has the following States:

- Exploration State
- Small-Place State
- Fish State
- Near-Fish State

- Fish-and-Small-Place State
- Fish-near-Small-Place State

The fourth Agent (Small-Places-Recording Agent) has the following Actions:

- Exploring Action
- Moving Action
- Recording Action
- Tracking Action
- Scanning Action

As shown in Figure 2.10, at first the small-recording agent is in exploration state (null state). He starts exploring and stays in (null state) until he or another agent finds a target, which is of importance for the small-recording agent. Then the agent starts moving to this target and when he reaches this target then he starts recording this target. Small-places-recording action lasts also for 5 steps (iterations). When this target is recorded then this target turns again into exploration state for the agent. It means that the agent doesn't care about this target and sees it as null.

Another state is the fish state. Since the fish is a dynamic target, then it should be tracked by the agent. The agent does tracking action when the fish is in his tracking range as shown in Figure 2.2. In this case, the state of the agent is: Near-fish-state. When the small-place agent meets the position of the fish then his state turns into fish state. When the agent is in fish state then he can also take tracking action.

Another possible state is: fish and small-place state. In this case, the agent should choose whether to track the fish or record the small-place.

Another possible state is: fish near small-place state. In this case, the agent should decide whether to record the small-place or track the fish.

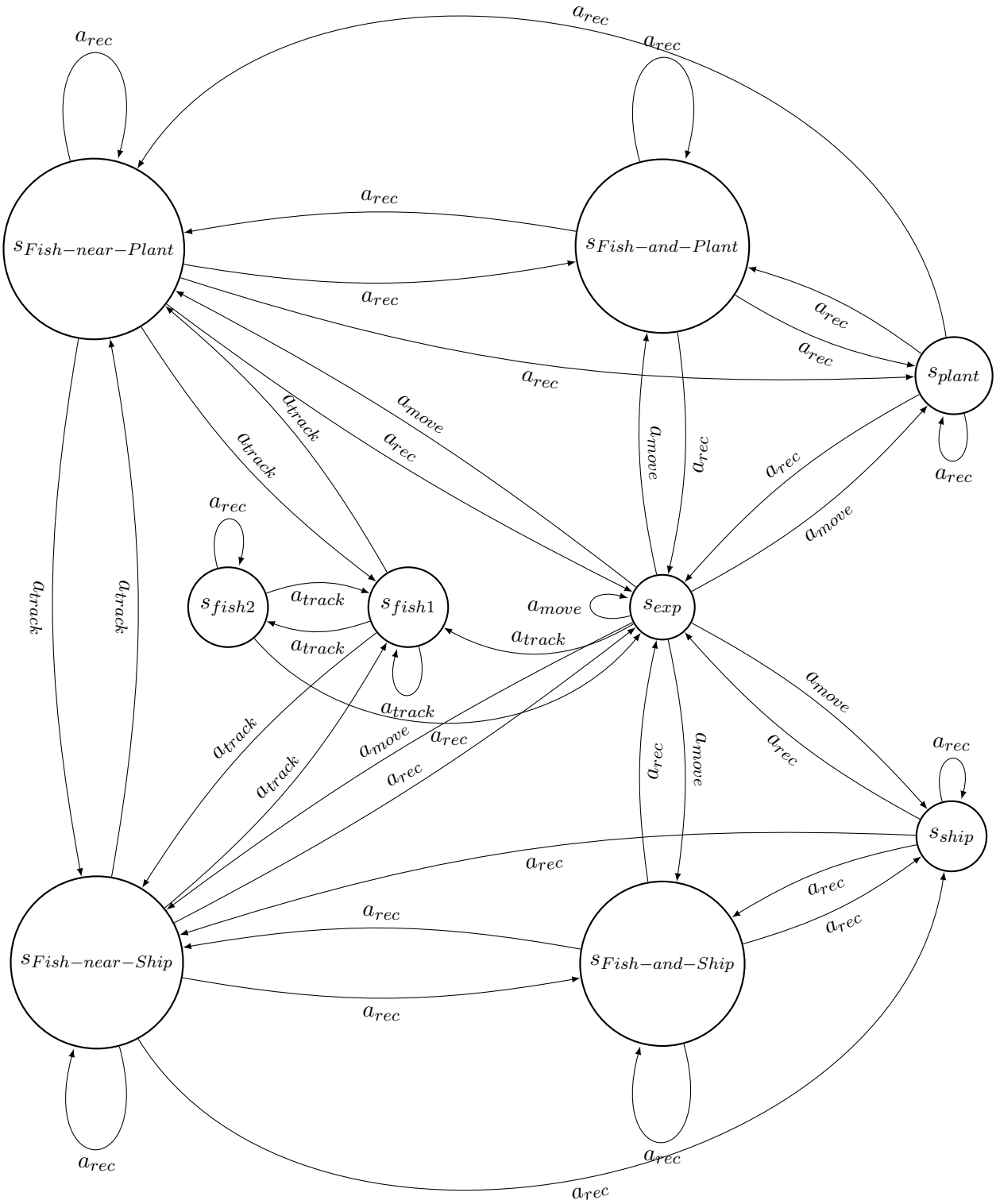

Figure 2.7: Recording Agent

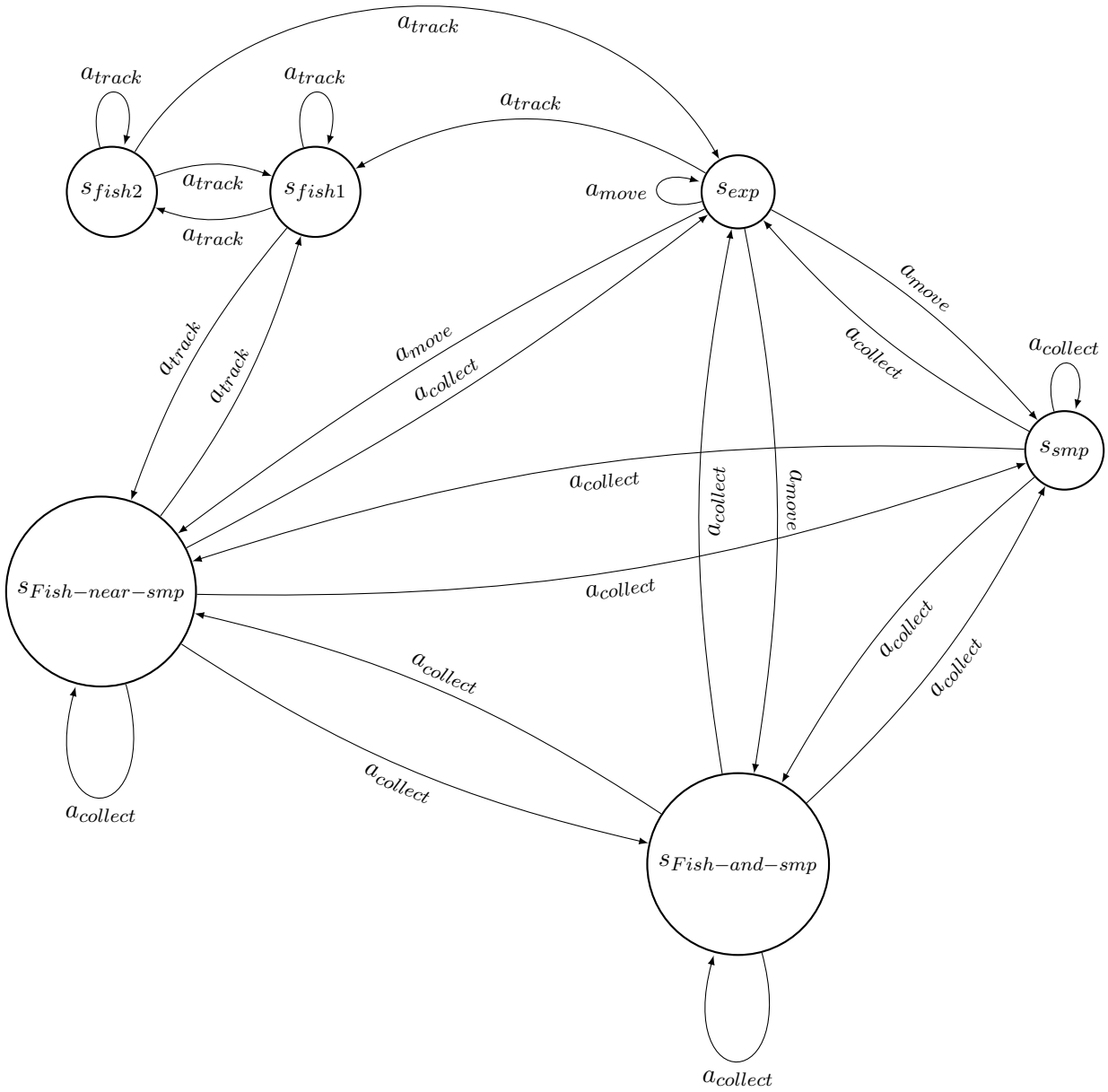

Figure 2.8: Samples Collecting Agent

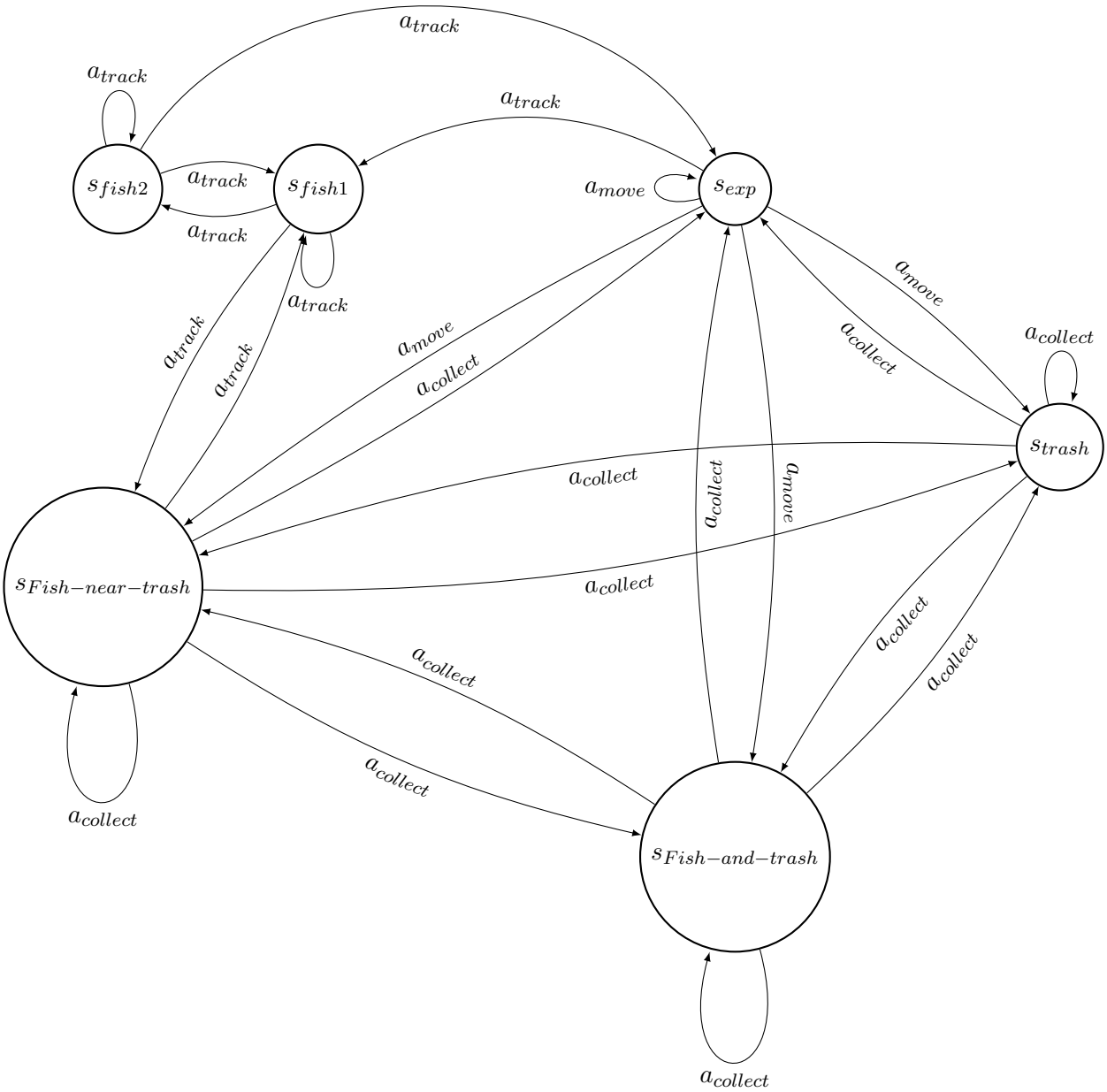

Figure 2.9: Trash Collecting Agent

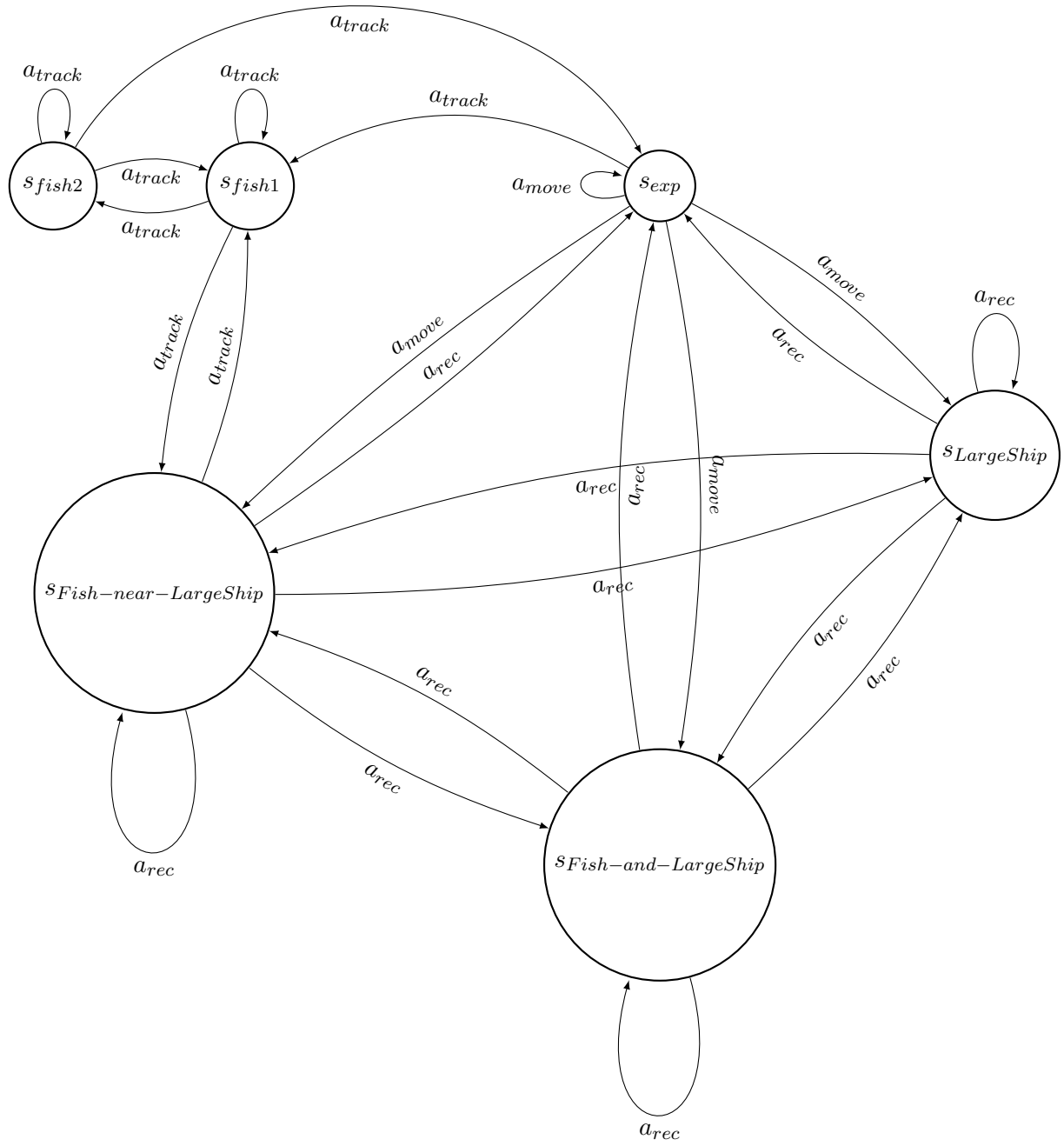

Figure 2.10: Small Recording Agent

3 Info-States

In this Study, Info-States are used to get statistics about the environment and embed them in every Joint-State to control the learning process.
In this Study the following Info-States are used:

- Ship-Info State.
  - This State has binary variable. It shows whether a Ship has been seen by any agent or not.
- Plant-Info State
  - This State has binary variable. It shows whether a Plant has been seen by any agent or not.
- Plant-Sample State:
  - This State has binary variable. It shows whether a Plant-Sample has been seen by any agent or not.
- Trash State
  - This State has binary variable. It shows whether a Trash has been seen by any agent or not.
- Small-Place State
  - This State has binary variable. It shows whether a Small-Place has been seen by any agent or not.
- Fish State
  - This State has binary variable. It shows whether a Fish has been seen by any agent or not.
- Exploration-info State
  - This info-State has an enumerator of 4 values. It shows how many areas have been explored in the world.

4 Rewards

In this study, since the whole mission is fully cooperative, the reward is calculated by adding the rewards from 4 agents. In this Study, different kinds of rewards with different values are used. The reason for this is the different actions, different targets, and the different needed time to reach the targets.
6 different Rewards with different values have been used:

- Exploration Rewards.
  - Exploration has a high priority in this study. Exploration is done for every grid-cell in this grid-world. All other actions depend on the exploration-action. In order to move to a target, this target has to be explored and found. In order to process a target (record, or collect), then the agent should move to the target after it has been explored. According to this, a positive reward of 1 has been used for an exploration of every new unexplored cell. However it's important to mention that Exploration is not only rewarded by choosing the exploration action. It can also be rewarded during the exploration of unexplored cells, by choosing move-action, scanning-action, and even track-action. In this way, the agents will not only care about the exploration of targets, but also about moving, tracking, scanning, and processing them.

- Moving, Scanning, and Tracking Rewards
  - Moving, Scanning, and Tracking actions are neither negatively nor positively rewarded. This kind of actions is zero rewarded.

- Recording of Static Targets:
  - Recording of static targets is positively rewarded. A positive reward of 1 is used for every record action.

- Recording of Dynamic Targets
  - To process dynamic targets, scanning, move, and tracking actions have to be used. According to this and in terms of time, dynamic targets are expensive to process. That's why dynamic targets are positively rewarded with 1.9.

- Delay Penalty Reward
  - The goal in this study is not only to explore and process all the targets, but also to finish the mission as fast as possible. According to this, a negative reward of 0.05 has been added to the sum of the 4 rewards in every iteration.

- Wrong Action Penalty Reward
  - When an agent does and action in the wrong place, then it should be punished. The value of the punishment has been calculated so that it influences the maximum sum of the rewards of all the agents. As an example for this, if rec-agent does rec-action for a fish then it gets +1.9. If smp-agent does collect-action for a sample then it gets +1. If trash-agent does collect-action for a trash then it gets +1. The sum of these 3 rewards is 3.9. If the fourth agent does wrong action then it should be higher than the sum of the positive rewards of other agents with negative sign. So it should be lower than -3.9. That's why in this study, this kind of rewards is chosen to be -4.