On the Mechanisms of Imitation in Multi-Agent Systems

Mehmet D. Erbas
Faculty of Engineering and Natural Sciences
Istanbul Kemerburgaz University
34217, Istanbul, Turkey
Email: mehmet.erbas@kemerburgaz.edu.tr

***Abstract*—Imitation is a social learning method in which an individual observes and mimics another's actions. To implement imitation on robots, a number of questions should be answered, including what information should be copied during imitation, how to choose the behaviors to be copied and how to translate the observed behaviors. In this research, we aim to answer the first two questions in an experiment scenario with simulated agents. First, based on the content of information that is copied during imitation, we compare two different imitation methods, namely, imitation of actions only and imitation of actions and perceptions. It is shown that if the observed behaviors are highly context specific, imitating perceptions along with actions is beneficial compared to imitating actions only. Second, to answer the question of which behaviors to copy, we compared different selection strategies. It is shown that the agents can choose which behaviors to copy by checking the utility of observed behaviors by a trial and error mechanism.**

I. Introduction

Imitation is a social learning method in which an individual observes and copies another individual's actions. A well-known definition from Thorndike [1] claims that imitation is the activity of *"learning to do some act from seeing it done"*. As the definition suggests, there is a direct link between imitation and learning. Because it allows information and skill transfer between agents, it has been seen as an important form of cultural learning [2]. As a results, imitation have been widely studied by both biologists and psychologists. Biological research on imitation is mainly interested in the adaptive value of imitation and psychological research on imitation examines the mechanisms and functions of imitation [3].

The study of imitation has received much attention in Robotics research. Dautenhahn et al. claimed that [4] the study of imitation in robotics holds the promise of overcoming the need to program every behavior a robot might need to perform. A robot that is able to imitate can learn new actions by observing the demonstrations of those actions. Bakker and Kuniyoshi [5] claimed that the observed actions that are copied by imitation is valuable as they are executed by an agent sharing the same environment. As a result, an agent with the ability to imitate has an increased level of adaptation to its environment. As the expectations from imitation is high, there have been many research on the topic, leading to the area of imitation learning. Some example research that used imitation to train robots are [6], [7], [8], [9].

Learning by imitation is different from other adaptive learning algorithms that have been widely used in robotics research, including evolutionary algorithms [10], reinforcement learning [11] or supervised learning [12] as learning by imitation exploits social interactions. An agent that is able to imitate can observe model behaviors that are executed by other agents that share the same environment. After the observation, the agent should find the matching behaviors of its own to imitate the observed behaviors. Yet, the observed behaviors should become a part of the individual learning process of the imitating agent. As a result, in order to implement imitation, a number of questions should be answered:

1) How to translate the observed behaviors.
2) What information should be copied during imitation.
3) How to choose the behaviors to be copied.

To answer the first question, an agent that imitates should translate the observed behaviors to its own set of actions. This can be done by matching each observed behavior with a behavior of its own. The problem of matching the actuators of the demonstrator agent to the imitator's actuators is presented as the correspondence problem [13]. This issue is solved by programming in most of the research that use imitation. That is, there are some procedures that automatically translate the observed actions into a set of actions that can be executed by the imitating agent.

The second question is about what information should be copied during imitation. Based on the content of the information that is copied during imitation, Winfield and Erbas [14] identified at least three types of imitation:

- *Imitation of actions only:* An agent copies another's sequence of movements, lights, or sounds. With this type of imitation, the imitator only records a sequence of actions that are executed by the demonstrator.
- *Imitation of actions and perceptions:* An agent copies another's sequence of actions along with the environmental effects that triggered those sequence of actions. The imitator can then enact the copied behaviors in a similar environmental context.
- *Imitation of goals:* An agent copies the goal or intentions of another's. It is possible that the imitator, once it copies the intentions of the demonstrator, may execute different set of actions to achieve the same goal.

In this research, two types of imitation methods, namely imitation of actions only, and imitation of actions and perceptions, are compared in an experiment scenario. The simulated agents learn to achieve a task by using Reinforcement Q Learning Algorithm [15] and imitation enhance their learning. The agents that imitate actions only copy the executed actions that are performed by other agents. The agents that imitate actions and perceptions copy the executed actions and also record in what context these actions are executed. It is shown that the agents that imitate actions and perceptions can learn faster if the executed actions are context specific.

To answer the third question, the imitator agent should be able to determine which demonstrated actions are beneficial to itself and copy specifically those actions that are beneficial (or at least, expected to be beneficial). In most of the research that use imitation in conjunction with individual learning, the agent that imitates is able to imitate a teacher or mentor agent and the observed behaviors are expected to be beneficial [16], [17], [18], [19]. In these research, the imitating agent has access to the internal state or expectations of the observed agent. The experiences or the expectations of the observed agent are used to train the imitator agent. In these research, the task that the imitator has to achieve is to learn the optimal policy or the unknown reward function of the expert agent which is supposed to be implicitly followed in the expert's behavior. Erbas et al. [20] had a different approach in which the imitator has no access to the internal state of the observed agent. The only information that is transferred between the agents is the set of actions that are performed by the demonstrator. Finding in what context these observed behaviors is beneficial (or if these actions are beneficial at all) is determined by a trial and error mechanism as the imitator tries those observed actions in a number of different states. In this way, it has been shown that imitation of purely observed behaviors speeds up learning.

In this research, different selection strategies are compared and it is shown that the agents can choose which behaviors to copy by checking the utility of observed behaviors in a trial and error mechanism.

The paper organized as follows: In section II, we explain the experimental setup. In section III the models for different types of imitation are presented. Section IV presents the experiments to compare the models of different types of imitation. In section V, different types of selection strategies are presented and their effects on the performance of the learning agents are examined. Finally section VI concludes the paper and mentions some further research questions.

II. Experiment Setup

To explore different aspects of imitation in a multi agent group, we simulated agents that employ the Q-learning algorithm [15] to achieve a foraging task. The environment in which the agents operate is a grid world (Fig. 1). In each experiment run, there are some randomly placed obstacles that limits the possible actions of the agents. During the experiments, in each time unit the agents can move to one of the eight neighboring cells of their current position. The agents start from their initial position and try to reach the target location.

A learning agent uses an ϵ-greedy algorithm in which it determines the action with the highest Q value in its current state. With a probability of $1-\epsilon$, it chooses that action and executes it. With a probability of ϵ, it chooses a random action and executes it. In this way, the agent updates the Q value for its current state and chosen action by using the formula below:

$$Q(s_t,a_c) = Q(s_t,a_c) + \alpha[r_{t+1} + \gamma max_a Q(s_{t+1},a) - Q(s_t,a_c)] \quad (1)$$

in which a_c is the chosen action, α is the learning rate, γ is the discount factor, and r_{t+1} is the reward for getting to state s_{t+1} from the state s_t. The only state that gives a reward is the goal state.

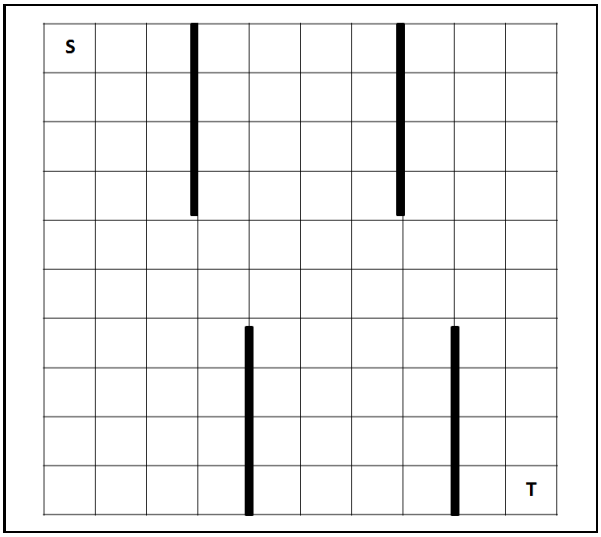

Fig. 1. The arena is a 10 x 10 grid world. The agents start from the top-left corner (S) and they try to reach the target which is at bottom-right corner of the arena (T). In the figure four obstacles that limit the possible moves of agents in adjacent cells, are placed on the arena (thick lines). In each run, two obstacles of length four units are placed on the top of the arena and two obstacles of length four units are placed on the bottom of the arena.

When the agents reach the target, they return to their starting position and try to reach the target again. One experiment run takes 100,000 time units, during which the Q values for each state action pair have enough time to converge to their final values. Every 100 time units, the shortest path to target location that can be achieved by the agent is determined by using a greedy action selection method on the current Q values.

III. The Models of Imitation

A. Imitation of Actions Only

The agent that can imitate actions only are able to observe and replicate the sequence of action that are executed by other agents. Based on this algorithm, at each time unit, the agent can be in one of the three states:

- *Moving:* The agent makes its move based on the ϵ-greedy Q learning algorithm, as explained in section II.

- *Observing:* While in *moving* state, with a probability equal to 0.01, the agent gets into the observing state and stays static for five time units. It chooses an agent to observe the actions that are executed by that agent. At the end of this period, the agent saves the observed list of actions into its memory and gets back to *moving* state. There is a cost of imitation such that the agents lose time while they watch other agents.
- *Imitating:* While in *moving* state, with a probability equal to $p_{imitate}$, the agent gets into the *imitating* state and enacts one of the list of actions that it previously observed, by executing the actions in the selected list of actions one by one while updating related Q values of the state action pairs. If these actions contradict the Q values of the agent, i.e. they have lower Q values compared to other actions in the same state, or if there is an obstacle that prevents the action to be executed, then the selected list of actions is abandoned and the agent acts according to its original ϵ-greedy algorithm.

The pseudo-code for the controller of the agents is given in Algorithm 1.

Once in imitating state, the agent executes the actions that are previously observed as it checks if those actions contradict its Q values. If the agent has an alternative action in its current state with a higher Q value, the observed action is rejected and the agent acts based on its original ϵ-greedy algorithm. Therefore, imitation provides model behaviors to the learning agent, if the agent does not have an action with a high Q value in its current state. This can only happen in the regions of the state-action space that the agent has not explored yet. As the agent executes observed actions, those actions become a part of its individual learning process and assists the agent in unexplored regions of the state-action space.

As stated above, while in *moving* state, the agent probabilistically selects a list of actions and with a probability equal to $p_{imitate}$, the agent gets into the *imitating* state (These procedures are executed at line 10 and line 11 of Algorithm 1). To choose which list of actions to enact and to calculate $p_{imitate}$, the agent performs a *Q value test* that is implemented in [20]. According to this method, the agent calculates the sum of Q values for all state-action pairs, and compares this value to the sum of Q values for all state-action pairs when the enacted list of actions is completed or abandoned. If there is an increase in the Q values, it means that the observed set of actions takes the agent closer to the goal state. If there is no increase, it means that the observed set of actions cause the agent to explore some part of the state-action space which do not contribute to the agent's overall performance. The method depends on the past experience of the agent, and it tries to determine the utility of observed set of actions based on the temporal differences in Q values.

The ratio of which there is an increase in the Q values, R_i, is calculated for each list of actions as follows:

n^i_{Q+} = *number of times Q values increased when list of actions i is enacted*

$n^i_{replicated}$ = *number of times list of actions i is enacted*

$$R_i = \frac{n^i_{Q+} + 1}{n^i_{replicated} + 1} \tag{2}$$

R_i is used to regulate the imitation probability of each list of actions and to choose which list of action to imitate. If the agent has *n* distinct list of actions in its memory, the probability of choosing the list of action i is calculated by:

$$P^i_{choose} = \frac{R_i}{\sum_{k=1}^{n} R_k} \tag{3}$$

The probability of enacting list of actions i once it is chosen is:

$$p^i_{imitate} = \beta R_i \tag{4}$$

in which β is a constant that regulates the initial imitation probability.

The agents can store up to ten lists of actions in their memory. Whenever the memory is filled up and a new list of actions is observed, among the lists of actions currently in the memory, the list of action that has the minimum R_i value is removed and the new list is recorded in the empty slot.

B. Imitation of Actions and Perceptions

As explained in the previous section, the agents that imitate actions and perceptions are able to copy the actions that other agents execute and they record in what context those actions are executed. For that purpose, along with each list of actions that is observed, the imitator store the starting position of each list; that is, the position of the demonstrator when the observation starts. The exact position of start and eight neighboring cells are declared as the *region of interest* for that list of actions. Based on this information, if the agent has *n* distinct list of actions in its memory, the formula (3) is updated as:

If the imitator is in region of interest of list of actions i

$$P^i_{choose} = \frac{R_i}{\sum_{k=1}^{n} R_k} \tag{5}$$

else

$$P^i_{choose} = 0 \tag{6}$$

So, whenever the agent goes into imitating state, it can only enact the list of actions that has been executed in the close proximity of the imitator's position. Other observed lists of actions, that have been executed in different regions of the arena, are ignored. The rest of the controller is exactly same with the agent that imitates actions only.

IV. Experiments on Different Imitation Methods

A. Agents Imitating an Inexperienced Agent

In the first set of experiments, three agents are placed on the arena. The first agent uses pure ϵ-greedy Q learning algorithm only, so it can not imitate. The second agent imitates actions only and the third agent imitates actions and perceptions. Both imitating agents, when they are in observing mode, observe the actions of the no-imitation agent. As shown in Fig. 1, four obstacles are randomly placed on the arena. The agents can not

Algorithm 1 Pseudocode for imitation of actions only algorithm

Input:
$Q(s,a) \leftarrow 0$ *for all state action pairs*
$currentState \leftarrow Moving, actionList \leftarrow \emptyset$
$s \leftarrow StartPosition, selectedPath \leftarrow 0$
for $time = 0 : 100,000$ **do**
  if $currentState = Moving$ **then**
    if $0.01 > random()$ **then**
      $currentState = Observing$
    else
      $i \leftarrow SelectActionList(actionList)$
      if $p^i_{imitate} > random()$ **then**
        $selectedPath \leftarrow i$
        $currentState \leftarrow imitating$
      else
        $a \leftarrow max_{a'}Q(s,a')$
        if $\epsilon > random()$ **then**
          $a \leftarrow SelectRandomAction()$
        end if
      end if
    end if
  else if $currentState = Imitating$ **then**
    $a \leftarrow GetNextAction(selectedPath)$
    if $\exists a' Q(s,a') > Q(s,a)$ **then**
      $a \leftarrow max_{a'}Q(s,a')$
      if $\epsilon > random()$ **then**
        $a \leftarrow SelectRandomAction()$
      end if
      $selectedPath \leftarrow 0$
    end if
  else if $currentState = Observing$ **then**
    $newList \leftarrow ObserveNewAction()$
    if $ListCompleted(newList)$ **then**
      $actionList \leftarrow AddNewList(newList)$
      $currentState \leftarrow Moving$
    end if
  end if
  $Q(s,a) \leftarrow Q(s,a) + \alpha[r + max_{a'}(s',a') - Q(s,a)]$
  $s \leftarrow s'$
  if $s = TargetPosition$ **then**
    $s \leftarrow StartPosition$
  end if
end for

cross these obstacles. At each 100 time units, the length of the shortest path that can be achieved by each agent is calculated and this metric is used for comparing the performance of different agents. One experiment run takes 100,000 time units and the same experiment is repeated 100 times.

Fig. 2 shows the results for this experiment set. As all agents start to learn at the same time, none of them is more experienced than the others. As a result, the agents can not fully benefit from their imitation activity. Nevertheless, it can be observed that, although statistically not significant, the agents that can imitate have slightly better performance than the no-imitation agent. The imitating agents lose time while they stay static during observation; however, the gain they achieve by imitation compensates for the loss of time during observation. This is due to the fact that, in some runs, the observed behaviors have a positive effect on the learning speed of the agents. By checking the utility of each observed list of actions, the agents are able to determine which list of actions is beneficial to them. However, as the demonstrator is not experienced at all, imitation does not help much. When we compare two different imitation methods, although the agent that imitates actions and perceptions has a slightly better performance, we can not observe any statistically significant difference in their relative performances in this experiment set.

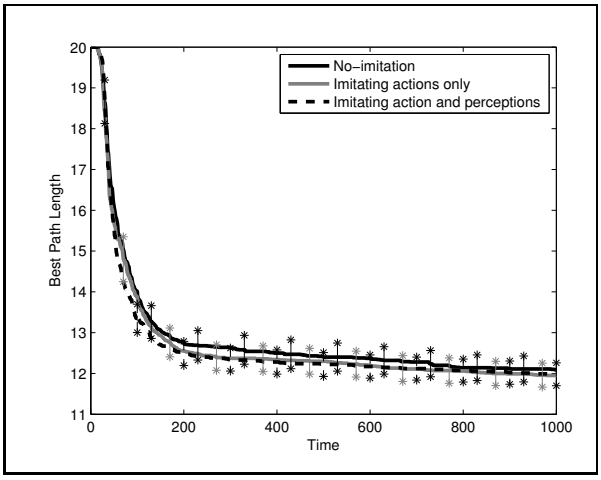

Fig. 2. Shortest achieved path length for imitating an inexperienced agent. The results are mean shortest path length from 100 experiment runs along with 95% confidence intervals (note that the confidence intervals are shown at specific intervals). The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. Parameters are set as follows: $\epsilon = 0.1, \beta = 0.5, \alpha = 0.2, \gamma = 0.7$. Time is given in 100 time units.

B. Agents Imitating an Experienced Agent

In the second set of experiments, initially we placed one agent that uses pure ϵ-greedy Q learning algorithm. The agent is trained for 90,000 time units while it becomes an experienced no-imitation agent. Then, at the $90,000^{th}$ time unit, two agents, one imitates actions only and the other imitates actions and perceptions, are placed on the arena. The imitating agents can observe the actions that are executed by the no-imitation agent. The no-imitation agent completes its life time at the $100,000^{th}$ time unit and the other agents continue to be trained for another 90,000 time units, by using the copied list of actions that they have in their memory. So each agent stay on the arena for 100,000 time units. Note that, at the $90,000^{th}$ time unit, the no-imitation agent may achieve the shortest path but because of its random move probability, it can still execute some actions that are not part of the shortest path. Once again, four obstacles are randomly placed on the arena and the same experiment is repeated 100 times.

Fig. 3 shows the results for this experiment set. As can be seen, both imitating agents have statistically significant better performance compared to the no-imitation agent. A pairwise ttest reveals that the difference between the no-imitation agent and the agent that imitates actions only is statistically significant until $50,000^{th}$ time unit as the difference between the no-imitation agent and the agent that imitates actions and perceptions is statistically significant during the whole experiment. So the agents that can imitate are able to get model behaviors from the experienced agent and enhance their learning by enacting those model behaviors. When we compare two types of imitation methods, although the difference is minimal after $20,000^{th}$ time unit, the agent that imitates actions and perceptions has a better performance compared to the agent that imitates actions only. The reason is that the agent that imitates actions and perceptions is able to test an observed model behavior in the environmental context in which the model behavior is demonstrated. As a results, it can highly exploit the information that it gets from the experienced agent.

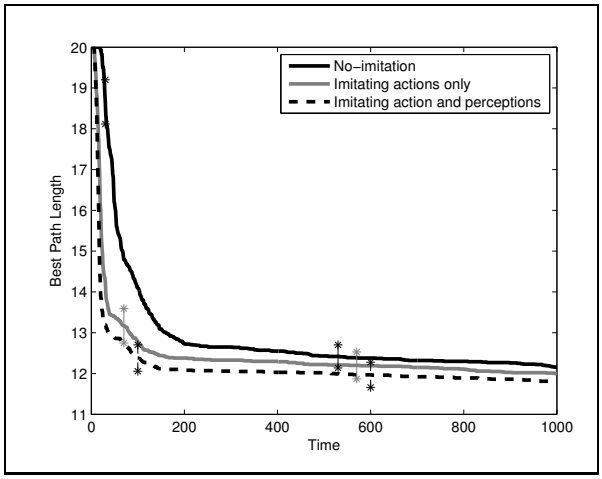

Fig. 3. Shortest achieved path length for imitating an experienced agent. The results are mean shortest path length from 100 experiment runs along with 95% confidence intervals. The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. Parameters are set as follows: $\epsilon = 0.1, \beta = 0.5, \alpha = 0.2, \gamma = 0.7$. Time is given in 100 time units.

Fig. 4 and fig. 5 shows two lists of actions that are recorded and then enacted highest number of times by an agent that imitates actions and perceptions during one experiment run. The shaded regions mark the region of interest of each list of action. As can be seen, the first list of action makes the agent avoid the first obstacle and move towards the middle of the arena. So it is only meaningful around the starting position of the agent. Similarly, the second list of action let the agent avoid another obstacle and move towards the target. The second list of action is meaningful in its region of interest.

C. Agents Imitating an Experienced Agent in an Environment with Bottleneck States

In the previous set of experiments, it is shown that the agent that imitates actions and perceptions is able to determine the

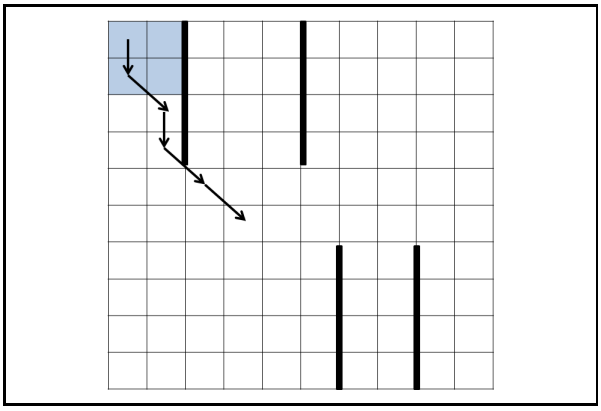

Fig. 4. List of actions that are copied and enacted highest number of times. It consist of one move towards South (S), followed by one move towards South-East (SE), one move towards S and two moves towards SE.

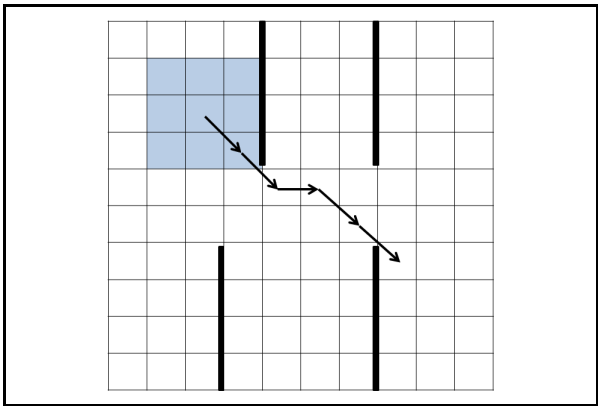

Fig. 5. List of actions that are copied and enacted highest number of times. It consist of two moves towards SE, followed by one move towards East (E), two moves towards SE.

observed behaviors that are beneficial to itself and enact those behaviors in the appropriate regions of the arena. Based on this observation, it can be hypothesised that if the actions that are observed are highly context specific, copying perception would be advantageous. To test this hypothesis, an arena with two bottleneck states is formed, as shown in fig. 6. These bottleneck states divides the environment into three distinct regions and these states should be visited to move from one region to another. The important property of this setting is that, the actions that should be executed to move towards the target in the first and third regions are different from the actions that should be executed in the second region. On this arena, an agent that can not imitate is trained for 500,000 time units. At the $490,000^{th}$ time unit, two agents, one imitates actions only, the other imitates actions and perceptions are placed on the arena. Each agent stays on the arena for 500,000 time units during which they learn the shortest path to the target location. Both imitating agents, when they are in observing mode, copy the actions of the no-imitation agent.

Fig. 7 shows the results for this experiment set. As can be seen, the difference in the learning speed of the no-imitation agent and the agent that imitates actions only is minimal but

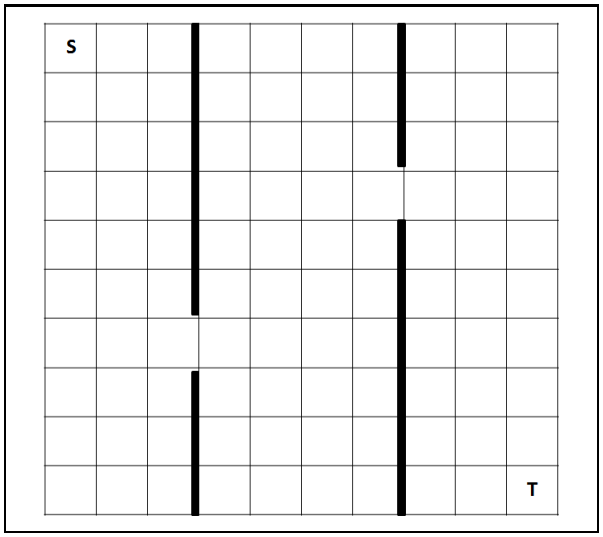

Fig. 6. Arena with bottlenecks. The obstacles that are placed on the arena roughly divides the arena into three distinct regions. These bottleneck states should be visited to reach the target.

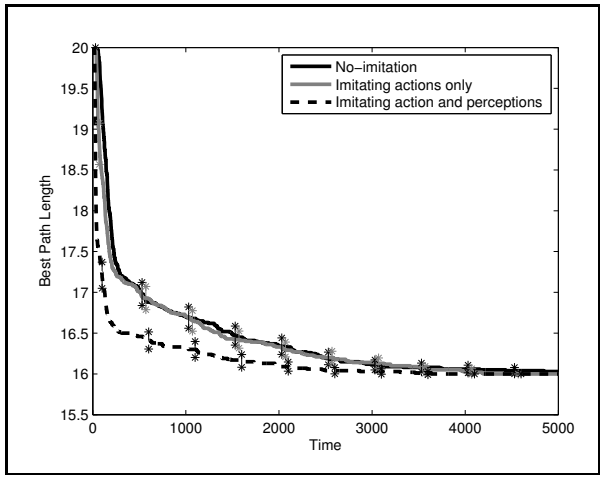

Fig. 7. Shortest achieved path length for imitating an experienced agent on an arena with bottleneck states. The results are mean shortest path length from 100 experiment runs along with 95% confidence intervals. The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. Parameters are set as follows: $\epsilon = 0.1, \beta = 0.5, \alpha = 0.2, \gamma = 0.7$. Time is given in 100 time units.

the agent that imitates actions and perceptions outperforms both other agents. The reason for this results can be explained as followed: the obstacles that are placed on the arena in this experiment roughly divides the arena into three distinct parts. The actions that are meaningful in first and third regions are different from the actions that are meaningful in the second region. Therefore an observed behavior in one of the regions is may not be helpful in another region. As the agent that imitates actions only has no information about the region in which the observed actions are executed, it is not able to choose appropriate observed behaviors for each region. However, the agent that imitates actions and perceptions is able to determine the lists of actions that are suitable to each region and enact those lists of actions accordingly. Hence, we can deduce that, copying perceptions along with actions is highly beneficial if the observed set of behaviors are context specific.

Fig. 8 further explains why the agent that imitates actions only can not benefit from imitation. In the figure, average number of imitation attempts and average number of actions executed in imitating state is shown for two imitating agents. As can be seen, the agent that imitates actions only makes more imitation attempts but has to abandon imitation very often. The agent that imitates actions and perceptions has low number of imitations attempts but once in imitating mode, it executes more actions based on its observation. Therefore, the agent that imitates actions and perception is able to exploit the information that it gains from imitation to enhance its learning speed.

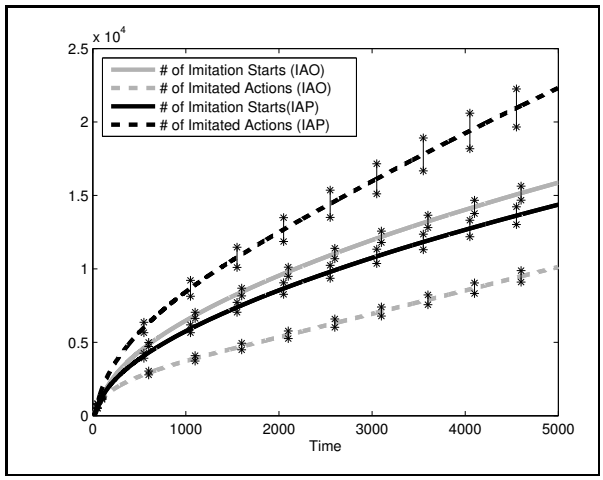

Fig. 8. Average number of imitation attempts and average number of actions executed in the imitating state for the agents with two different types of imitation. Gray line shows average imitation attempts and gray dashed line shows the average number of imitated actions for the agents that imitate actions only (IAO). The black line shows average number of imitation attempts and black dashed line shows average number of imitated actions for the agents that imitate actions and perceptions (IAP). Time is given in 100 time units.

V. Experiments on which agent to copy

In a multi-agent system, past experience of agents is a valuable source of information. To use imitation in a multi-agent system, the agent that imitates should be able to select which behavior to copy. As shown in the previous section, to enhance learning via imitation, the imitator agent should observe and replicate the actions of an experienced agent that shares the same environment. Therefore, the agents should have a mechanism to detect the actions that are beneficial to them. In most of the research that use agent to agent imitation, the imitator observes the behaviors of a teacher or mentor agent and the observed behaviors are expected to be beneficial [16], [17], [18], [19]. However, in a multi-agent system, the agents may interact with many agents with different levels of experience. In addition, an agent with a low level of experience may be more successful than an agent with high level of experience. In this section, we examine the question of which agent to copy in an experiment scenario with

multiple agents. For this purpose, we examined three different selection strategies:

1) The agents copy the most *experienced* agent. The agent select the agent that has the highest experience, in terms of spent time on the arena. This selection strategy is similar to the method that is used in past research as the agent has a mentor or teacher agent with high level of experience.
2) The agents copy the most *inexperienced* agent.
3) The agents copy a random agent. In this setting, it is possible that an agent selects a different agent to copy each time it gets into the *observing* mode.

In the experiments, initially an agent that can imitate is placed on the arena shown in fig. 1. Then, after every 1000 time units, a new agent is placed on the arena. Each agent stays on the arena for 100,000 time units and in this way 100 agents are trained in one experiment. So the first agent is active in between time units 0 to 100,000, the second agent is active in time units between 1000, and 101,000 time units and so on. One experiment run takes 199,000 time units and as usual, every 100 time units, the shortest path achieved by each agent is calculated.

Fig. 9 shows results for imitating actions only and Fig. 10 shows results for imitating actions and perceptions. As can be seen, for both imitation methods, the agent that uses the first selection method have a slightly better performance than no-imitation agents. As the agents that are selected for imitation have not much experience, the agents that imitate can not enhance their learning speed much by imitating the inexperienced agents. The agents that use second method and the agents that use the third method highly outperform no-imitation agents. Interestingly, agent that use the second method and the agents that use the third method have very similar performances. This result can be explained as follows: The agents that imitate are able to determine the list of actions that are beneficial to them. As the agents that select a random agent to imitate, copies a number of agents with different levels of experience, they are able to detect the ones with higher performance and imitate their demonstrated actions. As a result they can achieve a similar performance compared to the agents that are guaranteed to select the most experienced agents of the group.

Fig. 11 compares the performance of two imitation methods, namely, imitation of actions only and imitation of actions and perceptions, for each of the selection methods. As can be seen, for all three cases, the agents that imitates actions and perceptions have slightly better performance than agents that imitate actions only.

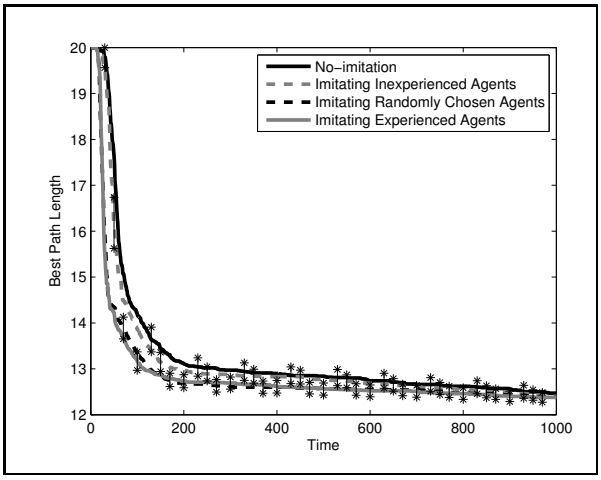

Fig. 9. A group of agents copying each other by using imitating actions only method. The results are mean shortest path length for 100 agents that are placed on the arena in one experiment run along with 95% confidence intervals. The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. Parameters are set as follows: $\epsilon = 0.1, \beta = 0.5, \alpha = 0.2, \gamma = 0.7$. Time is given in 100 time units.

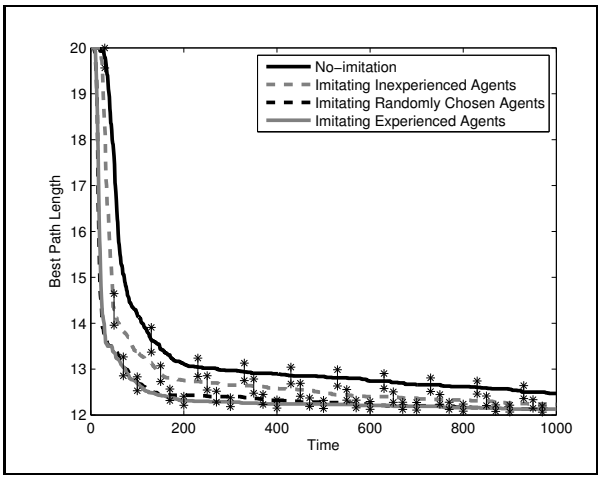

Fig. 10. A group of agents copying each other by using imitating actions and perceptions method. The results are mean shortest path length for 100 agents that are placed on the arena in one experiment run along with 95% confidence intervals. The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. The β value that regulates the initial imitation probability is set to 0.5 for these experiments. Time is given in 100 time units.

VI. Conclusion

In this research, we aim to investigate two different aspects of imitation in a multi-agent system. First, we attempt to find an answer to the question of what information should be copied during imitation. It is shown that if the actions that are executed are highly context specific, copying perceptions along with actions is beneficial. An imitating agent that copies model behaviors of other agents that share the environment, can further enhance its learning speed, if it checks the utility of the observed behaviors in a similar context with the demonstrator. The imitation method that is presented in this paper depends on pure observation as the imitating agent has no access to the internal state or expectations of the demonstrator agent. Second, we attempt to find an answer to the question of which behaviors should be copied. It is shown that the agents can utilise a trial and error mechanism to select the behaviors that are beneficial to them.

The two imitation methods are compared in an experiment scenario in which the agents learn to achieve a simple task

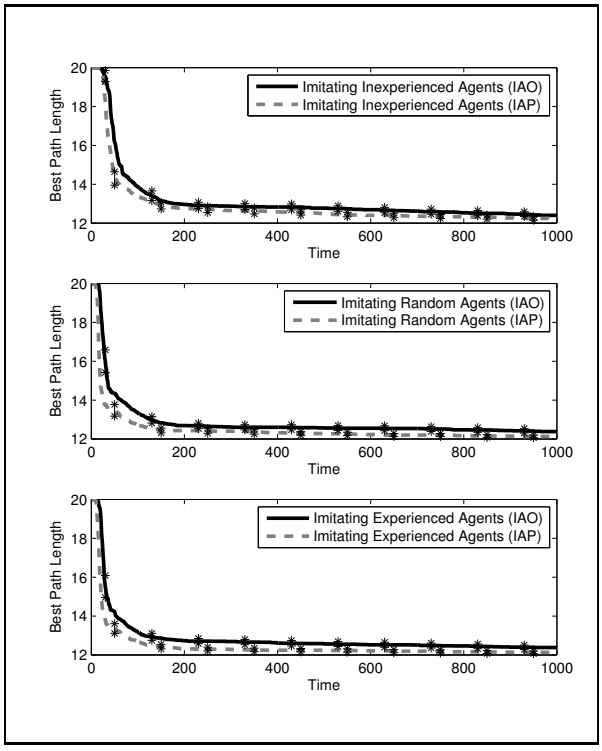

Fig. 11. Comparison of two imitation types, namely, imitation of actions only (IAO) and imitation of actions and perceptions (IAP) based on three different agent selection method. The results are mean shortest path length for 100 agents that are placed on the arena in one experiment run along with 95% confidence intervals. The shortest path achieved by each agent is calculated every 100 time units by using a greedy action selection algorithm on the current Q values. The β value that regulates the initial imitation probability is set to 0.5 for these experiments. Time is given in 100 time units.

of finding the shortest path to a target location on the arena. The task that the agents learn to achieve does not require any interaction between agents (except imitation) or interactions between the agents and the objects on the arena. With a more complex task which requires further interactions, such as collective foraging or collective transportation, it should be possible for the imitating agent to copy the interactions between agents and the interactions between agents and the environment. This can further explain the effects of copying perceptions during imitation.

The agents that can imitate in this research use a simple abstract model of perfect imitation in which the executed actions are transferred between agents. In another research, Erbas et al. [21] examined the effects of embodied imitation on the structure of imitated behaviors during multiple cycles of imitation. It was shown that the variations in the real robots' sensors and actuators allow certain behaviors to emerge during multiple cycles of imitation. These adapted behaviors appeared to be more robust to the uncertainties of the embodied imitation process so that they can be imitated with high fidelity. If the imitation of actions and perceptions is implemented and tested on real robots, we would observe sensor and actuator errors that would add variation to the observed actions and perceptions. Testing imitation of actions and perceptions on real robots may help us to further examine the effects of this imitation method on the learning speed of agents.

ACKNOWLEDGMENT

This work was supported by TUBITAK (research grant 113E588).

REFERENCES

[1] E. L. Thorndike, Animal Intelligence: An Experimental Study of the Associative Process in Animals, in Psychological Review Monograph, vol. 2, pp. 551-553, 1898.
[2] M. Tomasella, A. C. Kruger, and H. H. Ratner, Cultural Learning, in Behavioral and Brain Sciences, vol. 16(3), pp. 495-552, 1993.
[3] T. R. Zentall, Imitation in Animals: Evidence, Fucntion, and Mechanisms, in Cybernetics and Systems, vol. 32, pp. 63-96, 2001.
[4] K. Dautenhahn, C. L. Nehaniv, and A. Alissandrakis, Learning by Experience from Others - Social Learning and Imitation in Animals and Robots, Adaptivity and Learning: An Interdisciplinary Debate, R. Kahn, R. Menzel, U. Ratsch, M. M. Richter, I. O. Stamatescu, editors, pp. 217-241, Springer Verlag, Springer Verlag, 2003.
[5] P. Bakker, and Y. Kuniyoshi, Robot See, Robot Do: An Overview of Robot Imitation, in AISB96 Workshop on Learning in Robots and Animals, pp. 3-11, 1996.
[6] P. Gaussier, S. Moga, J. P Banquet, and M. Quoy, From Perception-Action Loop to Imitation Process: A Bottom-Up Approach of Learning by Imitation, Applied Artificial Intelligence, vol. 7(1), pp. 701-729, 1998.
[7] R. Dillmann, Teaching and Learning of Robot Tasks via Observation of Human Performances. Journal of Robotics and Autonomous Systems, vol. 47(2-3), pp. 109-116, 2004.
[8] M. Nicolescu, and M. J. Mataric, Task Learning through Imitation and Human-Robot Interaction, in C. L. Nehaniv and K. Dautenhahn, editors, Imitation and Scoail Learning in Robots, Humans and Animals, pp. 407-424, Cambridge University Press, 2007.
[9] S. Calinon, and A. Billard, Incremental Learning of Gestures by Imitation in a humanoid robot, in Proceedings of ACM/IEEE International Conference on Human-Robot Interaction, pp. 255-262, 2007.
[10] S. Nolfi, and D. Floreano, Evolutionary Robotics, MIT Press, 2000.
[11] A. G. Barto, S. J. Bradtke, and S. P. Singh, Learning to Act Using Real-time Dynamic Programming, Artificial Intelligence, vol. 6, pp. 105-122, 2004.
[12] D. E. Rumelhart, G. E. Hinton, and R. J. William, Learning by Representation by Back-Propagation, Nature, vol. 323, pp. 533-536, 1986.
[13] C. L. Nehaniv, and K. Dautenhahn, The Correspondence Problem, C. L. Nehaniv, and K. Dautenhahn, editors, Imitation in Animals and Artefacts, pp. 41-61, MIT Press, 2002.
[14] A. F. T. Winfield, and M. D. Erbas, On Embodied Memetic Evolution and the Emergence of Behavioural Traditions, Memetic Computing, vol. 3(4), pp. 261-270, 2011.
[15] A. Barto, and R. S. Sutton, Reinforcement Learning: An Introduction, MIT Press, 1998.
[16] B. Price, and C. Boutilier, Accelerating Reinforcement Learning through Implicit Imitation, Artificial Intelligence Research, vol. 19, pp. 569-629, 2003.
[17] D. C. Bentivegna, C. G. Atkeson, and G. Cheng, Learning from Observation and Practice Using Behavioral Primitives: Marble Maze, International Jorunal of Robotics Research, pp. 551-560, 2004.
[18] P. Abbeel, and A. Y. Ng. Apprenticeship Learning via Inverse Reinforcement Learning. Proc. ICML. 2004.
[19] P. Englert, A. Paraschos, M. P. Deisenroth, and J. Peters. Probabilistic Model-based Imitation Learning. Adaptive Behavior, vol. 21 (5), pp. 388-403, 2013.
[20] M. D. Erbas, A. F. T. Winfield, and L. Bull, Embodied Imitation-enhanced Reinforcement Learning in Multi-agent Systems, Adaptive Behavior, vol. 22(1), pp. 31-50, Sage publications, 2014.
[21] M. D. Erbas, L. Bull, and A. F. T. Winfield, On the Evolution of Behaviors through Embodied Imitation, Artificial Life, vol. 21, pp. 141-165, 2015.