This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

A Learning-based Autonomy Framework for Human-robot Collaboration

Md Khurram Monir Rabby, Ali Karimoddini, Mubbashar Altaf Khan, and Steven Jiang

In this paper, an adjustable autonomy framework is proposed for Human-robot Collaboration (HRC) in which a robot uses a Reinforcement Learning (RL) mechanism guided by a human operator's rewards in an initially unknown workspace. Within the proposed framework, the autonomy level of the robot is automatically adjusted in an HRC setting that is represented by a Markov Decision Process (MDP) model. When the robot reaches higher performance levels, it can operate more autonomously in the sense that it needs less human operator intervention. A novel $Q-$learning mechanism with an integrated ϵ-greedy approach is implemented for robot learning in order to capture the correct actions and robot's mistakes as a basis for adjusting the robot's autonomy level. The proposed HRC framework can adapt to changes in the workspace as well as changes in human operator reward (scaling and shifting) mechanism, and can always adjust the autonomy level. The autonomy level of the robot is automatically lowered when the workspace changes to allow the robot to explore new actions in order to adapt to the new workspace. In addition, the human operator has the ability to reset/lower the autonomy level of the robot to enforce the robot to re-learn the workspace if its performance is not satisfactory for the human operator. The developed algorithm is applied to a realistic HRC setting involving a humanoid robot, named Baxter. The experimental results are analyzed to assess the effectiveness of the proposed adjustable autonomy framework for different cases: for the case when the workspace does not change, then for the case when the robot autonomy level is reset/lowered by a human operator, and for the case when the workspace is changed by the introduction of new objects. The results confirm the capability of the developed framework to successfully adjust the autonomy level in response to changes in the human operator's commands or the workspace.

***Index Terms*—Human–Robot Collaboration (HRC), Markov Decision Process (MDP), Autonomy Level (AL), Reinforcement Learning (RL).**

I. Introduction

Recent studies on Human-robot Collaboration (HRC) aim at leveraging the interactions of humans and robots from highly constrained laboratories to meaningful collaborations for real-world applications [1]. As the robots are partial actors in an HRC, the degree of their roles and their acceptance to human co-workers depend upon their operation and performance to improve the joint performances in a workspace [2].

The traditional HRC approaches commonly use a pre-programmed robot, which does not necessarily require a robot to have learning capabilities [3]. However, with advances in machine learning and artificial intelligence, it is becoming possible to equip a robot with a learning mechanism and make it a more active and effective collaborator with a human operator in/on the loop [4, 5]. In [6–8], Markov chains, Markov Decision Process (MDP), and Partially Observable Markov Decision Process (POMDP) have been used to develop learning mechanisms for a robot in an HRC setting while capturing the uncertainties involved in a workspace and HRC actors (humans and robots). In [9], a TAMER framework is used that considers human rewards for training a robot in an environment captured by an MDP model. In [10], visual and force sensors have been used to observe and learn human motion for human-robot co-carrying tasks. Other learning based techniques such as imitation learning and supervised learning have been employed for training a robot in an HRC setting [11–13].

Despite the use of robot learning in the aforementioned studies, they do not consider any mechanism for automatic adjustment of robot autonomy based on the robot's capability of handling the shared tasks. To address this problem, a task assignment method is introduced in [14] that considers the task complexity associated with different autonomy levels in a Layered Adjustable Autonomy (LAA) model. A theoretical concept of autonomy adjustment using interaction with the human operator to achieve a common goal in an LAA model along with an Autonomy Analysis Module (AAM) has been used in [15] to control the robot's actions at different autonomy levels. Similarly, a Level of Autonomy (LA) approach is proposed in [16] for remotely controlling mobile robots to manually adjust their autonomy based on the interactions with a human operator. The work in [17] presents an Adjustable Autonomy Intelligent Environment (AAIE) model for developing a robot autonomy adjustment method in a dynamic environment. The concept of variable autonomy levels is implemented in [18] to explore its impact on the task completion period. A sliding scale autonomy is proposed for interactions with a human operator that allows autonomy levels to be changed during the robot operation [19]. The work in [20] presents a situation awareness mechanism for the cyber-physical systems with an integrated meta-model for multiple autonomy levels. In all of these works, the robot does not incorporate any learning capability. Instead, the robot is pre-trained/pre-programmed for different levels of automation and hence, the use of the term "*levels of autonomy*" might not accurately describe these frameworks. Therefore, to the best of our knowledge, this work is the first to propose an autonomy adjustment mechanism based on the change in the performance of a learning robot

M. Rabby, A. Karimoddini, and M. Khan are with the Department of Electrical and Computer Engineering, and S. Jiang is with the Department of Industrial and Systems Engineering, North Carolina Agricultural and Technical State University, Greensboro, NC 27411 USA.

Corresponding author: A. Karimoddini. Address: 1601 East Market Street, Department of Electrical and Computer Engineering North Carolina A&T State University Greensboro, NC, US 27411. Email: akarimod@ncat.edu (Tel: +13362853313).

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

in an HRC setting. In this paper, we refer to autonomy as the capability of a robot to perform a task with reduced human intervention/supervision.

This paper develops an adjustable autonomy framework for an HRC setting by augmenting the $Q-$learning algorithm with an $\epsilon-$greedy mechanism to capture the robot performance, adjust the autonomy level, and balance the exploration of the action space and exploitation of its knowledge base. Compared to supervised learning approaches which commonly use labeled-data to train a model with no interaction with the operator or environment [11], the adoption of $Q-$learning enables the robot to interact with the human operator and the environment to actively acquire and learn the required information and adapt to changes in the workspace. The imitation learning approaches can be a solution for this problem [12, 13]. However, imitation learning approaches often train a classifier to mimic human operator's behavior, i.e., first observe the actions of the human operator during the training phase, followed by learning a policy that mimics the actions demonstrated by the human operator, with limited or no active interaction with the operator, particularly during the training phase. In the proposed framework in this paper, however, the robot uses a greedy strategy to actively explore the action space and collect the required data by getting feedback signals from the human operator. In this way, the robot learns from its mistakes via interactions with the human operator, while adapting to changes in the workspace.

The contributions of this paper include:

- Developing an adjustable autonomy framework using the Reinforcement Learning (RL) mechanism in an HRC framework. The robot learns the correct intended choices of actions based on the received feedback from the human operator. This information is used as a basis for adjusting the robot's autonomy level and improving the robot's learning process. A finite-state Markov Decision Process (MDP) is developed to represent the proposed HRC framework.
- Developing a novel $Q-$learning mechanism and integrating an ϵ-greedy approach to adjust the robot autonomy level. In the proposed framework, in the lowest level of autonomy, the robot uses *exploration* of the action search space to maximally gain information from the human operator; in the intermediate autonomy level, depending on the knowledge about the workspace, the robot uses a mix of *exploration* and *exploitation*, and in the highest autonomy level, the robot primarily uses *exploitation* to take advantage of the experience that is acquired over the training process. The human operator has the authority to reduce the robot's autonomy level to enforce the robot to re-learn the workspace.
- Providing the analytical proof that the reward accumulation (irrespective of scaling and shifting in human operator reward) over the time changes the value of ϵ to improve the robot autonomy level to select the correct action and transition to a higher autonomy level. Conversely, the robot's mistakes are penalized with negative rewards which increase the ϵ value, resulting in lowering the robot's autonomy level.
- Applying the developed framework to a manufacturing case study, which includes different cases of changes in the workspace or human operator's commands for resetting/lowering the robot's autonomy level. To evaluate the proposed framework, experiments have been performed using the developed algorithm in the real-world on a $7-$DoF Baxter robot interacting with a human operator. The results show that with the developed algorithm, the autonomy level of the robot can be automatically adjusted in response to changes in the robot's learning capabilities, and the changes in the workspace and the human operator's commands.

The rest of this paper is presented as follows. Section II describes the proposed modeling of HRC in a shared workspace and formulates the problem of developing an adjustable autonomy framework for HRC. Section III presents the proposed adjustable autonomy framework for an HRC and the developed algorithm using RL. Section IV presents a manufacturing case study and the relevant experimental results for the evaluation of the proposed framework. Finally, the paper is concluded in Section V along with the provision of information about possible future research directions.

II. Proposed Model for Human-robot Collaboration and Problem Formulation

A. Human-robot Collaboration Model

We model an HRC as a finite-state MDP that is capable of capturing both the performances of the human operator(s) and the robot(s) in a shared workspace. We assume that the human operator always makes rational decisions, and correctly rewards the robot's actions. We consider that the state of the HRC system consists of both workspace state, $\mathcal{S}_{\mathcal{W}}$, and robot's state, $\mathcal{S}_{\mathcal{R}}$. The robot is assumed to be equipped with multiple sensors in order to assess the state of HRC (workspace state and robot's state) to select an intended action, a_{R_t}, from the action search space, $\mathcal{A}_R$, using the feedback received from the human operator. The selection of an intended action is based on a quantitative measure of the reward, r_H, which is instantaneously sent by the human operator for the intended action. If the robot's intended action is correct, the human operator provides the maximum reward, guiding the robot to execute that action on the workspace. Otherwise, the human operator minimizes the robot's reward to keep the robot looking for the correct action required to accomplish the desired task. This HRC can be captured by an MDP, $\mathcal{M}_{HRC}$, defined as follows:

$$\mathcal{M}_{HRC} = \langle \mathcal{S}, \mathcal{A}_R, \mathcal{T}, r_H, \gamma \rangle \tag{1}$$

where, $\mathcal{S} = \mathcal{S}_W \times \mathcal{S}_R$ is the state-space of $\mathcal{M}_{HRC}$, where $s_t = (s_{W_t}, s_{R_t}) \in \mathcal{S}$ consists of the workspace state $s_{W_t} \in \mathcal{S}_W$ and robot's state $s_{R_t} \in \mathcal{S}_R$ at a given time t; $\mathcal{A}_R = A_R \times \{0, 1\}$ is the action space, where A_R is the set of all available actions, $a_{R_t} \in A_R \times \{0\}$ and $a'_{R_t} \in A_R \times \{1\}$ refer to the intended and performed actions of the robot at time t, respectively; $\mathcal{T} : \mathcal{S} \times \mathcal{A}_R \times \mathcal{S}' \longrightarrow [0, 1]$ is the transition probability from the current state $s_t = (s_{W_t}, s_{R_t}) \in \mathcal{S}$ to the next state $s_{t+1} = (s_{W_{t+1}}, s_{R_{t+1}}) \in \mathcal{S}'$, given by $\mathcal{T}(s_t =$

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

$(s_{W_t}, s_{R_t}), a_{R_t}, s_{t+1} = (s_{W_{t+1}}, s_{R_{t+1}}))$; $r_H : \mathcal{S} \times \mathcal{A}_R \longrightarrow \mathbb{R}$ is the reward function, which determines the feedback to be provided for the robot's intended actions. The robot will receive $+r$ reward for selecting the correct intended actions and $-r$ for choosing the wrong actions, and $\gamma \in (0, 1]$ is the discounting factor.

B. Multilevel Autonomy

The term "autonomy" in the literature is context-based. The primary standard definition of autonomy from the application point of view is the SAE International's definition for autonomous cars [21], which was later adopted and enhanced by NHTSA [22]. The SAE standard quantifies the levels of autonomy based on the independence from the human operator's intervention (as a car becomes more independent from the human operator, its autonomy level increases and requires less supervision/intervention from the human operator). Apart from the SAE standard, in the HRC-related literature, the higher robot autonomy requires lower levels or less frequent and more sophisticated forms of intervention [23–25]. In fact, when a robot's performance improves, the human operator's trust in the robot increases, and as a result, the human operator can allow more independence to the robot and/or makes less intervention/supervision, which is interpreted as a higher level of autonomy. Accordingly, in the proposed HRC framework, as the robot learns correct action choices via interactions with the human operator, the robot's performance improves. In this situation, the robot requires less guidance from the human operator, and hence, the robot Autonomy Level (AL) increases. Without loss of generality, three levels of autonomy AL_0, AL_1, and AL_2 are considered for the proposed framework that is discussed in Section III. At the lowest autonomy level, AL_0, the robot does not have prior information about the workspace and hence, the robot goes through a trial-and-error procedure, requiring maximum interaction with the human operator for learning the correct choices of action selection. At the highest autonomy level, AL_2, the robot is experienced in selecting the correct actions for the workspace tasks and hence, the robot does not need to go through a trial-and-error procedure that it was using in AL_0. On the other hand, during the intermediate autonomy level, AL_1, the robot has some information about the selection of the correct actions for some situations but this acquired information is not enough to independently choose the correct action for all cases. Therefore, the robot uses both its already acquired knowledge and the trial-and-error procedure to fill the information gap.

Given an HRC framework with a robot being guided by a human operator, our aim is to develop a learning mechanism that can provide the robot with an opportunity to improve its performance and adjust its autonomy level via the guidance received from the human operator in the form of rewards, as formally stated below:

Problem 1. *Consider the HRC framework modeled by $\mathcal{M}_{HRC}$ given in (1). In this HRC framework, the robot chooses an action $a_{R_t} \in \mathcal{A}_R$ to apply to the workspace whose current state is captured as s_{W_t}. Also, consider a human operator who uses the reward function $r_H : \mathcal{S} \times \mathcal{A}_R \longrightarrow \mathbb{R}_H$ to provide a reward to the robot's choices of action based on the state of $\mathcal{M}_{HRC}$ captured by $(s_{W_t}, s_{R_t}) \in \mathcal{S} = \mathcal{S}_W \times \mathcal{S}_R$. Develop a learning mechanism in order to enable the robot to improve its performance and accordingly adjust its autonomy level based on the rewards received from the human operator.*

III. Proposed Adjustable Autonomy Framework

In this section, we develop an adjustable autonomy framework for an HRC. In the HRC model captured by $\mathcal{M}_{HRC}$ given in (1), it is assumed that the robot is equipped with a learning capability. Collaborating with a human operator, the robot learns through a human-reward mechanism about the actions to be performed to accomplish the assigned task(s).

A. Incorporating Reinforcement Learning into the Developed Adjustable Autonomy Framework

In the proposed collaborative framework, shown in Fig. 1, it is assumed that the robot already has the knowledge about the basic actions such as picking and placing an object or moving towards an object. Considering the robot's intended action, a_{R_t}, the current workspace status, s_{W_t}, and the current robot state, s_{R_t}, the human operator provides reward for the intended action as $r_{H_{t+1}} = r_H(s_t = (s_{W_t}, s_{R_t}), a_{R_t})$. The robot learns to choose the actions with the maximum reward at time t using a learning process. The robot's learning process and decision-making process in the proposed framework are divided into three modules, as discussed next.

1) Action selection mechanism

In the proposed HRC setting, the human-provided reward is used by the robot to update its state-action value function and it is defined as a $Q-$function for determining the correct choices of action. Here, the value of $Q-$function can be captured by the Bellman equation [26] as:

$$Q^\star(s_t, a_{R_t}) = r_H(s_t, a_{R_t}) + \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_t, a_{R_t}, s_{t+1}) \max_{a_{R_{t+1}} \in \mathcal{A}_R} Q^\star(s_{t+1}, a_{R_{t+1}}) \tag{2}$$

There are two techniques used by the robot to select an action among the available choices, namely, "*exploration*" and "*exploitation*."

Using the *exploration* technique, the robot takes a policy $\pi(a_R|s)$ to randomly choose an action in its search space as:

$$\pi(a_R|s) = \frac{1}{n}; \text{ for all } a_R \in \mathcal{A}_R \tag{3}$$

where $\pi(a_R|s)$ is the policy of choosing an action a_R at state s, and n is the total number of actions in the robot action

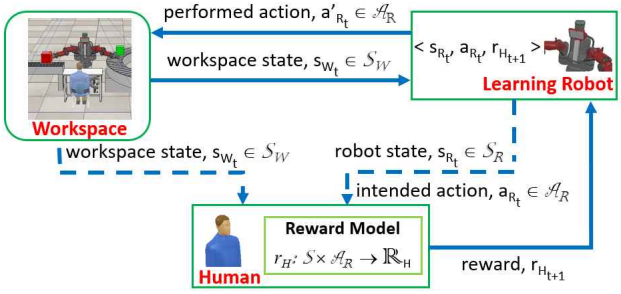


Fig. 1: The proposed learning-based HRC framework.

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

space. *Exploration* is a trial-and-error methodology that the robot uses to guess the correct choice of action(s) with all actions having an equal probability of being chosen.

On the other hand, the *exploitation* technique is used by the robot once it has reasonably learned about its actions through its previous experiences. In this case, the robot can use the learned information to maximize the $Q-$function. Using the *exploitation* technique, the robot chooses an action for receiving the highest reward, which can be achieved by the following policy:

$$\pi(a_R|s) = \begin{cases} 1 & ;\text{if } a^\star = \underset{a_R \in \mathcal{A}_R}{\operatorname{argmax}} Q(s, a_R) \\ 0 & ;\text{otherwise} \end{cases} \tag{4}$$

where $a^\star$ is the optimal choice of action. If there are multiple actions resulting in the maximum $Q-$value, then the robot randomly chooses one of them.

2) Human reward mechanism

In the proposed framework, the execution of the robot's current action is based upon the reward for its intended action, a_{R_t}. Also, the robot uses the human reward for improving its learning process. The reward function is given as:

$$r_H(s_t, a_{R_t}) = \begin{cases} -r & ;\text{if } a_{R_t} \in \mathcal{A}_R \text{ is wrong} \\ 0 & ;\text{if } (a_{R_t} \in \mathcal{A}_R \text{ is correct }) \wedge (s_{t+1} \neq s_{goal}) \\ +r & ;\text{if } (a_{R_t} \in \mathcal{A}_R \text{ is correct }) \wedge (s_{t+1} = s_{goal}) \end{cases} \tag{5}$$

where 0, $+r$, and $-r$ represent zero, positive, and negative rewards, respectively. The robot will be provided with 0 reward for correct actions that do not achieve the goal in the immediate next state; a positive reward, for a correct action that achieves the goal in the immediate next state, and a negative reward for a wrong action. If the task is a single-stage task (it contains only one action), then the robot is given a positive reward for each correct action selection, and the reward mechanism can be reduced to:

$$r_H(s_t, a_{R_t}) = \begin{cases} -r & ;\text{if } a_{R_t} \in \mathcal{A}_R \text{ is wrong} \\ +r & ;\text{if } a_{R_t} \in \mathcal{A}_R \text{ is correct} \end{cases} \tag{6}$$

Here, using the proposed MDP model in (1) and adopting the $Q-$learning method, the reward mechanism can be transformed (scaled and shifted) without changing the optimal policy, as formally stated in the following two lemmas:

Lemma 1. *Scaling human rewards in an MDP-based $Q-$learning does not change the optimal policy.*

Proof. See Appendix A for the proof. ■

Lemma 2. *Shifting human rewards by a constant in an MDP-based $Q-$learning does not change the optimal policy.*

Proof. See Appendix B for the proof. ■

Once the robot selects an action, the robot shares it with the human operator as an intended action for the task. The human operator provides a reward based on the robot's intended action according to the reward mechanism defined in (6). Then, the robot first updates its $Q-$matrix, and then performs the intended action only if the assigned reward is maximum, otherwise the robot continues to search for new action.

3) Autonomy level adjustment mechanism

Through the *exploration* and *exploitation* processes, the robot accumulates the received rewards for the choices of actions over the time. We introduce $0 \leq \epsilon \leq 1$ to capture the rates of the robot's mistakes. The value of ϵ can be greedily decreased when the robot's mistake rate reduces and its performance has improved due to the robot's learning capability. Initially, it is assumed that the robot starts its operation at the first autonomy level, AL_0, with the highest value of ϵ. Due to the high robot's mistake rate at this level, the robot only chooses *exploration* to explore the actions from its action search space. This provides the robot with a chance to learn correct choices of actions and reduce its mistake rate, which in turn reduces ϵ. Once ϵ reaches a certain threshold, it switches to AL_1.

In AL_1, even though the robot's performance has improved, it is not perfect yet. Therefore, the robot uses a combination of *exploration* and *exploitation*. We can use ϵ to assess the amount of training that the robot needs through the *exploration* process. For this purpose, the robot chooses *exploration* with the probability of ϵ and *exploitation* with the probability of $1 - \epsilon$. Therefore, combining (3) and (4), the action selection policy becomes:

$$\pi(a_R|s) = \begin{cases} \frac{\epsilon}{n} + 1 - \epsilon & ;\text{if } a^\star = \underset{a_R \in \mathcal{A}_R}{\operatorname{argmax}} Q(s, a_R) \\ \frac{\epsilon}{n} & ;\text{otherwise} \end{cases} \tag{7}$$

During this process, by reducing ϵ, the robot gradually reduces the use of *exploration* and increases the use of *exploitation* of the search space until an eventual transition to the next autonomy level, i.e., AL_2.

In AL_2, the robot reaches a high level of autonomy with a small rate of mistakes. Therefore, the robot uses only the *exploitation* and greedily updates ϵ by maximizing the received rewards as explained in (4). The next lemma and corollary show that employing the proposed reward mechanism, *exploitation* will lead to a more informed decision.

Lemma 3. *If the workspace does not change, with the reward mechanism in (7), the exploitation will always lead to a more informed decision, i.e., $Pr(a_R(t) = \text{ correct action})$ is larger under the exploitation as compared to the exploration.*

Proof. See Appendix C for the proof. ■

Corollary 1. *In an HRC setting with an $\epsilon-$Greedy policy for the zero initialization of the $Q-$matrix, choosing an action $a_{R_1} = \underset{a_R \in \mathcal{A}_R}{\operatorname{argmax}} Q_{\pi_1}(s, a_R)$ selected by policy π_1 will lead to a larger reward from the human operator than an action $a_{R_2} = \underset{a_R \in \mathcal{A}_R}{\operatorname{argmax}} Q_{\pi_2}(s, a_R)$ following a policy π_2, i.e., $r_H(s, a_{R_1}) \geq r_H(s, a_{R_2})$, if and only if $Q_{\pi_1}(s, a_{R_1}) \geq Q_{\pi_2}(s, a_{R_2})$.*

Proof. See Appendix D for the proof. ■

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

B. Proposed Algorithm for an Adjustable Autonomy

Initially, we set the autonomy level to AL_0 and $\epsilon = 1$. The human operator rewards the robot based on (6). Accordingly, the value of ϵ will be changed as:

$$\epsilon(t) = \epsilon(t-1)(1 - \kappa(r_H(s_t, a_{R_t}))) \tag{8}$$

with

$$\kappa(r_H(s_t, a_{R_t})) = \begin{cases} \kappa_+ & ;\text{if } r_H(s_t, a_{R_t}) > 0 \\ -\kappa_- & ;\text{if } r_H(s_t, a_{R_t}) < 0 \end{cases} \tag{9}$$

where a_{R_t} is an intended action at time t, and $\kappa_+, \kappa_- > 0$. As it is shown in Theorem 1, if condition (10) holds, the autonomy level of the robot is eventually elevated to AL_1 as the expected value of ϵ decreases below the threshold TL_0^-.

Theorem 1. *If the robot's autonomy level is at AL_0, with the reward mechanism in (6), the robot's autonomy level always eventually transitions from AL_0 to AL_1, i.e., $\mathbb{E}(\epsilon(t)) < TL_0^-$ at some $t \geq 0$ if and only if*

$$\kappa_+ > (n-1)\kappa_- \tag{10}$$

Proof. In the autonomy level AL_0 of the proposed human-reward-based $\epsilon-$Greedy Algorithm, the value of ϵ changes as described in (8). Since, ϵ and a_{R_t} are independent, based on (8), we have:

$$\mathbb{E}[\epsilon(t)] = \mathbb{E}[\epsilon(t-1)](1 - \mathbb{E}[\kappa(r_H(s_t, a_{R_t}))]) \tag{11}$$

According to the policy for *exploration*, governed by (3), the actions a_R in the search space are selected randomly as $\pi(a_R(t)|s) = \frac{1}{n}$. Since, there is only one correct action corresponding to each state, $Pr(a_{R_t}(t) = \text{the correct action}) = \frac{1}{n}$ and $Pr(a_{R_t}(t) = \text{a wrong action}) = \frac{n-1}{n}$. Therefore, $\mathbb{E}[\kappa(r_H(s_t, a_{R_t}))]$ will be:

$$\mathbb{E}[\kappa(r_H(s_t, a_{R_t}))] = \frac{1}{n}\kappa_+ - \frac{n-1}{n}\kappa_-. \tag{12}$$

Substituting (12) into (11), it can be revisited as:

$$\mathbb{E}[\epsilon(t)] = \mathbb{E}[\epsilon(t-1)](1 - \frac{\kappa_+ - (n-1)\kappa_-}{n}) \tag{13}$$

Clearly, $\mathbb{E}[\epsilon(t)]$ will be decreasing if and only if $\kappa_+ > (n-1)\kappa_-$. ■

In AL_1, the robot performs both *exploration* to search for the unknown states to handle different situations and the *exploitation* to infer the correct action based on $Q-$matrix using the policy provided in (7). In this situation, as it is shown in Theorem 2, the value of ϵ gradually decreases, and hence, we will eventually have more *exploitation* than *exploration*.

Theorem 2. *If the robot's autonomy level is at AL_1 and if the workspace does not change, with the reward mechanism in (6), the autonomy level of the robot always eventually transits from AL_1 to AL_2, i.e., $\mathbb{E}(\epsilon(t)) < TL_1^-$ at some $t \geq 0$ if*

$$\kappa_+ > (n-1)\kappa_- \tag{14}$$

Proof. Assume that the autonomy level of the robot is in AL_1. As it has been shown in the proof of Theorem 1, the expected value of ϵ changes according to (11). Applying the action selection policy provided in (7) for AL_1, $Pr(a_{R_t}(t) = \text{the correct action using } exploitation) = \frac{\epsilon}{n} + 1 - \epsilon$ and $Pr(a_{R_t}(t) = \text{a wrong action }) = \frac{n-1}{n}\epsilon$. Therefore, $\mathbb{E}[\kappa(r_H(s_t, a_{R_t}))]$ can be calculated as:

$$\mathbb{E}[\kappa(r_H(s_t, a_{R_t}))] = \kappa_+(\frac{\epsilon}{n} + 1 - \epsilon) - \kappa_- \frac{n-1}{n}\epsilon. \tag{15}$$

Substituting (15) into (11), it can be revisited as:

$$\begin{aligned} \mathbb{E}[\epsilon(t)] =& \mathbb{E}[\epsilon(t-1)] \\ & (1 - \frac{(\kappa_+ - (n-1)\kappa_-)\epsilon + \kappa_+(1-\epsilon)n}{n}) \end{aligned} \tag{16}$$

Since $0 \leq \epsilon \leq 1$, $\mathbb{E}[\epsilon(t)]$ decreases for $\kappa_+ \geq (n-1)\kappa_-$. Under this condition, the value of $\epsilon(t)$ eventually decreases below TL_1^- and the autonomy level switches to AL_2. Note that when the system is in AL_1, even though $\mathbb{E}[\epsilon(t)]$ will be decreasing, still there is a chance that the value of $\epsilon(t)$ increases to take it above the threshold TL_0^+ and accordingly, the autonomy level switches to AL_0. However, as it is proved in Theorem 1, again in AL_0 the value of $\mathbb{E}[\epsilon(t)]$ will decrease and eventually the value of $\epsilon(t)$ will fall below TL_0^- and the autonomy level will switch back to AL_1. On the other hand, as shown above, in AL_1, $\mathbb{E}[\epsilon(t)]$ is decreasing, and hence, the value of $\epsilon(t)$ eventually decreases below TL_1^- and the autonomy level switches to AL_2. ■

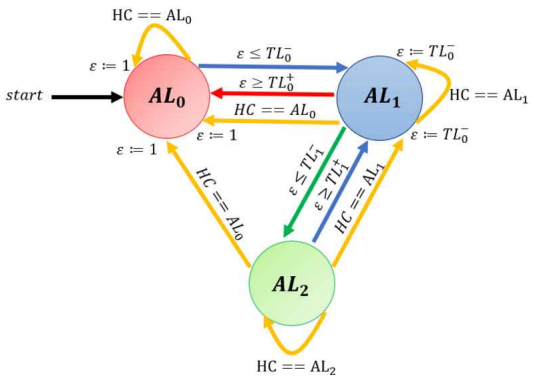


Fig. 2: Change of autonomy level in the proposed HRC setting, from the low level AL_0 to AL_1 and then AL_2, and vice versa, based on the human operator commands or value of ϵ, where $0 < TL_1^- < TL_1^+ < TL_0^- < TL_0^+ < 1$.

The state-diagram for the proposed adjustable autonomy framework is shown in Fig. 2, and is detailed in Algorithm 1 and Algorithm 2. Algorithm 1 implements the proposed $Q-$learning mechanism to learn from experiences in the form of *exploration* or *exploitation* to update the $Q-$matrix and Algorithm 2 adjusts the robot's autonomy level based on either the received human command (HC) or the updated $\epsilon-$value.

Algorithm 1 is initialized with zero $Q-$value function, implying that the robot does not have any prior information about the workspace (Line 1). In autonomy level AL_0, AL_1, and AL_2, the robot chooses an intended action a_R following

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

Algorithm 1: Human-reward-based $\epsilon-$Greedy Algorithm integrated with $Q-$Learning

1 **Initialization:** $\forall s \in \mathcal{S}$, $\forall a_R \in \mathcal{A}_R$: $Q(s, a_R) = 0$, $AL = AL_0$, $\epsilon = 1$;
2 **while** *(1)* **do**
3 **repeat**
4 **if** $AL = AL_0$ **then**
5 **Choose** a_{R_t} using the policy given in (3);
6 **else if** $AL = AL_1$ **then**
7 **Choose** a_{R_t} using the policy given in (7);
8 **else**
9 **Choose** a_{R_t} using the policy given in (4);
10 **end**
11 **Receive** reward r_H from the human operator;
12 **if** $r_H < 0$ **then**
13 $\epsilon = min(1, \epsilon \times (1 + \kappa_-))$;
14 **else**
15 $\epsilon = max(0, \epsilon \times (1 - \kappa_+))$;
16 performed action $\longleftarrow a_R$;
17 **update** system state $s_t \longleftarrow s_{t+1}$;
18 **end**
19 **Update** $Q(s_t, a_{R_t})$ using Bellman eq. [26];
20 **Receive** human operator command HC if any;
21 $AL =$ *AdjustAutonomy*(AL, ϵ, HC);
22 **until** $s \neq s_{goal}$;
23 **end**

Algorithm 2: Autonomy Adjustment

1 **Function** `AdjustAutonomy`$(AL,\ \epsilon,\ HC)$**:**
2 **if** $HC = AL_0$ **then**
3 $AL = AL_0$, $\epsilon = 1$;
4 **else if** $HC = AL_1$ *&* $(AL = AL_2$ *or* $AL = AL_1)$ **then**
5 $AL = AL_1$, $\epsilon = TL_0^-$;
6 **else if** $HC = AL_2$ *&* $AL = AL_2$ **then**
7 $AL = AL_2$;
8 **else**
9 **if** $AL = AL_1$ *&* $\epsilon \geq TL_0^+$ **then**
10 $AL = AL_0$;
11 **else if** $AL = AL_0$ *&* $\epsilon \leq TL_0^-$ **then**
12 $AL = AL_1$;
13 **else if** $AL = AL_2$ *&* $\epsilon \geq TL_1^+$ **then**
14 $AL = AL_1$;
15 **else if** $AL = AL_1$ *&* $\epsilon \leq TL_1^-$ **then**
16 $AL = AL_2$;
17 **end**
18 **end**
19 **return** AL;
20 **End Function**

the policies given by (3), (7), and (4), respectively (Lines 4-10). Then, the robot receives a reward from the human operator (Line 11). Based on the received reward value, the robot updates ϵ and decides whether or not to perform the intended action. If the reward value is negative, the value of ϵ will be increased and the robot will not proceed with the intended action (Lines 12-13). Otherwise, the robot will perform the action, and the value of ϵ will be decreased (Lines 14-18). Next, the robot updates the $Q-$matrix (Line 19) for each choice of action selection, followed by an adjustment in the autonomy level (Lines 20-21).

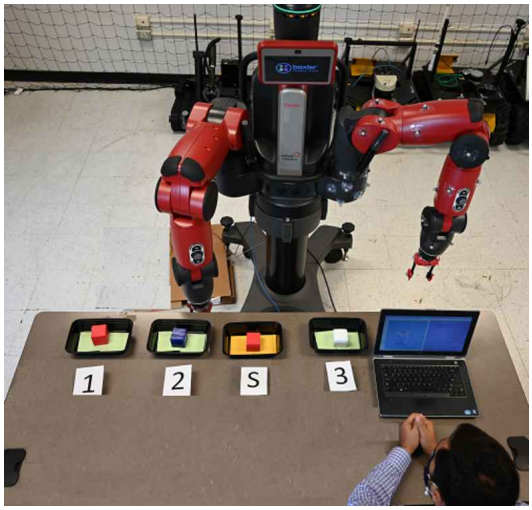


Fig. 3: The experimental setup.

Algorithm 2 is used to adjust the robot's autonomy level based on the received human operator command, HC, the updated value of ϵ, and the current autonomy level of the robot. The human operator can override the level of autonomy and can decrease the level of autonomy, but cannot increase the level of autonomy without allowing the robot an opportunity to learn and gain the required experiences. Therefore, by using human commands $HC = AL_0$, $HC = AL_1$, and $HC = AL_2$, the autonomy level can be degraded to or reinstated in AL_0, AL_1, and AL_2, respectively (Lines 2-7). Otherwise, based on the value of ϵ with respect to the defined threshold values, the robot can transition to appropriate autonomy levels (Lines 9-17). Hysteresis thresholding is applied to avoid Zeno phenomena.

IV. Experimental Results

In this section, the proposed adjustable autonomy framework is implemented on a robot, which collaborates with a human operator to inspect the incoming objects in order to sort and place them into appropriate destination containers as shown in Fig. 3.

A. Description of Experimental Case Study Set-up

The experimental setup is shown in Fig. 3. A Baxter robot [27] is used for this experiment. Baxter is a humanoid robot with $7-$DoF arms equipped with grippers for picking objects. We have made the robot capable of performing basic tasks such as picking an object and placing an object as shown in Fig. 3. In this experiment, the robot is expected to handle three types of objects with Red, Blue, and White colors, and the incoming objects need to be picked up from the Source Container and routed to Container-1, Container-2, or Container-3, depending on the scenario requirements. Accordingly, the workspace state is defined as $\mathcal{S}_W = \{b, r, w, empty\}$ where b, r, and w respectively represent the appearance of an incoming object of type blue, red, or white in the Source container, and empty represents the case when the there is no object in the

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

workspace. On the other hand, the robot state is defined as $\mathcal{S}_R = \{idle, busy\}$ depending on whether the robot is taking an action or not. Also, the set of available actions in this setup is $A_R = \{TC_1, TC_2, TC_3\}$, standing for transferring incoming objects to Containers 1, 2, or 3, respectively. In a manufacturing setting, these containers can represent the packaging stations/conveyors. The human operator inspects the objects and cognitively helps the robot by rewarding its correct intended choice of action selection through a ROS-operated workstation. The human operator chooses the maximum reward value for a correct intended action and the minimum reward value for an incorrect intended action. Moreover, the human operator physically adjusts the orientation of the object so that the robot can easily pick the object, whenever needed.

TABLE I: Experimental Setup Parameters

AL	**Parameter**	**Value**
All	α	0.8
All	γ	0.8
All	r	1
$-$	TL_0^+	0.85
$-$	TL_0^-	0.75
$-$	TL_1^+	0.45
$-$	TL_1^-	0.35
AL_0, AL_1	κ_+	0.2
AL_0, AL_1	κ_-	0.02
AL_2	κ_+	0.1
AL_2	κ_-	0.2

B. Analysis of the Experimental Results

In this experiment, we use the developed HRC framework to handle objects of three different types/colors with random arrival at the Source Container. The parameters involved in the proposed adjustable autonomy framework are given in Table I.

In our experiment, four successive cases have been considered to demonstrate the adjustment of the robot's autonomy level. The details of these four cases are as follows:

In $Case-1$, the robot initially starts at AL_0 and learns the workspace through interactions with the human operator to discover the correct actions for handling the incoming objects, as shown in episodes $1-100$ in Fig. 5 and time interval $[1, 259]$ in Figs. 6a and 7a, respectively. As the robot learns about the workspace, the total accumulated reward(s) per episode is improved (Fig. 5a) and the value of ϵ is gradually decreased over the time (Fig. 6a). As a result, the total *exploitation* count per episode is increased (Fig. 5c), and the total *exploration* count per episode is reduced (Fig. 5b). On the other hand, as ϵ decreases, the robot's autonomy level transitions to AL_1 when ϵ goes below $TL_0^- = 0.75$, and eventually it reaches AL_2 when ϵ becomes lower than $TL_1^- = 0.35$ as shown in Fig. 7a.

In $Case-2$, the human operator issues the command $HC == AL_0$ to reset the autonomy level to AL_0 at episode 101 when $t = 260min$. As shown in Fig. 5 over the episodes $101-193$ and over the time interval $[260, 465]$ in Figs. 6b and 7b, the robot starts learning about the workspace again, accumulates rewards, decreases ϵ, and adjusts its autonomy level accordingly. In this case, the total accumulated reward(s) per episode suddenly decreases but later increases (Fig. 5a).

Similarly, in $Case-3$, the human operator reduces the robot's autonomy level from AL_2 to AL_1 by issuing the command $HC == AL_1$ at episode 194 ($t = 466min$). As shown in Fig. 5 over episodes $194-285$ and over the time interval $[466, 660]$ in Figs. 6c and 7c, the total accumulated rewards per episode decrease insignificantly (Fig. 5a). As the robot already has prior information about the workspace, the robot quickly transitions from AL_1 to AL_2 as compared to $Case-1$ and $Case-2$.

In $Case-4$, while the robot continues to operate at the highest autonomy level, i.e. AL_2, a new object is introduced into the workspace at the episode 286 (at time instant $t = 661$min). In $Case-4$, as shown in Fig. 5 over episodes $286-373$ and over the time interval $[661, 930]$ in Figs. 6d and 7d, the robot does not have information about the new object in the workspace but since it is operating in AL_2, it initially follows the *exploitation* and consecutively makes wrong choices of actions, and hence, it consistently receives minimum rewards that increase the value of ϵ, reducing the level of autonomy to AL_1 when ϵ increases above the $TL_1^+ = 0.45$. Then, in AL_1, the robot starts learning about the workspace again by conducting a mix of *exploration* and *exploitation*, while accumulating rewards over the time that decreases ϵ. When the value of ϵ becomes lower than the threshold $TL_1^- = 0.35$, the AL of the robot is adjusted back to AL_2.

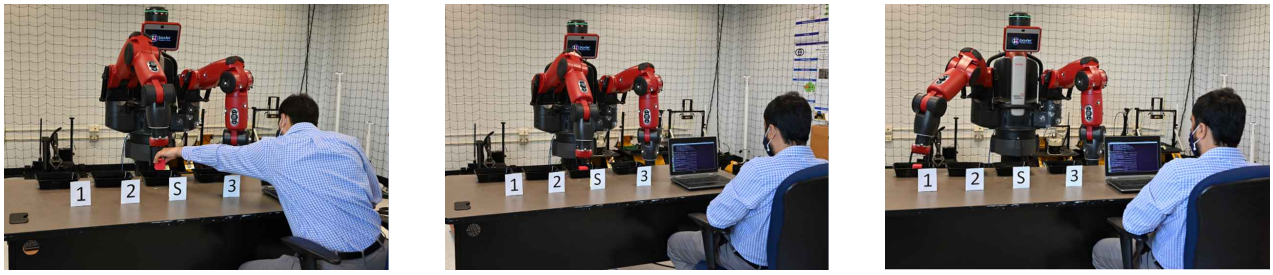
A time-lapse video of this experiment is available at: https://youtu.be/Sycrr_MqV_c.

V. Conclusions and Future Work

This paper developed an adjustable autonomy framework for an HRC setting to enable a robot to learn correct actions in an initially unknown workspace. An MDP model was developed and incorporated into the proposed framework to mathematically represent the collaboration setting between the human operator and the robot in a given workspace. A $Q-$learning mechanism with an integrated ϵ-greedy approach was developed for enabling the robot to learn the workspace and make correct intended choices of actions for adjusting the robot's autonomy level. The developed algorithm was applied to an HRC setting in a manufacturing process. In this process, a $7-$DoF Baxter robot collaborated with a human operator to inspect and sort the incoming objects. The experimental results showed the effectiveness of the proposed adjustable autonomy algorithm for adapting to different cases involving either changes in the workspace or human operator's commands. As future work, we will go beyond the laboratory setting experiments and will explore the application of the proposed framework to an HRC system in a manufacturing setting with more complex collaboration scenarios. We will also explore the impact of the human operator's performance and behavior change when the human operator does not consistently provide correct/rational reward values for the robot's actions. We will also extend the proposed framework to uniformly address both the high-level decision-making (action selection) and low-level control (motion planning and action execution).

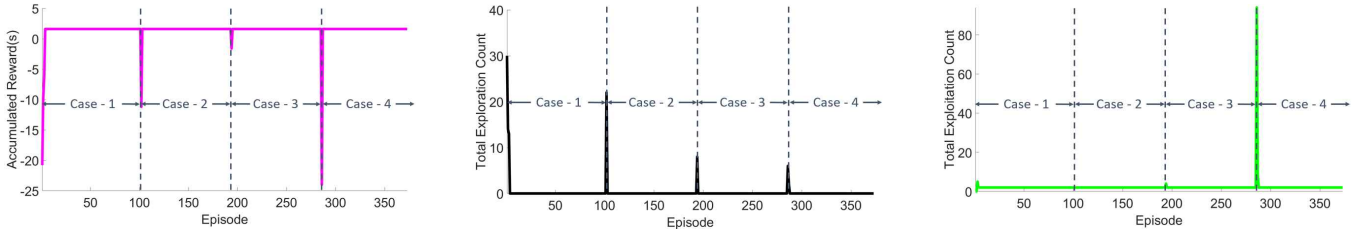
1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics



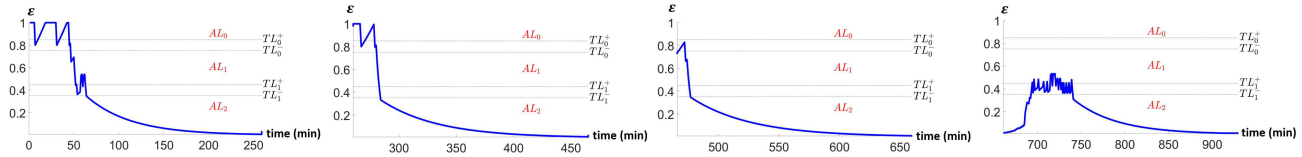
(a) (b) (c)

Fig. 4: Baxter robot collaborates with a human operator to inspect and sort objects using the proposed algorithm: (a) A human operator inspects an object and places it in the source container, (b) The Baxter robot picks the object from the Source Container, (c) The Baxter robot places the object in one of the containers.



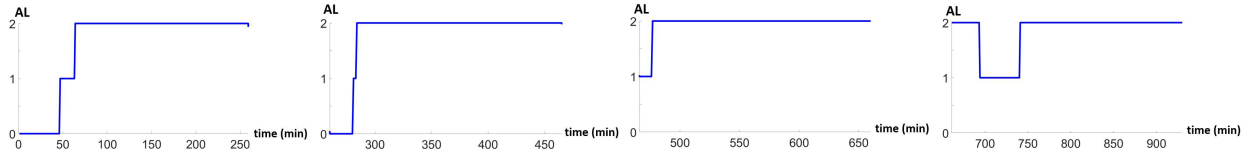
(a) Change in accumulated reward(s) (b) Change in total number of *exploration* (c) Change in total number of *exploitation*

Fig. 5: Change in reward(s) and the total number of *exploration* and *exploitation* for four different experimental cases.



(a) $Case-1$ (b) $Case-2$ (c) $Case-3$ (d) $Case-4$

Fig. 6: Change of ϵ over the time (min) for four different experimental cases.



(a) $Case-1$ (b) $Case-2$ (c) $Case-3$ (d) $Case-4$

Fig. 7: Change of AL over the time (min) for four different experimental cases.

APPENDIX A
PROOF FOR LEMMA 1

In an MDP-based $Q-$learning, assume that the optimal policy can be described as $\pi^{\star}(a_{R_{t+1}}|s_{t+1}) = \underset{a_{R_{t+1}} \in A_R}{arg\,max}\, Q^{\star}(s_{t+1}, a_{R_{t+1}})$, in which the optimal $Q-$value can be captured by the Bellman equation described in (2).

Suppose that there exists another reward mechanism as $r'_H(s_t, a_{R_t}) = cr_H(s_t, a_{R_t})$, where c is a positive scaling factor, resulting in a new $Q-$value function $Q^{\star\prime}$ as:

$$Q^{\star\prime}(s_t, a_{R_t}) = cr_H(s_t, a_{R_t}) + \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_t, a_{R_t}, s_{t+1}) \max_{a_{R_{t+1}} \in \mathcal{A}_R} Q^{\star\prime}(s_{t+1}, a_{R_{t+1}}) \tag{17}$$

Dividing both sides of (17) by c, we will have

$$\frac{1}{c}Q^{\star\prime}(s_t, a_{R_t}) = r_H(s_t, a_{R_t}) + \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_t, a_{R_t}, s_{t+1}) \max_{a_{R_{t+1}} \in \mathcal{A}_R} \frac{1}{c}Q^{\star\prime}(s_{t+1}, a_{R_{t+1}}) \tag{18}$$

From (18) we can conclude that $Q^{\star} = \frac{Q^{\star\prime}}{c}$. Therefore, as shown in (19), the new optimal policy is the same as the

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

previous optimal policy:

$$\begin{aligned}\pi^{\star\prime}(a_{R_{t+1}}|s_{t+1}) &= \underset{a_{R_{t+1}}\in A_R}{arg\,max}\, Q^{\star\prime}(s_{t+1}, a_{R_{t+1}}) \\ &= \underset{a_{R_{t+1}}\in A_R}{arg\,max}\; cQ^{\star}(s_{t+1}, a_{R_{t+1}}) \\ &= \underset{a_{R_{t+1}}\in A_R}{arg\,max}\; Q^{\star}(s_{t+1}, a_{R_{t+1}}) \\ &= \pi^{\star}(a_{R_{t+1}}|s_{t+1})\end{aligned} \tag{19}$$

APPENDIX B
PROOF FOR LEMMA 2

Similar to Lemma 1, assume that the optimal policy of the MDP-based $Q-$learning can be described as $\pi^{\star}(a_{R_{t+1}}|s_{t+1}) = \underset{a_{R_{t+1}}\in A_R}{arg\,max}\, Q^{\star}(s_{t+1}, a_{R_{t+1}})$, in which the optimal $Q-$value can be captured by the Bellman equation given in (2). Suppose that there exists another reward mechanism as $r'_H(s_t, a_{R_t}) = r_H(s_t, a_{R_t}) + c$, where c is a shifting value, resulting in a new $Q-$value function as:

$$\begin{aligned}Q^{\star\prime}(s_t, a_{R_t}) = r_H(s_t, a_{R_t}) + c + \\ \gamma \sum_{s_{t+1}\in\mathcal{S}} \mathcal{T}(s_t, a_{R_t}, s_{t+1}) \max_{a_{R_{t+1}}\in\mathcal{A}_R} Q^{\star\prime}(s_{t+1}, a_{R_{t+1}})\end{aligned} \tag{20}$$

Similar to the proof of Lemma 1, it can be shown that $Q^{\star} = Q^{\star\prime} - c$, concluding that $\pi^{\star\prime}(a_{R_{t+1}}|s_{t+1}) = \pi^{\star}(a_{R_{t+1}}|s_{t+1})$.

APPENDIX C
PROOF FOR LEMMA 3

Assume that in a $Q-$learning mechanism, $Q(s_t, a_{R_t})$ is an element of $Q-$matrix at time instant t corresponding to the state s for an intended action a_R. Now, if the robot uses the *exploitation* approach, according to (4), the robot chooses the intended action with maximum $Q-$values, i.e., $a_{R_t} = \underset{a_R\in\mathcal{A}_R}{\text{argmax}}\, Q(s_t, a_R)$, which is equivalent to the one that has received the maximum reward in the past experiences. With the rational reward mechanism described in (6), if there is only one action with maximum $Q-$value, it is the correct action, i.e., $Pr(a_R(t) = \text{ correct action}) = 1$. Even though it is less likely but it may happen, if there are multiple choices of actions with the maximum but equal $Q-$values, then the robot randomly chooses one of them. In the worst case, if all actions have the same $Q-$value, the *exploitation* would become equivalent to *exploration* (a completely random search). Therefore, $Pr(a_R(t) = \text{ correct action})$ is larger under the *exploitation* as compared to the *exploration*.

APPENDIX D
PROOF FOR COROLLARY 1

The proof is by induction. Initially, at $t = 0$, the matrix Q is set to be zero for both policies π_1 and π_2. Therefore, using Q-learning update formula [26], the value for policy π_k leading to an action a_{R_k} and receiving the reward $r_H(s_0, a_{R_k})$, $k = 1, 2$, will be updated as:

$$Q(s_0, a_{R_k}) = \alpha r_H(s_0, a_{R_k}) \tag{21}$$

Hence, at $t = 0$, if the robot chooses an action a_{R_1} following a policy π_1, the value of $Q-$matrix will be updated as $Q_{\pi_1}(s_0, a_{R_1}) = \alpha r_H(s_0, a_{R_1})$, which is greater than or equal to $Q_{\pi_2}(s_0, a_{R_2}) = \alpha r_H(s_0, a_{R_2})$ for an action a_{R_2} selected by policy π_2, if and only if $r_H(s, a_{R_1}) \geq r_H(s, a_{R_2})$. Now, assume that $Q_{\pi_1}(s_j, a_{R_1}) \geq Q_{\pi_2}(s_j, a_{R_2})$ at $t = j$. Following Q-learning formula [26], the updated $Q-$matrix for policy π_k leading to an action a_{R_k} and receiving the reward $r_H(s_j, a_{R_k})$ will be:

$$\begin{aligned}Q_{\pi_k}(s_j, a_{R_k}) =& (1-\alpha) Q_{\pi_k}(s_j, a_{R_k}) + \\ & \alpha\{r_H(s_j, a_{R_k}) + \gamma \max_{a_{R_k}} Q_{\pi_k}(s_{j+1}, a_{R_k})\}\end{aligned} \tag{22}$$

From (22), it can be seen that $Q_{\pi_1}(s_j, a_{R_1}) = (1-\alpha)Q_{\pi_1}(s_j, a_{R_1}) + \alpha\{r_H(s_j, a_{R_1}) + \gamma \max_{a_{R_1}} Q_{\pi_1}(s_{j+1}, a_{R_1})\} \geq Q_{\pi_2}(s_j, a_{R_2}) = (1-\alpha)Q_{\pi_2}(s_j, a_{R_2}) + \alpha\{r_H(s_j, a_{R_2}) + \gamma \max_{a_{R_2}} Q_{\pi_2}(s_{j+1}, a_{R_2})\}$, if and only if $r_H(s_j, a_{R_1}) \geq r_H(s_j, a_{R_2})$.

ACKNOWLEDGMENT

This work is supported by National Science Foundation under the award number 1832110 and the Air Force Research Laboratory and Office of the Secretary of Defense under agreement number FA8750-15-2-0116.

REFERENCES

[1] C. Furlough, T. Stokes, and D. J. Gillan, "Attributing Blame to Robots: I. the Influence of Robot Autonomy," *Human Factors*, p. 0018720819880641, 2019.

[2] G. Du, B. Zhang, C. Li, B. Gao, and P. X. Liu, "Natural Human-machine Interface with Gesture Tracking and Cartesian Platform for Contactless Electromagnetic Force Feedback," *IEEE Transactions on Industrial Informatics*, 2020.

[3] L. Han, W. Xu, P. Kang, and H. Yuan, "Unified Neural Adaptive Control for Multiple Human-robot-environment Interactions," *IEEE Transactions on Industrial Informatics*, 2020.

[4] S. Doltsinis, P. Ferreira, and N. Lohse, "A Symbiotic Human–machine Learning Approach for Production Ramp-up," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 3, pp. 229–240, 2017.

[5] Y. Zhang, A. Michi, J. Wagner, E. André, B. Schuller, and F. Weninger, "A Generic Human–machine Annotation Framework Based on Dynamic Cooperative Learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 3, pp. 1230–1239, 2019.

[6] X. Zhang and H. Lin, "Performance Guaranteed Human-robot Collaboration with POMDP Supervisory Control," *Robotics and Computer-Integrated Manufacturing*, vol. 57, pp. 59–72, 2019.

[7] W. Zheng, B. Wu, and H. Lin, "POMDP Model Learning for Human Robot Collaboration," in *2018 IEEE Confer-*

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

ence on Decision and Control (CDC), pp. 1156–1161, IEEE, 2018.

[8] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of Human Activity Patterns for Human–robot Collaborative Assembly Tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 3934–3942, 2018.

[9] W. B. Knox, P. Stone, and C. Breazeal, "Teaching Agents with Human Feedback: A Demonstration of the TAMER Framework," in *Proceedings of the Companion Publication of the 2013 International Conference on Intelligent User Interfaces Companion*, pp. 65–66, ACM, 2013.

[10] X. Yu, W. He, Q. Li, Y. Li, and B. Li, "Human-robot Co-carrying Using Visual and Force Sensing," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 9, pp. 8657–8666, 2020.

[11] F. Zhang and Y. Demiris, "Learning Grasping Points for Garment Manipulation in Robot-assisted Dressing," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9114–9120, IEEE, 2020.

[12] H. B. Amor, D. Vogt, M. Ewerton, E. Berger, B. Jung, and J. Peters, "Learning Responsive Robot Behavior by Imitation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3257–3264, IEEE, 2013.

[13] D.-W. Huang, G. Katz, J. Langsfeld, R. Gentili, and J. Reggia, "A Virtual Demonstrator Environment for Robot Imitation Learning," in *2015 IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, pp. 1–6, IEEE, 2015.

[14] S. A. Mostafa, S. S. Gunasekaran, M. S. Ahmad, A. Ahmad, M. Annamalai, and A. Mustapha, "Defining Tasks and Actions Complexity-levels via Their Deliberation Intensity Measures in the Layered Adjustable Autonomy Model," in *2014 International Conference on Intelligent Environments*, pp. 52–55, IEEE, 2014.

[15] S. A. Mostafa, M. S. Ahmad, M. Annamalai, A. Ahmad, and S. S. Gunasekaran, "A Conceptual Model of Layered Adjustable Autonomy," in *Advances in Information Systems and Technologies*, pp. 619–630, Springer, 2013.

[16] Á. Martínez-Tenor and J.-A. Fernández-Madrigal, "Smoothly Adjustable Autonomy for the Low-level Remote Control of Mobile Robots that is Independent of the Navigation Algorithm," in *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pp. 1071–1078, IEEE, 2015.

[17] M. Ball and V. Callaghan, "Explorations of Autonomy: An Investigation of Adjustable Autonomy in Intelligent Environments," in *2012 Eighth International Conference on Intelligent Environments*, pp. 114–121, IEEE, 2012.

[18] C. Robinson, I. B. Wijayasinghe, and D. O. Popa, "Quantitative Variable Autonomy Levels for Traded Control in a Pick-and-place Task," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, pp. 697–702, IEEE, 2019.

[19] M. Desai and H. A. Yanco, "Blending Human and Robot Inputs for Sliding Scale Autonomy," in *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pp. 537–542, IEEE, 2005.

[20] M. Gharib, L. D. Da Silva, H. Kavalionak, and A. Ceccarelli, "A Model-based Approach for Analyzing the Autonomy Levels for Cyber-Physical Systems-of-systems," in *2018 Eighth Latin-American Symposium on Dependable Computing (LADC)*, pp. 135–144, IEEE, 2018.

[21] SAE, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-road Motor Vehicles," tech. rep., 2016.

[22] NHTSA, "Federal Automated Vehicles Policy: Accelerating the Next Revolution in Roadway Safety," tech. rep., 2016.

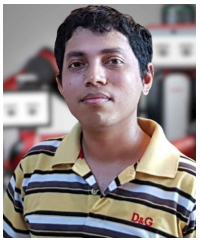
[23] J. M. Beer, A. D. Fisk, and W. A. Rogers, "Toward a Framework for Levels of Robot Autonomy in Human-robot Interaction," *Journal of Human-robot Interaction*, vol. 3, no. 2, p. 74, 2014.

[24] H.-M. Huang, K. Pavek, B. Novak, J. Albus, and E. Messin, "A Framework for Autonomy Levels for Unmanned Systems (ALFUS)," *Proceedings of the AUVSI's Unmanned Systems North America*, pp. 849–863, 2005.

[25] H.-M. Huang, E. Messina, R. Wade, R. English, B. Novak, and J. Albus, "Autonomy Measures for Robots," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 47063, pp. 1241–1247, 2004.

[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[27] X. Yu, W. He, H. Li, and J. Sun, "Adaptive Fuzzy Full-state and Output-feedback Control for Uncertain Robots with Output Constraint," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2020.



Md Khurram Monir Rabby is a Ph.D. student in Autonomous Cooperative Control of Emergent Systems of Systems (ACCESS) Lab in the Department of Electrical and Computer Engineering at North Carolina A&T State University, USA. He is also a member of Testing, Evaluation & Control of Heterogeneous Large-scale Systems of Autonomous Vehicle (TECHLAV) Center. He received his B.Sc. in Electrical & Electronic Engineering (EEE) and later completed his M.Sc. in Information & Communication Technology (ICT) from Bangladesh University of Engineering and Technology (BUET), Bangladesh. His current research interests include but not limited to Human-robot Collaboration (HRC), Automation & Control Systems, Formal Methods, Discrete Event Systems (DES), Multi-agent Systems, Machine Learning, and Robotics.

Ali Karimoddini is the Director of the NC Transportation Center of Excellence on Connected and Autonomous Vehicle Technology (NC-CAV) and the Autonomous Cooperative Control of Emergent System of Systems (ACCESS) laboratory at NC A&T State University and the Deputy Director of the TECHLAV DoD Center of Excellence in Autonomy. His research interests include control and robotics, human-machine interactions, flight control systems, cyber-physical systems, and multi-agent systems. He has received over $25 million in grants and contracts from federal funding agencies and industrial partners to research development of autonomous vehicles and their applications such as smart transportation systems and smart agricultural systems. He is a senior member of IEEE.

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3145567, IEEE Transactions on Industrial Informatics

Mubbashar Altaf Khan (Ph.D., MBA) is a Postdoctoral Research Scholar for the Center of Excellence on Advanced Transportation at the North Carolina Agricultural and Technical State University. In the past, Dr. Khan worked as an Assistant Professor at Mirpur University of Science and Technology, Pakistan. Dr. Khan holds a Ph.D. degree in Engineering from the University of Toledo (2018). Dr. Khan's research interests include Cognitive Radios, Quality of Service (QoS) and secondary radio spectrum, autonomous systems, connected autonomous vehicles, Fuzzy logic and Fuzzy Systems, Testing and evaluation of autonomous systems, Human-machine Collaboration and trust modeling for machines, and Machine Learning Algorithms.

Steven Jiang is an Associate Professor in the Department of Industrial and Systems Engineering at North Carolina Agricultural and Technical State University. He received his Ph.D. in Industrial Engineering from Clemson University, Clemson, South Carolina. His research interests include human systems integration, human trust in automation, human-computer interaction, visual analytics, and cognitive engineering. Dr. Jiang has received over $4 million in research funding as a PI or co-PI. He has authored or coauthored more than 150 technical publications on conferences and journals. Dr. Jiang is a member of the Human Factors and Ergonomics Society, the Institute of industrial and Systems Engineers, and the American Society for Engineering Education.

1551-3203 (c) 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.