Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy

Citation: Nikolaidis, Stefanos, and Julie Shah. “Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy.” In 2013 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 33-40. Institute of Electrical and Electronics Engineers, 2013.

As Published: http://dx.doi.org/10.1109/HRI.2013.6483499

Publisher: Institute of Electrical and Electronics Engineers

Persistent URL: http://hdl.handle.net/1721.1/81277

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Human-Robot Cross-Training: Computational Formulation, Modeling and Evaluation of a Human Team Training Strategy

Stefanos Nikolaidis
Department of Aeronautics and Astronautics
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: snikol@mit.edu

Julie Shah
Department of Aeronautics and Astronautics
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139
Email: julie_a_shah@csail.mit.edu

***Abstract*—We design and evaluate human-robot cross-training, a strategy widely used and validated for effective human team training. Cross-training is an interactive planning method in which a human and a robot iteratively switch roles to learn a shared plan for a collaborative task.**

We first present a computational formulation of the robot's interrole knowledge and show that it is quantitatively comparable to the human mental model. Based on this encoding, we formulate human-robot cross-training and evaluate it in human subject experiments ($n = 36$). We compare human-robot cross-training to standard reinforcement learning techniques, and show that cross-training provides statistically significant improvements in quantitative team performance measures. Additionally, significant differences emerge in the perceived robot performance and human trust. These results support the hypothesis that effective and fluent human-robot teaming may be best achieved by modeling effective practices for human teamwork.

I. Introduction

When humans work in teams, it is crucial for the members to develop fluent team behavior. We believe that the same holds for robot teammates, if they are to perform in a similarly fluent manner as members of a human-robot team. New industrial robotic systems that operate in the same physical space as people highlight the emerging need for robots that can integrate seamlessly into human group dynamics. Learning from demonstration [4] is one technique for robot training that has received significant attention. In this approach, the human explicitly teaches the robot a skill or specific task [5], [1], [22], [10], [2]. However, the focus is on one-way skill transfer from a human to a robot, rather than a mutual adaptation process for learning fluency in joint-action. In many other works, the human interacts with the robot by providing high-level feedback or guidance [7], [14], [11], [29], but this kind of interaction does not resemble the teamwork processes naturally observed when human teams train together on interdependent tasks [19].

In this paper we propose a training framework that leverages methods from human factors engineering, with the goal of achieving convergent team behavior during training and team fluency at task execution, as it is perceived by the human partner and is assessed by quantitative team performance metrics. We computationally encode a teaming model that captures knowledge about the role of the robot and the human team member. The encoded model is quantitatively comparable to the human mental model, which represents the interrole knowledge held by the human [19]. Additionally, we propose quantitative measures to assess human-robot mental model convergence, as it emerges through a training process, as well as mental model similarity between the human and the robot. We then introduce a human-robot interactive planning method which emulates cross-training, a training strategy widely used in human teams [19]. We compare human-robot cross-training to standard reinforcement learning algorithms through a large-scale experiment of 36 human subjects, and we show that cross-training improves quantitative measures of human-robot mental model convergence ($p = 0.04$) and mental model similarity ($p < 0.01$). Additionally, a post-experimental survey shows statistically significant differences in perceived robot performance and trust in the robot ($p < 0.01$). Finally, we observe a significant improvement in team fluency metrics, including an increase of 71% in concurrent motion ($p = 0.02$) and a decrease of 41% in human idle time ($p = 0.04$), during the actual human-robot task execution phase that succeeds the human-robot interactive planning process.

Section II presents our computational formulation of the human-robot teaming model, as well as methods to assess mental model convergence and similarity. Section III introduces human-robot interactive planning using cross-training, and Section IV describes the human subject experiments. Section V presents and discusses the experiment results, which show a significant improvement in team performance using cross-training, as compared to standard reinforcement learning techniques. We place our work in context of other related work in Section VI and conclude with future research directions in Section VII.

II. Mental Model Formulation

The literature presents various definitions for the concept of shared mental models [18]. In the proposed framework we use the definition in Marks et al. [19], that mental models contain "the content and organization of interrole knowledge held by team members within a performance setting ... and ... contain procedural knowledge about how team members should work together on a task within a given task domain, including information about who should do what at particular points in time." We present a computational team model in the form of a Markov Decision Process (MDP) that captures the knowledge about the role of the robot and the human for a specific task [25].

A. Robot Mental Model formulated as MDP

We describe how the robot teaming model can be computationally encoded as a Markov Decision Process. A Markov decision process is a tuple $\{S, A, T, R\}$, where:

- S is a finite set of states of the world; it models the set of world environment configurations.
- A is a finite set of actions; this is the set of actions the robot can execute.
- $T : S \times A \longrightarrow \Pi(S)$ is the state-transition function, giving for each world state and action, a probability distribution over world states; the state transition function models the variability in human action. For a given robot action a, the human's next choice of action yields a stochastic transition from state s to a state s'. We write the probability of this transition as $T(s, a, s')$. In this formulation, human behavior is the cause of randomness in our model, although this can be extended to include stochasticity from the environment or the robot actions, as well.
- $R : S \times A \longrightarrow \mathbb{R}$ is the reward function, giving the expected immediate reward gained by taking each action in each state. We write $R(s, a)$ for the expected reward for taking action a in state s.

The policy π of the robot is the assignment of an action $\pi(s)$ at every state s. The optimal policy π^* can be calculated using dynamic programming [25]. Under this formulation, the role of the robot is represented by the optimal policy π^*, whereas the robot's knowledge of the role of the human co-worker is represented by the transition probabilities T.

B. Evaluation of Mental Model Convergence

As the mental models of the human and robot converge, we expect the human and robot to perform similar patterns of actions. This means that the same states will be visited frequently and the robot uncertainty about the human's action selection will decrease. Additionally, if the mental models of the human and the robot converge, the patterns of actions performed will match the human preference, as it is elicited after the training.

To evaluate the convergence of the robot's computational teaming model and the human mental model, we assume a uniform prior and compute the *entropy rate* [12] of the Markov chain (Eq. 1). The Markov chain is induced by specifying a policy π in the MDP framework. For the policy π we use the robot actions that match human preference, as it is elicited by the human after training with the robot. Additionally, we use the states $s \in S$ that match the preferred sequence of configurations to task completion. For a finite state Markov chain X with initial state s_0 and transition probability matrix T the entropy rate is always well defined [12]. It is equal to the sum of the entropies of the transition probabilities $T(s, \pi(s), s')$, for all $s \in S$, weighted by the probability of occurrence of each state according to the stationary distribution μ of the chain (Equation 1).

$$H(X) = -\sum_{s \in S} \mu(s) \sum_{s' \in S} T(s, \pi(s), s') log \left[T(s, \pi(s), s') \right] \tag{1}$$

Interestingly, the conditional entropy, given by Eq. 1, also represents the robot's uncertainty about the human's action selection, which we expect to decrease as human and robot train together. We leave for future work the case where when the human has multiple preferences or acts stochastically.

C. Human-Robot Mental Model Similarity

Given the robot mental model formulation, we propose a similarity metric between the mental model of human and robot, based on prior work [18] on shared mental model elicitation for human teams. In a military simulation study [20], each participant was asked to annotate a sequence of actions to achieve mission completion, for himself, as well as for his other team members. Then, the degree of mental model similarity was calculated by assessing the overlap in action sequences selected by each of the team members. We generalize this approach on a human-robot team setting. In our study, the participant annotates a sequence of actions that he or she thinks that the human and the robot should do to complete the assigned task. We then elicit the similarity of the human and robot mental model by taking the ratio of the annotated robot actions that match the actions assigned by the optimal policy of the robot, to the total number of robot actions required for task completion. This describes how well the human preference for the robot actions matches the actual optimal policy for the MDP.

III. Human Robot Interactive Planning

Expert knowledge about the task execution is encoded in the assignment of rewards R, and in the priors on the transition probabilities T that encode the expected human behavior. This knowledge can be derived from task specifications or from observation of expert human teams. However, rewards and transition probabilities finely tuned to one human worker are not likely to generalize to another human worker, since each worker develops his or her own highly individualized method for performing manual tasks. In other words, a robot that works with one person according to another person's preferences is not likely to be good teammate. In fact, it has been shown in previous research that human teams whose members have similar mental models perform better than teams with more accurate but less similar mental models [19]. Even if the mental model learned by observation of a team of human experts is accurate, the robot needs to adapt this model when asked to work with a new human partner. The goal then becomes for the newly formed human-robot team to develop a shared-mental model. One validated and widely used mechanism for conveying shared mental models in human teams is "cross-training [19]." We emulate the cross-training process among human team-members by having the human and robot train together at a virtual environment. We use a virtual environment as, especially in high-intensity applications, the cost of training with an actual robot in the operational environment (e.g. on the assembly line or in space) can be prohibitive. In the following sections we briefly describe the cross-training process in human teams and then describe how we emulate this process in human-robot teams.

A. Cross-Training in Human Teams

There are three types of cross-training [6]: (a) positional clarification, (b) positional modeling, and (c) positional rotation. Findings [19], [8] suggest that positional rotation, which is defined as "learning interpositional information by switching work roles," is the most strongly correlated to improvement in team performance, since it provides the individual with hands-on knowledge about the roles and responsibilities of other teammates, with the purpose of improving interrole knowledge and team performance [19].

B. Cross-Training Emulation in Human-Robot Team

We emulate positional rotation in human teams by having the human and robot iteratively switch roles. We name the phase where the roles of the human and robot match the ones of the actual task execution as the *forward phase*, and the phase where human and robot roles are switched as *rotation phase*. In order for the robot's computational teaming model to converge to the human mental model:

1) The robot needs to have an accurate estimate of the human's role in performing the task, and this needs to be similar to the human's awareness of his or her own role. Based on the above, we use the human-robot forward phase of the training process to update our estimation of the transition probabilities that encode the expected human behavior.
2) The robot's actions need to match the expectations of the human. We accomplish this by using the human inputs in the rotation phase to update the reward assignments.

```
Algorithm : Human-Robot Cross-training
1. Initialize $R(s, a)$ and $T(s, a, s')$ from prior knowledge
2. Calculate initial policy $\pi$
3. while(number of iterations < $MAX$)
4.     Call Forward-phase($\pi$)
5.     Update $T(s, a, s')$ from observed sequence $s_1, a_1, s_2, \dots, s_{M-1}, a_{M-1}, s_M$
6.     Call Rotation-phase()
7.     Update $R(s_i, a_i)$ for observed sequence $s_1, a_1, s_2, a_2, \dots, s_N, a_N$
8.     Calculate new policy $\pi$
9. end while
```

Fig. 1. Human-Robot Cross-Training Algorithm

1) Cross-Training for Human-Robot Team: The Human-Robot Cross-training algorithm is summarized in Figure 1. In Line 1, rewards $R(s, a)$ and transition probabilities $T(s, a, s')$ are initialized from prior knowledge about the task. In Line 2, an initial policy π is calculated for the robot. In our implementation we used value iteration [25]. In Line 4, the Forward-phase function is called, where the human and robot train on the task. The robot chooses its actions depending on the current policy π, and the observed state and action sequence is recorded. In Line 5, $T(s, a, s')$ are updated based on the observed state-action sequence. $T(s, a, s')$ describes the probability that for a task configuration modeled by state s, and robot action a, the human will perform an action such that the next state is s'.

All transition probabilities as described above are given by multinomial distributions and are estimated by the transition frequencies, assuming a pre-observation count [11]. The pre-observation count corresponds to the size of a real or imaginary sample-set from which the transition probabilities of the MDP are initialized by the model designer, before the training. It is a measure of the confidence the model designer has on how close the initial model is to the expected behavior of the new human worker.

In the rotation phase (Line 6), the human and robot switch task roles. In this phase, the observed actions $a \in A$ are the actions performed by the human worker, whereas the states $s \in S$ remain the same. In Line 7, the rewards $R(s, a)$ are updated for each observed state s and human action a. We then use the new estimates for $R(s, a)$ and $T(s, a, s')$ to update the current policy (Line 8). The new optimal policy is computed using standard dynamic programming techniques [25].

In our implementation we update the rewards (Line 7) as follows:

$$R(s, a) = R(s, a) + r \tag{2}$$

The value of the constant r needs to be large enough, compared to the initial values of $R(s, a)$, for the human actions to affect the robot's policy. Note that our goal is not to examine the best way to update the rewards, something which has been shown to be task-dependent [17]. Instead, we aim to provide a general human-robot training framework and use the reward update of Eq. 2 as an example. Knox and Stone [16] evaluate eight methods for combining human inputs with MDP reward in a reinforcement learning framework. Alternatively, inverse reinforcement learning algorithms could be used to estimate the MDP rewards from human input [1].

We iterate the forward and rotation phases for a fixed number of *MAX* iterations, or until a convergence criterion is met.

2) Forward Phase: The pseudocode of the forward phase is given by Figure 2. In Line 1, the current state is initialized to the start step of the task episode. The FINAL_STATE in Line 2 is the terminal state of the task episode. In Line 3, the robot executes an action a assigned to a state s, based on the current policy π. The human action is observed (Line 4) and the $next\_state$ variable is set according to the $current\_state$, the robot action a and the human action. In our implementation, we use a look-up table that sets the next state for each state and action combination. Alternatively, the next state could be directly observed after the human and robot finish executing their actions. The state, action, and next state of the current time-step are recorded (Line 6).

```
Function: Forward-phase(policy $\pi$)
1.  Set current_state = START_STATE
2.  while(current_state != FINAL_STATE)
3.      Execute robot action $a$ according to current policy $\pi$
4.      Observe human action
5.      Set next_state to the state resulting from current_state, robot
        and human action
6.      Record current_state, $a$, next_state
7.      current_state = next_state
8.  end while
```

Fig. 2. *Forward Phase* of the Cross-Training Algorithm

3) Rotation Phase: The pseudocode of the rotation phase is given by Figure 3. In Line 3, the action a is the observed human action. In Line 4, a robot action is sampled from the transition probability distribution $T(s, a, s')$.

Just as the transition probability distributions of the MDP are updated after the forward phase, the robot policy is updated to match the human expectations after the rotation phase. This process emulates how a human mental model would change by working together with a partner. A key feature of the cross-training approach is that it provides an opportunity for the human to adapt to the robot, as well.

```
Function: Rotation-phase()
1.  Set current_state = START_STATE
2.  while (current_state != FINAL_STATE)
3.      Set action $a$ to observed human action
4.      Sample robot action from  T(current_state, $a$, next_state)
5.      Record current_state, $a$
6.      current_state = next_state
7.  end while
```

Fig. 3. *Rotation Phase* of the Cross-Training Algorithm.

C. Reinforcement Learning with Human Reward Assignment

We compare the proposed formulation to the interactive reinforcement learning approach where the reward signal of an agent is determined by interaction with a human teacher [30]. We chose as reinforcement learning algorithm Sarsa(λ) with greedy policy [27], for its popularity and applicability in a wide variety of tasks. In particular, Sarsa(λ) has been used to benchmark TAMER framework [15], as well as to test TAMER-RL [16], [17]. Furthermore, our implementation of Sarsa(λ) is identical to the Q-Learning with Interactive Rewards [29], if we remove eligibility traces on Sarsa (by setting $\lambda = 0$) and set a greedy policy for both algorithms. Variations of Sarsa have been used to teach a mobile robot to deliver objects [24], for navigation of a humanoid robot [21], as well as in an interactive learning framework, where the user gives rewards to the robot through verbal commands [28].

After each robot action, the human is asked to assign a good, neutral, or bad reward $\{+r, 0, -r\}$. In our current implementation we set the value of r, which is the reward signal assigned by the human, to be identical to the value of the reward update in cross-training (Eq. 2 in Section III-B) for comparison purposes.

IV. Human-Robot Teaming Experiments

A. Experiment Hypotheses

We conduct a large-scale experiment ($n = 36$) to compare human-robot cross-training to standard reinforcement learning techniques. The experiment tests the following three hypotheses about human-robot team performance.

- Hypothesis 1: Human-robot interactive planning with cross-training will improve quantitative measures of **human-robot mental model convergence** and **mental model similarity**, compared to human-robot interactive planning using reinforcement learning with human reward assignment. We base this on prior work showing that cross-training improves similarity of mental models of human team members [19], [8].
- Hypothesis 2: Participants that cross-trained with the robot will agree more strongly that **the robot acted according to their preferences**, compared to participants that trained with the robot by assigning rewards. Furthermore, we hypothesize that they will agree more strongly that **the robot is trustworthy**. We base this upon prior work [26] that shows that humans find the robot more trustworthy when it emulates the effective coordination behaviors observed in human teams.
- Hypothesis 3: Human-robot interactive planning with cross-training will improve **team-fluency metrics on task-execution**, compared to human-robot interactive planning using reinforcement learning with human reward assignment. We base this on the wide usage of cross-training to improve performance of human teams [19].

B. Experiment Setting

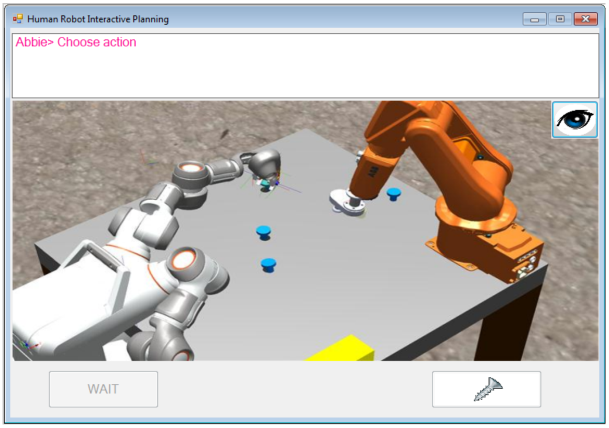
We apply the proposed framework to train a team of one human and one robot to perform a simple place-and-drill task, as a proof of concept. The human's role is to place screws in one of three available positions. The robot's role is to drill each screw. Although this task is simple, we found it adequate for testing of our framework, since there is a sufficient variety on how to accomplish the task among different persons. For example, some participants preferred to place all screws in a sequence from right-to-left and then have them drilled in the same sequence. Others preferred to place and drill each screw before moving on to the next. The participants consisted of 36 subjects recruited from MIT. Videos of the experiment can be found at: **http://tinyurl.com/9prt3hb**

C. Human-Robot Interactive Training

Before starting the training, all participants were asked to describe both verbally and in written form their preferred way of executing the task. We then initialized the robot policy from a set of prespecified policies so that it was clearly different from the participant's preference. For example, if the user preferred to "have the robot drill all screws as soon as they are placed, starting from left to right," we initialized the MDP framework so that the starting robot policy was to wait till all screws were placed before drilling. We did this to avoid the trivial case where the initial policy of the robot matches the preferred policy of the user, and to evaluate mental model convergence starting from different human and robot mental models.

The participants were randomly assigned to two groups, Group A and Group B. Each participant then did a training session in the ABB RobotStudio virtual environment with an industrial robot which we call "Abbie" (Figure 4). Depending on the assigned group, the participant participated in the following training session:

1) Cross-training session (Group A): The participant iteratively switches positions with the virtual robot, placing the screws at the forward phase and drilling at the rotation phase.
2) Reinforcement learning with human reward assignment session (Group B): This is the standard reinforcement learning approach, where the participant places screws and the robot drills at all iterations, with the participant assigning a positive, zero, or negative reward after each robot action [11].

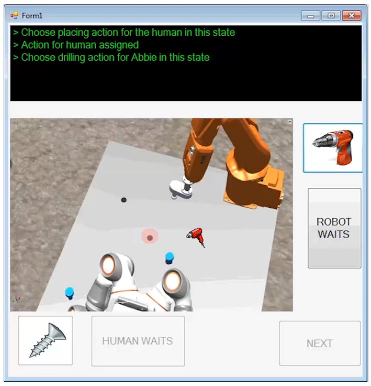


Abbie> Choose action

Fig. 4. Human-Robot Interactive Planning Using ABB RobotStudio Virtual Environment. The human controls the white anthropomorphic "Frida" robot on the left, to work with the orange industrial robot, "Abbie," on the right.

For the cross-training session, the policy update (Line 8 of Figure 1, Section III-B) was performed using value iteration with a discount factor of 0.9. The Sarsa(λ) parameters in the standard notation of Sarsa [27] were empirically tuned ($\lambda = 0.9, \gamma = 0.9, \alpha = 0.3$) for best performance on this task.

After the training session, the mental model of all participants was assessed with the method described in Section II-C. For each workbench configuration through task completion, participants were asked to choose a human placing action and their preference for an accompanying robot drilling action, based on the training they had together (Figure 5).



> Choose placing action for the human in this state
> Action for human assigned
> Choose drilling action for Abbie in this state

Fig. 5. Human-Robot Mental Model Elicitation Tool

D. Human-Robot Task Execution

We then asked all participants to perform the place-and-drill task with the actual robot, Abbie. To recognize the actions of the human we used a Phasespace motion capture system of eight cameras [23], which tracked the motion of a Phasespace glove worn by the participant (Figure 6). Abbie executed the policy learned from the training sessions. The task execution was videotaped and later analyzed for team fluency metrics. Finally, all participants were asked to answer a post-experiment survey.

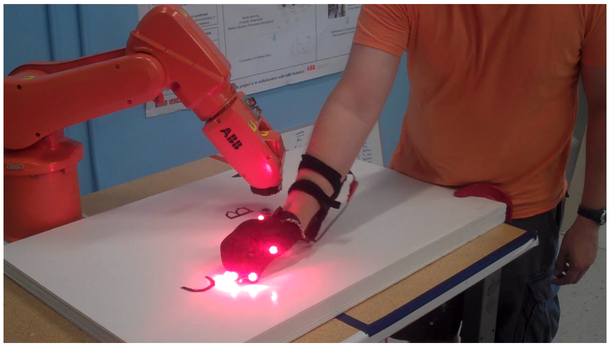


Fig. 6. Human-Robot Task Execution

V. Results and Discussion

Results of the human subject experiments show that the proposed cross-training method outperforms standard reinforcement learning in a variety of quantitative and qualitative measures. This is the first evidence that human-robot teamwork is improved when a human and robot train together by switching roles, in a manner similar to effective human team training practices. Unless stated otherwise, all the p-values in this section are computed for *two-tailed unpaired t-tests with unequal variance*.

A. Quantitative Measures

1) Mental Model Similarity: As described in Section II-C, we compute the mental model similarity metric as the ratio of the human drilling actions that match the actions assigned by the robot policy, to the total number of drilling actions required for task completion. Participants of Group A had an average ratio of 0.96, compared to an average ratio of 0.75 for Group B ($p < 0.01$). This shows that participants that cross-trained with the robot developed mental models that were more similar to the robot teaming model, compared to participants that trained with the robot by assigning rewards.

2) Mental Model Convergence: Mental model similarity was also reflected by similar patterns of actions during the training process, and by decreased robot uncertainty about the human's action selection, as computed by the entropy rate of the Markov Decision Process (Section II-B). We compute the entropy rate at each training round using the preferred robot policy, as elicited by the human with the mental model elicitation tool (Figure 5 of Section IV-C). Since the initial value of the entropy rate varies for different robot policies, we use the percent decrease, averaged over all participants of each group, as a metric to compare cross-training to reinforcement learning with human reward assignment. To calculate the entropy rate in the human reward assignment session, we update the transition probability matrix T from the observed state and action sequences, in a manner identical to how we calculate the entropy-rate for the cross-training session. We do this for comparison purposes, since Sarsa(λ) is a model-free algorithm and does not use T in the robot action selection [27].

Figure 7 shows the entropy rate after each training round for participants of both groups. We consider only the participants that did not change their preference (28 out of 36 participants). The difference for the two groups after the last training round is statistically significant ($p = 0.04$). This shows that the robot's uncertainty in the human participant's actions after the training is significantly lower for the group that cross-trained with the robot, than for participants who trained using reinforcement learning with human reward assignment.

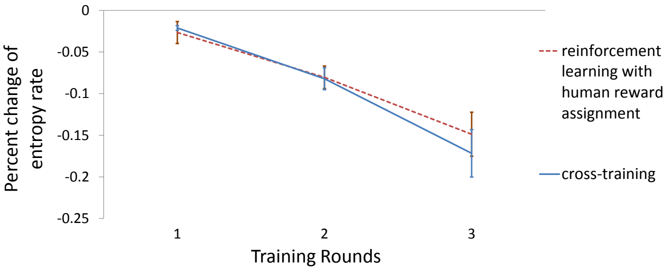


Fig. 7. Human-Robot Mental Model Convergence. The graph shows the percent decrease of entropy rate over training rounds.

Interestingly, for participants of Group A, there is a statistically significant difference in the entropy value after the last training round between those that kept their preference throughout the training and those that switched preferences ($p < 0.01$). This shows that the entropy rate could be a valid metric to detect changes in the human behavior or mistakes by the operator, and warrants further investigation.

We noticed that the cross-training session lasted slightly longer than the reinforcement learning with human reward assignment session, since switching roles on average took more time than assigning a reward after each robot action. Since participants often interrupted the training to interact with the experimenters, we were unable to reliably measure the training time for the two groups.

The above results support our first hypothesis that cross-training improves quantitative measures of human-robot mental model convergence.

B. Qualitative Measures

After each training round, the participant was asked to rate his or her agreement with the statement "In this round, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence" on a five-point Likert scale. Furthermore, after the end of the experiment, participants were asked to fill in a post-experimental survey. On a five-point Likert scale, subjects that cross-trained and then executed the task with Abbie, Group A, selected a significantly higher mark than those that trained with Abbie using the standard reinforcement learning method, Group B, when asked whether:

- "In this round, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence.":
(For the final training round) Group A: 4.52 [SD=0.96]; Group B: 2.71 [SD=1.21]; $p < 0.01$
- "In the actual task execution, Abbie performed her role exactly according to my preference, drilling the screws at the right time and in the right sequence.":
Group A: 4.74 [SD=0.45]; Group B: 3.12 [SD=1.45]; $p < 0.01$
- "I trusted Abbie to do the right thing at the right time.":
Group A: 3.84 [SD=0.83]; Group B: 2.82 [SD=1.01]; $p < 0.01$
- "Abbie is trustworthy.":
Group A: 4.05 [SD=0.71]; Group B: 3.00 [SD=0.93]; $p < 0.01$
- "Abbie does not understand how I am trying to execute the task.":
Group A: 1.89 [SD=0.88]; Group B: 3.24 [SD=0.97]; $p < 0.01$
- "Abbie perceives accurately what my preferences are.":
Group A: 4.16 [SD=0.76]; Group B: 2.76 [SD=1.03]; $p < 0.01$

The p-values above are computed for a two-tailed Mann-Whitney-Wilcoxon test. The results show that participants of Group A agreed more strongly that Abbie learned their preferences, compared to participants of Group B. Furthermore, cross-training had a positive impact on their trust in Abbie, in accordance with prior work [26]. This supports Hypothesis 2 of Section IV-A. The two groups did not differ significantly when subjects were asked whether they themselves were "responsible for most of the things that the team did well on this task," whether they were "comfortable working in close proximity with Abbie," or whether themselves and Abbie "were working towards mutually agreed upon goals."

C. Fluency Metrics on Task Execution

We elicit the fluency of the teamwork by measuring the concurrent motion of the human and robot and the human idle time during task execution phase, as proposed in [13]. The measurements of the above metrics were evaluated by an independent analyst who did not know the purposes of the experiment, nor the group of the participant. Additionally, we automatically compute the robot idle time and the human-robot distance. Since these metrics are affected by the human's preferred way of doing the task, we use only the subset of participants that self-reported their preferred strategy as the strategy of "while Abbie is drilling a screw, I will place the next one." The subset consists of 20 participants, and this is the largest subset of participants that reported the same preference on task execution.

We leave for future work the evaluation of task performance metrics such as task completion time, since for this experiment these measures largely depend on the preferred human strategy, and our current goal is to measure fluency in shared-location joint-action, rather than optimize task-specific metrics.

1) Concurrent Motion: We measured the time duration in which both human and robot were concurrently in motion during the task execution phase. Analysis shows that participants of Group A that preferred to "finish the task as fast as possible, placing a screw while Abbie was drilling the previous one" had a 71% increase in the time of concurrent motion with the robot, compared to participants of Group B that reported the same preference (A: 5.44 sec [SD = 1.13 sec]; B: 3.18 sec [SD = 2.15 sec]; $p = 0.02$). One possible explanation for these differences is that cross-training engendered more trust in the robot (supported by subjective results presented in Section V-B), and thereby participants of Group A had more confidence to act while the robot was moving.

2) Human Idle Time: We measured the amount of time the human spent waiting for the robot. Participants of Group A spent 41% less time idling, on average, than those of Group B, a statistically significant difference (A: 7.19 sec [SD = 1.71 sec]; B: 10.17 sec [SD = 3.32 sec]; $p = 0.04$). In some cases, the increase in idle time was caused because the participant was not sure on what the robot would do next, and therefore waited to see. In other cases, the robot had not learned correctly the human preference and did not act accordingly, with the result of forcing the human to wait, or confusing the human team-member.

3) Robot Idle Time: The time that the robot remained idle waiting for the human to make an action, such as place a screw, was calculated automatically by our task-execution software. We found the difference between Group A and Group B to be statistically significant (A: 4.61 sec [SD = 1.97 sec]; B: 9.22 sec [SD = 5.07 sec]; $p = 0.04$).

4) Human-Robot Distance: Statistically significant results across Group A and Group B were found for the distance of the human hand to the robot base, averaged over the time the robot was moving, and normalized to the baseline distance of the participant ($p = 0.03$). The difference resulted since some participants of Group B "stood back" while the robot was moving. Previous work has shown using physiological measures that mental strain of the operators is strongly correlated with the distance of a human worker to an industrial manipulator moving at high-speed [3]. We therefore suggest that cross-training with the robot may have a positive impact on emotional aspects such as fear, surprise and high-tension, but we leave further investigation for future work.

The above results confirm our third hypothesis that human-robot interactive planning with cross-training improves team fluency metrics on task execution, compared to human-robot interactive planning using reinforcement learning with human reward assignment.

VI. Related Work

In this study we benchmark our cross-training methodology against the Sarsa(λ) reinforcement learning approach where the reward signal is interactively assigned by the human. Both these techniques may be categorized as learning where the **human and machine engage in high-level evaluation and feedback**. In other approaches in this category, a human trainer assigns signals of positive reinforcement [7], [14], a method also known as "clicker training," or of both positive and negative reinforcement. Other methods, such as TAMER-RL [17], support the use of human input to guide an agent in maximizing an environmental reward. Prior investigations into TAMER-RL and Sarsa(λ) assume an objective performance metric is known and do not consider other metrics, such as trainer satisfaction. Q-Learning with Interactive Rewards [29] is identical to our version of Sarsa(λ), if we remove eligibility traces on Sarsa and set a greedy policy for both algorithms. A modified version [29] incorporating human guidance has been empirically shown to significantly improve several dimensions of learning. Human rewards have also been used as additional input to verbal commands, to simultaneously teach a system and learn a model of the user [11]. Additionally, active learning systems [9] have been used to improve objective measures of agent performance in a simulated driving domain from a few informative examples. In multi-agent settings, state-of-the-art behavior modeling based on the game-theoretic notion of regret and the principle of maximum entropy has been shown to accurately predict future behavior in newly encountered domains [31]. Our contribution is a human-team inspired approach to achieve fluency in action-meshing.

The other category for learning is where the **human provides demonstrations to the machine.** Our rotation-phase of the cross-training algorithm resembles this type of learning, since the reward function is inferred from the inputs of the human. Other work in learning from demonstration includes systems that learn a general policy for the task by passively observing a human expert executing the task; Atkeson and Schaal [5] address the challenge of teaching a robot arm to mimic the human expert's trajectory. Apprenticeship Learning [1] has enabled agents to perform as well as human experts in dynamic applications such as highway driving. In a number of recent works, the robot refines learned task representations using human demonstrations enriched with verbal instructions [22], multiple robots are taught to coordinate their actions using a GUI interface [10], the robot is physically guided by the human using trajectory and keyframe demonstrations [2], or the reinforcement learning agent receives guidance using a video-game environment [29]. While producing impressive results, the focus of these approaches is on one-way skill or behavior transfer to an agent, rather than the two-way mutual adaptation process that cross-training supports.

VII. Conclusion

We designed and evaluated human-robot cross-training, a strategy widely used and validated for effective human team training. Cross-training is an interactive planning method in which a human and a robot iteratively switch roles to learn a shared plan for a collaborative task. We first presented a computational formulation of the robot's teaming model and show that it is quantitatively comparable to the human mental model. Based on this encoding, we formulated human-robot cross-training and evaluated it in a large-scale experiment of 36 subjects. We show that cross-training improves quantitative measures of human-robot mental model convergence ($p = 0.04$) and mental model similarity ($p < 0.01$). Additionally, a post-experimental survey shows statistically significant differences in perceived robot performance and trust in the robot ($p < 0.01$). Finally, we observed a significant improvement in team fluency metrics, including an increase of 71% in concurrent motion ($p = 0.02$) and a decrease of 41% in human idle time ($p = 0.04$), during the human-robot task execution phase. These results provide the first evidence that human-robot teamwork is improved when a human and robot train together by switching roles, in a manner similar to effective human team training practices.

In this experiment we focused on a simple place-and-drill task, as a proof of concept. We are currently planning on using the robot uncertainty about the human's next action to influence the motion planning parameters for a robot working alongside a person. Future work includes extending the computational formulation of the robot's teaming model to a POMDP framework that incorporates information-seeking behavior, and testing the framework on more complex tasks. Additionally, we plan to investigate the proposed metric of mental model convergence as a method to automatically detect changes in the operator's behavior or human mistakes. Finally, it would be interesting to assess the impact of cross-training on emotional states of the operator, such as fear, surprise and discomfort.

References

[1] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. ICML*. ACM Press, 2004.

[2] B. Akgun, M. Cakmak, J. W. Yoo, and A. L. Thomaz, "Trajectories and keyframes for kinesthetic teaching: a human-robot interaction perspective," in *HRI*, 2012, pp. 391–398.

[3] T. Arai, R. Kato, and M. Fujita, "Assessment of operator stress induced by robot collaboration in assembly," *CIRP Annals - Manufacturing Technology*, vol. 59, no. 1, pp. 5 – 8, 2010.

[4] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robot. Auton. Syst.*, vol. 57, no. 5, pp. 469–483, May 2009.

[5] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *ICML*, 1997, pp. 12–20.

[6] C.-B. J. Blickensderfer E. and S. E., "Cross-training and team performance," *Making decisions under stress: Implications for individual and team training, American Psychological Association*, pp. 299–311, 1998.

[7] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson, "Integrated learning for interactive synthetic characters," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 417–426, Jul. 2002.

[8] B. E. B. C. Cannon-Bowers J.A., Salas E., "The impact of cross-training and workload on team functioning: a replication and extension of initial findings." *Human Factors*, pp. 92–101, 1998.

[9] S. Chernova and M. Veloso, "Multi-thresholded approach to demonstration selection for interactive robot learning," in *Proc. HRI*. New York, NY, USA: ACM, 2008, pp. 225–232.

[10] ——, "Teaching multi-robot coordination using demonstration of communication and state sharing," in *Proc. AAMAS*, Richland, SC, 2008.

[11] F. Doshi and N. Roy, "Efficient model learning for dialog management," in *Proc. HRI*, Washington, DC, March 2007.

[12] L. Ekroot and T. Cover, "The entropy of markov trajectories," *Information Theory, IEEE Transactions on*, vol. 39, no. 4, pp. 1418 –1421, jul 1993.

[13] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team," in *Proc. HRI*. New York, NY, USA: ACM, 2007, pp. 1–8.

[14] F. Kaplan, P.-Y. Oudeyer, E. Kubinyi, and A. Miklósi, "Robotic clicker training," *Robotics and Autonomous Systems*, vol. 38, no. 3-4, pp. 197–206, 2002.

[15] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The tamer framework," in *Proc. K-CAP*, September 2009.

[16] ——, "Combining manual feedback with subsequent mdp reward signals for reinforcement learning," in *Proc. AAMAS*, May 2010.

[17] ——, "Reinforcement learning from simultaneous human and mdp reward," in *Proc. AAMAS*, June 2012.

[18] J. Langan-Fox, S. Code, and K. Langfield-Smith, "Team mental models: Techniques, methods, and analytic approaches," *Human Factors*, vol. 42, no. 2, pp. 242–271, 2000.

[19] M. Marks, M. Sabella, C. Burke, and S. Zaccaro, "The impact of cross-training on team effectiveness," *J Appl Psychol*, vol. 87, no. 1, pp. 3–13, 2002.

[20] M. A. Marks, S. J. Zaccaro, and J. E. Mathieu, "Performance implications of leader briefings and team-interaction training for team adaptation to novel environments," *J Appl Psychol*, vol. 85, pp. 971–986, 2000.

[21] N. Navarro, C. Weber, and S. Wermter, "Real-world reinforcement learning for autonomous humanoid robot charging in a home environment," in *Proc. TAROS*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 231–240.

[22] M. N. Nicolescu and M. J. Mataric, "Natural methods for robot task learning: Instructive demonstrations, generalization and practice," in *Proc. AAMAS*, 2003, pp. 241–248.

[23] (2012) Phasespace motion capture http://www.phasespace.com.

[24] D. Ramachandran and R. Gupta, "Smoothed sarsa: reinforcement learning for robot delivery tasks," in *Proc. ICRA*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3327–3334.

[25] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[26] J. Shah, J. Wiken, B. Williams, and C. Breazeal, "Improved human-robot team performance using chaski, a human-inspired plan execution system," in *Proc. HRI*. New York, NY, USA: ACM, 2011, pp. 29–36.

[27] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[28] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseñor Pineda, "Dynamic reward shaping: training a robot by voice," in *Proc. IBERAMIA*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 483–492.

[29] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: evidence of feedback and guidance with implications for learning performance," in *Proc. AAAI*, 2006, pp. 1000–1005.

[30] A. L. Thomaz, G. Hoffman, and C. Breazeal, "C.: Real-time interactive reinforcement learning for robots," in *Proc. of AAAI Workshop on Human Comprehensible Machine Learning*, 2005.

[31] K. Waugh, B. D. Ziebart, and J. A. D. Bagnell, "Computational rationalization: The inverse equilibrium problem," in *Proc. ICML*, June 2011.