Human-Robot Interactive Planning using Cross-Training: A Human Team Training Approach

Stefanos Nikolaidis* and Julie Shah†

Massachusetts Institute of Technology, Cambridge, MA 02139, United States

Robots are increasingly introduced to work in concert with people in high-intensity domains, such as manufacturing, space exploration and hazardous environments. Although there are numerous studies on human teamwork and coordination in these settings, very little prior work exists on applying these models to human-robot interaction. In this paper we propose a novel framework for applying prior art in Shared Mental Models (SMMs) to promote effective human-robot teaming. We present a computational teaming model to encode joint action in a human-robot team. We present results from human subject experiments that evaluate human-robot teaming in a virtual environment. We show that cross-training, a common practice used for improving human team shared mental models, yields statistically significant improvements in convergence of the computational teaming model (p=0.02) and in the human participants' perception that the robot performed according to their preferences (p=0.01), as compared to robot training using a standard interactive reinforcement learning approach.

I. Introduction

We propose a novel framework that uses insight from prior art in human team coordination and shared mental models to increase the performance of human-robot teams executing complex tasks under uncertainty and time constraints. Shared Mental Models (SMMs)[5] are measurable models developed among team members prior to task execution and are strongly correlated to team performance. Although numerous studies have modeled the performance-linked characteristics of SMMs in human team coordination, very little prior work exists on applying these models to a human-robot interaction framework. We propose that valuable insights can be drawn from these works. For instance, a study evaluating teamwork in flight crews[12] has shown that teams with accurate but different mental models among team members perform worse than teams having less accurate but common models. Applying this insight to human-robot teaming leads to a hypothesis that, to promote effective teamwork, a robot must execute a task plan that is similar to the human partner's mental model of the execution.

In this paper, we outline a framework that leverages methods from human factors engineering to promote the development of teaming models that are shared across human and robot team members. The human factors community has developed and validated a widely used set of techniques for eliciting, quantitatively evaluating, and strengthening SMMs for human teamwork.[10,11,15] Studies of military tactical operations and aviation crews show that improved team performance and reduction of errors is strongly correlated to the similarity of the team members' mental models.[12,13] High quality shared mental models facilitate the use of implicit communications, which are documented to play a primary role in effective human team coordination.[17,19] These findings have potentially important implications for designing safe and effective human interaction with robots in safety-critical domains.

Our approach focuses on a human-robot **interactive-training phase**,[11,19,22] which preceeds task execution. The purpose of the planning phase is to derive a high-quality teamwork model that provides a perceptual common ground to predict co-workers' actions, and integrate predicted effects of the humans'

*Graduate Student, Department of Aeronautics and Astronautics, Massachusetts 77 Avenue, Cambridge, MA 02139, AIAA Student Member.

†Boeing Assistant Professor, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Massachusetts 77 Avenue, Cambridge, MA 02139, AIAA Professional Member.

and robots' actions. The proposed methodology builds on prior work showing that cross-training, and in particular positional rotation among team members, increases the similarity of the team members' mental models and subsequently improves team performance.[11,22]

II. APPROACH AND METHODOLOGY

A. Hypotheses

We have conducted human-subject experiments to compare the effectiveness of human-robot team training using cross-training versus the traditional ask-for-reward approach[7] where the human operator rates each robot action. We hypothesize:

- Hypothesis 1: We expect that positional-rotation cross-training will promote predictable patterns of human and robot action sequences. Specifically, we hypothesize that positional-rotation cross-training within a human-robot team will yield a statistically significant improvement in an objective measure of the robot's uncertainty about the human partner's action, compared to the traditional ask-for-reward approach.
- Hypothesis 2: We expect that positional-rotation cross-training will improve subjective measures of the robot performance. Specifically we hypothesize that human participants cross-training together with the robot will agree more strongly to statement that the robot perceives accurately what the human's preferences are.

B. Human-Robot Team Training Process

We outline the process for human-robot team training:
(1) The robot derives an initial teaming model from prior knowledge of the task or from observation of coordinated manual work performed by two or more expert human workers. The robot mental model is encoded as a Markov Decision Process (MDP).[1] For the experiments conducted, the robot was provided a naive model for human action selection where the human participant chooses any feasible action sequence with equal probability.
(2) Next, the robot and a new worker perform structured interactive planning of a simple task in a computer simulation environment. We computationally formulate the *positional rotation* cross-training type as an iterative reinforcement learning process. We compare this to learning through the standard "ask-for-reward" reinforcement learning framework.
(3) We assess the convergence of the learnt shared mental model of the human-robot team.

III. Mental Model formulation

The literature presents various definitions for the concept of Shared Mental Models.[9] In the proposed framework we use the definition in Marks et al.,[11] 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 that captures this knowledge about the role of the robot and the human for a specific task as a Markov Decision Processes. In the following section, we review the definition of a Markov Decision Processes.

A. Markov Decision Processes

A Markov decision process is a tuple { S, A, T, R }, where

- S is a finite set of states of the world;
- A is a finite set of actions;
- $T : S \times A \longrightarrow \Pi(S)$ is the state-transition function, giving for each world state and action, a probability distribution over world states;
- $R : S \times A \longrightarrow R$ 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);

In this formulation, a policy is a decription of the behavior of an acting agent. A policy is a function that maps each state to the action set:

$$\pi : S \longrightarrow A \tag{1}$$

The *optimal policy* that maximizes the total expected reward of the lifetime of an agent can be found using dynamic programming.[1]

B. Robot Mental Model formulated as MDP

Next, we describe how the human-robot teaming model can be computationally encoded as a Markov Decision Process.

- S: This is the set of world (environment and agent) configurations. For an assembly task, the set of states S may be modeled as the set of workbench configurations.
- A: This is the set of actions the robot can execute. We assume the robot actions are deterministic, in a sense that the stochastic transitions to subsequent states are determined only by the human behavior. This assumption is reasonable for settings where robot action completion can be modeled as deterministic, such as for industrial robots in assembly manufacturing applications.
- $T(s'|s,a)$: 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'. In other words, human behavior is the cause of randomness in our model.
- $R(s,a)$: The reward function is used to specify correct robot actions and is specified based on prior knowledge of successful task completion. As we explain in the next section, we use human inputs during the human-robot training phase to update the reward function.

The policy π of the robot is the assignment of an action π_s at every state s. Under this formulation, the *role* of the robot is represented by the policy π, whereas the knowledge of the robot about the role of the human co-worker is represented by the transition probabilies T.

IV. 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 at each planstep. 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 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.[11] Even if the mental model learnt by observation of the two 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".[11] 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[2] a) positional clarification (b) positional modeling and c) positional rotation. Findings[3,11,22] 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, as it "provides hands on approach to learning interpositional information by giving members experience on carrying out teammates' duties through active participation in each member's role".[11] The goal of positional rotation is to provide the individual with hands-on knowledge about the roles and responsibilities of other teammates, with the purpose of improving interrole knowledge and team performance.

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 at each planstep.

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.

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 $T(s'|s,a)$ are initialized from prior knowledge about the task. In Line 2, an initial policy π is calculated for the robot using value iteration.[16] 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 currenct policy π, and the observed state and action sequence is recorded. In Line 5, $T(s'|s,a)$ is updated based the observed state-action sequence. $T(s'|s,a)$ describes the probability that for a state s, and action a, the human will perform an action such that the next state is s'. At the same time, we update the probabilities of the transitions $T(\hat{s}'|s,\hat{a})$, where $\hat{s}' \neq s'$ and $\hat{a} \neq a$, that will occur for the same human action as the one observed in the transition $T(s'|s,a)$. The underlying assumption is that human actions depend only on current state s and not on the current robot action a, and also that the human and robot actions are not mutually exclusive. All transition probabilities as described above are given by multinomial distributions and are estimated by the transition frequencies, assuming a pre-observation count.[7] The pre-observation count corresponds to the size of a real or imaginary sample-set from which we built the MDP prior to the training process. It is a measure of the confidence we have on how close the 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, for each observed state s and human action a, the rewards are updated as follows:

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

Note that we can change the value of the constant c depending on how fast we want the robot to converge to the mental model of the human. In the extreme cases, a very small value will mean that human actions in the rotation phase will have small impact on the robot's policy in the forward phase, and a very large value will mean that the robot will imitate the human immediately after the first iteration.We then use the new estimated values for $R(s,a)$ and $T(s'|s,a)$ to update the current policy (Line 8). The new optimal policy given these values is computed using standard dynamic programming techniques.[1] We iterate the forward and rotation phases for a fixed number of M iterations, or until a convergence criterion is met, such as the difference between the human and robot action sequences across two consecutive iterations.

2. Forward phase

The forward phase is given by Figure 2. In Line 1, the current state is initialized to the start task-step. In Line 3, the robot executes an action a assigned to a state s, based on the current policy π. The human action is recorded (Line 4) and the *next_state* variable is set according to the *current_state*, the robot action a and the human action. This requires the use of 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 and action of the current time-step are recorded (Line 6), and the variable *current_state* is updated for the next iteration. The algorithm terminates when *current_state* is set to FINAL_STATE, which signals the end of the training round.

```
Algorithm : Cross-training
1.  Initialize R(s, a) and T(s' | s, a) from prior knowledge
2.  Calculate initial policy π
3.  while(number of iterations < MAX)
4.      Call Forward-phase(π)
5.      Update T(s' | s, a) from observed sequence s_1, a_1, s_2, a_2, .... , s_M, a_M
6.      Call Rotation-phase()
7.      Set R(s_i, a_i) = R(s_i, a_i) + c for observed sequence s_1, a_1, s_2, a_2, .... , s_N, a_N
8.      Calculate new policy π
9.  end while
```

Figure 1. Human-Robot Cross-training algorithm

```
Function: Forward-phase(policy π)
Set current_state = START_STATE
while(current_state != FINAL_STATE)
        Execute robot action a according to current policy π
        Observe human action
        Set next_state to the state resulting from current_state, robot and human action
        Record current_state, a
        current_state = next_state
end while
```

Figure 2. *Forward phase* of the Cross-training Algorithm

3. Rotation phase

The rotation phase is given by Figure 3. In Line 3, the action a is set according to the observed human action. In Line 4, robot action is sampled from the transition probability distribution $T(s'|s,a)$. We assume that for a state s and an action a there is only one action that leads to state s'. Just as the transition probability distributions of the MDP are updated in the forward phase, the robot policy is updated to match the human expectations in the rotation phase. This process emulates how a human mental model would change by working together with a partner. We also expect the human to learn the role of the robot and gradually adapt his or her own actions during the training process.

C. Interactive Reinforcement Learning

We compare the proposed formulation to the standard interactive reinforcement learning approach, where the reward signal is determined by a real-time interaction with a human teacher or coach.[16,21] The standard reinforcement learning algorithm we used is illustrated in Figure 4. In contrast to Cross-training algorithm (Figure 1), there is no rotation phase. After each training round, the rewards of observed states are updated from human reward inputs (Line 6), and a new policy is calculated (Line 7). To make the comparison valid between the two algorithms, we update the rewards and recalculate the policy after each training round, rather than after each robot action. Furthermore, we allow the human reward r to take values of $\{0, +c, -c\}$, corresponding to a neutral, good or bad reward. We use the same constant value c for the reward update at Line 7 of the Cross-training algorithm in Figure 1.

```
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(next_state | current_state, a)
5.         Record current_state, a
6.         current_state = next_state
7. end while
```

Figure 3. ***Rotation phase* of the Cross-training algorithm. In line 4, we assume that there is only a unique robot action that can lead to state $next\_state$ from $current\_state$ and action a.**

```
Algorithm : Standard Reinforcement Learning with Human Reward Assignment
Initialize R(s, a) and T(s' | s, a) from prior knowledge
Calculate initial policy π
while(number of iterations < MAX)
    Call Forward-phase-with-human-reward-inputs (π)
    Update T(s' | s, a) from observed state and action sequence s_1, a_1, s_2, a_2, .... , s_M, a_M
    Update R(s_i, a_i) = R(s_i, a_i) + r_i for observed state, action and human rewards s_1, a_1, r_1,
    s_2, a_2, r_2 , .... , s_N, a_N, r_N
    Calculate new policy π
end while
```

Figure 4. Standard reinforcement learning with human reward assignment.

D. Evaluation of mental model convergence

For a given policy π, a Markov Decision Process is reduced to a Markov chain.[4] To evaluate the convergence of human and robot mental model, we assume a uniform prior, and evaluate the *entropy rate*[6] of the Markov chain. For a finite state Markov chain $\mathtt{X}$ with initial state s_0 and transition probability matrix $\mathtt{T}$ the entropy rate is always well defined.[8] It is equal to the sum of the entropies of the transition probabilities $T(s'|s,\pi(s))$, for all $s' \in S$, weighted by the probability of occurrence of each state according to the stationary distribution μ of the chain (Equation 3). Given the transition proability distributions $T(s'|s,\pi(s))$, the μ_s can be estimated using Doeblin's Basic Theorem.[20]

$$H(\mathtt{X}) = -\sum_{s\in S}\mu_s \sum_{s'\in S} T(s'|s,\pi(s)) log\left[T(s'|s,\pi(s))\right]. \tag{3}$$

We choose conditional entropy (Eq. 3) as our metric to evaluate convergence of the robot's computational teaming model. As the mental models of 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 transition probability distributions that describe the stochastic transitions will increasingly reflect the human's preferences over action selection. Note that in our computational teaming model, the conditional entropy, given by Eq. 3, represents the robot's uncertainty about the human's action selection. As a result, we would expect the measure of conditional entropy in the Markov chain to decrease as the human and robot train together.

```
Function: Forward-phase-with-human-reward-inputs(policy π)
Set current_state = START_STATE
while(current_state != FINAL_STATE)
        Execute robot action a according to current policy π
        Observe human action
        Prompt human to enter reward r
        Set next_state to the state resulting from current_state, robot and human action
        Record current_state, a, r
        current_state = next_state
end while
```

Figure 5. ***Forward phase*** **of standard reinforcement learning with human reward assignments.**

V. User Study

A. A place-and-drill task

We apply the proposed framework to train a team of one human and one robot to peform a simple task in a virtual environment. The virtual environment, presented in Figure 6, was built in RobotStudio.[14]

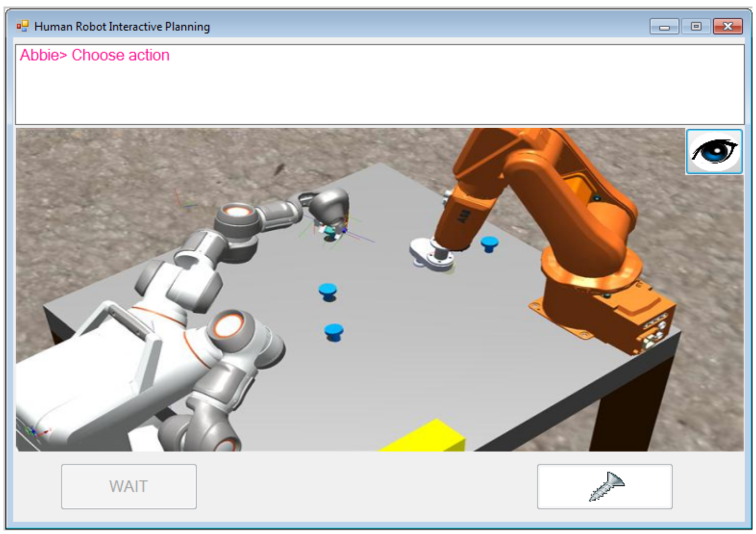

Figure 6. Human-Robot Interactive Planning using ABB RobotStudio virtual environment.

The human and robot team trained together to execute a place-and-drill task. The human's role was to place screws in one of three available positions. The robot's role was to drill each screw. Although this task is simple, we found it adequate for initial testing of our framework, as different persons have different preferences for accomplishing the task. For example, some participants preferred to place all screws before drilling any. Other participants preferred to place and drill each screw before moving on to the next.

B. User study setting

The participants consisted of 20 subjects recruited from MIT. The participants were asked to train with the Abbie, a small ABB industrial robot, using the virtual simulator. Each human subject participated in two training sessions:

1. Cross-training session: The participant iteratively switches positions with the virtual robot, placing the screws at the forward phase and drilling at the rotation phase.

2. Interactive Reinforcement Learning session (Interactive - RL): This is the traditional 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.[7]

We fix the number of training iterations to three for both sessions, with a final *execution round* where the human and robot execute the task in the forward phase using the learnt teaming model. All participants were provided a five-minute tutorial on how to use the virtual environment interface. Participants were divided into two groups: Group 1 participants were asked to write down beforehand how they would like to execute the place-and-drill task with the robot, whereas Group 2 started directly with the training. The order of the two sessions was randomly chosen for all participants to mitigate learning effects.

C. Results

1. *Entropy rate*

As described above, we evaluate the human-robot mental model convergence by calculating the entropy rate (Eq. 3) of the Markov Decision Process at each iteration using the robot policy computed for that iteration. We compare the entropy values, mean and standard deviation, for each iteration from the two training sessions in Figure 7. Results show that the conditional entropy of the computational teaming model converges faster through Cross-Training than through the traditional Interactive-RL approach. This means that the robot's uncertainty in the human participant's actions decreases faster with Cross-Training. Two-tailed within-subjects t-test (alpha = 0.05) indicates a statistically significant decrease (p=0.02) in entropy values at the last iteration for Cross-Training as compared to the last iteration for Interactive-RL. Participant group and the order of the training sessions did not produce statistically significant differences in the entropy values at each iteration.

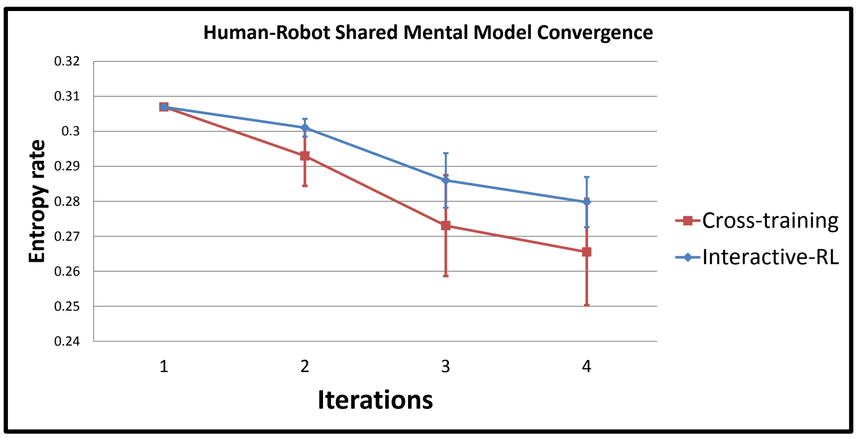

Figure 7. Human-Robot Shared Mental Model convergence, expressed as the entropy rate of the Markov Decision Process.

2. *Human perspective*

To gather information about the participant's subject experience, we asked each person to rate his or her agreement with the statement "In this round, the robot performed its role well according to my preference" on a 1-5 Likert scale, 1 for strongly disagree and 5 for strongly agree. Participants were prompted to respond to this statement after each of the three training rounds. After the final execution round each participant was asked to rate his or her agreement with the statements presented in Table 1 on a 1-5 Likert scale. The Likert questionnaire, similar to those used in previous research,[18] addressed the robot's performance, the robot's contribution to the team, its perception of the human preferences, shared goals, team fluency, and trust in the robot (Table 1).

Figure 8 presents the the averaged participant response to the statement "In this round, the robot performed is role well according to my preference". Two-tailed paired Wilcoxon-Mann-Whitney tests (alpha=0.05) found the difference of the values for the last training round to be statistically significant (p =

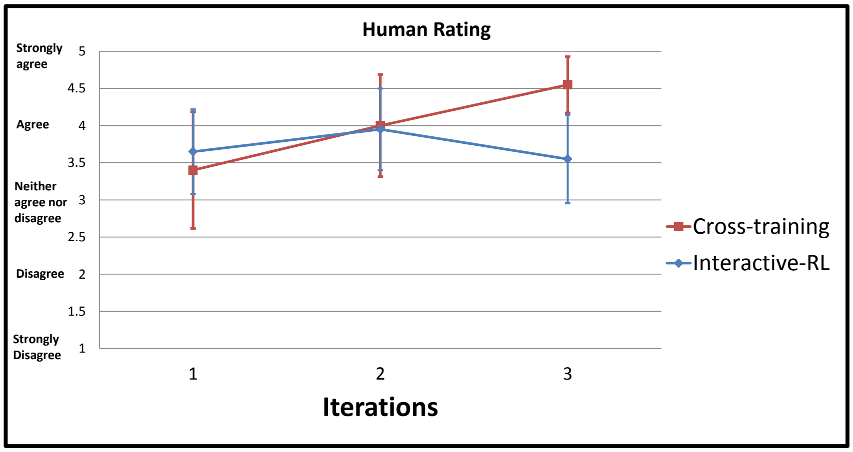

Figure 8. Average rating of the agreement of human subjects with the statement, "In this round, the robot performed its role well acording to my preference".

Table 1. Likert Questionnaire

1.	Abbie's performance was important for the success of the team
2.	Abbie performed well as part of the team
3.	Abbie perceives accurately what my preferences are
4.	The team worked fluently together
5.	Abbie does not understand how I am trying to execute the task
6.	Abbie and I are working towards mutually agreed upon goals
7.	Abbie was trustworthy

0.01). No statistically significant differences were found with respect to the participant groups or ordering of the sessions. Also, in most of the statements in Table 1, the answers for the cross-training session were moderately better, but no statistically significant differences were found.

It is interesting that for the statement #1 in Table 1, "Abbie's performance was important for the success of the team", both the participants' group and session order resulted in statistically significantly different answers for the Cross-training session. In particular, a Wilcoxon-Mann-Whitney test (alpha = 0.05) showed that participants who were asked after the Cross-training session to document their preferences for task execution beforehand (Group 1) agreed more strongly to the statement (p =0.01-0.03).

Group of the participants also played an important role for the statement #6, "Abbie and I are working towards mutually agreed upon goals", when the participants were asked to rate the statement after the Cross-training session. Group 1 participants agreed more strongly to the statement (p = 0.02).

3. Discussion

The results presented support hypothesis 1 that positional-rotation cross-training within a human-robot team yields a statistically significant improvement in an objective measure of the robot's uncertainty about the human partner's action sequences (p = 0.02), when compared to the transtional ask-for-reward approach. Analysis also indicates that participants agreed more strongly that "the robot performed well according to their preference" after the last round of cross-training, compared to after the last round of the standard reinforcement learning algorithm (p = 0.01). No statistically significant results were found for the Likert Questionnaire. Considering previously reported results,[18] it is surprising that participants did not agree more strongly with the statement that "Abbie is trustworthy" after the cross-training phase. One possible explanation is that working with an actual robot is necessary for significant differences in trust.

VI. Conclusion

In this study, we described the design and evaluation of a human-robot interactive-training phase, which emulates positional-rotation cross-training applied in human teams. The main strength of the proposed framework is an iterative training process which enables mutual co-adaptation of a human-robot team. We have evaluated human-robot cross-training in human subject experiments, in which a person works with a robot in a virtual environment to collaboratively perform a place-and-drill task, and compared it with the standard reinforcement learning approach. We show that cross-training promotes predictable patterns of human and robot action sequences, and reduces the robot's uncertainty on human actions ($p = 0.02$). Furthermore, participants trained with the proposed framework agreed more strongly that the robot performed its role according to their preference ($p = 0.01$).

The next step in this study would be to test how cross-training in the virtual environment affects the team performance of a human working with an actual industrial robot. It is also of interest to compare the proposed algorithm with the most recent state-of-the-art interactive reinforcement learning algorithms.

Acknowledgments

This work is supported in part by ABB, and is being conducted in collaboration with Thomas Fuhlbrigge, Gregory Rossano, Carlos Martinez, and Biao Zhang of ABB Inc., USCRC - Mechatronics. We would also like to acknowledge the Onassis Foundation as a sponsor, and thank Mark Gabriel of the Boeing Company for sharing his insight into assembly manufacturing applications.

References

[1]D. P. Bertsekas. *Dynamic Programming and Optimal Control, Vol. II, 2nd Ed.* Athena Scientific, Belmont, MA, 2001.

[2]Cannon-Bowers J.A. Blickensderfer E. and Salas E. Cross-training and team performance. *Making decisions under stress: Implications for individual and team trainin, American Psychological Association*, pages 299–311, 1998.

[3]Blickensderfer E. Bowers 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*, pages 92–101, 1998.

[4]C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems.* Springer, 2008.

[5]Nancy J. Cooke, Eduardo Salas, and Janis A. Cannon-bowers. Human factors, 42, 151-173. measuring team knowledge.

[6]Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory.* Wiley Series in Telecommunications, 1991.

[7]Finale Doshi and Nicholas Roy. Efficient model learning for dialog management. In *Proceedings of Human-Robot Interaction (HRI 2007)*, Washington, DC, March 2007.

[8]L. Ekroot and T.M. Cover. The entropy of markov trajectories. *Information Theory, IEEE Transactions on*, 39(4):1418 –1421, jul 1993.

[9]Janice Langan-Fox, Sharon Code, and Kim Langfield-Smith. Team mental models: Techniques, methods, and analytic approaches. *Human Factors*, 42(2):242–271, 2000.

[10]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. *Journal of Applied Psychology*, 85(6):971–986, 2000.

[11]M.A. Marks, M.J. Sabella, C.S. Burke, and S.J. Zaccaro. The impact of cross-training on team effectiveness. *J Appl Psychol*, 87(1):3–13, 2002.

[12]John E. Mathieu, Tonia S. Heffner, Gerald F. Goodwin, Janis A. Cannon-Bowers, and Eduardo Salas. Scaling the quality of teammates' mental models: equifinality and normative comparisons. *Journal of Organizational Behavior*, 26(1):37–56, 2005.

[13]John E. Mathieu, Tonia S. Heffner, Gerald F. Goodwin, Eduardo Salas, and Janis A. Cannon-Bowers. The influence of shared mental models on team process and performance. *Journal of Applied Psychology*, 85(2):273–283, 2000.

[14]ABB RobotStudio. http://www.abb.com.

[15]A.L. Rowe, N. Cooke, E. Hall, and T. Halgren. Toward an on-line knowledge assessment methodology: Building on the relationship between knowing and doing. *Journal of Experimental Psychology: Applied*, 2:31–47, 1996.

[16]Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Pearson Education, 2003.

[17]J. Shah and C. Breazeal. An empirical analysis of team coordination behaviors and action planning with application to human-robot teaming. *Human Factors*, 52, 2010.

[18]Julie Shah, James Wiken, Brian Williams, and Cynthia Breazeal. Improved human-robot team performance using chaski, a human-inspired plan execution system. In *Proceedings of the 6th international conference on Human-robot interaction*, HRI '11, pages 29–36, New York, NY, USA, 2011. ACM.

[19]R. Stout, J. Cannon-Bowers, E. Salas, and D. Milanovich. Planning, shared mental models, and coordinated performance: an empirical link established. *Human Factors*, 41:61–71, 1999.

[20]Daniel W. Stroock. *An Introduction to Markov Processes.* Springer, 2005.

[21]Andrea Lockerd Thomaz, Guy Hoffman, and Cynthia Breazeal. C.: Real-time interactive reinforcement learning for robots. In *In: Proc. of AAAI Workshop on Human Comprehensible Machine Learning*, 2005.

[22]C. Volpe, J. Cannon-Bowers, E. Salas, and P. Spector. The impact of cross training on team functioning. *Human Factors*, 38:87–100, 1996.