Improving Implicit Communication In Mixed Human-Robot Teams With Social Force Detection

Bradley Hayes
Yale University
Department of Computer Science
bradley.h.hayes@yale.edu

Brian Scassellati
Yale University
Department of Computer Science
scaz@cs.yale.edu

***Abstract*—One of the hallmarks of development is the transition of an agent from novice learner to able partner to experienced instructor. While most machine learning approaches focus on the first transition, we are interested in building an effective learning and development system that allows for the complete range of transitions to occur. In this paper, we present a mechanism enabling such transitions within the context of collaborative social tasks. We present a cooperative robot system capable of learning a hierarchical task execution from an experienced human user, collaborating safely with a knowledgeable human peer, and instructing a novice user based on the explicit inclusion of a feature within the planning and skill execution subsystems we've termed social force. We conclude with an evaluation of this feature's flexibility within a collaborative construction task, changing a robot's behaviors between student, peer, and instructor through simple manipulations of this feature's treatment within the planning subsystem.**

I. Introduction

Most robotics learning systems focus on developing a robot to be either a student or a teacher. In robot-learner applications, a robot is tasked with acquiring a skill from some collection of input signals, examples of which include human demonstrations, simulations, or solutions to complex sets of constraints. In robot-instructor applications, the robot focuses on transferring a set of knowledge to another agent, measuring competency and modulating the pace of knowledge transfer. These two domains are often treated as unrelated, with little crossover between communities. Between the two exists the relatively new field of socially collaborative robotics, a distinct but related research domain focusing on human-robot teaming and interactions. Collaborative robotics is interested in robots as capable learners, peers, and instructors, examining both the roles themselves and transitions between them. As robots transition out of isolation into roles where collaboration with humans or other robots is possible or even required, a focused effort must be made to engineer systems that can accommodate these complex requirements.

A multitude of challenging research problems must be addressed to enable fluent interactions between humans and robot agents [1]. Facilitating communication between humans and robots constitutes a substantial portion of these issues. Human teams constantly communicate explicitly and implicitly over a variety of verbal and non-verbal channels. Humans are capable of leveraging complex intention recognition capabilities in real-time, developing and updating models of their co-workers as they gain experience collaborating with or observing them. Further, we are able to infer a great deal about complex interactions between others merely through observations. This holds true at a distance when the actors are reduced to low fidelity representations of themselves, even in the absence of explicit context. While intention is tremendously complicated, interpreting basic motion is not, requiring very little from one's environment to provide context. This irrepressible, constantly utilized ability to infer intention from motion develops quickly in children around 9 months of age [2]. The limited processing requirement combined with the vast potential for understanding agent behaviors makes this an extremely attractive phenomenon to interpret and incorporate into collaborative robot systems.

Though people share a host of available avenues for communication, synchronizing mental models remains a challenging task for human teams. It has been demonstrated that increased implicit communication directly improves the performance of teams under stresses caused by temporal constraints or uncertainty, by enabling members to act in anticipation of teammates' actions [3]. To build robot systems capable of natural collaboration with humans, a targeted engineering effort and planned interaction methodology must be explored and developed regarding intentionality detection and expression. Additionally, integrating teammate-like behaviors into collaborative systems may dissuade notions of robots being unsuitable or unready for certain application domains.

Beyond the difficulties inherent to developing cooperative systems, complexities within skill and task representation further complicate the design and production of socially capable, collaborative robots. While humans often have little trouble verbalizing their understanding of a skill or task, robot systems' internal representations often do not afford such transparency. Designers must architect socially oriented solutions to the problems of skill acquisition, role definition, role selection, and action selection. Hierarchical Learning (HL) has yielded substantial success relating to the challenges of task representation, allowing for the scaling of skill complexity by dividing intricate tasks into collections of simpler, related sub-skills [4]. Concepts from HL have also been leveraged within individual skill acquisition to improve learning from demonstration and skill transfer between similar environments [5]. The weakness inherent to such HL techniques is the lack of explicit specification for human-accessible representations of information.

Addressing the challenges inherent to skill acquisition, task representation, and role selection introduced by building a collaborative system requires explicitly architected, socially

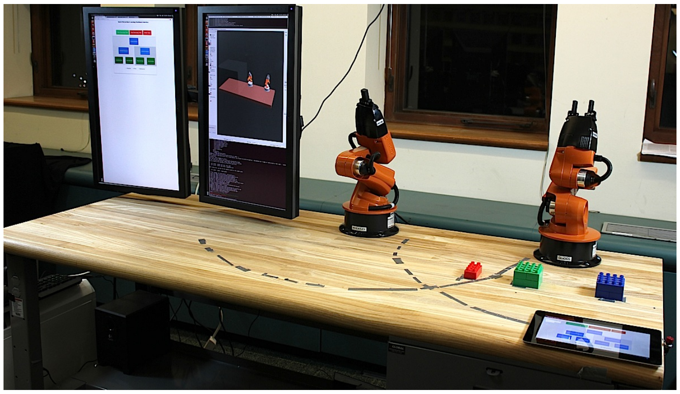

Fig. 1. The Collaborative Workbench used in the proof-of-concept implementation. The left monitor mirrors the tablet's display, showing live information about the task tree throughout task execution and allowing human participants to claim roles within the system. The right monitor displays a simulated environment mirroring the real world, along with social force fields as they are generated from user motion.

directed features within these core behaviors. We define Social Hierarchical Learning (SHL) as an extension to HL designed to facilitate the development of systems that can flexibly acquire skills from multiple sources, generalize these skills to cooperative tasks, and execute a complex plan collaboratively as part of a mixed human-robot team while maintaining transparency to its peers [6]. Our work is presented as a component that offers much benefit to an SHL-like system.

Within this paper we examine the utility of a feature we describe as *social force*, a projection of external agents' anticipated movements. Social force can carry vastly different meanings depending on the context in which it is applied. As an example of social force expression, consider two humans A and B selecting objects from a table. If A and B simultaneously reach for an object from the same region, the reaching action of A may cause B to select objects from a different region. The likelihood of a change in B's selection region depends on a number of factors, including the magnitude of repulsive social force expressed by A's action and B's commitment to the initially selected area. However, if A points towards an object and communicates intent for B to select it, the magnitude of the attractive social force expressed by A's gesture (modulated by insistence/sharpness of motion) and B's commitment to the initial selection will determine if B follows A's suggestion. We have integrated this feature into the skill execution and planning subsystems of a kinesthetically trainable, collaborative robotic workbench with two robot arms.

II. Background and Related Work

Skill acquisition within state of the art robotics systems can be accomplished through a variety of methods ranging from methods accessible to true novice users with no robotics or programming experience to methods requiring expert roboticists intricately familiar with the hardware, software, and sensor suites available to the robot. Techniques such as learning from demonstration (LfD) allow subject matter experts who may be novice robotics users to impart knowledge and abilities into robot systems [7]. These methods work exceptionally well for collaborative applications, as the robot's teammates may not have robotics or programming expertise. LfD is used frequently for human subjects experiments, providing valuable feedback to interaction designers [8]–[10]. Within LfD, kinesthetic teaching has shown exceptional utility as a method of learning skills from novices [11]–[14]. By physically guiding a robot to perform a skill, users are capable of directly imparting knowledge to a robot through example skill executions driven via a physical, intuitive interface. Other advantages of this method include no requirement of external sensing equipment, no danger of violating kinematic limitations of the robot, and minimal danger of unexpected self-collisions [15].

Once individual skills are learned, one must focus on task-level completion. Fluency of collaboration can only be attained once members of a team have a consistent mental model of the task to be completed. This mental model must include information relating orderings of skills, the mechanics of each skill's execution, and skill groupings that make up roles that can be assumed. Hierarchical task representation offers a convenient way to condense large quantities of information, exposing the proper level of sophistication required for the requesting agent or process. There exist several algorithms capable of decomposing tasks into representations suited for autonomous agents [4], [5], [16]–[19], but the addition of human co-workers introduces requirements for human-readable and human-communicable task representations [20], [21]. Recent contributions in this area have yielded human-inspired methods of action segmentation [22]. Statistical regularities in observed action sequences can be used to discern meaningful segmentations by identifying boundaries between probable action groupings. Other means of introducing shared mental models include exposing an interface by which a human can share his or her mental model of the task with a robot, overriding any existing representation [23]–[25]. Converging towards common mental representations of tasks represents significant progress towards fluent human-robot teaming, and can potentially simplify the task of interpreting the intentions broadcast by one's peers.

Once a common model of the task's components has been established, the next challenge facing a team lies in learning individual members' preferences for execution. Given potentially parallel branches of subtasks, it is important that teammates have accurate models of each other's expectations. Adaptive planning systems capable of modeling human behaviors with social understanding are beginning to emerge and face testing in real-world environments. Much active work remains in both the area of socially-oriented adaptive planners and that of addressing unfulfilled team commitments or execution-time variation in ordering constraint fulfillment [26], [27]. One effective method of learning teammate preferences is cross-training, a technique in human-human training in which participants trade roles. This approach has shown some promise when applied to human-robot teams, improving both the comfort of the human operator and the performance of the team [28].

One particularly promising avenue of exploration for building solutions to these collaboration-centric problems is intention detection and classification. In this domain, the unseen social forces that agents exert upon one another have been shown capable of yielding rich social, cultural, and intentional information [29]–[31]. It's been shown that details about inter-agent relationships and causal relationships between agents and

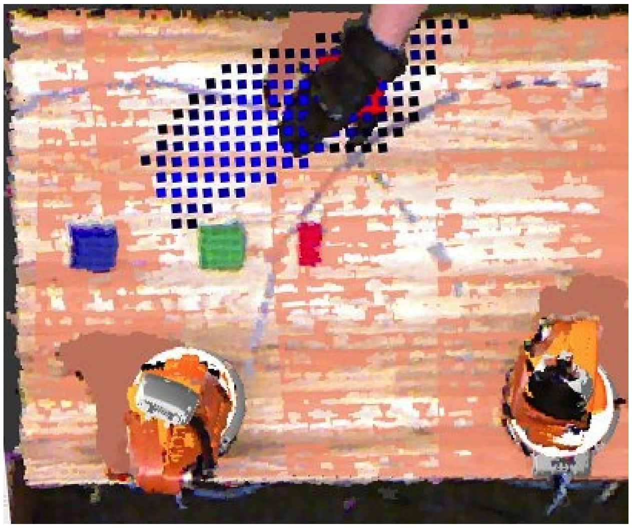

Fig. 2. Realtime visualization of a social force particle field generated from the user's motion patterns, rendered over point cloud data from the workbench sensors. Social force magnitude is represented by the blue channel, with the intensity of force increasing as particles range from black to blue. The estimated position of the user's hand is represented by a red square, shown under the gloved hand.

the world can be deduced from observations of impoverished and low-context trajectory data [32]. While social forces have been used in the past within perceptual tasks both to understand the actions of agents as intentional and to classify agency, we integrate social force understanding into a collaborative robot's planner to provide it the ability to dynamically transition between the roles of student, collaborator, and teacher.

III. Approach

Our system is implemented as a minimalist component of a larger SHL system, seeking to explore the addition of socially oriented algorithms to traditional Hierarchical Learning. The primary motivations behind SHL systems are to enable flexible skill acquisition from non-experts, generalize skills to social, cooperative tasks, and be able to collaboratively execute these skills within the context of complex tasks as part of a mixed human-robot team. We are also interested in facilitating bi-directional skill transfer within SHL. Many learning systems primarily focus on methods of inputting skills, but truly collaborative systems must also be able to effectively communicate learned knowledge to other agents.

A. Robot System

We have implemented our collaborative task execution proof-of-concept on our Collaborative Workbench (Figure 1), a 1.83m x 0.762m worktable equipped with two KUKA YouBot 5-DoF light manufacturing arms spaced 0.5m apart, with approximately 0.5m of overlapping operational envelope. Each arm is equipped with a 2-DoF gripper. Sensing is performed through registered point cloud data, captured via a Microsoft Kinect mounted above the workspace.

B. Software Architecture

The collaborative system we implemented to demonstrate social force is built in the Robot Operating System (ROS) software framework. Core data structures, such as skills, task trees, and robot controller interfaces are incorporated into a static software library. Other necessary, live components such as the social force publishing service, kinesthetic skill trainer, hand tracker, and cooperative execution program are implemented as separate ROS nodes. Following this type of module-oriented design pattern within ROS allows for near-trivial portability between application domains. User interfaces are implemented in HTML and Javascript, utilizing the ROS-Bridge web interface to allow for device portability, effective visualization, and rapid prototyping.

C. Kinesthetic Teaching

Skill acquisition within our system is accomplished via demonstration, through kinesthetic teaching. Kinesthetic teaching is a process by which a user physically manipulates the learner through an execution of the target action. While there are many valid ways of performing kinesthetic teaching, each with various strengths and weaknesses depending on the target domain, our system is based upon keyframing as opposed to pure trajectory-based recording. Keyframing is a process by which actions are recorded as a sparse set of important animation frames. Playing keyframes back in sequence can be used to recreate the skill as demonstrated. Individual trained skills within SHL are intended to be primitive action components that may be combined to achieve high-level functionality.

To train a skill within our system, a user must explicitly engage the robot into training mode through the provided web interface. Once training has started, the user is free to pose the robot to the desired position and set keyframes as necessary. Features to be tracked (e.g., motor positions, gripper distance to object, etc.) are indicated by the user prior to training, rather than learned during training. Keyframed state feature vectors are recorded as vertices within a skill's state graph, initially linked by edges reinforced from the explicit training example. As skills are executed, more densely populated paths are created, as explored states are recorded and linked. We explicitly simplify this component of the system to retain the minimally viable interactive feature subset to illustrate the flexibility and magnitude of effect of social force as a feature. Additionally, only a linear interpolation exploration function was used during skill execution, leaving the explicit user-trained paths as the only available options for the system to follow.

D. Task Structure and Execution

Complex tasks are represented as a hierarchical collection of primitive skills with varying goals and pre-/post-conditions. A task tree is a generalized representation of a complex task sequence. Branches represent divisions of labor along potentially parallel paths of execution, each relating to a possible role to be fulfilled by a participating agent. As tree nodes are claimed by human or robot agents for execution, all child nodes are also claimed for the agent. Ownership of subtasks can be released, returning the unfinished and newly unclaimed skills back to the available work pool. The task tree is managed by a process independent from working agents and is responsible for managing ownership requests and status updates. A web interface provides users of our system a visualization of the

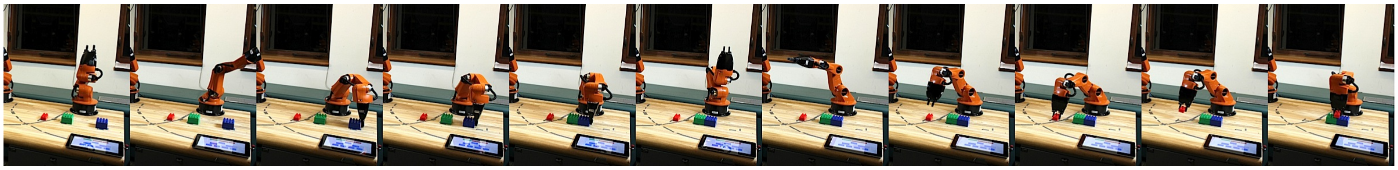

Fig. 3. Full demonstration of the task by the robotic agent. When multiple agents participate, it is possible to execute the subtask that joins the blue and green bases and the subtask that moves the red block into position in parallel.

task tree, including information such as branch ownership and completion status.

The collaborative execution of a task is accomplished through agents claiming ownership of subtasks from the task tree and executing the associated skills. It is the responsibility of each agent's planner to determine which subtasks to request ownership of and fulfill. For our tests, we utilized a simple planning algorithm that would choose the most promising role according to a social force value associated with each subtask at the time of the decision, fulfill it, and query the task tree for additional roles. An agent only claimed one branch of subtasks at a time. In the absence of social force measurements or in case of ties, decisions would be made randomly amongst candidate options. Throughout execution, agents would evaluate unclaimed roles against their chosen role, aborting their choice if the difference in desirability (measured solely through social force) between an unclaimed role and the currently assumed role passed a predetermined threshold. If no roles are unclaimed and the chosen role is evaluated to have low desirability beneath a predetermined threshold, the agent would halt its actions, continuing once the role's desirability increased above the threshold again.

IV. Social Force

Enabling collaboration between humans and robots is a major research challenge within the robotics community. Collaborative robots must be capable of learning co-worker role selection preferences in addition to modeling optimal role selection choices for themselves. Scenarios where multiple agents interact within a shared environment increase the likelihood of collisions between workers, both in terms of physical space occupancy and resource requirements. Collaborators must also be able to adapt to new situations, ranging from changes in skills required to perform one's duties to training new teammates. These high-level abilities all require some form of intention modeling to be effective.

One feature that we demonstrate as capable of assisting in fulfilling these requirements is that of social force. We define social force as a projection of one's intention through movement. While physical force generates straightforward behavior when applied to most agents, social force affects each agent differently and can be context-sensitive depending on the skill being executed. The social force exerted by an agent at a point in space can be computed given a model of the agent along with its motion history. In the absence of a kinematic model, the end effector position in space may be used.

Given a series of samples S of the target agent's end effector positions, one can compute an ellipsoid representative of likely future positions of the agent as a projection of anticipated motion. For work performed on the Collaborative Workbench, we used a 2D projection of the samples (Figure 2), as including the z-axis did not offer any accuracy or responsiveness gains for its computational cost. We define a temporally bounded set of positions $P = \{p : p \in S_{[t_{now}-t_{duration}, t_{now}]}\}$. For our experiments, we empirically set $t_{duration} = 0.75s$. Adjusting $t_{duration}$ changes the sensitivity of the social force projection, with increasing values softening the effect of sharp motions and decreasing values increasing their effect.

We obtain the covariance matrix of our samples

$$M = \begin{bmatrix} cov(P_x, P_x) & cov(P_x, P_y) \\ cov(P_y, P_x) & cov(P_y, P_y) \end{bmatrix}$$

which is guaranteed to be a real, symmetric matrix (for any dimensionality sample set). Given this guarantee, we can obtain real, orthogonal eigenvectors describing the principal components of $\bar{P}$ through the eigendecomposition $D = VMV^T$, extracting eigenvectors from columns of V and eigenvalues from the diagonal matrix D. Using the eigenvectors as axes and eigenvalues as axis lengths, we can construct an ellipse modeling the second order moments of the sample, providing a rough description of the data's shape and orientation while remaining robust against outliers from sensor fluctuations. Finally, we translate the ellipse's center by a value c representative of the inertia of the latest sampled position $p \in P$, given by $c = p - \bar{P}$.

Once the social force ellipse is defined, determining the social force value for a given point is a fast operation. A transformation matrix T can be constructed from scaling, rotating, then translating the unit circle into the ellipse described by the parameters extracted from the steps above. Given a test point x, one need only apply the transformation $x' = T^{-1}x$ and test for the presence of x' within the unit circle. The distance of x' from the origin along each axis may be used to scale the social force value assigned to x.

Simple changes in treatment of social force within a robot's action planning algorithm can be used to dramatically alter its behavior between that of student, peer, and instructor. The robot can use generated social force fields to evaluate potential skill or role choices. By obtaining readings of social force at the various locations its end effector will travel through at keyframes within the candidate skills, the robot can use social force to affect its skill selection decisions.

To illustrate the effectiveness of social force as a transformative feature within human-robot collaboration, we use a simple construction task as a test domain (Figure 3). Three blocks are set on the worktable, a red top piece, a green base piece, and a blue base piece. From the initial setup, the goal is to join the two base pieces with a single top piece in the middle of the working area. The task tree consists of two branches, one to move the blue base next to the green base and one to place the red piece on top of the joined bases. Each branch consists of three subtasks: a 'grasp' or 'locate' action, a 'place' action, and a 'return to resting pose' action. The 'return to resting pose' action is not displayed in the task tree visualization figures. The first actions of each branch can be done in parallel, but the red piece cannot be placed until the base pieces are moved together. This construction task constitutes the simplest possible task formulation that allows for meaningful expression of all modes of operation that we wish to demonstrate.

1) Robot as Student: If social force from a particular agent is treated as an attractive, positive force within the action planning system, the robot will behave as a learner, as if taking cues from an instructor (Figure 4). As the designated instructor gestures towards particular areas of the workspace, the generated social force will positively weight any decisions involving skills passing through that space. By enforcing a minimum, non-zero value of social force required before making a role or skill choice, one can ensure that the robot will remain inactive unless instructed towards a particular choice. The order of actions taken from this behavior can be used to learn task-level preferred action orderings that may not be obvious given data inherent to the skills in question (e.g., it's typically better to pour the cereal into the bowl before the milk).

During the construction task, the robot evaluates its skill choices within the task tree and chooses the skill that best fits the working area covered by the social force of the instructor. As the instructor motions towards the blue base block, the robot performs its locate blue block skill, moving its end effector to the side of the block, ready to push it. Once the instructor gestures towards the green block, the robot executes its push blue block skill, moving the block to its desired position. Similarly, if the instructor gestures towards the red block first, the robot will execute its pick up red block skill. Once the instructor moves the blue block to its final position, social force exerted near the newly combined base region will trigger the robot to execute its place red block action.

2) Robot as Collaborator: When social force is treated as a negative, repulsive force within the action planning system, the robot will behave as a peer, choosing subtasks that can be fulfilled in parallel with other agents while minimizing collisions (Figure 5). As other agents begin to execute their roles, the robot will initially choose from the minimally conflicting options. During execution, the robot becomes capable of reacting to invasions to its workspace, choosing to adopt a different role when it becomes apparent through other agents' social forces that it will be interrupted.

Throughout the construction task, the robot chooses to execute parallel, non-conflicting subtasks in concert with the agent it is working with. As an agent begins to perform an action, the robot is biased against choosing spatially similar actions to execute. If a robot has already chosen a particular task, but detects negative social force due to a conflicting task choice by another agent (e.g., human co-worker) operating in the same space, the robot aborts its selection and chooses a different, less conflicting action. This behavior may also lead to instances of turn-taking, which while typically avoided in collaborative exercises, can lead to safer operation in confined workspaces.

3) Robot as Instructor: If social force is used as a trigger once its value passes a particular threshold, the robot can behave as an instructor, indicating which skills it desires its student to complete and in which order (Figure 6). Leveraging the keyframing within the trained skills, a robotic instructor can step through the skill in stages until the social force trigger criterion is met. More formally, given a skill A consisting of keyframes $\mathrm{kf}_1, \mathrm{kf}_2, ..., \mathrm{kf}_n \in A$, the robot will execute keyframes in the set $K = \{\mathrm{kf}_i \in A : i \leq \text{(\# times demonstration has been repeated)}, \mathrm{kf}_i \notin \mathrm{GoalStates}(A)\}$. The robot first executes keyframes $k \in K$ in increasing order from $k_1, k_2, .., k_{|K|}$, then reverts to its start state by executing keyframes $k \in K$ in reverse order from $k_{|K|}, k_{|K|-1}, ..., k_1$. This staged execution has the benefit of progressively revealing more of the desired action while leveraging existing information to do so. Once the student begins to perform the desired action, the trigger criterion will be met and the robot is able to move on to the next desired skill.

When social force is treated as a triggering feature during the construction task, it can be leveraged to teach skills or orderings to the robot. Using the staged skill execution technique, the robot executes instructive behaviors from existing skill data. The robot chooses its preferred ordering of tasks to execute and directs another agent to perform them. In the construction task, the robot was trained to instruct the user to perform the following ordering: locate blue block, move blue block, pick up red block, place red block. While waiting for the user to indicate knowledge of the desired action and to demonstrate the intention to perform it, the robot begins to demonstrate the introductory motions of the skill. This demonstration is performed in increasingly complete motions, stopping just before completing the goal of the subtask. Within the context of the pick up red block skill, the robot first moves above the red block. If the user does not respond, the robot then begins again by moving over the red block, opening and orienting its gripper such that the block can be grasped. Finally, if there is no response from the user, the robot executes the entirety of the action with the exception of the final goal state: moving over the red block, orienting its gripper towards the block, opening its gripper, and placing the red block within its fingers before aborting and returning to a resting position.

With minor changes in treatment, social force is a feature that makes collaborative robotic systems capable of transitioning between student, peer, and instructor roles. With minor changes to the robot's skill execution algorithm, the robot becomes capable of teaching other agents kinematically demonstrated actions, in addition to teaching task ordering preferences.

V. CONCLUSION

We've introduced a rich feature called social force to a collaborative robot's planning and skill execution subsystems. This feature can be calculated in real-time and has been shown capable of providing robotic systems a range of social, collaborative functionality based on its treatment. By adding social force considerations into a robotic system's planner, the robot becomes capable of working with others as a peer as well as learning task ordering from others as a student. With minor changes to a robot's skill execution algorithm, it becomes

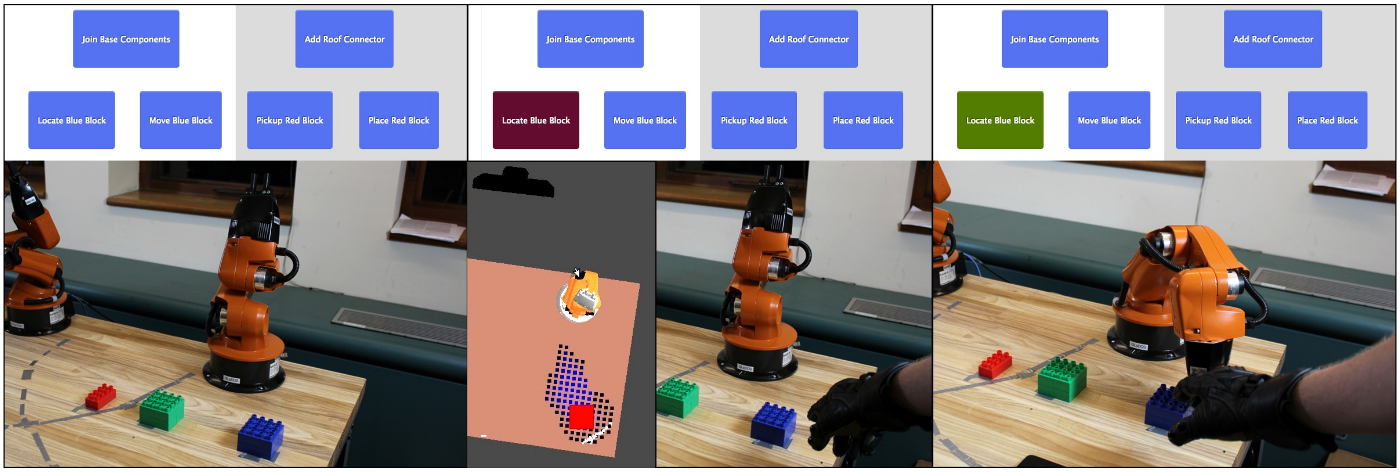

Fig. 4. An example of the ”student” behavior, achieved by attractive social force. In the first panel, the robot is waiting for guidance before choosing an action. In the second panel, the user has gestured near the blue block, exerting social force in the goal region of ”Locate Blue Block”. The red box around the skill name indicates that robot intends to complete this action. The final panel shows the action as ’completed’ in the task tree visualization (green box).

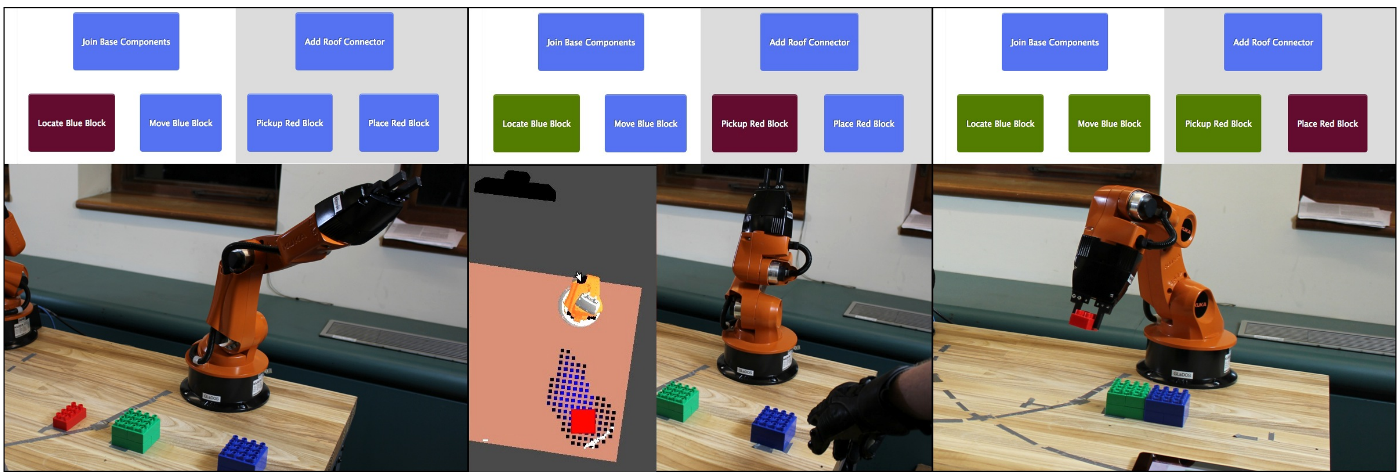

Fig. 5. An example of the ”peer” behavior, achieved by repulsive social force. In the first panel, the robot intends to complete ”Locate Blue Block”. The second panel shows an interruption of the action by a human user. Upon detecting strong social force in its current skill’s goal region, the robot aborts its execution and chooses a different, non-conflicting skill. The third panel shows the robot continuing to exercise conflict-avoidant behavior with the user.

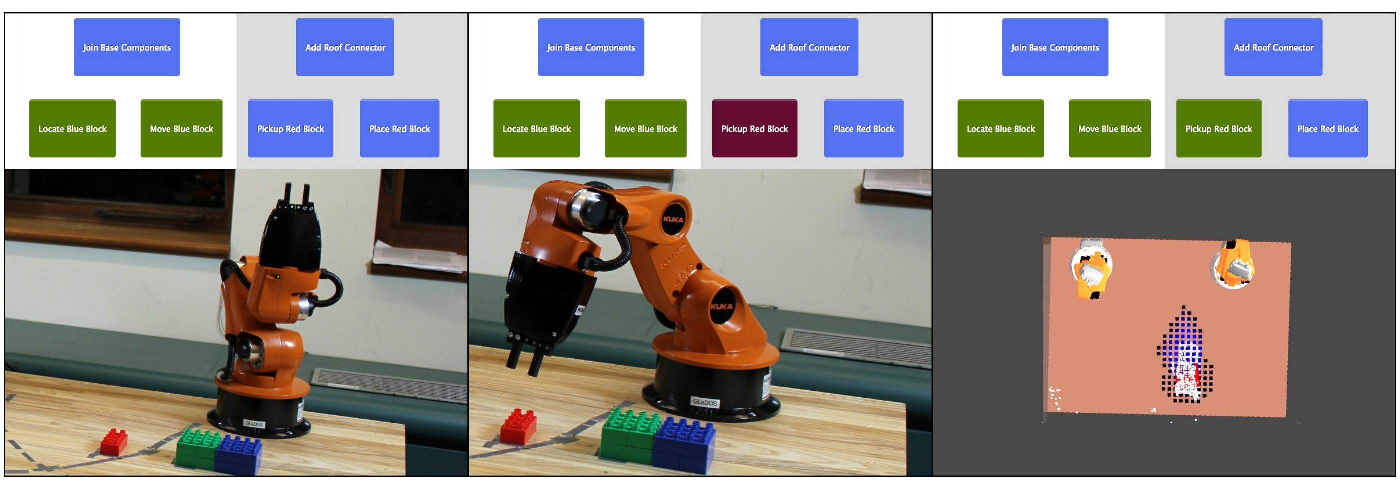

Fig. 6. An example of the ”instructor” behavior, achieved by a using social force as a trigger. The first panel shows the robot evaluating available actions to teach. The second panel shows the robot demonstrating part of the ”Pickup Red Block” subtask without actually completing it. In the third panel, the user has picked up the red block as previously pointed to by the robot, triggering the robot to mark the task as complete.

capable of using social force to teach skills and task orderings to others. We've demonstrated this functionality through a proof-of-concept implementation on a collaborative workbench equipped with a robotic, lightweight manufacturing arm. The advantages offered by this approach can result in improved team efficiency and robot integration into social roles.

Incorporating social force as a feature within a collaborative planner and skill execution algorithm yields promising results, meriting further study through inclusion in more complex, more capable collaborative systems. Future work exploring the benefits of this feature includes correlating the robot's own projected social force values with environmental changes that occur during skill execution to learn about the consequences of one's action paths. Agents can also learn models of collaborators' social force tolerances, and use this data to optimally adapt skill execution to minimize cross-agent conflicts. Socially collaborative robots can also correlate social force values with collaborators' reactions to determine whether an agent is looking to occupy a student, peer, or instructor role when interacting with the robot.

Acknowledgment

This material is based upon work supported by Office of Naval Research grant #N00014-12-1-0822 ("Social Hierarchical Learning") and grants from the National Science Foundation under contracts No. 1139078, No. 1117801, and No. 0835767.

References

[1] B. Hayes and B. Scassellati, "Challenges in shared-environment human-robot collaboration," in *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2013) Workshop on Collaborative Manipulation*.

[2] P. Rochat, T. Striano, R. Morgan *et al.*, "Who is doing what to whom? young infants' developing sense of social causality in animated displays," *PERCEPTION-LONDON-*, vol. 33, no. 3, pp. 355–370, 2004.

[3] J. Shah and C. Breazeal, "An empirical analysis of team coordination behaviors and action planning with application to human–robot teaming," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 52, no. 2, pp. 234–245, 2010.

[4] G. Konidaris, S. Kuindersma, A. Barto, and R. Grupen, "Constructing skill trees for reinforcement learning agents from demonstration trajectories," *Advances in neural information processing systems*, vol. 23, pp. 1162–1170, 2010.

[5] G. Konidaris and A. Barto, "Skill discovery in continuous reinforcement learning domains using skill chaining," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1015–1023, 2009.

[6] B. Hayes, "Social hierarchical learning," in *Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2013) Pioneers Workshop*.

[7] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[8] J. Sweeney and R. Grupen, "A model of shared grasp affordances from demonstration," in *Humanoid Robots, 2007 7th IEEE-RAS International Conference on*. IEEE, 2007, pp. 27–35.

[9] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*. ACM, 2007, pp. 255–262.

[10] C. Atkeson and S. Schaal, "Robot learning from demonstration," in *International Conference on Machine Learning*, 1997, pp. 11–73.

[11] A. G. Billard, S. Calinon, and F. Guenter, "Discriminative and adaptive imitation in uni-manual and bi-manual tasks," *Robotics and Autonomous Systems*, vol. 54, no. 5, pp. 370–384, 2006.

[12] S. Calinon and A. Billard, "Statistical learning by imitation of competing constraints in joint space and task space," *Advanced Robotics*, vol. 23, no. 15, pp. 2059–2076, 2009.

[13] M. Hersch, F. Guenter, S. Calinon, and A. Billard, "Dynamical system modulation for robot learning via kinesthetic demonstrations," *Robotics, IEEE Transactions on*, vol. 24, no. 6, pp. 1463–1467, 2008.

[14] S. Bitzer, M. Howard, and S. Vijayakumar, "Using dimensionality reduction to exploit constraints in reinforcement learning," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 2010, pp. 3219–3225.

[15] B. Akgun, M. Cakmak, J. Yoo, and A. Thomaz, "Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective," in *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*. ACM, 2012, pp. 391–398.

[16] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto, "Robot learning from demonstration by constructing skill trees," *The International Journal of Robotics Research*, vol. 31, no. 3, pp. 360–375, 2012.

[17] L. Cobo, C. Isbell Jr, and A. Thomaz, "Automatic task decomposition and state abstraction from demonstration," in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 2012, pp. 483–490.

[18] P. Stone and M. Veloso, "Task decomposition and dynamic role assignment for real-time strategic teamwork," *Intelligent Agents V: Agents Theories, Architectures, and Languages*, pp. 293–308, 1999.

[19] S. Calinon, F. Guenter, and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 37, no. 2, pp. 286–298, 2007.

[20] R. Parasuraman, T. B. Sheridan, and C. D. Wickens, "Situation awareness, mental workload, and trust in automation: Viable, empirically supported cognitive engineering constructs," *Journal of Cognitive Engineering and Decision Making*, vol. 2, no. 2, pp. 140–160, 2008.

[21] P. S. Tsang and M. A. Vidulich, "Mental workload and situation awareness," *Handbook of Human Factors and Ergonomics, Third Edition*, pp. 243–268, 2006.

[22] J. Shim and A. Thomaz, "Human-like action segmentation for option learning," in *RO-MAN, 2011 IEEE*. IEEE, 2011, pp. 455–460.

[23] C. Breazeal, G. Hoffman, and A. Lockerd, "Teaching and working with robots as a collaboration," in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 3*. IEEE Computer Society, 2004, pp. 1030–1037.

[24] D. Glas, S. Satake, T. Kanda, and N. Hagita, "An interaction design framework for social robots," *Robotics: Science and Systems VII*, p. 89, 2012.

[25] P. Rybski, K. Yoon, J. Stolarz, and M. Veloso, "Interactive robot task training through dialog and demonstration," in *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*. IEEE, 2007, pp. 49–56.

[26] J. Shah, J. Wiken, B. Williams, and C. 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*. ACM, 2011, pp. 29–36.

[27] J. Shah, "Fluid coordination of human-robot teams," Ph.D. dissertation, Massachusetts Institute of Technology, 2011.

[28] S. Nikolaidis and J. Shah, "Human-robot cross-training: Computational formulation, modeling and evaluation of a human team training strategy," 2013.

[29] C. Crick, M. Doniec, and B. Scassellati, "Who is it? inferring role and intent from agent motion," in *Development and Learning, 2007. ICDL 2007. IEEE 6th International Conference on*. IEEE, 2007, pp. 134–139.

[30] L. Talmy, "Force dynamics in language and cognition," *Cognitive science*, vol. 12, no. 1, pp. 49–100, 1988.

[31] P. Wolff, "Representing causation," 2007.

[32] C. Crick and B. Scassellati, "Inferring narrative and intention from playground games," in *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*. IEEE, 2008, pp. 13–18.