Anticipating human actions for collaboration in the presence of task and sensor uncertainty

Kelsey P. Hawkins Shray Bansal Nam N. Vo Aaron F. Bobick

***Abstract*— A representation for structured activities is developed that allows a robot to probabilistically infer which task actions a human is currently performing and to predict which future actions will be executed and when they will occur. The goal is to enable a robot to anticipate collaborative actions in the presence of uncertain sensing and task ambiguity. The system can represent multi-path tasks where the task variations may contain partially ordered actions or even optional actions that may be skipped altogether. The task is represented by an AND-OR tree structure from which a probabilistic graphical model is constructed. Inference methods for that model are derived that support a planning and execution system for the robot which attempts to minimize a cost function based upon expected human idle time. We demonstrate the theory in both simulation and actual human-robot performance of a two-way-branch assembly task. In particular we show that the inference model can robustly anticipate the actions of the human even in the presence of unreliable or noisy detections because of its integration of all its sensing information along with knowledge of task structure.**

I. Introduction

Robots can potentially be effective coworkers to humans in a broad range of applications including industrial manufacturing, logistics, and personal healthcare services [1]. Most prior work in human-robot collaboration has focused on two research topics: acquiring skills by demonstration or teaching, and how to properly engage with users [2]. In many of these efforts the difficult questions revolve around how to generalize the observed human action into an appropriate robot action plan or determining what type of robot interaction is most effective in terms of assisting the human agent.

In these investigations, the observation of the human is often presumed to be unambiguous and accurate. By unambiguous we mean that the robot can assume that human task plans are deterministic; by accurate we mean that the sensing of the human state is engineered to be precise. But in the real world, the human agent may be performing one of a variety of tasks and the execution of those tasks may vary from execution to execution. Furthermore, the ability to sense human actions varies with task and environmental conditions. Thus, a robot must maintain beliefs about the past, current, and future state of the human in the face of human variability and perceptual uncertainty to provide effective human-robot interaction [3].

All authors are affiliated with the Center for Robotics and Intelligent Machines and The School of Interactive Computing, Georgia Institute of Technology, Atlanta, GA. {kphawkins, namvo, sbansal34}@gatech.edu, afb@cc.gatech.edu This work was supported in part by BMW Project# RD441.

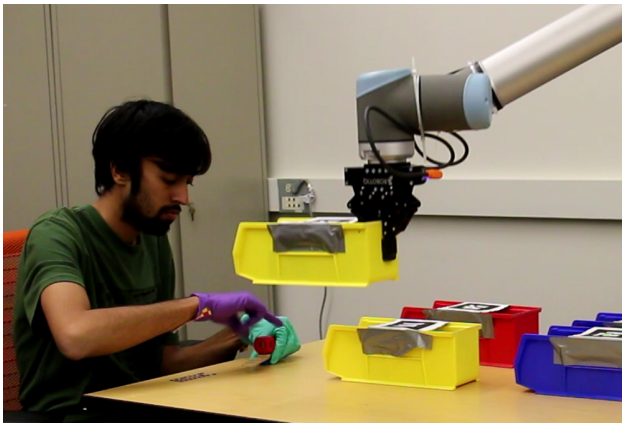


Fig. 1: Experimental Station. A Universal UR10 robot assists a human by fetching and removing bins as needed by anticipating actions of the human.

In this paper we focus on the specific interaction scenario in which a human is performing one of a variety of assembly task variations during which the robot must anticipate which parts the human will need and when. We use this context to identify a variety of uncertainties that arise in the robot perceiving the human actions and making predictions as to when future actions will occur. First is *task ambiguity*: the human agent may choose to perform the same task plan with varying action orders or step omissions, or perform entirely distinct task plans. Second is *task timing uncertainty*: variations in the duration of the steps in the task. Finally there is *action sensing reliability*: sensors designed to respond to human actions are corrupted by environmental factors which introduce both noise into measurements and characteristic disruptions such as when occlusion prevents an action from being sensed at all.

We present here a representation of human action and an inference method which probabilistically models the human's task, infers which task variation the human is doing and predicts when elements of that task will be performed. By modeling detector reliability, incorporating timing distributions, and reasoning over the full sensor history, the robot makes efficient use of its knowledge, integrating evidence over time to improve its belief about the human. After inferring the human's state, we utilize a cost-based planner to optimize the robot's action plan with respect to the posterior human action distributions, reducing the expected cost for an arbitrarily defined system cost function.

The work presented in this paper greatly expands prior results in [4] by accommodating task descriptions more complex than simple linear graphs with no task ambiguity.

In that work, the only inference to be performed was when future actions were likely to be performed and for the robot to anticipate accordingly. Here, the system can model and respond to more ambiguous task orderings corresponding to paths through acyclic polytrees where the branch decisions are determined by the human and must be implicitly inferred by the robot through its sensors.

We organize the remainder of our paper as follows. After discussing selected related works, we develop the representation and inference method for modeling multi-path tasks, assessing the likelihood that a given branch of the task is being performed, and predicting when a particular sub-task will occur. Using these predictions we employ a planning and action system for the robot that attempts to minimize a cost function based upon expected human wait time. We demonstrate the theory in both simulation and actual human-robot performance of a simple two-way-branch assembly task. In particular we show that the inference model can robustly anticipate the actions of the human even in the presence of unreliable or noisy detections because it integrates its entire sensing history with knowledge of task structure and environmental constraints.

II. Related Work

In robotics there has been significant recent study on the role of prediction on the fluency of human-robot interactions, along with the development of learning and planning algorithms that perform action selection in a collaborative context; such work usually presumes sensing is straightforward and that the challenge is making the right action decision. For example, [5] uses an adaptive Markov model to assign confidence about predictions of the human partner's actions. The uncertain predictions are used in a cost-based framework to select the best action. In both that work and subsequent efforts [6] the benefits of employing anticipatory actions in a human-robot task are well observed in human trials. In all these systems the actions of the human are presumed to be clearly and reliably observed.

In the robotics literature there are a variety of approaches to anticipating the actions of humans. These efforts vary in how much a priori knowledge the system has about the task or domain. Huber et. al. [7] provides the robot has complete knowledge of the sub-tasks performed by the human. Fish et. al. [8] and Tenorth [9] collect detailed statistics about the human performance of the specified task and use predict duration variability over time. Koppula and Saxena [10] learn likely sequences of human action from training data. At run time, the robot instantiates a set of probabilistically weighted "anticipatory temporal conditional random fields" to predict which actions the human may take and when. The work presented here also explicitly models possible future sub-tasks sequencings and maintains a probability for each based upon prior info and current observations. However, our possible futures are constrained by an a priori task description.

Wilcox et. al. [11] use strict temporal constraints to develop robotic schedules for human-robot collaborative assembly with the addition of preferences which optimize the plan over the constraints . While they accommodate human variability by using different preferences for different behavior models, they do not address the issue of perceptual ambiguity. We note that the work presented here also frames action selection as minimum cost planning in the face of probabilistic beliefs about when the human will perform various sub-tasks.

Finally, computer vision research, specifically activity recognition, has also developed many approaches to modeling activities composed of sequences of actions. Perhaps the most relevant work is that of Shi et.al. [12] where a Dynamic Bayes Network variant was proposed to recognize partially ordered sequential action. Related, Albanese et al. [13] uses Probabilistic Petri Nets to detect events and [14] learns an activity's decomposable structure of "actionlets" with a probabilistic suffix tree; given that data structure, early prediction of sub-action can be made. In [15], Tang et. al. demonstrated how to use a variable-duration Hidden Markov Model (HMM) to learn an action's latent temporal structure and showed it helps improve detection results in the presence of noisy sensors. This is similar to the work here where sensing information is integrated with a structural description of the task to improve action detection.

III. Representation and Inference

In [4], we developed a representation, inference procedure, and reactive planning system in the context of simple linear graph task descriptions. The system modeled the task as a known sequence of human actions, incorporating duration knowledge, task constraints, and detector observations simultaneously. Given a history of task constraints and detector observations up to the current time and an estimate of these values in the future, the system inferred the distribution over when human actions occurred or will occur.

The key development in that work was representing the linear graph as a sequential Bayes net where the state variables are the beginning and ending times of each of the actions. Duration models allowed for conditioning the end times upon start values, detectors were designed that provided diagnostic information as to when an action was occurring, and online inference procedures were developed that incorporated not only all detections viewed up to the current time but also task constraints such as whether the robot had performed a necessary action that would enable the human to progress in his task.

In this paper we significantly extend that work to allow for task variations where the human is not limited to a strict linear path but can be considered as "multi-path": the task may be a partially ordered one where certain actions can occur in a variety of orderings, or even a set of multiple tasks where some actions may be skipped altogether.

A. AND-OR tree task representation

We begin by defining a representation for the multi-path task which governs what human action plans are considered. A task is an ordered AND-OR tree with leaves called *primitive actions*. Primitive actions are discrete human states

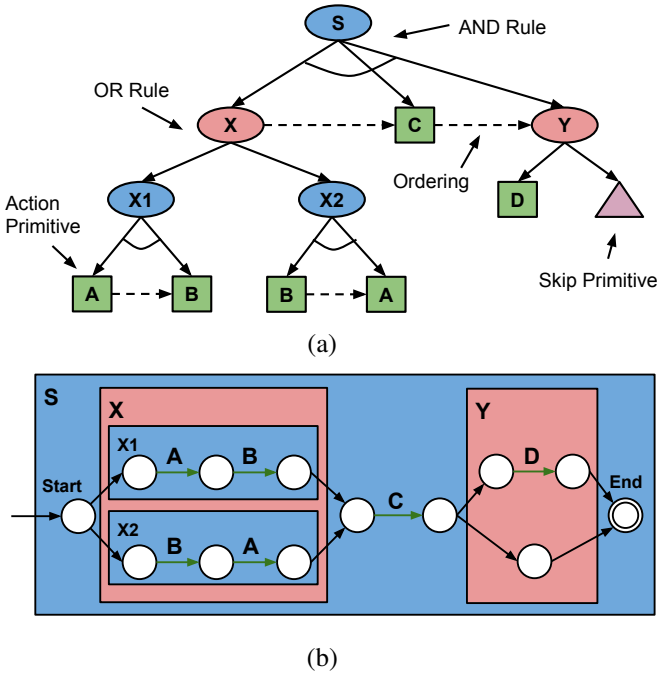


Fig. 2: A complex activity represented by **(a)** an AND-OR tree representation of a task and **(b)** the equivalent acyclic FSM

associated with a start and end time and a probabilistic detector which provides evidence about the transition timings. Inner nodes are composition elements which collect children and indicate either an AND rule or OR rule. An AND rule designates that if this node is executed by the human, all of the children will be performed in the order specified. OR rules assert that exactly one of its children is an acceptable path. Skip primitives are leaves of an OR node that indicate the execution of the node can be skipped entirely.

Figure 2(a) illustrates a task with action primitives *A,B,C* and *D*. The tree encodes that first *A* and *B* can be performed in any order, followed by *C* in either case, and finally, *D* can be performed, but is optional. We can see that tasks could also be encoded as acceptable strings to an acyclic, nondeterministic finite state machine (FSM) (Fig. 2(b)). Thus, the proposed representation is analogous in prior work from both robotics (e.g. [16]) and computer vision [12, 17]. However, the AND-OR tree representation can only encode action plans of finite length.

In the following sections, we discuss conversion of our AND-OR tree representation into a probabilistic graphical model for predicting human actions and their timings.

B. Primitive actions and detectors

For each primitive action A we denote A_s and A_e to be the probabilistic variables representing its start and end time. We utilize a discrete representation of time, where each variable admits integer values. Each value represents a continuous time interval with a scaling that depends on the ratio of T, the number of time intervals, to T_{max}, a maximum duration for the task. Each action detector variable Z^A is conditioned on an primitive action's start and end times which can be potentially reused for other actions. Furthermore, each end

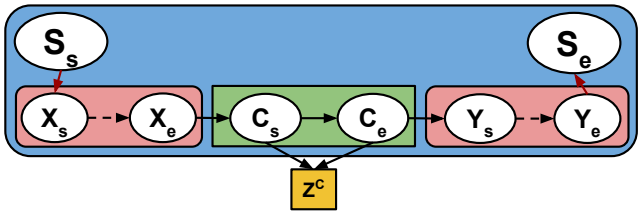


Fig. 3: The *S* AND rule from figure 2 converted into a Bayes net of three subtasks. Since the green *C* action is a primitive, the start and stop times are conditioned on the sensor Z^C. The dotted lines indicate a further expansion as seen in figure 4.

time A_e is conditioned on the action's start A_s, and each start time B_s is conditioned on the previous action's end A_e (see action *C* in fig. 3). Note that while this notion is always valid in our previous work [4], the intra-action conditionals will soon be modified for some connections.

We assume a duration prior for each action, $P(A_e|A_s)$, distributed according to a trimmed Gaussian whose parameters are learned offline. In our experiments, for the action transition conditional $P(B_s|A_e)$ we used a gating function which enforces the environmental constraint that if the conditions for the human to perform the next action have not yet been met, then the transition cannot yet have been made (see [4]). We also produce an observation likelihood $P(Z^A|A_s, A_e)$ from the raw action detector $F_A[\alpha, \beta]$. The detector represents how consistent the observed data is with the action starting at time α and ending at time β for every possible(α, β) of the entire input sequence. Then, the likelihood can be computed based on that detection: $P(Z^A|A_s = \alpha, A_e = \beta) = h_A F_A[\alpha, \beta]$ for a constant h_A. The choice of F_A reflects the sensitivity and reliability of the sensing system in being able detect the action A. If, for example, there was no available sensing, then F_A would be a constant, effectively eliminating any impact on the inference.

C. Sequence of actions: AND

An AND-rule conversion is shown in Fig. 3. We set the start and the end of a composition according to its start and end of its subtasks ($S_s = X_s$, $S_e = Y_e$, denoted by red arrows). Given all the conditional probability tables (CPT) have been computed, we can use a message-passing algorithm to perform exact inference. The local output will be the posterior distributions of the start and end of every actions: $P(Z)$ and $P(S_s|Z), P(S_e|Z), P(X_s|Z), P(X_e|Z), \ldots$ [4].

D. Branching: OR

An OR-rule conversion is shown in Fig. 4. Here we describe a more complicated composition: the subtask *X* suggests that the human will take either path *X1* or *X2* with discrete probability p_{X1}, $p_{X2} = 1 - p_{X1}$ (Fig. 4). The network will include the nodes X_s, X_e and recursively all components in $X1, X2$. If the human takes path *X1*, we would have $X_s = X1_s$, $X_e = X1_e$, and we need to represent that *X2* does not happen, which we denote $X2_s = X2_e = -1$. We use $\exists X1$ and $!X1$ to denote the event *X1* happens $(X1_s, X1_e > 0)$ or not $(X1_s = X1_e = -1)$.

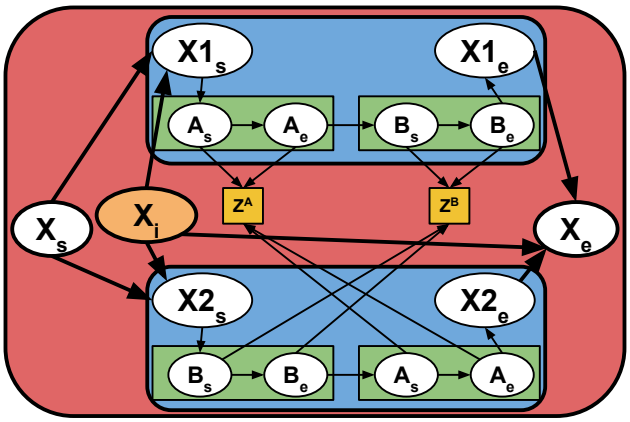


Fig. 4: The *XOR* rule from figure 2 converted into a Bayes net with two possible paths. In this subtask, action primitives *A* and *B* can be performed in either order. The switching variable X_i is added to help manage the OR relationship. In this instance, the action detectors are shared across branches.

A standard approach to realizing "OR" in Bayes network is using a "switching" variable X_i [18].

The timing of *X* can be presented in terms of the timing of *X1* and *X2* as

$$P(X_e = \alpha, Z) = W_1 P(X1_e = \alpha, Z^{X1} | \exists X1) + W_2 P(X2_e = \alpha, Z^{X2} | \exists X2)$$

$$W_1 = P(\exists X1)\, P(Z^{X2} | !X2);\; W_2 = P(\exists X2)\, P(Z^{X1} | !X1)$$

If *X1* is a primitive action, then $P(Z^{X1}|!X1) = h_{X1} F_{X1}[-1,-1]$; we choose $F_{X1}[-1,-1]$ to have the "null" score of F_{X1} defined to be its average score computed over all training data. If *X1* is a composition, then $P(Z^{X1}|!X1) = \prod_{\text{M in X1}} P(Z^M|!M)$

More simply, the inference's forward and backward processes are performed on *X1* and *X2*, then the results are combined for *X*. The weights of *X1* and *X2* depend on two factors: the prior probabilities $P(\exists X1)$, $P(\exists X2)$ and the likelihood $P(Z^x | m_s, m_e)$ for every primitive action m in *X1* and *X2*. For example, strong detection of actions in subtask *X1* which are consistent with the duration priors would make *X* more likely to be *X1* than *X2*.

Note that while the themes and applications make this model seem like an HMM, this model is very different. While in an HMM, state variables represent the actual state of the system at a particular time step, our representation's state variables are the timings of the beginning and ending of states. This approach allows us to enforce strict task structure constraints, reducing variability, while still maintaining healthy alternative hypothesises.

E. Inference

The message-passing algorithm is used to perform inference. Besides values between 1 and T, the distribution of the timings now also include special value -1, which indicates the human never performs the action. The inference returns as output: the posterior probabilities of whether each branch was taken, and the distributions of when an action starts or ends, given the action was performed. From this output, we can also compute the probability that the action x is being performed at time step t, for every x and t.

Inference is performed at a particular point in time and as more sensor information is received, inference should be run again to produce updated predictions. Likelihood values for detections which have not yet occurred are set to a uniform constant. The running time for inference is roughly $\mathrm{O}(NT^2)$ where N is the number of action primitives and T is the number of time intervals in the discrete time state. This hints that the primary trade-off in efficiency is reduced precision in the timing densities. For details on the inference algorithms refer to Appendix A.

IV. Example Application Description

We first present a human-robot collaborative application we use to motivate our investigation. A human sits at a table across from a robot collaborator who is safely out of reach of the human, but who can move a set of bins both into and out of the reach of the human (Fig. 1). Each bin contains a variable number of Baufix toys, a wooden construction set of screws, nuts, and bolts, which can be used to make small model vehicles and other designs. The bins are kitted so that a subset of the bins could be used to construct a few different models.

For the task, the human is instructed to begin building a model from the pieces in the bins. Their reaches are generally restricted to withdraw one part from a bin at a time. Since the human cannot remove a part from a bin not in reach, this imposes a task constraint which the robot must satisfy for the pair to complete the task. When the human needs to reach for a part from a bin not in the workspace, they are instructed to wait until the robot has delivered the bin they need. Based on observations of the human gathered from sensors in the environment, combined with a model of the task, the robot begins delivering bins the human might need. There are only M slots ($M = 3$ for our experiments) in the human's workspace into which the robot can place bins, so eventually the robot must decide to remove unneeded bins and deliver more demanded ones. When more than one construction is possible, the knowledge of which one the human is performing is not made explicit a priori and must be inferred by the activity of the human.

We define each primitive action as a draw of a particular piece inside a particular bin. The start of this "drawing action" is defined as the moment when the hand touches the piece (and the end of the action is the start of the next one). We use a likelihood function

$$F_A[\alpha, \beta] = N(H_\alpha(\text{bin(A)}); \mu_{Pos(A)}, \sigma_{Pos(A)}) + w_m$$

where bin(A) is the bin corresponding to action A, and $H_t(b)$ is the position of the closest hand to bin b, represented in the local coordinate frame of b, at time step t. Parameters $\mu_{Pos(A)}$ and $\sigma_{Pos(A)}$ are learned during training. $\sigma_{Pos(A)}$ represents confidence about the detector's accuracy, while w_m is a uniform distribution that represents the confidence

about the detector's recall. With the defined parameters, a high confidence detector will have small $\sigma_{Pos(A)}$ and small w_m. As these 2 parameters get bigger, the system is less confident in its sensing. A very high value of $\sigma_{Pos(A)}$ or w_m would make $F_A[\alpha, \beta]$ nearly constant , which is equivalent to no available sensing information. We also compute the expected detection score and assign this value to $F_A[-1, -1]$.

A special "waiting action" is included before every "drawing action" to add the constraint that a bin must be in the workspace before an draw can occur. Instead of separate duration and detection likelihoods for this special action, we define the multiplication of both as: $P(A_e = \beta, Z^A | A_s = \alpha) = 1$ if bin(A) is not available during the interval $[\alpha, \beta - 1]$ and becomes available at time step β, or otherwise 0.

Using the inference output, a prediction can be made about what actions have already happened, what actions are going to be performed next and when. The planner needs to know when a bin is needed and not. Hence we will use the distribution of the start of the action of waiting to draw the first piece from the bin as when that bin is needed, and the last piece from the bin as when that bin is no longer needed. We presented in our previous paper [4] a cost-based planner which optimizes bin delivery and removal timings given the posterior distributions . The planner attempts to minimize expected sum squared wait times, which we use to reduce both total wait time and the maximum wait period. The planning is mostly identical, except that costs are now weighted by the posterior branch probabilities which come from the OR-rules.

V. Evaluation

A. Task Descriptions

We developed a simple, illustrative task, to demonstrate the types of behavior our system exhibits in a collaborative assembly scenario. The human attempts to assemble one of two possible toys whose parts are each separated into 4 bins and the robot has no prior knowledge as to which toy the human will be assembling. The two toy models have an identical assembly structure and the base structure, in bin A, is the same for both. However, each model is a different color, and all successive parts past the base are in different bins. Thus, the human needs bins B_1, B_2, and B_3 for model B and C_1, C_2, and C_3 for model C. We require that the human perform one reach for each part in the bin and there are total of 14 parts that need to be assembled, 6 in bin A, 1 in bin B_1/C_1, 1 in bin B_2/C_2, and 6 in bin B_3/C_3. The bin A is already in the human's workspace when the task starts and takes enough time that both bin B_1 and bin C_1 can be delivered before the human finishes with it.

Since the robot cannot determine which model the human is building before reaching into one of the bins in a branch, the robot almost always begins by delivering both B_1 and C_1. Assuming the robot does not remove bins preemptively a problem we explored more heavily in our previous work [4], the best case scenario is when the robot only delivers bins $*_2$ and $*_3$ for the branch the human performing. In the worst case, the same bins are delivered in the branch the robot is not performing, followed by the two in the branch they are performing.

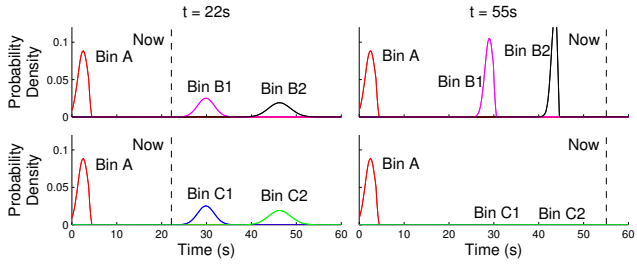


Fig. 6: Inference result at two different times where the human reaches into bins A, B1, and B2. As more evidence is observed, the robot understands that the human does not need C1 or C2 and the timing predictions become sharper.

B. Simulation

We developed a simulator which allowed us to evaluate both our inference and our planner in a controlled environment. The human agent simulation was programmed to reach towards bins based on random times drawn from our duration model. If a necessary bin was not available in the workspace, the agent would remain stationary and wait.

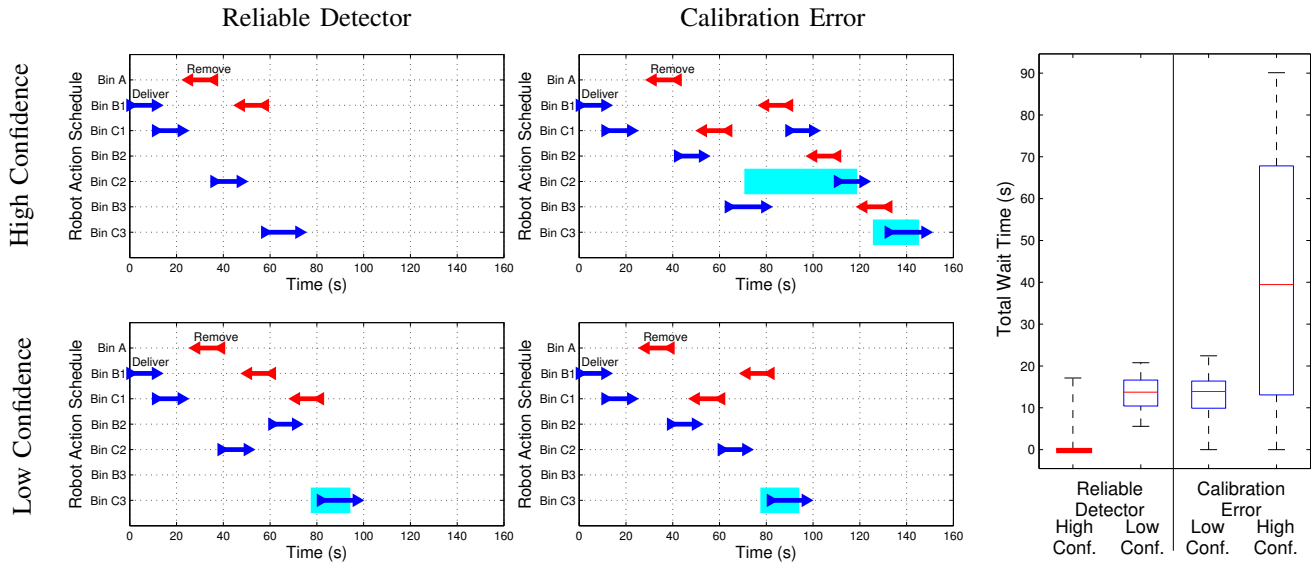
In order to investigate the behavior of our system in the face of action detection ambiguity, we modified the detector to present a false hand position to the detectors. A calibration error is introduced which shifts the perceived hand positions so that a reach into one bin often looks more like a reach into the next bin to the side.

Likewise, in the robot's inference model, we alter the parameters of the sensor model to control the "detector confidence". The detector's precision confidence is modified to be lower by widening the detector's σ value so that it has less precision but higher recall.

In Fig. 6 we illustrate how the inference resolves both task uncertainty and model uncertainty simultaneously. Early on, the robot projects the timings of future human actions to be wide and attributes equal probability to both paths. Once the robot detects reaches, it knows both when the human began using the bins and which bins they need.

We compared a well-calibrated reliable detector to one with a calibration error when the robot had either a high and low confidence in the precision of the detector. The results of our simulation trials can be found in Fig. 5. In the high confidence case, the robot committed to the path it saw first, delivering bins $*_2$ and $*_3$ as soon as possible. When the detector was reliable, this meant the human never had to wait. However, when it had a calibration error, after incorrectly perceiving a reach into the other bin, it overcommitted down the wrong path, delivering two bins the human did not need.

The calibration error was overcome by lowering the confidence in the sensor. In the low confidence cases, the robot made the human wait slightly longer than the first condition, but not as long as when it overcommitted. It accomplished this by covering its bases, delivering both $*_2$ bins, regardless of which it saw first. We can see that for the reliable detector, this meant that it did not perform as well as the high



(a) Robot action schedules on example cases from simulation (b) Results

Fig. 5: Results from simulated trials of a simple 2-path grammar for each of four detector performance/detector confidence combinations: a well-calibrated reliable detector versus one with calibration error, and a high confidence in the reliability versus a low confidence. **(a)** In each case, the simulated human was trying to build from bins A, C_1, C_2, and C_3, in that order. The cyan overlays indicate the human was waiting on that bin during that interval. **(b)** Distribution of results from $N = 60$ simulated trials, for each of the four conditions, varying the human action durations. The red mid-line is the median; the boxes, the middle two quartiles; the extrema, the minimum and maximum values. As the robot's estimation about its detector reliability better matches the actual stochastic processes, the human's wait time decreases, the variance of the wait times is reduced, and its worst-case wait time is significantly limited.

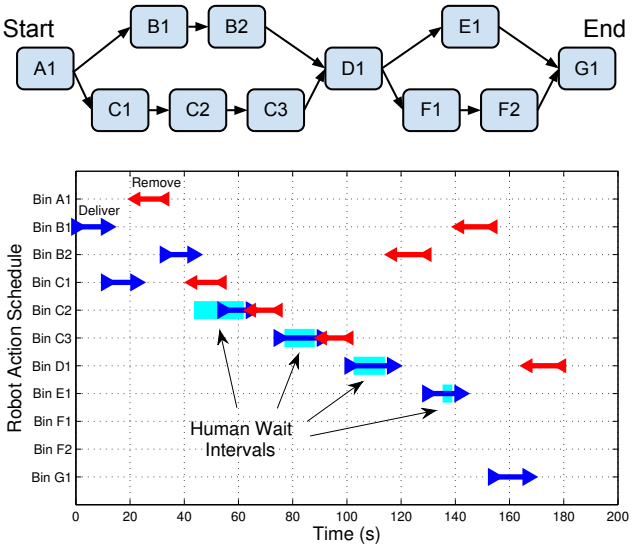


Fig. 7: The system applied to a more complex grammar. **Top:** The human task action component ordering. Each block represents a bin potentially needed by the human. From the robot's perspective, the human could perform the task in one of 4 different paths. **Bottom:** An example run of the simulated system where the human took the C branch followed by the E branch. The system had a calibration error and moderately tuned parameters.

confidence case. Thus, correctly tuning the robot's belief about its detector to its underlying performance is important for improving system performance.

We also applied the simulator to a more complex grammar to demonstrate it can model arbitrary tasks (Fig. 7). The robot began by delivering both B_1 and C_1, to determine which branch the human was doing. The robot first mistakenly delivered B_2, but then recovered when it saw no reaches. It delivered the C bins, followed by D_1 and E_1. When E_1 was delivered, it immediately got positive information, so it skipped the F bins entirely, immediately delivering G_1. Note that it waited until D1 was delivered to finally remove the B bins, and instead removing the C bins very quickly. This shows how the robot is optimizing over residual ambiguity to both follow its most likely paths while still covering its bases as best as possible.

C. Robot implementation

In this section we present experiments performed by a human-robot collaborative team performing the task described above. The robot was a 6-DOF Universal Robots UR-10 mounted to a steel table with a Robotiq C-model parallel jaw gripper (Figure 1). Above the robot, a webcam was mounted to track the positions and orientations of the bins, affixed with ARTag augmented reality tags. To the side of the human and bins, a Kinect RGB-D sensor was mounted to sense the behavior of the human. The entire system is calibrated such that the locations of the bins are known with respect to both the robot and the human sensing.

The task the human performed is exactly the same task described in section IV that involves building either of two toy models, A and B. To track the human collaborator's hands, we used brightly colored surgical gloves and implemented a color blob tracker on the RGB-D sensor. A relatively simple model of euclidean distance between the bin and the hand closest to it, is used to compute the detection score F_A, as described in detail in section IV.

The approach we take with the human-robot collaboration mirrors that taken in simulation. For the sensing system we consider three cases. The first, "reliable", is denoted RD and is where the detectors performance reflects the statistics seen in the training data. The second and third, each "unreliable", have some form of sensing perturbation introduced. One disturbance, analogous to the simulation, is Calibration Error (CE) where we altered the extrinsic calibration of the Kinect by 6 cm. This has a similar effect to the tripod mounting the Kinect being inadvertently *shifted*. The other perturbation, referred to as False Positives (FP), involved attaching a glove having the same color as one of those being tracked to the bin corresponding to the bin B_2 in view of the Kinect. It produced the effect of having a high detection score for that bin throughout the duration of the task, hence that bin receives many false positives.

Also following the simulation approach, we varied the model parameterization used for inference. The first of these we refer to as High Confidence (HC). These are the parameters learned from training data which was collected without the above mentioned perturbations. The two remaining parameters settings we refer to as "Low" confidence, LC1 and LC2. These parameters reflect two possible types of sensor uncertainty. LC1 models inaccuracy in detection as an increase in the variance of detector measurement. LC2 models the unreliability of actual detection such as when occlusion might prevent any detection of an action.

Table I shows the average human total waiting times for a task performed in each of the conditions. The High Confidence parameterization - learned from unperturbed sensing - performs well in RD but causes the human to wait for long periods of time in the other cases. Specifically, CE causes the detection scores to become quite small which lead to the system being unsure of when a reach into a bin is made. This causes the system to wait an extended time before deciding that a bin is no longer needed. The incorrect detections also leads to ambiguity in terms of which task is being performed making the system slow in recovering after committing to an incorrect model. In FP, the high detection score for B_2 causes a bias toward task B, so the system always commits by delivering B_3, B_4. Thus, when task C is actually being performed, the human has to wait for a long period.

The low confidence parameterization LC1 for CE was produced by increasing the variance on the detection Gaussians. Notice that this parameterization does not presume having knowledge of the actual perturbation - in this case an actual bias. Rather, it simply permits noisy detections. This has the effect of increasing the detection scores when reaches into the bins are far from the learned positions. This causes the system's task uncertainty to increase causing the robot to not commit to any particular task until a later time. For instance, when a reach is made into B_2 but the calibration causes the reach to appear near C_2 as well, the robot delivers both B_3 and C_3. This leads to a reduced waiting time as the system understands there is uncertainty in the sensors and reacts by being conservative and waiting a small period instead of making a mistake that may cause a long wait.

Mean Human Total Wait Time (s)

	Reliable Detect	Calib Error	False Positives
High Conf	3.0	84.1	57.0
Low Conf Calib	4.6	19.6	
Low Conf FP	7.9		37.4

TABLE I: Averaged total wait times for $N = 6$ real-world trials for each condition. Waiting Times and appropriate detection confidence parameterizations for Calibration Error and False-Positive perturbations. As can be seen from the first columns when the sensors are reliable the waiting times are lower. Also, lower confidence parameters perform better than a high confidence model when faced with systematic perturbations.

The low confidence LC2 settings in FP were found simply by scaling down the raw detection score for the bin with the glove attached. This is analogous to reducing the net effect of that sensor. Again, the model of the uncertainty used is not aware of the actual defect in the sensing, only that the sensor for some particular action is "stuck" on. This reduces the confidence in detections of B_2 and thus in the confidence that the task B is being performed. It was noticed that in this case the system, owing to a large number of false positives committed initially to task B. However, it was able to recover from this mistake fairly quick leading to the average waiting time being less than those of the High Confidence parameterization. While such a perturbation and confidence setting may seem contrived, we provide this only as another example of the system being able to easily accommodate particular sensor degradation by a simple adjustment to the parameters of our model, not by having an accurate assessment of the sensor failure mode.

Another observation made through these experiments is that even though the High Confidence works best for Reliable Detection, the other two settings are still able to perform well in both the Reliable Detection case and the perturbed situations. The Calibration Error situation directly mirrors the effect seen in simulation. The False Positive perturbation demonstrates the effect of task ambiguity on our system. Although, the noise model in the real experiments differs from that in simulation. We believe that the behavior of the system in terms high and appropriately confident parameterizations with respect to the presence and absence of systematic perturbations remain the same.

VI. Discussion and Conclusion

We have proposed a significant extension of our previous work which allows us to model multi-path branching in a probabilistic manner. By maintaining densities over multiple branch possibilities, the robot can act in a way that does not require it to commit to only one particular branch belief. By encoding the task structure, the robot can continually integrate new information and propagate it forward and backward in time, always improving its perception of the human's past, current, and future states.

Furthermore, the robot can use its knowledge to be more or less conservative when it comes to making predictions about which actions the human will need the robot to do next. The robot can leverage this information to optimize its

execution to reduce the number of supporting actions it must perform and to improve collaborative task efficiency.

We have performed experiments which demonstrate that by simply adjusting the confidence in the detectors, the system can behave more appropriately in the face of uncertainty and perceptual perturbations. We also show that even naive detectors like the one used here with appropriate parameters can handle complex perturbations.

In future work, we will perform a more rigorous real-world evaluation of the system with more trials and novice users. Specifically we will investigate the impact of the system inference settings on the human agent's sense of fluency.

REFERENCES

[1] "A Roadmap for U.S. Robotics," Robotics Technology Consortium, 2013.

[2] B. Hayes and B. Scassellati, "Challenges in Shared-Environment Human-Robot Collaboration," in *Collaborative Manipulation Workshop at HRI*, 2013.

[3] G. Hoffman and C. Breazeal, "Collaboration in Human-Robot Teams," in *AIAA 1st Intelligent Systems Technical Conference*. Reston, Virigina: American Institute of Aeronautics and Astronautics, Sept. 2004, pp. 1–18.

[4] K. Hawkins, N. Vo, S. Bansal, and A. Bobick, "Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration," in *International Conference on Humanoid Robots*. IEEE, Oct. 2013.

[5] G. Hoffman and C. Breazeal, "Cost-Based Anticipatory Action Selection for HumanRobot Fluency," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 952–961, Oct. 2007.

[6] ——, "Effects of anticipatory perceptual simulation on practiced human-robot tasks," *Autonomous Robots*, vol. 28, no. 4, pp. 403–423, Dec. 2009.

[7] M. Huber and A. Knoll, "When to assist?-Modelling human behaviour for hybrid assembly systems," in *Robotics (ISR), 2010 41st ...*, 2010, pp. 165–170.

[8] L. Fish, C. Drury, and M. Helander, "Operatorspecific model: An assembly time prediction model," *Human Factors and ...*, vol. 7, no. 3, pp. 211–235, 1997.

[9] M. Tenorth, F. D. Torre, and M. Beetz, "Learning Probability Distributions over Partially-Ordered Human Everyday Activities," in *International Conference on Robotics and Automation*, 2013, pp. 4524–4529.

[10] H. S. Koppula and A. Saxena, "Anticipating Human Activities using Object Affordances for Reactive Robotic Response," in *Robotics: Science and Systems*, Berlin, 2013.

[11] R. Wilcox, S. Nikolaidis, and J. Shah, "Optimization of temporal dynamics for adaptive human-robot interaction in assembly manufacturing," in *Robotics: Science and Systems*, 2012.

[12] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa, "Propagation networks for recognition of partially ordered sequential action," in *Computer Vision and Pattern Recognition*, vol. 2. IEEE, 2004, pp. 862–869.

[13] M. Albanese, R. Chellappa, V. Moscato, A. Picariello, V. Subrahmanian, P. Turaga, and O. Udrea, "A constrained probabilistic petri net framework for human activity detection in video," *Multimedia, IEEE Transactions on*, vol. 10, no. 6, pp. 982–996, 2008.

[14] K. Li, J. Hu, and Y. Fu, "Modeling complex temporal composition of actionlets for activity prediction," in *Computer Vision–ECCV*. Springer, 2012, pp. 286–299.

[15] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1250–1257.

[16] H. Goto, J. Miura, and J. Sugiyama, "Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model," in *HRI Workshop on Collaborative Manipulation*, 2013.

[17] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 852–872, 2000.

[18] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques (Adaptive Computation and Machine Learning series)*, 2009.

APPENDIX

The whole network is constructed recursively, the message-passing inference is also a recursive algorithm, consisting of 4 steps:

Inputs: (1) $P(\exists S)$: the prior probability of S happening, (2) $P(S_s|\exists S)$: The prior probability of the start of S, (3) $P(Z^{end}|S_e, \exists S)$: the likelihood representing the constraint on the end of S, and (4) the CPT $P(A_e|A_s)$ and $P(Z^A|A_s, A_e)$ for all primitive action A (recall that our random variables have discrete values between 1 and T. The special value $A_s = A_e = -1$ means $\exists A$, the case where the action A happens)

Step 1, Forward phase: Given $P(A_s, Z^{pre(A)}|\exists A)$, one can compute $P(A_e, Z^{pre(A),A}|\exists A)$ for every action A (where $Z^{pre(A)}$ stands for the observation of all actions happening before A). If A is a primitive action, then compute the joint $P(A_s, A_e, Z^{pre(A),A}|\exists A)$ and perform marginalization. If A is defined as M AND N, then recursively compute $P(M_e, Z^{pre(A),M}|\exists M)$ and $P(N_e, Z^{pre(A),M,N}|\exists N)$ then we have the distribution of A_e the same as N_e. On the other hand if A is defined as M OR N, then the distribution of A_e will be weighted combination of M_e and N_e according to equation 3.

The forward process starts with $P(S_s|\exists S)$ and recursively compute $P(A_s, Z^{pre(A)}|\exists A)$, $P(A_e, Z^{pre(A),A}|\exists A)$ for every action A

Step 2, Backward phase: similarly, this process starts with $P(Z^{end}|S_e, \exists S)$ and recursively compute $P(Z^{post(A)}|A_e, \exists A)$, $P(Z^{A,post(A)}|A_s, \exists A)$ for every action A (here $Z^{post(A)}$ stands for observation of all actions happening after A).

Step 3, compute the posteriors: this is done simply by multiplying the forward and backward messages, we obtain $P(A_s, Z|\exists A)$ and $P(A_e, Z|\exists A)$ for every action A. Additionally we can have $P(Z) = \sum_{t>0} P(S_s = t, Z)$

Step 4, compute the posterior probabilities of an action happening: starting with $P(\exists S|Z) = P(\exists S) = 1$, evaluate $P(\exists A|Z)$ for every symbol A recursively.

Given S is defined as A AND B, then $P(\exists A|Z) = P(\exists B|Z) = P(\exists S|Z)$.

Given S is defined as A OR B, one can compute (apply similar formulas for B):

$$P(\exists A|Z) = P(\exists S|Z)\frac{P(\exists A, Z|\exists S)}{P(\exists A, Z|\exists S) + P(\exists B, Z|\exists S)} \quad (1)$$

where $P(\exists A, Z|\exists S)$ can be calculated:

$$P(\exists A, Z|\exists S) = P(\exists A|\exists S)\sum_{t>0} P(A_e = t, Z|\exists A) \quad (2)$$

Output: the probability of action A happening $P(\exists A|Z)$, and if that the case, the distribution of the start and end $P(A_s, Z|\exists A)$, $P(A_e, Z|\exists A)$. We can compute:

$$P(A_s|Z) = P(\exists A|Z)\frac{P(A_s, Z|\exists A)}{\sum_{t>0} P(A_s = t, Z|\exists A)} \quad (3)$$

for values between 1 and T. Note that $P(A_s = -1|Z) = P(!A|Z) = 1 - P(\exists A|Z)$.