Probabilistic human action prediction and wait-sensitive planning for responsive human-robot collaboration

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

***Abstract*— A novel representation for the human component of multi-step, human-robot collaborative activity is presented. The goal of the system is to predict in a probabilistic manner when the human will perform different subtasks that may require robot assistance. The representation is a graphical model where the start and end of each subtask is explicitly represented as a probabilistic variable conditioned upon prior intervals. This formulation allows the inclusion of uncertain perceptual detections as evidence to drive the predictions. Next, given a cost function that describes the penalty for different wait times, we develop a planning algorithm which selects robot-actions that minimize the expected cost based upon the distribution over predicted human-action timings. We demonstrate the approach in assembly tasks where the robot must provide the right part at the right time depending upon the choices made by the human operator during the assembly.**

I. Introduction

Human-robot collaboration is rapidly gaining interest in a broad range of applications, including industrial manufacturing and assembly [1], as well as personal services [2]. In such scenarios, human and robots work in a shared space, focused on accomplishing a joint task. The successful performance of this task requires that the human and robot work as an effective *team* implying some notion of joint intention where both parties maintain a set of shared beliefs about the state of the world and the task being performed [3, 4]. One goal of maintaining such shared beliefs is that both the human and the robot should know when to perform specific actions in support of the collaborative task; this synchronization is informed by both knowledge of the task being performed and perception of the human and environment.

Much prior work has focused on the first element above: providing the robot with knowledge of the task either through demonstration [5] or by explicit teaching. However, when considered in a collaborative setting, most research assumes that the sensing of the execution of the task is straightforward and ignores real-world sensor effects like noise, drop-out, and occlusion.

Despite great strides in recent years, in many uncontrolled environments human action perception is still noisy and unreliable. Furthermore, human collaborators can exhibit great variability in the manner they perform tasks, such as action speed and personal discretion. It is in this context that the robot must determine the state of the collaborative task being performed and it must infer both what to do and

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.

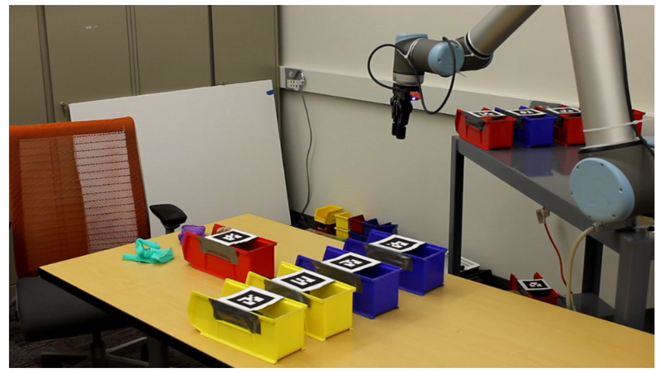

Fig. 1. Station where a Universal UR10 robot assists a human by fetching and removing bins as needed by anticipating actions of the human.

when to do it. In this paper we introduce a new, probabilistic model and inference mechanism that permits the robot to seamlessly infer the current state of the task and to predict the distribution of when particular robot actions would be appropriate. We will show how this representation allows for the modeling of sensor uncertainty, human variability, environmental constraints, and task structure to more accurately deduce the timings of the human's actions as more sensor information is observed.

We organize the remainder of our paper as follows. After discussing selected related work, we describe an example task that will be used both to derive and explain the necessary theory as well as to demonstrate in both simulation and with actual human-robot execution. In Section IV we develop a representation of activity that decomposes a task into a set of sub tasks and permits reasoning about when a particular sub-task is likely to be executed; this inference allows the robot to appropriately anticipate when it should perform the necessary collaborative action. Section V describes how an action plan is formulated given the probabilistic assessment of when human sub-tasks will occur. Simulations are then provided that show how the system naturally handles various uncertainties in a unified fashion, for example altering the timing of planned actions based upon the certainty of perceptual measurements. Finally, we demonstrate a robot appropriately assisting a human by fetching correct parts at the right time and removing used part-carriers when certain they are no longer needed.

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. A straightforward demonstration of the importance of prediction on the fluency on HR collaboration is seen in [6]. In that work a joint assembly task is specified by a provided finite state machine representation (as in [2]) and the robot learns to predict the next action of the human more quickly by noticing repeated patterns of low level actions such as grasping a part. By assuming that repeated low level action imply repeated high level sub-task performance, the robot learns to anticipate the human action and can more quickly respond with the necessary assistive action. A more sophisticated state/action model is found in [7] which 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 [8] 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.

Wilcox et. al. 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 [1]. 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.

Also from the robotics domain there has been a variety of work on how to anticipate the actions of humans. These efforts lie on a spectrum reflecting how much a priori knowledge the system has about the task or domain. A simple yet elegant approach is demonstrated by Huber et.al. [9] where the robot has complete knowledge of the sub-tasks performed by the human and it uses the sub-task complexities to predict execution durations. Using a very precise, top-down description of human behavior during an assembly task, Fish et. al. are able to collect detailed statistics about the human on a per-step basis, and use a cognitive model to predict both duration variability over time and error statistics [10]. Tenorth et. al. also observe human performing actions to learn a probabilistic description of the distributions of partially ordered human activities [11]. At the far end of the spectrum is the work of Koppula and Saxena [12] where the actions performed by a human in a given domain is learned from observation 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. They only report the accuracies of the predictions and do not develop an action selection process based upon them. The work presented here also explicitly model possible future sub-tasks sequencings and maintain a probability for each based upon prior info and current observations. But our possible futures are defined by an a priori task description.

The other domain from which prior work is drawn is computer vision, specifically activity recognition, where there is a vast number of approaches to modeling activities composed of sequences of actions. Perhaps the most relevant work is that of Shi et.al. [13] where a Dynamic Bayes Network variant was proposed to recognize partially ordered sequential action. The network encodes activity's structural and temporal information. Inference is done by particle filtering, where the states of particles represent the state distribution of the activity at a particular time. In contrast, the work developed here uses a graphical model that explicitly represents when each action occurs whether it be the past or future. Related to this line of work concerning parsing long activity with complex structure, Albanese et al. [14] uses probabilistic Petri nets to detect events while [15] learns an activity's decomposable structure of "actionlets" with a probabilistic suffix tree; given that data structure, early prediction of sub-action can be done. In [16], Tang et al. demonstrated how to use a variable-duration Hidden Markov Model to learn an action's latent temporal structure and showed it helps to improve detection results in the presence of noisy sensors. Kitani et.al. [17] leverage Hidden variable Markov Decision Processes to learn to predict activity from noisy sensor measurements. Finally Ryoo [18] explicitly develops a dynamic bag of words approach to recognize partially completed action and predict near future action.

III. 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 (Fig. 2).

For the task, the human is instructed to begin building a specific model from the pieces in the bins. They are asked to restrict their bin reaches to one part from a bin at a time. Since the human cannot withdraw from a bin not in reach, this imposes a task constraint which the robot must satisfy for the pair to complete the task. 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.

Though the order in which the human needs the bins is fixed, the robot is allowed to be more flexible. In order to accommodate the possibility of mistakenly removing bins before the human was done with them, the robot might deliver bins in any order or redeliver bins previously removed.

Fig. 2. Example assembly task constructions. The human sits at table and assembles a variety of objects. (a) The parts required for each sub-task are in separate bins which are to be delivered to the human. (b) A few different models the human could make.

IV. Task Representation

Our system models the task as a known sequence of human actions, incorporating duration knowledge, task constraints, and detector observations simultaneously (Sec. IV-A). 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 infers the distribution over when human actions occurred or will occur (Sec. IV-C). Using human action prediction, we can reason about the human's demands on task constraint satisfaction (Sec. IV-D).

A. Linear chains: representing sequential actions

Figure 3 shows a schematic Bayes network of a task consisting of K actions, where g_k^s and g_k^e are random variables representing the starting time and ending time of action k. In this formulation, time will be discrete and for now we assume a maximum fixed total task execution time T, therefore $1 \leq g_k^s < g_k^e \leq T$. In the conclusion we discuss why this is not a limitation.

Our linear chain makes a strong Markovian assumption that the start of each action is dependent only upon the end of the preceding action.[1] We specify the conditional dependency at each node in the network:

$$P(g_k^e | g_k^s) \propto D_k(g_k^e - g_k^s) \tag{1}$$

where $D_k \geq 0$ is the distribution representing the duration of action k. In our experiment, D_k is a truncated normal distribution either learned from training data or explicitly provided. Though we assume that each action starts immediately after the preceding one ends:

$$P(g_{k+1}^s | g_k^e) = \begin{cases} 1 & \text{if } g_{k+1}^s = g_k^e \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

that is not essential as we could have a distribution statistically describing the gap between any given two actions.

We also incorporate gating task constraints which must be true for the human to perform a particular action. This permits the system to reason about the impact of robot actions on human actions; that if a necessary condition for a given human action is not satisfied by the robot at the current time, the human cannot proceed. In our example task, it allows the system to recognize that if a particular bin is not in the workspace, the human cannot have started withdrawing parts from that bin. For a given action k we define an observable constraint C_k: if at time step t_i, the human is supposed to perform action k, but the bin is not available $C_k(t_i) = 0$, then the human has to wait until a later time t_j, $j > i$ when the bin becomes available $C_k(t_j) = 1$. The conditional probability is modified to model that constraint.

The system only incorporates this information with respect to the current time and assumes that all bins will become available when needed after the current time: $C_k(t_i > t_{now}) = 1$. This is effectively making an optimistic assumption on the performance of the robot, that it is able to deliver every needed bin in the future with perfect timing. In the planning section developed later the system considers the delay that will actually occur when delivering bins.

The final piece in our network is action k's measurement Z^k. It is the observation over the whole task — *all T time steps from both past and future*— indicating, for every interval during the task, the likelihood of the observed evidence given the interval describing when action k occurs:

$$P(Z_{1:T}^k | g_k^s, g_k^e) \tag{3}$$

The idea is that all observations about all actions impact the belief as to when a given action occurs. Because the system must consider that an action occurs in the future, it theoretically must consider future observations. Of course, for future observations — observations later than the current time step — only a prior distribution can be assumed. In the next section, we describe model for detecting primitive actions, and the detection score there will be used as an observation likelihood.

A prior on $P(g_1^s)$ is required to perform inference. For simplification purposes, we simply use a uniform distribution over the first several seconds.

B. Action detection

In our task, we define each primitive action to be getting a piece from a bin and assembling it. The start of an action will be when the hand touches the piece inside the bin, and the end will be when the hand reaches for the next piece (which is the start of the next action). The detection of an action k is performed based on the positions of the hands and of

[1]This is assumption prevents us from modeling global effects such as speed - one user might be uniformly faster than another [9]. In practice such a global parameter can be easily accommodated in the inference.

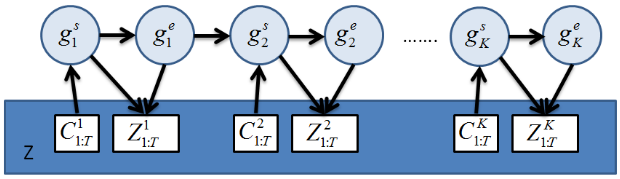

Fig. 3. The Bayes network representing the task structure.

the particular bin $b(k)$ that has the corresponding piece. Let $H_t(b(k))$ be the 3D position of the closest hand to $b(k)$ at time step t and represented in $b(k)$ local frame coordinates. If action k starts at some time step t_i, then $H_{t_i}(b(k))$ is likely to have a distinctive value (learned from training data) independent of the bin's position. We model that position by a Gaussian distribution. Our detectors are therefore driven by detecting such "action start" events where the hand touches the bins:

$$f_h(g_k^s, g_k^e) = N(H_{g_k^s}(bin(k)); \mu_h, \Sigma_h)$$

That is, in this particular experiment, our detectors are a function only of when the action starts g_k^s and not when it ends g_k^e.

The detector above is subscripted with an h - indicating that the detector is functioning correct and correctly observed the action. It is a "hit" in pattern recognition parlance. While this Gaussian component can handle systematic variance in the offset of the hand, it does not represent the possibility sensors completely fail to detect anything meaningful during the start of an action. This sensor unreliability can occur from either a weak detector or from occlusion by the robot or some other body in the workspace. We therefore make use of a second component with uniform value:

$$f_m(g_k^s, g_k^e) = w_m$$

representing a miss. When we are considering the likelihood of a future observation, we also use a uniform distribution, assuming the action detection could happen any time up to T:

$$f_f(g_k^s, g_k^e) = w_f$$

The overall likelihood will be computed as weighted combination of those 3 components:

$$F(g_k^s, g_k^e) = \begin{cases} f_h(g_k^s, g_k^e) + w_m & \text{if } g_k^s \leq t_{now} \\ w_f & \text{otherwise} \end{cases} \tag{4}$$

The probability distribution is then computed by normalizing the observation likelihood:

$$P(Z_{1:T}^k | g_k^s, g_k^e) \propto F(g_k^s, g_k^e) \tag{5}$$

The weights $w_m, w_f > 0$ control both the belief about how the reliability of the sensor and the ability of the sensor to render negative evidence about the state of the human. A relatively high w_m value will indicate low confidence in the sensor and vice versa. A relatively high value for $w_f - w_m$ will mean that low f_h values should be considered as a greater likelihood it will happen in the future, whereas when the two are equal $w_f = w_m$, low f_h values will not increase the likelihood the event happened now.

C. Inference

Our network has a chain-like structure, which allows for efficient inference. The factorization of the entire network is:

$$P(g, Z) = \prod_{k=1}^{K} P(g_k^s | g_{k-1}^e) P(g_k^e | g_k^s) P(Z^k | g_k^s, g_k^e) \tag{6}$$

where $Z = Z^1, Z^2 ... Z^K$ and $g = g_1^s, g_1^e, g_2^s, g_2^e ..., g_K^s, g_K^e$

Given the network with the full conditional probability table computed, we use the message-passing/junction-tree algorithm to perform exact inference (Algorithm 1). In the forward phase, as messages are passed from the beginning to the end of the chain, the probability of each action takes into account observations of all previous actions. In the backward phase, it is the other way around. The final step combines all messages to output the posterior of each action taking into account all observations (including observations of other actions both before and after). The inference finally outputs the posteriors: $P(g_k^s|Z), P(g_k^e|Z)$.

Algorithm 1 Message passing on our network

Input: $P(g_1^s), P(g_{k+1}^s | g_k^e), P(g_k^e | g_k^s), P(Z^k | g_k^s, g_k^e) \forall k$

Note that all these formulas are computed for every value of g_k^s and g_k^e, which are $1 \to T$

[Forward phase]

for $k = 1 \to K$ **do**

$P(Z^k, g_k^e | g_k^s) = P(g_k^e | g_k^s) P(Z^k | g_k^s, g_k^e)$

$P(g_{k-1}^e | Z^{1:k-1}) = \sum_{g_{k-1}^s} P(g_{k-1}^s, g_{k-1}^e | Z^{1:k-1})$

$P(g_k^s | Z^{1:k-1}) = \sum_{g_{k-1}^e} P(g_k^s | g_{k-1}^e) P(g_{k-1}^e | Z^{1:k-1})$

$P(g_k^s, g_k^e | Z^{1:k}) \propto P(g_k^s | Z^{1:k-1}) P(Z^k, g_k^e | g_k^s)$

end for

[Backward phase]

for $k = K \to 1$ **do**

$P(Z^{k+1:K} | g_{k+1}^s) = \sum_{g_{k+1}^e} P(Z^{k+1}, g_{k+1}^e | g_{k+1}^s) P(Z^{k+2:K} | g_{k+1}^e)$

$P(Z^{k+1:K} | g_k^e) = \sum_{g_{k+1}^s} P(g_{k+1}^s | g_k^e) P(Z^{k+1:K} | g_{k+1}^s)$

end for

[Compute posterior by combining the messages]

for $k = 1 \to K$ **do**

$P(g_k^s, g_k^e | Z) \propto P(g_k^s, g_k^e | Z^{1:k}) P(Z^{k+1:K} | g_k^e)$

$P(g_k^s | Z) = \sum_{g_k^e} P(g_k^s, g_k^e | Z)$

$P(g_k^e | Z) = \sum_{g_k^s} P(g_k^s, g_k^e | Z)$

end for

Output: $P(g_k^s|Z), P(g_k^e|Z) \forall k$

D. Human Constraint Satisfaction Demand

Using this inference result which models the human's actions, their probabilistic demand on the bins can be estimated. Let b_j^s and b_j^e be random variables which represent the time at which the demand for bin j starts and ends, respectively. We define b_j^s to be the time at which the last reach from the previous bin ends. This represents the time at which the human starts waiting on the current bin if it's unavailable, or the time they make the first reach for the bin if it is available. We define b_j^e to be the time at which the last reach from the current bin starts, the time when all of the reaches from the current bin have been completed.

V. PLANNING BY ANTICIPATING

When the robot is not busy performing an action, it plans actions which anticipate the human's demands. Since the number of bins which can be in the human workspace is limited, the robot must eventually remove bins in order to deliver others. This is the primary constraint driving robot action decision: a competition between removing a bin before the human is finished with it, and delivering a bin later than they needed it. If the bin is removed too early, it can be redelivered, but generally at the cost of a long wait. On the other hand, we would prefer the robot anticipate the human's needs and keep them from waiting on a bin needed in the near future.

A. Planner Outline

To plan, the robot first heuristically determines which bins will be most needed for delivery in the future. It ranks bins based on their expected starting time $E[b_j^s]$, but penalizes bins which are probably no longer needed ($P[b_j^e < t_{now}] \approx 1$). Here, t_{now} refers to the time at which the inference was performed. The top 3 bins are selected from those not in the human workspace for delivery, or fewer if their expected need was excessively low. Several delivery sequences were generated which deliver those bins in any permutation. For each sequence, the robot plans delivers and removes such that it delivers bins until there are no more empty slots, then alternates removing and delivering.

Each action plan is associated with an optimization problem whose solution space is a schedule S of deliver and remove action times for each step in the plan. We denote S's values as a_j^d and a_j^r to represent the times at which a bin j is delivered or removed, respectively. For both of these actions, we define cost functions $\phi^d(a_j^d, b_j^s, b_j^e)$ and $\phi^r(a_j^r, b_j^s, b_j^e)$, described below (Sec. V-B). We use a constrained nonlinear optimization algorithm (Matlab's *fmincon*) to minimize the sum expected cost for executing each action in the plan at a given time:

$$S^* = \arg\min_S \sum_{a_b \in S} E_{P(b_j^s, b_j^e | a_b)}[\phi] \tag{7}$$

The optimizer enforces the constraint on S that each subsequent action cannot begin until the previous is completed. The action schedule with the lowest sum expected cost is selected for decision making. Based on the timing of the first action in the plan, the robot decides to either perform that action, or wait until the next planning iteration.

We should note that though the sequence of deliveries is known before optimizing, the sequence of removes is not. The optimization function handles this by selecting the bin with lowest removal cost among the bins in the workspace at the time a removal is demanded. We will also point out that the robot continuously re-plans, keeping no history of its previous plans. Thus, plans for actions occurring later than the planning cycle are subject to change.

B. Cost functions

In this section we qualitatively describe our cost heuristics. In general, we try to make the cost equal to squared waiting time to accomplish the goal of reducing both total wait time and longest wait time. Though we make no claim that these functions are optimal, they have shown reasonable results in practice. The cost functions in our planner are developed to account for several mutually exclusive events occurring.

When a bin is not in the workspace and the robot is deliberating whether to deliver it, we consider 4 different cases:

- If the bin demand has not started ($b_j^s > t_{now}$), then if it delivers late ($a_j^d > b_j^s$), then the cost is the squared waiting time from the time the human started demanding the bin to the time it is expected to be delivered.
- If it has not started and delivers early ($a_j^d \le b_j^s$), a small reward is granted to deliver bins sooner when the robot is otherwise idle.
- If the bin demand has already started but not yet ended ($b_j^s \le t_{now} < b_j^e$), then, under the assumptions of our system, the bin has been removed preemptively, so we give it a cost equal to the squared wait from the time the bin was last removed to when it would be redelivered.
- Finally, if the bin demand has ended ($b_j^e \le t_{now}$), we give zero cost.

When a bin is in the workspace and the robot is deliberating whether to remove it, we consider 3 different cases:

- If the bin is removed before its demand has ended ($a_j^r < b_j^e$), then if it removes the bin, we penalize it a constant value, equal to the squared wait time required to remove the bin and deliver it back.
- If the bin is removed after its demand has ended ($a_j^r \ge b_j^e$), we give zero cost
- Finally, if the probability that demand has ended is very low, we give an enormous penalty. This keeps the optimizer from delivering bins superfluously.

VI. EXPERIMENTATION

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 is attempts to assemble a toy whose parts are separated into 4 bins. The human's workspace can only maintain 3 workspace slots, requiring that the robot must eventually remove the first bin the human works on. The first bin is already in the human's workspace when the task starts. We require that the human perform one reach for each part in the bin and there are total of 17 parts which need to be assembled, 6 in bin 1, 2 in bin 2, 1 in bin 3, and 8 in bin 4. Since the time between the last reach into bin 1 and the first reach into the bin 4 is short with respect

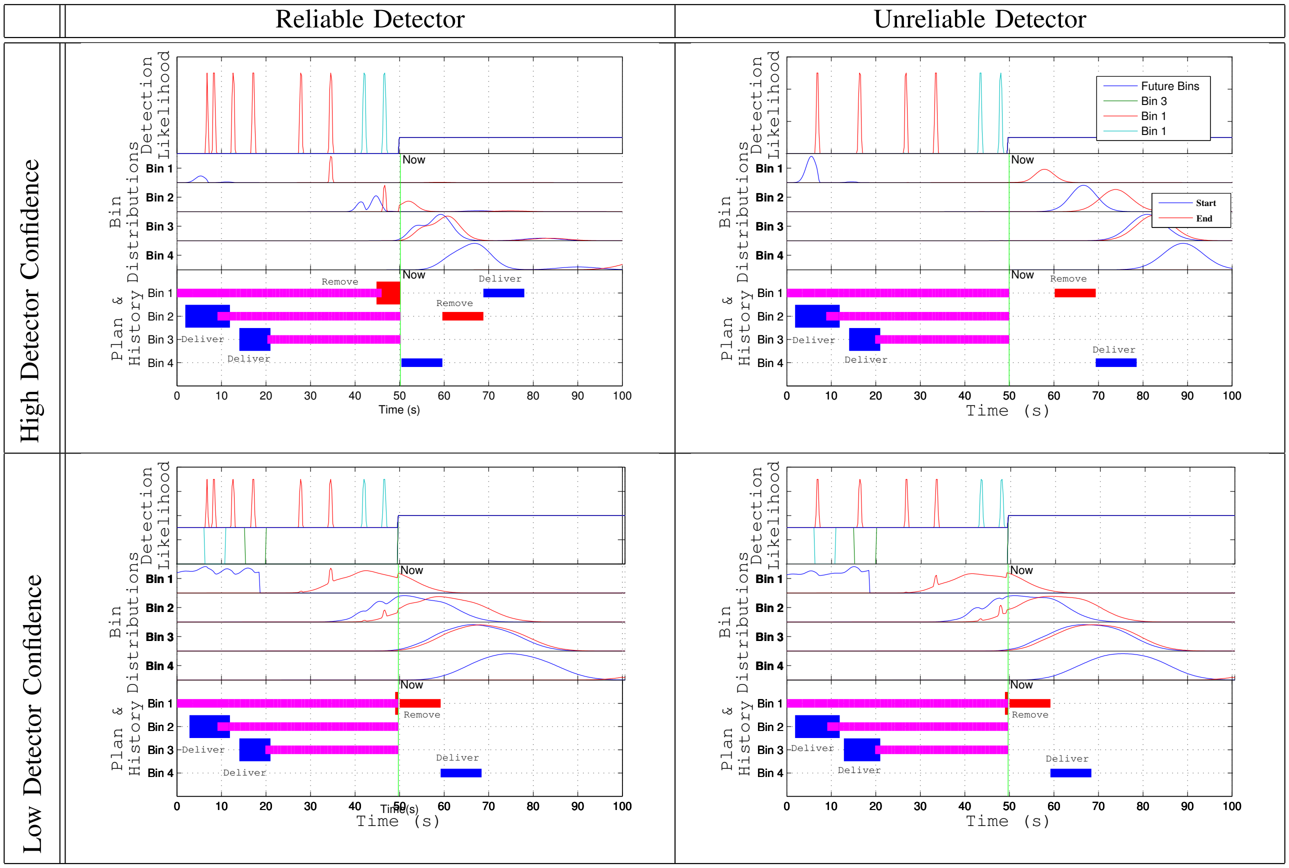

Fig. 4. Examples of the simulation trials detailed in figure 5. Each box contains three plots representing the planning circumstances at the current time t_{now} (the green vertical line), given different simulation detectors and different sensor models. **Within each box: Top:** detection likelihood (Eq. 4) over time; **Middle:** The start and ending distributions for each bin's demands (b_j^s and b_j^e); **Bottom:** The execution history and projected plan of the planner. The thinner magenta bars represent the times each bin was in the human workspace over time. The blue boxes left of t_{now} represent delivers the robot performed in the past, while red boxes represent removals. The boxes to the right represent the robot's best plan of action at the moment.

to the robot's replacement time, the robot must replace bin 1 very soon after its last reach in order to keep the human from waiting on bin 4.

B. Simulation

We developed a simulator which allowed us to evaluate our planner in a controlled environment. The human agent 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 demonstrate the behavior of our system in the face of detector unreliability, we purposefully altered the behavior of the agent to simulate these effects. To simulate detector failure, for two bin reaches, the detector generated no response.

By altering the parameters of the sensor model, w_m, w_f, the system can seamlessly interpolate between relying on its duration model and relying on its detections. We developed 2 sensor models, one which has high confidence in its detector's reliability, and one which has low confidence. We ran $N = 6$ trials each for each of the 2 simulator conditions against the each of the 2 sensor models. To measure performance, we found the total wait time and the total execution time for each run. The execution time is the time between the first action in the task, and the last action.

Figure 4 illustrates a few exemplars of each trial set. The differences between the four conditions can be seen in the Detection Likelihood plots. For both of the reliable detectors, we see 6 peaks for the Bin 1 detector, while we only see 4 peaks in the unreliable detector cases. For the low confidence detectors, we can see that the miss likelihood is much higher than the high confidence ones, representing far more of the density.

These action schedules demonstrate that when your noise model matches reality, your performance is improved. When the high confidence model is applied to a reliable detector, the system is very accurate in its estimation of when the final reach for Bin 1 occurred. This can be seen in the Bin Distributions plots where the distributions have far less entropy than in its low confidence counterpart. Since the demand for Bin 4 is rapidly rising, it wastes no time and is already in the process of removing Bin 1. The low confidence model, on the other hand, excessively distrusts the detector, causing the system to wait about 6s more before swapping.

When the low confidence model is used on the unreliable sensor, the system correctly relies on its duration priors, whose influence can be seen in the wide Gaussian distri-

Simulation Results (s)

	Reliable Det.		Unreliable Det.	
Sensor Model	Wait	Exec.	Wait	Exec.
High Conf.	1.5	102.2	23.6	125.3
Low Conf.	7.8	108.9	10.2	111.8

Fig. 5. Results from a set of $N = 6$ simulated trials for each condition, presenting the average wait times and execution times in seconds for each sensor model and simulation condition. Two sensor models, each with low or high confidence in their sensor reliability, are executed against two different simulator conditions: a reliable detector and an unreliable detector.

butions. However, when the system overestimates the sensor reliability, it ends up waiting for missed detections which will never arrive.

The results from a series of 6 trials for each case are found in figure 5. These statistics again lend support to the fact that while having a better detector can improve performance, matching the system with an appropriate sensor model can improve the performance even more significantly.

C. Human-Robot Experiments

We also ran experiments with a real human-robot collaborative team. The robot was a 6-DOF Universal Robots UR-10 mounted to a steel table with a Robotiq C-model parallel jaw gripper. Above the robot, a webcam was mounted to track the positions and orientations of the bins, affixed with Alternate Reality (AR) tags. Above and in front of the human, an Asus Xtion RGB-D sensor was mounted to sense the behavior of the human. The entire system was 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 linear task we tested in simulation. To track the human collaborator's hands, we used brightly colored surgical gloves and implemented a color blob tracker on the RGB-D sensor. Despite this seemingly reliable detector, frequent occlusion by the robot causes the system to miss several bin reaches, making this a fairly unreliable sensor (Fig. 6).

We ran 5 trials each on the parameters tuned for low and high reliability sensors. The results for each of the trials, sorted by total wait time, can be found in figure 7. These results seem to reaffirm the results obtained in figure 5, that overconfidence in the detector can cause significant performance loss. It seems that twice in the high confidence case, occlusions were infrequent and the detector performed as expected. However, in three others, the detector failed, costing the system performance dearly. Airing on the safe side will keep the system from suffering catastrophic failures at the cost of being consistently slower than the high confidence's best cases.

VII. Discussion and Conclusion

In both simulation and on the real robot, wait time is at its lowest value when the system's confidence in its detector is matched with reality. When the simulation paired a reliable detector with a high confidence sensor model, the

Fig. 6. In the trial experimentation, the robot often occludes the human collaborator's hands. Despite this, we were able to find parameters which kept the wait times consistently low.

Human-Robot Results (s)

Sensor Model	Sorted Waiting Times				
High Conf.	0.0	0.5	24.5	32	35*
Low Conf.	3.5	3.6	4.3	4.5	6.6

Fig. 7. Total wait times from a set of $N = 5$ human-robot trials for a high confidence and low confidence sensor model applied to an unreliable detector. Entries marked with an asterisk demonstrated a preemptive bin removal. Though a high confidence detector can occasionally produce little to no wait time, it can also suffer from severe failures. A low confidence detector, however, can produce consistently decent results.

system was able to exploit this knowledge to reduce the waiting times significantly and become more responsive to the collaborator's actions. The application of the reliable detector to the low confidence model shows the costs of being overly conservative. Likewise, the overconfident detectors were shown in both simulation and in real experimentation that they either become too risky, and mistakenly remove a bin too early, or become delayed, since they are counting on those missed detections to arrive.

For sensing humans using visual sensors, noisy and unreliable detectors are, for many applications, unavoidable. Furthermore, human variability will make incorporating sensors necessary for fluent human-robot collaboration. Thus, being able to gracefully tune the confidence of the robot in it's perception of the state of the world will allow robots to more effectively exploit their sensing and task knowledge and better anticipate the needs of human collaborators.

In this paper we have presented a human-robot collaboration system which leverages both a duration model and a sensor model to account for human variability and noisy or unreliable sensors. By modelling the task probabilistically, we can produce distributions which allow us to make the right decision over many possible futures. Our planner leverages these distributions to weigh the cost of making the costly error of removing a bin early versus the eventual cost of a growing late time. We have evaluated the system on toy cases in both simulation and in a real-world experiment.

We mention that having a fixed execution time T is not a limitation. If T is chosen large enough to cover the entire task the system observes all the actions and reasons about them appropriately. However, the challenge arises that the computational complexity of Algorithm 1 is $O(T^2)$. We are currently developing a variable time resolution method that uses higher temporal precision around the current time and uses coarser intervals further in the past and future. This makes large T computations to remain real-time on conventional hardware.

In future work, we will present a simple extension of this work which reasons over multiple potential paths the human can take. By accommodating human non-deterministic behavior, this system can be applied to a much broader class of task descriptions. In addition, we should perform a more rigorous evaluation of the system. For example, we should show that our model works under a large range of tasks by generating random task models performed by the human, and available to the robot.

References

[1] 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.

[2] H. Goto, J. Miura, and J. Sugiyama, "Human-Robot Collaborative Assembly by On-line Human Action Recognition Based on an FSM Task Model," in *Human-Robot Interaction 2013 Workshop on Collaborative Manipulation*, 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] S. Nikolaidis and J. Shah, "Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy," in *IEEE international conference on Human-robot interaction*, Mar. 2013, pp. 33–40.

[5] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, pp. 109–116, 2004.

[6] P. Dominey, G. Metta, F. Nori, and L. Natale, "Anticipation and initiative in human-humanoid interaction," *Humanoids 2008 - 8th IEEE-RAS International Conference on Humanoid Robots*, pp. 693–699, Dec. 2008.

[7] 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.

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

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

[10] 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.

[11] 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.

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

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

[14] 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.

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

[16] 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.

[17] K. M. Kitani, B. D. Ziebart, J. A. Bagnell, and M. Hebert, "Activity Forecasting," in *European Conference on Computer Vision*, 2012, pp. 1–14.

[18] M. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in *Computer Vision (ICCV), 2011 IEEE International*, no. Iccv, 2011.