Assisting Operators of Articulated Machinery with Optimal Planning and Goal Inference

Ehsan Yousefi[1] and Dylan P. Losey[2] and Inna Sharf[1]

***Abstract*— Operating an articulated machine is a complex and hierarchical task, involving several levels of decision making. Motivated by the timber-harvesting applications of these machines, we are interested in developing a collaborative framework for operating an articulated machine/robot in order to increase its level of autonomy. In this paper, we consider two problems in the context of collaborative operation of a feller-buncher: first, the problem of planning a sequence of cut/grasp/bunch tasks for the trees in the vicinity of the machine. Here we propose a human-inspired planning algorithm based on our observations of the operators in the field. Then, a Markov Decision Process (MDP) framework is provided, which enables us to obtain an optimal sequence of tasks. We provide numerical illustrations of how our MDP framework works. Second is the problem of inferring the operator's goal from the motions of the machine. The goal inference algorithm presented here enables the robot equipped with the planning intelligence to perceive the human's intent in real-time. We evaluate the performance of our goal inference algorithm through a user-study with a feller-buncher simulator. The results show the benefits of our algorithm over a robot that assumes the human is moving to the closest target.**

1. Introduction

Similar to driving a car, operating an articulated machine for pick and place tasks is a hierarchical process, which includes strategical (e.g., route planning) and low-level controls (e.g., cabin rotation) [1]. Accordingly, human-robot cooperation can occur at different levels within this hierarchy, including the high-level strategic and planning tasks. In addition, the behavior of the operator can be described by a *well-structured sequence* of repetitive subtasks [2].

We specifically focus on human-robot collaboration in feller-buncher articulated machines (see Figure 1). These hydraulic mobile robots are employed in timber harvesting industry applications—for cutting trees, picking them up, and then delivering these trees to a storage location. What makes operating these robots challenging is that the articulated machine is very *dexterous*: operators must correctly interact with 9 different inputs to guide the robot arm. Moreover, the tasks that the human and robot must perform are repetitive and hierarchical: the human must make high-level decisions (e.g., which group of trees to cut next) and fine-grained motions (e.g., how to cut and grasp a specific tree).

Within the industrial state-of-the-art, the autonomy of machines for forestry (such as the feller-buncher) is much lower than in other comparable industries, such as mining [3]–[5]. In other words, when humans use these forestry tools they are completely responsible for both the high-level decisions and the low-level motions. Inspired by the recent successes of shared autonomy approaches for assistive robot arms [6]–[11], we here seek to introduce *partial autonomy* for large-scale forestry machines. We separate this partial autonomy into two main parts: *identifying optimal plans* for high-level tasks (i.e., determine the optimal sequence for cutting and delivering trees) and *predicting human goals* for low-level motions (i.e., inferring which tree the human currently wants to reach). Our overall approach to partial autonomy is guided by our observations of expert operators:

Experts in the field often cluster nearby trees, and interact with all the trees in a cluster before moving on.

Although this observation motivates our approach, it is not required: our user studies show that our method extends to inexperienced operators who have less efficient strategies.

In collaborative tasks — like the ones considered in this paper — it is imperative that the human and robot possess a shared mental model of the task process to achieve an effective and mutually understandable collaboration [12]–[14]. Thus, the decision making process should be mutually understandable to the agents in the system. With this insight in mind, we first explore how forestry robots should plan an optimal sequence of high-level subtasks. We formalize this setting as an instance of a Markov decision process (MDP) and the robot solves for the optimal sequence of visiting, cutting, and delivering trees. The key here is our choice of reward function. If we encourage the robot to optimize for the same reward as the human operator, then we can ensure that the human and robot possess a shared mental model of the task. Alternatively, if we select a new reward function (e.g., the amount of fuel used), then we can utilize the resulting MDP output to train and improve human operators.

Our MDP formulation enables the robot to suggest high-level plans to the human operator. To help the human execute those plans (and predict what the human is reaching for), we next introduce a low-level intent inference framework for articulated forestry machines. Here we start with existing cognitive models from robotics and cognitive science [9], and then modify those models to capture and anticipate the human's behavior within articulated machines. To test our low-level inference algorithm, we perform a user study on a simulated feller-buncher machine. The results suggest that our approach accurately predicts which cluster of trees the human is trying to reach.

[1]Dept. of Mechanical Engineering, McGill University, Montreal, QC, Canada, ehsan.yousefi@mail.mcgill.ca, inna.sharf@mcgill.ca

[2]Dept. of Mechanical Engineering, Virginia Tech, Blacksburg, VA, USA, losey@vt.edu

(a) (b)

Fig. 1: (a) Feller-buncher machine; (b) Operator's view of the end-effector.

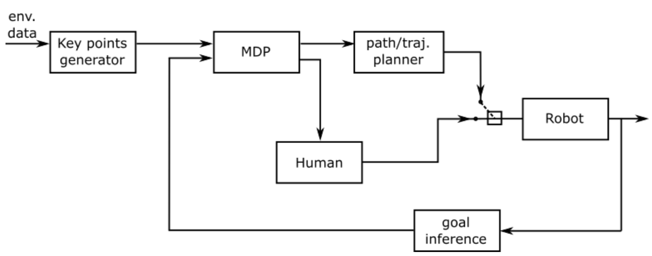


Fig. 2: Closed-loop system for human-robot collaboration.

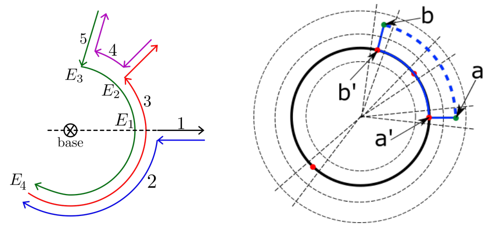
Taking our high-level planning and low-level prediction algorithms together, we propose a partial autonomy approach that (a) assists the operator's decision making process and (b) anticipates the operator's motions. This paper is organized as follows: Section 2 formalizes planning problem, defines a Markov decision process, and introduces the goal inference algorithm. Section 3 contains numerical results to demonstrate the MDP framework, while Section 4 presents the results from our goal inference user study.

2. Problem Formulation

As shown in Figure 2, the proposed holistic framework for human-robot collaboration is comprised of a number of blocks: *key points generator* processes the environment data, clusters the nearby objects (i.e., trees) and provides the key points as goal locations. Next, the *MDP* block provides the optimal sequence of actions to operate in between the key points. This block is a human-inspired autonomous decision-maker, the output of which can directly feed a standard *path/trajectory generator* block and that offers a full autonomous operation. However, there might be occasions when the operator intervenes and changes the thread of actions. The *goal inference* block is designed to infer the human-operated robot's goal, acting like a sensor so that the robot can take over, while simultaneously, the MDP re-plans the optimal sequence of actions from this moment onwards. In case the human wants to retain the control authority, the intelligent robot continues to suggest the optimal next action and senses how the human is proceeding using the inference block. In this paper, we focus on the high-level blocks and provide formulation for the MDP and goal inference.

A. Human-Inspired Path Planning

Here, we introduce a new concept to analyze the human's high-level pattern of behavior while operating the crane of the feller-buncher machine, called the Envelope of Manipulation (E^M). Figure 3a shows a schematic of how an operator carries out path planning between clusters of objects and,



(a) (b)

Fig. 3: (a) Illustration of Envelope of Manipulation (E^M) for a set of key-points, E_1-E_4. The latter is the storage point. The arcs indicate the end-effector paths between key points and the straight line segments indicate the motion of the end-effector to the tree cluster; (b) Sample of human-adaptive cell decomposition. The proposed motion planning method is based on cell decomposition and is adaptive to human's preferred path planning.

hence, the top view of the envelope. In addition, a set of key points, namely, E_n at the storage point, and E_1, E_2 up to E_{n-1} are introduced. The curve connecting these key points is referred to as the *envelope* in this work, and its simplest realization is a circle with radius r_E. Hence, the motion of the arm along E^M is in fact a one-DoF motion which is affected by rotating the cabin about the vertical axis with the arm in a fixed configuration. The proposed motion planning method is based on cell decomposition [15] and is adaptive to human's preferred path planning; we call it *human-adaptive cell decomposition* with the key points as *via points*, as illustrated in Figure 3b.

A typical tree-cutting operation sequence proceeds as follows and is illustrated in Figure 3a. Starting at key point E_1, the operator follows path no. 1 to reach out to a cluster in that direction. This path is followed back to E_1, and then following path no. 2 to E_4. Since the location of E_4 is known in advance, during the time spent traversing path no. 2, the operator is able to select the next target location and the path to reach there through the envelope. This process repeats itself at successive key points, with the operator planning at least one step ahead. The idea of action primitives can also be leveraged here. In particular, in the context of the envelope of manipulation, the process of divergence from and convergence to the set of key points $\mathcal{E} = \{E_1, E_2, E_3, ..., E_n\}$ can be considered as an action *primitive*, called $\mathcal{P}_1$. Additionally, motion along the path on the envelope between the key points is another action primitive, $\mathcal{P}_2$. Therefore, for the scenario of Figure 3a, the sequence or thread of action primitives can be expressed as an ordered set:

$$\mathcal{P} = \{\mathcal{P}_1^{E_1},\ \mathcal{P}_2^{\widehat{E_1E_4}}, \mathcal{P}_2^{\widehat{E_4E_2}}, \mathcal{P}_1^{E_2},\ \mathcal{P}_2^{\widehat{E_2E_3}},\ \mathcal{P}_1^{E_3},\ \mathcal{P}_2^{\widehat{E_3E_4}}\}, \quad (1)$$

where $\mathcal{P}_1^{E_i}$ is the action primitive $\mathcal{P}_1$ at key point E_i, and $\mathcal{P}_2^{\widehat{E_iE_j}}$ is the action primitive $\mathcal{P}_2$ between the key points E_i and E_j. From the hierarchical point of view, the set $\mathcal{P}$ can be considered as human's low-level operational controls. The operator's high-level decision-making process also involves constructing the set $\mathcal{E}$. As already noted, the next key point selection is carried out during the execution of the previous action primitive. The efficiency and skill level of an operator are directly related to the ability to perform these actions simultaneously, and therefore, to minimize some measure of energy expended during these actions.

In our MDP formulation, we will employ the capacity for Maneuverability, denoted as CfM, which is the remaining actuation capacity for a human intervention before actuator saturation occurs [16]. In particular, one of the constraints in the tree-cutting operation involves the capacity for maneuverability of the arm, CfM_{arm}, which ensures that the operation happens within the reachable space of the robot end-effector. Moreover, the capacity for maneuverability of the end-effector, CfM_{ee}, measures the remaining capacity of the end-effector claws to contain objects and/or the remaining actuation capacity in the end-effector.

B. MDP Formulation

Suppose that the set of key points is defined as $\mathcal{E} = \{E_0, E_1, ..., E_n\}$, with E_0 and E_n denoting the starting and the storage points, respectively. An example topology of a set of key points and the Envelope of Manipulation are shown in Figure 4a. An agent, which can be a driver or a driver-in-the-loop AI, is expected to cut/grasp a set of objects at each of these key points and unload them at the storage point. Multiple unloadings in a thread of actions are also possible. The constraints are the maximum Capacity for Maneuverability of the robot arm, $\overline{CfM}_{arm}$, and of the end-effector, $\overline{CfM}_{ee}$. The first constraint is enforced *a priori* by limiting the objects under consideration to lie within the reachable space of the arm. We now define the elements of the MDP representation of the problem as follows:

State, s. A state $s \triangleq (s_1,\ s_2,\ s_3)$ is defined as a tuple with three elements, as follows:

- s_1: Key point number, E_i, or node number.
- s_2: CfM_{ee}. This is a value between 0 and 1, which changes in finite increments as objects are grabbed by the end-effector. We discretize it by dividing the range $[0,1]$ into N segments and assign an integer value to each segment.
- s_3: State of the environment defined as a tuple with binary entries associated with the key points. For example, if the agent has only visited E_1, the associated state element value is $s_3 = (1,0,...,0)$. We assume that all objects within the cluster associated with that key point are collected once a particular key point has been visited. This assumption can be relaxed in future work but it is in line with our current field observations.

Action, a. An action is defined as the direction of the cabin rotation, i.e., $dir = CW$ or CCW, and the destination key point number. Therefore, an action j at key point i is represented by $a_i^j = (dir, E_j)$ for $j = 1,...,n$ and $j \neq i$.

Reward function, $R(s,a,s')$. We define the following reward function that breaks the task into five parts:

$$\begin{aligned} R(s,a,s') &= R_1 + R_2 + R_3 + R_4 + R_5 \\ &= -r_E(\phi^{s_1 \to s'_1})(a_1 + a_2 C_1 s_2) + C_2 \mathbf{1}(isEnd) - C_3 \mathbf{1}(s'_1 = E_n) \\ &\quad - C_4 \mathbf{1}(failure) + C_5(s'_2)\mathbf{1}(!failure\ \&\ s'_1 \neq E_n), \end{aligned} \tag{2}$$

where rewards R_1 to R_5 are defined, term-wise, as follows: cost of consumed energy to move the robot with the end-effector and payload (R_1), a successful operation of a thread (R_2), consumed energy for an unloading process (R_3), any failure (R_4), and any successful step within a thread (R_5). Also, $\phi^{s_1 \to s'_1}$ is the angle of rotation between the designated key points, and a_1 and a_2 are the coefficients related to the energy consumed to move the robot with or without the end-effector payload (grasped objects). Moreover, C_1 is a measure of accumulated mass in the end-effector, which depends on the density of the objects; C_2 is the reward of finishing a thread, which happens when all of the objects are collected. If an unloading process happens, C_3 is a measure of consumed energy for the unloading process. C_4 is a negative reward incurred if an unsatisfactory state is visited such as any state violating the physical constraints of the model, for example, $CfM_{ee} < 0$, which is considered to be a *failure*. C_5 is the reward of collecting any new object(s) in case of non-failure. The function $\mathbf{1}(.)$ is defined such that if the condition inside the parentheses is true, it returns 1, otherwise it returns 0.

Transition probability, $T(s,a,s')$. The probability of transitioning to state s' from state s by taking action a. At each key point E_i, $i = 1,...,n$, considering both CW and CCW are possible directions of motion there are $L = 2(n-1)$ options.

Policy. The purpose of solving the MDP problem is to find a mapping from states to actions, $\pi = S \to A$. The following intuitive policy structure is suggested for the agent as the *baseline policy*:

- We define $P|_n$ as the probability of transitioning to the storage point E_n by taking any action $a = (-, E_n)$. The probability $P|_n$ depends on s_2 (i.e, CfM_{ee}) and is assumed to be uniformly distributed over all possible transitions to the storage point, thus taking the form:

$$P|_n = \begin{cases} 1, & isEnd \\ max(2/L, 1 - (s_2|_s)/(\overline{CfM}_{ee}), & \text{otherwise} \end{cases} \tag{3}$$

- For the remaining $L-2$ decisions, we assume a uniform probability distribution as follows:

$$P = (1 - P|_n)/(L-2). \tag{4}$$

It is straightforward to demonstrate that $\sum_a P(a|s) = 1$. This baseline policy is not likely to be optimal; however, it is a meaningful policy given the semantics of our problem. In order to find the optimal policy π_{opt}, we improve the baseline policy via recursive policy evaluation and value iteration (5) until the state values induced by the policy remain unchanged within a threshold ε:

$$V_\pi^{(t)}(s) = \max_{a \in \mathcal{A}(s)} \sum_{s' \in \mathcal{S}} T(s,a,s')[R(s,a,s') + \gamma V_\pi^{(t-1)}], \tag{5}$$

where γ is a discount factor, $\mathcal{A}$ is the set of all actions, and $\mathcal{S}$ is the set of states.

C. Goal Inference of Human-Operated Robot

Effective human-robot collaboration also requires the robot to understand the human's goal through the lens of the operator's actions as they are reflected in the robot outputs, in other words, through the information *leaks* [17] from the human operator. Thus, we are now concerned with the problem from the perspective of the intelligent robot.

TABLE I: Envelope of manipulation and parameter values used for MDP simulation results

input data/parameters	value
set of key points $\mathcal{E}$	{0, 1, 3, 2, 4, 5, 6}
number of objects at each key point	{0, 4, 1, 5, 2, 1, 0}
ang. loc. of key points wrt x-axis [deg]	{90, 30, 100, 170, 220, 270, 0}
discretization factor N	5
discount factor γ	0.9
a_1, a_2, C_1-C_5	1, 100, 2000, 10000, 500, 10000, 10

Suppose that the human intends to move the robot to a goal location $g \in G$, where g is one of the key points and $G \subset \mathcal{E}$ is the set of discrete goals that an operator might intend to visit. The objective here is to infer the human operated robot's goal, where we assume a first-order mental model [18]. In this regard, we follow Luce's axiom of choice [19] to distribute the probability of choosing a particular goal g^* among G and use the Boltzmann model of noisily-rational behavior [11], [12], [20], where $p_d(g^*) \propto \exp(\beta V_{g^*})$. This assumes the human operator is trying to reach their intended goal by noisily optimizing a reward function such as [9]:

$$V_g = \gamma^{||\mathbf{x}_g - \mathbf{x}_h||} U - c(\gamma - \gamma^{||\mathbf{x}_g - \mathbf{x}_h||})/(1-\gamma), \qquad (6)$$

where $\mathbf{x}_g$ contains the goal locations (i.e., key points location), $\mathbf{x}_h$ is the location of the robot's end-effector, $U, c \in \mathbb{R}^+$ are constants, γ is the discount factor, and the parameter β characterizes how likely the operator's behavior deviates from the rational path as an element of noise. One caveat here is that in the context of following the envelope of manipulation, the operator might move the robot close to a key point (potential goal), while only intending to move across it in order to reach another goal key point. Intuitively, the operator shows an intention of slowing down the arm angular speed ($\dot{\theta}_h$) below some threshold value $\bar{\dot{\theta}}_h$ when the operator approaches a particular goal location with intention of carrying out the cutting operation. This statement can be mapped to a value (V_{vel}) using a membership function employed in fuzzy logic [21]. We have used a bell-shaped function of the form:

$$V_{vel}(\dot{\theta}_h; \bar{\dot{\theta}}_h) = 1/(1 + |\dot{\theta}_h / \bar{\dot{\theta}}_h|^{2\bar{\dot{\theta}}_h/3}). \qquad (7)$$

Thus, the probability of slowing down around any goal is proportional to the obtained value V_{vel}, i.e., $p_{vel} \propto V_{vel}$. The joint probability of a goal is therefore defined by $p(g^*) = p_d(g^*) p_{vel}$. Throughout a thread of actions, we employ the above-mentioned algorithm to infer the intended goal.

3. MDP Simulation Results

We consider the envelope of manipulation with the configuration depicted in Figure 4a, defined precisely with parameters in Table I and reward function parameters listed in the same table. It is assumed that each object takes up 20% of the end-effector capacity and thus, for the configuration chosen, more than one unloading is necessary. Using (5), we iteratively obtain the optimal value function V_π induced by the optimal policy, and the algorithm terminates when the value function remains unchanged, i.e., $||\Delta V|| \leq \varepsilon$, depicted in Figure 4b. The optimal policy, π_{opt}, for our scenario is presented in Table II. The optimal thread of actions, therefore,

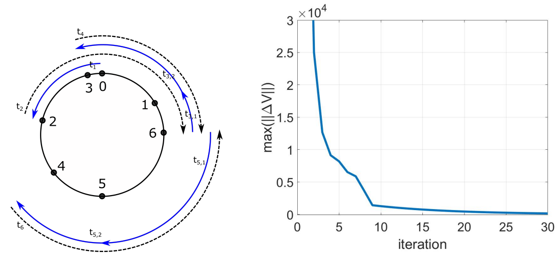


Fig. 4: (a) Envelope of manipulation: scenario for MDP analysis (solid black curve) and illustration of optimal thread of actions from MDP solution. Proximity of arcs to the envelope of manipulation corresponds to the ordering of actions. Blue arcs represent action primitive $\mathcal{P}_1$. End arrows indicate a stop to execute action primitive $\mathcal{P}_2$ (either to cut/grasp or to unload). Dashed arcs indicate return to the storage point.; (b) MDP value iteration error, $\max(\|\Delta V\|)$, for each iteration, using (5) and considering $\varepsilon = 10^{-6}$ as the convergence threshold.

TABLE II: Optimal policy results

state, $s = (s_1, s_2, s_3)$	**action**, $a = (dir, E_i)$
(0, 5, (0, 0, 0, 0, 0, 0))	('ccw', 2)
(2, 0, (0, 0, 1, 0, 0, 0))	('cw', 6): unload
(6, 5, (0, 0, 1, 0, 0, 1))	('ccw', 1)
(1, 1, (1, 0, 1, 0, 0, 1))	('ccw', 3)
(3, 0, (1, 1, 1, 0, 0, 1))	('cw', 6): unload
(6, 5, (1, 1, 1, 0, 0, 1))	('cw', 5)
(5, 4, (1, 1, 1, 0, 1, 1))	('cw', 4)
(4, 2, (1, 1, 1, 1, 1, 1))	('ccw', 6): unload
(6, 5, (1, 1, 1, 1, 1, 1))	none

is "$E_0E_2E_6E_1E_3E_6E_5E_4E_6$", as can be deduced from the first column of Table II. Based on these results, for the specified form of the reward function and the baseline policy, the MDP we designed identifies the optimal policy, i.e., the optimal thread of actions for our application. We recognize that the MDP formulation presented here is not unique and further improvements can be made [22], [23]. However, the novelty of this part of our work lies in providing a methodology to study path planning and, subsequently, goal inference for Feller-Bunchers and other articulated machines. It is also important that these optimal results are aligned with the expert behaviors we observed in the field.

4. Goal Inference Results from Human Participant Study

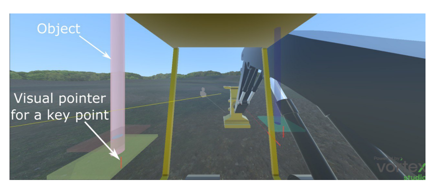
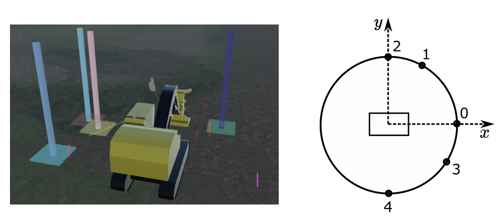


Fig. 5: Virtual world with visual pointers representing the key points



(a) **(b)**

Fig. 6: (a) Virtual world with visual pointers representing key points: perspective view of the scene; (b) Configuration of key points.

Our setup is a virtual world with a robot (a model of Feller-Buncher machine) with several objects (trees) in the vicinity, as shown in Figure 5, with visual pointers indicating the key points. This simulator platform was developed in *Vortex Studio* [24], in combination with Python scripts and it provides high-fidelity dynamics simulation and realistic visualization. Using this platform, we are able to record human operator's input data (joystick signals) as well as robot motion as output data, through an interface to python scripts for real-time data processing.

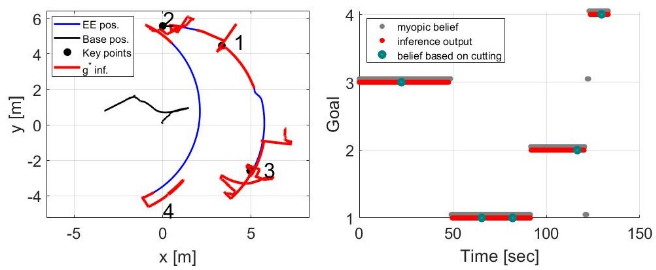
As a proof-of-concept and to showcase the capabilities of our simulation setup, we conducted a study involving 11 participants from McGill community, with little or no previous exposure to the type of material and the applications discussed in this paper. Each participant was asked to carry out a tree harvesting operation in the world scene shown in Figure 6a using a Logitech Gamepad F310 as an input device. A schematic of the key points for the virtual world in Figure 6a is shown in Figure 6b; we note that there are two objects for cutting near key point 1 unlike the others. The user's task was to move the robot arm and/or base so as to align the end-effector with a desired object for the automatic cut/grasp emulation to occur and repeat this process until all objects have been cut and bunched at the storage location. The user was allowed to unload the object(s) grasped at any time at the designated storage location (E_4). The users were not aware of the inference algorithm during the operation, nor of any optimal thread of actions. They were only introduced to the key points, as indicated by the visual pointers.

Our inference algorithm attempts to obtain the intended goal by observing the position of a point on the robot's end-effector (the saw at the bottom), the arm (cabin) angular rate, and the knowledge of key points' locations. Intuitively, we would expect the values of parameters γ, β, U, c, and $\dot{\bar{\theta}}_h$, employed in (6)-(7) of the goal inference algorithm to be user dependent and, in practice, learned from the behaviour of the user. However, for the present study, in the absence of a learning algorithm, we converged by trial and error to the values $\gamma = 0.5$, $\beta = 0.9$, $U = 10$, $c = 1$, and $\dot{\bar{\theta}}_h = 30$ deg/sec; these parameters are used for all participants. We compare the performance of our inference algorithm to two other schemes: the first is the *myopic belief* over human intention, which assumes that the closest goal is the intended goal [7]. The second baseline is a *gold standard* where we measure the human's true goals. Here we record the human's thread of actions across the entire interaction, and then mark the key points where they stopped to pick-up, or unload the trees. Note that this approach — although accurate — can only be used to predict the human's goals after they have completed the task, and as such is not suitable for online assistance. We refer to this idealized approach as *belief based on cutting*. Moving forward, we expect our approach to lie between the myopic baseline (i.e., predicting the closest key point) and the gold standard (i.e., looking back at their actual goals after the task is complete). We note that *belief based on cutting* misses out on times where the human changes their mind along the way, and assumes that the human always wanted to reach their recorded key points.

We observed vastly different trajectories and sequencing among the users for this relatively simple setting. We attribute this disparity to two causes: 1) *Edge cases* — these occur when a user changes their mind after moving close to a goal, or when a user takes extra time to think and plan when they are near a goal; 2) *Driving habits* — At the early stages of inferring a goal and also at the very last stages of leaving that goal, participants stopped make decisions, this hesitation caused our inference output to slightly oscillate. Some participants drove very slowly: in this case, our inference scheme reduced to a myopic result (since the human's speed was always below the inference threshold). In reporting our results, we consider the entire thread of actions for each user, since the individual user's data set reflects that user's driving habits and allows us to demonstrate how our algorithm with fixed parameter values works for different users and their specific driving habits. From the observations of 11 participants who completed a thread of multiple loadings and unloadings, we categorize our results into two groups:

Group I: Our algorithm performs accurately for the entire thread of actions for this group comprised of seven out of 11 users. For some users, we observed short intervals of uncertainty at the beginning and at the end of the inference zones. A sample robot trajectory is shown in Figure 7a for one user from Group I. In this figure, the blue curve shows the path of the end-effector in the $x-y$ plane, while the red segments are overlayed over parts of the path where the inference algorithm assigns a goal to that particular segment. The inference results for this user for their entire thread of actions are included in Figure 7b. In this Figure, the gray lines depict the myopic belief over human intent, the green circles show where and when cutting took place and the red lines show the output of our inference algorithm. If the probability of a goal is greater than 0.5, we assume that goal is the human's intended target. An important episode occurs after the 4th cutting, between $t = 117$ and $t = 130$ seconds, when the operator is heading for key point #4, as they pass through key points #1 and #3. The beliefs for the four goal locations over this episode are plotted in Figure 8, from which we observe that our inference algorithm does not assign a high probability to any of the key points until #4, in contrast to the myopic result.

Group II: For this group of four out of 11 users, our algorithm sometimes predicts that the human is going to the closest goal (i.e., our approach reverts to the myopic baseline) even when the human really wants a more distant target. This occurs for the reasons discussed earlier (edge cases and driving habits); however, our approach never predicts the wrong goal when the human is getting ready to pick-up or put-down a tree (i.e., *inference* matches *belief based on cutting*). The paths and the inference data over the entire thread for one user in Group II are shown in Figure 9. Again, a noteworthy episode occurs between $t = 6$ and $t = 57$ seconds, the beliefs for which are plotted in Figure 10. The user on the way to key point #2 slows down around key point #4, but no operation happens there. Our algorithm incorrectly infers #4 to be a goal (i.e., a myopic prediction) but corrects itself much faster than the myopic scheme. Since a cutting ultimately happens near key point #2 at $t = 57$ seconds, we can assume that it was the intended goal. However, prior to that, the operator moves the robot to key point #1 (still without an operation) and even approaches key point #3, as indicated with the myopic belief, finally returning to #2 and cutting an object. In this thread of actions, our algorithm performs well overall.



(a) **(b)**

Fig. 7: Sample user results from Group I: (a) end-effector and base paths: blue indicates end-effector path in $x-y$ plane, red is overlayed on segments where goal is inferred; (b) inference and belief over a thread: myopic belief (gray), occurrence of cutting (green circles), our inference algorithm (red).

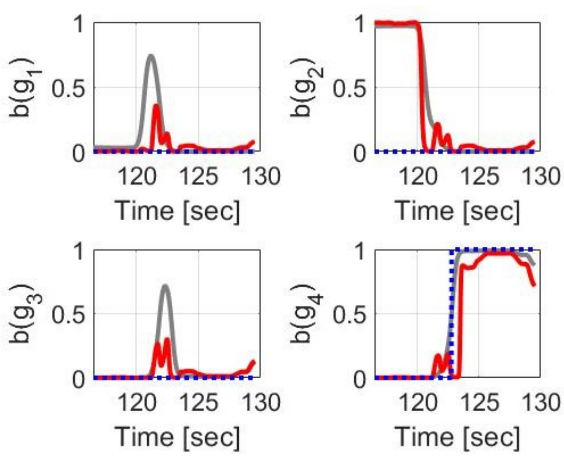
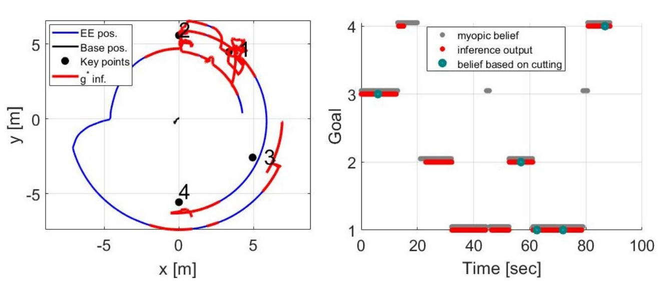


Fig. 8: Goal inference results for sample user from Group I: time history of goal inference (red) vs belief over goal (gray) for time period between $t = 116.5$ and $t = 129.5$ seconds.



(a) **(b)**

Fig. 9: Sample user results from Group II: (a) end-effector and base paths: blue indicates end-effector path in $x-y$ plane, red is overlayed on segments where goal is inferred; (b) inference and belief over a thread: myopic belief (gray), occurrence of cutting (green circles), our inference algorithm (red).

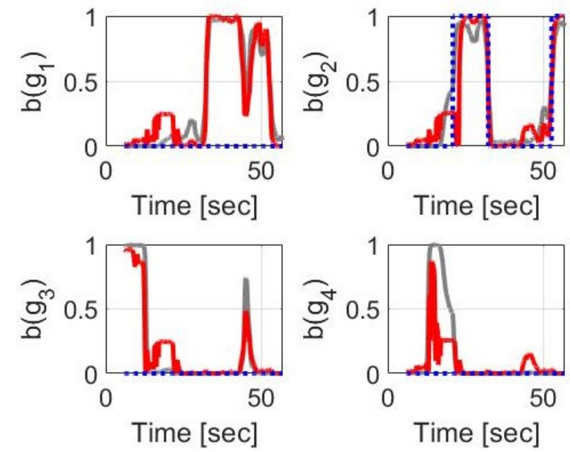


Fig. 10: Goal inference results for sample user from Group I: time history of goal inference (red) vs belief over goal (gray) for time period between $t = 5.967$ and $t = 56.85$ seconds.

In all, given the diversity of users' driving habits, the fact that the algorithm employed robot data only, and that the same set of parameter values was used for all users, the proposed inference algorithm works well.

5. Conclusion and Future Work

In this work, we introduced a framework for assisting humans operating dexterous, high-dimensional, and articulated machines. Specifically, in the context of tree harvesting tasks, we introduced an MDP formulation that enables the robot to autonomously plan a sequence of *high-level* actions (e.g., determining which trees to cut and when to deliver them). We encouraged the robot to optimize a reward function that considers the energy consumption: the resulting plan provides high-level guidance and suggestions to human operators. In order to help the human execute this plan, we next proposed an intent inference algorithm that identifies the human's *low-level* goal (i.e., which tree they want to reach next). Our approach was based on existing cognitive models, and we adapted these models for timber-harvesting applications. Our user study results with a simulated Feller-Buncher machine demonstrated that our approach is more accurate than simply assuming that the human wants to reach the closest object.

Our next step is to extend this formulation into a shared control framework. We envision combining both the high-level planning and low-level inference to assist human operators as they work with articulated machines in forestry tasks. The application of this framework and the ideas presented within, however, are not limited to timber harvesting applications and can be applied to articulated machinery in other domains, such as mining and construction.

Acknowledgment

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN), and Summer Undergraduate Research in Engineering (SURE) programs. The authors also acknowledge the valuable contributions of Brice Diomande and Denis Kartachov to the development of the simulator.

REFERENCES

[1] C. Guo, C. Sentouh, J. C. Popieul, and J. B. Haué, "Predictive shared steering control for driver override in automated driving: A simulator study," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 61, pp. 326–336, 2019.

[2] S. Westerberg, *Semi-Automating Forestry Machines*. PhD thesis, Umeå University, Sweden, 2014.

[3] O. Lindroos, P. La Hera, and C. Häggström, "Drivers of advances in mechanized timber harvesting - a selective review of technological innovation," *Croatian Journal of Forest Engineering*, vol. 38, no. 2, pp. 243–258, 2017.

[4] R. Fukui, T. Niho, M. Nakao, and M. Uetake, "Imitation-based control of automated ore excavator: improvement of autonomous excavation database quality using clustering and association analysis processes," *Advanced Robotics*, vol. 31, no. 11, pp. 595–606, 2017.

[5] J. Song and I. Sharf, "Time Optimal Motion Planning with ZMP Stability Constraint for Timber Manipulation," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 4934–4940, 2020.

[6] D. P. Losey, H. J. Jeon, M. Li, K. Srinivasan, A. Mandlekar, A. Garg, J. Bohg, and D. Sadigh, "Learning latent actions to control assistive robots," *Autonomous robots*, pp. 1–33, 2021.

[7] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.

[8] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[9] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, "Goal inference improves objective and perceived performance in human-robot collaboration," 2018.

[10] S. Jain and B. Argall, "Probabilistic human intent recognition for shared autonomy in assistive robotics," *ACM Transactions on Human-Robot Interaction*, vol. 9, no. 1, pp. 1–23, 2019.

[11] A. Bobu, D. R. R. Scobee, J. F. Fisac, S. S. Sastry, and A. D. Dragan, *LESS is More: Rethinking Probabilistic Models of Human Behavior*, p. 429–437. New York, NY, USA: Association for Computing Machinery, 2020.

[12] L. Chris, R. Rebecca, C. L. Baker, J. B. Tenenbaum, and R. R. Saxe, "Goal Inference as Inverse Planning," in *Proceedings of the Annual Meeting of the Cognitive Science*, 2007.

[13] E. Yousefi, *An Investigation of the Effects of Human Dynamics on System Stability and Performance*. M.Sc. thesis, Bilkent University, 2018.

[14] T. Yucelen, Y. Yildiz, R. Sipahi, E. Yousefi, and N. Nguyen, "Stability limit of human-in-the-loop model reference adaptive control architectures," *International Journal of Control*, vol. 91, pp. 2314–2331, oct 2018.

[15] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[16] E. Eraslan, Y. Yildiz, and A. M. Annaswamy, "Shared control between pilots and autopilots: Illustration of a cyber-physical human system," 2019.

[17] B. Mutlu, F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita, "Nonverbal leakage in robots: Communication of intentions through seemingly unintentional behavior," in *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, HRI '09, (New York, NY, USA), p. 69–76, Association for Computing Machinery, 2009.

[18] A. Tabrez, M. B. Luebbers, and B. Hayes, "A Survey of Mental Modeling Techniques in Human–Robot Teaming," *Current Robotics Reports*, vol. 1, no. 4, pp. 259–267, 2020.

[19] R. D. Luce, "The choice axiom after twenty years," *Journal of Mathematical Psychology*, vol. 15, no. 3, pp. 215–233, 1977.

[20] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI Conference on Artificial Intelligence*, vol. 8, pp. 1433–1438, 2008.

[21] D. J. Dubois and H. Prade, *Fuzzy Sets and Systems: Theory and Applications*. Academic Press, 1980.

[22] D. K. Grady, M. Moll, and L. E. Kavraki, "Extending the Applicability of POMDP Solutions to Robotic Tasks," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 948–961, 2015.

[23] S. Feyzabadi and S. Carpin, "Multi-objective planning with multiple high level task specifications," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 5483–5490, 2016.

[24] CM-Labs. `https://www.cm-labs.com/vortex-studio/software/modular-software-architecture-simulation/editor/`.