Mixed Observability MDPs for Shared Autonomy with Uncertain Human Behaviour

Clarissa Costen , **Marc Rigter** , **Bruno Lacerda** and **Nick Hawes**
Oxford Robotics Institute, University of Oxford, UK
{clarissa, mrigter, bruno, nickh}@robots.ox.ac.uk

Abstract

Shared autonomy allows humans and AI operators to work towards a common goal. Typically, shared autonomy systems are modelled by combining a single model for human behaviour, and a model for the AI behaviour. In this paper, we attempt to provide a richer human model, which accounts for variation in performance due to factors that are not directly observable. Our shared autonomy system will maintain a belief over the unobservable factors, and update its belief as they make observations. The new belief is used to decide who should operate the shared autonomy system. We show that using our model with a richer human representation results in better performance than using a simplistic human model.

1 Introduction

Shared autonomy (SA) systems allow cooperation between human operators and robot controllers working towards a shared goal [Jansen *et al.*, 2016]. The use of SA platforms allows the human operator to relinquish control and reduce their workload. Self-driving cars are an example of this [Basich *et al.*, 2020]. In this paper, the autonomous driver controls the car until the safety is compromised. The SA system aims to minimise the human intervention required while maintaining safety. However, the system presented in [Basich *et al.*, 2020] does not consider the competence of the human driver, and only evaluates the competence of the autonomous driver. Instead, the SA system assumes that a human driver is always safer than the autonomous driver. An alternative scenario in SA is where the autonomous system is assisting an imperfect human user [Cubuktepe *et al.*, 2019]. The human starts with control of the system, but the robot planner overrides when the human strategy conflicts with safety measures.

In both of these systems described above, the choice between the human and the robot controller can be modelled as a Markov Decision Process (MDP). To model this choice as an MDP, models of both the human operator and the robot controller are required. The MDP can then compare the models of the operators to make an action choice. Such an approach assumes that the human operators' behaviour can be described by a single model. This does not consider the fact that the human operators' abilities can vary depending on factors outside of the SA system. Examples of these factors include their individual ability, fatigue levels and their workload. Many of these factors are not directly observable, so one cannot assume to have access to the true underlying human state. In this paper we propose a method to include the uncertainty over the human state, and hence over their performance, into the SA system model.

We are interested in modelling the variation of performance in human operators. In particular, we would like to examine situations where the state of the human operator is not directly observable. For example, situations when a human operator's performance is affected by fatigue or workload. However, the agent would not be able to directly measure these factors. We propose to model such situations using a Mixed Observability MDPs (MOMDPs) [Ong *et al.*, 2009]. The state space in a MOMDP is split between observable and hidden states. We use this feature to define the hidden states as the human operator's states which are not directly observable.

The main contribution of this paper is posing a SA system as a MOMDP, which allows us to build a richer human model. We empirically show that modelling a SA system as a MOMDP rather than a MDP yields better performance. We outline how to model a SA system using a MOMDP, and present an algorithm to solve the MOMDP. To motivate our approach, we introduce a simple example MOMDP, which encodes a series of tasks. Each task can either done by a robot controller or a human operator. At the start of a run, the state of the human operator is unknown to the agent. We use this model to demonstrate how we model a SA system with a MOMDP.

In our experiments, we apply our SA model to two scenarios. The first scenario is a surveillance problem, which was previously tackled in [Feng *et al.*, 2016]. The original paper modelled the human operator becoming fatigued after performing n_f actions. We extend the model to consider uncertainty over when the human enters a fatigued state, which is not directly observable. The second scenario we consider is when a known AI and an unknown player are working together to play the game Angry Birds. The identity of the unknown player is the hidden state factor in this scenario.

2 Related Work

There are broadly two views to shared autonomy systems. One is where the human steps in to assist an autonomous system to achieve a goal [Basich *et al.*, 2020] [Rigter *et al.*, 2020] [Duchetto *et al.*, 2018] [Abdel-Illah *et al.*, 2010]. The other is where the autonomous system assists the human operator to achieve a goal [Cubuktepe *et al.*, 2019] [Feng *et al.*, 2016] [Anderson *et al.*, 2009].

When the SA system has a human involved to recover the system when the robot controller has failed, the system typically consider the human to be a perfect controller. For example, [Basich *et al.*, 2020] focuses on modelling the different levels of competence an autonomous driver has in any given situation. Their system aims to maximise efficiency while minimising the human assistance needed. [Rigter *et al.*, 2020] uses reinforcement learning to teach an autonomous controller how to execute tasks through human demonstration. The cost of the human time required to teach the robot is weighed against the time for the human to recover a failed robot. [Abdel-Illah *et al.*, 2010] treats the human operator as a supervision unit, who takes over when the system is too complex for the AI. All of these systems assume the human performs perfectly.

The other application of a shared autonomy system is where the autonomous system helps the human to achieve their goal. [Cubuktepe *et al.*, 2019] considers a scenario where the human operator is not aware of their surroundings. The human operator will make uninformed decisions, and their performance is modelled as a Markov Chain (MC). The robot controller is fully informed about the surroundings their behaviour is modelled as a MDP. The paper aims to blend the human operator's actions and the robot's actions, creating a shared control scenario. We are more interested in how interactions between the human and the system can inform the agent about future choices. [Feng *et al.*, 2016] models a SA system where the human operator has multiple profile states. The profile states are defined by the fatigue level of the human and their current workload, and they are associated with different success probabilities for the task in the SA system. The paper models the transitions between the performance states deterministically. The SA system is modelled using a MDP, where the agent chooses between a human and an autonomous driver. Here, the agent aims to minimise the number of failed tasks in the SA system.

We use MOMDPs to model the SA system. MOMDPs were originally proposed as a model for planning problems. [Ong *et al.*, 2009] outline the advantages of using MOMDPs over Partially Observable MDPs (POMDP) to model robotic tasks. The size of the hidden state space is smaller in a MOMDP than a POMDP. This means the time taken to compute policies is shorter. [Ferrari *et al.*, 2017] use a POMDP keep track of the mental state of a participant in a human-robot cooperation task. Examples of how a MOMDP can be used to model an uncertain environment is shown in [Ong *et al.*, 2010].

3 Preliminaries

3.1 Markov Models

A Discrete Time Markov Chain (MC) is a stochastic model, which we will use to model the performance of an operator.

Definition 1 (MC)**.** *A MC is defined by the tuple $\langle S, s_0, T\rangle$, where:*

- *S is the finite set of the possible states in the system;*
- *s_0 is the initial state of the MC;*
- *$T : S \times S \to [0, 1]$ is the transition probability function from a state s to s'.*

Markov Decision Processes (MDPs) extend MCs to consider action selection.

Definition 2 (MDP)**.** *A MDP is a tuple $\langle S, s_0, A, T, R\rangle$, where:*

- *S is the finite set of the possible states in the system;*
- *s_0 is the initial state of the MDP;*
- *A is the finite set of action choices that can be made by the agent;*
- *$T : S \times A \times S \to [0, 1]$ is the transition function, mapping the probability of going to state s' when action a was taken in state s;*
- *$R : S \times A \to \mathbb{R}$ is the reward function, mapping the reward collected when taking action a in state s.*

An MDP is solved to return an optimal policy. An optimal policy maps a state to the optimal action to maximise the rewards collected.

A mixed observability MDP (MOMDP) is a MDP where a subset of the state factors cannot be directly observed.

Definition 3 (MOMDP)**.** *A MOMDP is defined by the tuple $\langle S_o, S_h, s_o, A, O, b_0, T, \Phi, R\rangle$, where*

- *S_o is the set of the observable state factors;*
- *S_h is the set of the hidden state factors. The total state space of the MOMDP is $S = S_o \times S_h$, and a single state is defined by the tuple (s_o, s_h);*
- *s_0 is the initial observable state of the MOMDP;*
- *A is the finite set of actions;*
- *O is the finite set of observations;*
- *b_0 is the initial belief distribution of the agent. A belief distribution $b_i(s_h)$ gives the probability of the agent being in a hidden state s_h at time step i;*
- *$T : S \times A \times S \to [0, 1]$ is the transition function;*
- *$\Phi : S \times A \times O \to [0, 1]$ is the observation function;*
- *$R : S \times A \to \mathbb{R}$ is the reward function.*

We use the sampling-based search algorithm POMCP [Silver and Veness, 2010] to solve MOMDPs. The POMCP algorithm returns the optimal history-dependent policy. History is defined by the sequence of actions and observations made up until the current state. The optimal policy maps the history to the optimal action. We use the POMCP algorithm to solve the MOMDP, as it allows us to deal with the curse of dimensionality [Kaelbling *et al.*, 1998], which afflicts computing exact solutions for MOMDPs.

4 MOMDPs as models of Shared Autonomy systems

4.1 Problem Formulation

Let $D = \{d_1, d_2, \cdots d_N\}$ be the set of operators that can control the system. The SA agent has an action choice of $A_{SA} = \{a_1, a_2, \cdots a_N\}$, where the action a_i puts the operator d_i in control. The SA system has a set of environment states S^E, which defines the problem the SA system is trying to solve. It also has a reward over the states, and the goal is to maximise that reward by choosing who takes the action.

The operator d_i has L_i performance profiles. A performance profile determines how an operator will act in the environment state, s^E. For example, an expert performance profile would model an operator with a high success rate when attempting the tasks. The performance profiles make up a profile state space $X^i = \{\chi^i_1, \chi^i_2, \cdots, \chi^i_{L_i}\}$ in the operator model. From the example above, if the operator d_i has a high success rate, they will be in the expert profile state χ^i_j. In this section, we formally defined each component of the SA system and then formalise the shared autonomy MOMDP.

4.2 Performance Profile Model

A performance profile model describes how an operator d^i in profile state χ^i_j would behave if they were in control of the SA system.

Definition 4 (Performance Profile Model)**.** *The performance profile model is defined as the MC $\mathcal{MC}^i_j = \langle S^E, s^E_0, T^i_j \rangle$, where:*

- *S^E is the environment state space for the SA system;*
- *s^E_0 is the initial environment state for the SA system;*
- *$T^i_j : S^E \times S^E \to [0, 1]$ represents the probability that operator d^i changes the environment state from s^E to $s^{E'}$, given that they are in profile state χ^i_j.*

4.3 Operator Model

We would like to model the behaviour of the operator in the SA system. At each time step of the SA system, the operator d^i could either be chosen by the agent to perform an action or not be involved in controlling the system. MC $\mathcal{O}^i_{active}$ models the dynamics of the operators' performance profile when they are selected to control the SA system. MC $\mathcal{O}^i_{dormant}$ models the dynamics of the operators' performance profile when they are not involved in controlling the system.

Definition 5 (Active Operator Model)**.** *The active operator model for operator d_i is defined by the MC $\mathcal{O}^i_{active} \langle S^i, s_0, T^i_{active} \rangle$, where:*

- *the state space $S^i = S^E \times X^i$ is defined by the environment states of the SA system and the possible profile states for the operator;*
- *$s_0 = (s^E_0, \chi^i_0)$ is the initial environmental and profile state;*
- *$T^i_{active}((s^E, \chi^i_j), (s^{E'}, \chi^i_{j'}))$ is the probability of the current environment s^E and profile χ^i_j transitioning to $s^{E'}$ and $\chi^i_{j'}$, given the SA system is controlled by operator d_i.*

$$\begin{aligned} T^i_{active}((s^E, \chi^i_j), (s^{E'}, \chi^i_{j'})) = \\ Pr(s^{E'}|s^E, \chi^i_j) \cdot Pr(\chi^i_{j'}|s^E, \chi^i_j) = \\ = T^i_j(s^E, s^{E'}) \cdot Pr(\chi^i_{j'}|s^E, \chi^i_j), \end{aligned} \tag{1}$$

where $T^i_j(s^E, s^{E'})$ is defined by the performance profile model $\mathcal{MC}^i_j$, and $Pr(\chi^i_{j'}|s^E, \chi^i_j)$ is the probability of the operator in profile state χ^i_j and environment state s^E changing to profile state $\chi^i_{j'}$.

The transition probability for the profile states are dependent on the environment state, as tasks can affect the operator in different ways. For example, an operator who is remotely driving a vehicle in a dark, narrow and hazardous environment is more likely to go into a tired profile state than an operator driving through a bright and clear path.

We consider the scenario when the operator i has not been selected by the agent to control the SA system.

Definition 6 (Dormant Operator Model)**.** *The dormant operator model is defined by the MC $\mathcal{O}^i_{dormant} = \langle X^i, T^i_{dormant} \rangle$, where:*

- *the state space X^i is the set of profile states for operator i.*
- *$T^i_{dormant}(\chi^i_j, \chi^i_{j'})$ gives the transition probability of operator i moving to profile state $\chi^i_{j'}$ from χ^i_j when the system is controlled by a different operator.*

The dormant operator model does not include the task state space S_o, as the operator is not engaged with the SA system. Hence, the environment state of the SA does not influence the operator's profile state dynamics.

The active operator model defines the evolution of the environment states and the operator's profile states when the operator is in control of the SA system. The dormant operator model defines the change in the operator's profile state when they are not in control of the system. As the operator is not directly interacting with the SA environment, the environment states are not factored into the dormant operator model.

4.4 MDP as a Model of the SA system

The SA system when there are N operators can be expressed as a MDP.

Definition 7 (SA-MDP)**.** *The SA system can be modelled as an MDP $SA\text{-}MDP = \langle S, s_0, A, T, R \rangle$, where:*

- *$S = S^E \times X^1 \times X^2 \times \cdots X^N$, where S^E is the environment state factors for the SA system, and X^i gives the profile state factors for the ith operator;*
- *s_0 is the initial state of the system;*
- *$A = \{a_1, a_2, \cdots a_N\}$, which allows the agent to choose the operator;*
- *T is the transition function defined by Equation 2. When the agent chooses an action a_i, the operator i takes control of the system. Therefore, the task state transition*

from s^E to $s^{E'}$ and the operator i's transition from profile state χ_u^i to $\chi_{u'}^i$ is defined by the active operator model $\mathcal{O}_{active}^i$. All of the other operators are not in control for this action, so they are dormant. Therefore, their transition probabilities are dependent their dormant operator models.

$$T((s^E, \chi_p^1, \chi_q^2, \cdots, \chi_r^N), a_i, (s^{E'}, \chi_{p'}^1, \chi_{q'}^2, \cdots, \chi_{r'}^N)) = T_{active}^i((s^E, \chi_u^i), (s^{E'}, \chi_{u'}^i)) \times \prod_{\substack{k=1 \\ k \neq i}}^{N} T_{dormant}^k(\chi_p^k, \chi_{p'}^k); \tag{2}$$

- $R : S \times A \to \mathbb{R}$ *is the reward function.*

4.5 MOMDP as a Model of the SA system

In the MDP model for the SA system, the current profile state for every operator is known. However, this may be unrealistic, as the parameters defining the profile states may not be observable. Let us assume there is one operator whose profile states are not observable. For simplicity, we will define a SA system where there are two operators. However, the formulation of the SA system can be extended to any number of visible operators and any number of hidden operators.

We consider a SA system with two operators, where one operator is a human and the other is an autonomous system. They have profile state factors X^h and X^a. The SA system agent is unable to observe the profile states for the human operator. Therefore, the SA system can be modelled as a MOMDP, $\mathcal{MO}_{SA}$.

Definition 8 (SA-MOMDP)**.** *Our MOMDP modelling a SA system is defined by the tuple $\mathcal{MO}_{SA} = \langle S^E \times X^a, X^h, (s_0^E, \chi^a), A, O, b_0, T_{MO}, \Phi, R \rangle$, where*

- $S^E \times X^a$ *are the observable states;*
- X^h *are the hidden states;*
- (s_0^E, χ_0^a) *is the initial observable state;*
- $A = \{a_h, a_a\}$ *is the set of actions;*
- $O = S^E \times X^a$ *is the set of observations;*
- *the initial belief distribution b_0 is over the hidden profile states for the human operator, X^h;*
- *the transition function T_{MO} is defined by:*

$$T_{MO}((s^E, \chi_i^a, \chi_j^h), a_a, (s^{E'}, \chi_{i'}^a, \chi_{j'}^h)) = T_{active}^a((s^E, \chi_i^a), (s^{E'}, \chi_{i'}^a)) \times T_{dormant}^h(\chi_j^h, \chi_{j'}^h) \tag{3}$$

and

$$T_{MO}((s^E, \chi_i^a, \chi_j^h), a_h, (s^{E'}, \chi_{i'}^a, \chi_{j'}^h)) = T_{active}^h((s^E, \chi_j^h), (s^{E'}, \chi_{j'}^h)) \times T_{dormant}^a(\chi_i^a, \chi_{i'}^a). \tag{4}$$

If $A = a_a$, the transition function is Equation 3. The state transition probability is defined by the autonomy operator active model, $\mathcal{O}_{active}^a$. The human is dormant, so the human's profile transition probability is defined by $\mathcal{O}_{dormant}^h$.

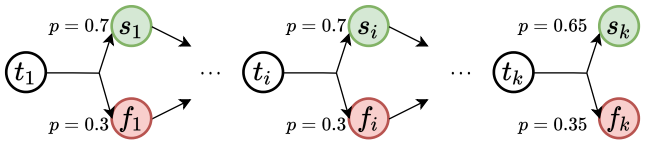

Figure 1: MC for the robot performance profile χ_1^{robot}.

If $A = a_h$, the transition function is Equation 4. The state transition probability is defined by the human operator active model, $\mathcal{O}_{active}^h$. The autonomy is not in control, so the autonomy's profile transition probability is defined by $\mathcal{O}_{dormant}^a$;

- $$\Phi(o, (s^E, \chi^a, \chi^h)) = \begin{cases} 1 & \text{if } o = (s^E, \chi^h) \\ 0 & \text{else} \end{cases} \tag{5}$$

  is the observation function;
- $R : (S^E \times X^a \times X^h) \times A \to \mathbb{R}$ *is the reward function.*

4.6 Example: A Simple SA system

To illustrate the concepts introduced above and the benefits of considering hidden human states, we present a simple example of a SA system. The SA system environment is a series of k tasks. At the start of the first task, the agent is in a environment state t_1. The agent chooses between the autonomous system and the human operator to attempt the task. The operator can either end up in the success state, s_1, or the fail state f_1. This process is repeated k times. When the agent reaches a fail state, a negative reward is collected. The 1st to the $k-1$th task have the same difficulty, while the kth task is set to have a higher failure rate for both operators. The negative reward collected at the failure state for the k th task is also set to be much higher. While the behaviour of the robot controller is fully known, the human could either be a novice or an experienced operator. The agent does not know the identity of the human.

The robot controller has one performance profile, so $X^{robot} = \{\chi_1^{robot}\}$. The MC for a robot in the profile χ_1^{robot} is shown in Figure 1. The 1st to the $k-1$th tasks have a 0.7 probability of entering the success states, while the kth task has a 0.65 probability of entering the success state. As there is only one possible profile for the robot controller to be in, the robot controller model is equal to the performance profile model for χ_1^{robot}.

The human operator has two performance profiles, $X^{human} = \{\chi_{nov}^{human}, \chi_{exp}^{human}\}$. χ_{nov}^{human} is the profile state a human operator would be in if they were a novice, while χ_{exp}^{human} is the profile state for an experienced human operator. For the novice performance profile, the 1st to $k-1$th tasks have a 0.6 probability of going to the success state, and the kth task has a 0.55 probability of going to the success state. For the experienced performance profile, the 1st to $k-1$th tasks have a 0.8 probability of going to the success state, and the kth task has a 0.75 probability of going to the success state. The MCs for the profiles in X^{human} will have the structure of Figure 1, with the aforementioned transition probabilities.

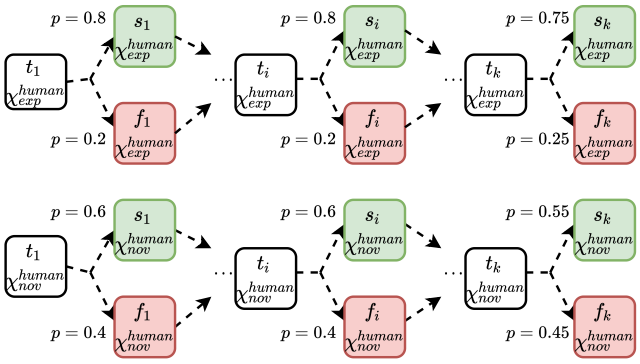

Figure 2: MC $\mathcal{O}^{human}_{active}$ for the human operator when they have control of the SA system.

The two profiles for the human operator are used to form the human operator MC. We do not include any transitions between the two profile states, and the MC $\mathcal{O}^{human}_{active}$ is shown in Figure 2. The MC $\mathcal{O}^{human}_{dormant}$ has two states $\{\chi^{human}_{nov}, \chi^{human}_{exp}\}$, and there are no transitions between the two states.

The human operator model and the robot controller model are combined to form the MOMDP $\mathcal{MO}_{simple}$ for our simple SA system. For simplicity, we outline the MOMDP when $k = 2$. $\mathcal{MO}_{simple}$ is defined by:

- the observable environment states, $S^E = \{t_1, s_1, f_1, t_2, s_2, f_2\}$. The robot's profile state space X^{robot} is not included, as there is only one state, so the profile state is redundant;
- the initial observable state is $s^E_0 = t_1$;
- the hidden states, $S_h = X^{human} = \{\chi^{human}_{nov}, \chi^{human}_{exp}\}$;
- the action choices, $A = \{a_{human}, a_{robot}\}$;
- the observations $O = S^E$;
- the initial belief distribution:

$$b_0(\chi^{human}_{nov}) = b_0(\chi^{human}_{exp}) = 0.5; \tag{6}$$

- the observation function, $\Phi(o, s^E) = \delta(o, s^E)$;
- the transition functions, as shown in Figure 3;
- the reward function, R. $R = -5$ when the agent visits a state with $S^E = f_1$, and $R = -20$ when the agent visits a state with $S^E = f_2$.

The MOMDP $\mathcal{MO}_{simple}$ is solved using the POMCP algorithm. The history-dependent policy is shown in Figure 4. The optimal policy when the agent is in the initial state t_1 is to take the action a_{human}. If the human operator takes the SA system to the success state f_1, the optimal action at t_2 is action a_{human} again. Alternatively, if the human operator takes the SA system to the fail state f_1, the optimal action at t_2 is the action a_{robot}. The optimal policy demonstrates that the MOMDP formulation allows the agent to gather information about the human operator is the first task, and the collected information is used to make an informed choice in the second task.

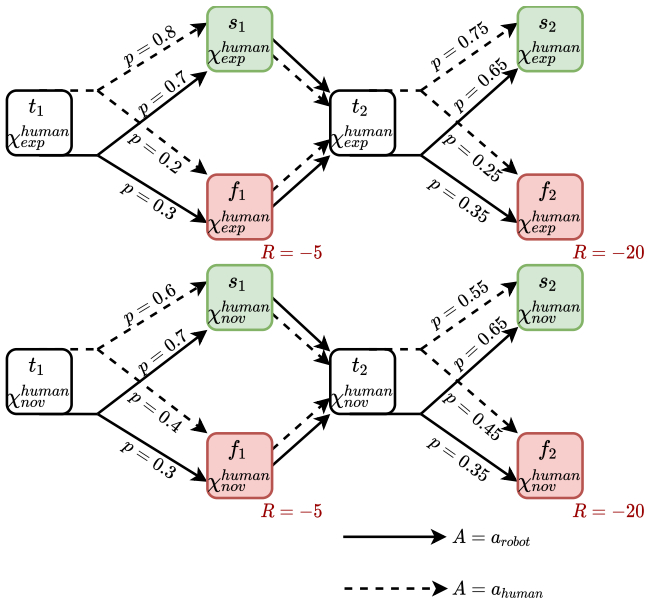

Figure 3: MOMDP $\mathcal{MO}_{simple}$ for the simple SA system.

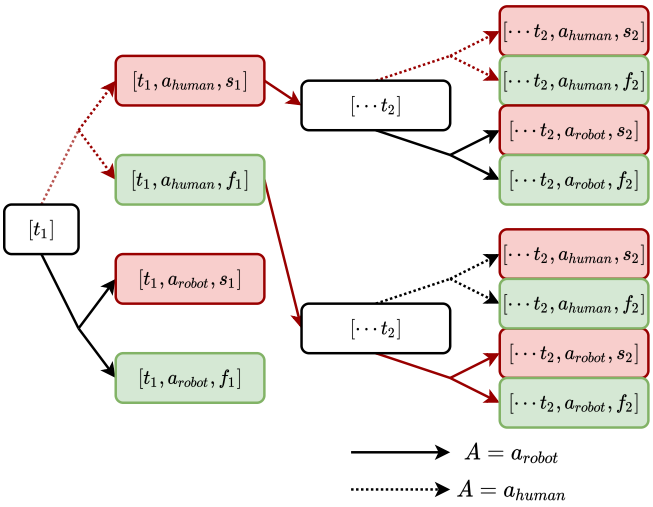

Figure 4: The optimal policy for the simple SA system is shown in the red lines. The transitions between the success/fail states to the next task are deterministic, so are not represented here in full.

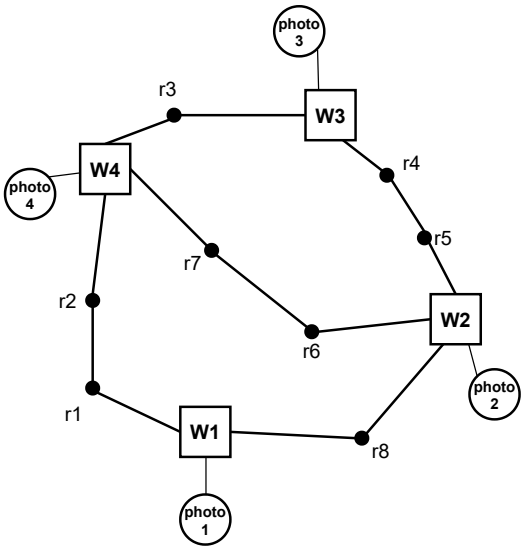

Figure 5: Map of the world the UAV travels around

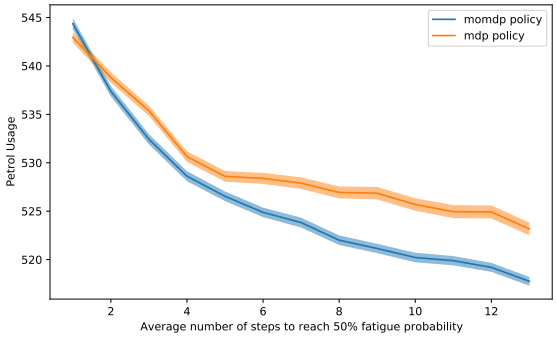

Figure 6: Petrol Usage of the UAV when they complete the surveillance task using the MDP policy v.s. the MOMDP policy

5 Experiments

5.1 Surveillance Model

[Feng *et al.*, 2016] considers a SA problem where a human or an autonomous system remotely controls an Unmanned Aerial Vehicle (UAV) to complete a surveillance task. The paper models how fatigue and workload affect the success rate of a task completed by a human. The paper models the SA system as an MDP, and the number of actions completed by the human is counted and stored as part of the state space. The model uses the number of actions completed by the human to determine the fatigue level. When the number of actions counted is less than n_f, the human is classified as normal. When the number of actions counted greater or equal than n_f, the human is classified as fatigued.

We considered a surveillance task where the SA system flies to all the waypoints in a map. At each waypoint, the UAV will attempt to take a "good" photo of the waypoint. If a good photo is not taken, the UAV will fly around and attempt to take a good photo until it does. The normal human, the fatigued human and the autonomous system all have different probabilities of taking a good photo of a waypoint. The task aims to visit all the waypoints while minimising the amount of petrol used. We adapted the map used in [Feng *et al.*, 2016] to create the map in Figure 5 for our experiment. The UAV flying around a node to take a good photo used 20 units of petrol, while flying between nodes uses 60 units of petrol.

The paper modelled the human operator as a MDP, where the human would transition to the fatigued state after n_f actions by the human. We set the probability of a normal human taking a good photo of a waypoint to 0.9, and the probability of a fatigued human taking a good photo to 0.3. The probability of the autonomous system taking a good photo at any waypoint was set to 0.6. The SA-MDP for a deterministic human was solved to return a optimal policy mapping states to actions.

We applied our MOMDP model to this surveillance task problem. The human normal/fatigue states formed the hidden state space in our MOMDP, $\mathcal{MO}_{UAV}$. The method [Feng *et al.*, 2016] used to determine the human fatigue state does not consider individual variations in the number of steps completed before the human operator becomes fatigued. We model the human operator such that a normal human taking control of the system has a $p = 1-0.5^{\frac{1}{n_f}}$ probability of entering the fatigue state. Therefore, after n_f steps completed by a human operator, the cumulative probability of them being in the fatigued state is 0.5.

$\mathcal{MO}_{UAV}$ has an observable state space $S_o = S_w \times P_1 \times P_2 \times P_3 \times P_4$. S_w is the set of waypoints the UAV can be at, as shown in Figure 5. $P_i = \{0, 1\} \forall i \in \{1, 2, 3, 4\}$ gives the state of the photo taken at W_i. If the UAV has not taken a good photo at W_i yet, $P_i = 0$ and vice versa. The initial observable state is $S_o = (W_1, 0, 0, 0, 0)$. We assume the human operator is in the normal state at the start, so the initial belief is $b_0(\chi^{human}_{norm}) = 1.0,\ b_0(\chi^{human}_{fatigue}) = 0.0$. At each node, the action choice allows the agent to choose the operator in control of the next step, so $A = \{a_{human}, a_{auto}\}$. We solved $\mathcal{MO}_{UAV}$ using the POMCP algorithm. This gave us a history-dependent policy.

We produced the MDP policies from the paper's model and the MOMDP policies from our MOMDP model when the value of n_f is varied between 1 and 13. We ran the policies on the MOMDP model 5000 times, and we recorded the petrol used in each run. The results of this are shown in Figure 6. When $n_f \geq 2$, the MOMDP policy resulted in less petrol being used by the UAV than the MDP policy. As n_f increased, the difference in petrol usage between the MDP and the MOMDP policy increased.

5.2 Angry Birds

We considered the game Angry Birds as a SA system, where two players can take turns playing the game. Angry Birds is a puzzle game, where the player uses a catapult to hit "pigs" hidden in a structure. The player is given a set number of "birds" to catapult, and the aim is to maximise the number of pigs hit. The birds are shot sequentially, and each shot can be taken by a single player. In the future, we will test our framework with a human controlling an autonomous system.

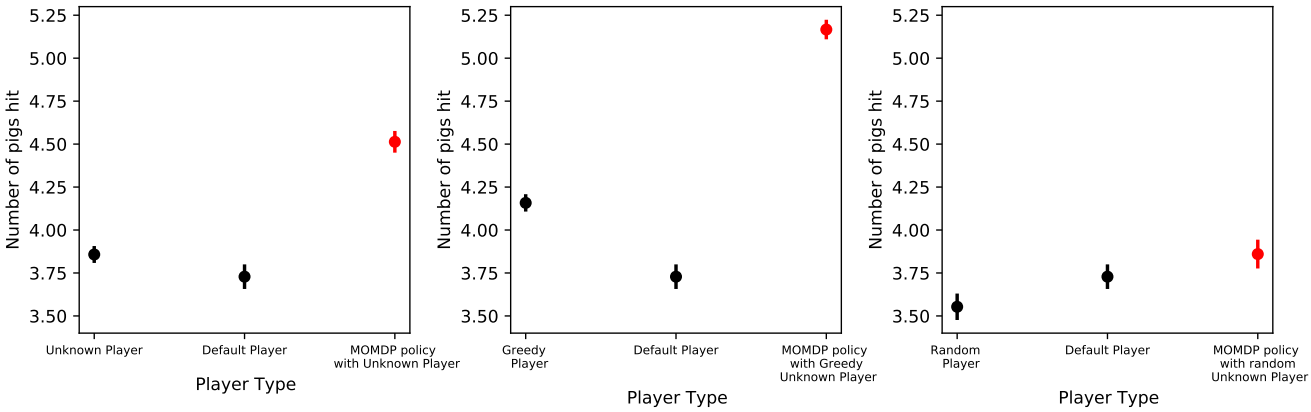

Figure 7: Average number of pigs hit by each agent

To build towards this, we have decided to use Angry Birds as a testbed as it allows us to easily bring human operators to work alongside an AI.

In this version, the two players are a known player and an unknown player. The known player is an automated player which considers all the possible shots to hit the pigs and chooses an unblocked trajectory. If there are no clear shots possible, the known player will randomly choose a pig to shoot at.

The unknown player could either be a random AI player or a greedy AI player. The unknown player has an equal probability of being either player. The identity of the unknown player does not change during a game. The random player is an AI player which randomly chooses a pig, and aims at it. No consideration on objects blocking the trajectory is made. The greedy player is an AI player made by Datalabs [Borovička *et al.*, 2014], which won the Angry Birds AI competition [Renz *et al.*,] in 2013 and 2014.

We model the SA Angry Birds game with our MOMDP model. The number of birds and pigs left defines the observable environment state factors, and at the start of the game there are five birds and six pigs. For example, after two shots, and one pig has been hit, the game is in the observable state $s_o = (\text{birds} : 3,\ \text{pigs} : 5)$. The hidden state space is the set of identities for the unknown player, $\{\chi^{unknown}_{random}, \chi^{unknown}_{greedy}\}$, and the initial belief distribution is set to $b_0(\chi^{unknown}_{random}) = b_0(\chi^{unknown}_{greedy}) = 0.5$. At each shot, the agent chooses an action from the set $A = \{a_{known}, a_{unknown}\}$. We collected the results of the known, unknown random and greedy players playing 150 rounds each. We used the results to create the MC for each profile. The transition probabilities in the MCs were used in the MOMDP. The MOMDP was solved using the POMCP algorithm to give us a history-dependent policy.

The game was played with the unknown player being random and greedy 150 times each following the MOMDP policy. The average number of birds hit is shown in Figure 7. An unknown player with an equal probability of being a random or greedy player hit 3.86 ± 0.05 pigs and the default player hit 3.73 ± 0.07 birds in a game. When the unknown player and the default player take turns according to the MOMDP policy, the mean number of birds hit is 4.51 ± 0.06.

The results can be split into two cases, random and greedy unknown players. The greedy player has a mean of 4.16 ± 0.05 birds hit per game. In the SA game, if the unknown player is the greedy player, the mean number of birds hit is 5.17 ± 0.06. The random player hit 3.55 ± 0.08 birds in a game. When the unknown player is the random player, the SA game has 3.86 ± 0.08 birds hit in a game.

We can compare the results from the MOMDP policy to when the SA system was modelled using a MDP. The transition probabilities for the random and greedy players are combined to create a single profile for the unknown player. The SA system is modelled as a SA-MDP, where the agent chooses between a known player and a unknown player for each shot. We solved the MDP using value iteration to get a policy mapping the current pig and bird states to the optimal action. The comparison of the results from using the MDP policy to the MOMDP policy can be seen in Figure 8. When the unknown player and known player are working together in the SA system, following the MDP policy hits 4.32 ± 0.06 birds, while following the MOMDP policy hits 4.51 ± 0.06 birds. Therefore, modelling the SA system in Angry Birds as a MOMDP result in a higher number of birds hit than modelling the system with a MDP.

6 Conclusion

We have presented a method for modelling variations in operator performance in a SA system using MOMDPs. Firstly, we modelled operators with multiple profile states, where each profile state encodes how the operator will act in the environment state. We considered a scenario where the agent cannot directly observe an operator's profile state. The unobservable operator's profile states defined the hidden states in the MOMDP. The agent held a belief over the hidden profile states and updated the belief using action-observation pairs. The MOMDP was solved using a POMCP algorithm to give us a history-dependent optimal policy.

We applied our MOMDP model to two SA systems, a

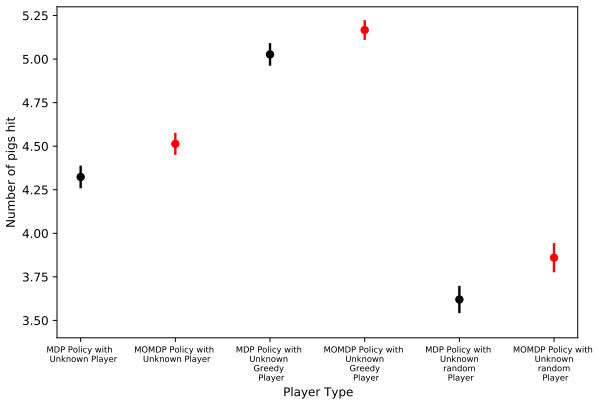

Figure 8: Average number of birds hit in a SA environment when following the MOMDP policy or the MDP policy. The results from the MDP policy are shown in black, and the results from the MOMDP policy are shown in red.

surveillance task and a computer game. We compared our MOMDP policy to a MDP policy used in previous papers. In both of the SA systems, we found that our MOMDP policy outperforms the MDP policy with significance.

To further this work, we will replace the unknown players in the Angry Birds game with human participants. We would also like to consider parametrizing the operator model over the profiles. We hope that this can capture the behaviour that does not neatly fit into a single profile state.

Acknowledgments

This work was supported by the Defence Science and Technology Laboratory, and the EPSRC Programme Grant 'From Sensing to Collaboration' (EP/V000748/1).

References

[Abdel-Illah *et al.*, 2010] Mouaddib Abdel-Illah, Zilberstein Shlomo, Beynier Aurelie, and Jeanpierre Laurent. A decision-theoretic approach to cooperative control and adjustable autonomy. *Frontiers in Artificial Intelligence and Applications*, 215(ECAI 2010):971–972, 2010.

[Anderson *et al.*, 2009] Sterling J Anderson, Steven C. Peters, Karl D. Iagnemma, and Tom E. Pilutti. A unified approach to semi-autonomous control of passenger vehicles in hazard avoidance scenarios. In *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 2032–2037, October 2009. ISSN: 1062-922X.

[Basich *et al.*, 2020] Connor Basich, Justin Svegliato, Kyle Hollins Wray, Stefan Witwicki, Joydeep Biswas, and Shlomo Zilberstein. Learning to Optimize Autonomy in Competence-Aware Systems. *arXiv:2003.07745 [cs]*, March 2020. arXiv: 2003.07745.

[Borovička *et al.*, 2014] Tomáš Borovička, Radim Špetlík, and Karel Rymeš. http://aibirds.org/2014-papers/datalab-birds.pdf, 2014.

[Cubuktepe *et al.*, 2019] Murat Cubuktepe, Nils Jansen, Mohammed Alsiekh, and Ufuk Topcu. Synthesis of Provably Correct Autonomy Protocols for Shared Control. *arXiv:1905.06471 [cs, math]*, May 2019. arXiv: 1905.06471.

[Duchetto *et al.*, 2018] Francesco Del Duchetto, Ayse Kucukyilmaz, Luca Iocchi, and Marc Hanheide. Do not make the same mistakes again and again: Learning local recovery policies for navigation from human demonstrations. *IEEE Robotics and Automation Letters*, 3(4):4084–4091, 2018.

[Feng *et al.*, 2016] L. Feng, C. Wiltsche, L. Humphrey, and U. Topcu. Synthesis of Human-in-the-Loop Control Protocols for Autonomous Systems. *IEEE Transactions on Automation Science and Engineering*, 13(2):450–462, April 2016. Conference Name: IEEE Transactions on Automation Science and Engineering.

[Ferrari *et al.*, 2017] Fabio-Valerio Ferrari, Laurent Jeanpierre, and Abdel-Illah Mouaddib. Flexible pomdp framework for human-robot cooperation in escort tasks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, page 1538–1540, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

[Jansen *et al.*, 2016] Nils Jansen, Murat Cubuktepe, and Ufuk Topcu. Synthesis of Shared Control Protocols with Provable Safety and Performance Guarantees. *arXiv:1610.08500 [cs]*, October 2016. arXiv: 1610.08500.

[Kaelbling *et al.*, 1998] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, May 1998.

[Ong *et al.*, 2009] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Pomdps for robotic tasks with mixed observability. In Jeff Trinkle, Yoky Matsuoka, and José A. Castellanos, editors, *Robotics: Science and Systems V, University of Washington, Seattle, USA, June 28 - July 1, 2009*. The MIT Press, 2009.

[Ong *et al.*, 2010] Sylvie C. W. Ong, Shao Wei Png, David Hsu, and Wee Sun Lee. Planning under Uncertainty for Robotic Tasks with Mixed Observability. *The International Journal of Robotics Research*, 29(8):1053–1068, July 2010. Publisher: SAGE Publications Ltd STM.

[Renz *et al.*,] Jochen Renz, XiaoYu Ge, Peng Zhang, Ekaterina Nikonova, and Vimukthini Pinto. http://aibirds.org/angry-birds-ai-competition.html. Accessed: 2021-06-29.

[Rigter *et al.*, 2020] Marc Rigter, Bruno Lacerda, and Nick Hawes. A Framework for Learning From Demonstration With Minimal Human Effort. *IEEE Robotics and Automation Letters*, 5(2):2023–2030, April 2020.

[Silver and Veness, 2010] David Silver and Joel Veness. Monte-carlo planning in large pomdps. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, NIPS'10, page 2164–2172, Red Hook, NY, USA, 2010. Curran Associates Inc.