Hierarchical Approach to Transfer of Control in Semi-Autonomous Systems

Kyle Hollins Wray and **Luis Pineda** and **Shlomo Zilberstein**

College of Information and Computer Sciences
University of Massachusetts, Amherst, MA 01003
{wray,lpineda,shlomo}@cs.umass.edu

Abstract

Semi-Autonomous Systems (SAS) encapsulate a stochastic decision process explicitly controlled by both an agent and a human, in order to leverage the distinct capabilities of each actor. Planning in SAS must address the challenge of transferring control quickly, safely, and smoothly back-and-forth between the agent and the human. We formally define SAS and the requirements to guarantee that the controlling entities are always able to act competently. We then consider applying the model to Semi-Autonomous VEhicles (SAVE), using a hierarchical approach in which micro-level transfer-of-control actions are governed by a high-fidelity POMDP model. Macro-level path planning in our hierarchical approach is performed by solving a Stochastic Shortest Path (SSP) problem. We analyze the integrated model and show that it provides the required guarantees. Finally, we test the SAVE model using real-world road data from Open Street Map (OSM) within 10 cities, showing the benefits of the collaboration between the agent and human.

1 Introduction

Autonomous systems have been deployed in a wide variety of applications such as space exploration [Zilberstein *et al.*, 2002], reservoir control [Castelletti *et al.*, 2008], energy conservation [Kwak *et al.*, 2012], and autonomous driving [Wray and Zilberstein, 2015; Wray *et al.*, 2015]. These systems, however, almost universally require human intervention or interaction at some point in order to achieve their objectives (e.g., the Mars rovers), or recover from failure (e.g., the Roomba vacuum cleaner). Within the proposed automated planning solutions to these problems, few if any approaches take full advantage of this collaboration. Instead, they commonly resort to default hard-coded behaviors instead of integrating human capabilities into the planning process [Biswas and Veloso, 2013]. Semi-Autonomous Systems (SAS) capture explicitly this collaborative process in which a human and an agent—or any number of actors—work together to achieve a goal, smoothly transferring control over the system back and forth, while proactively considering each actor's capabilities and the human's preferences [Zilberstein, 2015].

New challenges arise in semi-autonomous systems because the overall plan must factor the inherent uncertainty and unpredictability associated with human behavior. We consider a semi-autonomous driving domain where the vehicle can operate autonomously only on well-mapped roads under ideal conditions. To reach a distant destination, the vehicle may require the human to occasionally take control. This transfer of control process requires second-to-second monitoring as various messages are conveyed to the driver. It is also not always successful given an allotted time window and the driver's state (e.g., distracted). These factors must be taken into consideration as the system is planning its long-term route. Additionally, this process of transfer of control must incorporate these factors to provide a measure of safety for the system. Car companies are already developing nascent semi-autonomous capabilities and user interfaces to support transfer of control [Nissan Motor Company Ltd, 2016], but research has been sparse on generalized planning models.

Previous work on semi-autonomous systems has focused on preventing or reacting to human error [Anderson *et al.*, 2009], for example, automatically correcting an undesired lane change [Jung and Kelber, 2004] or human-reactive implementations of adaptive cruise control [Rajamani and Zhu, 2002]. While a long line of research exists on collaboratively controlling a system [Connell and Viola, 1990], planning with the explicit consideration of the human in the plan execution cycle has been lacking [Fong *et al.*, 2003]. No existing algorithm explicitly tackles the transfer of control problem.

Our proposed collaborative multiagent framework is quite distinct from existing approaches for collaboration such as Shared Plans [Grosz and Kraus, 1996], Teamwork [Tambe, 1997], and Dec-POMDP [Bernstein *et al.*, 2002]. First, a SAS requires exactly one actor to be in control of plan execution at any given time. Second, this fact requires explicit mechanisms for transferring control among the actors. Finally, a SAS must proactively plan to leverage each actor's capabilities (or lack thereof) as it efficiently moves in the state space.

Our primary contributions are: (1) a formal definition of a SAS and its key properties, (2) a general transfer of control model, (3) a hierarchical approach for integrating domain action planning with transfer of control, and (4) an analysis showing the hierarchical model is a *strong SAS*. Finally, we provide semi-autonomous driving experiments for 10 cities using real road data that show the benefits of the method.

2 Semi-Autonomous Systems

Semi-Autonomous Systems (SAS) rely on collaboration between a human and an agent in order to achieve some goals while maintaining a measure of safety [Zilberstein, 2015]. We consider semi-autonomy within the context of automated planning, extending a Markov Decision Process (MDP) to support semi-autonomy, as formally defined below.

Definition 1. *A **semi-autonomous system** is represented by a tuple $\langle \mathcal{A}, S_+, A_+, T_+, C_+, G, L \rangle$.*

- *$\mathcal{A}$ is a set of actors (controlling entities).*
- *$S_+ = S \times \mathcal{A}$ is a set of factored states: a standard state set S and the current controlling actor $\mathcal{A}$.*
- *$A_+ = A \times \mathcal{A}$ is a set of factored actions: a standard action set A and the next desired actor $\mathcal{A}$.*
- *$T_+ : S_+ \times A_+ \to \triangle^{|S_+|}$ is a transition function, comprised of a state transition $T_{\mathbf{a}} : S \times A \to \triangle^{|S|}$ for each actor $\mathbf{a} \in \mathcal{A}$, and control transfer function $\rho : S_+ \times \mathcal{A} \to \triangle^{|\mathcal{A}|}$.*
- *$C_+ : S_+ \times A_+ \to \mathbb{R}^+$ is a cost function.*
- *$G \subseteq S_+$ is a set of goal states.*
- *$L \subseteq S_+$ is a set of live states, such that for actor capability function $\psi : S \to 2^{\mathcal{A}}$, $L = \{\langle s, \mathbf{a} \rangle \in S_+ | \mathbf{a} \in \psi(s)\}$.*

The actors $\mathcal{A}$ of the system describe controlling entities, which include at a minimum a human λ and an autonomous agent ν; we focus in this paper on situations involving these specific two actors. The states must record who is in control at any given time, and the actions must record intentions to switch control to new actors. In SAS, we cannot always assume that transfer of the control has a flawless execution. Hence our T_+ (Definition 4) is factored into two components: $T_{\mathbf{a}}$ and ρ (Definitions 2 and 3).

Definition 2. *An **actor state transition function**, denoted $T_{\mathbf{a}} : S \times A \to \triangle^{|S|}$, describes how an actor $\mathbf{a} \in \mathcal{A}$ can operate in the world when in control (n-simplex $\triangle^n$).*

Definition 3. *A **control transfer function**, denoted $\rho : S_+ \times \mathcal{A} \to \triangle^{|\mathcal{A}|}$, describes the result of attempting to transfer control from the current actor in a given state.*

Definition 4. *The **SAS state transition function** for $s_+ = \langle s, \mathbf{a} \rangle$, $a_+ = \langle a, \hat{\mathbf{a}} \rangle$, and $s'_+ = \langle s', \mathbf{a}' \rangle$ is:*

$$T_+(s_+, a_+, s'_+) = \begin{cases} T_{\mathbf{a}}(s, a, s'), & \text{if } \mathbf{a}{=}\hat{\mathbf{a}}{=}\mathbf{a}' \\ T_{\mathbf{a}}(s, a, s')\rho(s_+, \hat{\mathbf{a}}, \mathbf{a}'), & \text{if } \mathbf{a}{\neq}\hat{\mathbf{a}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In Equation 1, the first component corresponds to keeping the current actor, which simply follows the actor's state transition. The second component describes the actor still in control but seeking to switch to a different actor at the next state. The third component indicates that it is impossible to take control from an actor without the desire to transfer.

We develop a hierarchical approach to transfer control that treats each decision of the high-level planning process as a macro-action or an *option* [Sutton *et al.*, 1999], which involves micro-actions to support the successful transfer of control. This hierarchical design seems particularly suited for our target domain of semi-autonomous driving. Here, we perform path planning on large, world-scale roads in time scales of minutes or hours. Transfer of control, however, requires much more care, and is done in time scales of seconds. If we were to path plan with transfer of control at full detail everywhere along the route, the state spaces would be astronomically large. For example, for the *smallest* problem instance in our experiments (Pittsburgh), this would blow up the state space by a factor of 387, resulting in a POMDP with approximately 7.6×10^4 states. Instead, we take advantage of the fact that transfer of control is a generic process that depends on a handful of context variables such as the time remaining to complete the transfer and some general driving conditions (e.g., transferring control on a straight road, turns, low-speed, and high-speed). Apart from that, the way in which the transfer of control is performed is largely independent of the remaining route and destination. This enables us to generalize the transfer process, and model it as a compact state transition at the higher level following ρ in the form of an *option*.

Following Zilberstein [2015], a **SAS of type I (SAS-I)** does *not* explicitly model the human in the execution loop, whereas a **SAS of type II (SAS-II)** does. Thus, we have presented a SAS-II as we explicitly model the human within our set of actors ($\lambda \in \mathcal{A}$). We proceed with additional properties.

Within a SAS, we define two types of histories. Definition 5 formalizes the meaning of any trajectory over states and actions, given the limits of the stochastic state transition.

Definition 5. *A **realizable history** is a sequence of the form $\hbar = \langle s^0_+, a^0_+, \ldots, s^\ell_+ \rangle$ such that for all s^i_+, a^i_+, and s^{i+1}_+, $T_+(s^i_+, a^i_+, s^{i+1}_+) > 0$. The set of all realizable histories starting at $s^0_+ \in S_+$ with horizon $\ell \in \mathbb{N}$ is denoted $\bar{H}(s^0_+, \ell)$.*

Next, we define a more constrained history with respect to a specific *policy* in Definition 6. A policy $\pi : S_+ \to A_+$ is a mapping from factored states to factored actions.

Definition 6. *Given policy π, a **policy realizable history**, is a realizable history $\hbar$ such that $\forall i, a^i_+ = \pi(s^i_+)$. We denote the set of all policy realizable histories starting at state $s^0_+ \in S_+$ with horizon $\ell \in \mathbb{N}$ as $\bar{H}_\pi(s^0_+, \ell)$.*

Given a policy π, the agent incurs a cost per time step given by $C_+ : S_+ \times A_+ \to \mathbb{R}^+$ as it tries to reach a *goal state* from $G \subseteq S_+$. A policy is *optimal* if it minimizes the expected cost over time, also called the *value* of a state $V_+ : S_+ \to \mathbb{R}$. For initial state $s^0 \in S_+$, the optimal policy π^* minimizes:

$$V_+(s^0) = \mathbb{E}\Big[\sum_{t=0}^{\infty} C^t_+(s^t, \pi^*(s^t)) | s^0 \Big] \quad (2)$$

Given an initial state, this defines a Stochastic Shortest Path (SSP) MDP. Bellman's optimality equation for state s is:

$$V_+(s) = \min_{a \in A_+} \{ C_+(s, a) + \sum_{s' \in S_+} T_+(s, a, s') V_+(s') \} \quad (3)$$

Following Bertsekas and Tsitsiklis [1991], this equation produces an optimal policy π^* under two assumptions. First, a *proper policy* must exist that can reach a goal with probability 1 from s. Second, all *improper policies* must incur infinite cost at states that cannot reach a goal with probability 1. Such SSPs can be solved using search methods such as

LAO* [Hansen and Zilberstein, 2001] as well as conventional value iteration.

So far we have described a concrete model that explicitly represents the current actor (controlling entity) and the dynamics for attempting to transfer control among actors. We now introduce **actor capability constraints** for SAS, which specify limits on the abilities of actors to control the system under certain conditions, through the function in Definition 7.

Definition 7. *An **actor capability function (ACF)** $\psi : S \rightarrow 2^{\mathcal{A}}$ maps states to the actors capable of acting in that state.*

For example, in the semi-autonomous driving domain, the autonomous agent may not be able to drive on every road, but only on well-mapped roads or under certain weather conditions. Thus, the planner must incorporate the limited capabilities of the autonomous agent, as well as the uncertainty regarding transfer of control between the human and agent, in order to construct a route from a starting location to a destination. Definition 8 formalizes the notion of *live state* in which an actor can control the system.

Definition 8. ***Live states*** $L = \{\langle s, \mathbf{a} \rangle \in S_+ | \mathbf{a} \in \psi(s)\}$ *are states which satisfy the ACF ψ.*

Live states are states in which the system is considered *active* or *safe*, because the controlling entity can act there. Definition 9 states constraints that must be satisfied for all SAS.

Definition 9. *The **live state constraints** for SAS are:*

1. $G \subseteq L$
2. $\forall s_+ \notin L, \forall a_+ \in A_+, \forall s'_+ \in L, T_+(s_+, a_+, s'_+) = 0$

Unlike general *dead ends* [Kolobov *et al.*, 2012], which are states from which the goal becomes unreachable, our live state constraints form a particular *structured* type of dead end that is easier to analyze (largely because these conditions are explicitly captured by L). In fact, we will show that our semi-autonomous vehicle formulation produces policies that guarantee avoidance of non-live states. In general, this requires us to prove that the given transfer of control model produces a ρ that never enters a non-live state. This key mechanism is described in the following section. We now formalize this.

Our objective is to characterize *policies* and *systems* in terms of their ability to *maintain live state*. Definitions 10, 11, and 12 present three key properties of policies.

Definition 10. *A policy π is **strong** if for all $s^0_+ \in L$ and $\ell \in \mathbb{N}$, and for all $\hbar \in \bar{H}_\pi(s^0_+, \ell)$ and $i \in \{0, \ldots, \ell\}$, $s^i_+ \in L$.*

Definition 11. *A policy π is **conditionally strong** if there exists an $s^0_+ \in L$ and $\ell \in \mathbb{N}$, such that for all $\hbar \in \bar{H}_\pi(s^0_+, \ell)$ and $i \in \{0, \ldots, \ell\}$, $s^i_+ \in L$.*

Definition 12. *A policy π is **weak** if it is not strong or conditionally strong.*

Next, we extend these terms from policies to an entire SAS. A SAS is said to be **strong** (**conditionally strong**) if there exists a strong (conditionally strong) policy π^* that is optimal. Otherwise, the SAS is said to be **weak**.

3 Transfer of Control

Transfer of Control (TOC) is the critical method that enables effective and safe transference of the controlling entity within the stochastic decision-making process. TOC both to and from a human requires the optimal selection of various messages (e.g., visual or auditory) in order to prompt the human to reengage and ensure smooth transference. Each message type presents a trade-off between the efficacy of alerting the human to the agent's intention and the human's amiable perception of the agent (e.g., aggregated annoyance). For example, a continuous alarm is effective but undesirable, and a blinking light is not as effective but more favorable. Additionally, the system receives noisy observations of the human's state of engagement due to the limited sensing capabilities available. Thus, it instead must make decisions based on a belief regarding the engagement level. Finally, this is a time-sensitive sequential optimization problem due to the limited time window in which control may be transferred. For example, in semi-autonomous driving, the vehicle may not be able to operate on insufficiently mapped roads and must seamlessly relinquish control before reaching these roads. We model this process using a POMDP. First, we formally define the TOC problem, then POMDPs, and finally construct the POMDP model of TOC.

3.1 The Transfer of Control Problem

The **Transfer of Control (TOC) Problem** is a tuple $\langle \mathcal{H}, \mathcal{M}, \mathcal{O}, \mathcal{T}, \mathcal{P}_h, \mathcal{P}_c, \mathcal{P}_o, \mathcal{C} \rangle$. $\mathcal{H}$ is a set of human states. $\mathcal{M}$ is a set of available messages to inform the user of the desire to transfer control. The absence of a message is indicated by $\varnothing \in \mathcal{M}$ (i.e., no operation or 'NOP') and is always available. $\mathcal{O}$ is the set of observations made by sensors, which provide partial information about the human's state. $\mathcal{T} = \{1, \ldots, \tau\}$ is a set of limited time steps for the transfer of control to complete (e.g., τ seconds). $\mathcal{P}_h : \mathcal{H} \times \mathcal{M} \times \mathcal{T} \rightarrow \triangle^{|\mathcal{H}|}$ is the probability of the human state transitioning from h to h' given message m was sent t time steps ago, such that we have $\mathcal{P}_h(h, m, t, h') \equiv Pr(h' \mid h, m, t)$. $\mathcal{P}_c : \mathcal{H} \times \mathcal{M} \rightarrow \triangle^{|\mathcal{T}|}$ is the probability that control will be transferred given the human state h and that message m was sent t time steps ago. If control is transferred, then the process terminates; the agent knows when this occurs. $\mathcal{P}_o : \mathcal{H} \rightarrow \triangle^{|\mathcal{O}|}$ is the probability of making a sensor observation o given the human state is h, such that $\mathcal{P}_o(h, o) \equiv Pr(o|h)$. $\mathcal{C} : \mathcal{H} \times \mathcal{M} \times \mathcal{T} \rightarrow \mathbb{R}^+$ is the cost of sending message m given human state h and t time steps since sending the last message (i.e., $\mathcal{C}(h, m, t)$). The agent can always **abort**, ending the transfer attempt.

The human has a true hidden state $h \in \mathcal{H}$. This changes over time as the agent selects a message $m_t \in \mathcal{M}$ for each $t \in \mathcal{T}$, forming sequence $m = \langle m_1, \ldots, m_\tau \rangle$. The objective is to *minimize the total sum of message costs*; however, failing to transfer control without safely aborting should strictly be avoided. Thus, the agent must also decide when to abort.

Importantly, control can be transferred either way in this model. That is, it captures requesting control to be *both* taken from and given to the agent. Furthermore, different TOC problems may be defined, each encoding a *different* environment or scenario in which control must be transferred. For example, a vehicle taking control on a highway turn, or a human taking control on a quiet suburban road, are both different transfer of control instances.

3.2 Background on POMDPs

A **Partially Observable Markov Decision Process (POMDP)** is defined by a tuple $\langle \bar{S}, \bar{A}, \bar{\Omega}, \bar{T}, \bar{O}, \bar{R} \rangle$ [Kaelbling *et al.*, 1998]. $\bar{S}$ is a set of n states. $\bar{A}$ is a set of m actions. $\bar{\Omega}$ is a set of z observations. $\bar{T} : \bar{S} \times \bar{A} \to \triangle^{|\bar{S}|}$ is a state transition function such that $\bar{T}(s, a, s') \equiv Pr(s'|s, a)$. $\bar{O} : \bar{A} \times \bar{S} \to \triangle^{|\bar{\Omega}|}$ is an observation function such that $\bar{O}(a, s', \omega) \equiv Pr(\omega|a, s')$. $\bar{R} : \bar{S} \times \bar{A} \to \mathbb{R}$ is a reward function denoted $\bar{R}(s, a)$. Actions are performed at each time step, up to the horizon $\ell \in \mathbb{N}$, discounting the reward obtained by $\gamma \in (0, 1)$. The agent, however, must select actions without knowing the true state of the system. Instead, it maintains a *belief* over the true state $b \in \triangle^n$, or a collection of r beliefs $\bar{B} \subseteq \triangle^n$ (standard n-simplex). Given belief b, action a, and subsequent observation ω, for each state s', the belief updates (normalizing constant $\eta = Pr(\omega|b, a)^{-1}$) via: $b'(s') = \eta \bar{O}(a, s', \omega) \sum_{s \in \bar{S}} \bar{T}(s, a, s') b(s)$.

The goal is to find a *policy* $\pi : \bar{B} \to \bar{A}$ that maximizes the expected reward over time, i.e., *value function* $\bar{V} : \bar{B} \to \mathbb{R}$. This function is piecewise linear and convex [Kaelbling *et al.*, 1998], so we use a set of α-*vectors* $\Gamma = \{\alpha_1, \ldots, \alpha_r\}$ with each vector $\alpha_i = [\alpha_i(s_1), \ldots, \alpha_i(s_n)]^T$. Each $\alpha_i(s_j)$ denotes the value of state $s_j \in S$. The optimal value (resulting in an optimal policy $\bar{\pi}^*$) at time t for belief b is:

$$\bar{V}^t(b) = \max_{a \in \bar{A}} \sum_{s \in \bar{S}} b(s) \bar{R}(s, a) + \sum_{\omega \in \bar{\Omega}} \max_{\alpha \in \Gamma^{t-1}} \sum_{s \in \bar{S}} b(s) \bar{V}^t_{sa\omega\alpha} \tag{4}$$

with $\bar{V}^t_{sa\omega\alpha} = \gamma \sum_{s' \in \bar{S}} \bar{O}(a, s', \omega) \bar{T}(s, a, s') \alpha(s')$.

3.3 Transfer of Control POMDP Formulation

We model the control transfer problem as a POMDP called a **TOC POMDP**. The state space is $\bar{S} = \mathcal{T} \times \mathcal{H} \times \mathcal{M} \times \mathcal{T} \cup \mathcal{E}$ with a set of 'end result' states $\mathcal{E} = \{\oplus, \ominus, \oslash\}$ denoting 'success,' 'failure,' and 'aborted,' respectively. Each state captures the time remaining, current human state, the previous message sent, and how long it has been since that message was sent, as well as the outcome of the transfer of control. The action space $\bar{A} = \mathcal{M} \cup \{\oslash\}$ is the messages to send and the 'abort' action (denoted $\oslash$). The observation space $\bar{\Omega} = \mathcal{O} \cup \mathcal{E}$ represents the observations, and lets the model know the end result, as per the problem definition. The state transition function needs to encapsulate the notions of human state, as well as the success or failure of transferring control. We break this into two scenarios.

The first scenario examines transitions only among non-end result states such that $s, s' \notin \mathcal{E}$, with state factors denoted $s = \langle t, h, m, t_m \rangle$ and $s' = \langle t', h', m', t'_m \rangle$. This scenario has two non-zero cases each with the same probability. In both, control has not successfully transferred yet, so we always update the human state and count down the timer (via constraint $t' = t - 1 \geqslant 0$). The first case encodes the effect of sending a message from $\mathcal{M} \backslash \{\varnothing\}$. The second case encodes the effect of a 'NOP' message $\varnothing$. Formally, for any s, a, and s':

$$\bar{T}(s, a, s') = \begin{cases} \bar{\mathcal{P}} & \text{if } a \notin \{\oslash, \varnothing\} \wedge m' = a \wedge t'_m = 0 \\ \bar{\mathcal{P}} & \text{if } a = \varnothing \wedge m' = m \wedge t'_m = \min\{t_m + 1, \tau\} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

with $\bar{\mathcal{P}} = (1 - \mathcal{P}_c(h, m, t_m)) \mathcal{P}_h(h, m, t_m, h')$ above.

The second scenario examines transitions to an end result state: successor $s' \in \mathcal{E}$. This scenario has four non-zero cases. The first case is simply the absorbing states $\mathcal{E}$. The second case is immediate termination via the abort action $\oslash$. The third case captures the ever-possible chance of successful control transfer ($\oplus$). The fourth case handles a failure ($\ominus$) transition by running out of time. Thus, for a state s, action a, and successor s' we have:

$$\bar{T}(s, a, s') = \begin{cases} 1 & \text{if } s = s' \in \mathcal{E} \\ 1 & \text{if } s \notin \mathcal{E} \wedge a = s' = \oslash \\ \mathcal{P}_c(h, m, t_m) & \text{if } s \notin \mathcal{E} \wedge a \neq \oslash \wedge s' = \oplus \\ 1 - \mathcal{P}_c(h, m, t_m) & \text{if } s \notin \mathcal{E} \wedge t = 0 \wedge a \neq \oslash \wedge s' = \ominus \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

with $s = \langle t, h, m, t_m \rangle$ above provided $s \notin \mathcal{E}$.

The observation transition function only needs to model two components. First, the agent always has perfect knowledge of the final outcome state. Second, the agent makes noisy observations from sensors which hint at the true human state (e.g., a face or eye tracker in a vehicle). Formally, for action a, successor state s', and observation ω:

$$\bar{O}(a, s', \omega) = \begin{cases} 1 & \text{if } \omega = s' \in \mathcal{E} \\ \mathcal{P}_o(h', \omega) & \text{if } s' \notin \mathcal{E} \wedge \omega \in \mathcal{O} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

with states $s' = \langle t', h', m', t'_m \rangle$ above provided $s' \notin \mathcal{E}$.

The reward function has four components. First, there are costs associated with all normal messages, as defined by the TOC problem. Second, an arbitrarily small $\epsilon > 0$ cost is given for a NOP $\varnothing$. Third, there is a large penalty for unnecessary aborting. Fourth, failure repeatedly incurs the maximal cost. Thus, for state s and action a:

$$\bar{R}(s, a) = \begin{cases} -\mathcal{C}(h, m, t_m) & \text{if } s \notin \mathcal{E} \wedge a \notin \{\varnothing, \oslash\} \\ -\epsilon & \text{if } s \notin \mathcal{E} \wedge a = \varnothing \\ -\mathcal{C}^* & \text{if } s \notin \mathcal{E} \wedge a = \oslash \wedge t > 0 \\ -\mathcal{C}^* & \text{if } s = \ominus \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

with $s = \langle t, h, m, t_m \rangle$ and a non-success penalty $\mathcal{C}^*$ (e.g., let $\mathcal{C}^* = \ell \mathcal{C}_{max}$ with $\mathcal{C}_{max} = \max_{h, m, t_m} \mathcal{C}(h, m, t_m)$).

Within a TOC POMDP, the end result terminal states are fully observable, as well as each state factor *except* for the true human state $\mathcal{H}$. Thus, all beliefs, including initial belief b^0, only contain uncertainty regarding these human states.

4 Application to Semi-Autonomous Driving

The TOC formulation as a POMDP enables us to incorporate semi-autonomy into stochastic path planning problems. Again, our main motivation is the semi-autonomous driving domain. Route decisions are made at intersections of roads; however, only well-mapped main roads are capable of autonomy. While the driver can drive on any road, the longer, uninteresting, boring highways are assumed to be roads in which the human prefers autonomy, meaning that control should be transferred to the vehicle. All costs are proportional to the time spent on the road. The uncertainty stems from the transfer of control, also decided at road intersections. We first formally define the problem, then describe the full model of a specific SAS called a Semi-Autonomous VEhicle (SAVE).

4.1 Problem Definition

The **Semi-Autonomous VEhicle (SAVE) Problem** begins with a strongly connected weighted directed graph $\langle V, E, w\rangle$. V is a set of vertices forming intersections. $E \subseteq V \times V$ is a set of pairs of vertices (intersections) defining edges which form roads. $w : E \to \mathbb{R}^+$ defines a positive weight for each edge (road) which captures the time spent on the road. There are initial and goal vertices $v^0, v^g \in V$.

Additionally, the system may be driven by the human or the agent (vehicle) itself; however, given the allotted time between each vertex (intersection), there is uncertainty if control transfer will be successful. This micro-level behavior is modeled using TOC POMDPs. (These may be solved offline for different scenarios, as described in the previous section.) We label edges (roads) $E_c \subseteq E$ as **autonomy-capable**, meaning the set of roads in which the vehicle is capable of driving autonomously. Similarly, $E_p \subseteq E_c$ are **autonomy-preferred** roads in which the human prefers autonomous driving. Thus, at each intersection the agent must decide if control should be maintained or transferred as well as which road to take next, in order to *minimize the sum of traversed weights (travel time)*. If control transfer is required, fails to succeed, but the agent aborted, then the system is assumed to safely pause at the vertex, i.e., the vehicle safely pulls over to the side of the road. Otherwise, the SAS will enter an unsafe state in which the vehicle is in control but cannot drive on the road; this should be avoided at all costs.

4.2 Semi-Autonomous Vehicle Formulation

A **Semi-Autonomous VEhicle (SAVE)** is a SAS with $\langle \mathcal{A}, S_+, A_+, T_+, C_+, G, L\rangle$. Actors $\mathcal{A} = \{\lambda, \nu, \sigma\}$ encode the current controlling agent of the vehicle: either the *human* λ, the *vehicle* itself ν, or no active actor as it *safely waits* on the side of the road σ. $S_+ = V \times \mathcal{A}$ have standard states V corresponding to the vertices (intersections) of the map. $A_+ = D \times \mathcal{A}$ is the action set with D denoting the possible directions (roads) to take at a vertex (intersection) (e.g., $D = \{\leftarrow, \uparrow, \rightarrow\}$). This notation is commonly overloaded such that $A_+(s)$ returns the set of actions available at state s. Let $\theta : V \times D \to V$ map a vertex (intersection) and an action (direction) to the subsequent vertex following E. Additionally, we assume: (1) the map is expanded to include a 'failure' absorbing vertex $v^f \in V$, with $\theta(v^f, d) = v^f$ for all $d \in D$, and (2) the goal is also absorbing with $\theta(v^g, d) = v^g$ for all $d \in D$.

The state transition function T_+ follows Equation 1, and introduces the uncertainty from our TOC POMDP's transfer of control process given by ρ. First, for the human actor λ, the actor transition function T_λ simply follows the map. Next, for vehicle actor ν, the actor transition function T_ν ensures that: (1) autonomy-capable states follow the path, and (2) non-autonomy-capable states enter the absorbing vertex v^f. Finally, for the safely parked vehicle σ, the actor transition function T_σ always self-loop since it is on the side of the road.

We now define the control transfer function ρ. Formally, for $s = \langle v, \mathbf{a}\rangle \in S_+$ with action $\langle d, \hat{\mathbf{a}}\rangle \in A_+(s)$, the allotted travel time is $\tau = \lfloor w(\langle v, \theta(v, d)\rangle)\rfloor$. We have a particular TOC POMDP given the intersection v, current controlling actor $\mathbf{a}$, and desired successor actor $\hat{\mathbf{a}}$ from optimal SAVE policy $\pi^*(s) = \langle d, \hat{\mathbf{a}}\rangle$. We assume the TOC POMDP's $\mathcal{T} = \{1, \ldots, \tau\}$ has units in seconds without loss of generality. Given the solved TOC POMDP, we would like to compute the expected result from transfer of control. Formally, we sample trajectories over the unobserved true state, observations, and resultant action following the TOC POMDP's optimal policy. Due to the structure of the TOC POMDP, this always results in a collapsed belief with a known end result from $\mathcal{E}$. Formally, let $J = \{s_1, \ldots, s_k\}$ be a set of k final 'end result' states from the TOC POMDP's $\mathcal{E}$ which are determined by random state-action-observation trajectories following the TOC POMDP.

In the case with $\mathbf{a} = \sigma$, initially the car is safely on the side of the road. Either (1) the car remains on the side of the road, (2) the human TOC succeeds, or (3) the human TOC fails:

$$\rho(s, \hat{\mathbf{a}}, \mathbf{a}') = \begin{cases} 1, & \text{if } \hat{\mathbf{a}}{\neq}\lambda \wedge \mathbf{a}'{=}\sigma \\ \frac{1}{k}\sum_{i=1}^{k}[s_i{=}\oplus], & \text{if } \hat{\mathbf{a}}{=}\lambda \wedge \mathbf{a}'{=}\lambda \\ \frac{1}{k}\sum_{i=1}^{k}[s_i{\neq}\oplus], & \text{if } \hat{\mathbf{a}}{=}\lambda \wedge \mathbf{a}'{=}\sigma \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

with $[\cdot]$ denoting Iverson brackets.

In the case with $\mathbf{a} \neq \sigma$, either the human λ or vehicle ν is the controlling entity. Either (1) no transfer is requested and the actor remains the same, (2) the TOC succeeds to switch actors, (3) the TOC fails to switch actors, or (4) the TOC is aborted and safely pulls over to the side of the road:

$$\rho(s, \hat{\mathbf{a}}, \mathbf{a}') = \begin{cases} 1, & \text{if } \hat{\mathbf{a}}{=}\mathbf{a} \wedge \mathbf{a}'{=}\hat{\mathbf{a}} \\ \frac{1}{k}\sum_{i=1}^{k}[s_i{=}\oplus], & \text{if } \hat{\mathbf{a}}{\neq}\mathbf{a} \wedge \mathbf{a}'{=}\hat{\mathbf{a}} \\ \frac{1}{k}\sum_{i=1}^{k}[s_i{=}\ominus], & \text{if } \hat{\mathbf{a}}{\neq}\mathbf{a} \wedge \mathbf{a}'{=}\mathbf{a} \\ \frac{1}{k}\sum_{i=1}^{k}[s_i{=}\oslash], & \text{if } \hat{\mathbf{a}}{\neq}\mathbf{a} \wedge \mathbf{a}'{=}\sigma \\ 0, & \text{otherwise} \end{cases} \tag{10}$$

Trivially, in the limit as the number of sampled trajectories grows $k \to \infty$, this converges to the exact probabilities of obtaining each resulting actor, our desired result. This formulation of the expected value with number of samples k enables us to sample a finite number of times and obtain an approximation of ρ in practice.

The cost function C_+ simply measures the time traveling on a road, given its length and speed limit. We assume ties between actions are broken following the autonomy-preferred roads in E_p. In other words, if autonomy is preferred and the current actor is the human λ, then the next action will attempt to transfer to the vehicle ν. For $s = \langle v, \mathbf{a}\rangle$ and $a = \langle d, \hat{\mathbf{a}}\rangle$:

$$C_+(s, a) = \begin{cases} w(\langle v, \theta(v, d)\rangle), & \text{if } v \neq v^g \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

The goal is $G = \{\langle v^g, \lambda\rangle\}$ and the initial state is $s^0 = \langle v^0, \lambda\rangle$. Following the problem definition, the human is capable of acting in all states, and the vehicle can be safely on the side of the road in any state. Thus, for a vertex v, $\psi(v)$ is $\{\lambda, \nu, \sigma\}$ if the road v is autonomy-capable, and $\{\lambda, \sigma\}$ otherwise. Trivially, the failure state has no actors ($\psi(v^f) = \varnothing$). This defines L following Definition 8.

4.3 Theoretical Analysis

We now integrate all three concepts we have introduced thus far: SAS, TOC POMDP, and SAVE, in order to show that the

transfer of control within our stochastic path planning model provably maintains live state guarantees. Due to space constraints, we only provide proof sketches.

Proposition 1. *SAVE satisfies the live state constraints.*

Proof (Sketch). Both cases in Definition 9 are satisfied because (1) the goal is a live state, and (2) T_λ forces all non-live states to transition to the absorbing states at v^f. □

Next, we establish Lemma 1 which states that the only uncertainty regarding the true state is over the human factor. This fact allows the agent to know and thus act appropriately at the final time step.

Lemma 1. *Belief uncertainty within the TOC POMDP is only over the human state factor $\mathcal{H}$; all other factors are known.*

Proof (Sketch). Proof by induction following the belief update equation from b^0. Base case is trivial. Induction step considers the update to b' for any a or ω. Equation 5's $\bar{\mathcal{P}}$ is non-zero following $\mathcal{P}_c$ and $\mathcal{P}_h$; all other state factors deterministic (both cases). Equation 7 follows $\mathcal{P}_o$, only stochastic over human states. Thus, b' has uncertainty only over $\mathcal{H}$. □

Establishing this property allows us to prove that the TOC POMDP never enters the failure state $\ominus$ in Proposition 2. This is a critical requirement in order to prove that SAVE is a strong SAS in Proposition 3.

Proposition 2. *Following a TOC POMDP's optimal policy π^*, for any horizon ℓ, the underlying true state $s^\ell \neq \ominus$.*

Proof (Sketch). By Lemma 1, we examine actions taken at reachable beliefs with $t = 0$, since this factor is known to the agent. By Equation 8, the abort action $\oslash$ is always optimal. By Equation 6, the failure state is unreachable following π^*. Thus, $s^\ell \neq \ominus$. □

Proposition 3. *SAVE is a strong SAS.*

Proof (Sketch). SAVE enters a failure v^f from some v only for T_ν with $v \notin E_c$ and zero time remaining ($t = 0$). By Equation 10 and ψ's definition, the only way to enter a non-live state is for an optimal TOC POMDP policy to enter $\ominus$ with ν in control when $v' \notin E_c$. This is impossible by Proposition 2, implying a SAVE optimal policy is strong (Definition 10). Thus, SAVE is a strong SAS. □

5 Experiments

We present a series of trials with subsets of 10 cities' road data from Open Street Map (OSM). Distant start and goal addresses are selected as one would do using a Global Positioning System (GPS) device. All main roads with a speed limit of 30 or greater are marked as autonomy-preferred. We compare our approach with a human driver following the GPS (λ), only the autonomous car (ν), and the collaboration between human and vehicle (λ & ν). We use three metrics for comparison. First, we check if the goal was reachable from the initial road (G). Second, we determine the percentage of time the vehicle drives autonomously, provided it could do so, for roads along its route (%). Third, we record the average travel time to compare efficiency along each of the routes (T).

			λ			ν			λ & ν		
City	$\lvert S\rvert$	$\lvert A\rvert$	G	%	T	G	%	T	G	%	T
Austin	303	12	Y	0	128	N	100	—	Y	13	128
Balt.	315	12	Y	0	146	Y	100	232	Y	46	154
Boston	912	18	Y	0	136	N	100	—	Y	95	140
Chic.	258	12	Y	0	99	N	100	—	Y	85	142
Denver	348	15	Y	0	128	N	100	—	Y	81	132
L.A.	291	12	Y	0	120	N	100	—	Y	42	120
N.Y.C.	960	15	Y	0	294	N	100	—	Y	54	313
Pitts.	198	12	Y	0	81	N	100	—	Y	8	89
San Fr.	504	18	Y	0	151	Y	100	183	Y	80	174
Seattle	366	12	Y	0	111	Y	100	138	Y	0	111

Table 1: Results for human λ, vehicle ν, and λ & ν drivers.

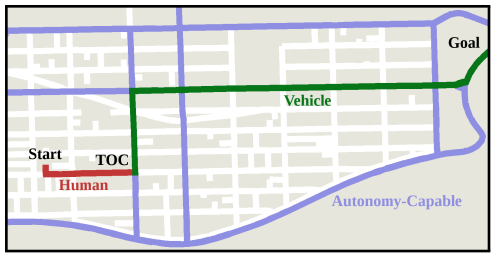

Figure 1: SAVE policy with TOC in Boston.

Our experiments solve the TOC POMDP with Point-Based Value Iteration (PBVI) [Pineau *et al.*, 2003] and the SAVE SAS using LAO* [Hansen and Zilberstein, 2001]. Table 1 shows our results for 100 trials for each city. Figure 1 depicts a sample collaborative policy in which the human and vehicle gracefully transfer control along the route. The driver is always able to reach the goal, but never drives autonomously, even when the car is capable of doing so. The autonomous vehicle always succeeds in autonomously driving, but is only able to reach the goal in 3 of the 10 scenarios. When it does drive autonomously, it has to take long main roads, causing travel time to be greatly increased. Interestingly, the human and autonomous vehicle collaboration always reaches the goal, and drives autonomously for large portions of the route. Also, the average travel times are relatively similar between the human and collaborative scenarios. This collaborative approach selects routes that properly balance main road autonomous driving and back road human driving.

6 Conclusion

We present a hierarchical approach to the transfer of control in semi-autonomous systems, which facilitates efficient planning for a human-agent collaboration. The hierarchical model captures explicitly and optimizes the critical transfer of control process using a POMDP. We show how to apply the general framework to SAS for semi-autonomous vehicles and demonstrate its benefits. Furthermore, we analyze the SAS with TOC model, showing that it maintains live state and thus is a strong SAS. The experiments show that the hierarchical approach is able to leverage the capabilities of the human and agent as it optimizes the desired objective.

Future work will include experiments with humans in a full-scale driving simulator. We will also explore other SAS domains such as assistive technologies (e.g., physical therapy) and disaster response (e.g., search-and-rescue). Finally, we will provide our source code to facilitate the creation of a wide variety of strong semi-autonomous systems.

Acknowledgments

We thank Claudia Goldman for feedback on early versions of this work. This research was supported by the National Science Foundation grant number IIS-1405550.

References

[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 *IEEE International Conference on Systems, Man and Cybernetics*, pages 2032–2037, 2009.

[Bernstein *et al.*, 2002] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

[Bertsekas and Tsitsiklis, 1991] Dimitri P. Bertsekas and John N. Tsitsiklis. An analysis of stochastic shortest path problems. *Mathematics of Operations Research*, 16(3):580–595, 1991.

[Biswas and Veloso, 2013] Joydeep Biswas and Manuela M. Veloso. Localization and navigation of the CoBots over long-term deployments. *International Journal of Robotics Research (IJRR)*, 32(14):1679–1694, 2013.

[Castelletti *et al.*, 2008] Andrea Castelletti, Francesca Pianosi, and Rodolfo Soncini-Sessa. Water reservoir control under economic, social and environmental constraints. *Automatica*, 44(6):1595–1607, 2008.

[Connell and Viola, 1990] Jonathan Connell and Paul Viola. Cooperative control of a semi-autonomous mobile robot. In *Proceedings of the 5th IEEE International Conference on Robotics and Automation (ICRA)*, pages 1118–1121, 1990.

[Fong *et al.*, 2003] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A survey of socially interactive robots. *Robotics and Autonomous Systems*, 42(3-4):143–166, 2003.

[Grosz and Kraus, 1996] Barbara J. Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[Hansen and Zilberstein, 2001] Eric A. Hansen and Shlomo Zilberstein. LAO*: A heuristic search algorithm that finds solutions with loops. *Artificial Intelligence*, 129(1-2):35–62, 2001.

[Jung and Kelber, 2004] Cláudio R. Jung and Christian R. Kelber. A lane departure warning system based on a linear-parabolic lane model. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 891–895, 2004.

[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, 1998.

[Kolobov *et al.*, 2012] Andrey Kolobov, Mausam, and Daniel Weld. A theory of goal-oriented MDPs with dead ends. In *Proceedings of the 28th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 438–447, Corvallis, Oregon, August 2012.

[Kwak *et al.*, 2012] Jun-Young Kwak, Pradeep Varakantham, Rajiv Maheswaran, Milind Tambe, Farrokh Jazizadeh, Geoffrey Kavulya, Laura Klein, Burcin Becerik-Gerber, Timothy Hayes, and Wendy Wood. SAVES: A sustainable multiagent application to conserve building energy considering occupants. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 21–28, 2012.

[Nissan Motor Company Ltd, 2016] Nissan Motor Company Ltd. Together we ride. http://youtu.be/9zZ2h2MRCe0, April 2016.

[Pineau *et al.*, 2003] Joelle Pineau, Geoff Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1025–1032, 2003.

[Rajamani and Zhu, 2002] Rajesh Rajamani and Chunlin Zhu. Semi-autonomous adaptive cruise control systems. *IEEE Transactions on Vehicular Technology*, 51(5):1186–1192, September 2002.

[Sutton *et al.*, 1999] Richard S. Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112:181–211, 1999.

[Tambe, 1997] Milind Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 2:83–124, 1997.

[Wray and Zilberstein, 2015] Kyle H. Wray and Shlomo Zilberstein. Multi-objective POMDPs with lexicographic reward preferences. In *Proceedings of the 24th International Joint Conference of Artificial Intelligence (IJCAI)*, pages 1719–1725, 2015.

[Wray *et al.*, 2015] Kyle Hollins Wray, Shlomo Zilberstein, and Abdel-Illah Mouaddib. Multi-objective MDPs with conditional lexicographic reward preferences. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 3418–3424, 2015.

[Zilberstein *et al.*, 2002] Shlomo Zilberstein, Richard Washington, Daniel S. Bernstein, and Abdel-Illah Mouaddib. Decision-theoretic control of planetary rovers. In *International Seminar on Advances in Plan-Based Control of Robotic Agents*, pages 270–289, 2002.

[Zilberstein, 2015] Shlomo Zilberstein. Building strong semi-autonomous systems. In *Proceedings of the 29th Conference on Artificial Intelligence (AAAI)*, pages 4088–4092, 2015.