Probabilistic System Modeling for Complex Systems Operating in Uncertain Environments

Parisa Pouya and Azad M. Madni

Abstract Complex systems that continuously interact with dynamic uncertain environments need the ability to adapt their decision-making based on observed outcomes of their decisions and actions. Traditional deterministic modeling approaches are invariably inadequate for modeling systems whose models are not fully known initially. For such systems, we need the ability to start with an incomplete model and then progressively complete the model with observations made along the way. To address this problem type, we propose an extendable-partially observable Markov decision process (extendable-POMDP) to model the system's state space and decision-making in the presence of uncertainties. The extendable-POMDP model is able to account for unknown-unknowns by incorporating "new hidden states" that result in expanding the model state space which in turn extends the associated probability distributions. This paper provides an online algorithm for solving a POMDP model by employing a heuristic search algorithm that estimates long-term rewards in a finite-horizon look-ahead in a sense-plan-act cycle. Heuristics are employed in model definition, expansion, and online look-ahead search to contain the otherwise inevitable computational complexity arising from state-space explosion.

Keywords Decision-making and planning · Probabilistic modeling · Partially observable Markov decision processes · Complex systems · Model-based systems engineering (MBSE) · Heuristic search

P. Pouya (✉) · A. M. Madni
University of Southern California, Los Angeles, USA
e-mail: pouya@usc.edu

A. M. Madni et al. (eds.), *Recent Trends and Advances in Model Based Systems Engineering*, https://doi.org/10.1007/978-3-030-82083-1_11

1 Introduction

Understanding the dynamic behavior of complex systems that undergo state changes as a result of ongoing interaction with the environment is a challenging system modeling problem. Existing techniques that are employed for these purposes are either deterministic or stochastic. Deterministic models fully determine a model based on setting up initial assumptions and conditions, while stochastic models exhibit required randomness needed to characterize stochastic properties. Recent studies on modeling a system's dynamic behavior can be classified as the following: (1) inferring the underlying model and parameters for classifying and making predictions about the future and (2) identifying the conditional dependencies, relationships between variables, correlations, and changes in variables over time (Robinson and Hartemink 2009). The main objective of the former class of problems is to find patterns and parameters from the behavioral data and make predictions about the future. Examples of this class of problems are speech recognition, activity or behavior detection, and anomaly detection. Markovian models, such as Markov chains and hidden Markov models (HMMs), are usually employed in these applications. The latter class of problems focuses on identifying the underlying (system or environment) states and correlations resulting from a system's interaction with its environment. The main goal in these problems is to design an abstract representation (model) of the real system-environment interaction and use it for understanding and reasoning about the system (Madni and Sievers 2018). This paper focuses on adapting existing modeling techniques for the latter class of problems that require ongoing decision-making in complex systems operating in dynamic uncertain environments.

Various techniques, such as time-varying Gaussian graphical models (Talih and Hengartner 2005; Xuan and Murphy 2007), dynamic Bayesian networks (DBNs) (Robinson and Hartemink 2009), and different Markovian models, are employed for designing models based on state variables and correlations. In this paper, we focus on Markov model family because they offer a strong mathematical framework and probabilistic structure capable of modeling systems for a wide range of applications. These models perform well in practice if applied in the right way when modeling complex systems (Rabiner 1989). For instance, HMMs (also viewed as stochastic generalization of finite-state automata) are examples of Markovian models in which state variables and dependencies are modeled as states and probability distributions, respectively. Markovian models are also employed for developing modeling tools, such as state diagrams that are mainly used in MBSE and control system design (Madni and Sievers 2018; Wray and Zilberstein 2019).

To make decisions and plan actions with respect to state variables and their correlations, decisions and their influences on state variables should be embedded within the system model. Markov decision processes (MDPs) are Markov models that capture the transitions and correlations between various state variables during system-environment interactions that occur during system's decision-making. Basically, MDPs can be viewed as Markov chains that include decisions within the model that allows for making decisions over time (Alagoz et al. 2010). POMDPs are generalization of MDP models extended to a probabilistic domain in which uncertainty regarding the state of the system model is allowed, and state variable information is completely hidden or only partially known. This implies that an observation (signal) from the environment (1) can identify more than one state at a time or (2) cannot be explained based on the existing state variables and correlations in the model. The former implies that the most probable states should be considered, instead of one unique state, while the latter means that a new "hidden state" is required in the model to represent the new observation. POMDPs are widely used in modeling sequential planning in various applications in which systems interface noisy and uncertain environments (Cassandra 1998). For instance, Hubmann et al. (2017), Song et al. (2016), and Ulbrich and Maurer (2013) employ POMDPs for decision-making in autonomous vehicles (AVs) where there exist uncertainties in observed information and intensions of passengers and drivers. Machine maintenance, structural inspection, machine vision, search and rescue, and target identification are other examples of complex systems with uncertain environments where POMDPs are successfully employed for planning and decision-making (Cassandra 1998). Generally, with POMDP applications, the former definition of hidden information is widely addressed through the probability distributions within the model. However, the latter definition of hidden information, i.e., unknown-unknowns, should also be considered in the model design to ensure accurate and correct response when faced with unknown-unknowns.

In this paper, we propose an extended version of the standard POMDP models that accounts for unknown-unknowns and unexplainable signals, in addition to uncertainties, by incorporating new hidden states, expanding the model, and retuning the probability distributions when needed (Madni et al. 2018a; b; Sievers et al. 2019a, b). Also, an online look-ahead heuristic search algorithm is provided that solves the extended POMDP model by estimating the possible future decision paths for each available decision and calculating the expected long-term reward associated with that decision and path. An example of model setup and decision-making using the new POMDP model and online algorithm for a simulated autonomous vehicle (AV) in a specific scenario is also provided.

2 Review of Markov Models and Decision Processes

2.1 Markov Models and Hidden Markov Models (HMMs)

A Markov model (chain) is defined as a triplet $<S, T, \pi>$ in which S denotes a finite set of states that are directly observable from the system-environment interaction, $T : S \times S \to [0, 1]$ is the transition function (matrix) that includes the probabilities associated with transitioning from a state to another, and $\pi : p(s_i) \to [0, 1]$ where $s_i \in S$ is the initial state probability distribution. In contrast with the Markov models

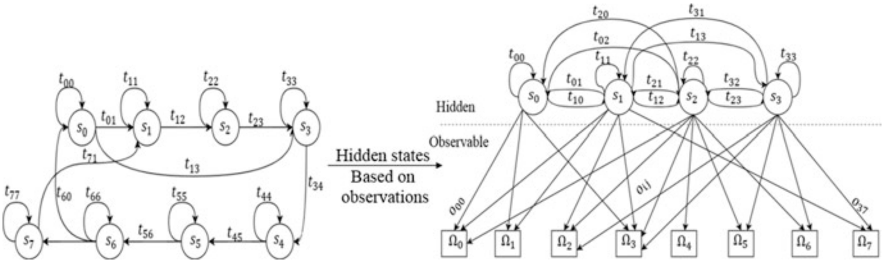


Fig. 1 (**a**) A Markov chain with eight states. (**b**) An HMM with four hidden states and eight observations. $t_{i,\,j}$ and $o_{i,\,k}$ transition and emission probabilities, respectively

that assume state variables and their changes over time are directly observable, HMMs assume an underlying "hidden" process associated with state variables that is modeled as a Markov chain and that process is obtained from noisy observations. In other words, an HMM is a statistical Markov model. Basically, an HMM can be defined as a tuple <S, Ω, T, O, π> in which S denotes the state space in the model; $\Omega = \{\mathrm{o}_0, \ldots, \mathrm{o}_\mathrm{n}\}$ is the observation space in which distinct observations associated with states are defined; $T : S \times S \rightarrow [0, 1]$ shows the transition matrix; $O : S \times \Omega \rightarrow [0, 1]$ represents the observation probabilities in the emission matrix, where $O_i(o_j)$ is the probability associated with observing o_j at state s_i at time t; and $\pi : p(s_i) \rightarrow [0, 1]$ defines the probabilities associated with being at a state at time $t = 0$ (Rabiner 1989). Figure 1a shows a Markov model where various observations from the system-environment interaction are modeled as individual states, whereas Fig. 1b shows the same example with hidden states identified based on observations.

2.2 Markov Decision Processes (MDPs)

To keep track of the impacts of transitions in between states, values can be embedded within the model definition. These values can be defined with respect to an objective or goal defined for the model under a specific scenario. Moreover, the capability of making decisions or reacting to changes in state variables can also be integrated with a Markov model. Adding the ability of making decisions based on observed changes in state variables to a Markov model and storing the impacts of changes and transitions defines an MDP model. In general, an MDP model is defined as a tuple <S, A, T, R> where S identifies a set of finite states (state space), A identifies an action space, $T : S \times A \times S \rightarrow [0, 1]$ represents the transition function that identifies the transition probabilities between states based on actions, and $R : S \times A \times S \rightarrow \mathfrak{R}$ shows the reward function which identifies the rewards or penalties associated with being in a state and making a decision. The overall objective in MDPs is to find the most optimal mapping between the actions and states, so-called optimal policy that maximizes the sum of long-term rewards by achieving the goal of the MDP using minimum possible number of decisions or in the shortest time. A commonly applied approach for finding an optimal policy associated with an MDP model is using value iteration (Eq. 2) that employs dynamic programming for solving the Bellman's equation in an iterative process until the optimal value is achieved. For each state at time t, the action corresponding to the maximum value is considered the optimal mapping between that state and available actions (Eq. 3):

$$V_t^*(s) = \max_{a \in A} \sum_{s'} p\left(s|s', a\right)\left[R\left(s|s', a\right) + \gamma V_{t-1}^*\left(s'\right)\right] \tag{2}$$

$$\pi_t^*(s) = argmax_{a \in A} \sum_{s'} p\left(s|s', a\right)\left[R\left(s|s', a\right) + \gamma V_{t-1}^*\left(s'\right)\right] \tag{3}$$

2.3 Partially Observable Markov Decision Processes (POMDPs)

POMDPs are generalization of MDPs to uncertain environments where partially available data could potentially result in incomplete information about the state space. Uncertainty may appear as (1) uncertainty in actuation, whether an action is carried out successfully; (2) uncertainty in sensor and data interpretation due to sensor noise and limited sensor capabilities; (3) uncertainty about the environment; and (4) uncertainty about intensions of other systems in the environment (Koenig and Simmons 1998; Bai et al. 2015). In contrast with the MDP models that assume full access to state space, partial observability implies that the system only receives an indication of its current state that only allows for probabilistic identification of the state. A POMDP model can be defined as a tuple <S, A, Ω, T, O, R> in which S determines a finite state space which is hidden; A identifies a finite set of actions; $\Omega = \{o_0, \ldots, o_n\}$ is a finite set of observations; $T : S \times A \times S \rightarrow [0, 1]$ is the transition function that identifies the probabilities associated with transitions in between states; $O : S \times A \times \Omega \rightarrow [0, 1]$ defines the emission function (or matrix), which provides the probabilities associated with performing an action in a state and observing an observation from the observation space; and finally $R : S \times A \times S \rightarrow \Re$ is the reward function that provides rewards/penalties associated with performing an action in a state and transitioning to another state (Spaan 2012). Figure 2 shows the differences between a problem modeled using both an MDP and a POMDP model with four states $S = \{s_0, s_1, s_2, s_3\}$ and three actions $A = \{a_0, a_1, a_2\}$. In this figure, $[t_0, t_1, t_2]_{i,j}$ and $[r_0, r_1, r_2]_{i,j}$ represent the transition probabilities and rewards/penalties of performing actions $[a_0, a_1, a_2] \in A$ at state s_i and transitioning to state s_j in both MDP and POMDP models, respectively. In addition, $[o_0, o_1, o_2]_{i,k}$ are the emission probabilities of observing $o_k \in \Omega$ after performing $[a_0, a_1, a_2] \in A$ at state s_i in the POMDP model.

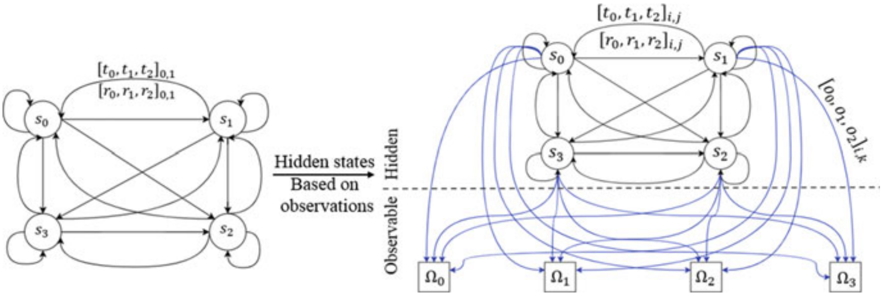


Fig. 2 A problem modeled with an MDP model if the states are observable (left) and a POMDP model where the full-observability assumption is relaxed

Since the state space is only partially observable, POMDPs employ a probabilistic distribution, so-called belief, over the state space to determine the most recent state based on received observations. At time $t = 0$, with no available observation, the belief can be initialized as a uniform distribution over all the states. Later, as the system interacts with the environment and receives feedback (i.e., observations and rewards), the belief vector gets updated based on Bayes' rule as follows:

$$b^{t+1}(s_i) = p\left(s_i | b^t, a, o\right) = \frac{p(o|s_i, a) \sum_{s \in S} p(s_i|s, a)\, b^t(s)}{\sum_{s' \in S} p(o|s', a) \sum_{s \in S} p(s'|s, a)\, b^t(s)} \tag{4}$$

where $p(o|\, s_i, a)$ and $p(s_i|\, s, a)$ show the emission probability of performing a at s_i and observing o and the probability of transitioning to s_i after performing action a at state s. A POMDP can be formulated as an MDP to find the optimal policies associated with all possible belief states by solving Bellman's equation using techniques such as dynamic programming (Eq. 5):

$$V_t^*\left(b^t\right) = \max_{a \in A}\left[\sum_{s \in S} b^t(s) R(s, a) + \gamma \sum_{o \in \Omega} p\left(o|b^t, a\right) V_{t-1}^*\left(b|b^t, a, o\right)\right] \tag{5}$$

The techniques, such as dynamic programming, that evaluate every imaginable belief and action pair and provide an optimal policy prior to execution are known as "offline algorithms." Offline algorithms assume that the initial model setup and the environment are fixed. While the offline algorithms can achieve very good performance, they often take significant amount of time, e.g., hours, to solve slightly large problems in which there exist numerous possible situations to consider (Ross et al. 2008). On the other hand, online algorithms circumvent the complexity of computing a policy by only considering the current belief and a small horizon to search for contingency plans. Since online algorithms evaluate the actual belief achieved from real interactions between a system and its environment, they can handle changes in the environment (e.g., changes in goals) without recomputing the full policy for the whole model (Sunberg and Kochenderfer 2018; Ye et al. 2017).

3 Proposed POMDP and Solution

In dynamic and uncertain environments, there is no guarantee that all information is initially known and considered in the model. This means that there may be observations that cannot be explained using the existing states in the model, which require expanding the current state space to include "new hidden states," when discovered during the execution phase. The current definition of the POMDPs and offline solutions are not able to accommodate the issues associated with unknown-unknowns.

To accommodate this issue, we further extended the standard definition of the POMDP models to allow for expanding the model and incorporating new hidden states. Thus, the definition of the POMDP updates to a tuple $<S^+, A, \Omega^+, T^+, O^+, R^+>$ in which $S^+ : S \cup H$ is the extended finite set of states including the hidden state(s), H, and $\Omega^+ : \Omega \cup \Theta$ is the extended finite set of observations. Initially, H and Θ are empty sets and the model is a tuple of $<S, A, \Omega, T, O, R>$. As the model discovers unknown-unknowns during its interaction with its environment, the sets are accumulated by the new observations and hidden states. $T^+ : S^+ \times A \times S^+ \rightarrow [0, 1]$ is the extended transition function that includes the probabilities of transitioning to and from the hidden states, $R^+ : S^+ \times A \times S^+ \rightarrow \Re$ determines the extended reward function, and $O^+ : S^+ \times A \times \Omega^+ \rightarrow [0, 1]$ identifies the extended observation function that contains the probabilities of observing both $o \in \Omega$ and $o^{'} \in \Theta$ (Fig. 3). On the other hand, definition of states, observations, and reward function in the proposed POMDP model are slightly different from a standard POMDP model. **State space** is defined based on various, high-level events that generally describe different conditions in the system-environment interaction. Generalization of the state space to high-level events helps with reducing the number of state space and decreasing the related computational complexities.

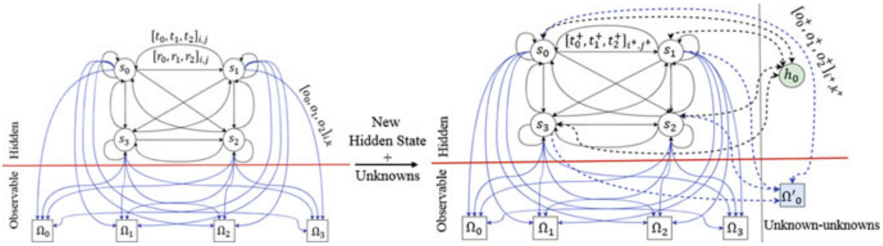


Fig. 3 Example of how a POMDP expands as a new hidden state is added to incorporate an unknown-unknown

Moreover, variables that identify the goal and failure events are also modeled as states in the model. Since the goal and failures are also embedded as states within a model, the **reward function** assigns rewards/penalties ($r > 0$ or $r < 0$) to the states depending on whether a state is identified as goal, failure, or transient, which changes the reward matrix to a reward vector $R^+ : 1 \times S^+ \rightarrow \Re$. **Optimal policy** in the proposed POMDP is identified as an action that updates the belief so that, considering all possible observations in the model, the goal state has a higher probability (failure has less probability) in the future belief. In other words, the optimal policy is the mapping between the belief and actions that maximizes the long-term sum of rewards by constantly moving along a plan trajectory that improves the belief state (Eq. 6):

$$\pi_t^*\left(b^t\right) = argmax_{a \in A}\left[\sum_{o \in \Omega^+}\sum_{s \in S^+} p\left(o|s,a\right)\left(b^{t+1}(s)|b^t(s),a,o\right)R^+(s)\right] \tag{6}$$

Thus, since the belief represents the probabilities assigned to most probable states and optimal policy improves the belief probability distribution by assigning a higher probability to the goal state, the combination can be used for explaining the decisions made and reasoning about the model.

3.1 *Initializing New Hidden States*

Hidden states are associated with the events that cannot be interpreted from the initialized state in the model state space (Sievers et al. 2019a, b). When a new hidden state is initialized in the proposed POMDP model, the transition and emission probabilities of the hidden state are empirically initialized using the following heuristics depending on the model and scenario:

Heuristic 1: "Expected Outcome: Assign higher probabilities to the most expected states/observations based on how the action changes the state variables of the model." After performing a certain action (e.g., a), depending on the influence of the action on the predetermined state variables, the most expected observations or states are assigned with higher probabilities, and the less expected ones will have lower probabilities.

Heuristic 2: "Safest Outcome: Assign probabilities so that the optimal policy associated with the hidden state is the safest action (e.g., neutral action) or updates the belief so that the safest/most neutral state receives a higher probability."

On the other hand, transition and emission probabilities *from known states* to the hidden state(s) are initially assigned with small probabilities (e.g., 0.01), because this transition (or emission) is not repeated enough compared to the known states.

3.2 N-Step Look-Ahead (Online Policy Estimation Algorithm)

Since the hidden states and unexplainable observations are discovered during the execution phase, there exists no prior information, such as pre-planned policies, associated with them. Thus, even if an offline solution can efficiently calculate and estimate optimal policies for a model prior to execution, the solution still lacks the optimal policies associated with the newly added hidden state. In other words, offline algorithms also lack the ability of updating a pre-estimated solution when a model or environment changes that results in changing the model objective or goal (Ross et al. 2008). This implies that the offline algorithms are not applicable to the problems with highly dynamic environments and objectives, because they require to recompute the optimal policies after any changes. On the other hand, online algorithms that rely on combining offline calculations in estimating optimal policies during the look-ahead search in estimated future beliefs are not sufficient, since there is no prior information that exists for newly initialized hidden states. To this end, we implemented the online, "N-Step Look-Ahead," policy estimation algorithm that (1) defines a belief tree with the current belief state as its top node, "root"; (2) recursively, explores the possible plan/decision paths by traversing the expected beliefs located on the lower levels of the tree; and (3) calculates the expected long-term rewards for available possible plans in that tree to select the plan with the highest long-term reward.

The algorithm recursively expands the belief states at each level ($l \leq N$) until it reaches the deepest level (N) or a termination condition for a belief is met. The tree is explored bottom to top and left to right, meaning that initially the values associated with the leftmost branches are calculated starting from the bottom of the tree and moving to the top node, then the next branch is explored, and value is calculated until all branches are traversed (Eq. (7)). A learning rate $0 < \gamma < 1$ is also considered, so that the largest rewards are collected as early as possible. The depth of the tree determines the finite horizon for the look-ahead search. Basically, at each level of the tree, the expected beliefs at level $l < N$ are calculated based on the beliefs at level $l - 1$, available actions, and possible observations:

$$V^N\left(b^t, a\right) = \sum_{o\in\Omega^{+'}} \Pr\left(o|\, b^t, a\right) * \gamma \sum_{a'\in A} V^N_{l+1}\left(b^{t+1}_{a,o}, a'\right) \tag{7}$$

Since this algorithm is designed to provide a policy anytime it's required, the execution time of the algorithm (time complexity) is very important. As N increases, the search algorithm explores deeper levels and the estimated long-term reward becomes more accurate. On the other hand, larger N requires more computation time, so there is a trade-off between the accuracy and execution time. Various techniques, such as sampling and heuristic search, are employed to reduce the computation time of a search algorithm (Ye et al. 2017; Kurniawati and Yadav 2016; Etminan and Moghaddam 2018a; b). We employ a heuristic search that reduces the time complexity of the search by only considering the exploration of the belief nodes that satisfy the conditions defined in our heuristics. Our heuristic summarizes as follows:

Search Heuristic1: "Expand non-terminal/non-failure belief nodes." Using this heuristic, the search algorithm only expands and explores the belief nodes that have low probabilities assigned to failure states.

Search Heuristic2: "Expand the belief nodes using possible actions and associated reachable observations only." Based on this heuristic, the belief nodes in the lower levels (*l-1*) of the tree are calculated based on observations with high probabilities from a parent belief node at level *l* and an action *a*. Reachable observations are identified using the criteria represented in Eq. 8:

$$\mathrm{o} \in \Omega^{+'}\left(b^t, a\right) \; iff \; \sum_{s \in S^+} b^t(s) p\left(o|s, a\right) \geq L \tag{8}$$

Where $\Omega^{+'} \subseteq \Omega^+$ denotes the reachable observation and $0 \leq L \leq 1$ is the minimum reachability probability defined empirically based on the size or the problem and emission matrix. Figure 4 shows how applying the heuristic search reduces the computation time in the N-Step Look-Ahead search from an exponential growth rate to a linear growth rate for a given model that includes four states and three actions.

4 An Exemplar POMDP Model

In an exemplar scenario, we simulated an AV in a multilane freeway using PythonVTK. As shown in Fig. 5a, the AV (green) is surrounded by traffic in different lanes (different relative distances and velocity/speed). The AV has two main objectives with respect to its surrounding environment (freeway + traffic). These are (1) safely drive within one lane and (2) safely change lanes when it becomes necessary (Pouya and Madni 2020a).

For the purpose of this paper, we define a POMDP model, including the states and probabilities associated with the former objective, and test the model in the simulated multilane freeway using the N-Step Look-Ahead function for constant decision-making. Later, we tune the parameters and demonstrate model expansion for including hidden states. The initial step in POMDP model definition is identification of candidate states with respect to various high-level general conditions. However, due to partial observability (e.g., sensor noise and hidden driver intensions), the states can be identified based on observations. Figure 5b demonstrates various observation classes (candidate states) defined based on vehicle speed and relative distances (difference between distance in front and rear, $dF - dR$) using the simulated traffic data around the AV with two different traffic setups.

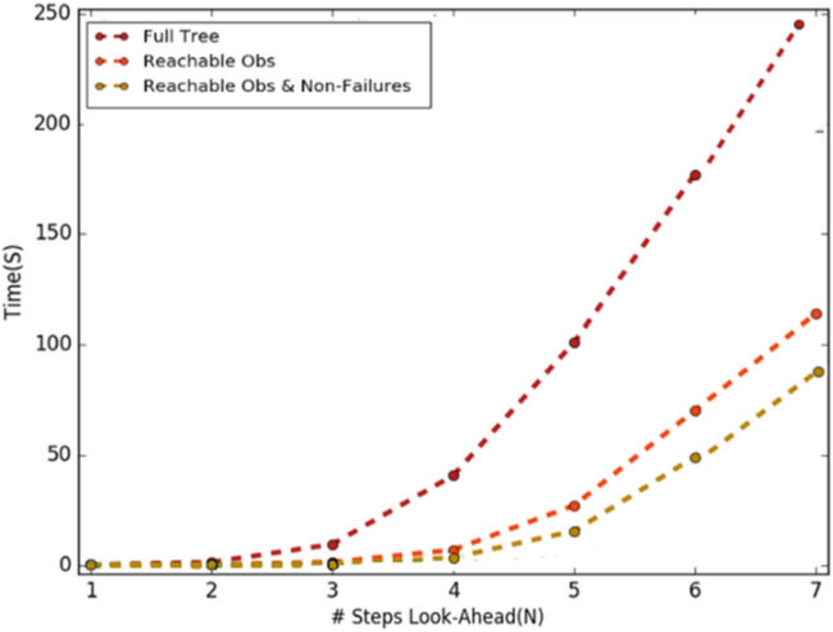
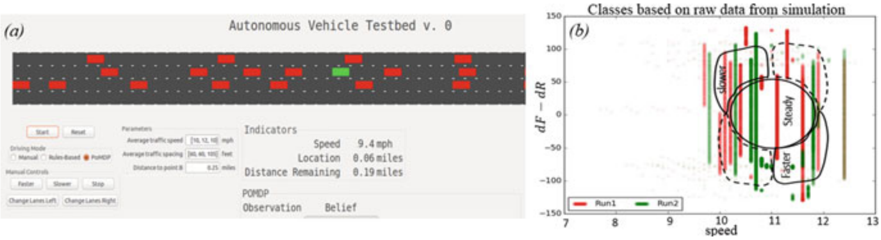


Fig. 4 Look-ahead policy estimator computation time as different heuristics are applied



Parameters
Average traffic speed [10, 12, 10] mph
Average traffic spacing [60, 60, 100] feet
Distance to point B 0.25 miles

Indicators
Speed 9.4 mph
Location 0.06 miles
Distance Remaining 0.19 miles

POMDP
Observation Belief

Fig. 5 (**a**) Traffic simulation dashboard. (**b**) High-level traffic conditions (patterns) based on data

The datapoints that fall into the dashed classes are observations that cannot fully determine a state, which we refer to as noisy and partially available observations. In addition, according to pre-determined speed and distance limits in the simulation, the datapoints that exceed these limits are identified as failures or crashes. The next step after identifying the state candidates is defining the probabilities and reward values. Since the objective of the model is to drive safely within a lane, the goal state is s_2 : *safe and steady* with $R(s_2) = +10$, the failure state is s_3 : *crashed/failure* with $R(s_3) = -20$, and s_0 : *slower*, s_1 : *faster* are transient states with $R(s_0) = R(s_1) = +1$ reward values. In addition, actions associated with this model are a_0 : *maintain status quo*, a_1 : *speed up*, and a_2 : *slow down*. The transition and emission matrices can be learned from the simulation data or can be initialized based on expert’s judgment and tuned within the simulation. Later, the probability and reward matrices are expanded as the model receives an observation that cannot be explained using the current state space.

To expand and initialize the probability matrices when a new observation is realized, we have applied the “most expected outcome” heuristic. As an example, the transition probability associated with h_0 and a_1 has the highest probability assigned to s_1, because the vehicle expects to drive faster as it applies a_1 and speeds up.

In this simulation and for the purpose of this exemplar scenario, hidden observations are generated as outputs of a random function invoked in random times in the simulation. After tuning the probabilities and defining the extension technique within the POMDP model, the model is tested in the simulation by having the N-Step Look-Ahead function evaluate the possible decisions at every time step based on the most recent belief. N equal to 2 is selected for the depth of look-ahead search with a sampling rate of 0.1s, and the POMDP decisions and performance are compared to a rule-based algorithm designed based on time-to-collision measurements.

Figure 6 (left) shows a series of changes in the AV’s belief, and the right figure represents the values for possible actions estimated for each belief with N = 2.

As a new hidden state is identified and the belief is expanded at $t = 15$, the look-ahead value estimation decides to maintain status quo (a_0) as long as the belief probability assigned to the hidden state is high but changes its decision as soon as the belief in the known states goes higher. The dashed line demonstrates the sum of long-term rewards associated with the belief series. Figure 7 demonstrates the performance of the POMDP model in comparison with a rule-based algorithm that makes decisions based on TTC criteria with full observability. As shown in the figure, the overall pattern of the decisions made by the POMDP matches the rule-based with full-observability pattern. However, the number of the changes in decisions made by the POMDP is smaller (smoother pattern) than the rule-based, which implies that the rule-based is more aggressive and reacts to every single

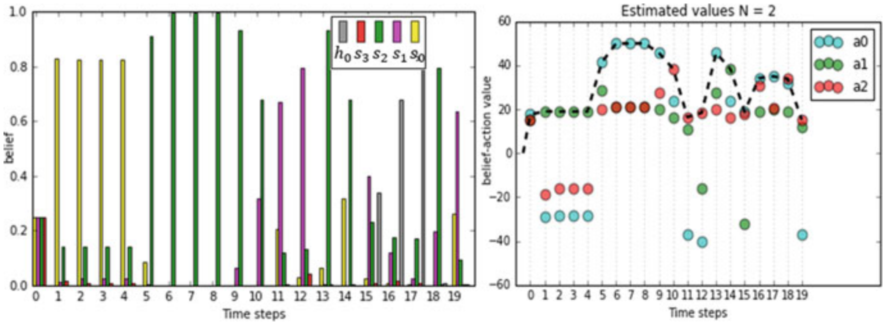


Fig. 6 (left) Belief updates, expanded at t = 15 to include a new hidden state; (right) estimated long-term reward (dashed line)

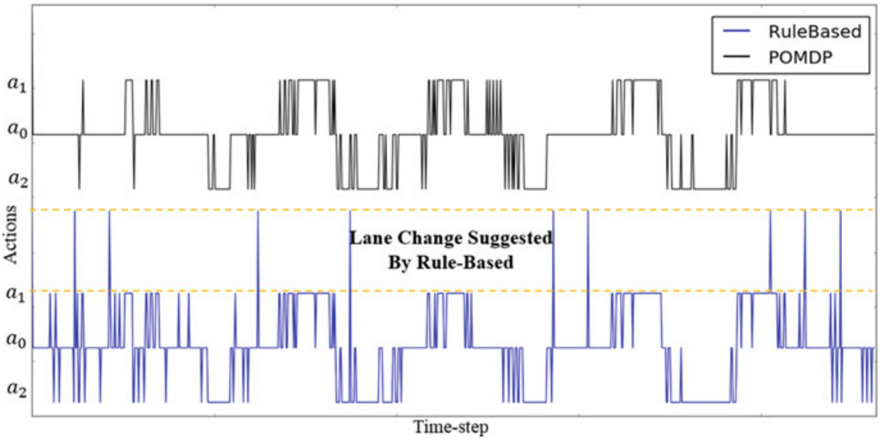


Fig. 7 POMDP performance evaluation and comparison with a TTC rule-based. (Pouya and Madni 2020b)

observation. In contrast, the POMDP ensures about the consistency of the received observation and reacts when its belief with respect to the received observation is high.

5 Summary and Future Work

In this paper, we emphasized the importance of understanding the behavior of a complex system from its interactions with its environment to design accurate models for resilient decision-making with partially available data. We presented an extendable-POMDP model that is initialized using available information, and then adapts to new information by incorporating new hidden states, and thereby extends the related probability distributions using heuristics, so they can be learned incrementally. The flexibility introduced by incorporating new hidden states results in risk associated with evaluating the accuracy of decisions made for the hidden states with less or no prior information about the state, which we manage by employing heuristics in model expansion. To address the risk associated with computational complexity, the N-Step Look-Ahead online value estimation algorithm is employed. This algorithm uses heuristic search, to solve the extendable-POMDPs in an anytime fashion. We intend to extend the work presented in this paper to realize a probabilistic modeling paradigm that can be used for decision-making and planning of complex systems and system of systems that operate in highly dynamic, uncertain environments. For model testing and verification purposes, we currently compare with rule-based algorithms and full observability, but in the future, we intend to employ machine learning (e.g., Q-learning (Pouya and Madni 2020c)) and formal reasoning methods to create a formal verification technique for POMDP models.

References

Alagoz, O., H. Hsu, A.J. Schaefer, and M.S. Roberts. 2010. Markov Decision Processes: A Tool for Sequential Decision Making Under Uncertainty. *Medical Decision Making 30* (4): 474–483.

Bai, H., S. Cai, N. Ye, D. Hsu, and W.S. Lee. 2015. Intention-aware Online POMDP Planning for Autonomous Driving in a Crowd. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, pp. 454–460.

Cassandra, A.R. 1998. A Survey of POMDP Applications. In *Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes*, Vol. 1724.

Etminan, A., and M. Moghaddam. 2018a. Electromagnetic Imaging of Dielectric Objects Using a Multidirectional-Search-Based Simulated Annealing. *IEEE Journal on Multiscale and Multiphysics Computational Techniques 3*: 167–175.

——— 2018b. A Novel Global Optimization Technique for Microwave Imaging Based on the Simulated Annealing and Multi-Directional Search. In *2018 IEEE International Symposium on Antennas and Propagation & USNC/URSI National Radio Science Meeting,* IEEE, pp. 1791–1792.

Hubmann, C., M. Becker, D. Althoff, D. Lenz, and C. Stiller. 2017. Decision Making for Autonomous Driving Considering Interaction and Uncertain Prediction of Surrounding Vehicles. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, 1671. 1678: IEEE.

Koenig, S., and R. Simmons. 1998. Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models. *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, (Partially), pp. 91–122.

Kurniawati, H., and V. Yadav. 2016. An Online POMDP Solver for Uncertainty Planning In Dynamic Environment. In *Robotics Research*, 611–629. Cham: Springer.

Madni, A.M., and M. Sievers. 2018. Model-Based Systems Engineering: Motivation, Current Status, and Needed Advances. In *Disciplinary Convergence in Systems Engineering Research*, 311–325. Cham: Springer.

Madni, A.M., M. Sievers, A. Madni, E. Ordoukhanian, and P. Pouya. 2018a. Extending Formal Modeling for Resilient Systems Design. *INSIGHT 21* (3): 34–41.

Madni, A., D. Erwin, A. Madni, E. Ordoukhanian, and P. Pouya. 2018b. *Formal Methods in Resilient Systems Design using a Flexible Contract Approach* (No. SERC-2018-TR-119). SYSTEMS ENGINEERING RESEARCH CENTER HOBOKEN NJ HOBOKEN United States.

Pouya, P., and A.M. Madni. 2020a. Expandable-Partially Observable Markov Decision-Process Framework for Modeling and Analysis of Autonomous Vehicle Behavior. *IEEE Systems Journal*. https://doi.org/10.1109/JSYST.2020.30.

———. 2020b. Leveraging Probabilistic Modeling and Machine Learning in Engineering Complex Systems and System-of-Systems. In *AIAA Scitech 2020 Forum*, p. 2117.

———. 2020c. A Probabilistic Online Policy Estimation for Autonomous Systems Planning and Decision Making. *In 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* IEEE.

Rabiner, L.R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE 77* (2): 257–286.

Robinson, J.W., and A.J. Hartemink. 2009. Non-stationary Dynamic Bayesian Networks. *Advances in Neural Information Processing Systems*: 1369–1376.

Ross, S., J. Pineau, S. Paquet, and B. Chaib-Draa. 2008. Online Planning Algorithms for POMDPs. *Journal of Artificial Intelligence Research 32*: 663–704.

Sievers, S., A.M. Madni, and P. Pouya. 2019a. Trust and Reputation in Multi-agent Resilient Systems. In *2019 International Conference on Systems, Man, Cybernetics (SMC)*, 741–747. IEEE.

Sievers, M.M., A.M. Madni, and P. Pouya 2019b. Assuring Spacecraft Swarm Byzantine Resilience. In *AIAA Scitech 2019 Forum*, p. 0224.

Song, W., G. Xiong, and H. Chen. 2016. Intention-Aware Autonomous Driving Decision-Making in an Uncontrolled Intersection. In *Mathematical Problems in Engineering, 2016*.

Spaan, M.T. 2012. Partially Observable Markov Decision Processes. In *Reinforcement Learning*, 387–414. Berlin/Heidelberg: Springer.

Sunberg, Z.N., and M.J. Kochenderfer. 2018. Online Algorithms for POMDPs with Continuous State, Action, and Observation Spaces. In *Twenty-Eighth International Conference on Automated Planning and Scheduling*.

Talih, M., and N. Hengartner. 2005. Structural Learning with Time-Varying Components: Tracking the Cross-Section of Financial Time Series. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67* (3): 321–341.

Ulbrich, S., and M. Maurer. 2013. Probabilistic Online POMDP Decision Making for Lane Changes in Fully Automated Driving. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 2063–2067. IEEE.

Wray, K.H., and S. Zilberstein. 2019. Generalized Controllers in POMDP Decision-Making. In *2019 International Conference on Robotics and Automation (ICRA)*, 7166–7172. IEEE.

Xuan, X., and K. Murphy. 2007. Modeling Changing Dependency Structure in Multivariate Time Series. In *Proceedings of the 24th International Conference on Machine Learning*, 1055–1062. ACM.

Ye, N., A. Somani, D. Hsu, and W.S. Lee. 2017. Despot: Online Pomdp Planning with Regularization. *Journal of Artificial Intelligence Research 58*.