



POMDP inference and robust solution via deep reinforcement learning: an application to railway optimal maintenance

Giacomo Arcieri¹ · Cyprien Hoelzl¹ · Oliver Schwery² · Daniel Straub³ · Konstantinos G. Papakonstantinou⁴ · Eleni Chatzi¹

Received: 3 April 2023 / Revised: 23 November 2023 / Accepted: 26 April 2024 /
Published online: 31 May 2024
© The Author(s) 2024

Abstract

Partially Observable Markov Decision Processes (POMDPs) can model complex sequential decision-making problems under stochastic and uncertain environments. A main reason hindering their broad adoption in real-world applications is the unavailability of a suitable POMDP model or a simulator thereof. Available solution algorithms, such as Reinforcement Learning (RL), typically benefit from the knowledge of the transition dynamics and the observation generating process, which are often unknown and non-trivial to infer. In this work, we propose a combined framework for inference and robust solution of POMDPs via deep RL. First, all transition and observation model parameters are jointly inferred via Markov Chain Monte Carlo sampling of a hidden Markov model, which is conditioned on actions, in order to recover full posterior distributions from the available data. The POMDP with uncertain parameters is then solved via deep RL techniques with the parameter distributions incorporated into the solution via domain randomization, in order to develop solutions that are robust to model uncertainty. As a further contribution, we compare the use of Transformers and long short-term memory networks, which constitute model-free RL solutions and work directly on the observation space, with an approach termed the belief-input method, which works on the belief space by exploiting the learned POMDP model for belief inference. We apply these methods to the real-world problem of optimal maintenance planning for railway assets and compare the results with the current real-life policy. We show that the RL policy learned by the belief-input method is able to outperform the real-life policy by yielding significantly reduced life-cycle costs.

Keywords Partially observable Markov decision process · Reinforcement learning · Deep learning · Model uncertainty · Optimal maintenance

Editors: Emma Brunskill, Minmin Chen, Omer Gottesman, Lihong Li, Yuxi Li, Yao Liu, Zonging Lu, Niranjani Prasad, Zhiwei Qin, Csaba Szepesvari, Matthew Taylor.

Extended author information available on the last page of the article

1 Introduction

Partially Observable Markov Decision Processes (POMDPs) offer a mathematically sound framework to model and solve complex sequential decision-making problems (Drake, 1962; Sondik, 1971; Cassandra, 1998). POMDPs account for the uncertainty associated with observations in order to derive optimal policies, namely a sequence of optimal decisions that minimize/maximize the total costs/rewards over a prescribed time horizon, under stochastic and uncertain environments. Stochasticity can indeed be incorporated both in the evolution of the hidden states over time, i.e., the transition dynamics, and in the process that generates the observations, which reflect only a partial and/or noisy information of the actual states.

POMDPs form a potent mathematical framework to model optimal maintenance planning for deteriorating engineered systems (Papakonstantinou & Shinozuka, 2014a). In such problems, a perfect information of the system's condition (state) is generally not available or feasible to acquire, due to the problem's scale, inherent noise of sensing instruments, and associated costs limitations. By using sensors and inferred associated condition indicators, Structural Health Monitoring (SHM) tools, as described by Farrar and Worden (2012) and Straub et al. (2017), can provide estimates of the structural state. However, the provided observations are often incomplete and susceptible to noise, which limits their ability to accurately determine the true state of the system. Consequently, decision-making must occur in the face of uncertainty. Within a POMDP scheme, the decision maker (or agent) receives an observation from the environment, which in these cases reflects a measurement that is delivered by an SHM system, and uses this to form a belief about the current state of the system. Based on this belief, the agent takes an action, which will impact the future condition of the system. The POMDP objective is to find the optimal sequence of maintenance actions that minimizes the expected total costs over the operating life-cycle.

POMDP modeling has repeatedly been applied in the context of optimal maintenance planning. Madanat and Ben-Akiva (1994) model the highway pavement maintenance planning as a POMDP, where the deterioration level is discretized in 8 hidden states, accessed through noisy observations that are delivered by different measurement possibilities. Ellis et al. (1995) apply the framework to the problem of maintenance of highway bridges, where 5 deterioration levels are used, under availability of uncertain inspection information, and 4 maintenance actions. Memarzadeh et al. (2015) propose POMDP modeling of the wind farm maintenance problem using 3 damage states of the turbine, 4 types of available noisy observations and 3 available maintenance actions. In Schöbi and Chatzi (2016) a deteriorating bridge maintenance planning problem is modeled as a POMDP using a continuous space of deterioration levels, coupled with a discrete set of actions and observations that are available from both monitoring and inspection. Papakonstantinou et al. (2018) illustrate two different POMDP formulations on the problem of maintenance for deteriorating bridge structures characterized by both stationary and non-stationary dynamics. In tackling non-stationarity, they adopt a high dimensional vector of discrete hidden states, 4 possible discrete observations, and 10 available actions, while combining inspection and maintenance decisions into the actions space. Kivanç et al. (2022) formulate and solve the maintenance problem of a regenerative air heater system in a coal-based thermal power plant, composed of 6 components, using a POMDP model with a factored structure, where each component can assume between 2 or 3 different discrete hidden states and a set of 2 possible (maintenance) actions is available per component.

POMDP solutions assume knowledge of the transition dynamics and the observation generating process. This implies strict prior assumptions on the POMDP model parameters that govern the deterioration, the effects of maintenance actions, and the relation of observations to latent states and variables. When a POMDP model is available, the solution can be computed via Dynamic Programming (DP) (Bertsekas, 2012) and approximate methods (Papakonstantinou & Shinozuka, 2014b) with optimality convergence guarantees, when the complexity of the problem is not prohibitive, or via Reinforcement Learning (RL) schemes (Sutton & Barto, 2018) through samples and trial and error learning. While RL methods can relax some assumptions on the POMDP knowledge, a simulator that can reliably describe the POMDP model is still necessary for inference and testing purposes, particularly for engineering problems and in infrastructure asset management applications.

However, a full POMDP model of the problem is rarely available in real-world applications, and the inference of all POMDP parameters that form the transition dynamics and the observation generating process of the problem can be quite challenging. The availability of such a model is a key issue that prevents wide adoption of the POMDP framework and its solution methods (including RL) for real-world applications. Available literature on the theme of maintenance planning is focused on developing RL methods to solve complex POMDP problems, as pioneered by the work of Andriotis and Papakonstantinou (2019, 2021), while assuming knowledge of the POMDP transition and observation models, i.e., by for example assuming that the POMDP inference has already been carried out. Only few papers deal with the POMDP inference, which poses a challenge in itself, while best practices are not generally available. Papakonstantinou and Shinozuka (2014a), Song et al. (2022) and Wari et al. (2023) propose methods to estimate the state transition probability matrix for deterioration processes, but without demonstrating inference on the transition matrices associated with maintenance actions. Guo and Liang (2022) propose methods to estimate both the transition and the observation models, but do not consider model uncertainty and the implementation examples do not involve real-world data but only simulated ones.

In Arcieri et al. (2023), we tackle this key inference issue by proposing a framework to jointly infer all transition and observation model parameters entirely from available real-world data, via Markov Chain Monte Carlo (MCMC) sampling of a Hidden Markov Model (HMM), which is conditioned on actions. The framework, which can be practically implemented and can be tailored to the problem at hand, estimates full posterior distributions of POMDP model parameters. By considering these distributions in the POMDP evaluation, optimal policies that are robust with respect to POMDP model uncertainties are obtained.

In this work, we combine the POMDP inference with a deep RL solution. Most previous works on deep RL methods focus on fully observable problems, with RL solutions for POMDPs having received notably lower attention. Partial observability is usually overcome with deep learning architectures that are able to infer hidden states through memory and a history of past observations. Schmidhuber (1990) is one of first works that applied Recurrent Neural Networks (RNNs) for RL problems. Subsequently, Long Short-Term Memory (LSTM) networks have become the standard to handle partial observability (Dung et al., 2008; Zhu et al., 2017; Meng et al., 2021). Recent works propose to replace LSTM architectures with Transformers (GTrXL) (Parisotto et al., 2020). A third modeling option, which constitutes a hybrid approach between a DP and a RL solution, exploits the POMDP model to compute beliefs via Bayes' theorem, which are then fed to the deep RL algorithm as inputs to classical feed-forward Neural Networks (NNs) (Andriotis & Papakonstantinou, 2019, 2021; Morato et al., 2023). Namely, the POMDP problem is converted into the belief-MDP (Papakonstantinou & Shinozuka, 2014b; Andriotis et al., 2021) and then

solved with deep RL techniques. We compare these three available solution methods and propose a joint framework of inference and robust solution of POMDPs based on deep RL techniques, by combining MCMC inference with domain randomization of the RL environment in order to incorporate model uncertainty into the policy learning.

We showcase the applicability of these methods and of the proposed framework on the real-world problem of optimal maintenance planning for railway infrastructure. The observations in this case are delivered as on-board railway monitoring data, namely the so-called “fractal values” condition indicator, computed from field measurements and provided by SBB (the Swiss Federal Railways). The fractal value indicator is currently used over the Swiss railway network to detect track substructure damage and guide associated maintenance action, e.g. minor repair (so-called “tamping”) or renewal. However, the indicator, albeit useful, is an indirect and noisy observation and, thus, far from a perfect estimate of the actual railway condition. As such, the problem of optimal maintenance planning for railway assets, based on on-board monitoring data, can be naturally modeled as a POMDP.

The contributions of this work can be summarized as follows:

- We highlight two key issues that affect the implementation of RL solutions for real-world problems, which often tend to only be partially observable, namely i) the lack of availability of a POMDP model or simulator thereof, and ii) the lack of robustness of the solution to model uncertainty over the environment parameters.
- We address the above issues through a combined framework of POMDP inference and robust solution based on deep RL methods. The former is tackled by proposing a joint inference of all transition and observation model parameters entirely from available real-world data, via MCMC sampling of a HMM conditioned on actions. By recovering posterior samples over the uncertain parameters, the inference technique allows the incorporation of solutions methods that enhance the robustness over epistemic (environment) uncertainty. To this end, we propose a domain randomization of the environment parameters through the inferred posterior samples, enabling the RL agent to learn a policy optimized over all plausible parameters space.
- We demonstrate the efficacy of our approach by comparing this for three state-of-the-art deep RL solution methods for POMDPs, namely the use of LSTMs and Transformers directly on the observations space, and a third method, here termed the belief-input method, which exploits a (learned or known) model of the environment to transform the POMDP into a belief-MDP and works with classical feed-forward NNs on the belief space. To the best of our knowledge, no other work experimentally compares the performance of the latter method to the first two.
- Finally, the real-life railway application forms a salient contribution in itself, promoting the use of POMDP modeling and RL solution methods for infrastructure maintenance planning and demonstrating the applicability of the proposed methods starting from real-world (measurement) data.

The remainder of this paper is organized as follows. Section 2 provides the necessary background on POMDPs and prior work. Section 3 describes the considered maintenance planning problem of railway assets and the monitoring data. Section 4 describes the POMDP inference and its implementation to the problem here considered. Section 5 evaluates the three available modeling options of deep RL solutions for POMDPs, namely LSTM, GTrXL, and the belief-input case. Section 6 proposes our joint framework of POMDP inference and robust solution via deep RL and domain randomization. Finally, Sect. 7

concludes with a highlight and a discussion of the contributions, and outlines possible future work.

2 Preliminaries

2.1 Partially observable Markov decision processes

A POMDP can be considered as a generalization of a Markov Decision Process (MDP) for modeling sequential decision-making problems within a stochastic control setting, with uncertainty incorporated into the observations. A POMDP is defined by the tuple $\langle S, A, Z, R, T, O, b_0, H, \gamma \rangle$, where:

- S is the finite set of hidden states that the environment can assume.
- A is the finite set of available actions.
- Z is the set of possible observations, generated by the hidden states and executed actions, which provide partial and/or noisy information about the actual state of the system.
- $R : S \times A \rightarrow \mathbb{R}$ is the reward function that assigns the reward $r_t = R(s_t, a_t)$ for assuming an action a_t at state s_t .
- $T : S \times S \times A \rightarrow [0, 1]$ is the transition dynamics model that describes the probability $p(s_{t+1} | s_t, a_t)$ to transition to state s_{t+1} if action a_t is taken at state s_t .
- $O : S \times A \times Z \rightarrow \mathbb{R}$ is the observation generating process that defines the emission probability $p(z_t | s_t, a_{t-1}, z_{t-1})$, namely the likelihood to observe z_t if the system is at state s_t and action a_{t-1} was taken.
- b_0 is the initial belief on the system's state s_0 .
- H is the considered horizon of the problem, which can be finite or infinite.
- γ is the discount factor that discounts future rewards to obtain the present value.

In the POMDP setting, the agent can take a decision based on a formulated belief over the system's state. Such a belief is defined as a probability distribution over S , which maps the discrete finite set of states into a continuous $|S| - 1$ dimensional simplex (Papakonstantinou & Shinozuka, 2014b). It is a sufficient statistics over the complete history of actions and observations. Solving a POMDP is thus equivalent to solving a continuous state MDP defined over the belief space, termed the belief-MDP (Papakonstantinou & Shinozuka, 2014b; Andriotis et al., 2021). The belief over the system's state is updated according to Bayes' rule every time the agent receives a new observation:

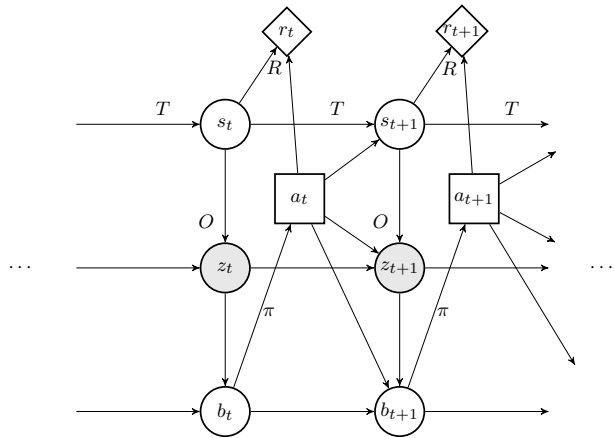
$$b(s_{t+1}) = \frac{p(z_{t+1} | s_{t+1}, a_t)}{p(z_{t+1} | \mathbf{b}, a_t)} \sum_{s_t \in S} p(s_{t+1} | s_t, a_t) b(s_t) \quad (1)$$

where the denominator is the normalizing factor:

$$p(z_{t+1} | \mathbf{b}, a_t) = \sum_{s_{t+1} \in S} p(z_{t+1} | s_{t+1}, a_t) \sum_{s_t \in S} p(s_{t+1} | s_t, a_t) b(s_t) \quad (2)$$

The objective of the POMDP is to determine the optimal policy π^* that maximizes the expected sum of rewards:

Fig. 1 Probabilistic graphical model of a POMDP



$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^H \gamma^t r_t \right] \quad (3)$$

Algorithms based on DP (Bertsekas, 2012) can be used to compute the optimal policy. These algorithms rely on two key functions: the value function V^π , which calculates the expected sum of rewards for a policy π starting from a given state until the end of the prescribed horizon, and the Q-value function Q^π (Sutton & Barto, 2018), which estimates the expected value for assuming action a_t in state s_t , and then following policy π .

Finally, a POMDP can be represented as a special case of influence diagrams (Morato et al., 2022; Luque & Straub, 2019), which form a class of probabilistic graphical models. Figure 1 illustrates the influence diagram for the POMDP considered in this work. Circles and rectangles correspond to random and decision variables, respectively, while diamonds correspond to utility functions (Koller & Friedman, 2009). Shaded shapes denote observed variables, while edges encode the dependence structure among variables.

The graphical model in Fig. 1 as well as the POMDP mathematical definitions above refer to a special POMDP case with a direct dependency among the observations, as this is the formulation used in this work to model the problem at hand and the available data (see Arcieri et al. (2023) for more details on why this direct dependency is necessary). Nevertheless, it is possible to present this special POMDP case without loss of generalization because the conditional probability $p(z_t | s_t, a_{t-1}, z_{t-1})$ simply reduces to $p(z_t | s_t, a_{t-1})$ if z_{t-1} does not provide further information directly (i.e., if there is not a direct dependency) and the standard formulation is recovered.

2.2 Belief-MDPs

By converting the POMDP problem, originally defined over the observation space, into a belief-MDP, the objective becomes to determine the optimal policy π^* that maximizes the expected sum of rewards defined over the belief space, essentially mapping beliefs to actions:

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[\sum_{t=0}^H \gamma^t R(s_t, \pi(b_t)) \right] \quad (4)$$

It is, thus, possible to rewrite the value function V^π from its original form into the new belief-based form (Papakonstantinou & Shinozuka, 2014b):

$$V^{\pi^*}(\mathbf{b}_t) = \max_{a_t \in A} \left[\sum_{s_t \in S} b(s_t) R(s_t, a_t) + \gamma \sum_{z_{t+1} \in Z} p(z_{t+1} | \mathbf{b}, a_t) V^{\pi^*}(\mathbf{b}_{t+1}) \right] \quad (5)$$

Likewise, it is possible to rewrite other RL fundamentals and re-derive popular RL solution algorithms in terms of the belief variable. Andriotis and Papakonstantinou (2019) first combine these known results from POMDP theory with deep RL solution methods, in order to solve complex partially observable problems. The observation acquired at each time-step is used to update the belief variable over all hidden states via Eq. 1, along with a (learned or known) model of the environment. The updated belief is then passed as input of classical feed-forward NNs, which learn the optimal policy via popular model-free RL algorithms, though based on the belief variable, thus avoiding to carry a history of observations at each time-step and the use of more complex networks [e.g., LSTMs or Transformers, for which the interested reader is referred to Zhu et al. (2017), Meng et al. (2021) and Parisotto et al. (2020) for a detailed overview of these broadly adopted schemes] to handle such input structures. This approach has been successfully applied in subsequent works (Andriotis & Papakonstantinou, 2021; Morato et al., 2023) to solve complex POMDP problems in the field of maintenance planning of engineered infrastructure.

The belief variable encodes information on the hidden states that is extracted from the uncertain observations, conditioned on the learned/assumed model. This extracted information is thus not required to be learned by, for instance, a NN (e.g., via LSTM cells, which infer network hidden states from observations). The use of the belief variable allows to ease the learning process and alleviate the *curse of history* in POMDP formulations. By exploiting a more informative and compact representation of the observations, this methodology is expected to lead to improvements with respect to directly applying model-free RL algorithms over the observation space, as demonstrated in Sect. 5.

2.3 Bayesian decision making for robust solution

Arcieri et al. (2023) combine Bayesian decision making with the POMDP framework to derive optimal solutions that are robust to the epistemic uncertainty over the POMDP environment parameters. In Bayesian decision theory (Berger, 2013), the optimal action is the one that maximizes the expected utility $U(\theta, a)$ with respect to the entire problem parameter distribution $p(\theta)$, namely:

$$a^* = \arg \max_{a \in A} \mathbb{E}_{\theta \sim p(\theta)} [U(\theta, a)] \quad (6)$$

Cast into the sequential decision-making scheme, the utility function U is replaced by the objective function of the problem (or the objective function of the solution algorithm in the case of approximate methods, as is the case in POMDP problems).

In Arcieri et al. (2023), this framework is devised for POMDP cases by utilizing the Q_{MDP} approximate solution method:

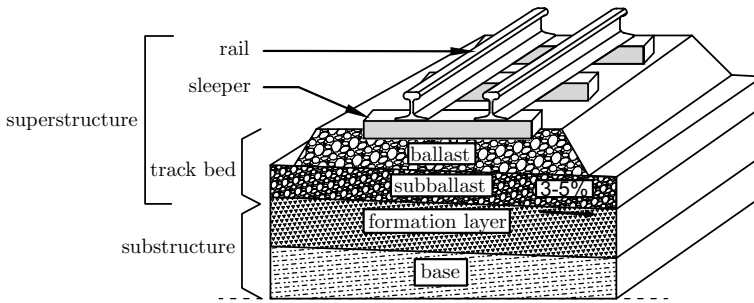


Fig. 2 Structure of the railway track. Figure reproduced from Profillidis (2016)

$$a_t^* = \arg \max_{a \in A} \mathbb{E}_{\theta \sim p(\theta)} \left[\sum_{s \in S} b_{\theta}(s_t) Q_{\theta}^{\pi^*}(s, a, t) \right] \quad (7)$$

The robust optimal action can be computed at each step by approximating the expectation with an average over (e.g., MCMC) samples. In this work, we extend this framework to deep RL approaches, which are more generally applicable to a larger variety of complex problems. To this end, we propose a domain randomization (Tobin et al., 2017) of the environment parameters over the inferred MCMC samples to train a policy that is robust over the POMDP epistemic uncertainty, as presented in Sect. 6.

3 The railway maintenance problem

We apply and test the proposed methodology on the problem of optimal maintenance planning for railway infrastructure assets on the basis of availability of regularly acquired monitoring data. The railway track comprises various components, as illustrated in Fig. 2, such as rails, sleepers, and ballast, which are exposed to harsh environments and high operating loads, leading to accelerated degradation. Among these infrastructure components, the substructure—in particular—is especially important in this degradation process. The substructure undergoes repeated loading from the superstructure (tracks, sleepers and ballast), prevents soil particles from rising into the ballast, and facilitates water drainage. A weakened substructure typically results in distortions of the track geometry. Tamping (Audley & Andrews, 2013), a maintenance procedure that uses machines to compact the ballast underneath the railway track, restoring its shape, stability and drainage system, is often applied when the substructure condition is considered moderately deteriorated. However, in case of poor substructure condition, such as intrusion of clay or mud or water clogging, tamping provides only a short-term remedy, and replacing the superstructure and substructure is the most appropriate long-term solution.

The optimization of maintenance decisions for these critical infrastructure components benefits from information that is additional to the practice of scheduled visual inspections, which are typically conducted on-site by experts. Such additional information can be delivered from monitoring data derived by diagnostic vehicles. In this work, we specifically exploit the *fractal values*, a substructure condition indicator extracted from the longitudinal level, which is measured by a laser-based system mounted on a diagnostic vehicle, to guide decisions for substructure renewal. The longitudinal level represents the deviations of the rail from

a smoothed vertical position (Wang et al., 2021). On the basis of this measurement the fractal values can be computed, via appropriate filtering and processing steps. The fractal value indicator describes the degree of “roughness” of the track at varying wavelength scales. For the interested reader, the detailed steps of the fractal value computation are reported in Landgraf and Hansmann (2019) and Arcieri et al. (2023). In particular, long-wave (25–70 m) fractal values, which are employed in this work, have shown a significant correlation to substructure damage (Hoelzl et al., 2021), and are used by railway authorities as an indicator which can instigate repair/maintenance actions, such as tamping.

In this work, we use actual track geometry measurements, carried out via a diagnostic vehicle of the SBB between 2008 and 2018 across Switzerland’s railway network. The track geometry measurements were collected twice a year for the investigated portion of track. The fractal values are computed every 2.5 m from the measured longitudinal level. The performed maintenance actions have been logged for the analyzed tracks over the same considered period. These logs contain information on the maintenance, repair, or renewal actions taken on a section of the network at a specific date.

We model the railway track maintenance optimization with a POMDP scheme, relying on diagnostic vehicle measurements of long-wave fractal values. The true but unobserved railway condition is discretized in 4 hidden states, s_0 , s_1 , s_2 , and s_3 , reflecting various grades, from perfect to highly deteriorated state. This is chosen to coincide with the number of grade levels assumed by the Swiss Federal Railways for classifying substructure condition. It should be noted, that in the POMDP inference setting, the number of hidden states is not fixed. To this end, we evaluated further possible dimensions of the hidden states vector, as part of the POMDP inference presented in the next section, and a dimension of four yielded improved convergence and better-defined distributions. The fractal values are assumed as the (uncertain) POMDP observations, which correlate with the actual state of the substructure, but offer only partial and noisy information thereof. Unlike classical POMDP modeling of optimal maintenance planning problems, where observations are usually discrete, fractal values comprise (negative) continuous values, rendering the considered POMDP inference and solution quite complex. The problem definition is supplemented with information on the available maintenance actions. Three possible actions are considered, corresponding to the real-world setting, namely action a_0 do-nothing, and the aforementioned tamping and replacement actions, denoted as a_1 and a_2 , which can be interpreted as a minor and a major repair, respectively. The fractal value indicators are derived via measurements of the diagnostic vehicle every 6 months, which thus represents the time-step of the decision-making problem. Considering the almost 10 years of collected measurements, our real-world dataset is ultimately composed of time-series of 20 fractal values, per considered railway section, complete with information on respective maintenance actions (with “action” do-nothing included), i.e., $(z_0, a_0, \dots, a_{19}, z_{20})$. Finally, the (negative) rewards representing costs associated with actions and states have been elicited from SBB and are reported in Table 1 in general cost units.

4 POMDP inference

To formulate the POMDP problem, the transition dynamics and the observation generating process must be inferred. In the RL context, the POMDP inference is necessary to generate samples for the policy learning, for inference of a belief over the hidden states, and/or for testing purposes. To tackle this key issue, we propose an MCMC inference of a HMM conditioned on actions, which jointly estimates parameter distributions of both

Table 1 Costs of the POMDP model

State condition	s_0	s_1	s_2	s_3
Maintenance action				
a_0	0	0	0	0
a_1	-50	-50	-50	-50
a_2	-2050	-2710	-3370	-4050
Condition cost	-100	-200	-1000	-8000

the POMDP transition and observation models based on available data. While we implement the proposed scheme on the problem of railway maintenance planning based on fractal value observations, its applicability is general. Therefore, we further suggest possible extensions to help researchers and practitioners tailor the POMDP model inference to the problem at hand. In addition, we provide a complementary tutorial¹ illustrating the code implementation on various simulated case-studies, in order to support exploitation for real-world applications.

In the context of discrete hidden states and actions, the transition dynamics are modeled via Dirichlet distributions:

$$\begin{aligned}
 T_0 &\sim \text{Dirichlet}(\alpha_0) \\
 s_0 &\sim \text{Categorical}(T_0) \\
 T &\sim \text{Dirichlet}(\alpha_T) \\
 s_t \mid s_{t-1}, a_{t-1} &\sim \text{Categorical}(T)
 \end{aligned} \tag{8}$$

where T_0 are the parameters of the probability distribution of the initial state s_0 , and α_0 and α_T are the prior concentration parameters. T_0 can be assigned a uniform flat prior α_0 , unless some prior knowledge on the initial state distribution is available. By contrast, it is beneficial to regularize T with informative priors α_T , which regularize the deterioration or the repairing process. For example, the transition matrix related to the action do-nothing, which describes the deterioration process of the system, can be regularized with higher prior probabilities on the diagonal and on the upper-right triangle, and near-zero on the lower-left triangle. Likewise, the transition matrices associated with maintenance actions would present higher prior probabilities on the left triangle and near-zero on the right triangle, in order to inform the model that a repair action is expected to be followed by improvements of the system.

The dimensionality of the Dirichlet distribution that models the transition dynamics T is $S \times S \times A$, namely one transition matrix per action. The extension to time-dependent transition dynamics is straightforward by enlarging the distribution by a further dimension representing time, i.e., $S \times S \times A \times H$.

In the context of continuous observations, the observation generating process can differ on the basis of whether the observation follows a deterioration or a repairing process. In addition, similarly to the inference of the first hidden state according to T_0 , an initial observation process can be necessary to model the first observation. Tailoring to the nature of the fractal value monitoring data, the initial, deterioration, and repairing processes are modeled via Truncated Student's t processes, as follows:

¹ Code available on [GitHub](#).

$$\begin{aligned}
z_0 &\sim \text{TruncatedStudentT}(\mu_{s_0}, \sigma_{s_0}, \nu_{s_0}, \text{ub} = 0) \\
z_t - z_{t-1} &\sim \text{TruncatedStudentT}(\mu_{d|s_t}, \sigma_{d|s_t}, \nu_{d|s_t}, \text{ub} = -z_{t-1}) \\
z_t &\sim \text{TruncatedStudentT}(k_{r|a_{t-1}} * z_{t-1} + \mu_{r|s_t}, \sigma_{r|s_t}, \nu_{r|s_t}, \text{ub} = 0)
\end{aligned} \tag{9}$$

where *ub* stands for “upper bound”, and all parameters governing the processes are assigned priors described in Arcieri et al. (2023).

The use of Truncated Student’s *t* processes was tailored to the mathematical characteristics of the fractal values, which (1) assume only negative values, (2) exhibit a negative trend in absence of repairing actions, (3) their values are dependent on the previous observations, and (4) the studied dataset, as is common in real-world measurements, presents outliers and measurement errors, modeled by the Student’s *t* fat tails. Naturally, other distributions can also be employed as part of the proposed framework in order to model the data at hand related to each application. For instance, in absence of the previous limiting characteristics, simpler (unbounded) Gaussian emissions could have been used, as further shown in the tutorial. In the case of discrete observations, the observation model would be represented by a probability matrix $S \times Z$, which can be again modeled via a Dirichlet distribution. In the case of more than one possible inspection action or monitoring tool, as in Papakonstantinou et al. (2018), the Dirichlet distribution can be simply enlarged by a further dimension representing the number of possibilities. Finally, dependencies in multi-component systems could be modeled via a Bayesian hierarchical model (Gelman et al., 1995), enabling solutions as proposed in Andriotis and Papakonstantinou (2019, 2021) and Morato et al. (2023).

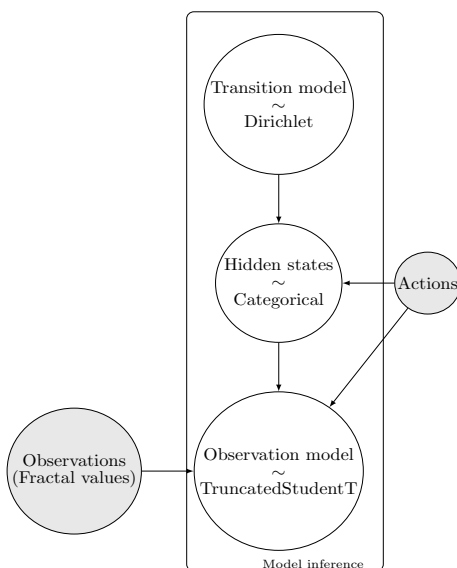
The graphical model of the entire HMM is reported in Fig. 3. The MCMC inference is run on a final dataset of 62 time-series with the No-U-Turn Sampler (NUTS) (Hoffman & Gelman, 2014). Four chains are run with 3000 samples collected per chain. The inference results, which present good post-inference diagnostic statistics, with no divergences and high homogeneity between and within chains, are reported in Figs. 9, 10, 11, 12, 13 and 14 in Appendix A.

5 RL for POMDP solution

POMDP problems have been tackled via deep RL with common methods augmented with LSTM architectures and a history of past observations (and possibly actions) as inputs (Zhu et al., 2017; Meng et al., 2021). More recently, motivated by the breakthrough success of Transformers over LSTMs in natural language processing, Parisotto et al. (2020) designed a new Transformer architecture, namely GTrXL, which yielded significant improvements in terms of performance and robustness over LSTMs on a set of partially observable benchmarking tasks.

Both LSTM and GTrXL architectures compose fully model-free deep RL solutions to POMDPs. A third modeling option, which comprises a model-based/model-free hybrid solution, pertains to transformation of the POMDP problem into the belief-MDP by computing beliefs via Bayes Theorem (Eq. 1). The belief-MDP is then solved via classical deep model-free RL methods with feed-forward NNs (Andriotis & Papakonstantinou, 2019; Morato et al., 2023). We here compare the performance of the two model-free and the hybrid solution, referred to as “belief-input” case, on the real-world POMDP problem of railway maintenance planning that has been presented in Sect. 3, with parameter inference

Fig. 3 A graphical model of the HMM inference. Arrows indicate dependencies, while shaded nodes indicate observed variables



described in Sect. 4. While Parisotto et al. (2020) demonstrate the superiority of Transformers over LSTMs on simulated tasks, our work offers a further comparison of the two methods, and confirms the superiority of the former, on a real-world, stochastic (both in the transition dynamics and in the observation generating process), partially observable problem.

For this comparison we set the POMDP parameters to the mean values of the distributions reported in Appendix A, in order to evaluate the methods without model uncertainty, with the latter case tackled in the next section. For all modeling options, the policy is learned via the Proximal Policy Optimization (PPO) algorithm with clipped surrogate objective (Schulman et al., 2017). The overall evaluation algorithm is reported in pseudocode format in Algorithm 1. In addition, the code of the experiment is made available online². We consider 50 time-steps, i.e., 25 years (1 time-step equals 6 months), as the decision horizon H of the problem, as discussed with our SBB partners.

² Code available on [GitHub](#).

Algorithm 1 Evaluation algorithm

```

1: Initialize policy network  $\pi_\phi$ 
2: Initialize replay buffer  $\mathcal{D} \leftarrow \emptyset$ 
3: Set environment parameters  $\hat{\theta}$  of  $T_0, O_0, T, O$  to the mean values of  $p(\theta \mid D)$ 
4: for training episode = 0 to  $N$  do
5:   Sample initial  $s_0 \sim T_{0_{\hat{\theta}}}$  and  $z_0 \sim O_{0_{\hat{\theta}}}$ 
6:   Initialize belief to initial state distribution  $b_0 \leftarrow T_{0_{\hat{\theta}}}$ 
7:   for timestep  $t = 0$  to  $H$  do
8:     if belief-input case then
9:       Input  $y_t = b_t$ 
10:    else if LSTM then
11:      Input  $y_t = (z_t, a_{t-1}, \dots, z_{t-h+1})$   $\triangleright h = 3$ 
12:    else if GTrXL then
13:      Input  $y_t = (z_t, a_{t-1}, \dots, z_0)$ 
14:    end if
15:     $a_t \sim \pi_\phi(y_t)$ 
16:     $s_{t+1} \sim T_{\hat{\theta}}(s_t, a_t), z_{t+1} \sim O_{\hat{\theta}}(s_{t+1}, a_t, z_t)$ 
17:    Compute  $b_{t+1}$  via Equation 1
18:     $\mathcal{D} \leftarrow \mathcal{D} \cup \{(y_t, a_t, R(s_t, a_t))\}$ 
19:  end for
20:  every  $K$  total timesteps do  $\triangleright K = 4,000$ 
21:    Update  $\pi_\phi$  with PPO and replay buffer  $\mathcal{D}$ 
22:  every 5 updates do
23:    Run 500 policy evaluation episodes without exploration
24: end for

```

For all methods, the policy networks are updated every 4000 training time-steps. Every 5 updates, 500 evaluation episodes are run with different random seeds in order to average the results over the stochasticity of the environment. In addition, the entire analysis is repeated over 10 different random seeds to further average the results over the stochasticity of the NN training. Grid-searches are performed over the hyperparameters for all methods and the selected values are reported in Table 4 in Appendix B. The mean performance over 250 evaluation iterations (5 million training time-steps) is plotted in Fig. 4 along with the shaded regions representing one standard deviation over the 10 different random seeds. Along with the three evaluated methods, three additional benchmarking solutions are reported. The first option refers to the Q_{MDP} method (Littman et al., 1995), which constitutes a POMDP solution based on DP, and which turns out to be an effective solution for the characteristics of this problem (Arcieri et al., 2023). The second option is the optimal MDP solution, namely the optimal policy computed and evaluated on the underlying MDP, i.e., when the hidden states are fully observable. The latter constitutes an upper bound to any POMDP solution, which cannot be exceeded, given the irreducible inherent uncertainty of the observations, and only serves as a benchmarking reference. Finally, the total cost achieved with the current maintenance policy implemented by the Swiss Federal Railways is also reported (dashed black line). This policy is based on optimized thresholds on the fractal values to guide tamping and renewal actions in real life. The costs of all three

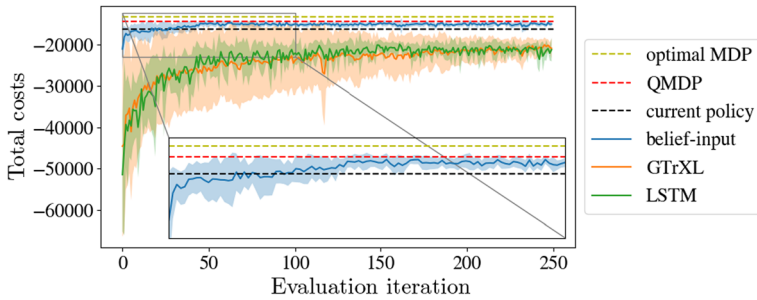


Fig. 4 Comparison of the performance of LSTM (green), GTrXL (orange), and the belief-input case (blue) over 250 evaluation iterations. At every iteration, 500 trial episodes are evaluated with different random seeds and the average results are returned. The entire training is repeated over 10 different random seeds. The plotted learning curves denote the mean performance, while the shaded regions represent one standard deviation over the 10 different random seeds. An evaluation iteration is run after 5 policy updates and a policy update is performed every 4000 training time-steps, for a total of 5 million time-steps. The performance is further benchmarked against the Q_{MDP} method (dashed red), the optimal MDP policy (dashed yellow), and the current real-life policy implemented by the Swiss Federal Railways (dashed black). On the right corner, a zoomed-in plot of the belief-input performance over the first 100 evaluation iterations is shown (Color figure online)

reference policies (Q_{MDP} , optimal MDP, current policy) are accurately estimated by averaging 100,000 simulations.

The belief-input method outperforms the other two model-free RL solutions and already shows strong performance at the first evaluation iterations. The method matches the performance of the current real-life implemented policy in about 25 evaluation iterations (500,000 training time-steps), outperforming it and converging to the best policy in about 50 evaluation iterations (1 million training time-steps), as shown in the zoomed-in view of the first 100 evaluation iterations reported in the lower right figure inset, converging close to the performance of the Q_{MDP} method. Because the number of training time-steps evaluated may not be sufficient for convergence of the other two model-free RL methods, we continue training up to 2000 evaluation iterations (40 million training time-steps). This could however negatively impact the performance of the belief-input method, which already converged and may begin to suffer from overfitting. The extended training is reported in Fig. 5, where a rolling average window of 5 steps is further applied for illustration purposes.

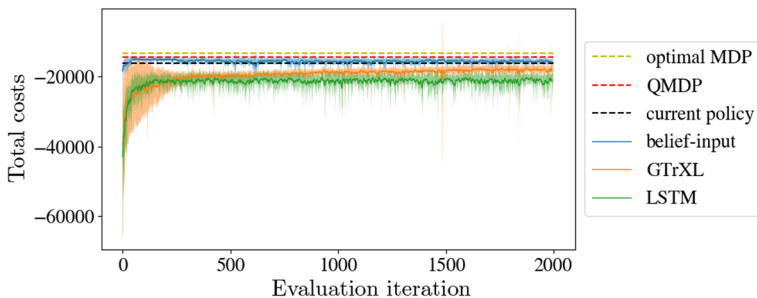


Fig. 5 Comparison of the performance of LSTM (green), GTrXL (orange), and the belief-input case (blue) over 2000 evaluation iterations, for a total of 40 million training time-steps. The performance is further plotted with an average rolling window of 5 steps for displaying purposes

Table 2 Performance of the best models, as inferred during the training process, evaluated over 100,000 simulations

Method	Avg. costs	SE	Max	Min
Optimal MDP	−13,315	27	−5000	−93,980
Q_{MDP}	−14,374	35	−5050	−123,800
Belief-input	−14,677	36	−5050	−121,950
Current policy	−16,295	42	−5000	−161,650
GTrXL	−17,196	46	−5700	−188,600
LSTM	−18,167	42	−5100	−404,150

As expected, the performance of the belief-input method slightly decreases over time, yet it still stays above the current real-life policy. The GTrXL is proven to deliver a better architecture than the LSTM for POMDP applications, also for this particular case of application on a real-world problem. The GTrXL, after the first iterations, is indeed less affected by variance and eventually converges to a better policy, albeit still far from current real-life policy and the best policy with the belief-input method.

Finally, for all three methods we saved the best models, as determined during training, and evaluated the learned policies over 100,000 trials. The results are reported in Table 2 in terms of average performance (i.e., average total costs), Standard Error (SE), best (Max) and worst (Min) trial. In the table, the belief-input case average performance is close but slightly worse than the Q_{MDP} method. This is likely due to the fact that the best model was picked based on an average over 500 trials, which is still subject to a significant standard error. While we explained in Sect. 2.2 why the belief-input case is expected to deliver improved performance with respect to the two alternate schemes, we are not aware of other works that experimentally assess its superiority in solving POMDP problems against state-of-the-art deep RL methods (LSTMs and Transformers) that operate directly on the observation space. In addition, the belief-input method is able to improve the current real-life policy by yielding significantly reduced costs.

6 Domain randomization for robust solution

Further to the challenge of POMDP inference, another key issue is the robustness of the deep RL solutions. RL methods generally learn an optimal policy by interacting with a simulator. When the trained RL agent is deployed to the real-world, the performance can deteriorate, or altogether fail, due to the “simulation-to-reality” gap (Zhao et al., 2020; Salvato et al., 2021), if the solution is not robust to model uncertainty.

In Arcieri et al. (2023), we propose a framework in combination with the POMDP inference to enhance the robustness of DP solutions to model uncertainty. Namely, the POMDP parameter distributions inferred via MCMC sampling are incorporated into the solution by merging DP algorithms with Bayesian decision making, as mentioned in Sect. 2.3. In Arcieri et al. (2023) we incorporate DP methods into the Bayesian optimal action formula (Eqs. 6–7 in this work) to derive solutions that maximize the expected value with respect to the entire model parameter distributions, hence rendering the solution robust to model uncertainty.

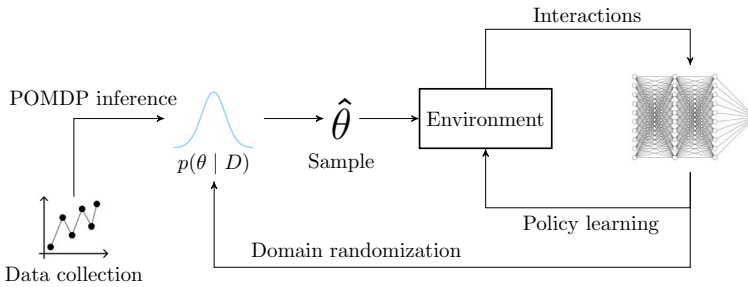


Fig. 6 The POMDP inference and robust solution framework via domain randomization and deep reinforcement learning

In this work, we bring and extend this framework into the RL training scheme. The utility function is represented by the RL algorithm objective function, e.g., the PPO clipped surrogate objective in this case. We propose the use of domain randomization (Tobin et al., 2017) of the POMDP environment, which is enabled by our POMDP inference scheme through the recovery of parameter distributions, in order to enhance the robustness of the RL solution to model uncertainty. At every episode, a different POMDP configuration is sampled from the parameter distributions. The RL agent interacts with this POMDP configuration until the end of the episode. Afterwards, a new configuration of the environment is sampled. At the end of the training, the RL agent will have optimized the learned policy over all possible problem parameters to derive a solution robust to model uncertainty. The expectation in Eq. 6 is thus implemented in practice via stochastic gradient ascent/descent steps over varying randomized problem parameters. It should be reminded that the (Bayesian) robust optimal policy may be sub-optimal for a specific value θ , while maximizing the expected value with respect to the entire model parameter distribution. The domain randomization technique can thus be used in combination with the model inference proposed in Sect. 4 to establish a joint framework of POMDP inference and robust solution based on RL. The framework is depicted in the graphical model in Fig. 6.

We showcase the implementation of this framework with the belief-input method, but it is also applicable with the other methods reported in Table 2, given its general validity. The evaluation algorithm is similar to Algorithm 1, with the only difference that the POMDP parameters $\hat{\theta}$ are sampled at every episode from the inferred posterior distributions $p(\theta | D)$. The policy updates are again performed every 4000 training time-steps and an evaluation iteration is run every 5 policy updates. Similarly to Fig. 5, the performance during training is averaged at each evaluation iteration over 500 episodes with different random seeds. The training is then repeated 10 times with 10 different random seeds to also average over the stochasticity of the NN training. The resulting average performance is plotted in Fig. 7. Given the more challenging learning task, owing to model uncertainty, the average training performance decreases and demonstrates a higher variance than the belief-input performance without domain randomization, shown in Fig. 5. For this case, the hyper-parameter tuning was also restricted to a minimal grid-search. While the results are already satisfying, the RL agent performance can likely be further increased via a more thorough hyperparameter optimization.

Again, the best performing models shown in the evaluations during training are saved and the learned policy is evaluated over 100,000 simulations. The results are shown in Table 3 and compared against the robust Q_{MDP} policy described in detail in Arcieri et al.

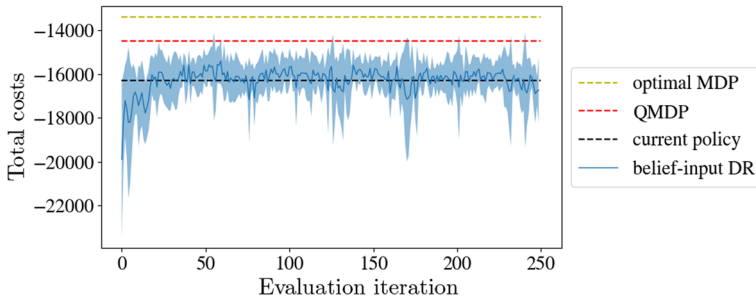


Fig. 7 Performance of the belief-input method (blue) over 250 evaluation iterations with domain randomization, i.e., a different POMDP model is sampled at every episode, both for training and evaluation. At every iteration, 500 trial episodes are evaluated with different random seeds and the average results are returned. The entire training is repeated over 10 different random seeds. The plotted learning curve denotes the mean performance, while the shaded regions represent one standard deviation over the 10 different random seed. An evaluation iteration is run after 5 policy updates and a policy update is performed every 4000 training time-steps, for a total of 5 million time-steps. The performance is further benchmarked against the robust Q_{MDP} method (dashed red), the robust optimal MDP policy (dashed yellow), and the current real-life policy implemented by the Swiss Federal Railways (dashed black), all evaluated under model uncertainty, as in Arcieri et al. (2023) (Color figure online)

(2023) and the upper bound optimal MDP policy evaluated with full observability, both assessed under model uncertainty. In addition, we report the result of the best model of the RL agent from the previous analysis, namely with the policy optimized without model uncertainty incorporated into the training (i.e., no domain randomization), evaluated now in the context of model uncertainty. This further analysis resembles a real-world deployment, where the environment parameters can differ from those inferred, inducing the aforementioned simulation-to-reality gap. The performance of the agent trained with no domain randomization deteriorates, while the agent trained with domain randomization is able to learn and deliver a more robust policy in the context of model uncertainty. We also report the results of the current real-life policy evaluated under model uncertainty and demonstrate that the policy learned by the belief-input RL agent is able to significantly improve the current policy by yielding substantially reduced costs also in this further context.

Finally, Fig. 8 shows two trials of the sequential maintenance actions planned by the belief-input model, which has been trained with domain randomization, over the considered problem horizon. The environment true states are reported in the second subplot from the top, which however are never accessed by the RL agent and are here reported only for comparison and interpretation. The hidden states indeed produce the observations (fractal

Table 3 Performance of the best models, as inferred during training, evaluated over 100,000 simulations in the context of model uncertainty with domain randomization. In particular, we report on the evaluation of the belief-input agent trained with (DR) and without Domain Randomization (no DR). The former achieves a significantly improved and more robust policy

Method	Avg. costs	SE	Max	Min
Optimal MDP	-13,374	33	-5000	-190,450
Q_{MDP}	-14,526	39	-5050	-197,050
Belief-input DR	-14,648	38	-5050	-168,600
Belief-input no DR	-14,901	39	-5050	-205,100
Current policy	-16,259	44	-5000	-201,850

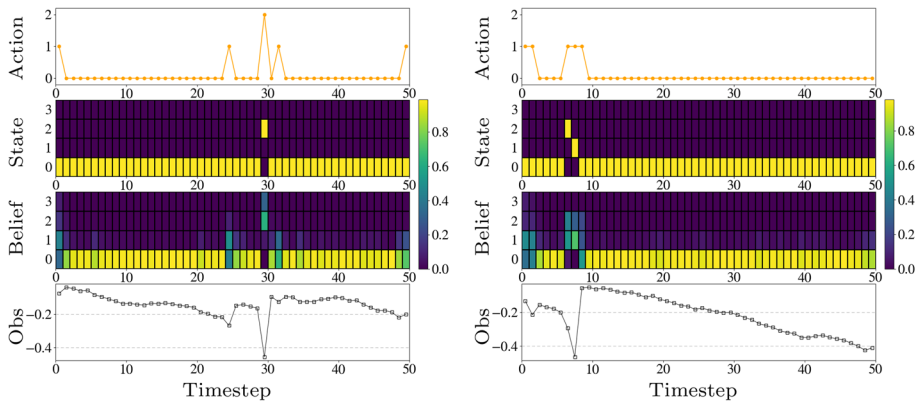


Fig. 8 Two trials of the maintenance actions planned by the belief-input model trained with domain randomization. From bottom to top: the observations (fractal values); the beliefs, namely a probability distribution over hidden states, computed via Bayes' formula and fed to the policy networks; the true hidden states, which are not accessed by the agent and/or the model; the actions planned by the RL agent

values), reported in the bottom subplot. These are used to compute the beliefs (third subplot) via Bayes' formula, which are fed to the policy networks. Based on these beliefs, the RL agent plans the maintenance actions, reported in the top subplot. While some higher uncertainty is present in the formed belief in some specific time-steps of the trials (e.g. time-step 25 in the left plot and time-step 9 in the right plot), which lead to non-optimal maintenance actions, these are explained by the observations that generated these beliefs, which indeed allow outliers/measurement errors in the simulations. Besides these isolated cases, it is possible to appreciate how the beliefs are generally accurate compared to the true hidden states, although these are never accessed for their computation, and effectively lead to optimal actions. To explain the plots further, one can notice, for example, how the maintenance action a_2 at time-step 30 on the left plot significantly improves the ground truth hidden state and also the inferred belief, yielding a substantially increase in the observation as well. Likewise for the maintenance actions at time-steps 7–9 in the right plot.

7 Conclusion

This work tackles two key issues related to adoption of RL in real-world partially observable planning problems. First, an environment (POMDP) model, which enables the RL training via simulations, is often unknown and generally non-trivial to infer, with unified best practices not available in the literature. This constitutes a main obstacle against broad adoption of the POMDP scheme and its solution methods for real-world applications. Second, RL solutions often lack robustness to model uncertainty and suffer from the simulation-to-reality gap.

In this work, we tackle both issues via a combined framework for inference and robust solution of POMDPs based on deep RL algorithms. The POMDP inference is carried out via MCMC sampling of a HMM conditioned on actions, which jointly estimates the full

distributions of plausible values of the transition and observation model parameters. Then, the parameter distributions are incorporated into the solution via domain randomization of the environment, enabling the RL agent to learn a policy, which is optimized over the space of plausible problem parameters and is, thus, robust to model uncertainty. We compare three common RL modeling options, namely a Transformer and an LSTM-based approach, which constitute model-free RL solutions, and a hybrid belief-input case. We implement our methods for optimal maintenance planning of railway tracks based on real-world monitoring data. While the Transformer delivers generally better performance than the LSTM, both methods are significantly outperformed by the hybrid belief-input case. In addition, the latter method outperforms the current real-life policy implemented by the Swiss Federal Railways by yielding significantly reduced costs. Ultimately, we demonstrate on the belief-input method that an RL agent trained with domain randomization is able to learn an improved policy, which is robust to model uncertainty, than an RL agent trained without domain randomization.

A possible limitation of this work is that, while our methods allow for incorporation of rather complex extensions, e.g., time-dependent dynamics and hierarchical components, and are here demonstrated on the quite difficult case of continuous observations, the POMDP inference under continuous multi-dimensional states and actions is still to be investigated. Future work will focus on the development of methods that can scale to these cases, e.g., via coupling with deep model-based RL methods (Arcieri et al., 2021).

Appendix A: Inference results

Transition model parameters

See Figs. 9, 10, and 11.

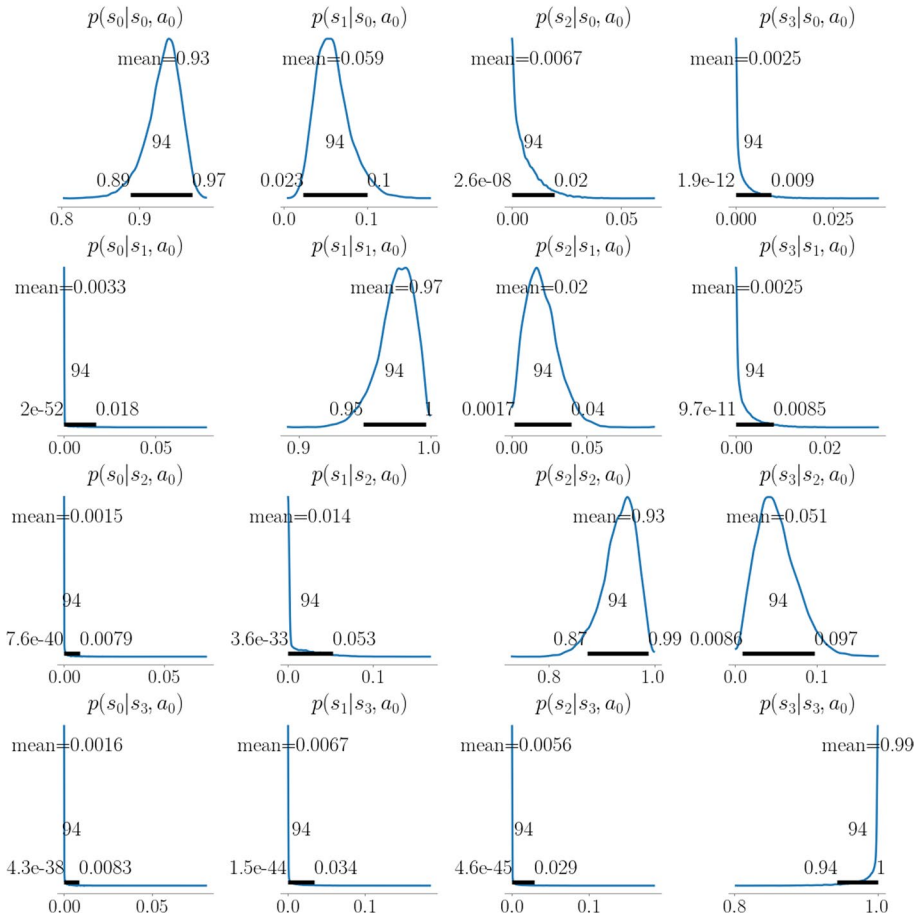


Fig. 9 Transition matrix related to action do-nothing a_0 . The distribution at row i and column j is associated with the probability to transition from state i to j when action a_0 is taken. Consistent with what is expected in deterioration processes the highest probabilities are assigned to the state remaining invariant (diagonal entries), lower probabilities exist for deterioration transitions (upper right triangle), and almost zero probability is assigned to improvements of the system (lower left triangle)

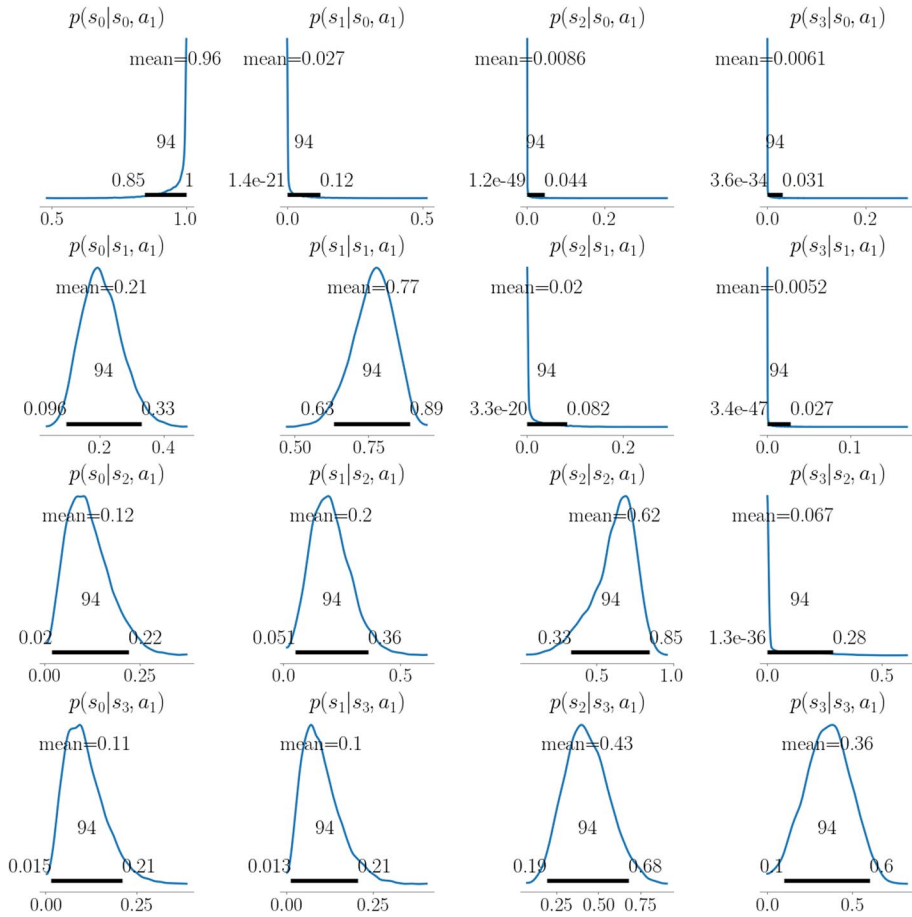


Fig. 10 Transition matrix related to action a_1 (tamping). The distribution at row i and column j is associated with the probability to transition from state i to j when action a_1 is taken. Deterioration of the system (upper right triangle) reflects an almost zero probability, while it appears most probable to remain in the same condition or improve by a maximum of one state, which reflects the reduced influence of this action

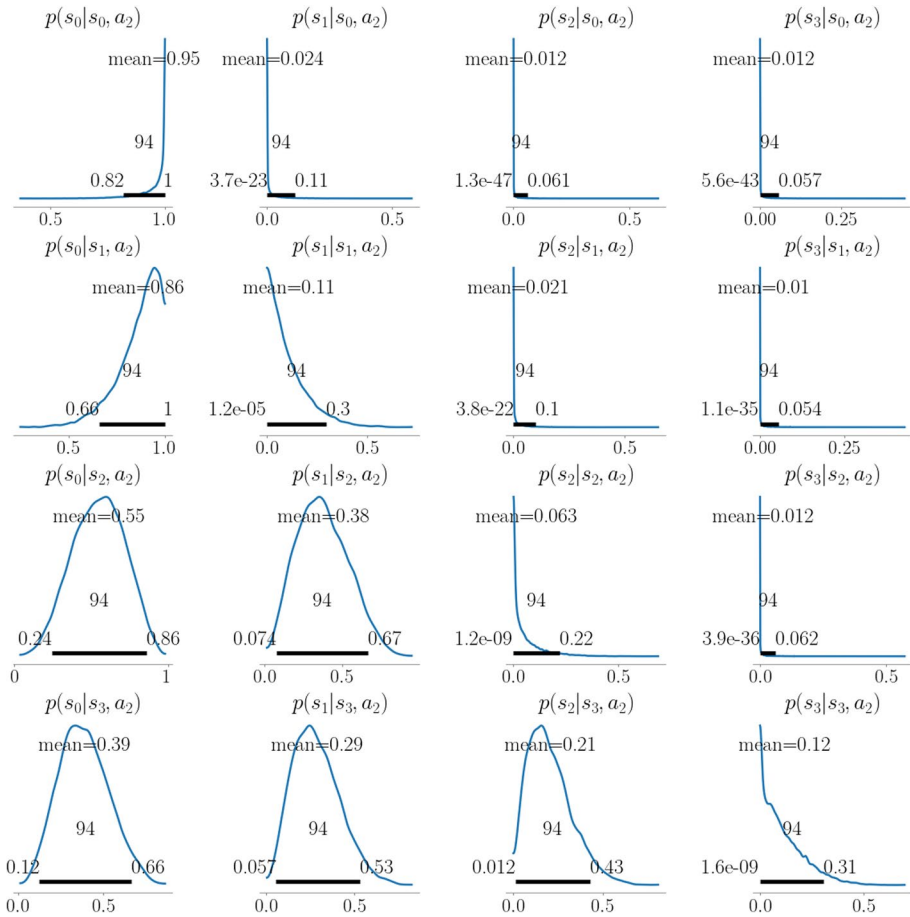
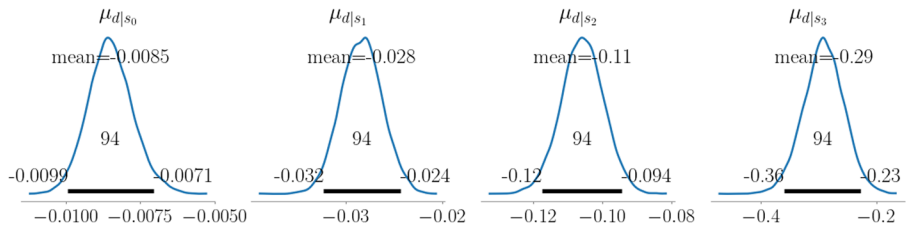


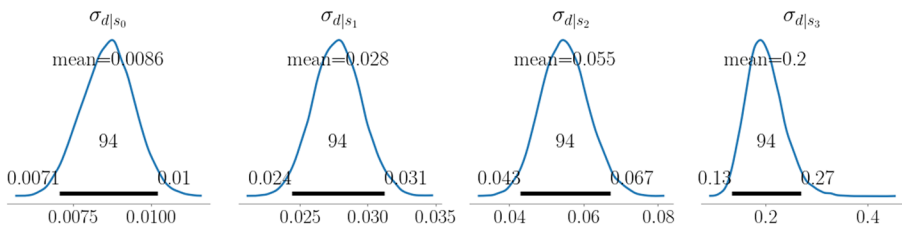
Fig. 11 Transition matrix related to action a_2 (renewal plus tamping). The distribution at row i and column j is associated with the probability to transition from state i to j when action a_2 is taken. Transition to the best possible state s_0 is consistently assigned the highest probability, regardless of the starting state, reflecting the higher repairing effect of this maintenance action

Observation model parameters

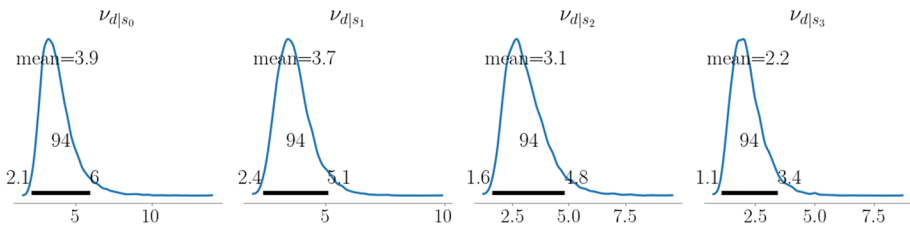
See Figs. 12, 13 and 14.



(a) Posterior distributions of state-dependent parameters $\mu_d|s_t$.

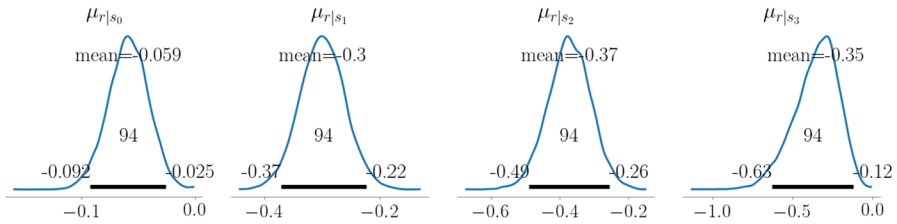
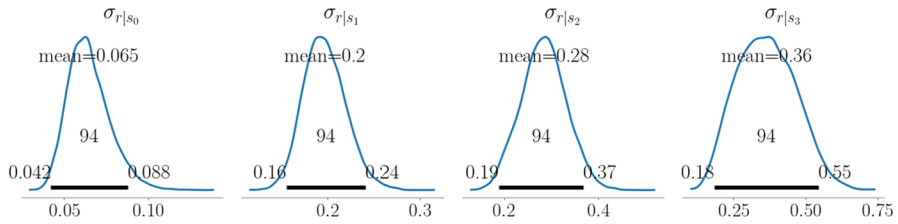
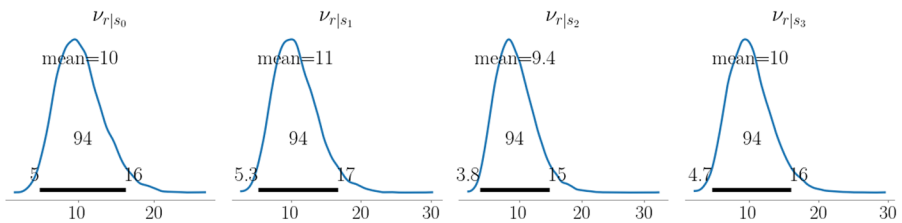
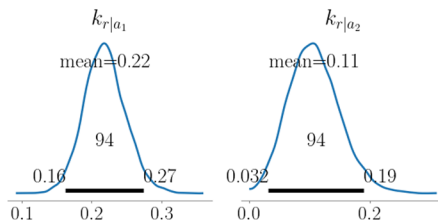


(b) Posterior distributions of state-dependent parameters $\sigma_d|s_t$.



(c) Posterior distributions of state-dependent parameters $\nu_d|s_t$.

Fig. 12 Posterior distributions of observation model parameters (deterioration process)

(a) Posterior distributions of state-dependent parameters $\mu_r|s_t$ (b) Posterior distributions of state-dependent parameters $\sigma_r|s_t$.(c) Posterior distributions of state-dependent parameters $\nu_r|s_t$.(d) Posterior distributions of the autoregressive parameters $k_r|a_t$ for a_1 (left) and a_2 (right).**Fig. 13** Posterior distributions of observation model parameters (repair process)

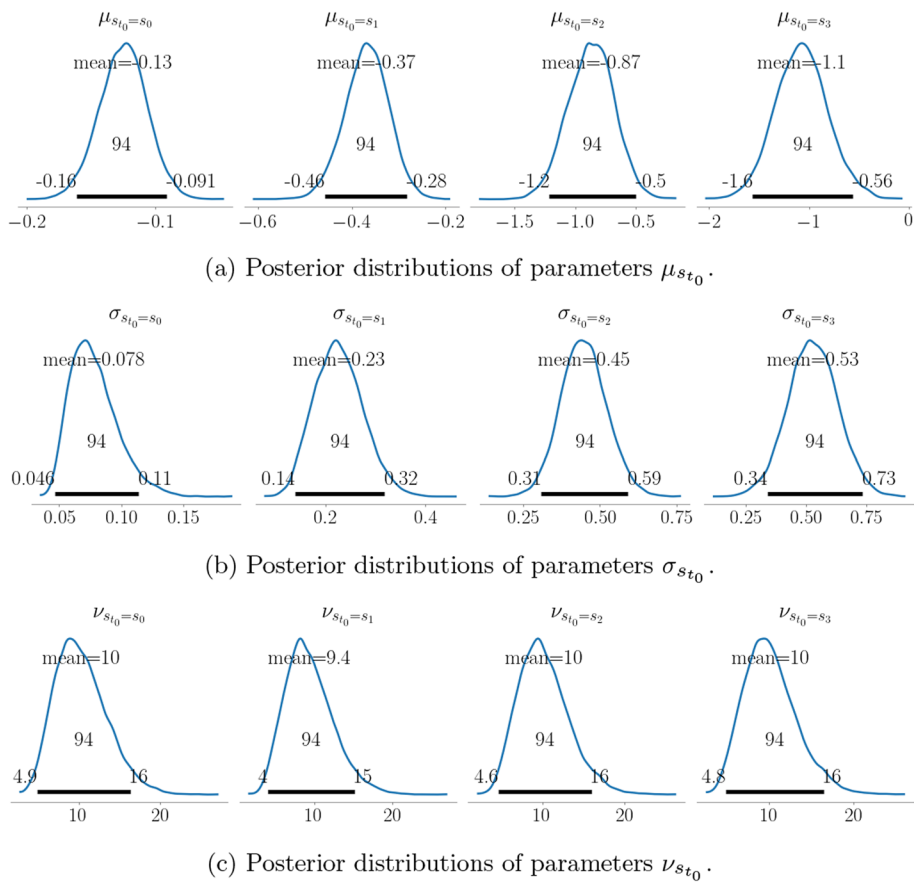


Fig. 14 Posterior distributions of observation model parameters (initial observation)

Appendix B: Hyperparameters

See Table 4.

Table 4 Best hyperparameters from the grid-search optimization

Hyperparameter	Belief (no DR)	Belief (DR)	GTrXL	LSTM
Hidden layers	3	3	2 × GTrXL	1 × LSTM + 2 × MLP
Hidden size	100	100	–	100
Learning rate	0.0001	0.0001	0.001	0.001
Heads	–	–	8	–
Head dimension	–	–	32	–
Max seq. length	–	–	50	3
Memory	–	–	50	–
Use prev. actions	–	–	Yes	Yes
Clip parameter	0.01	0.01	0.3	0.3

Acknowledgements The authors acknowledge the support of the Swiss Federal Railways (SBB) as part of the ETH Mobility Initiative project REASSESS. The authors thank the ETH cluster support for their precious help with the availability of computational power. Dr. Papakonstantinou would like to acknowledge the support by the U.S. National Science Foundation under Grant No. 1751941.

Author Contributions GA Conceptualization; Data curation; Formal analysis; Investigation; Methodology; Software; Visualization; Roles/Writing—original draft; Writing—review and editing. CH Data curation; Roles/Writing—original draft. OS Funding acquisition; Validation. DS Methodology; Supervision; Validation; Writing—review and editing. KGP Methodology; Supervision; Validation; Writing—review and editing. EC Conceptualization; Methodology; Funding acquisition; Project administration; Resources; Supervision; Validation; Writing—review and editing.

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich. The authors acknowledge the support of the Swiss Federal Railways (SBB) as part of the ETH Mobility Initiative project REASSESS.

Availability of data and material The real-world monitoring data used in this research paper is SBB proprietary and cannot be published.

Code availability All code of the experiments of this research paper is made available on GitHub in public repositories linked in the paper.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable. No further consent is needed for publication of this research paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Andriotis, C. P., & Papakonstantinou, K. G. (2019). Managing engineering systems with large state and action spaces through deep reinforcement learning. *Reliability Engineering and System Safety*, 191, 106483.
- Andriotis, C. P., & Papakonstantinou, K. G. (2021). Deep reinforcement learning driven inspection and maintenance planning under incomplete information and constraints. *Reliability Engineering and System Safety*, 212, 107551.
- Andriotis, C. P., Papakonstantinou, K. G., & Chatzi, E. N. (2021). Value of structural health information in partially observable stochastic environments. *Structural Safety*, 93, 102072.
- Arcieri, G., Hoelzl, C., Schwery, O., Straub, D., Papakonstantinou, K. G., & Chatzi, E. (2023). Bridging POMDPs and Bayesian decision making for robust maintenance planning under model uncertainty: An application to railway systems. *Reliability Engineering and System Safety*, 109496.
- Arcieri, G., Wölfe, D., & Chatzi, E. (2021). Which model to trust: Assessing the influence of models on the performance of reinforcement learning algorithms for continuous control tasks. [arXiv:2110.13079](https://arxiv.org/abs/2110.13079)

- Audley, M., & Andrews, J. D. (2013). The effects of tamping on railway track geometry degradation. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 227.
- Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis*. Berlin: Springer.
- Bertsekas, D. (2012). *Dynamic programming and optimal control* (Vol. 1). New York: Athena scientific.
- Cassandra, A. R. (1998). A survey of POMDP applications. Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes (Vol. 1724).
- Drake, A. W. (1962). Observation of a Markov process through a noisy channel (Doctoral dissertation, Massachusetts Institute of Technology). <https://dspace.mit.edu/bitstream/handle/1721.1/11341/33167429-MIT.pdf?sequence=2>
- Dung, L. T., Komeda, T., & Takagi, M. (2008). Reinforcement learning for POMDP using state classification. *Applied Artificial Intelligence*, 22(7–8), 761–779.
- Ellis, H., Jiang, M., & Corotis, R. B. (1995). Inspection, maintenance, and repair with partial observability. *Journal of Infrastructure Systems*, 1(2), 92–99.
- Farrar, C. R., & Worden, K. (2012). *Structural health monitoring: A machine learning perspective*. New York: Wiley.
- Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (1995). *Bayesian data analysis*. New York: Chapman and Hall/CRC.
- Guo, C., & Liang, Z. (2022). A predictive Markov decision process for optimizing inspection and maintenance strategies of partially observable multi-state systems. *Reliability Engineering and System Safety*, 226, 108683.
- Hoelzl, C., Dertimanis, V., Chatzi, E. N., Winklehner, D., Züger, S., & Oprandi, A. (2021). Data driven condition assessment of railway infrastructure. In *Bridge maintenance, safety, management, life-cycle sustainability and innovations* (pp. 3251–3259). CRC Press.
- Hoffman, M. D., Gelman, A., et al. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(1), 1593–1623.
- Kıvanç, İ., Özgür-Ünlüakın, D., & Bilgiç, T. (2022). Maintenance policy analysis of the regenerative air heater system using factored POMDPs. *Reliability Engineering and System Safety*, 219, 108195.
- Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. New York: MIT Press.
- Landgraf, M., & Hansmann, F. (2019). Fractal analysis as an innovative approach for evaluating the condition of railway tracks. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 233.
- Littman, M. L., Cassandra, A. R., & Kaelbling, L. P. (1995). Learning policies for partially observable environments: Scaling up. In *Machine learning proceedings* (pp. 362–370). Elsevier.
- Luque, J., & Straub, D. (2019). Risk-based optimal inspection strategies for structural systems using dynamic Bayesian networks. *Structural Safety*, 76(68), 80.
- Madanat, S., & Ben-Akiva, M. (1994). Optimal inspection and repair policies for infrastructure facilities. *Transportation Science*, 28(1), 55–62.
- Memarzadeh, M., Pozzi, M., & Zico Kolter, J. (2015). Optimal planning and learning in uncertain environments for the management of wind farms. *Journal of Computing in Civil Engineering*, 29(5), 04014076.
- Meng, L., Gorbet, R., & Kulić, D. (2021). Memory-based deep reinforcement learning for POMDPs. In *2021 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 5619–5626).
- Morato, P. G., Andriotis, C. P., Papakonstantinou, K. G., & Rigo, P. (2023). Inference and dynamic decision-making for deteriorating systems with probabilistic dependencies through Bayesian networks and deep reinforcement learning. *Reliability Engineering and System Safety*, 109144.
- Morato, P. G., Papakonstantinou, K. G., Andriotis, C. P., Nielsen, J. S., & Rigo, P. (2022). Optimal inspection and maintenance planning for deteriorating structural components through dynamic Bayesian networks and Markov decision processes. *Structural Safety*, 94, 102140.
- Papakonstantinou, K. G., & Shinozuka, M. (2014a). Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation. *Reliability Engineering and System Safety*, 130, 214–224.
- Papakonstantinou, K. G., & Shinozuka, M. (2014b). Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part I: Theory. *Reliability Engineering and System Safety*, 130, 202–213.
- Papakonstantinou, K. G., Andriotis, C. P., & Shinozuka, M. (2018). POMDP and MOMDP solutions for structural life-cycle cost minimization under partial and mixed observability. *Structure and Infrastructure Engineering*, 14(7), 869–882.
- Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., & Jayakumar, S., et al. (2020). Stabilizing transformers for reinforcement learning. In *International conference on machine learning* (pp. 7487–7498).

- Profillidis, V. (2016). *Railway management and engineering*. London: Routledge.
- Salvato, E., Fenu, G., Medvet, E., & Pellegrino, F. A. (2021). Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning. *IEEE Access*, 9, 153171–153187.
- Schmidhuber, J. (1990). Reinforcement learning in Markovian and non-Markovian environments. In *Advances in Neural Information Processing Systems* (Vol. 3).
- Schöbi, R., & Chatzi, E. N. (2016). Maintenance planning using continuous state partially observable Markov decision processes and non-linear action models. *Structure and Infrastructure Engineering*, 12(8), 977–994.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. [arXiv:1707.06347](https://arxiv.org/abs/1707.06347)
- Sondik, E. J. (1971). The optimal control of partially observable Markov decision processes. Ph.D. thesis, Stanford University.
- Song, C., Zhang, C., Shafieezadeh, A., & Xiao, R. (2022). Value of information analysis in non-stationary stochastic decision environments: A reliability assisted POMDP approach. *Reliability Engineering and System Safety*, 217, 108034.
- Straub, D., Chatzi, E., Bismut, E., Courage, W., Döhler, M., Faber, M. H., et al. (2017). Value of information: A roadmap to quantifying the benefit of structural health monitoring. In *ICOSSAR-12th international conference on structural safety and reliability*.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. London: MIT Press.
- Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., & Abbeel, P. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International conference on intelligent robots and systems (IROS)* (pp. 23–30).
- Wang, H., Berkers, J., van den Hurk, N., & Layegh, N. F. (2021). Study of loaded versus unloaded measurements in railway track inspection. *Measurement*, 169, 108556.
- Wari, E., Zhu, W., & Lim, G. (2023). A discrete partially observable Markov decision process model for the maintenance optimization of oil and gas pipelines. *Algorithms*, 16(1), 54.
- Zhao, W., Queralta, J. P., & Westerlund, T. (2020). Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *2020 IEEE symposium series on computational intelligence (SSCI)* (pp. 737–744).
- Zhu, P., Li, X., Poupart, P., & Miao, G. (2017). On improving deep reinforcement learning for POMDPs. [arXiv:1704.07978](https://arxiv.org/abs/1704.07978)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Giacomo Arcieri¹  · Cyprien Hoelzl¹ · Oliver Schwery² · Daniel Straub³ · Konstantinos G. Papakonstantinou⁴ · Eleni Chatzi¹

✉ Giacomo Arcieri
giacomo.arcieri@ibk.baug.ethz.ch

Cyprien Hoelzl
hoelzl@ibk.baug.ethz.ch

Oliver Schwery
oliver.schwery@sbb.ch

Daniel Straub
straub@tum.de

Konstantinos G. Papakonstantinou
kpapakon@psu.edu

Eleni Chatzi
chatzi@ibk.baug.ethz.ch

- ¹ Institute of Structural Engineering, ETH Zürich, 8093 Zürich, Switzerland
- ² Swiss Federal Railways SBB, 3000 Bern, Switzerland
- ³ Engineering Risk Analysis Group, Technical University of Munich, 80333 Munich, Germany
- ⁴ Department of Civil and Environmental Engineering, Pennsylvania State University, University Park, PA 16802, USA