

Adaptive planning for risk-aware predictive digital twins

Marco Tezzele^[0000-0001-9747-6328] and
Steven Carr^[0000-0002-5661-3768] and
Ufuk Topcu^[0000-0003-0819-9985] and
Karen E. Willcox^[0000-0003-2156-9338]

Abstract This work proposes a mathematical framework to increase the robustness to rare events of digital twins modelled with graphical models. We incorporate probabilistic model-checking and linear programming into a dynamic Bayesian network to enable the construction of risk-averse digital twins. By modeling with a random variable the probability of the asset to transition from one state to another, we define a parametric Markov decision process. By solving this Markov decision process, we compute a policy that defines state-dependent optimal actions to take. To account for rare events connected to failures we leverage risk measures associated with the distribution of the random variables describing the transition probabilities. We refine the optimal policy at every time step resulting in a better trade off between operational costs and performances. We showcase the capabilities of the proposed framework with a structural digital twin of an unmanned aerial vehicle and its adaptive mission replanning.

1 Introduction

When creating a digital twin (DT) it is important to ensure its robustness to different possible scenarios and rare event occurrences. Moving from considering typical operational settings to more general scenarios that account for the involved risks

Marco Tezzele
University of Texas at Austin, 78712, Austin, TX, US e-mail: marco.tezzele@austin.utexas.edu

Steven Carr
University of Texas at Austin, 78712, Austin, TX, US e-mail: stencarr@utexas.edu

Ufuk Topcu
University of Texas at Austin, 78712, Austin, TX, US e-mail: utopcu@austin.utexas.edu

Karen E. Willcox
University of Texas at Austin, 78712, Austin, TX, US e-mail: kwillcox@oden.utexas.edu

will foster the development of robust and risk-averse digital twins. The aversion to risk is particularly important during the deployment of the DT, that is after the calibration phase is performed, to avoid interruption of the operations due to the occurrence of rare events. To achieve reliability in an engineering system, it is important to account for the uncertainties characterizing the physical asset under consideration, in addition to the uncertainties in its operating environment. The main contribution of this work is to present a method to increase the robustness of DTs by using a probabilistic graphical model (PGM) formulation, parametric Markov decision processes (MDPs), and risk measures to account for rare scenarios. The potential impact on real-world systems is twofold: reliability of DT deployments can be increased and operations can be optimized, resulting in lower operational costs and more effective predictive maintenance.

A DT is distinguished from traditional modeling and simulation through its personalization, that is, it is linked to the underlying physical asset or process, allowing it to represent the particular attributes of that specific asset or process. This DT representation is also dynamically evolving, continually incorporating data from the physical world, and refining the DT's predictive capabilities accordingly [1, 2, 3, 4, 5]. Within the digital twin framework, we need to accomplish different tasks such as data acquisition, solving inverse problems, building computational models, accounting for uncertainty, and performing future state predictions. In recent years, DTs have been proposed for diverse applications such as health monitoring and maintenance planning for spacecrafts [6], railway systems [7], civil engineering structures [8, 9], and unmanned aerial vehicles [10, 11, 12].

A common framework for modeling decision-making in the presence of uncertainty is based on Markov decision processes [13]. MDPs are widely used in reinforcement learning [14], robotics, and planning [15]. Oftentimes, the transitions in an MDP are unrealistically assumed to be precisely known. In this work, we leverage parametric MDPs with transition probabilities modeled as random variables with a given distribution. Parametric MDPs describe models where some probabilities are unknown but explicitly related. Such models form a natural pairing with a digital twin framework where information on one transition probability may inform future decisions. Probabilistic model-checking [16] provides a general framework to verify whether an implemented policy satisfies a wide range of constraints.

The probabilistic graphical model we use is a dynamic Bayesian network with decision nodes, also called dynamic decision network [17, 15, 18, 19]. The PGM encodes the relationships between the variables describing the physical asset and its digital counterpart. The edges in the PGM represent relationships between the connected variables through conditional probability distributions. The uncertainty is thus propagated in the entire graph, making it a powerful tool to account for risk. Among these conditional probability distributions the state transition probability distributions, which encode how the state evolves after taking a given action, are usually set a priori. In this paper, we propose a numerical method to dynamically update the state transition probabilities with a Bayesian approach, feeding the parametric MDP with the new information, and making the graphical model more interconnected with

the physical asset. Better estimates translate into better policies and a more effective digital twin.

A sketch of the end-to-end pipeline is depicted in Figure 1, where we consider a specific physical asset for the sake of clarity. The computational flow starts with the data acquisition from sensors placed on the unmanned aerial vehicle (UAV) wing. The noisy data are used to estimate the digital state and the PGM is populated with another time step. The unrolling is realized by duplicating the nodes and edges of the graph, at a given time, for the next time step. The policy is then updated by solving the MDP using the new state transition probability, and the optimal action is issued, continuing the computational cycle. Future state predictions can be computed for any time step, using the current estimate of the transition probability.

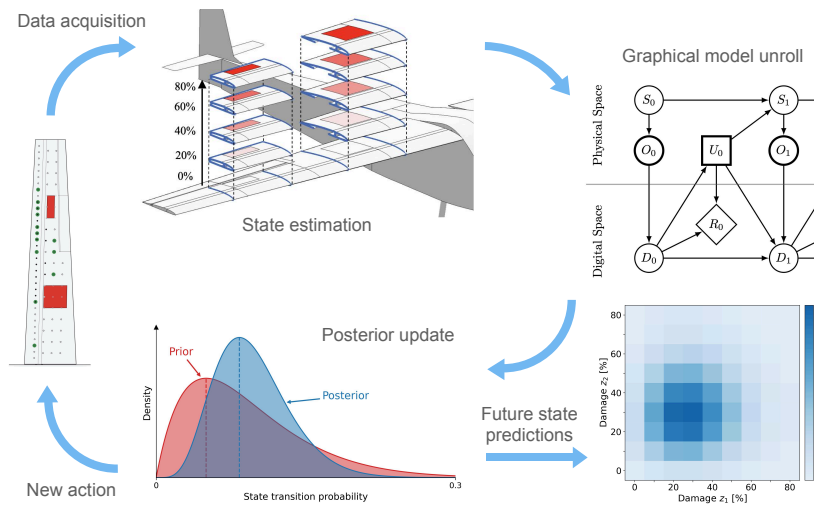


Fig. 1: Abstract representation of the information flow within the DT formulation, from sensor data acquisition to the MDP’s optimal policy update. Starting from strain sensors placed on the wing, we estimate the digital state connected with the structural integrity of the UAV. We then unroll the graphical model with all the variables for a new time step and we update the posterior estimates of the transition probability. We recompute the new optimal policy and we issue a new action to perform. This policy is also used to compute future actions in the prediction phase happening in the digital space.

The paper is organized as follows. In Section 2 we describe the PGM and its predictive capabilities. Section 3 presents the underlying parametric MDP and how the optimal policy is updated using different risk measures. The numerical results with a test mission and two different scenarios are explained in Section 4. Finally, we draw conclusions in Section 5.

2 Probabilistic graphical model formulation for digital twins

This section introduces the PGM and how the variables are connected with the MDP. We follow the approach of Ref. [10] for the PGM formulation.

Figure 2 depicts the encoding of the interactions between the physical and the digital assets through a dynamic decision network, starting from $t = 0$ to the current time $t = t_c = 2$. We indicate with t a generic non-dimensional discretized time step. Circle nodes in the graph represent random variables, diamonds indicate output functions, while square nodes denote the decisions taken to enact specific actions. Following the edges we track the dependencies between variables. Notice how we can unroll the directed graphical model for every time step just by repeating its core structure. We consider the following time-dependent random variables: the unknown underlying physical state S , the observed sensor measurements O , the digital state D , the actions U (also called controls) that can be issued, and the reward R . The direct solid edges encode the temporal dependencies between the variables through conditional probability distributions.

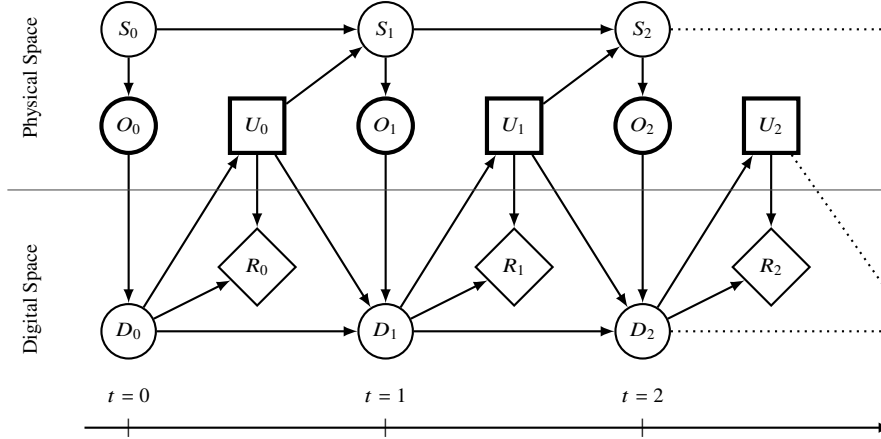


Fig. 2: Representation of the dynamic decision network used to encode the relationships between the physical and the digital spaces for the first 3 time steps. Square nodes denote actions, while diamond nodes denote the reward function. Bold outlines represent observed (deterministic) quantities, while thin outlines stand for estimated (random) variables.

Decision-making is achieved by leveraging MDPs. Let \mathcal{S} be the set of possible states, \mathcal{A} be the set of possible actions, \mathcal{P} be a set of transition probabilities, and \mathcal{R} be the reward set. In the remainder of the work we assume all these sets are finite. A finite Markov decision process is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. The state transition probability $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ describes the probability of

transitioning from the state s to s' at time $t - 1$ after taking the action u . It is defined as:

$$p(s' | s, u) := \Pr\{S_t = s' | S_{t-1} = s, U_{t-1} = u\}. \quad (1)$$

Analogously the reward function R is defined as:

$$R(s', s, u) := \Pr\{r_t | S_t = s', S_{t-1} = s, U_{t-1} = u\}, \quad (2)$$

where r_t is the reward at time t . The probability of taking an action u at time t is given by a policy $\pi_t : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ defined as:

$$\pi_t(s, u) := \Pr\{U_t = u | S_{t-1} = s\}. \quad (3)$$

To issue an actual action u_t we take the $\text{argmax}_{u \in \mathcal{A}} \pi_t(s, u)$. A classical MDP satisfies the Markov property [13], that is

$$\Pr\{S_t = s' | S_{t-1} = s_{t-1}, U_{t-1} = u_{t-1}, \dots, S_1 = s_1, U_1 = u_1\} = \Pr\{S_t = s' | S_{t-1} = s_{t-1}, U_{t-1} = u_{t-1}\}. \quad (4)$$

The goal is to leverage the resolution of an MDP to find the optimal sequence of actions to issue to maximize the expected reward. The MDP is thus a key component to provide optimal decision-making capabilities to the PGM. For more general cases where the Markov property does not hold, the topology of the graph should reflect that by adding more edges connecting all the necessary time instants.

We now show how uncertainty is propagated through the graph from data acquisition to future state predictions. We denote by $p(d_t)$ the probability distribution of $D_t = d_t$, for any possible digital state d_t , with similar notation employed for the other random variables comprising the graphical model. We have $S_t \sim p(s_t)$ for the physical state, $O_t \sim p(o_t)$ for the observational data, $U_t \sim p(u_t)$ for the action, and $R_t \sim p(r_t)$ for the reward. We can factorize the joint distributions of some of the variables conditioned on the observed ones, as in the following:

$$p(D_0, \dots, D_{t_c}, R_0, \dots, R_{t_c} | o_0, \dots, o_{t_c}, u_0, \dots, u_{t_c}) \propto \prod_{t=0}^{t_c} \left[\phi_t^{\text{data}} \phi_t^{\text{dynamics}} \phi_t^{\text{reward}} \right],$$

where the factors are:

$$\phi_t^{\text{data}} = p(O_t = o_t | D_t), \quad (5)$$

$$\phi_t^{\text{dynamics}} = p(D_t | D_{t-1}, U_{t-1} = u_{t-1}), \quad (6)$$

$$\phi_t^{\text{reward}} = p(R_t | D_t, U_t = u_t). \quad (7)$$

The factor ϕ_t^{data} encodes the assimilation of the sensor measurements. With ϕ_t^{dynamics} we factorize the belief about the digital state at the current time step, given the digital state at the previous time step and the last enacted control. Finally ϕ_t^{reward} encodes the evaluation of the reward function which is a performance measure.

The state space and the action space are discrete so the conditional probabilities involving these spaces are encoded in the graph through conditional probability tables (CPTs). These CPTs are usually set a priori, extrapolated from historical mission logs or manufacturer's blueprints of the physical asset. The structure of the CPTs describing ϕ_t^{dynamics} , also called transition probability matrices, is further analyzed in Section 3.1.

The prediction for future time steps is performed using just a subset of the variables comprising the DT framework, as depicted in Figure 3. To decide which action to perform we query the policy π , mapping the current digital state to a distribution of actions. The action with the highest density is the one that maximizes the total future rewards. After updating the transition probability estimates in the graph at every time step, we need to recompute the corresponding optimal policy. The policy used for all the times $t \geq t_c$ is denoted with π_{t_c} and incorporates the most updated state transition probability estimates for all the possible actions. Notice how in Figure 3 the actions U_t are depicted as random variables and not anymore as decision nodes since we propagate the uncertainty throughout all the prediction stages. We have indeed that $U_{t_c+i} \sim \pi_{t_c}(D_{t_c+i})$ for $i = 0, 1, 2$.

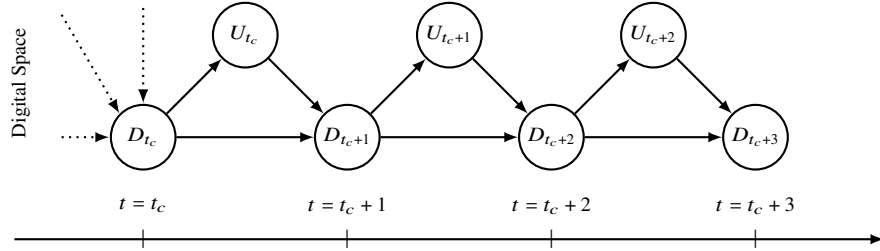


Fig. 3: Dynamic Bayesian network used for the prediction of the digital state evolution with the associated uncertainty. We assume to have the estimation of D_{t_c} given the observations from the physical asset O_{t_c} , the previous issued action U_{t_c-1} , and the previous digital state D_{t_c-1} . From the current time t_c we show the graph used to predict the next 3 actions and digital states, using the policy π_{t_c} .

The target belief state is extended as in the following:

$$\begin{aligned}
 & p(D_0, \dots, D_{t_p}, R_0, \dots, R_{t_c}, U_{t_c+1}, \dots, U_{t_p} \mid o_0, \dots, o_{t_c}, u_0, \dots, u_{t_c}) \\
 & \propto \prod_{t=0}^{t_c} [\phi_t^{\text{data}} \phi_t^{\text{dynamics}} \phi_t^{\text{reward}}] \prod_{t=t_c+1}^{t_p} [\phi_t^{\text{step}} \phi_t^{\text{control}}], \quad (8)
 \end{aligned}$$

where $\phi_t^{\text{control}} = p(U_t \mid D_t)$ is given by the last computed policy π_{t_c} , and $\phi_t^{\text{step}} = p(D_t \mid D_{t-1}, U_{t-1})$.

3 Online adaptivity of risk-aware parametrized policies

Section 3.1 introduces parametric Markov decision processes defined by random variables and how the state transition probability is refined after each data acquisition step. Section 3.2 presents how risk measures are incorporated into the PGM formulation.

3.1 Parametric Markov decision processes

For every possible action, there is a corresponding transition probability matrix encoding the conditional probabilities for all the combinations of states. The transition probability matrix P associated to the action $u \in \mathcal{A}$ is defined as

$$P(u) := (p_{ij})_{i,j=1}^n = (p(s_j | s_i, u))_{i,j=1}^n, \quad \text{for } s_i, s_j \in \mathcal{S}, \quad (9)$$

that is:

$$P(u) = \begin{pmatrix} p(s_1 | s_1, u) & p(s_2 | s_1, u) & \dots & p(s_n | s_1, u) \\ p(s_1 | s_2, u) & p(s_2 | s_2, u) & \dots & p(s_n | s_2, u) \\ \vdots & \vdots & \ddots & \vdots \\ p(s_1 | s_n, u) & p(s_2 | s_n, u) & \dots & p(s_n | s_n, u) \end{pmatrix}, \quad (10)$$

where $p_{ij} \geq 0$, and $\sum_{j=1}^n p_{ij} = 1$, for all $i \in [1, \dots, n]$. This matrix can take different forms depending on the underlying MDP.

We assume there exists a set of actions, denoted by $\mathcal{A}^D \subset \mathcal{A}$, which produces a deterministic outcome such as moving towards a known direction, for example. We define \mathcal{A}^D as

$$\mathcal{A}^D := \{u \in \mathcal{A} \mid \forall s \in \mathcal{S}, \exists s' \in \mathcal{S} \text{ s.t. } p(s' | s, u) = 1\}. \quad (11)$$

We denote its complementary set with $\mathcal{A}^N = \mathcal{A} \setminus \mathcal{A}^D$ and we define it as

$$\mathcal{A}^N := \{u \in \mathcal{A} \mid \forall s \in \mathcal{S} \setminus \{\bar{s}\}, \exists s', s'' \in \mathcal{S}, s' \neq s'', \exists q \in (0, 1) \text{ s.t.} \\ p(s' | s, u) = q, p(s'' | s, u) = 1 - q\}, \quad (12)$$

where \bar{s} is the only state for which $p(\bar{s} | \bar{s}, u) = 1$.

We restrict to the case for which only two outcomes are possible, i.e., the state varies smoothly without abrupt changes that lead to the transition to many possible states. This assumption is not too restrictive because we can always decrease the acquisition time step up to a certain computational threshold. Moreover, s' and s'' in Eq. (12) will always be two consecutive discretized states.

Let us consider an action $u^D \in \mathcal{A}^D$, then we have that $P(u^D)$ is a permutation of the identity matrix, which is known a priori. If, instead, we consider an action $u^N \in \mathcal{A}^N$, then we can parametrize $P(u^N)$ with the parameter q introduced in Eq. (12).

We introduce $P(u^N; q)$, an upper bidiagonal matrix¹, which has the following form:

$$P(u^N; q) = \begin{pmatrix} 1-q & q & 0 & \dots & 0 & 0 \\ 0 & 1-q & q & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1-q & q \\ 0 & 0 & 0 & \dots & 0 & 1 \end{pmatrix}. \quad (13)$$

We refer to this case as state-invariant transition, because the probability of transitioning from the current state to the contiguous discretized state does not depend on the current state. In particular, if s is a scalar, the superdiagonal elements are equal to q , while the main diagonal elements are $1 - q$, except the last one which is 1. Figure 4 illustrates the different transition probability matrices arising from our setting. The general case with a state-dependent transition, denoted as $P(u; q(s))$ in Figure 4, will be considered in future studies.

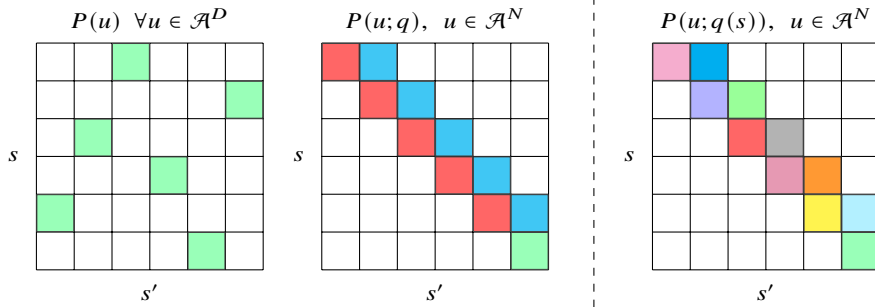


Fig. 4: Illustration of the types of transition matrices considered in this work. On the left panel the matrix for the actions in \mathcal{A}^D , where green stands for 1 and white for 0. In the central panel we have the upper bidiagonal with fixed transitions for actions in \mathcal{A}^N , where red stands for $1 - q$ and blue for q . On the right panel the generic upper bidiagonal for transition probabilities varying between states.

In general, q is unknown and possibly varying in time. When the parameter q is incorporated to the transition probabilities, we can model the system as a parametric MDP, which describes sets of sets on the transition probabilities. We model q with a random variable Q . By accounting for the unknown nature of q , we may compute robust policies covering the worst case transitions [20]. As we gather more information on the true value of q the policy will dynamically improve. We leverage the PGM formulation and the acquired observations to refine our belief on q at every time steps. Other approaches to describe uncertainty in parametric MDPs

¹ In general the number of diagonals is equal to 2^d , where d is the dimension of the state s .

use intervals [21, 22, 23], robust MDPs [24] which consider nonconvex sets for the transition probabilities [25], or temporal specifications [26].

Suppose that the random variable Q describing the state transition probability of our parametric MDP is beta-distributed with left parameter $\alpha > 1$ and right parameter $\beta > 1$, that is $Q \sim \mathcal{Be}(\alpha, \beta)$. Let us also suppose that $X = (X_1, X_2, \dots)$ is a sequence of indicator random variables such that given $Q = q \in (0, 1)$, X is a conditionally independent sequence with

$$\Pr\{X_i = 1 \mid Q = q\} = q, \quad \forall i \in \mathbb{N}^+, \quad (14)$$

then X is called a beta-Bernoulli process with parameters α and β . X_i represents the outcome of the i -th trial, with 1 denoting a success and 0 a failure. In our particular case $X_i = 1$ if the state at time i is different from the state at time $i - 1$, and $X_i = 0$ otherwise.

It can be proven that the conditional distribution of q given $(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n)$ is a beta with left parameter $\alpha + k$ and right parameter $\beta + n - k$, with $k = \sum_{i=1}^n x_i$. Since the prior $p(q)$ and the posterior $p(q \mid X = x)$ belong to the same probability distribution family they are called conjugate distributions. In particular, the beta distribution is the conjugate prior to the Bernoulli distribution. This posterior distribution could then be used as the new prior, without the need for expensive MCMC. The parameters α and β are updated incorporating new experience as it comes in a computationally efficient way.

3.2 Risk measures

As we showed above, computing the posterior distribution of the continuous random variable Q is computationally efficient. A common choice for the best estimate of the transition probability would be the maximum a posteriori (MAP) estimate. The MAP estimate does not account for rare events and the associated risk, an important aspect in engineering applications where we need to be aware of the risks connected to failures. To overcome these limitations we use the conditional value at risk (CVaR) for the posterior estimate of the transition probabilities, thus enhancing the reliability and the risk-awareness of the digital twin. CVaR was first introduced for portfolio risk analysis [27, 28, 29], and successfully used over the years in engineering applications such as in [30, 31, 32, 33, 34].

To define the CVaR we first introduce the value at risk (VaR) [35, 36], also called α -quantile. For continuous distributions the VaR reads as follows:

$$\text{VaR}_{1-\alpha}(Q) := \inf_{t \in \mathbb{R}} \{t : \Pr(Q \leq t) \geq 1 - \alpha\}, \quad \alpha \in (0, 1]. \quad (15)$$

CVaR, also called average value at risk or superquantile, is a coherent risk measure, even though it is derived from the value at risk, which is not. It measures the expected loss in the lower tail given that a particular threshold has been crossed. It

provides a measure of the average loss magnitude beyond the α -quantile level. For continuous distributions, the CVaR is defined as

$$\text{CVaR}_{1-\alpha}(Q) := \frac{1}{\alpha} \int_0^\alpha \text{VaR}_{1-t}(Q) dt = \mathbb{E}[Q \mid Q \geq \text{VaR}_{1-\alpha}(Q)], \quad \alpha \in (0, 1].$$

It follows that for every $\alpha \in (0, 1]$ this inequality holds: $\text{VaR}_{1-\alpha}(Q) \leq \text{CVaR}_{1-\alpha}(Q)$.

Figure 5 depicts the relation between the MAP estimate, the VaR, and the CVaR for a beta-distributed random variable.

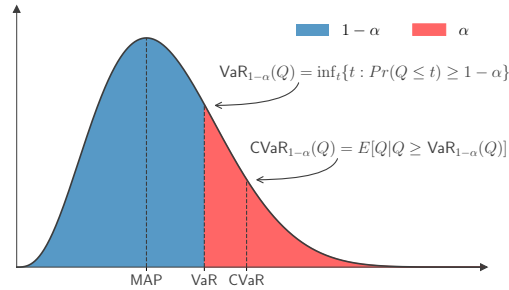


Fig. 5: Maximum a posteriori (MAP) estimate, value at risk (VaR), and conditional value at risk (CVaR) for a generic probability distribution followed by the random variable Q .

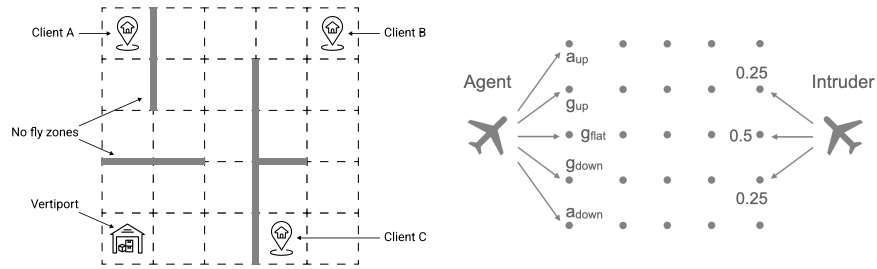
4 Numerical results

This section presents two approaches to planning using robust policy optimization for a parametric MDP. The first is a risk-averse planning structure, whereby the model checking tool will compute the set of policies that are guaranteed to satisfy a given safety specification and then select the optimal. The second is used when we cannot make an almost-sure guarantee on safety but we seek to minimize the total cost of a high-impact safety incident (such as wing failure). In the next section we will describe the conditions that allow for the first form and planning and the subsequent modifications that inform the second. The probabilistic graphical model framework is implemented using the open source Python library pgmpy [37].

4.1 Example scenarios

We present two case studies for UAV applications where our proposed framework provides impact when planning. These cases are described as:

1. **UAV delivery** (Figure 6a) – a UAV is delivering items in a gridworld environment with a series of target points. At each time interval there is a time-based cost for transit and a non-zero probability $q \in (0, 1]$ that the wings may incur some damage in the panels, i.e., local stiffness degradation, shown in Figure 1. The UAV may elect to take an *aggressive* or *gentle* maneuver. An aggressive maneuver results in a faster transition between states (i.e., a higher reward $r_{AGG} > r_{GEN}$) but it has a higher probability of incurring damage on one of the wing panels (i.e., $q_{AGG} > q_{GEN}$).
2. **Aircraft collision avoidance** (Figure 6b) – a modified MDP model [38] in a similar environment to the UAV delivery, an aircraft is attempting to deconflict with an oncoming vehicle by selecting the appropriate altitude band. At each time-interval they may select from one of five decisions for their next altitude band, three decisions $\{g_{up}, g_{flat}, g_{down}\}$ form the gentle band (which carry probability q_{GEN} of causing damage to the wing panel) and two decisions $\{a_{up}, a_{down}\}$ are more aggressive maneuvers that reach further states at the cost of higher probability q_{AGG} of causing damage to the wings. The scenario is concluded after the vehicles occupy the middle band of points. If both vehicles are at the same altitude in this trajectory then they are determined to have a risk of crash and are considered safe otherwise. The opponent moves stochastically between altitude bands according to some known distribution [38].



(a) The UAV departs from a warehouse or a vertiport and has to deliver the package to one of the clients in the served area. The agent computes the best path and the associated maneuvers to take at every time step.

(b) Aircraft attempts to avoid an incoming vehicle using the same x - y trajectory. It may select a band of altitude with a gentle maneuver (g_{up} , g_{flat} , g_{down}) or an aggressive one (a_{up} , a_{down}).

Fig. 6: (a) Last mile delivery and (b) Vertical air collision avoidance examples.

These scenarios can be modeled by a parametric MDP with two parameters: the probability that the aggressive maneuver causes damage (q_{AGG}) and the probability

that the gentle maneuver causes damage (q_{GEN}). There are numerous methods for synthesizing policies that ensure an agent modeled as a parametric MDP will adhere to a temporal logic specification [39, 40, 41]. In this work, we must both ensure a probability of reaching a goal state with the minimization of an expected cost and therefore we employ the probabilistic model checker Storm [42]. Using Storm, we instantiate the worst case values for the damage probability and formulate the problem as a linear program [43]. The computed policy is the policy that minimizes the expected cost to reach the goal while ensuring the UAV will remain below the damage threshold. Additionally, since Storm is an efficient probabilistic model checker, we may update the expected value of the parameters as the aircraft makes decisions and computes updates to the policy online.

In these examples, the worst case value of the parameter is predictable and therefore we chose an efficient tool to compute the policy. When it is uncertain how a given parameter value will affect the property, one may employ methods such as sequential convex programming [20]. For other approaches for verification of uncertain parametric MDPs see [44].

In each case above the objective of the aircraft can be described using a temporal logic formula such as $\phi = \neg C \cup G$, i.e., do not crash until you reach the goal where C describes a crash state such as wing damage, occupying the same altitude band as an opponent or violating a no-fly region, and G is some goal condition such as delivering an object or passing an opponent in a different altitude band.

In both scenarios we seek to ensure that:

1. we successfully navigate to a goal G ;
2. without incurring catastrophic structural damage to our wings $\neg C$;
3. with the minimum expected cost of arrival.

At each decision point, there is a trade-off between taking a cheap, aggressive but higher-risk action against an expensive, gentle but lower-risk action. For an almost-sure guarantee on arriving to the goal without incurring catastrophic damage there needs to be either a zero probability action (i.e., $q_{\text{GEN}} = 0$) for wing damage or a minimum number of decisions smaller than the number of damage bins on the wing (e.g., 5 steps to the goal and 10 bins for structural damage transitions to occur). While there exist many situations where $q_{\text{GEN}} = 0$, we are most interested in presenting the trade-off between ensuring safety and optimizing for time taken as well as showing the value of the predictive input of an updating digital twin on the plan. Accordingly, most results, unless otherwise specified will be minimizing the expected cost, where the cost of a failure condition is graded as significantly higher than any time-based action.

For all the test cases we assume that the calibration of the structural DT has already been performed as in [10]. The initial prior on the transition probabilities is provided as a beta distribution $\mathcal{Be}(2, \beta)$ with left parameter $\alpha = 2$ and given mode. The formula to compute the mode of a beta-distributed random variable given its parameters is the following:

$$\frac{\alpha - 1}{\alpha + \beta - 2}, \quad \text{for } \alpha, \beta > 1. \quad (16)$$

By using Equation (16) we compute the corresponding right parameter β . To ensure the computational model interpretability we round β to the closest integer.

4.2 Digital state estimation

This section describes how the digital state is estimated from the noisy observational data coming from the sensors placed on the wing of the UAV. We emphasize there are many different methods that can be leveraged to solve the inverse problem from data to parameters comprising the digital state. The reader can refer to Bayesian state estimation techniques or more generally to data assimilation methods [45, 46]. The crucial aspect is to compute the uncertainty associated with the estimations to fully leverage the graphical model. In this application the digital state is a four-dimensional vector composed by $\mathbf{x} \in \mathbb{N}^2$, representing the deterministic UAV position coordinates, and by $\mathbf{z} \in \mathbb{D} \subset \mathbb{R}^2$, which is a vector comprising the structural health parameters associated to the UAV wing. Since the UAV position is not affected by uncertainty in our setting, we focus on \mathbf{z} . The vector \mathbf{z} is defined as

$$\mathbf{z} = [z_1, z_2] \in \mathbb{D} \subset \mathbb{R}^2, \quad \mathbb{D} := \left\{ \left(\frac{i}{10}, \frac{j}{10} \right) \mid i, j \in [0, 1, \dots, 8] \right\}. \quad (17)$$

It represents the percentage of stiffness degradation (from 0% to 80%) of the two red regions on the wing showed in Figure 1. The vector \mathbf{z} is estimated at every time step through a set of strain measurements $\boldsymbol{\epsilon} = \{\epsilon^j\}_{j=1}^{24}$, coming from sensors placed on the UAV. We assume these measurements are corrupted with additive white Gaussian noise:

$$\hat{\epsilon}_t^j = \epsilon_t^j + v_t, \quad v_t \sim \mathcal{N}(0, \sigma), \quad (18)$$

where the subscript denotes the time instant, and $\sigma = 10$ so that the sensor model does not match the simulated measurements. To estimate \mathbf{z} we solve the following inverse problem

$$\min_{\boldsymbol{\theta}} \frac{1}{2} \|F(\boldsymbol{\theta}) - \hat{\boldsymbol{\epsilon}}\|_2^2 + \|\boldsymbol{\theta}\|_2, \quad (19)$$

where $\boldsymbol{\theta} \in [0, 0.8]^2$ and $\hat{\boldsymbol{\epsilon}} \in \mathbb{R}^{24}$ is the vector comprising the noisy strain measurements. We use an L_2 regularization to favor low norm solutions. The $\boldsymbol{\theta}^*$ that realizes the minimum is then approximated with the nearest point in the discrete space \mathbb{D} . The map $F : [0, 0.8]^2 \rightarrow \mathbb{R}^{24}$ describes the evaluation of the reduced order model based on the static condensation reduced basis element (SCRBE) method developed in [47, 48, 49] and presented in [50, 51] for the UAV we are considering. The SCRBE technique brings together the scalability of component-based formulations and the accuracy of the reduced basis method.

The resulting accuracy of the state estimation is equal to 75.4%. The accuracy is measured by sampling 100 realizations of the noisy sensor data and then computing the average of the confusion matrix' diagonal. In particular, if we look at the estimation of the single components of the vector \mathbf{z} , we observe that z_1 is estimated with

a 100% accuracy, and all the errors are due to z_2 , which is in line with the findings in [10]. These measures are factored in the PGM through ϕ_t^{data} .

4.3 Risk-averse case using CVaR

The simulated mission is composed of 40 time steps and it starts from an initial state $\mathbf{z} = [0.2, 0.2]$. In Figure 7 we show how the data assimilation performs for z_1 (bottom part) and z_2 (top part). The underlying transition probabilities are 3% and 10% for the gentle and aggressive actions, respectively. As we can see the tracking of z_1 is perfect, while for z_2 we need some time steps to identify a state change. This is observed also in [10] and it is due to the chosen state discretization levels and the position on the wing, closer to the tip, which is more sensitive to noise.

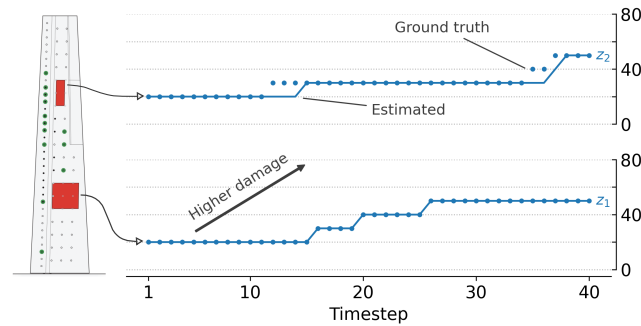


Fig. 7: Evolution of the digital states z_1 and z_2 for the test mission. The initial state \mathbf{z} at time $t = 1$ is set to $[0.2, 0.2]$.

In Figure 8 we report the expected cost to complete the mission at every time step, and the actual cumulative cost spent to follow the optimal policy. To estimate the transition probabilities we use the conditional value at risk with $\alpha = 0.25$, i.e., the CVaR at the upper quantile. The adaptive policy results in a reduced cost equal to 22%. This reduction is due to the policy refinement and the posterior updates. If we consider the last-mile delivery scenario, the cost reduction translates to faster deliveries without incurring a fatal crash, while accounting for the risk associated with extreme events. By looking at the actual cost slope we can identify the switch between gentle and aggressive actions, which correspond to a cost equal to 25 and 10, respectively. The priors associated to the gentle and aggressive maneuvers are $\mathcal{B}e(2, 66)$ and $\mathcal{B}e(2, 20)$, respectively. They are reported in Figure 9 together with the corresponding posteriors after 40 time steps. We can see that in both cases we are converging to the unknown underlying transitions.

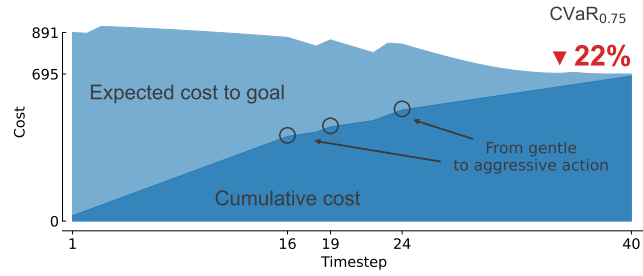


Fig. 8: Expected and effective cost to complete a mission using the conditional value at risk to estimate the transition probability matrices.

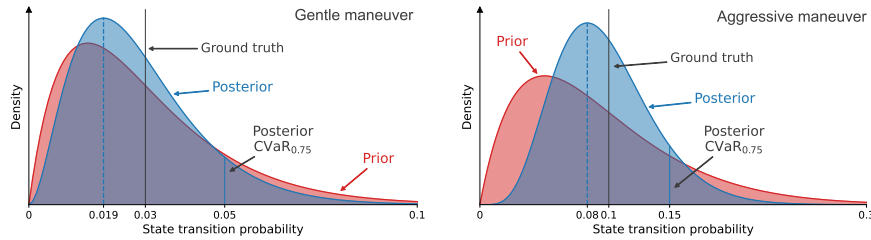


Fig. 9: Prior and posterior distributions corresponding to the gentle maneuver (left panel) and to the aggressive maneuver (right panel) for the risk-averse test case.

4.4 Best estimate case using MAP

For this test case we use the maximum a posteriori estimate of the beta distributions to approximate the transition probabilities. The initial state at $t = 1$ and the costs associated with the two actions are the same as in the previous test case. The underlying transition probability for the gentle maneuver is equal to 2%, while for the aggressive maneuver is 10%.

By comparing the expected cost to goal and the actual cumulative cost we notice a similar reduction as before equal to 22.07%, as we can see in Figure 10. The similarities are due to the fact that we have a limited set of possible actions and the length of the mission is also the same in the two scenarios.

The priors for gentle and aggressive actions are $\mathcal{Be}(2, 14)$ and $\mathcal{Be}(2, 5)$, respectively. They correspond to a MAP equal to 7% and 20%. In Figure 11 we also plotted the posteriors after the last time step $t = 40$. For the aggressive maneuver the MAP estimate identifies almost perfectly the ground truth mode, while for the gentle action we need more time steps.

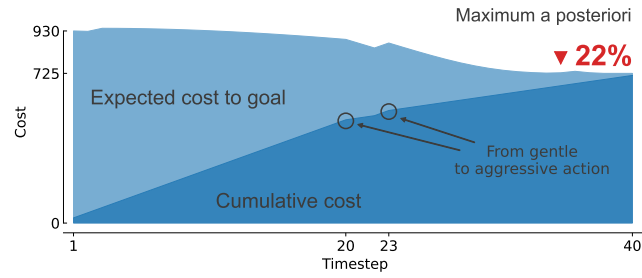


Fig. 10: Expected and effective cost to complete a mission using the maximum a posteriori estimate to estimate the transition probability matrices.

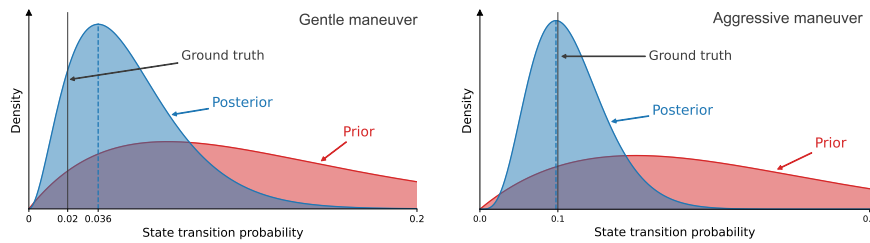


Fig. 11: Prior and posterior distributions corresponding to the gentle maneuver (left panel) and to the aggressive maneuver (right panel) for the maximum a posteriori estimate test case.

4.5 Digital state predictions

The PGM allows making predictions by propagating all the uncertainties in the graph. By leveraging the subgraph presented in Figure 3, we compute the evolution of the digital state with the policy π_{t_c} . Figure 12 depicts the digital state starting from $t_c = 0$, where we have a discrete distribution corresponding to a 75% probability of being at the state $[0, 0]$ and a 25% probability of being in $[0, 0.1]$. The prediction horizon is extended over 70 time steps in the future so that $t_p = t_c + 70$. In the central panel we show the estimates at half of the time horizon, while in the right panel the final estimates. In this example we employ a posterior estimate of the transition probabilities equal to 2% and 10% for the gentle and aggressive actions, respectively. The DT formulation informs about the expected future degradation of structural health, allowing to planning of preventive interventions and maintenance.

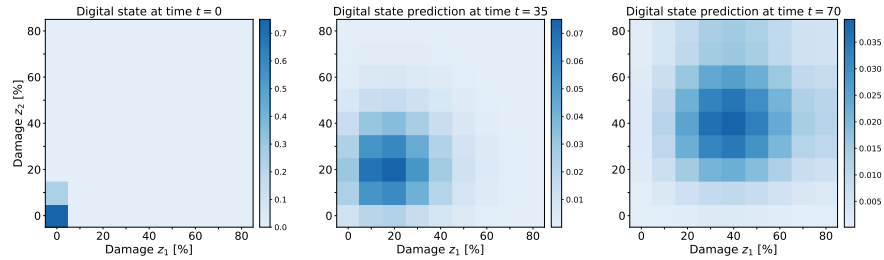


Fig. 12: Digital state predictions. Initial estimate in the left panel ($t = 0$), predictions after 35 time steps in the central panel, and after 70 time steps in the right panel. The colorbar represents the probability of being in a particular state $[z_1, z_2]$.

5 Conclusions and future perspectives

In this paper we integrated uncertainty quantification, risk-awareness, adaptive planning for autonomous systems, and predictive capabilities using a probabilistic graphical model for digital twins. We showed how it is possible to dynamically update the underlying assumptions in the creation of a PGM using parametric Markov decision processes. We demonstrated that the decision-making agent will deploy a robust policy while dynamically refining its belief from new sensor data. The dynamic update of the policy is a crucial characteristic that every digital twin should have since it increases personalization and provides better decision-making while accounting for risk and rare events. Not only is the DT able to monitor the physical asset in a more realistic way, but also the graphical model improves the DT's prediction accuracy. The increased personalization of the DT represents a stepping stone for enhanced predictive maintenance and monitoring.

The need of recomputing the optimal policy at every time step could be computationally demanding depending on the dimension of the discretized variables and on the actual computational power on board. Therefore, the ability to accurately describe the real-world scenario by the MDP is limited by the actual controller installed on the UAV. Nevertheless, the methods presented in this work aim at setting a general formulation for digital twins that are able to adapt to sensed scenarios by recomputing optimal policies, and accounting for risk. Future studies will be conducted considering more general parametrized transition probabilities.

Acknowledgements The authors acknowledge support from the NASA University Leadership Initiative under Cooperative Agreement 80NSSC21M0071. The authors also thank Michael Kapteyn for the UAV's wing illustrations, and Akselos for the structural analysis and reduced order modeling software used (Akselos Integra v4.5.9).

Competing Interests The authors have no conflicts of interest to declare that are relevant to the content of this chapter.

References

1. AIAA Digital Engineering Integration Committee. Digital twin: Definition & value. *AIAA and AIA Position Paper*, 2020.
2. Eric J Tuegel, Anthony R Ingraffea, Thomas G Eason, and S Michael Spottswood. Reengineering Aircraft Structural Life Prediction Using a Digital Twin. *International Journal of Aerospace Engineering*, 2011:154798, 2011.
3. Michael Grieves and John Vickers. Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdisciplinary perspectives on complex systems: New findings and approaches*, pages 85–113, 2017.
4. Adil Rasheed, Omer San, and Trond Kvamsdal. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access*, 8:21980–22012, 2020.
5. Steven A Niederer, Michael S Sacks, Mark Girolami, and Karen E Willcox. Scaling digital twins from the artisanal to the industrial. *Nature Computational Science*, 1(5):313–320, 2021.
6. Yumei Ye, Qiang Yang, Fan Yang, Yanyan Huo, and Songhe Meng. Digital twin for the structural health management of reusable spacecraft: a case study. *Engineering Fracture Mechanics*, 234:107076, 2020.
7. Giacomo Arcieri, Cyprien Hoelzl, Oliver Schwery, Daniel Straub, Konstantinos G Papakonstantinou, and Eleni Chatzi. Bridging pomdps and bayesian decision making for robust maintenance planning under model uncertainty: An application to railway systems. *Reliability Engineering & System Safety*, page 109496, 2023.
8. Matteo Torzoni, Marco Tezzele, Stefano Mariani, Andrea Manzoni, and Karen E. Willcox. A digital twin framework for civil engineering structures. *Computer Methods in Applied Mechanics and Engineering*, 418:116584, January 2024.
9. Pablo G Morato, Charalampos P Andriotis, Konstantinos G Papakonstantinou, and Philippe Rigo. Inference and dynamic decision-making for deteriorating systems with probabilistic dependencies through bayesian networks and deep reinforcement learning. *Reliability Engineering & System Safety*, 235:109144, 2023.
10. Michael G Kapteyn, Jacob VR Pretorius, and Karen E Willcox. A probabilistic graphical model foundation for enabling predictive digital twins at scale. *Nature Computational Science*, 1(5):337–347, 2021.
11. Michael G Kapteyn and Karen E Willcox. Design of Digital Twin Sensing Strategies via Predictive Modeling and Interpretable Machine Learning. *Journal of Mechanical Design*, 144(9):091710, 08 2022.
12. Andrew McClellan, Joseph Lorenzetti, Marco Pavone, and Charbel Farhat. A physics-based digital twin for model predictive control of autonomous unmanned aerial vehicle landing. *Philosophical Transactions of the Royal Society A*, 380(2229):20210204, 2022.
13. Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.
14. Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.
15. Stuart J Russell. *Artificial Intelligence: A Modern Approach*. Pearson Education, London, UK, 2020.
16. Marta Kwiatkowska, Gethin Norman, and David Parker. Probabilistic model checking: Advances and applications. In *Formal System Verification*, pages 73–121. Springer International Publishing, jun 2017.
17. Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, Cambridge, MA, 2009.
18. Thomas Dean and Keiji Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, 1989.
19. Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, Cambridge, MA, 2012.
20. Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Convex Optimization for Parameter Synthesis in MDPs. *IEEE Transactions Automatic Control*, 67(12):6333–6348, 2022.

21. Benoît Delahaye, Kim G Larsen, Axel Legay, Mikkel L Pedersen, and Andrzej Wąsowski. Decision problems for interval Markov chains. In *Language and Automata Theory and Applications: 5th International Conference, LATA 2011, Tarragona, Spain, May 26-31, 2011. Proceedings 5*, pages 274–285. Springer, 2011.
22. Robert Givan, Sonia Leach, and Thomas Dean. Bounded-parameter Markov decision processes. *Artificial Intelligence*, 122(1-2):71–109, 2000.
23. Alberto Puggelli, Wenchao Li, Alberto L Sangiovanni-Vincentelli, and Sanjit A Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *Computer Aided Verification: 25th International Conference, CAV 2013, Saint Petersburg, Russia, July 13-19, 2013. Proceedings 25*, pages 527–542. Springer, 2013.
24. Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.
25. Arnab Nilim and Laurent El Ghaoui. Robust control of markov decision processes with uncertain transition matrices. *Operations Research*, 53(5):780–798, 2005.
26. Eric M Wolff, Ufuk Topcu, and Richard M Murray. Robust control of uncertain Markov decision processes with temporal logic specifications. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pages 3372–3379. IEEE, 2012.
27. R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2(3):21–41, 2000.
28. R Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, 2002.
29. R Tyrrell Rockafellar and Johannes O Royset. Superquantiles and Their Applications to Risk, Random Variables, and Regression. In *Theory Driven by Influential Applications*, chapter 8, pages 151–167. INFORMS, 2013.
30. Weini Zhang, Hamed Rahimian, and Güzin Bayraksan. Decomposition Algorithms for Risk-Averse Multistage Stochastic Programs with Application to Water Allocation under Uncertainty. *INFORMS Journal on Computing*, 28(3):385–404, 2016.
31. JO Royset, L Bonfiglio, G Vernengo, and S Brizzolara. Risk-adaptive set-based design and applications to shaping a hydrofoil. *Journal of Mechanical Design*, 139(10):101403, 2017.
32. Huanhuan Yang and Max Gunzburger. Algorithms and analyses for stochastic optimization for turbfan noise reduction using parallel reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 319:217–239, 2017.
33. Anirban Chaudhuri, Boris Kramer, Matthew Norton, Johannes O Royset, and Karen E Willcox. Certifiable risk-based engineering design optimization. *AIAA Journal*, 60(2):551–565, 2022.
34. Facundo Airaud, Harbir Antil, Rainald Lohner, and Umarmkhon Rakhimov. *On the Use of Risk Measures in Digital Twins to Identify Weaknesses in Structures*. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, 2024.
35. Matthew Pritsker. Evaluating Value at Risk Methodologies: Accuracy versus Computational Time. *Journal of Financial Services Research*, 12(2-3):201–242, 1997.
36. Glyn A Holton. *Value-at-Risk: Theory and Practice*. Academic Press, Elsevier, 2003.
37. Ankur Ankan and Abinash Panda. pgmpy: Probabilistic graphical models using Python. In *Proceedings of the 14th Python in Science Conference (SCIPY 2015)*. Citeseer, 2015.
38. Mykel J Kochenderfer, Jessica E Holland, and James P Chryssanthacopoulos. Next-Generation Airborne Collision Avoidance System. *Lincoln Laboratory Journal*, 19(1):17–33, 2012.
39. Christian Dehnert, Sebastian Junges, Nils Jansen, Florian Corzilius, Matthias Volk, Harold Brountjes, Joost-Pieter Katoen, and Erika Ábrahám. Prophesy: A probabilistic parameter synthesis tool. In *Computer Aided Verification: 27th International Conference, CAV 2015, San Francisco, CA, USA, July 18-24, 2015, Proceedings, Part I 27*, pages 214–231. Springer, 2015.
40. Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Synthesis in pMDPs: A tale of 1001 parameters. In *International Symposium on Automated Technology for Verification and Analysis*, pages 160–176. Springer, 2018.
41. Ernst Moritz Hahn, Tingting Han, and Lijun Zhang. Synthesis for PCTL in parametric Markov decision processes. In *NASA Formal Methods Symposium*, pages 146–161. Springer, 2011.

42. Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In *CAV (2)*, volume 10427 of *Lecture Notes in Computer Science*, pages 592–600. Cham, 2017. Springer.
43. Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, Cambridge, MA, 2008.
44. Thom Badings, Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Scenario-based verification of uncertain parametric MDPs. *International Journal on Software Tools for Technology Transfer*, 24(5):803–819, 2022.
45. Christopher K Wikle and L Mark Berliner. A Bayesian tutorial for data assimilation. *Physica D: Nonlinear Phenomena*, 230(1-2):1–16, 2007.
46. Sebastian Reich and Colin Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, 2015.
47. Dinh Bao Phuong Huynh, David J Knezevic, and Anthony T Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.
48. Jens L Eftang and Anthony T Patera. Port reduction in parametrized component static condensation: approximation and a posteriori error estimation. *International Journal for Numerical Methods in Engineering*, 96(5):269–302, 2013.
49. Kathrin Smetana and Anthony T Patera. Optimal local approximation spaces for component-based static condensation procedures. *SIAM Journal on Scientific Computing*, 38(5):A3318–A3356, 2016.
50. Michael G Kapteyn, David J Knezevic, and Karen Willcox. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*, 2020.
51. Michael G Kapteyn, David J Knezevic, DBP Huynh, Minh Tran, and Karen E Willcox. Data-driven physics-based digital twins via a library of component-based reduced-order models. *International Journal for Numerical Methods in Engineering*, 2020.