ENSURE: Towards Reliable Control of Cyber-Physical Systems under Uncertainty

Wenhua Yang, Chang Xu, Minxue Pan, Yu Zhou, and Zhiqiu Huang

***Abstract*—Cyber-physical systems (CPS) are complex ensembles of physical and cyber components that cooperate to offer dynamic and adaptive functionalities. Uncertainty can arise from a plethora of sources in the entangled components, ranging from the unreliable perception, the non-deterministic action effects, to even the changes in the environment. Existing controlling approaches, such as those using Markov decision process, have limited ability in handling uncertainty. To address the challenge, in this paper, we novelly propose using partially observable Markov decision processes (POMDPs) to model CPS under uncertainty and show that common types of uncertainties can be modelled by partial observations and non-deterministic actions over probabilistic distributions. With POMDPs, strategies that can optimally control CPS are synthesised. We further propose a strategy-wise verification method, which resolves the difficult problem of verifying the entire POMDP, to offer reliable controlling strategies. Experiments on two representative cases of CPS show promising results compared to existing approaches.**

***Index Terms*—Cyber-physical systems; Uncertainty; POMDP.**

I. INTRODUCTION

CYBER-PHYSICAL systems (CPS) are physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core [1]. They have wide applicability in various economically vital domains, including aircraft and air-traffic control, transportation vehicles and intelligent highways, autonomous robotic systems, and manufacturing, to name a few [1], [2]. CPS consist of two main components: physical elements modelling physical components of the systems and physical environments and cyber elements representing the controlling software and communication links. The heterogeneity, which spans the cyber and physical worlds and hardware and software in CPS, makes it an arduous task to design the controlling strategies which need to manage a large variety of system behaviour.

This challenge is exacerbated by uncertainty, which is prevalent in CPS [3]. As a combination of cyber and physical elements, CPS behaviour is characterised to be affected by the uncertainty in the ever-changing physical environment, which makes them often experience uncertainty in interacting with the open and dynamic environment [4]. Meanwhile, CPS consist of heterogeneous physical units such as sensors and actuators, where their imprecision may cause uncertain results. These particular forms of uncertainty in CPS, such as the sensor's perception error, the actuator's deviation when performing an action, or the dynamic change of the environment, put CPS into a state of incomplete knowledge about themselves and the environment. Hence, handling uncertainty in a graceful manner during the real operation of CPS is critical. As the complexity of CPS increases fast in real life, it becomes increasingly challenging to design a reliable control scheme of CPS in the presence of uncertainty.

Regarding the problem of CPS control design, researchers have recently conducted many studies [5]. Although techniques for the control design of CPS have been proposed in different categories, e.g., model-based techniques [6]–[8], contract-based techniques [9], [10], and control theory-based techniques [11], there is not yet a widely-used control design methodology [12]. Furthermore, these studies have not explicitly considered and dealt with the uncertainty in CPS when designing the control strategy. Meanwhile, there is no doubt that uncertainty affects the behaviour of the system. As reported in [13], an uncertain interaction among the control modules and the physical environment led to a safety violation of a DARPA Urban Challenge vehicle. It is an urgent need to design a control strategy that allows CPS to make decisions under uncertainty.

To address these challenges, we propose to use the partially observable Markov decision processes (POMDPs) [14] as the designing language to design the control for CPS under uncertainty. POMDPs extend the well-known planning paradigm—the fully observable Markov decision processes (MDPs) [15] to domains where only partial observability of the current system state is available. Most existing work focus on the policy synthesis [16]–[18] and notion extension [19]–[21] for POMDP. There are a few recent works on CPS using POMDP as the underlying framework to the reinforcement learning of the controlling policy [22]. Our work studies the approach of designing CPS under uncertainty that can effectively specify the system behaviour and the major forms of uncertainty. We leverage the general form of POMDPs as the designing language and propose a designing approach that can specify the major forms of uncertainty in CPS, including the sensing error, the actuator's deviation, and the dynamic change of the environment.

In addition, to develop a comprehensive design approach, we also study the methods of control policy synthesis from

W. Yang (Corresponding author), Y. Zhou, and Z. Huang are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China. W. Yang and Y. Zhou are also with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China. E-mail: {ywh, zhouyu, zqhuang}@nuaa.edu.cn.

C. Xu is with the State Key Laboratory for Novel Software Technology and the Department of Computer Science and Technology of Nanjing University, Nanjing, China E-mail: changxu@nju.edu.cn.

M. Pan is with the State Key Laboratory for Novel Software Technology and the Software Institute of Nanjing University, Nanjing, China. E-mail: mxp@nju.edu.cn.

Manuscript received xxxx xx, 202x; revised xxxx xx, 202x.

the POMDP and the correctness assurance for the POMDP. With a POMDP, we can derive a strategy that maximises the expected total reward. Though it is notoriously hard to synthesise strategies for POMDPs by finding exact solutions [23], several promising new methods (e.g, approximate [24], point-based [25], [26], and Monte Carlo-based [27] methods) have emerged recently, making it possible to effectively and efficiently solve the problem. However, these methods do not provide any formal guarantees for synthesised strategies, while the CPS behaviour needs to obey certain properties, e.g., safety, liveness or specifications that cannot be expressed using the reward functions. It is desirable to have a strategy that is optimal with respect to some reward measures and meanwhile satisfies the given properties. Existing tool PRISM-POMDP [28] can verify a POMDP by approximating the belief space into a fully observable belief MDP but is restricted to small examples. It is still impractical to verify the entire POMDP, especially the general form of POMDP, against the given properties.

We propose to verify the synthesised strategy that actually controls the CPS. The verification of a POMDP strategy has been studied in previous works. However, their methods are all restricted to simple forms of POMDPs that use deterministic observation functions, whereas in the CPS context, a general form of POMDP that uses stochastic observation function is more suitable to model uncertainty. We give an approach that yields an induced Markov chain from a general POMDP with a strategy, so that verification methods can efficiently certify common properties for millions of states [29] for this simpler model. In this way, the complexity of verification is significantly reduced, while we still can learn whether the synthesised strategy satisfies the property.

This paper presents an approach, named ENSURE, for reliable control of cyber-physical systems under uncertainty. ENSURE adopts POMDPs as the designing language to specify CPS, and in its framework, uncertainty is modelled by partial observations and non-deterministic actions over probabilistic distributions. Strategies are synthesised and verified to provide reliable and optimal control for the modelled system. We applied ENSURE to two representative cases of CPS for control design, and the experimental results show that ENSURE provides more reliable control of CPS compared to existing approaches.

Although CPS is an active research area, uncertainty in CPS is still relatively unexplored. The objective of studying uncertainty in CPS is to come up with a system that can tackle the inevitable uncertainty immersed from the interaction between the physical elements and cyber elements. This requires that, as a designing approach, it should explicitly model and embrace uncertainty in the design [30]. However, existing approaches for CPS modelling have limited power in modelling uncertainty, which motivates us to propose using the general form of POMDP. The safety of CPS is critical, while the verification of POMDP is difficult, and this motivates us to study effective verification methods. In summary, the contributions of this paper are:

- We novelly propose to model CPS under uncertainty using POMDP with which a variety of uncertainties can be modelled.
- We propose an approach to verifying the synthesised strategies for the general form of POMDP, which is more efficient than verifying the entire model, to provide reliable control.
- We conducted a series of experiments on typical CPS, and the results show that ENSURE can produce more effective strategies compared to alternative approaches, and meanwhile, provide formal correctness guarantees for the strategies.

The rest of the paper is organised as follows. Section II presents an introduction of uncertainty in CPS with an illustrative example. Section III introduce the system modelling and strategy synthesis, and Section IV presents the strategy verification. Section V describes the experimental evaluation. Section VI discusses the threats to validity. Section VII surveys related work, and Section VIII concludes this paper.

II. UNCERTAINTY IN CPS

The concept of uncertainty has been extensively explored by many scientific disciplines, such as economics, physics, and psychology, and a number of enlightening and pioneering works [3], [31]–[33] can help us better understand the uncertainty. Regarding CPS, as they bring together the discrete and powerful logic of computing to monitor and control the continuous dynamics of physical and engineered systems, the precision of computing must interface with the uncertainty in the physical environment [1]. However, since the physical operating environment of CPS is inherently complex and unpredictable, operating under uncertainty must be baked into the design of CPS [30]. This section does not make an attempt to enumerate all kinds of uncertainties in CPS, but instead, will discuss the generally recognised uncertainties in CPS, including uncertainties due to the dynamic change of the environment, the sensing error, and the actuator's deviation, and particularly, their impact on the CPS control design.

Firstly, due to the design intention and the inherent characteristics, CPS are generally running in a complex and dynamic environment, e.g., the running environment of a typical CPS application—the autonomous robotic system for target tracking and obstacle avoidance. The environment of such a system is not static and can change to a state that is very different from the developer's expectations when designing the system. These dynamic and unexpected environmental changes are uncertain to CPS. Secondly, CPS often consist of heterogeneous physical units (e.g., sensors and actuators) interacting with the physical world, software and humans. Due to both the internal (the limitation and deterioration of sensor measurement precision or sensor failure) and the external (environment condition perturbation or variation) factors, sensors are prone to manifest themselves less precise and more vulnerable to faults than as they are initially designed. It is impossible for CPS to know exactly the degree of the sensor's specific deviation during running. Therefore, the perception is also uncertain. Additionally, the actuator can also introduce uncertainty to CPS since the actuator cannot be guaranteed to be perfect due to its physical characteristics. It is possible that the actuator

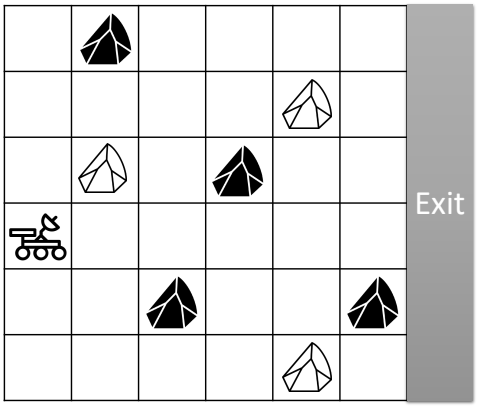


Fig. 1. A *RockSample* instance.

cannot perform the action precisely as expected. For example, when the autonomous robot system controls the robot to move one meter forward, the robot may not be able to move for exact one meter, but 0.9 meters or 1.1 meters instead.

While these uncertainties are present in CPS due to their interaction with the dynamic and open physical environment [34], in the area of CPS, uncertainty is still relatively unexplored [3], [30]. How to explicitly model and deal with uncertainty in the design of cps is even less studied, which is exactly the aim of this work. Uncertainty affects a cyber-physical system's accurate understanding of itself and its running environment. It can put CPS into a state of limited knowledge where it is impossible to exactly describe the existing state. For example, the sensing uncertainty will cause CPS to be unable to accurately understand the actual condition of the physical environment. As a consequence, uncertainty can cause many problems in the running of CPS, even huge losses in safety-critical areas, if not handled properly. The key problem with uncertainty is that it is unavoidable: we cannot eliminate uncertainty from CPS, but only reduce it and seek to manage it intelligently. That is, when a system's state is not fully observed, a decision-maker must act based on what information is available: knowledge of system structure and the history of past decisions and partial state observations. However, in spite of the fact that uncertainty is prevalent in CPS and should be a pivotal concern for decision making, it has not been explicitly considered in the control design of CPS. In the following, we explain through an illustrative example to show how uncertainty affects the running of CPS.

This illustrative example is a variant of the *RockSample* problem that models rover science exploration [35]. Figure 1 depicts the running scenario of the system. A rover explores a terrain, where "scientifically valuable" rocks may be hidden. The rover can achieve reward by sampling rocks in the immediate area, and by reaching the exit at the right side of the map during its traverse. The locations of the rocks and the rover are known during the running. However, it is unknown whether the rocks have the type "good" or "bad". Once the rover moves to the immediate location of a rock, it can sample its type. Sampling a rock is expensive, so the rover is equipped with a noisy long-range sensor that returns an estimate on the type of the rock. The rover can observe the status of the rock with some noise when executing a sensing action. The accuracy of the sensor decreases with increased distance to the rock. The sensing results can be used to help determine whether a rock is good before choosing whether to approach and sample it. Additionally, based on the consideration of energy, while the rover is exploring, we also require that the rover should complete the task within a certain number of steps at a high probability.

An instance of *RockSample* with the terrain being a grid of size $n \times n$ and k rocks is described as $RockSample[n, k]$. The rover can perceive through the equipped sensor to check the type of the rock, move forward in four directions, and sample the rock. Uncertainty inevitably exists in this CPS application, as the long-range sensor always contains unpredictable noise, and may not faithfully reflect the type of the rock. The movement of the rover can also be affected by uncertainty, which is caused by the limitation of the physical actuator nowadays. When the rover is performing a movement to a grid cell, there is a probability that the movement cannot be completed as expected. Due to these uncertainties, the system cannot obtain complete information about its actual situation. It is obvious that the system cannot ignore the existence of these uncertainties and their impact on the system when making decisions. Therefore, we should explicitly consider uncertainty in the control design of CPS.

III. Control Design

The aim of the control design of CPS is to provide a strategy that can decide the actions to be taken by CPS under certain circumstances. ENSURE consists of two components: generating strategies for CPS and verifying the generated strategies. This section introduces the first component of ENSURE.

A. Modelling with POMDPs

As a generalisation of MDPs, POMDPs are more expressive. An MDP provides a mathematical framework for sequential decision making. It can model a system that evolves through states. In each state, one of several actions is chosen, and the system stochastically evolves to a new state based on the current state and the chosen action. In addition, a reward is associated with each state and transition, and the central question is to find a strategy of choosing the actions that maximise the total rewards obtained over the run of the system. MDPs are a widely used model for many kinds of systems' decision making, since they can specify the probability and allow non-deterministic choices [36], to model the uncertainty in the action effects. Take the RockSample for instance. When the rover is moving to a target grid cell, it has a high probability of moving into the expected grid cell and lower probabilities to other nearby grid cells. This kind of uncertainty is supported by MDPs. However, the expressiveness of MDPs is not enough for CPS as there are other complex uncertainties existing in CPS. In the RockSample example, the long-range sensor is imprecise and cannot check the type of rocks for sure. As a result, the rover is unable to determine the current state accurately from the partially observed information. This partial observability is not supported by an MDP since the MDP requires its current state to be determined at any time.

Fortunately, POMDPs are capable of specifying the partial observability and are more general than MDPs in modelling real-world sequential decision processes. Therefore, we propose to model CPS under uncertainty with POMDPs and synthesise the control strategy based on them.

The formal definitions of the models used in this paper are given below. The notations employed in this paper are relatively straightforward. $\mathbb{R}$ denotes the set$(-\infty, +\infty)$. A probability distribution over a finite or countably infinite set X is a function $\mu : X \to [0, 1]$ with $\Sigma_{x\in X}\mu(x) = \mu(X) = 1$. The set of all distributions on X is $Dist(X)$.

Definition 1 (MDP): An MDP is a tuple $\mathcal{M} = (S, s_0, A, \mathcal{T}, \mathcal{L}, \mathcal{R})$, where

- S is a finite non-empty set of states, and $s_0 \in S$ is the initial state;
- A is a finite non-empty set of actions, and the set of actions enabled from state s is denoted by $A(s)$;
- $\mathcal{T} : S \times A \times S \to [0, 1]$ is a transition probability function such that for $s \in S, a \in A$, either $\sum_{s'\in S} \mathcal{T}(s, a, s') = 1$ (a is enabled) or $\sum_{s'\in S} \mathcal{T}(s, a, s') = 0$ (a is disabled), for each $s \in S$ there exists at least one action enabled from s;
- $\mathcal{L} : S \to 2^{AP}$ is a state labelling function, mapping states to a subset of the atomic propositions AP;
- $\mathcal{R} : S \times A \to \mathbb{R}$ is the immediate reward received after transitioning from state $s \in S$ to another state due to action $a \in A$.

MDPs are fully observable, which means their current state is accessible. An MDP $\mathcal{M}$ describes how the state of a system can evolve in discrete time steps through states from the set S. Each state $s \in S$ of $\mathcal{M}$ has a set $A(s)$ of available actions. The choice between which available action is chosen in a state is non-deterministic. In a state s, if action $a \in A(s)$ is selected, then the probability of moving to state s' equals $\mathcal{T}(s, a, s')$. A *strategy* of an MDP resolves the non-deterministic choices for the MDP. It specifies how the action is chosen. Once an action is selected in a state, a transition to a successor state occurs randomly according to the transition probability function. A strategy is *memoryless* if its choices depend only on the current state. POMDPs are an extension of MDPs. In POMDPs, the current state cannot be directly determined and can only be partially observed by strategies that control them.

Definition 2 (POMDP): A POMDP is a tuple $\mathcal{M} = (S, s_0, A, \mathcal{T}, \mathcal{L}, \mathcal{R}, \mathcal{O}, \mathcal{Z})$, where

- $(S, s_0, A, \mathcal{T}, \mathcal{L}, \mathcal{R})$ is an MDP;
- $\mathcal{O}$ is a finite set of *observations*;
- $\mathcal{Z} : S \times A \to Dist(\mathcal{O})$ is an observation function $\mathcal{Z}(s, a, o)$, which defines the probability of observing $o \in \mathcal{O}$ after taking an action $a \in A$ and reaching state $s \in S$.

Above is the general form of POMDPs which are used to model CPS in this paper. To ease the presentation, we assume that the initial state is fully observable. For a POMDP, the actual state is partially observable. This corresponds to the situation where a cyber-physical system has to make decisions based on the incomplete information it observes. A well-designed cyber-physical system can use the incomplete information to make an assumption about the states that it is currently in and acts on its assumption, which seems like the cyber-physical system has a "belief" about its current state. In POMDPs, the concept of *belief* can also be expressed, which is a probability distribution over all possible states $b : S \to [0, 1]$ with $\Sigma_{s\in S}b(s) = 1$. The set of beliefs $\mathcal{B} = \{b : S \to [0, 1]|\Sigma_{s\in S}b(s) = 1\}$ is known as *belief space*. Note that a transition $\mathcal{T}_\mathcal{B}$ in belief space is a deterministic function $b' = \mathcal{T}_\mathcal{B}(b, a, o)$, i.e., given an action $a \in A$ and an observation $o \in \mathcal{O}$, the updates to beliefs are deterministic based on Equation 1:

$$b'(s') = \alpha\mathcal{Z}(s', a, o)\sum_{s\in S}\mathcal{T}(s, a, s')b(s) \tag{1}$$

where α is a normalisation constant.

We denote by $\omega = (b_0, a_1, o_1, b_1, a_2, b_2, \cdots)$ the observation-action sequence in the belief space, such that for all $i > 0$, the belief updates satisfy the transition function $\mathcal{T}_\mathcal{B}$, i.e., $b_i = \mathcal{T}_\mathcal{B}(b_{i-1}, a_i, o_i)$, where $a_i \in A$ is an action and $o_i \in \mathcal{O}$ is an observation. The $(i + 1)$th belief b_i of the observation-action sequence ω is denoted $\omega(i)$ and, if ω is finite, $last(\omega)$ denotes its final belief. The set of all finite observation-action sequences for a POMDP $\mathcal{M}$ is denoted by $ObsSeq_\mathcal{M}$. Because of partial observability, the set of all strategies for a POMDP $\mathcal{M}$ only includes the observation-based strategies. A strategy for a POMDP also resolves the non-deterministic choices in the POMDP by assigning distributions over actions. Formally, an observation-based strategy is defined below.

Definition 3 (Observation-based strategy): A *strategy* of a POMDP $\mathcal{M} = (S, s_0, A, \mathcal{T}, \mathcal{L}, \mathcal{R}, \mathcal{O}, \mathcal{Z})$ is a function $\sigma : \mathcal{B}\to A$ that maps a belief $b \in \mathcal{B}$ to an action $a \in A$. A policy σ defines a set of observation-action sequences $\Omega_\sigma = \{\omega = (b_0, a_1, o_1, \cdots)|\forall i > 0, a_i = \sigma(b_{i-1})$ and $o_i \in \mathcal{O}\}$. For each observation-action sequence $\omega \in \Omega_\sigma$, the action a_i at each step i is chosen by the strategy.

Before we can synthesise strategies to control CPS, we first need to model them. As introduced in Definition 2, a POMDP $\mathcal{M}$ contains eight elements, of which six can be intuitively mapped into CPS for modelling. The states in a POMDP are used to model the states of CPS which are the abstraction of the internal system status and external environment status of the CPS. Actions and observations come into use when the system has a close interaction with the physical environment. The cyber processes monitor and control the physical processes via sensors and actuators, usually with feedback loops where physical processes affect computations and vice versa. Actions in a POMDP are operations that the system can perform, which can be actions performed by the actuator or adjustments within the system. Observations can be mapped to the different kinds of results perceived by the sensors in the system. Propositions are conditions that need to be satisfied and rewards represent the gains during the running of CPS, which can be specified by the designer regarding the specific CPS applications.

The key parts of modelling are the transition probability function $\mathcal{T}$ and the observation function $\mathcal{Z}$ in $\mathcal{M}$. The function $\mathcal{T}$ specifies the probability of the system transitioning to another state $s' \in S$ after performing an action $a \in A$ in one state $s \in S$. The state to which the system transits after

the execution of an action is affected by the effect of the action. In CPS, the effect of an action can be probabilistic due to uncertainty. For instance, an actuator may perform an action in an imperfect way, resulting in that the action effect to be probabilistic. Indeed, specifications of many actuators would state that the actuators have a certain defect rate due to physical limitations of the device. In the following, we present the modelling method. To specify the probability, we can leverage the distribution law of discrete random variables. Specifically, for an action $a \in A$ that can cause probabilistic effect E, the possible effects are e_i $(i = 1, 2, \cdots)$ and the probability that E takes every possible value is $P\{E = e_i\} = p_i, i = 1, 2, \cdots$. According to the distribution law of the action a to be performed and the state s, the transition probability function $\mathcal{T}$ can be constructed: we first determine which state the system will transit to based on state s and action a's effect $E = \{e_1, e_2, \cdots\}$, then assign $P\{E = e_i\}$ to $\mathcal{T}(s, a, s')$, where s' is the transited state due to action a's effect e_i.

The function $\mathcal{Z}$ defines the probability of observing $o \in \mathcal{O}$ while having executed action $a \in A$ and being in state $s \in S$. The observation $\mathcal{O}$ comes from the outputs of equipped sensors in CPS. Often, $o \in \mathcal{O}$ is an incomplete projection of the physical environment contaminated by sensor noise. Therefore, this function can be seen as the sensor model. In general, the sensor's noise can be described by the distribution law or probability distribution. As a result, we can define the observation function through the noise distribution of the sensor. The sensor's perceptual noise is generally given by the manufacturer or can be obtained through a number of experimental statistics and analysis, which has been widely studied in the existing work [37]. If there are multiple sensors, the observation function is determined by their joint distribution.

This modelling method is flexible. Uncertainty can be conveniently considered through the transition probability function and the observation function. Meanwhile, if the uncertainty's specification changes or new uncertainty arises, we need to update these two functions to deal with the uncertainty. For example, the dynamic environmental changes can invalidate part of the sensing results about the environments, which can also change the observations and their probabilities.

Example. We take the $RockSample[n, k]$ problem introduced in Section II for example to explain the modelling. The state space is the cross product of $k + 1$ features: the k binary features $RockType_i = \{Good, Bad\}$ indicating which of the rocks are good, and the one feature of $Position = \{(1, 1), (1, 2), \cdots, (n, n)\}$ indicating the current position of the rover. There is an additional terminal state reached when the rover moves off the right-hand edge of the map. The actions of the rover include $\{East, West, North, South, Sample\}$. The first four are single-step motion actions to the four directions. The $Sample$ action samples the rock at the rover's current location. If the rock is good, the rover receives a reward of 10 and the rock becomes bad (indicating that nothing more can be gained by sampling it). If the rock is bad, it receives a penalty of -10. Moving into the exit area yields a reward of 10. The transition probability function can be defined according to the current state of the rover and the action to be performed. A state indicates the position of the rover, and an action has a defect rate resulting in different possible effects e_i $(i = 1, 2, \dots)$, e.g, moving into the target cell or the cells adjacent to the target. The probability of each e_i is defined by the action's semantics and the defect rate. For example, action $East$ can move the rover to its eastern cell at probability $P\{E = e_1\}$ and can also move the rover to its southern cell at probability $P\{E = e_2\}$ due to $East$'s multiple effects. The observation function models the rover's noisy sensor. It can return a noisy observation from {*Good*, *Bad*} for the rock's type. The noise in the long-range sensor reading is determined by the efficiency η, which decreases exponentially as a function of Euclidean distance from the target. When $\eta = 1$, the sensor always returns the correct value. When $\eta = 0$, it has a 50/50 chance of returning *Good* or *Bad*. Thus, the observation function can be directly defined according to η and the state.

B. Strategy Synthesis

For a POMDP of a cyber-physical system, we can synthesise the strategy to control the system. In this paper, we focus on synthesising memoryless strategies for a POMDP, which means its choices only depend on the current state.

To solve POMDPs exactly is computationally intractable. Recently, point-based algorithms have made impressive progress to large POMDPs by computing approximate solutions close to the optimal ones [25], [26]. Therefore, we use the point-based algorithm SARSOP [26] as the underlying strategy synthesis algorithm. POMDP solving algorithms typically operate in the model's belief space $\mathcal{B}$. A *belief* is a probability distribution over all possible states $b : S \to [0, 1]$ with $\Sigma_{s \in S} b(s) = 1$. The set of beliefs $\mathcal{B} = \{b : S \to [0, 1] | \Sigma_{s \in S} b(s) = 1\}$ is known as belief space. When solving a POMDP, after having taken an action $a \in A$ and observing $o \in \mathcal{O}$, its belief needs to be updated based on $b'(s') = \alpha \mathcal{Z}(s', a, o) \Sigma_{s \in S} \mathcal{T}(s, a, s') b(s)$, where α is a normalization constant. Intuitively, the difficulty of solving POMDPs is due to the "curse of dimensionality": in a POMDP, the belief space has dimensionality equal to $|S|$. Point-based algorithms sample a set of point from $\mathcal{B}$ and use it as an approximate representation of $\mathcal{B}$. Early algorithms sample only the subset $\mathcal{R}(b_0)$ of belief points reachable from a given initial point $b_0 \in \mathcal{B}$. SARSOP improves them by sampling a smaller subset $\mathcal{R}^*(b_0)$, i.e., the subset of belief points reachable from b_0 under optimal sequences of actions. Thus, it only explores relevant regions that can be reached from the initial belief point under optimality conditions. The idea of SARSOP is to compute successive approximations of the subset and converge to it iteratively. Since the targeted subset $\mathcal{R}^*(b_0)$ is unknown in advance, it relies on heuristics and information gathered from earlier samples of $\mathcal{R}(b_0)$ to guide the sampling and improve the sampling distribution over time. Furthermore, by using value function bounds, it tries to avoid sampling in regions that are unlikely to be reachable under any optimal conditions. SARSOP has shown state-of-the-art performance in terms of scalability. Meanwhile, a tool that implements the algorithm is available, and we employ it to synthesise the strategy for our POMDPs.

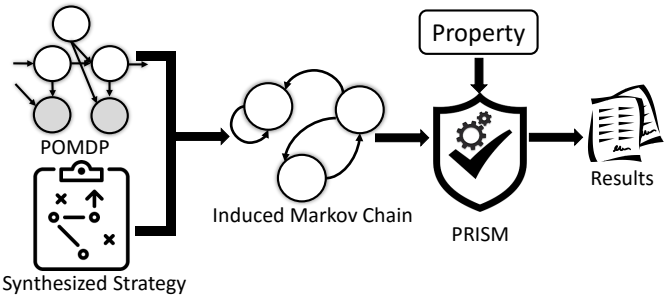


Fig. 2. The overview of strategy verification method.

IV. Strategy Verification

This section introduces the second component of ENSURE, i.e., strategy verification. CPS are often deployed in safety-critical areas such as autonomous vehicles and smart medical devices. It is necessary to guarantee that the strategies satisfy the desired properties. To achieve this, for CPS modelled with POMDPs, a straightforward idea would be to verify the entire POMDP $\mathcal{M}$ to ensure all possible strategies from $\mathcal{M}$ satisfy the property φ, i.e., $\mathcal{M} \models \varphi$. However, the problem is generally undecidable [38], and existing works [28] that study the verification of entire POMDPs are restricted to small examples. A dual problem of verifying an entire POMDP, which is to find a strategy such that the strategy satisfies the given properties, has also been studied, but the current solution does not scale well either [28]. Moreover, they only support the deterministic observation function (i.e., of type $S \rightarrow \mathcal{O}$) and do not handle the stochastic observation function, while modelling real-world CPS requires the most general form of the observation function, i.e., the stochastic observation function of type $S \times A \rightarrow Dist(\mathcal{O})$ as introduced in Definition 2. Therefore, for now, it is still impractical to implement this simple idea into real-world CPS.

We observe that, for the problem of CPS control, we only need one or several feasible strategies, and if we want to determine whether a strategy σ of $\mathcal{M}$ satisfies the given property φ, we can just verify this strategy against φ, rather than verify the entire POMDP $\mathcal{M}$ against φ (i.e., checking if $\mathcal{M} \models \varphi$). This will greatly reduce the complexity of the problem. As [39] shows that a strategy σ for a POMDP $\mathcal{M}$ resolves all non-deterministic choices and partial observability, yielding an induced Markov chain $\mathcal{M}^\sigma$. However, existing works only give the solutions for the simple forms of POMDPs that use deterministic observation functions. We propose our strategy-wise verification method for the general POMDPs that uses stochastic observation function, and the overview is shown in Figure 2. We first transform the cyber-physical system's POMDP $\mathcal{M}$ and its synthesised strategy σ to an induced Markov chain $\mathcal{M}^\sigma$, and the goal turns into checking whether $\mathcal{M}^\sigma \models \varphi$. The induced Markov chain is much simpler than the POMDP, and existing methods can verify the induced Markov chain against common properties efficiently [29]. We employ the model checker PRISM [40] to verify $\mathcal{M}^\sigma$ against the given property φ.

A. Transformation

Given a POMDP modelled for a cyber-physical system and a synthesised strategy, ENSURE uses the strategy to resolve all non-determinism and partial observability in the POMDP, resulting in an induced Markov chain.

Definition 4 (Induced Markov chain): For a POMDP $\mathcal{M} = (S, s_0, A, \mathcal{T}, \mathcal{L}, \mathcal{R}, \mathcal{O}, \mathcal{Z})$ and and a strategy σ, the Markov chain induced by $\mathcal{M}$ and σ is given by $\mathcal{M}^\sigma = (ObsSeq_\mathcal{M}, b_0, P^\sigma)$ where:

$$P^{\omega,\omega'} = \begin{cases} \mathcal{T}_\mathcal{B}(last(\omega), a, b') & \text{if } \omega' = \omega a b' \text{ and} \\ & a = \sigma(last(\omega)) \\ 0 & otherwise \end{cases}$$

A Markov chain is a stochastic model describing a sequence of possible events in which the probability of each event depends only on the state attained in the previous event [41]. The state space of the induced Markov chain is $ObsSeq_\mathcal{M}$, which is the set of finite observation-action sequences for a POMDP $\mathcal{M}$, as introduced in Section III-A, $b_0 \in \mathcal{B}$ is the initial state, and P^σ specifies the transition probability between states.

The transformation from the POMDP $\mathcal{M}$ together with the strategy σ to the induced Markov chain $\mathcal{M}^\sigma$ is as follows. Since we assume that the initial state is fully observable, the initial state b_0 of $\mathcal{M}^\sigma$ is simple, i.e., $b_0(s_0) = 1$ and for other $s' \in S$, $b_0(s') = 0$. Meanwhile, we add $\omega = b_0$ to $ObsSeq_\mathcal{M}$. Then, the strategy σ would map an action $a \in A$ for $\mathcal{M}$ to take. After taking a, $\mathcal{M}$ would perform an observation o. Another belief b_1 will be obtained according to Equation 1 in Section III-A that updates beliefs. Hence, we add $\omega' = b_0, a, o, b_1$ to $ObsSeq_\mathcal{M}$, and the transition probability $P^{\omega,\omega'} = \mathcal{T}_\mathcal{B}(b_0, a, b_1)$. And so forth, we can obtain the induced Markov chain for the corresponding $\mathcal{M}$ and σ. For this finite induced Markov chain, we apply model checking, which in polynomial time reveals whether $\mathcal{M}^\sigma \models \varphi$.

B. Verification

There exist multiple excellent and off-the-shelf model checkers that can efficiently verify Markov chains against common properties. In this paper, we choose PRISM [40] as the underlying probabilistic model checker to verify the induced Markov chain. PRISM provides an easy-to-use language to encode the models to be verified and supports a wide range of quantitative properties.

CPS are often required to operate in partially observable environments. They must reliably execute a specified objective even with incomplete information about the state of the environment. In addition to the need for an optimal control strategy, it is also necessary to ensure that the behaviour of the system under this strategy complies with certain properties. For CPS modelled with POMDPs, one wants to synthesise a strategy such that the probability of satisfying a given property respects a given bound, denoted by $\varphi = \mathbb{P}_{\sim\lambda}(\psi)$, where $\sim \in \{<, \leq, \geq, >\}$, $\lambda \in [0, 1]$, and ψ is a formula expected to hold in all the states. In this paper, we consider safety properties that need to be satisfied by the entire induced

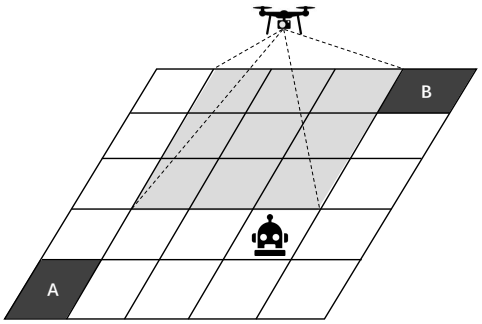


Fig. 3. The problem of Drone surveillance.

Markov chain, and most of these properties can be specified by first-order logic formula.

We first encode the induced Markov chain $\mathcal{M}^\sigma$ and the property φ using the PRISM language. Then, they are fed to PRISM for automatic verification. The result returned by PRISM will tell whether the induced Markov chain satisfies the property or not. If yes, it suggests that the synthesised strategy is guaranteed to satisfy the property.

V. Evaluation

The overarching goal of ENSURE is to synthesise control strategies for CPS in offline scenarios while providing formal guarantees for the strategies. To evaluate ENSURE, we carry out experiments to investigate its *effectiveness* and *performance*. Thus, we explore the following research questions:

- **RQ1:** *Can* ENSURE *synthesise more rewarding strategies compared to other alternatives?*
- **RQ2:** *Can* ENSURE *effectively detect property violations in the synthesised strategies?*
- **RQ3:** *What is the performance of* ENSURE *for synthesising and verifying strategies?*

A. Experimental Subjects

We selected two typical CPS applications: RockSample [35] and Drone surveillance [42], as our experimental subjects. The RockSample problem models rover science exploration, and its running scenario has been introduced in detail in Section II. The problem $RockSample[n, k]$ is easy to extend. By modifying the value of n or k, we will obtain different RockSample problems with different POMDPs. Generally, as the value of n and k increases, so does the complexity of the problem. The drone surveillance problem is inspired by work [42]. An aerial vehicle must survey regions in the corners of a grid-like environment while avoiding a ground agent, as illustrated in Figure 3. The drone and the ground agent moving synchronously and independently in a shared environment, which is divided into a grid of $x \times y$ equally sized square regions. The drone can observe the location of the ground agent only if it is in its field of view delimited by a 3×3 area centred at the drone location. The location of the drone is known during the running. The drone can move in the grid from its current region to any adjacent region in the grid or hover over the current region. The ground agent moves probabilistically in the grid, e.g., moving randomly across the whole grid. The downward-facing camera mounted on the drone provides feedback about the relative position of the ground agent. The camera is not perfect, and cannot determine in which region of the grid the ground agent currently is. It only determines its relative position to the current position of the drone, e.g., in the northeast or southwest of the drone or out of the view.

A POMDP can be constructed for the drone surveillance problem. The state space is the cross product of the location of the drone and the location of the ground agent. There is an additional terminal state, corresponding to the area outside of the grid. The set of actions includes the actions available to the drone. The transition probability function combines the movement of the drone and the motion model of the ground agent. The set of observations consists of the observations transmitted from the camera and an additional observation for the terminal state. With the location of the drone and the observed relative position of the ground agent from the camera, the position of the agent can be estimated with a probability, from which the observation function can be given. If the drone surveys regions A or B for the first time, it receives a reward 10. Moving out of the grid or meeting the ground agent both yield reward -10. The drone surveillance problem is also easy to extend. We can obtain many variants of the problem by changing the size $x \times y$ of the grid. To further assess the synthesised strategies for RockSample and Drone surveillance, we implemented a simulator for each of them.

B. Experimental Design

RQ1: ENSURE is to synthesise control strategies for CPS, and a major purpose is to find an optimal strategy that maximises the total reward obtained over the run of the system. Considering this, the first research question is to evaluate whether ENSURE can synthesise better optimised strategies, by investigating the total reward obtained during the running of the system under the control of those synthesised strategies. To answer RQ1, we compare ENSURE with other alternatives that can control CPS.

Since existing CPS modelling methods that can be used to derive control strategies have not explicitly modelled uncertainty, we take the frequently used approaches–MDP and the Finite-State Machines (FSMs) based method [43] as baselines for comparisons, to show the impacts of uncertainty modelling in control strategy synthesis. MDPs are a principal mathematical framework for sequential decision making. Compared to POMDPs, the current state of an MDP is accessible and thus it does not include observations and the observation function. For the two experimental subjects, we can extract the underlying MDPs of their POMDPs and use classical methods [44] to generate strategies for the MDPs. FSMs are a type of frequently used models in the model-based design—a paradigm for system design in which the design process begins with the creation of models. They have been proposed to model CPS and applied on industrial cases in the field of CPS, particularly in automotive and avionics applications [12]. An FSM is a system with inputs, states, and outputs taking values from discrete sets that are updated at discrete transitions

TABLE I
OBTAINED REWARDS FOR SUBJECTS RUNNING UNDER THE CONTROL OF DIFFERENT MODELS.

Problem	Model	Model Size	Reward
RockSample [4, 5]	POMDP	$\|S\| = 513$, $\|A\| = 5$, $\|\mathcal{O}\| = 2$	30
	MDP	$\|S\| = 513$, $\|A\| = 5$	-90
	FSM	$\|S\| = 17$, $\|T\| = 34$	-10
RockSample [5, 6]	POMDP	$\|S\| = 1,601$, $\|A\| = 5$, $\|\mathcal{O}\| = 2$	40
	MDP	$\|S\| = 1,601$, $\|A\| = 5$	-180
	FSM	$\|S\| = 26$, $\|T\| = 67$	-30
RockSample [6, 7]	POMDP	$\|S\| = 4,609$, $\|A\| = 5$, $\|\mathcal{O}\| = 2$	50
	MDP	$\|S\| = 4,609$, $\|A\| = 5$	-250
	FSM	$\|S\| = 37$, $\|T\| = 82$	-90
RockSample [7, 8]	POMDP	$\|S\| = 12,545$, $\|A\| = 5$, $\|\mathcal{O}\| = 2$	50
	MDP	$\|S\| = 12,545$, $\|A\| = 5$	-290
	FSM	$\|S\| = 50$, $\|T\| = 116$	-130
RockSample [8, 9]	POMDP	$\|S\| = 32,769$, $\|A\| = 5$, $\|\mathcal{O}\| = 2$	60
	MDP	$\|S\| = 32,769$, $\|A\| = 5$	-340
	FSM	$\|S\| = 65$, $\|T\| = 145$	-180
Drone Surveillance [4, 4]	POMDP	$\|S\| = 257$, $\|A\| = 5$, $\|\mathcal{O}\| = 6$	20
	MDP	$\|S\| = 257$, $\|A\| = 5$	-50
	FSM	$\|S\| = 33$, $\|T\| = 82$	0
Drone Surveillance [5, 5]	POMDP	$\|S\| = 626$, $\|A\| = 5$, $\|\mathcal{O}\| = 6$	20
	MDP	$\|S\| = 626$, $\|A\| = 5$	-120
	FSM	$\|S\| = 51$, $\|T\| = 131$	-20
Drone Surveillance [6, 6]	POMDP	$\|S\| = 1,297$, $\|A\| = 5$, $\|\mathcal{O}\| = 6$	20
	MDP	$\|S\| = 1,297$, $\|A\| = 5$	-270
	FSM	$\|S\| = 73$, $\|T\| = 136$	-60
Drone Surveillance [7, 7]	POMDP	$\|S\| = 2,402$, $\|A\| = 5$, $\|\mathcal{O}\| = 6$	20
	MDP	$\|S\| = 2,402$, $\|A\| = 5$	-490
	FSM	$\|S\| = 99$, $\|T\| = 264$	-110
Drone Surveillance [8, 8]	POMDP	$\|S\| = 4,097$, $\|A\| = 5$, $\|\mathcal{O}\| = 6$	10
	MDP	$\|S\| = 4,097$, $\|A\| = 5$	-610
	FSM	$\|S\| = 129$, $\|T\| = 328$	-180

triggered by its inputs. At every transition, the states and the outputs of the FSM are updated. FSMs are suitable for modelling how a system should behave in response to the inputs, which makes it a candidate for controlling CPS. Take our experimental subjects as an example. The conditions of the rover and the drone, plus the status of the environment, can be modelled as the inputs of FSMs; and the actions of the rover and the drone can be modelled as outputs of FSMs. We asked two experts with more than eight years of experiences of using FSMs to devise the FSMs for RockSample and Drone surveillance, respectively.

RQ2: ENSURE provides formal guarantees for the synthesised strategies by verifying them against the given safety properties. If the strategy does not satisfy the property, a *counterexample* will be returned by PRISM. It indicates that the system's behaviour will violate the property if the system executes according to the strategy. Due to the safety-critical nature of CPS, it is vital to know whether the systems' behaviour is reliable. The verification in ENSURE is designed for this purpose. To validate the effectiveness of verification, we use ENSURE to verify its synthesised strategies for the experimental subjects. We propose two properties for each kind of the experimental subjects to be verified, respectively. Due to uncertainty in the system, the properties are all probabilistic. For Rocksample, we require that the rover *sample all good rocks* `and` *complete that within one hundred steps* (denoted as ψ_1), or the rover *sample all good rocks* `and` *not sample any bad rock* `and` *complete that within two hundred steps* (denoted as ψ_2). The two properties to be verified independently are $\varphi_1 = \mathbb{P}_{>0.9}(\psi_1)$ and $\varphi_2 = \mathbb{P}_{>0.95}(\psi_2)$. For drone surveillance, we require that the drone *visit regions A and B* `and` *complete that within one hundred steps* (denoted as ψ_3), or the drone *not visit regions A or B* `or` *encounter the ground agent* `or` *get out of the grid* (denoted as ψ_4). Then, the two properties for drone surveillance to be verified independently are $\varphi_3 = \mathbb{P}_{>0.9}(\psi_3)$ and $\varphi_4 = \mathbb{P}_{\leq 0.05}(\psi_4)$.

If PRISM returns a counterexample for the strategy, it means that ENSURE detects a property violation for the strategy. To further confirm the found problems, we validate them in the simulators of the experimental subjects. Specifically, for a POMDP's synthesised strategy σ that has been found to violate a property φ, we employ the problematic strategy σ to control the running of the system modelled with the POMDP in the simulator. Then, based on the results of the running, we check whether the system really violates the property.

RQ3: POMDPs are difficult to solve exactly and verify entirely. ENSURE proposes to use a point-based method to solve the POMDP approximately and verify a strategy of the POMDP against the property rather than verifying the entire model. Through such a compromise, the difficulty of solving and verifying POMDPs will be significantly reduced. Under these circumstances, we are curious about the performance of ENSURE. To answer RQ3, we evaluate the performance of ENSURE with respect to the strategy synthesis and strategy verification. More comprehensive evaluation results can be obtained by using POMDPs of different levels of complexity. Therefore, we extend the $RockSample[n, k]$ and Drone surveillance problems by modifying the number of n and k and the grid size in Drone surveillance to obtain more variants with different complexity. Afterwards, we apply ENSURE to synthesise strategies for these problems and then verify the synthesised strategies. Meanwhile, we record the time and memory cost for synthesis and verification.

C. Experimental Results

1) RQ1: Reward maximisation in CPS controlling: Three approaches were applied to the experimental subjects for CPS controlling. We synthesised strategies from the POMDPs and MDPs of the subjects and used the strategies to control the experimental subjects' running in the simulators. FSMs can be used for controlling a system's behaviours in response to inputs, and thus we directly employed the experimental subjects' FSMs for controlling their running in the simulators. We recorded the obtained rewards during the subjects' running. The rewards are obtained according to the actions and the state of the system, e.g., sampling a good rock receives a reward of 10. Table I shows the obtained rewards for using different models to control the subjects' running.

Column 1 presents the considered problems of RockSample and Drone Surveillance with different levels of complexity. The type and the size of each model used for CPS controlling are given in Column 2. Specifically, for POMDPs, the numbers of states, actions, and observations are given; for MDPs, the numbers of states and actions are given; and for FSMs, their sizes are measured by the numbers of states and transitions.

TABLE II
TIME (IN SECOND) AND MEMORY (IN MB) COST OF ENSURE.

Problem	Synthesis		Verification	
$\|S\|/\|A\|/\|\mathcal{O}\|/\varphi$	Time	Memory	Time	Memory
RockSample				
$[4,4]$, $257/5/2/\varphi_1$	0.02	126.4	0.01	20.5
$[4,5]$, $513/5/2/\varphi_1$	0.14	184.5	0.05	34.3
$[5,5]$, $801/5/2/\varphi_1$	0.52	214.6	0.09	40.6
$[5,6]$, $1,601/5/2/\varphi_1$	1.6	256.7	0.11	51.8
$[6,6]$, $2,305/5/2/\varphi_1$	3.8	313.9	0.29	57.1
$[6,7]$, $4,609/5/2/\varphi_2$	6.1	421.1	1.20	62.9
$[7,7]$, $6,273/5/2/\varphi_2$	11.2	501.3	2.55	67.4
$[7,8]$, $12,545/5/2/\varphi_2$	24.5	542.1	3.86	72.6
$[8,8]$, $16,385/5/2/\varphi_2$	30.3	604.7	4.98	78.3
$[8,9]$, $32,769/5/2/\varphi_2$	53.6	659.0	6.28	81.0
Drone Surveillance				
$[4,4]$, $257/5/6/\varphi_3$	18.8	207.4	0.02	45.2
$[5,5]$, $626/5/6/\varphi_3$	34.2	290.6	0.08	57.4
$[6,6]$, $1,297/5/6/\varphi_3$	52.9	464,3	0.19	63.3
$[7,7]$, $2,402/5/6/\varphi_3$	90.1	510.5	0.41	67.8
$[8,8]$, $4,097/5/6/\varphi_3$	176.3	604.1	1.08	71.5
$[9,9]$, $6,562/5/6/\varphi_4$	212.4	675.0	1.62	86.5
$[10,10]$, $10,001/5/6/\varphi_4$	357.6	732.7	3.01	92.2
$[11,11]$, $14,642/5/6/\varphi_4$	409.1	864.6	5.55	99.1
$[12,12]$, $20,737/5/6/\varphi_4$	688.5	983.5	8.57	104.6
$[13,13]$, $28,562/5/6/\varphi_4$	997.8	1033.8	12.14	117.2

Column 3 presents the obtained rewards of using different models to control CPS. *From Table I we can see that when under the control of strategies synthesised from POMDPs, all the experimental subjects obtain the largest rewards.* Additionally, the results of POMDPs are significantly superior to the ones of MDPs and FSMs in terms of the obtained rewards. FSMs come second, with MDPs ranked the last.

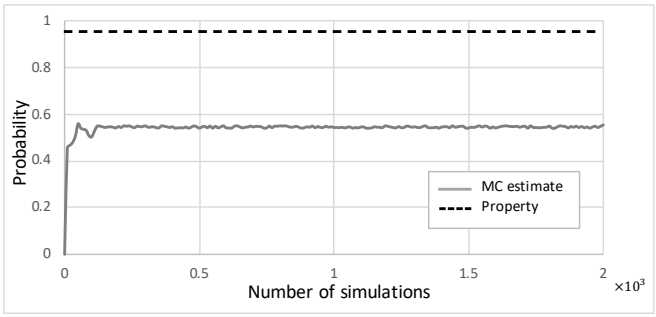
The reason for such results is that uncertainty existing in CPS can cause non-determinism and partial observability that affect the systems' running. Using POMDPs to model CPS enables the explicit consideration of uncertainty. Hence, as demonstrated by the results, the largest reward for each problem was obtained by using POMDPs. We observed from the simulation that because MDPs totally ignore the partial observability in CPS, they often made incorrect judgements of the current state, resulting in inferior decision makings. This, therefore, causes their results even worse than FSMs, which rely on the current actual inputs to react. These experimental results clearly indicate the importance of considering uncertainty in CPS modelling and controlling.

2) RQ2: Effectiveness of strategy verification: We applied the verification function in ENSURE on different strategies for the experimental subjects of different sizes. We found 4 counterexamples, which were from $RockSample[7,8]$ and φ_2, $RockSample[8,9]$ and φ_2, Drone Surveillance $[8,8]$ and φ_3, and Drone Surveillance $[8,8]$ and φ_4, respectively. We empirically validated all the synthesised strategies provided by ENSURE against the respective properties that need to be satisfied in simulators, which accounts to 20 cases (10 strategies*2 properties) validated. Since the properties are about probabilities, to reduce the bias of randomness, we have simulated the running of the systems for up to 2,000 times when checking whether the property was violated or not under the control of the strategies. Then the probabilities were estimated using a Monte Carlo estimator. *The results show that* ENSURE *has detected all the property-violation strategies, and all the detected violations are real ones.* Figure 4 shows the simulation results for the four counterexamples that failed to satisfy the property. The dotted line in Figure 4 (a) specifies the minimum probability (0.95) required by the property. However, the actual probability estimated from the subject's running in simulation is less than the minimum probability. Therefore, the property is violated. The situation in Figure 4 (b) and Figure 4 (c) is similar to the situation described in Figure 4 (a). In Figure 4 (d), the property requires the probability should be less than 0.05, while the actual estimated probability is larger than 0.05. It indicates that the systems' behaviours cannot satisfy the properties under the control of these synthesised strategies. *This confirms that* ENSURE *can effectively identify property-violation problems in the synthesised strategies.*

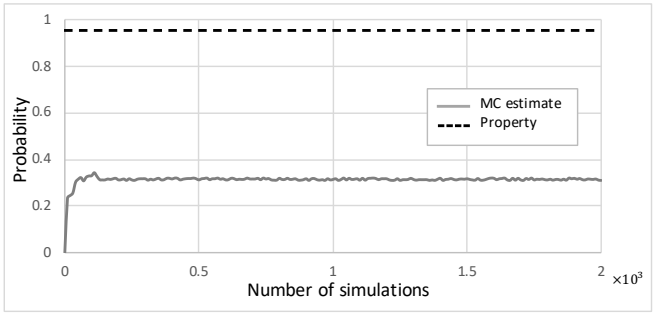
3) RQ3: Performance: The results of time and memory cost for the synthesis and verification components in ENSURE are illustrated in Table II. The experiments are run on a 3.2 GHz machine with a 32 GB memory. Column 1 gives the details of the POMDPs used for experiments, including their model size and the properties used in verification. We conducted experiments on 10 RockSample problems and 10 Drone Surveillance problems with different levels of difficulty. Columns 2 and 3 show the time and memory cost for synthesis and Columns 4 and 5 are for verification. As we can see, the cost of synthesis and verification is very low for small-size models, and as the difficulty of the problem increases, the cost increases proportionally, which indicates the scalability of the approach. Even for the most complex RockSample problem and the Drone Surveillance problem in the experiments where the numbers of states have reached 32,769 and 28,562, respectively, ENSURE can still complete within 1 minute and 17 minutes on an ordinary desktop computer. Such a cost is acceptable, especially considering that the synthesis and verification are mostly conducted offline before the CPS are deployed in real environments.

VI. THREATS TO VALIDITY

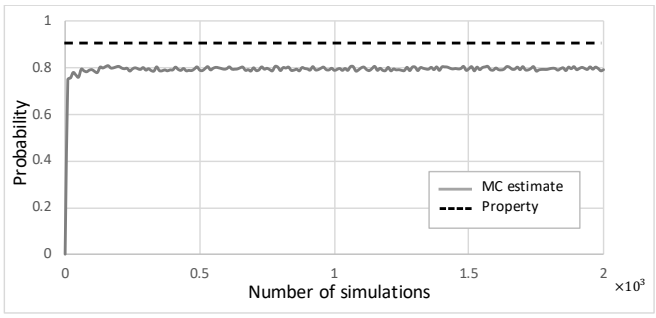
In this section, we discuss the threats to validity. First, threat can occur to the strategy synthesis algorithm for solving POMDPs, as the solution provided by the algorithm is approximate. Due to the complexity of the problem, the algorithm cannot compute the solution exactly. However, the approximation is also a commonly adopted practice for solving POMDPs. To address this concern, we employ the algorithm that has shown state-of-the-art performance for solving POMDPs. Though the synthesised strategy is not guaranteed to be a globally optimal strategy, it could be the optimal we can obtain for now. The second threat is that we verify the strategy rather than the entire system. The problem of verifying an entire POMDP is generally undecidable. To reduce the complexity, we propose to verify a strategy that actually controls the CPS rather than the entire POMDP, and transform the strategy with the POMDP to an induced Markov chain that can be efficiently verified. This is a trade-off. We do not verify all possible strategies and choose to verify the synthesised



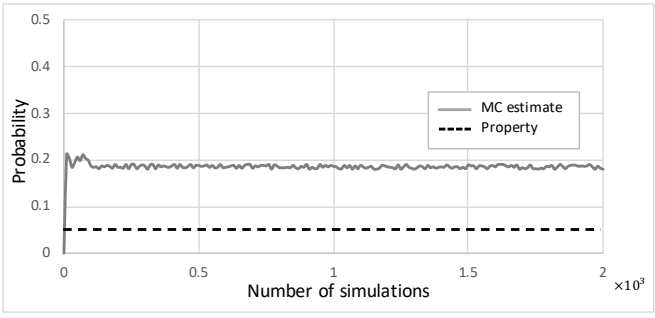
(a) $RockSample[7,8]$, $\varphi_2 = \mathbb{P}_{>0.95}(\psi_2)$



(b) $RockSample[8,9]$, $\varphi_2 = \mathbb{P}_{>0.95}(\psi_2)$



(c) *Drone Surveillance* $[8,8]$, $\varphi_3 = \mathbb{P}_{>0.9}(\psi_3)$



(d) *Drone Surveillance* $[8,8]$, $\varphi_4 = \mathbb{P}_{\leq 0.05}(\psi_4)$

Fig. 4. Comparisons of the actual estimated probabilities and the required probabilities of properties.

optimal strategy. However, it is possible for our approach to verify other strategies when needed, e.g., to verify several backup strategies, by simply eliciting more strategies using the strategy synthesis component. The third threat concerns the modelling of POMDP. Modelling is essential for model-based system control approaches. While the modelling requires considerable manual effort, the benefit overweights the effort, since the strategy synthesis and verification can be conducted automatically and have formal guarantees. We aim to reduce the manual effort by proposing the general form of POMDPs with which the partial observation can be more straightforwardly modelled compared to the simple POMDPs and other modelling approaches for CPS, e.g., hybrid automata and contracts. The fourth threat concern the generalisation of our conclusions, which may not generalise to other CPS. To reduce this threat, we selected two representative CPS applications from existing literature and generated many variants for them by increasing the level of difficulty to conduct the experiments. The results consistently support our conclusions, and we try to make them applicable to other CPS applications. Still, there is a need for evaluating our approach with other CPS applications.

VII. Related Work

In this section, we briefly review the literature on uncertainty handling, control design, and verification for CPS. The literature on these topics of research is extensive. We focus our discussion here to those works that are of utmost relevance.

A. Uncertainty Handling in CPS

Researchers have conducted lots of research on the uncertainty in various kinds of software systems [31]–[33]. To understand uncertainty in the context of CPS, Zhang et al. propose a conceptual model that is mapped to three logical levels of CPS, i.e., application, infrastructure, and integration [3]. There are many sources of uncertainty in CPS, e.g., unreliable sensing, imperfect actuators, unstable networks, unpredictable environmental changes, and human in CPS. While some of them are prevalent in many different systems, uncertainty immersed from the interaction between the physical and cyber worlds has a particular place in CPS and is investigated in this work, since serious vulnerabilities are often created by the interactions between the physical and the cyber sides [45]. To further characterise uncertainty in CPS, researchers propose several uncertainty modelling methods. Work [46] depicts uncertainty in CPS with SysML requirements diagrams for tracing uncertainty information through artefacts. A probabilistic extension of clock constraint specification language is proposed for specifying the uncertain occupant behaviours in CPS [47]. Because of the prevalent existence of uncertainty in CPS, unless properly handled, the uncertainty could affect systems' running and potentially cause fatal consequences. One option to discover these problems is by focusing on testing uncertainties' impact on CPS. To support model-based testing of CPS facing uncertainty, work [48] proposes a framework to create test ready models for capturing the expected behaviour of CPS in the presence of uncertainty. Study [49] pays special attention to testing self-healing behaviours of CPS under uncertainty. To test whether self-healing behaviours of CPS can correctly heal faults under uncertainties to ensure their reliability, it learns the optimal policies for invoking

operations and introducing uncertainties from test executions. Despite the importance of studying uncertainty, it is still an emerging topic in CPS area. Existing literature mainly focuses on modelling and characterising uncertainty in CPS, or testing the behaviour of CPS under the influence of uncertainty. There is little work considering how to deal with uncertainty at the control design stage of CPS, i.e., designing the control for CPS that can cope with uncertainty. This motivates our work.

B. Control Design for CPS

CPS is an interdisciplinary area involving computer science, control, and automation, etc. Each discipline has its own focus on CPS research. As a kind of software-intensive system, CPS have attracted much attention of the software engineering community, especially on the ways to engineering CPS and the quality assurance of CPS. This echoes the motivation of this paper. We aim to devise a reliable control scheme that takes uncertainty into account for CPS. Speaking of the control design of CPS, there is a series of related work. In general, there are several types of common design methods. Model-based design is one of them [6]. It seeks to place an emphasis on abstract, mathematical modelling as a first step before getting into low-level details of the implementation. Various models are leveraged for the control design of CPS, e.g., functional models [8] and hybrid automata [7]. Model-based tools such as Simulink [50] and Modelica [51] are popular in the design of CPS for modelling the heterogeneous subsystems. Our work proposes to use the general form of POMDP for the model-based CPS design. A similar modelling language, MDP, has been frequently used in modelling CPS [52]. Besides the fact that MDP is less effective in modelling uncertainty in CPS, they are often used in a distinctive way: MDP is to describe the process of reinforcement learning of the control policy. Differently, our work falls in the domain of model-based design, which aims to use POMDP as the designing language to specify CPS and proposes an approach to explicitly modelling uncertainty. Contract-based design uses contracts to specify and abstract the components of CPS [9], [10]. Contracts are specifications of the conditions for correctness of element integration, for a lower level of abstraction to be consistent with the higher ones, and for abstractions of available components to be faithful representations of the actual parts. Contract-based design is carried out as a sequence of refinement steps from a high-level specification to an implementation built out of a library of components at the lower level. Some control theory techniques are also applied for the control design of CPS, e.g., a PID controller is employed for controlling a cyber-physical home system [11]. Recently, researchers have leveraged CEGIS (Counterexample Guided Inductive Synthesis) [53] based on SMT solvers for the synthesis of controllers. Our approach differs significantly from the CEGIS-based approaches [54], [55] in terms of their objectives, which results in different implementations. Our approach focuses on handling uncertainty in CPS control and thus takes the POMDP approach. CEGIS-based approaches, on the other hand, do not target uncertainty but synthesise candidate solutions based on iteratively added counterexamples. For more details on the control design of CPS, we refer the interested readers to [5], [12]. The above methods provide impressive support for the control design of CPS. However, they lack explicit consideration of uncertainty in the design process. Our approach, falling in the category of model-based design, takes uncertainty in CPS into account through partially observable Markov decision processes.

C. Verification of CPS

CPS are expected to meet a multitude of constraints, e.g., safety constraints, timing constraints, energy consumption, memory usage, and often with uncertain environment or occupant behaviour. Currently, the verification of CPS remains a challenging problem. To alleviate this problem, lots of studies have been carried out. For example, Thacker et al. propose to model CPS with the labelled hybrid Petri-net and remove the details from the model that are irrelevant to the property of interest for verification [56]. Complete verification of CPS is often impractical due to the complexity of the systems and the use of heterogeneous formalism for modelling different components. Work [57] introduces behaviour relations to relate the different semantic formalism and presents structured constructs of nested conjunctive and disjunctive analyses to enable verification of system-level properties from the properties of heterogeneous models. In terms of the verified property, it involves safety [58], robustness [59], security [60] and temporal correctness [61], [62], etc. Technically, model checking is the most used verification algorithm for CPS. Various model checkers, e.g., UPPAAL and SPIN, have been employed for verifying CPS in existing works [62], [63]. The purpose of verifying CPS is to assure that CPS satisfy the given property under all possible conditions. However, it is arduous to verify the entire CPS considering the complexity of the system and environment. For example, verifying CPS based on hybrid automata cannot scale to large systems since it takes hours to verify a hybrid automaton with dozens of states [64]. Meanwhile, a particular challenge for the verification of CPS is the presence of uncertainty which increases the state space to be verified. Different from these works, our approach seeks to a strategy for CPS modelled with POMDPs that explicitly consider uncertainty, and verifies the strategy rather than the entire POMDP to reduce the complexity of verification.

VIII. Conclusion and Future Work

In this paper, we present ENSURE, an approach to synthesising and verifying control strategies for CPS. To support the controlling of CPS in the presence of uncertainty, ENSURE novelly proposes to model CPS with the general form of POMDPs, which can specify the partial observability and non-determinism caused by uncertainty in CPS. Based on the POMDPs, an optimised strategy for controlling CPS is synthesised. To guarantee that the strategy satisfies the given properties, ENSURE uses a strategy-wise checking method that can efficiently conduct the verification. The experimental results on two typical CPS applications have shown the effectiveness of ENSURE in delivering reliable controlling strategies. The contribution not only lies in the proposed approach but also in the entire framework that integrated all the

methods together to address the technical challenge of using model-based approaches for reliable controlling of CPS under uncertainty.

ENSURE verifies the synthesised strategy to answer whether it satisfies the property or not. If the verification results indicate that the strategy fails to satisfy the property, currently ENSURE reports the results, with which the designers can decide whether or not to discard the synthesised strategy. In the future work, we are going to investigate whether the verification results can be further utilised to help ENSURE to generate other optimised strategies that can satisfy the property. Specially, we will study how to leverage counterexamples to refine the models. On the other hand, we are working with our partners to apply our approach to industrial cases.

ACKNOWLEDGMENT

This work was supported by the Leading-edge Technology Program of Jiangsu Natural Science Foundation (No. BK20202001), the National Natural Science Foundation of China (Nos. 61972193, 61932021, 61972197), the Fundamental Research Funds for the Central Universities (No. NS2021069), the Natural Science Foundation of Jiangsu Province (No. BK20201292), and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Design Automation Conference*, 2010, pp. 731–736.

[2] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, 2008, pp. 363–369.

[3] M. Zhang, B. Selic, S. Ali, T. Yue, O. Okariz, and R. Norgren, "Understanding uncertainty in cyber-physical systems: A conceptual model," in *Modelling Foundations and Applications*. Cham: Springer, 2016, pp. 247–264.

[4] W. Yang, C. Xu, M. Pan, X. Ma, and J. Lu, "Improving verification accuracy of cps by modeling and calibrating interaction uncertainty," *ACM Trans. Internet Technol.*, vol. 18, no. 2, jan 2018. [Online]. Available: https://doi.org/10.1145/3093894

[5] S. K. Khaitan and J. D. McCalley, "Design techniques and applications of cyberphysical systems: A survey," *IEEE Systems Journal*, vol. 9, no. 2, pp. 350–365, 2015.

[6] J. C. Jensen, D. H. Chang, and E. A. Lee, "A model-based design methodology for cyber-physical systems," in *2011 7th International Wireless Communications and Mobile Computing Conference*, 2011, pp. 1666–1671.

[7] A. Banerjee and S. K. S. Gupta, "Spatio-temporal hybrid automata for safe cyber-physical systems: A medical case study," in *2013 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2013, pp. 71–80.

[8] J. Wan, A. Canedo, and M. A. Al Faruque, "Functional model-based design methodology for automotive cyber-physical systems," *IEEE Systems Journal*, vol. 11, no. 4, pp. 2028–2039, 2017.

[9] A. Sangiovanni-Vincentelli, W. Damm, and R. Passerone, "Taming dr. frankenstein: Contract-based design for cyber-physical systems*," *European Journal of Control*, vol. 18, no. 3, pp. 217 – 238, 2012.

[10] P. Nuzzo, A. L. Sangiovanni-Vincentelli, D. Bresolin, L. Geretti, and T. Villa, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2104–2132, 2015.

[11] W. W. Shein, Y. Tan, and A. O. Lim, "Pid controller for temperature control with multiple actuators in cyber-physical home system," in *2012 15th International Conference on Network-Based Information Systems*, 2012, pp. 423–428.

[12] S. A. Seshia, S. Hu, W. Li, and Q. Zhu, "Design automation of cyber-physical systems: Challenges, advances, and opportunities," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 9, pp. 1421–1434, 2017.

[13] S. Mitra, T. Wongpiromsarn, and R. M. Murray, "Verifying cyber-physical interactions in safety-critical systems," *IEEE Security Privacy*, vol. 11, no. 4, pp. 28–37, 2013.

[14] E. J. Sondik, "The optimal control of partially observable markov processes over the infinite horizon: Discounted costs," *Oper. Res.*, vol. 26, no. 2, pp. 282–304, Apr. 1978.

[15] R. A. Howard, "Dynamic programming and markov processes." 1960.

[16] D. Hadfield-Menell, E. Groshev, R. Chitnis, and P. Abbeel, "Modular task and motion planning in belief space," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 4991–4998.

[17] K. He, M. Lahijanian, L. E. Kavraki, and M. Y. Vardi, "Towards manipulation planning with temporal logic specifications," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 346–352.

[18] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 639–646.

[19] P. Poupart, A. Malhotra, P. Pei, K.-E. Kim, B. Goh, and M. Bowling, "Approximate linear programming for constrained partially observable markov decision processes," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 29, no. 1, Mar. 2015. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/9655

[20] P. Rodrigues Quemel e Assis Santana, S. Thiebaux, and B. Williams, "Rao*: An algorithm for chance-constrained pomdp's," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10423

[21] P. Hou, W. Yeoh, and P. Varakantham, "Solving risk-sensitive pomdps with and without cost observations," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10402

[22] G. Yoo, M. Yoo, I. Yeom, and H. Woo, "rocorl: Transferable reinforcement learning-based robust control for cyber-physical systems with limited data updates," *IEEE Access*, vol. 8, pp. 225 370–225 383, 2020.

[23] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra, "Solving pomdps by searching the space of finite policies," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 417–426.

[24] M. Hauskrecht, "Value-function approximations for partially observable markov decision processes," *J. Artif. Int. Res.*, vol. 13, no. 1, pp. 33–94, Aug. 2000.

[25] J. Pineau, G. Gordon, and S. Thrun, "Point-based value iteration: An anytime algorithm for pomdps," in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, ser. IJCAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 1025–1030.

[26] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces," in *In Proc. Robotics: Science and Systems*, 2008.

[27] D. Silver and J. Veness, "Monte-carlo planning in large pomdps," in *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'10. Red Hook, NY, USA: Curran Associates Inc., 2010, pp. 2164–2172.

[28] G. Norman, D. Parker, and X. Zou, "Verification and control of partially observable probabilistic systems," *Real-Time Systems*, vol. 53, no. 3, pp. 354–402, 2017.

[29] C. Baier and J.-P. Katoen, *Principles of model checking*. MIT press, 2008.

[30] A. Chatterjee and H. Reza, "Toward modeling and verification of uncertainty in cyber-physical systems," in *2020 IEEE International Conference on Electro Information Technology (EIT)*, 2020, pp. 568–576.

[31] B. H. Cheng, P. Sawyer, N. Bencomo, and J. Whittle, "A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty," in *Proceedings of the 12th International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 468–483.

[32] A. J. Ramirez, A. C. Jensen, and B. H. C. Cheng, "A taxonomy of uncertainty for dynamically adaptive systems," in *2012 7th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, 2012, pp. 99–108.

[33] N. Esfahani and S. Malek, *Uncertainty in Self-Adaptive Software Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 214–238.

[34] M. N. Asmat, S. U. R. Khan, and S. Hussain, "Uncertainty handling in cyber–physical systems: State-of-the-art approaches, tools, causes, and future directions," *Journal of Software: Evolution and Process*, vol. n/a, no. n/a, p. e2428. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/smr.2428

[35] T. Smith and R. Simmons, "Heuristic search value iteration for pomdps," in *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*. Arlington, Virginia, USA: AUAI Press, 2004, pp. 520–527.

[36] J. Filar and K. Vrieze, *Competitive Markov Decision Processes*. Berlin, Heidelberg: Springer-Verlag, 1996.

[37] K. Jerath, S. Brennan, and C. Lagoa, "Bridging the gap between sensor noise modeling and sensor characterization," *Measurement*, vol. 116, pp. 350 – 366, 2018.

[38] O. Madani, S. Hanks, and A. Condon, "On the undecidability of probabilistic planning and related stochastic optimization problems," *Artificial Intelligence*, vol. 147, no. 1, pp. 5 – 34, 2003, planning with Uncertainty and Incomplete Information.

[39] S. Junges, N. Jansen, R. Wimmer, T. Quatmann, L. Winterer, J.-P. Katoen, and B. Becker, "Finite-state controllers of pomdps via parameter synthesis," in *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*. AUAI Press, 2018, pp. 519–529.

[40] M. Kwiatkowska, G. Norman, and D. Parker, "Prism 4.0: Verification of probabilistic real-time systems," in *Computer Aided Verification*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 585–591.

[41] P. Gagniuc, *Markov Chains: From Theory to Implementation and Experimentation*. John Wiley & Sons, 2017.

[42] M. Svoreňová, M. Chmelík, K. Leahy, H. F. Eniser, K. Chatterjee, I. Černá, and C. Belta, "Temporal logic motion planning using pomdps with parity objectives: Case study paper," in *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 233–238.

[43] D. B. Rawat, J. J. P. C. Rodrigues, and I. Stojmenovic, *Cyber-Physical Systems: From Theory to Practice*. USA: CRC Press, Inc., 2015.

[44] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 1994.

[45] E. A. Lee, "CPS foundations," in *Design Automation Conference*, 2010, pp. 737–742.

[46] T. Bandyszak, M. Daun, B. Tenbergen, P. Kuhs, S. Wolf, and T. Weyer, "Orthogonal uncertainty modeling in the engineering of cyber-physical systems," *IEEE Transactions on Automation Science and Engineering*, pp. 1–16, 2020.

[47] D. Du, P. Huang, K. Jiang, and F. Mallet, "pcssl: A stochastic extension to marte/ccsl for modeling uncertainty in cyber physical systems," *Science of Computer Programming*, vol. 166, pp. 71 – 88, 2018.

[48] M. Zhang, S. Ali, T. Yue, R. Norgren, and O. Okariz, "Uncertainty-wise cyber-physical system test modeling," *Softw. Syst. Model.*, vol. 18, no. 2, pp. 1379–1418, Apr. 2019.

[49] T. Ma, S. Ali, T. Yue, and M. Elaasar, "Testing self-healing cyber-physical systems under uncertainty: a fragility-oriented approach," *Software Quality Journal*, vol. 27, no. 2, pp. 615–649, 2019.

[50] T. M. Inc, "Simulink," http://www.mathworks.com/products/simulink, 2020, [Online].

[51] M. Association, "Modelica," https://www.modelica.org, 2020, [Online].

[52] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for uav attitude control," *ACM Trans. Cyber-Phys. Syst.*, vol. 3, no. 2, feb 2019. [Online]. Available: https://doi.org/10.1145/3301273

[53] A. Solar-Lezama, L. Tancau, R. Bodik, S. Seshia, and V. Saraswat, "Combinatorial sketching for finite programs," *SIGARCH Comput. Archit. News*, vol. 34, no. 5, pp. 404–415, Oct. 2006. [Online]. Available: https://doi.org/10.1145/1168919.1168907

[54] H. Ravanbakhsh and S. Sankaranarayanan, "Counter-example guided synthesis of control lyapunov functions for switched systems," in *2015 54th IEEE Conference on Decision and Control (CDC)*, 2015, pp. 4232–4239.

[55] ——, "Robust controller synthesis of switched systems using counterexample guided framework," in *2016 International Conference on Embedded Software (EMSOFT)*, 2016, pp. 1–10.

[56] R. A. Thacker, K. R. Jones, C. J. Myers, and H. Zheng, "Automatic abstraction for verification of cyber-physical systems," in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '10. New York, NY, USA: ACM, 2010, pp. 12–21.

[57] A. Rajhans and B. H. Krogh, "Heterogeneous verification of cyber-physical systems using behavior relations," in *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, ser. HSCC '12. New York, NY, USA: ACM, 2012, pp. 35–44.

[58] H. Geuvers, A. Koprowski, D. Synek, and E. van der Weegen, "Automated machine-checked hybrid system safety proofs," in *Interactive Theorem Proving: First International Conference, ITP 2010, Edinburgh, UK, July 11-14, 2010. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 259–274.

[59] P. Tabuada, S. Y. Caliskan, M. Rungger, and R. Majumdar, "Towards robustness for cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3151–3163, Dec 2014.

[60] R. Akella, H. Tang, and B. M. McMillin, "Analysis of information flow security in cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 3, no. 3, pp. 157 – 173, 2010.

[61] L. Guo, Q. Zhu, P. Nuzzo, R. Passerone, A. Sangiovanni-Vincentelli, and E. A. Lee, "Metronomy: A function-architecture co-simulation framework for timing verification of cyber-physical systems," in *2014 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, 2014, pp. 1–10.

[62] S. Balasubramaniyan, S. Srinivasan, F. Buonopane, B. Subathra, J. Vain, and S. Ramaswamy, "Design and verification of cyber-physical systems using truetime, evolutionary optimization and uppaal," *Microprocessors and Microsystems*, vol. 42, pp. 37 – 48, 2016.

[63] Y. Sun, B. McMillin, X. F. Liu, and D. Cape, "Verifying noninterference in a cyber-physical system the advanced electric power grid," in *Proceedings of the Seventh International Conference on Quality Software*, ser. QSIC '07. USA: IEEE Computer Society, 2007, pp. 363–369.

[64] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi, "Hytech: A model checker for hybrid systems," *Int. J. Softw. Tools Technol. Transf.*, vol. 1, no. 1–2, pp. 110–122, Dec. 1997.