



# Performance modeling and quantitative evaluation for cyber-physical systems based on LTS

Zhen Li<sup>1,2</sup> · Zining Cao<sup>1,2,3,4</sup> · Chao Xing<sup>1</sup>

Accepted: 12 September 2023 / Published online: 3 October 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Cyber-physical systems (CPS) are complex systems that can be understood as a hybrid system, a real-time system with probabilistic behavior, and a concurrent system. With the increasing use of CPS, there is a growing demand for higher reliability and performance. Additionally, the large amount of information in CPS requires processing, exacerbating the state space explosion problem caused by model checking methods used for the quantitative evaluation of CPS. Therefore, evaluating the functionality, performance, and reliability of CPS is not only an important research topic but also an inevitable challenge. This paper aims to establish a comprehensive and efficient CPS performance model and a quantitative evaluation method and proposes solutions to existing problems based on the specific practical application environment of the system. First, we extend the labeled transition system (LTS) and propose a hybrid probability time cost transition system (HPTCTS) that provides a detailed description of the continuous behavior, probabilistic behavior, time characteristics, and performance of CPS. Furthermore, we propose HPTCTS temporal logic (HPTCTS-TL) to describe the performance characteristic of HPTCTS. To alleviate the state space explosion problem when quantitatively evaluating the HPTCTS model, we propose the corresponding quantitative evaluation algorithm based on symbolic model checking. Finally, we discuss a typical example of CPS to demonstrate the feasibility of our approach. Overall, our work contributes to a better understanding of CPS and provides a more effective and comprehensive way to evaluate CPS performance, which is crucial for the successful development and deployment of CPS in real-world applications.

**Keywords** Cyber-physical systems · Performance modeling · Quantitative evaluation · LTS · Symbolic model checking

# 1 Introduction

## 1.1 Context and motivation

As one of the pivotal technologies driving Industry 4.0, cyber-physical systems (CPS) have become indispensable in safety-critical domains, including aerospace, automotive, smart homes, medical equipment, and military applications [1–3]. The designers and developers of CPS have introduced numerous innovative advancements that lay the groundwork for future industrial technologies, continuously pushing the boundaries of innovation. However, with the increased interaction between cyber systems with discrete behavior and physical systems with continuous behavior, coupled with the exponential growth in size, complexity, and cost of CPS, they have become imperative to ensure that they satisfy critical properties such as correctness, reliability, security, and performance [4, 5]. Therefore, it is essential to rigorously characterize the functions and properties of CPS and verify them using formal methods.

Formal methods have the rigorous mathematical specification and theoretical basis and they are very suitable for modeling and verifying safety-critical systems such as CPS. The existing formal models (automata and their extensions, process algebra, LTS) all provide good support for formal verification techniques (model checking, mutual simulation, theorem proving.) [6–8]. However, a CPS is an intricate system that not only involves communication between cyber and physical systems but also comprises complex characteristics like real-time behavior and probabilistic behavior. Thus, CPS exhibit various features such as discrete behavior, continuous behavior, temporal characteristics, uncertainty, and randomness. At the same time, a running CPS has performance requirements in terms of cost (energy consumption, number of successfully delivered and discarded message packets.) [9, 10]. Although existing work is devoted to modeling CPS using formal methods, they still cannot fully describe the above CPS and have some limitations. Our work aims to model CPS using formal methods considering the characteristics of CPS in a comprehensive manner.

We believe that labeled transition system (LTS) [11–13] is a promising approach for formal modeling of CPS. It provides a concise representation of the target system's behavior and a powerful ability to intuitively depict system state transitions. Additionally, LTS can function as a semantic model of process algebra [14] to describe concurrent systems. CPS are complex systems with multiple characteristics, and although LTS cannot directly represent CPS in great detail, it can cover various aspects of CPS. However, since LTS cannot model the continuous behavior, probabilistic behavior, and time characteristics of CPS, this will directly affect the formal verification of CPS. In addition, the LTS model also cannot directly express the cost consumed during the operation of CPS, which results in us not being able to obtain performance metrics about CPS from the model. Therefore, we are exploring the extended LTS as a potential solution for a more detailed description and comprehensive formal verification of CPS.

Typically, model checking provides a qualitative answer to whether a system satisfies certain properties, with the results being “true” or “false.” However, not only a CPS has real-time and probabilistic properties, but also it has specific performance requirements. Therefore, in addition to verifying whether the system satisfies the properties qualitatively, we also need to consider the cost and time of CPS satisfying the properties, such as the probability of operation errors, power consumption, and running time [15, 16]. Probabilistic model checking is an extension of traditional model checking that enables both qualitative and quantitative verification of formal CPS models. Temporal logic plays a crucial role in model checking techniques, and the primary temporal logics used to describe system properties include probabilistic computation tree logic (PCTL), and continuous stochastic logic (CSL) [6, 17]. Similar to other model checking methods, the main challenge faced in probabilistic model checking is state space explosion, as the number of system states increases exponentially with the number of concurrent components, making it infeasible to thoroughly search the state space.

Symbolic model checking is a useful technique for handling larger systems and tackling the state explosion problem. In this approach, the system state is implicitly represented by predicates, as well as the initial state and transition relations of the system. Many studies, such as [6, 18–20], have shown that symbolic representations are generally more efficient than explicit ones. Additionally, this method utilizes advanced data structures such as Binary Decision Diagrams (BDD) or Multi-Terminal BDD (MTBDD) [21–24], which can effectively store and explore a large number of system states simultaneously. Therefore, in order to tackle the challenges of managing time, probability, and cost information in CPS while maintaining the requirement of low spatiotemporal complexity for quantitative evaluation, this paper draws on the main concepts of probabilistic model checking and symbolic model checking. This approach has significant practical implications for the quantitative assessment of CPS.

## 1.2 Contributions

In this paper, we focus on the modeling and quantitative evaluation of CPS using a formal approach whose main contributions and processes are shown in Fig. 1.

- The first step aims to achieve formal modeling and properties representation of CPS. In order to comprehensively capture the continuous behavior, time characteristics, probability behavior, and operational cost information of CPS, we first propose a hybrid probability time cost transition system (HPTCTS) based on LTS. Next, we refer to the syntax and semantics of PCTL to propose HPTCTS-temporal logic (HPTCTS-TL) for depicting the critical properties of CPS.
- We employ symbolic model checking to alleviate the state explosion problem in the quantitative evaluation of the HPTCTS model. We use the ordered reduced binary decision diagram (ORBDD) symbolic representation of the HPTCTS model  $M$ . Moreover, we treat the formula  $\varphi'$  that represents the quantitative index values from the HPTCTS-TL formula into the CTL for-

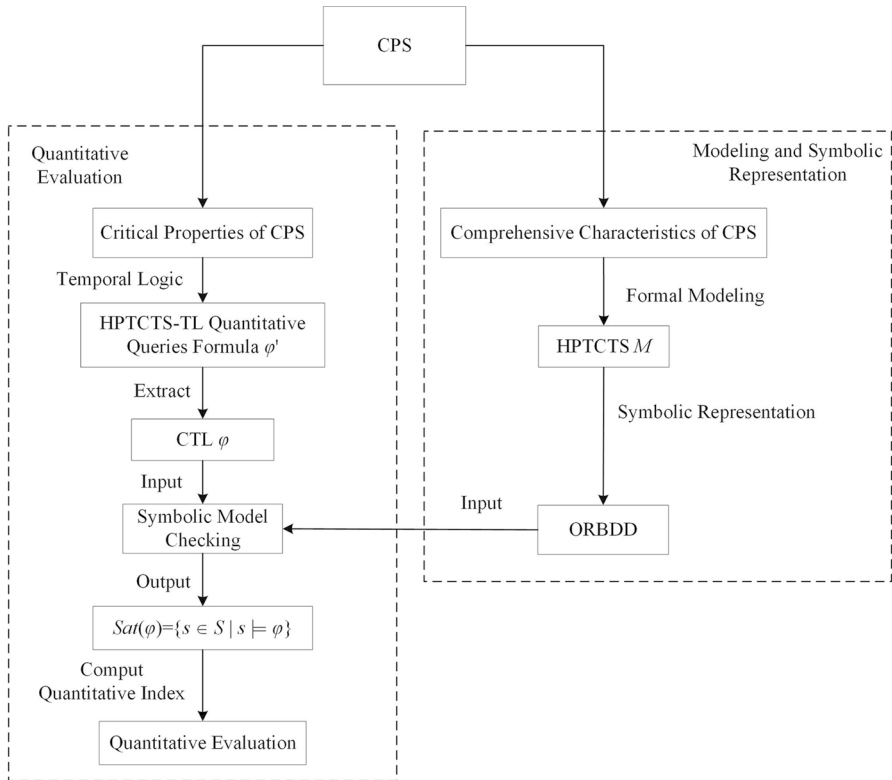


Fig. 1 An overview of our approach

mula  $\varphi$ . Then, the symbolic representations of  $M$  and  $\varphi$  are used as inputs for the symbolic model checking. The set  $S$  of states that satisfy the CTL formula  $\varphi$  is computed as  $Sat(\varphi)$ , denoted as  $\{s \in S \mid s \models \varphi\}$ .

- Finally, we calculate  $\varphi'$  to obtain the result of the quantitative query, i.e., the probability, cost, and time of the HPTCTS satisfying the formula  $\varphi$ . This is used as the quantitative evaluation metric of the performance of the CPS.

After the preliminaries in Sect. 2, we extend LTS and propose HPTCTS in Sect. 3 for modeling CPS. In Sect. 4, we introduce how to symbolically represent the states and transitions of HPTCTS. In Sect. 5, we propose a temporal logic suitable for describing the attributes of the HPTCTS model, and provide the process of symbolizing temporal logic and performance calculation. In Sects. 6–7, we present the implementation details and illustrate the performance modeling/quantitative evaluation procedure of CPS through a case study. In Sect. 8, we review related work. In Sect. 9, we present conclusions and future prospects.

## 2 Preliminaries

In this section, we introduce the background knowledge necessary for the formal modeling and efficient quantitative evaluation of CPS. We first describe LTS, which is a widely used mathematical model for representing CPS behavior. Then we introduce PCTL, which is a temporal logic used for specifying probabilistic properties of systems. Next, we describe ORBDD, which is a data structure used for representing Boolean functions. Finally, we introduce efficient symbolic model checking algorithm for alleviating the state space explosion problem.

### 2.1 Labeled transition system (LTS)

LTS is a highly expressive system model, and both probabilistic and hybrid system models can be obtained as variants of LTS [11]. LTS is also used as a semantic model of process algebra to describe concurrent systems and thus can be obtained by compiling process algebra. The state transitions of the LTS system model can be represented by state transition diagrams, which can visually represent the functional behavior of the system. Next, we give a definition based on the standard concept of LTS.

**Definition 1** A labeled transition system (LTS) is a tuple  $(S, Act, \longrightarrow, s_0, AP, L)$ , Where

- $S$  is a set of states.
- $Act$  is a set of actions.
- $\longrightarrow \subseteq S \times Act \times S$  is a transition relation.
- $s_0 \in S$  is the initial state.
- $AP$  is a set of atomic propositions.
- $L : S \rightarrow 2^{AP}$  is a label function, which maps the state  $s \in S$  to a set of atomic propositions with true values in the current state.

An LTS can give an intuitive and accurate execution of a system where a state  $s$  is transferred to a state  $s$  through an action  $a$ , which we usually denote as  $(s, a, s) \in \longrightarrow$  and abbreviate it as  $s \rightarrow s$ .

### 2.2 Probabilistic computation tree logic (PCTL)

PCTL is a temporal logic used for specifying and verifying probabilistic properties of systems. PCTL extends classical temporal logic by allowing probabilistic operators to be used in temporal formulas [6].

**Definition 2** The syntax of PCTL includes Boolean operators, temporal operators, and probabilistic operators. The temporal operators include next ( $X$ ), until ( $U$ ), and

eventually ( $F$ ), while the probabilistic operators include probability ( $P$ ). PCTL formulas are constructed using the following syntax:

$$\phi ::= a \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid P_{\bowtie p}[\psi]$$

- $a$ : a propositional symbol denoting a property of a state
- $\neg\phi$ : negation of a formula
- $\phi_1 \wedge \phi_2$ : conjunction of two formulas
- $P_{\bowtie p}[\psi]$ : probability operator, quantifying the probability of satisfying the path formula  $\psi$  with respect to the proposition  $\alpha$ , where  $\bowtie \in \{<, \leq, >, \geq\}$ ,  $p \in [0, 1]$
- $\psi ::= X\phi \mid \phi_1 U\phi_2$  denotes the path formula.  $X\phi$ : next operator, stating that  $\phi$  must hold in the next state.  $\phi_1 U\phi_2$ : until operator, stating that  $\phi_1$  must hold until  $\phi_2$  eventually holds.

PCTL formulas can also be extended to include the logical operators  $\vee$  (disjunction) and  $\rightarrow$  (implication), as well as the temporal operator  $G$  (globally). Additionally, PCTL can be combined with the branching-time logic CTL (Computation Tree Logic) to form the logic PCTL-CTL. Moreover, each PCTL formula evaluates to a Boolean by virtue of the comparison of the actual probability of the set of paths with the probability bound  $p$ . If the outermost operator of a PCTL formula is  $P_{\bowtie p}[\psi]$ , we can omit the bound  $\bowtie p$  and simply compute the probability instead [25–27],  $P_{=?}[\psi]$  is used to denote the value of probability from a PCTL formula. The PCTL model checking algorithm computes the actual probability anyway, so no extra cost is incurred.

**Definition 3** The semantics of PCTL can be defined on a probabilistic model  $\mathcal{M} = (S, P, s_0)$ , where  $S$  is a set of states,  $P : S \times S \rightarrow [0, 1]$  is a transition probability function, and  $s_0 \in S$  is an initial state. In addition, the path of  $\mathcal{M}$  is  $\pi$ . A path  $\pi$  in a probabilistic structure is an infinite sequence of states  $\pi = s_0, s_1, s_2, \dots$  such that for each  $i > 0, i \in N, P[s_i, s_{i+1}] > 0$ .

The satisfaction relation  $\mathcal{M}, s \models \phi$  denotes that formula  $\phi$  is satisfied in state  $s$  of  $\mathcal{M}$ , and is defined inductively as follows:

- $(\mathcal{M}, s) \models a$  iff  $s \in L(a)$ , where  $L(a)$  is the set of states where proposition  $a$  is true;
- $(\mathcal{M}, s) \models \neg\phi$  iff  $(\mathcal{M}, s) \not\models \phi$ .
- $(\mathcal{M}, s) \models \phi_1 \wedge \phi_2$  iff  $(\mathcal{M}, s) \models \phi_1$  and  $(\mathcal{M}, s) \models \phi_2$ .
- $(\mathcal{M}, s) \models P_{\bowtie p}[\psi]$  iff  $\mathcal{P}(s, \psi) \bowtie p$ , where  $\mathcal{P}(s, \psi)$  is the probability of satisfying  $\psi$  from state  $s$ , and  $\bowtie \in \{<, \leq, >, \geq\}$ .

For the path formula  $\psi ::= X\phi \mid \phi_1 U\phi_2$ , the satisfaction relation for any path  $\pi$  is as follows.

- $(\mathcal{M}, \pi) \models X\phi$  iff  $P(s_i, s_{i+1}) > 0$  for some  $s_{i+1} \in S$  such that  $(\mathcal{M}, s_{i+1}) \models \phi$ .
- $(\mathcal{M}, \pi) \models \phi_1 U\phi_2$  iff there exists a path  $\pi = s_0, s_1, \dots, s_n$  in  $\mathcal{M}$  such that  $s = s_0$  and  $(\mathcal{M}, s_n) \models \phi_2$ , and for all  $0 \leq i < n, (\mathcal{M}, s_i) \models \phi_1$ .

Overall, PCTL provides a rich language for expressing a wide variety of probabilistic properties of systems, and its semantics are well defined and widely used in the field of probabilistic model checking.

### 2.3 Ordered reduced binary decision diagram (ORBDD)

An Ordered Reduced Binary Decision Diagram, abbreviated as ORBDD [19], is a data structure used for the representation and manipulation of Boolean functions. It finds common applications in the fields of computer science and electronic engineering, particularly in the design and optimization of logical circuits. ORBDD is a specialized form of Binary Decision Diagram (BDD) with a primary objective of efficiently representing Boolean functions in a compact manner, making it a powerful tool for the analysis, manipulation, and optimization of these functions.

ORBDD is a variant of BDD and inherits its fundamental concepts. BDD, which stands for Binary Decision Diagram, is a directed acyclic graph employed for representing the truth tables of Boolean functions. Each node in the graph represents a Boolean variable and has two child nodes, one for the value of variable value being 0 and the other for the value being 1. In BDD, 0 typically represents “False,” while 1 represents “True.” This ordered and reduced characteristic of ORBDD allows it to efficiently handle large Boolean functions while minimizing storage requirements and computational complexity.

### 2.4 Symbolic model checking

Symbolic model checking is a technique for verifying system properties by manipulating symbolic representations of the system’s state space and transition relation [18, 19, 28]. Instead of exploring the entire state space of a system, which can be exponentially large, symbolic model checking uses algorithms such as BDD-based reachability analysis and fixed-point computation to efficiently explore only relevant portions of the state space.

Symbolic model checking is well suited for verifying complex systems, such as concurrent and distributed systems and for handling temporal properties. The input to a symbolic model checker is typically a formal specification of the system in a temporal logic such as PCTL, and the output is a counterexample trace or a proof of correctness. Symbolic model checking has been successfully applied to a wide range of applications, including hardware and software verification, protocol analysis, and biological systems modeling. However, it can still face scalability challenges for large-scale systems and requires careful consideration of the system model and property specification to ensure effective verification.

### 3 Hybrid probabilistic time cost transition system (HPTCTS)

In this section, we take into account the modeling requirements of CPS and extend LTS to describe in detail various characteristics and behaviors of CPS. We introduce hybrid probabilistic time cost transition system (HPTCTS) as a novel model for CPS.

#### 3.1 Definition of HPTCTS

During the operation of CPS, the system performs functional action transitions to achieve certain functionalities. Moreover, CPS has some temporal operations that can keep the system in a state for a specific period of time. In addition, CPS is a complex system that is deeply integrated by hardware/software, network and physical environment, with continuous, concurrent and uncertain behaviors. Meanwhile, the performance (energy consumption, time and other costs) of CPS also needs to be considered. In order to describe the various characteristics and behaviors of CPS in detail, the formal representation of HPTCTS model is given in this section.

**Definition 4** (*Hybrid probability time cost transition system*). A hybrid probability time cost transition system (HPTCTS) is a tuple  $(S, s_0, \Lambda, C, Var, F, I, E, P, L)$ , where:

- $S$  is a set of states. A state represents a particular situation of a system at a given moment in time and describes a mode in which the system is in, and a system contains a set of modes. For example, a system may be in one of the different modes “starting,” “running” or “stopped.” The state cost is a function  $\rho : S \rightarrow R_{\geq 0}$ . Intuitively,  $\rho(s)$  is the cost acquired in state  $s$  per time-step.
- $s_0$  is the initial state,  $s_0 \in S$ .
- $\Lambda = Act \cup [T]$ , it is a set of actions,  $\alpha \in \Lambda$ ,  $Act = \{act_1, act_2, act_3, \dots\}$ ,  $act_i = a|c?|c!|\tau$ ,  $a$  is a discrete action,  $c?$  is a receive action, it means that the channel  $c$  receives the message.  $c!$  is a sending action, it means getting message from channel  $c$  and sending it to a value,  $\tau$  is an internal action.  $[T] = \{[t]|t \in \mathbb{R}\}$ , it is a set of time actions. The function action  $act$  represents the action performed by the system to realize a function, and the time action  $[t]$  represents the system staying in a certain state for a period of time.
- $C : S \times Act \times S \rightarrow R_{\geq 0}$  is a transition cost function of the system state to the target state.
- $Var$  represents the set of variables. These variables can be the state of various components of the system, such as position, velocity and temperature. Here,  $Var$  can include not only the set of discrete variables  $dVar$ , and the set of continuous variables  $cVar$ ,  $Var = dVar \cup cVar$  but also different types of variables, including system data variables and time variables.
- $F : s \rightarrow De$  represents the rate of change of variable when the system undergoes continuous variations. It assigns a flow condition  $De$  to the state  $s$ ,  $s \in S$ .



$De$  is a differential equation that is typically expressed as a first-order derivative equation involving the variable and time.

- $I : S \rightarrow Scon$  is the operating condition of the system state, which can be expressed as the invariant condition of the hybrid automata. This article argues that the system state  $s \in S$  is constrained by variables, where  $scon \in Scon$  is the invariant condition, which is a conditional expression of the variable  $var$  and the constant  $c$ ,  $scon := var \sim c | scon_1 \wedge scon_2, c \in \mathbb{R}, \sim \in \{<, >, \leq, \geq, =\}$ . If  $scon = true$ , state  $s$  continues, otherwise, the system should execute the next state.
- $E \subseteq S \times Gcon \times \Lambda \times T \times C \times S$  is the transition relations of the states.  $Gcon$  is a set of guard conditions,  $gcon \in Gcon$ ,  $gcon$  represents the precondition of execution state transition.
- $P : E \rightarrow [0, 1]$  is a transition probability function of the system state to the target state.  $\forall s \in S, \sum_{st \in S, act \in Act} P(s, act, st) = 1$ .
- $L : S \rightarrow 2^{PF}$  is a labeling function of predicate logic that can be used to represent the properties that the system should satisfy,  $PF$  is a set of predicate logic formulas.

**Definition 5 (Infinite path)** The execution of the system can be represented by the paths in the system model  $M$  of HPTLTS. A path is a sequence of states:  $\pi = s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \xrightarrow{gcon_1, act_1, t_1, c_1} p_1 s_2 \dots \xrightarrow{gcon_{i-1}, act_{i-1}, t_{i-1}, c_{i-1}} p_{i-1} s_i \dots$ , where  $s_i \in S, p_i \in [0, 1]$  is the probability value,  $act_i \in Act, i \geq 0$ . The infinite set of paths starting from state  $s$  is denoted by  $Path_s$ .

**Definition 6 (Finite path)** The execution of the system can be represented by the paths in the system model  $M$  of HPTLTS. A path is a sequence of states:  $\pi = s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \xrightarrow{gcon_1, act_1, t_1, c_1} p_1 s_2 \dots \xrightarrow{gcon_{n-1}, act_{n-1}, t_{n-1}, c_{n-1}} p_{n-1} s_n$ , where  $s_i \in S, p_i \in [0, 1]$  is the probability value,  $act_i \in Act, n \geq i \geq 0$ . The finite set of paths starting from state  $s$  is denoted by  $Path_s^{fin}$ .

**Definition 7 (Probability measure)** For the system model  $M$  of HPTLTS and the state  $s \in S$ , the probability of a path starting from state  $s_0$  can be measured by  $Pr_{s_0}^M$  (or  $Pr$  for short). In order to correlate probabilities with events in  $M$ , the probability space of the  $M$  is defined by  $(\Omega, \xi, Pr)$ . The sample space  $\Omega$  denotes the infinite path in  $M, \Omega = Path_s$ . The event set  $\xi$  is the smallest  $\sigma$ -algebra formed by the cylinder set  $Cyl(s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \dots \xrightarrow{gcon_{n-1}, act_{n-1}, t_{n-1}, c_{n-1}} p_{n-1} s_n)$ , where  $Cyl(s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \dots \xrightarrow{gcon_{n-1}, act_{n-1}, t_{n-1}, c_{n-1}} p_{n-1} s_n)$  is caused by a finite path starting from state  $s_0$ . For the path  $\pi = s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \dots \xrightarrow{gcon_{n-1}, act_{n-1}, t_{n-1}, c_{n-1}} p_{n-1} s_n, Cyl(\pi) = \{\pi' \in Path_s | \pi \text{ is the prefix of } \pi'\}$ . Then the probability of this cylinder set is defined as  $Pr(Cyl(s_0 \xrightarrow{gcon_0, act_0, t_0, c_0} p_0 s_1 \dots \xrightarrow{gcon_{n-1}, act_{n-1}, t_{n-1}, c_{n-1}} p_{n-1} s_n)) = \prod_{i=0}^n P(s_i, act_i, s_{i+1})$ . The

probability measure  $Pr : \xi \rightarrow [0, 1]$ . Moreover, for any finite path sets  $R \subseteq Path_s$ , the probability measure can be calculated by formula  $Pr(R) = \sum_{\pi_i \in R} (Cyl(\pi_i)), 1 \leq i \leq n$ .

**Definition 8 (Cumulative execution time)** The cumulative execution time for path  $\pi$  of HPTCTS is defined as  $CumTime(\pi) = \sum_{i=0}^{n-1} (t_i)$ ,  $t_i$  denotes the time spent in state  $\pi(i)$ .

**Definition 9 (Cumulative transition cost)** The cumulative transition cost of the path  $\pi$  of HPTCTS is defined as  $CumTranCost(\pi) = \sum_{i=0}^n C(\pi(i - 1), act_i, \pi(i))$ .

**Definition 10 (Cumulative cost)** The cumulative cost  $CumCost(\pi)$  of a path  $\pi$  contains the state cost  $\rho(s)$  and the transition cost  $C(s_{i-1}, act_i, s_i)$ . The cumulative cost of the path  $\pi$  of HPTCTS is defined as  $CumCost(\pi) = \rho(\pi(n)) + \sum_{i=0}^{n-1} \rho(s_i) + CumTranCost(\pi)$ .

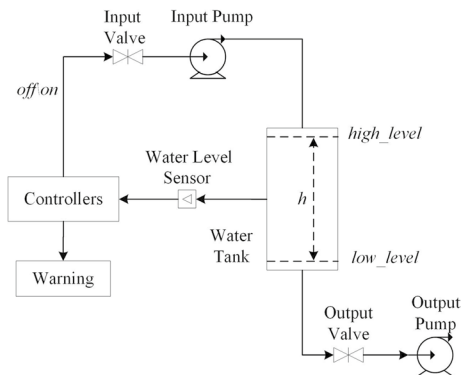
According to the definition of LTS in the Sect. 2.1 and the related definition of HPTCTS in this subsection, HPTCTS not only captures all the semantics that can be expressed by LTS, but it also describes continuous, temporal as well as stochastic behaviors of a CPS. In addition, HPTCTS can express the cost consumed by the system at runtime, which is not directly expressed by the original LTS.

### 3.2 An example

This subsection illustrates the HPTCTS modeling capability through an example of Water Tank Control System [29] as shown in Fig. 2. The system is composed of a water tank, a water level sensor, a controller, an warning device, and an input/output valve and pump. The system can be represented as

$$\text{System} = \text{WaterTank} \parallel \text{Sensor} \parallel \text{Controller} \parallel \text{WarningDevice}.$$

**Fig. 2** The water tank control system



The controller obtains the water level through the sensor and determines whether to close and open the water inlet system according to the preset water level threshold.

Errors were introduced in the controller. When the water level is lower than the water level threshold  $low\_level$  or higher than the water level threshold  $high\_level$ , and the controller has migration constraint time  $trTime$  greater than  $t_1$  and  $t_2$ , the fault may not execute the on and off actions of the water inlet valve. This fault occurs randomly. After the error occurs, the warning device is activated, and the error rates are  $1 - q$  and  $1 - p$ , respectively. After the water inlet system is opened, the water tank reaches  $high\_level$  within  $t_1$  time. After the water inlet system is closed, the water tank drops to  $low\_level$  within  $t_2$  time. The time for closing and opening the water inlet is  $tr_1$  and  $tr_2$ , and the corresponding costs are  $c_1$  and  $c_2$ . The time for the system to switch to the faulty state is  $tr_3$ , and the corresponding cost is  $c_3$ . The time for opening the error warning device is  $tr_4$ , and the corresponding cost is  $c_4$ . For the description of the above system, The HPTCTS model for this system is shown in Fig. 3.

In Fig. 3, based on Definition 4, the discrete behavior, continuous behavior, probabilistic behavior, running cost of the water tank control system can be modeled by HPTCTS. Additionally, as a semantic model of process calculus, HPTCTS can represent the runtime states of multiple process components in the system. At the same time, HPTCTS can be used to model the performance of CPS. In contrast to the HPTCTS model shown in Fig. 3, according to Definition 1, we can employ LTS to model the discrete behaviors of the water tank control system. However, LTS is limited in its ability to describe the continuous evolution of system states over time. Moreover, the states within the LTS model only encompass instantaneous actions during transitions, thus neglecting to capture the duration for which the system remains in a specific state. Additionally, it is important to note that the LTS model is unable to account for probabilistic choices and the

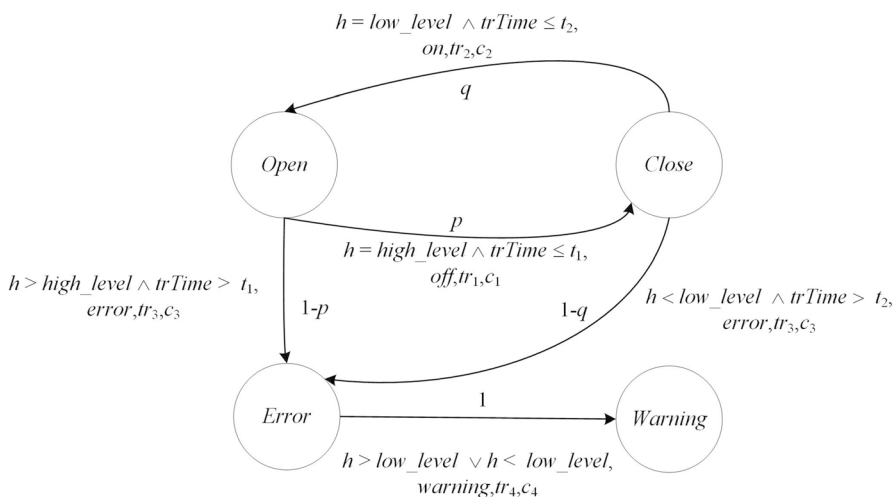


Fig. 3 HPTCTS of water tank control system

costs incurred during system state transitions. These show that modeling the CPS using HPTCTS will provide a more comprehensive representation of the characteristics and behavior for the system.

## 4 Symbolic representation of HPTCTS

By utilizing symbolic encoding techniques, we aim to efficiently represent the HPTCTS model as a Boolean expression and convert it into a ordered reduced binary decision diagram (ORBDD). This approach can help alleviate the state space explosion issue when performing quantitative evaluations of HPTCTS.

### 4.1 The symbolization of states

Let  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$  be HPTCTS. All elements in the state set  $S$  are encoded to complete the symbolic representation of the state in HPTCTS. First, a fixed sequence  $pf_1, pf_2, \dots, pf_n$  is assigned to all predicate logic formulas in  $PF$ . A boolean variable  $x_i$  is assigned to each  $pf_i$ , and a Boolean expression  $f_s(x_1, x_2, \dots, x_n)$  is defined to represent the state, if  $pf_i \in L(s)$ ,  $l_i = x_i$ , otherwith,  $l_i = \bar{x}_i$ . The set  $S = \{s_1, s_2, \dots, s_m\}$  can be expressed as  $f_s = f_{s_1} \vee f_{s_2} \vee \dots \vee f_{s_m}$ ,  $m = |S|$ ,  $n = \lceil \log_2(m) \rceil$ ,  $n$  denote upward rounding based on the corresponding calculation.

For the set of states  $S = \{Open, Close, Error, Warning\}$  in Fig. 3, the corresponding specified Boolean variables order  $x_1, x_2$ . Let  $Open = s_0, Close = s_1, Error = s_2, Warning = s_3$ , the state can be represented as follows:  $f_{s_0} = x_1 \bar{x}_2, f_{s_1} = \bar{x}_1 x_2, f_{s_2} = \bar{x}_1 \bar{x}_2, f_{s_3} = x_1 x_2$ . The Boolean expression of state set  $S$  is  $f_S = x_1 \bar{x}_2 + \bar{x}_1 x_2 + \bar{x}_1 \bar{x}_2 + x_1 x_2$ .

### 4.2 The symbolization of transition relations

We can see that the state probabilistic transition relation is a function of  $S \times Gcon \times \Lambda \times T \times C \times S \rightarrow [0, 1]$  form in the HPTCTS model. We encode a state  $s$  by the Boolean vector  $(x_1, x_2, \dots, x_n)$  and the Boolean vector  $(x'_1, x'_2, \dots, x'_n)$  to encode next state  $s'$ . Formally, we define an injection  $e : S \rightarrow \{0, 1\}^n$ . We use also Boolean vector  $(y_1, y_2, \dots, y_k)$  and  $(z_1, z_2, \dots, z_r)$  to encode transition conditions and actions, respectively. Where  $v = |Gcon|, k = \lceil \log_2(v) \rceil, j = |Act|, r = \lceil \log_2(j) \rceil$ ,  $k$  and  $r$  denote upward rounding based on the corresponding calculations. A valuation  $x$  of the Boolean vector  $(x_1, x_2, \dots, x_n)$  encodes a state  $s$  by  $e(s)$ ,  $s \in S$ . A valuation  $y$  of the Boolean vector  $(y_1, y_2, \dots, y_k)$  encodes a transition condition  $gcon$  by  $e(gcon)$ ,  $gcon \in Gcon$ . A valuation  $z$  of the Boolean vector

**Table 1** Encoding the set of states of HPTCTS  $\mathcal{M}$  ( $\mathcal{M}$  in Fig. 3)

$s \in \mathcal{S}$	$s_0$	$s_1$	$s_2$	$s_3$	$e(s_i)$
$s_0$	0	1	1	0	10
$s_1$	1	0	1	0	01
$s_2$	0	0	0	1	00
$s_3$	0	0	0	0	11

**Table 2** Encoding the transition relations of HPTCTS  $\mathcal{M}$  ( $\mathcal{M}$  in Fig. 3)

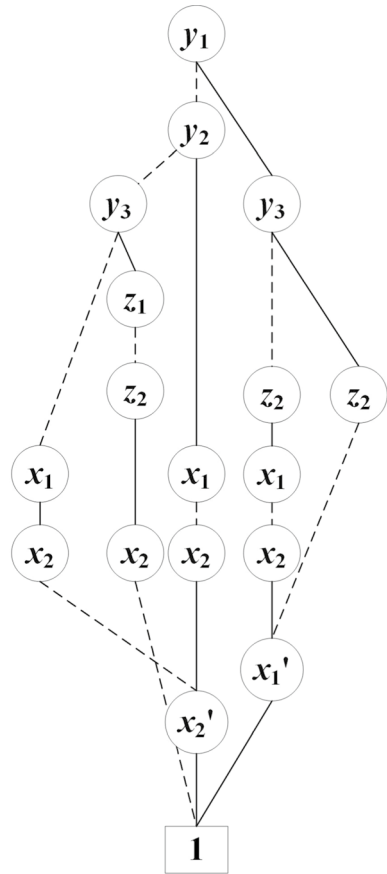
$\rightarrow$	$y$	$z$	$x_1$	$x_2$	$x'_1$	$x'_2$
$s_0 \rightarrow s_1$	(000)	(00)	1	0	0	1
$s_0 \rightarrow s_2$	(001)	(01)	1	0	0	0
$s_1 \rightarrow s_0$	(011)	(11)	0	1	1	0
$s_1 \rightarrow s_2$	(110)	(01)	0	1	0	0
$s_2 \rightarrow s_3$	(111)	(10)	0	0	1	1

$(z_1, z_2, \dots, z_r)$  encodes a action  $a$  by  $e(a)$ ,  $a \in Act$ . Define a Boolean function  $f : \{0, 1\}^k \times \{0, 1\}^r \times \{0, 1\}^n \rightarrow \{1\}$ .

We consider the HPTCTS  $\mathcal{M}$  (Fig 3).  $\mathcal{M}$  contains the set of states  $S_{\mathcal{M}} = \{s_0, s_1, s_2, s_3\}$ , the set of condition  $Gcon$ , the set of actions  $Act = \{on, off, error, warning\}$ . We use  $(x_1, x_2)$  to encode the set of states in  $S_{\mathcal{M}} : e(s_0) = (10), e(s_1) = (01), e(s_2) = (00), e(s_3) = (11)$ , we use the set  $(y_1, y_2, y_3)$  to encode the conditions:  $e(gcon_1) = (000), e(gcon_2) = (001), e(gcon_3) = (011), e(gcon_4) = (110), e(gcon_5) = (111)$ , and we use the set  $(z_1, z_2)$  to encode the actions:  $e(on) = (00), e(error) = (01), e(off) = (11), e(warning) = (10)$ , respectively. The Table 1 and Table 2 summarizes the process of encoding the states and transition relations function.

According to Tables 1 and 2, we can see that when  $y_1 = y_2 = y_3 = z_1 = z_2 = x_2 = x'_1 = 0, x_1 = x'_2 = 1, f(y_1, y_2, y_3, z_1, z_2, x_1, x_2, x'_1, x'_2) = 1$ . When  $y_1 = y_2 = z_1 = x_2 = x'_2 = 0, y_3 = z_2 = x_1 = 1, f(y_1, y_2, y_3, z_1, z_2, x_1, x_2, x'_1, x'_2) = 1$ . When  $y_1 = x_1 = x'_2 = 0, y_2 = y_3 = z_1 = z_2 = x_2 = x'_1 = 1, f(y_1, y_2, y_3, z_1, z_2, x_1, x_2, x'_1, x'_2) = 1$ . When  $y_3 = z_1 = x_1 = x'_1 = x'_2 = 0, y_1 = y_2 = z_2 = x_2 = 1, f(y_1, y_2, y_3, z_1, z_2, x_1, x_2, x'_1, x'_2) = 1$ . When  $z_2 = x_1 = x_2 = 0, y_1 = y_2 = y_3 = z_1 = x'_1 = x'_2 = 1, f(y_1, y_2, y_3, z_1, z_2, x_1, x_2, x'_1, x'_2) = 1$ . The transition relations of HPTCTS for water tank control system represented by ORBDD is shown in Fig. 4. In the ORBDD, the dashed line indicates that a value of 0 is assigned to  $x_i, y_i, z_i$  and the solid line indicates that a value of 1 is assigned to  $x_i, y_i, z_i$ .

**Fig. 4** ORBDD encoding the transition relations of HPTCTS  $\mathcal{M}$ ( $\mathcal{M}$  in Fig. 3)



## 5 Quantitative evaluation

Temporal logic is the basis for model checking to describe the properties of the system. In this section, we apply improved HPTCTS temporal logic (HPTCTS-TL) to describe the performance requirements that HPTCTS model needs to satisfy. Meanwhile, we symbolized HPTCTS-TL and provided a method to calculate the performance formula of HPTCTS-TL that includes a symbolic model checking algorithm for quantitatively evaluating HPTCTS model.

### 5.1 HPTCTS-TL (HPTCTS-temporal logic)

For the CPS comprehensive model HPTCTS proposed in Sect. 3.1, we also need a suitable temporal logic to describe the properties of the HPTCTS model accurately and effectively. That is, it has the ability to describe data constraints, time constraints, probability, cost and other properties that the HPTCTS model needs to satisfy. Moreover, this temporal logic has a strict syntactic and semantic

expressiveness. We give the complete syntax and semantics of HPTCTS-TL, which has a similar composition structure to PCTL.

Since HPTCTS-TL can constrain the data variables, the properties of the inscribed system model are not only propositional logic formulas, but also predicate logic formulas and variables. In order to describe the properties of variables within a certain range of values, this paper also introduces the universal quantification  $\forall$  and the existential quantification  $\exists$ .

**Definition 11** (*Syntax*). The syntax of HPTCTS-TL is defined as follows.

$$\phi ::= pf(x_1, \dots, x_n) \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid (\exists x:V)\phi \mid (\forall x:V)\phi \mid P_{\sim p}(\phi) \mid C_{\sim c}(\phi) \mid T_{\sim t}(\phi)$$

Where:

- $pf(x_1, \dots, x_n)$  is an  $n$ -ary predicate formula,  $(x_1, \dots, x_n)$  are variables,  $pf$  represents an  $n$ -ary predicate.
- $\neg\phi$  is negative.
- $(\exists x : V)\phi$  and  $(\forall x : V)\phi$  describe existential or any variable  $x$  of type  $V$ , for which  $\phi$  is valid.  $\phi_1 \wedge \phi_2$  is conjunction formula.
- $P_{\sim p}(\phi)$  is probability formula. The probability of HPTCTS model meeting the path formula  $\phi$  is within the bound  $p$ . Here,  $p \in [0, 1]$ ,  $\sim \in \{<, \leq, >, \geq\}$ .
- $C_{\sim c}(\phi)$  is cumulative cost formula. The cumulative cost of the system meeting the path formula  $\phi$  is within the bound  $c$ ,  $c \in R$ . Here, cost refers to the consumption of resources other than time. For example, the energy consumption of the system, the price spent, the memory footprint.
- $T_{\sim t}(\phi)$  is cumulative time formula. The cumulative time of the system meeting the path formula  $\phi$  is within the bound  $t$ ,  $t$  denotes a natural number.
- $\phi ::= E\psi \mid A\psi$  denotes a path formula with path quantifiers  $E$  and  $A$ .
- $\psi ::= X\phi \mid \phi_1 U_{[a,b]}\phi_2 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid pf$  denotes a path-formula.  $X\phi$ : next operator, stating that  $\phi$  must hold in the next state.  $\phi_1 U_{[a,b]}\phi_2$ : within the time constraint  $[a, b]$ , the HPTCTS model satisfies the formula  $\phi_1 U\phi_2$ , iff  $\phi_2$  holds at  $s_i$ , and  $\phi_1$  holds at every preceding state. Here,  $s_i \in S, 0 \leq i \leq n$ , where  $n$  is a natural number,  $[a, b]$  is a time interval with  $a \in Z, b \in Z \cup \infty$ , and  $a \leq b$ .

If the outermost operators of HPTCTS-TL formulas are  $P_{\sim p}(\cdot)$ ,  $C_{\sim c}(\cdot)$ , and  $T_{\sim t}(\cdot)$ , we simplify our calculations by disregarding the bounds ( $\sim p, \sim c, \sim t$ ) and focusing solely on computing the probability, cost, and time. We define  $\phi' ::= \phi'_1 \mid \phi'_2 \mid \phi'_3$  represents the quantitative queries formula from HPTCTS-TL formulas. Formula  $\phi'_1 ::= P_{=?}(\phi) \mid P_{max=?}(\phi) \mid P_{min=?}(\phi)$  as the probability to be computed, which can be represented as a function that maps a given state space  $S$  of HPTCTS model to the interval  $[0,1]$ :  $S^\pi \rightarrow [0, 1]$ . Formula  $\phi'_2 ::= C_{max=?}(\phi) \mid C_{min=?}(\phi)$  represents the cost to be computed and can be expressed as a function:  $S^\pi \rightarrow R$ . Similarly, formula  $\phi'_3 ::= T_{max=?}(\phi) \mid T_{min=?}(\phi)$  represents the time to be computed and can be expressed as a function:  $S^\pi \rightarrow R$ .

Where:

- $P_{=?}(\phi)$ : What is the probability that HPTCTS model meets the path formula  $\phi$ ? What are the maximum and minimum probabilities that  $P_{max=?}(\phi)$

and  $P_{min=?}(\varphi)$  indicate that the HPTCTS model satisfies the path formula  $\varphi$ , respectively?

- What are the maximum and minimum cumulative cost that  $C_{max=?}(\varphi)$  and  $C_{min=?}(\varphi)$  indicate that the HPTCTS model satisfies the path formula  $\varphi$ , respectively?
- What are the maximum and minimum cumulative time that  $T_{max=?}(\varphi)$  and  $T_{min=?}(\varphi)$  indicate that the HPTCTS model satisfies the path formula  $\varphi$ , respectively?

For a given HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$ , the semantics of HPTCTS-TL is defined by the satisfaction relation  $(\mathcal{M}, s) \models \phi$ . Let  $s \in S$  be a state,  $Path_s^{fin}$  is an finite path in  $\mathcal{M}$ , and  $I(s)$  is the condition constraint of the state. For a finite path  $\pi$ ,  $\pi(i)$  denotes the  $i$ -th state on a path  $\pi$ .

**Definition 12 (Semantics)** The satisfaction relationship of HPTCTS-TL is defined as follows.

- $(\mathcal{M}, s) \models pf(x_1, \dots, x_n)$  iff  $I(s) \models pf(x_1, \dots, x_n)$
- $(\mathcal{M}, s) \models \neg\phi$  iff  $(\mathcal{M}, s) \not\models \phi$
- $(\mathcal{M}, s) \models \phi_1 \wedge \phi_2$  iff  $(\mathcal{M}, s) \models \phi_1$  and  $(\mathcal{M}, s) \models \phi_2$
- $(\mathcal{M}, s) \models (\exists x : V)\phi$  iff  $\exists v \in V, (\mathcal{M}, s) \models \phi\{v/x\}$
- $(\mathcal{M}, s) \models (\forall x : V)\phi$  iff  $\forall v \in V, (\mathcal{M}, s) \models \phi\{v/x\}$
- $(\mathcal{M}, s) \models P_{\sim p}(E\psi)$  iff  $\exists \pi, Pr\{\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi\} \sim p$
- $(\mathcal{M}, s) \models C_{\sim c}(E\psi)$  iff  $\exists \pi, CumCost(\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi) \sim c$
- $(\mathcal{M}, s) \models T_{\sim t}(E\psi)$  iff  $\exists \pi, CumTime(\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi) \sim t$
- $(\mathcal{M}, s) \models P_{\sim p}(A\psi)$  iff  $\forall \pi, Pr\{\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi\} \sim p$
- $(\mathcal{M}, s) \models C_{\sim c}(A\psi)$  iff  $\forall \pi, CumCost(\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi) \sim c$
- $(\mathcal{M}, s) \models T_{\sim t}(A\psi)$  iff  $\forall \pi, CumTime(\pi \in Path_s^{fin} \mid (\mathcal{M}, \pi) \models \psi) \sim t$
- $(\mathcal{M}, s) \models E\psi$  iff  $(\mathcal{M}, \pi) \models \psi$  for some  $\pi \in Path_s^{fin}$
- $(\mathcal{M}, s) \models A\psi$  iff  $(\mathcal{M}, \pi) \models \psi$  for all  $\pi \in Path_s^{fin}$ , where  $A\psi$  can be expressed by  $\neg(E\psi)$ .

For path formula  $\psi ::= X\phi \mid \phi_1 U_{[a,b]} \phi_2 \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid pf$ , the satisfaction relation for any path  $\pi$  is as follows.

- $(\mathcal{M}, \pi) \models X\phi$  iff  $(\mathcal{M}, \pi(1)) \models \phi$
- $(\mathcal{M}, \pi) \models \phi_1 U_{[a,b]} \phi_2$  iff  $\exists i, a \leq i \leq b \wedge (\mathcal{M}, \pi(i)) \models \phi_2 \wedge \forall j, 0 \leq j < i, (\mathcal{M}, \pi(j)) \models \phi_1$
- $(\mathcal{M}, \pi) \models \neg\phi$  iff  $(\mathcal{M}, \pi) \not\models \phi$
- $(\mathcal{M}, \pi) \models \phi_1 \wedge \phi_2$  iff  $(\mathcal{M}, \pi) \models \phi_1$  and  $(\mathcal{M}, \pi) \models \phi_2$
- $(\mathcal{M}, \pi) \models pf$  iff  $s$  is the first state of  $\pi$ ,  $(\mathcal{M}, s) \models pf$

Using HPTCTS-TL, it is possible to construct two types of queries on the given HPTCTS model  $\mathcal{M}$ , namely, qualitative queries and quantitative queries.



1. Qualitative queries: Query the set of states of  $\mathcal{M}$  satisfying the formulas  $P_{\sim p}(\varphi)$ ,  $C_{\sim c}(\varphi)$ , and  $T_{\sim t}(\varphi)$ .
2. Quantitative queries:  $\varphi$  is computed and solved as follows:
  - $P_{=?}(\varphi) = Pr\{R \subseteq Path_s^{fin} \mid \forall \pi_i \in R, (M, \pi_i) \models \psi\}$   
 $P_{max=?}(\varphi) = max(\{Pr(Cyl(\pi_i)) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$   
 $P_{min=?}(\varphi) = min(\{Pr(Cyl(\pi_i)) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$
  - $C_{max=?}(\varphi) = max(\{CumCost(\pi_i) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$   
 $C_{min=?}(\varphi) = min(\{CumCost(\pi_i) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$
  - $T_{max=?}(\varphi) = max(\{CumTime(\pi_i) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$   
 $T_{min=?}(\varphi) = min(\{CumTime(\pi_i) \mid \pi_i \in R, (M, \pi_i) \models \psi\})$

The quantitative query operation  $P_{=?}(\cdot)$  can also have a wider range of metrics, including probabilistic metrics for average execution (waiting) time and worst-case execution (waiting) time. The set of states for the system at either the average execution time or the worst-case execution time can be obtained through model checking, and then solved according to the set of paths and probabilistic metric operations based on the composition of the set of states.

Functional requirements are mainly used to determine whether the system can meet the expected working requirements under normal conditions, that is, to verify whether the system meets specific requirements under the assumption that all components of the system are working as expected. The safety characteristic refers to whether the system can continue to operate without causing system failure in the case of failure, that is, in the case of one or more components failure, verifying whether the system can continue to operate will not lead to system-level failure. In this work, we mainly focus on the quantitative queries of HPTCTS model for CPS. We assume that Fig. 3 meets the following functional and safety requirements, and calculate the probability, cost, and time that  $\mathcal{M}$  satisfies: within 60 time units, what is the probability that the water level in the water tank is greater than *high\_level* but not greater than *full*, causing the alarm device to be turned on? What is the minimum cost consumed by the system after this event? What is the minimum execution time? The above description is expressed using the formula  $\varphi$  as follows:

- (1)  $P_{=?}(true \ U_{[0,60]} high\_level < h < full \wedge Warning = true)$ .
- (2)  $C_{min=?}(true \ U_{[0,60]} high\_level < h < full \wedge Warning = true)$
- (3)  $T_{min=?}(true \ U_{[0,60]} high\_level < h < full \wedge Warning = true)$

We give a detailed description of the syntax and semantics of HPTCTS-TL, which is suitable for describing properties of system performance and reliability for states and transition paths in HPTCTS model. Referring to the theorem and proof of the equivalence relation between PTCL and CTL, the probability formula  $P_{p \geq 0}(\psi)$  for HPTCTS-TL is equivalent to the CTL formula  $E\psi$  [6]. However, since formulas  $C_{\sim c}(\phi_1 U_{[a,b]} \phi_2)$  and  $T_{\sim t}(\phi_1 U_{[a,b]} \phi_2)$  describe the cost and time properties of the system, although they have some relation with CTL formulas, there is no direct equivalence relation. Moreover, HPTCTS-TL offers a rich expression capability for system characteristics and can simultaneously capture various quantitative aspects of

HPTCTS model such as data constraints, time, probability and cost. We can see that HPTCTS-TL has stronger expressive power than PCTL and CTL.

## 5.2 The symbolization of HPTCTS-TL

The properties of PCTL and HPTCTS-TL have been introduced in the previous section. For the symbolization of HPTCTS-TL properties, we can refer to the symbolization of CTL properties.

The symbolization of HPTCTS-TL properties can be referred to the CTL property symbolization. In the CTL property description, the main timing operators are  $EX$ ,  $EU$ ,  $AX$  and  $AU$ , where  $AX$  and  $AU$  can be expressed by  $EX$  and  $EU$ , respectively [7]. Although the basic operator  $EX$  cannot be represented by an immovable point, it is known from the semantics of  $EX$  that  $EXf = EV'(f \wedge R(V, V'))$ .  $V$  is a set of propositional variables, each variable in  $V$  is a formula,  $f$  and  $g$  are formulas,  $\neg f$ ,  $f \vee g$ , and  $f \wedge g$  are also formulas.  $R$  is the Boolean function  $R(V, V')$  representing the state transition relationship of system model  $\mathcal{M}$ , and it is complete, that is, for each state  $s \in S$ , there is at least one state  $s' \in S$ , so that  $(s, s') \models R(V, V')$ . The symbolization of the formula  $E[fUg]$  can be denoted as  $g \vee (f \wedge EXE[fUg])$ .

The formula  $\phi_1 U_{[a,b]} \phi_2$  can be represented as follows:

$$\phi_1 U_{[a,b]} \phi_2 = \begin{cases} \phi_1 \wedge X(\phi_1 U_{[a-1,b-1]} \phi_2), & \text{if } a > 0, b \geq a \\ \phi_2 \vee (\phi_1 \wedge X(\phi_1 U_{[0,b-1]} \phi_2)), & \text{if } a = 0, b > 0 \\ \phi_2, & \text{if } a = 0, b = 0 \end{cases} \quad (1)$$

The formula  $E(\phi_1 U_{[a,b]} \phi_2)$  can be represented as follows:

$$E(\phi_1 U_{[a,b]} \phi_2) = \begin{cases} \phi_1 \wedge EXE(\phi_1 U_{[a-1,b-1]} \phi_2), & \text{if } a > 0, b \geq a \\ \phi_2 \vee (\phi_1 \wedge EXE(\phi_1 U_{[0,b-1]} \phi_2)), & \text{if } a = 0, b > 0 \\ \phi_2, & \text{if } a = 0, b = 0 \end{cases} \quad (2)$$

The computation procedure of formulas  $\phi_1 U_{[a,b]} \phi_2$  and  $E(\phi_1 U_{[a,b]} \phi_2)$  is implemented through the operation of ORBDD in the symbolic model checking algorithm [7, 28], and for the quantitative querying of HPTCTS-TL formulas, we will refer to such an implementation in order to obtain the corresponding results. HPTCTS-TL has a similar expression to the relationship between PCTL and CTL with  $P_{\geq 0}(\phi_1 U_{[a,b]} \phi_2) = E(\phi_1 U_{[a,b]} \phi_2)$  and then obtain the set  $S$  of HPTCTS satisfying the formula by calculating the symbolized formula (2), and finally we calculate the formula according to the relevant definition.

### 5.3 Quantitative evaluation

In this paper, for the HPTCTS-TL quantitative evaluation problem on the HPTCTS model, including state space explosion and performance analysis, we solve the problem by combining the methods used in the symbolic model checking and the PCTL formula probability model checking process. In the process of quantitative evaluation, we conduct two main studies:

(1) Solve for the set of states  $Sat(\varphi) = \{s \in S \mid s \models \varphi\}$  that satisfy the formula  $\varphi$  based on symbolic model checking.

We first deal with the formula  $\varphi'$  to obtain the corresponding formula  $\varphi$ . The transition relationship of HPTCTS represented by ORBDD, which is used to get  $Sat(\varphi)$  based on symbolic model checking. However, the output of the symbolic model checking algorithm is the set of Boolean expressions for all states that satisfy the formula  $\varphi$ , which is denoted by ORBDD, obtained by the solution of the function *Check* [7, 30].

For finite state systems, model checking is decidable, that is, it can be determined automatically in a limited time by a computer program. This paper focuses on quantitative evaluation for symbolic model checking of integer type constraint in a finite domain.

**Definition 13** (*Finite domain mode*) A decidable subset of HPTLTS with the system model is defined so that the value of each variable is limited in the mode corresponding to each state, condition variables, and time constraint. A mode  $Mode = [s_0, \dots, s_n \mid v_1, \dots, v_m \mid t]$ , the value of  $v_i$  is set to a finite number,  $i = 1, \dots, m$ . Where  $m \in \mathbb{Z}^+ \wedge m \leq \mathbb{Z}_{max}$ ,  $n \in \mathbb{Z} \wedge n \leq \mathbb{Z}_{max}$ ,  $t \in \mathbb{Z} \wedge t \leq \mathbb{Z}_{max}$ ,  $\mathbb{Z}^+$  is a positive integer field and  $\mathbb{Z}_{max}$  is a positive integer in  $\mathbb{Z}^+$ .

The basic process is shown in Algorithm 1.

---

**Algorithm 1** Solve for the set of states  $Sat(\varphi)$ 


---

**Input:** HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$

HPTCTS-TL quantitative queries formula  $\varphi$ ,  $\varphi = P_{=?}(\varphi) \mid P_{max=?}(\varphi)$

$\mid P_{min=?}(\varphi) \mid C_{max=?}(\varphi) \mid C_{min=?}(\varphi) \mid T_{max=?}(\varphi) \mid T_{min=?}(\varphi)$

$\varphi ::= E\psi \mid A\psi$

**Output:**  $Sat(\varphi) = \{s \in S \mid s \models \varphi\}$

**for** each state  $s \in S$  **do**

$f_s \leftarrow s : (x_1, x_2, \dots, x_n)(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$

$f_{s'} \leftarrow s : (x'_1, x'_2, \dots, x'_n)(\bar{x}'_1, \bar{x}'_2, \dots, \bar{x}'_n)$

**end for**

**for** each transition  $\rightarrow$  **do**

$f_{gcon} \leftarrow gcon : (y_1, y_2, \dots, y_k)$

$f_{act} \leftarrow act : (z_1, z_2, \dots, z_r)$

$ORBDD.add\_expr((f_{gcon}, f_{act}, f_s) \rightarrow (1))$  //Call the *add\_expr* method of ORBDD to generate a mapping relationship between three nodes and one terminal node.

**end for**

$ORBDD_{sat} = Check(ORBDD, \varphi)$

$Sat(\varphi) \leftarrow \emptyset$

$states \leftarrow [s_0]$  // Initialize a list of states with the initial state.

**while**  $states$  non-empty **do**

$s \leftarrow$  remove a state from  $states$

**if**  $ORBDD_{sat}.is\_expr(f_s)$  **then**

$Sat(\varphi) \leftarrow Sat(\varphi) \cup \{s\}$

**end if**

**for** each successor  $f_{s'}$  of  $f_s$  in  $ORBDD_{sat}$  **do**

**if**  $ORBDD_{sat}.is\_expr(f_{s'})$  **then**

add  $s'$  to  $states$

**end if**

**end for**

**end while**

**return**  $Sat(\varphi)$

---

## (2) Quantitative queries

Based on  $Sat(\varphi)$ , the probability, time and cost of the formula  $\varphi'$  are solved to analyze the performance of HPTCTS. The corresponding solution process is completed by the following algorithm. In the following algorithm we only solve quantified queries with path formula  $\varphi = E(\phi_1 U_{[a,b]} \phi_2) \mid A(\phi_1 U_{[a,b]} \phi_2)$ .

Algorithm 2 computes the probability that satisfying formula  $E(\phi_1 U_{[a,b]} \phi_2)$  or  $A(\phi_1 U_{[a,b]} \phi_2)$ .

---

**Algorithm 2** Compute  $P_{=?}(\varphi)$ 


---

**Input:** HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$ ;

$P_{=?}(\varphi)$ ;  $a, b$ : integers in  $[0, \infty)$ , and  $a \leq b$ .

**Output:**  $p$ : the probability of satisfying  $\varphi$  starting from state  $s_0$ .

```

if  $a = 0, b = 0$  then
  if  $\varphi = E(\phi_1 U_{[a,b]} \phi_2)$  then
     $R = \{\pi_j \in path_s^{fin} \mid \forall \pi_j = (s_0, \dots, s_n), \exists i, a \leq i \leq b, s_i \in Sat(\phi_2),$ 
       $\forall (i-1), s_{i-1} \in Sat(\phi_2)\}$ 
     $p = Pr(R)$ 
  end if
  if  $\varphi = A(\phi_1 U_{[a,b]} \phi_2)$  then
    if  $\forall s_i \notin Sat(\varphi)$  then
       $p = 0$ 
    else if  $\forall \pi_j \models \phi_1 U_{[a,b]} \phi_2$  then
       $R = \{\pi_j \in path_s^{fin} \mid \forall \pi_j = (s_0, \dots, s_n), s_i \in Sat(\phi_2)\}$ 
    end if
  end if
else if  $a \geq 0, b > 0$  then
  if  $\varphi = E(\phi_1 U_{[a,b]} \phi_2)$  then
     $R = \{\pi_j \in path_s^{fin} \mid \exists \pi_j = (s_0, \dots, s_n), \exists i, a \leq i \leq b, s_i \in$ 
       $Sat(\phi_2), \forall (i-1), s_{i-1} \in Sat(\phi_1)\}$ 
     $p = Pr(R)$ 
  end if
  if  $\varphi = A(\phi_1 U_{[a,b]} \phi_2)$  then
    if  $\forall s_i \notin Sat(\varphi)$  then
       $p = 0$ 
    else if  $\forall \pi_j \models \phi_1 U_{[a,b]} \phi_2$  then
       $R = \{\pi_j \in path_s^{fin} \mid \forall \pi_j = (s_0, \dots, s_n), s_i \in Sat(\phi_2)\}$ 
       $p = Pr(R)$ 
    end if
  end if
end if
return  $p$ 

```

---

Algorithm 3 computes the minimum probability that satisfying formula  $E(\phi_1 U_{[a,b]} \phi_2)$  or  $A(\phi_1 U_{[a,b]} \phi_2)$ .

---

**Algorithm 3** Compute  $P_{min=?}(\varphi)$

---

**Input:** HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$ ;  
 $P_{min=?}(\varphi)$ ;  $a, b$ : integers in  $[0, \infty)$ , and  $a \leq b$ .  $\infty$  denotes  
a sufficiently large real number.

**Output:**  $p_{min}$ : the minimum probability of satisfying  $\varphi$  starting from state  $s_0$ .

**if**  $a = 0, b = 0$  **then**

**for**  $\forall \pi_j = (s_0, \dots, s_n) \in path_s^{fin}$  **do**  
**if**  $s_i \in Sat(\varphi) \wedge Pr\{Cyl(\pi_j)\} < p_{min}$  **then**  
 $p_{min} = Pr\{Cyl(\pi_j)\}$   
**else if**  $\forall s_i \notin Sat(\varphi)$  **then**  
 $p_{min} = 0$   
**end if**

**end for**

**else if**  $a \geq 0, b > 0$  **then**

**for**  $\forall \pi_j = (s_0, \dots, s_n) \in path_s^{fin}$  **do**  
**if**  $\exists i, a \leq i \leq b, s_i \in Sat(\phi_2), \forall (i-1), s_{i-1} \in Sat(\phi_1), Pr\{Cyl(\pi_j)\} <$   
 $p_{min}$  **then**  
 $p_{min} = Pr\{Cyl(\pi_j)\}$   
**else if**  $\forall s_i \notin Sat(\varphi)$  **then**  
 $p_{min} = 0$   
**end if**

**end for**

**end if**

**return**  $p_{min}$

---

Algorithm for computing  $P_{max=?}(\varphi)$  is similar to  $P_{min=?}(\varphi)$ , we only need to replace “ $p_{max}$ ” with “ $p_{min}$ ”, “if  $Pr\{Cyl(\pi_j)\} > p_{max}$ ” with “if  $Pr\{Cyl(\pi_j)\} < p_{min}$ ”, and “return  $p_{max}$ ” with “return  $p_{min}$ ”.

Algorithm 4 computes the minimum cumulative cost that satisfying formula  $E(\phi_1 U_{[a,b]} \phi_2)$  or  $A(\phi_1 U_{[a,b]} \phi_2)$ .  $C_{max=?}(\varphi)$  can be computed the similar as  $C_{min=?}(\varphi)$ .

---

**Algorithm 4** Compute  $C_{min=?}(\varphi)$

---

**Input:** HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$ ;

$C_{min=?}(\varphi)$ ;  $a, b$ : integers in  $[0, \infty)$ , and  $a \leq b$ .

**Output:**  $c_{min}$ : the minimum cost of satisfying  $\varphi$  starting from state  $s_0$ .

```

if  $a = 0, b = 0$ 
  for  $s_i \in Sat(\varphi)$  do
    if  $\rho(s_i) < c_{min}$  then
       $c_{min} = \rho(s_i)$ 
    end if
  end for
if  $s_i \notin Sat(\varphi)$ 
   $c_{min} = \infty$ 
end if
else if  $a \geq 0, b > 0$ 
  for  $\forall \pi_j = (s_0, \dots, s_n) \in path_s^{fin}, \exists i, a \leq i \leq b, s_i \in Sat(\phi_2), \forall (i - 1), s_{i-1} \in Sat(\phi_1)$  do
    if  $CumCost(\pi_j) < c_{min}$  then
       $c_{min} = CumCost(\pi_j)$ 
    end if
  end for
if  $\forall s_i \notin Sat(\varphi)$  then
   $c_{min} = \infty$ 
end if
end if
return  $c_{min}$ 

```

---

Algorithm 5 computes the minimum cumulative execution time that satisfying formula  $E(\phi_1 U_{[a,b]} \phi_2)$  or  $A(\phi_1 U_{[a,b]} \phi_2)$ . Algorithm for computing  $T_{max=?}(\varphi)$  is similar to  $T_{min=?}(\varphi)$ , we only need to replace “ $t_{max}$ ” with “ $t_{min}$ ,” “if  $CumTime(\pi_j) > t_{max}$ ” with “if  $CumTime(\pi_j) < t_{min}$ ,” and “return  $t_{max}$ ” with “return  $t_{min}$ .”

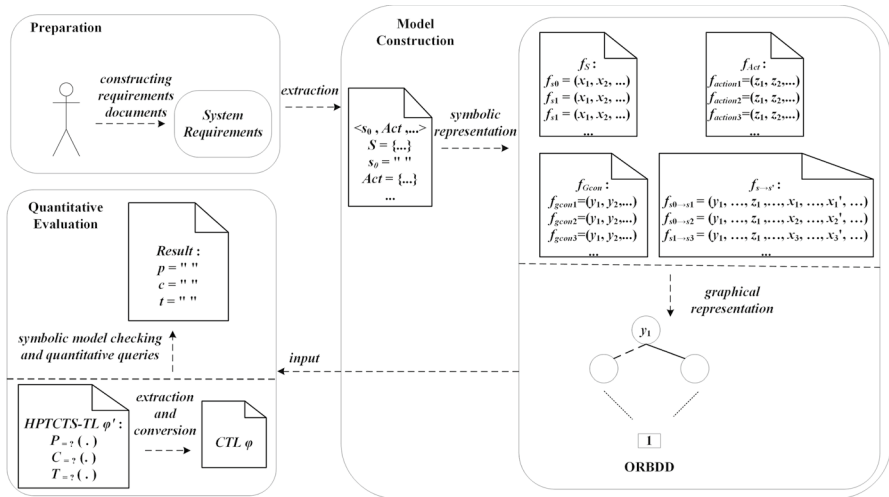


Fig. 5 The workflow of the prototype system

**Algorithm 5** Compute  $T_{min=?}(\varphi)$

**Input:** HPTCTS  $\mathcal{M} = (S, s_0, \Lambda, C, Var, F, I, E, P, L)$ ;

$T_{min=?}(\varphi)$ ;  $a, b$ : integers in  $[0, \infty)$ , and  $a \leq b$ .

**Output:**  $t_{min}$ : the minimum time of satisfying  $\varphi$  starting from state  $s_0$ .

```

if  $a = 0, b = 0$ 
  for  $s_i \in Sat(\varphi)$  do
    if  $t_i < t_{min}$  then
       $t_{min} = t_i$ 
    end if
  end for
if  $s_i \notin Sat(\varphi)$ 
   $t_{min} = \infty$ 
end if
else if  $a \geq 0, b > 0$ 
  for  $\forall \pi_j = (s_0, \dots, s_n) \in path_s^{fin}, \exists i, a \leq i \leq b, s_i \in Sat(\phi_2), \forall (i - 1), s_{i-1} \in Sat(\phi_1)$  do
    if  $CumTime(\pi_j) < t_{min}$  then
       $t_{min} = CumTime(\pi_j)$ 
    end if
  end for
if  $\forall s_i \notin Sat(\varphi)$  then
   $t_{min} = \infty$ 
end if
end if
return  $t_{min}$ 

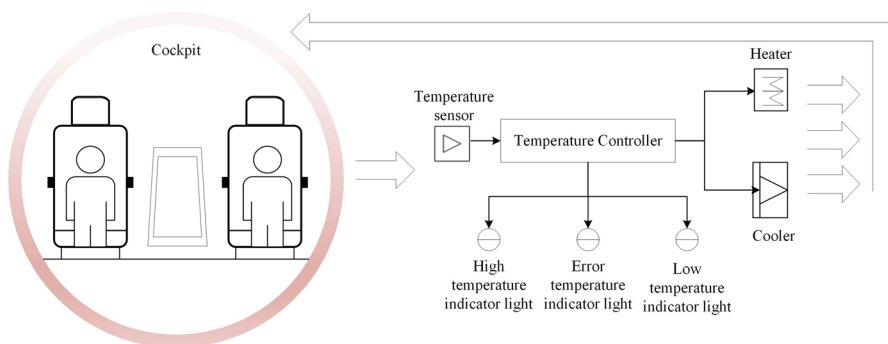
```



Algorithms 1–5 can be utilized to quantitative query the HPTCTS-TL formulas. Due to the fact that CPS modeled by HPTCTS is a hybrid system, we define a bounded domain to solve the performance metrics of the HPTCTS model based on symbolic model checking. In this case, the results computed by Algorithm 2–5 are approximate rather than completely accurate. In addition, if any of the states do not satisfy the system requirements and the system cannot automatically fix them, the time and cost consumed to run the system can be immeasurably large. Nevertheless, the approach presented in this paper still has the following advantages for quantitatively evaluating the performance of CPS.

First, we symbolically represent the constructed HPTCTS model of the CPS as a ORBDD of complexity  $O(m \log(k + r + n))$  based on a Boolean formula, where  $m$  is the number of non-zero terminal nodes, the  $k, r,$  and  $n$  is the dimension of the Boolean vector. Similarly, we design HPTCTS-TL symbolic representations that can portray CPS properties, which will simplify the system model and drastically reduce the stored system state space. Second, symbolic model checking is a formal verification method that effectively avoids memory state explosion, and it has been shown that the method can handle systems with up to  $10^{120}$  states. The real-world CPS can be a large-scale system accompanied by concurrent, hybrid, real-time, and stochastic characteristics. Algorithm 1 implements the symbolic model checking based on the HPTCTS model, which has a computational complexity of  $|\mathcal{M}| \times O(2^{|\phi|})$ . In addition, for the quantitative querying requirements of CPS, we can use the corresponding algorithms to compute and obtain the values of the corresponding performance metrics that enable qualitative evaluation of CPS.

Therefore, it is undoubtedly difficult to ensure precise quantification of performance metrics for large and complex systems. However, the approximate values of performance metrics can meet the needs of system developers and users. Based on symbolic model checking, the quantitative evaluation of CPS can efficiently solve its performance metrics, and the results are close to the performance requirements of system developers and users. Based on the obtained metrics, we can perform performance analysis of the HPTCTS model for CPS, and formulate a plan for the next step of improvement for the system.



**Fig. 6** Aircraft cockpit temperature control system

## 6 Design of the prototype system

This paper designs a prototype system to implement performance modeling and quantitative evaluation of CPS on a Windows 10 machine with an Intel Pentium CPU running at 3.20 GHz and 8GB of memory. Figure 5 describes the workflow of the prototype system.

### (1) Preparation

In the preparation phase, we first construct a text file based on the user's requested CPS functionalities and related attributes. Then, we extract the contents of the file to facilitate the model construction stage.

### (2) Model construction

During the CPS model construction stage, we extract the state, variables, constraints, behavior of the state transition process, transition probabilities, and operational costs from the document content prepared in the previous stage. With these elements as a basis, we construct the HPTCTS model of the CPS. The prototype system can symbolically represent the HPTCTS model by describing it as a Boolean expression and then expressing it in the form of ORBDD. When quantitatively evaluating CPS, it is necessary to first verify that the system functions correctly.

### (3) Quantitative evaluation

Taking into account the practical situation of the system, we designed functional requirements for different scenarios and calculated performance metrics to meet these requirements. We provided the prototype system with a quantitative query formula for HPTCTS-TL and calculated the satisfiable state set of the formula using Algorithm 1, and the performance metrics satisfying the formula using Algorithms 2–5.

In our actual implementation, the prototype system is mainly implemented by the Java programming language. We will demonstrate the effectiveness of the performance modeling and quantitative evaluation for CPS in Sect. 7. According to the verification results, it will be helpful to realize the development of CPS with multiple characteristics and behaviors.

## 7 Example analysis

In this section, an aircraft cockpit temperature control system will be modeled with HPTCTS, and its performance attributes will be characterized with HPTCTS-TL. Based on the quantitative evaluation algorithm for HPTCTS model, we calculate the HPTCTS-TL formula to perform performance analysis on the system.

### 7.1 System description

We carried out an investigation on a CPS case (a teaching and representative aircraft cockpit temperature control system [31], as depicted in Fig. 6) using HPTCTS for modeling, and quantitatively assessed the performance of the HPTCTS model. The aircraft cockpit temperature control system is a CPS with behaviors such as discrete, continuous, concurrent, and random. It is composed of a set of sensor, controller, indicator light, heater, and cooler. These components can receive and process external information, and then control and adjust the temperature inside the cockpit to meet the pilot’s comfortable driving experience. The system can be represented as

$$\text{System} = \text{Sensor} \parallel \text{Controller} \parallel \text{Indicator Light} \parallel \text{Heater} \parallel \text{Cooler}.$$

Temperature sensor will continuously monitor the temperature inside the cockpit and send temperature signals to the temperature controller. The temperature controller will control the heater and cooler to maintain the desired temperature.

If the temperature inside the cockpit exceeds 30 °C, the cooler will activate to lower the temperature. Conversely, if the temperature drops below 22 °C, the heater will activate to raise it. The temperature inside the cockpit should be kept between 22 °C and 30 °C. Furthermore, the high-temperature and low-temperature indicator lights will illuminate if the temperature gets too hot or too cold. Additionally, the error indicator light will turn on in case of a system malfunction, alerting the pilot to take prompt action. Over time, fluctuations in temperature can cause changes in aircraft fuel consumption, which can impact the performance and safety of aircraft. Therefore, the temperature control system inside the cockpit is crucial.

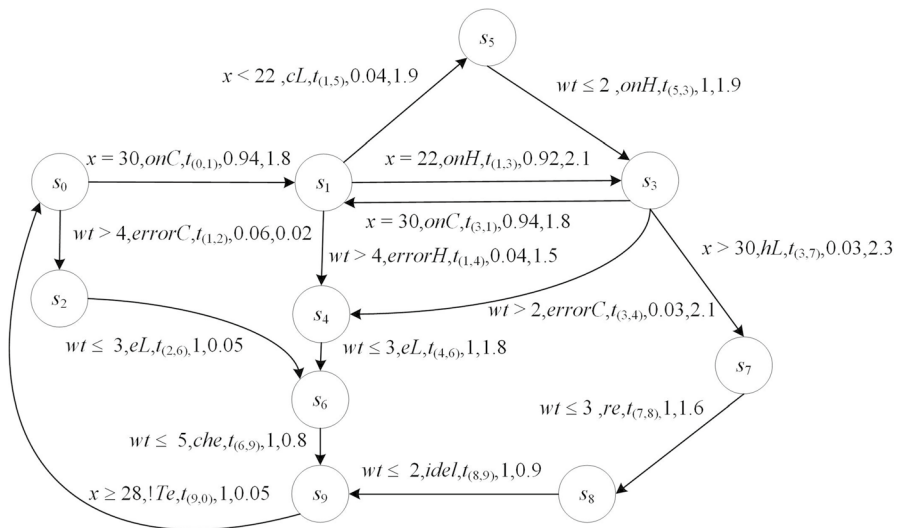


Fig. 7 HPTCTS of aircraft cockpit temperature control system

**Table 3** Encoding the set of states of HPTCTS  $\mathcal{M}$  ( $\mathcal{M}$  in Fig. 7)

$s \in \mathcal{S}$	$s_0$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	$s_9$	$e(s_i)$
$s_0$	0	1	1	0	0	0	0	0	0	0	0000
$s_1$	0	0	0	1	1	1	0	0	0	0	0001
$s_2$	0	0	0	0	0	0	1	0	0	0	0010
$s_3$	0	1	0	0	1	0	0	1	0	0	0100
$s_4$	0	0	0	0	0	0	1	0	0	0	0101
$s_5$	0	0	0	1	0	0	0	0	0	0	0110
$s_6$	0	0	0	0	0	0	0	0	0	1	0111
$s_7$	0	0	0	0	0	0	0	0	1	0	1000
$s_8$	0	0	0	0	0	0	0	0	0	1	1001
$s_9$	1	0	0	0	0	0	0	0	0	0	1010

**Table 4** Encoding the transition relations of HPTCTS  $\mathcal{M}$  ( $\mathcal{M}$  in Fig. 7)

$\rightarrow$	$y$	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x'_1$	$x'_2$	$x'_3$	$x'_4$
$s_0 \rightarrow s_1$	(0000)	<i>onC</i> (0000)	0	0	0	0	0	0	0	1
$s_0 \rightarrow s_2$	(0001)	<i>errorC</i> (0001)	0	0	0	0	0	0	1	0
$s_1 \rightarrow s_3$	(0010)	<i>onH</i> (0010)	0	0	0	1	0	1	0	0
$s_1 \rightarrow s_4$	(0001)	<i>errorH</i> (0011)	0	0	0	1	0	1	0	1
$s_1 \rightarrow s_5$	(0100)	<i>cL</i> (0100)	0	0	0	1	0	1	1	0
$s_2 \rightarrow s_6$	(0101)	<i>eL</i> (0101)	0	0	1	0	0	1	1	1
$s_3 \rightarrow s_1$	(0000)	<i>onC</i> (0000)	0	1	0	0	0	0	0	1
$s_3 \rightarrow s_4$	(0110)	<i>errorC</i> (0001)	0	1	0	0	0	1	0	1
$s_3 \rightarrow s_7$	(0111)	<i>hL</i> (0110)	0	1	0	0	1	0	0	0
$s_4 \rightarrow s_6$	(0101)	<i>eL</i> (0101)	0	1	0	1	0	1	1	1
$s_5 \rightarrow s_3$	(1000)	<i>onH</i> (0010)	0	1	1	0	0	1	0	0
$s_6 \rightarrow s_9$	(1001)	<i>che</i> (0111)	0	1	1	1	1	0	1	0
$s_7 \rightarrow s_8$	(0101)	<i>re</i> (1000)	1	0	0	0	1	0	0	1
$s_8 \rightarrow s_9$	(1000)	<i>idel</i> (1001)	1	0	0	1	1	0	1	0
$s_9 \rightarrow s_0$	(1010)	<i>!Te</i> (1010)	1	0	1	0	0	0	0	0

We set the initial state of the system as  $s_0$ , the initial temperature as 28 °C, and  $x$  as the temperature variable,  $wt$  is the time variable, and  $t_{(i,j)}$  is the duration of the transition from state  $s_i$  to  $s_j$ . The cockpit temperature control system for the aircraft can be described as follows.

- (1) At initial state  $s_0$ , the temperature in the cockpit rises at a rate of 0.5 over time. When the temperature inside the cockpit reaches 30 °C, the system executes *onC* action with a probability of 0.94 to open the cooler. The system transitions from state  $s_0$  to  $s_1$ , and the aircraft consumes 1.8 units of fuel. After the cooler is opened, the temperature in the cockpit is reduced at a rate of 2 over time. If the cooler fails to open, the system executes *errorC* action with a probability

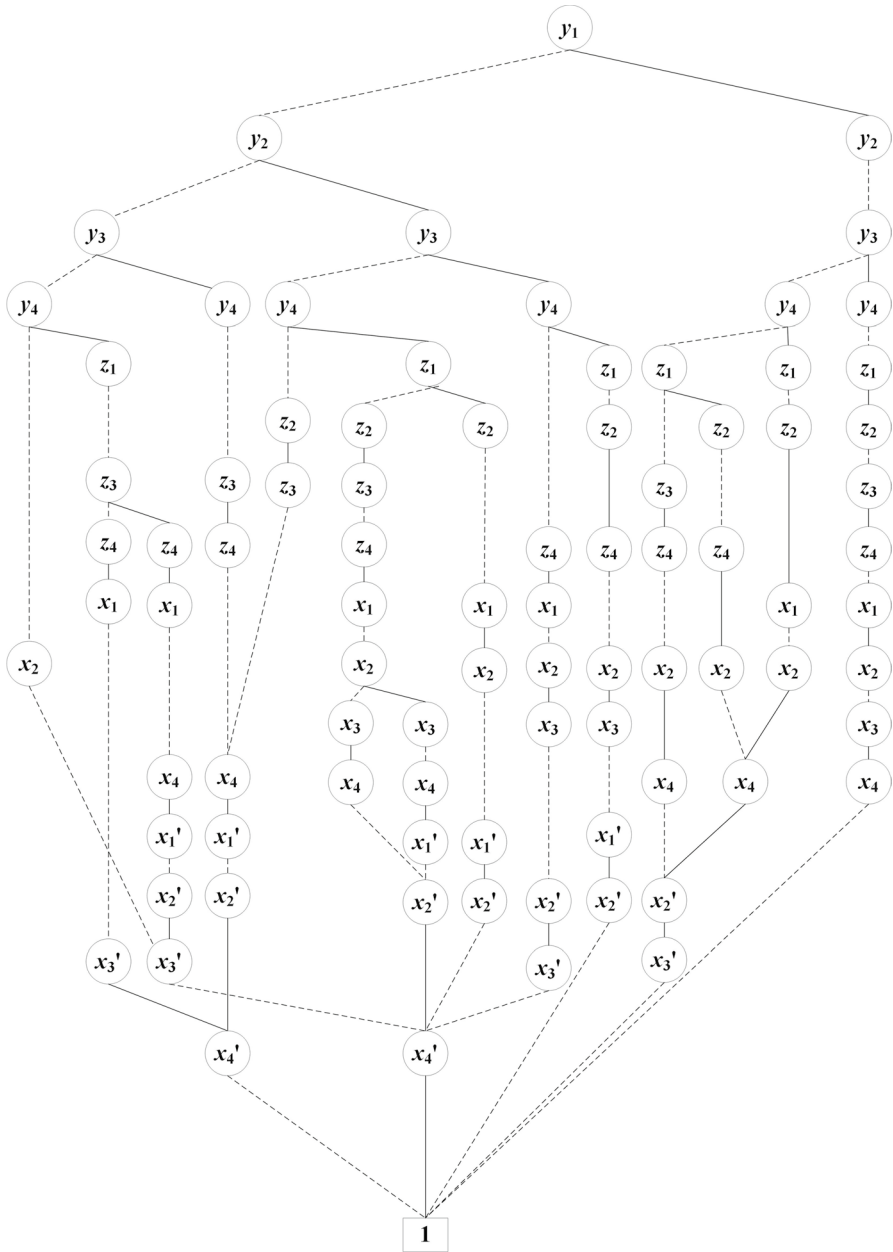


Fig. 8 ORBDD encoding the transition relations of HPTCTS  $\mathcal{M}$  ( $\mathcal{M}$  in Fig. 7)

of 0.06. In this case, the system transitions from state  $s_0$  to  $s_2$ , and the aircraft consumes 0.02 units of fuel. Within 3 time units, the system's error indicator light will be turned on by  $eL$  action. The system transitions from state  $s_2$  to  $s_6$ , and the aircraft consumes 0.05 units of fuel. After the error indicator light is turned on, the system enters the detection state for 5 time units, i.e., the system transitions from state  $s_6$  to  $s_9$ . The system executes the action of sending the ambient temperature,  $!Te$ , at state  $s_9$  and transitions to state  $s_0$ , where the aircraft consumes 0.05 units of fuel.

- (2) When the temperature inside the cockpit drops below 22 °C, the system executes  $onH$  action with a probability of 0.92 to open the heater. After the heater is turned on, the temperature in the cockpit gradually increases at a rate of 4 over time. The system enters state  $s_3$ , which is transitioned from  $s_1$ , and the aircraft consumes 2.1 units of fuel. However, if an error occurs when the heater is turned on at state  $s_1$ , the system executes  $errorH$  action with a probability of 0.04. In this case, the system transitions from state  $s_1$  to  $s_4$ , and the aircraft consumes 1.5 units of fuel. Then, the system continues to transition to  $s_6$ . As the heater cannot be turned on, the temperature inside the cockpit will remain below 22 °C. The system also executes  $cL$  action with a probability of 0.04, and the low-temperature indicator light is turned on. At this point, the system transitions from state  $s_1$  to  $s_5$ , and the aircraft consumes 1.9 units of fuel. Within 2 time units, the system continues to execute  $onH$  action to turn on the heater, and the system transitions from state  $s_5$  to  $s_3$ , where the aircraft also consumes 1.9 units of fuel.
- (3) When the system executes the  $onC$  action to open the cooler, there is a 0.03 probability of encountering an error and executing the  $errorC$  action. In this case, the system transitions from  $s_3$  to  $s_4$ , consuming 2.1 units of fuel before continuing to transition to  $s_6$ . If the temperature inside the cockpit is higher than 30 °C and the cooler cannot be opened, the system executes  $hL$  action with a 0.03 probability, causing the high-temperature indicator light to turn on and transitioning from  $s_3$  to  $s_7$ , consuming 2.3 units of fuel. Within 3 time units, the system executes  $re$  action to turn off the heater and release heat, transitioning from  $s_7$  to  $s_8$ , and consuming 1.6 units of fuel. The system then enters the detection state by executing the  $idel$  action within 2 time units, transitioning from  $s_8$  to  $s_9$ , and consuming 0.9 units of fuel before sending the ambient temperature and transitioning back to  $s_0$ .

### 7.2 Modeling system using HPTCTS and symbolic representation

For the description of the above system, the combined system model HPTCTS is shown in Fig. 7.

For the state set  $S = \{s_0, \dots, s_9\}$ , the corresponding specified Boolean variables order  $x_1, x_2, x_3, x_4$ . The state can be represented as follows:

$$\begin{aligned}
 f_{s_0} &= \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4, f_{s_1} = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4, f_{s_2} = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4, f_{s_3} \\
 &= \bar{x}_1 \bar{x}_2 x_3 x_4, f_{s_4} = \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4, f_{s_5} = \bar{x}_1 x_2 \bar{x}_3 x_4, f_{s_6} \\
 &= \bar{x}_1 x_2 x_3 \bar{x}_4, f_{s_7} = \bar{x}_1 x_2 x_3 x_4, f_{s_8} = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4, f_{s_9} \\
 &= x_1 \bar{x}_2 \bar{x}_3 x_4
 \end{aligned}$$

**Table 5** The numerical results

Num	Quantitative querye	Result	Symbolic		Explicit	
			Memory (KB)	Runtime (s)	Memory (KB)	Runtime (s)
(1)	$P_{=i}(trueU_{[0,20]}(LTLight = on \wedge x < 22))$	$p = 0.07$	5.27	0.75	13.57	1.07
	$C_{min=i}(trueU_{[0,20]}(LTLight = on \wedge x < 22))$	$c_{min} = 18.5$	3.89	0.68	12.64	0.95
	$T_{min=i}(trueU_{[0,20]}(LTLight = on \wedge x < 22))$	$t_{min} = 10$	6.08	0.62	11.51	1.13
(2)	$P_{=i}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$	$p = 0.16$	2.91	0.63	8.03	1.09
	$C_{min=i}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$	$c_{min} = 0.39$	3.18	0.71	7.18	1.33
	$T_{min=i}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$	$t_{min} = 9$	3.05	0.65	7.95	1.28
(3)	$P_{=i}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$	$p = 0.03$	9.58	0.96	25.92	1.15
	$C_{min=i}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$	$c_{min} = 32.8$	8.96	1.73	23.06	2.08
	$T_{min=i}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$	$t_{min} = 17$	8.33	1.46	17.39	2.27

Tables 3 and 4 summarize the process of encoding the states and transition relations function of HPTCTS for aircraft cockpit temperature control system, and the corresponding ORBDD is shown in Fig. 8.

### 7.3 Quantitative evaluation of HPTCTS model for aircraft cockpit temperature control system

We will use the quantifiable evaluation algorithm in Sect. 5.3 to calculate the HPTCTS-TL formula that must be satisfied by the aircraft cockpit temperature control system, the minimum cumulative cost and time of the system will be given priority consideration. Below we give some quantifiable evaluation formulas of HPTCTS-TL, and the numerical results of each formula are shown in Table 5.

(1) What is the probability that the temperature falls below 22 °C when the cockpit temperature control system triggers a low-temperature indicator within 20 units of time? And what are the minimum cost and time required for this event to occur? The formula can be expressed as:

- $P_{=?}(trueU_{[0,20]}(LTlight = on \wedge x < 22))$
- $C_{min=?}(trueU_{[0,20]}(LTlight = on \wedge x < 22))$
- $T_{min=?}(trueU_{[0,20]}(LTlight = on \wedge x < 22))$

(2) What is the probability of error indicator turning on due to heater or cooler switching failure when cockpit temperature is not higher than 22 °C or not lower than 30 °C within 20 units of time? And what are the minimum cost and time consumption of the system and execution time of this scenario? The formula can be expressed as:

- $P_{=?}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$
- $C_{min=?}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$
- $T_{min=?}(trueU_{[0,20]}((cooler \vee heater = false) \wedge Elight = on))$

(3) The cockpit temperature fluctuates between 22°C and 30°C within 50 time units. When the temperature exceeds 30°C, how likely is it that the radiator opens to cool the cockpit below 30°C? What is the minimum cost and execution time of this system? The formula can be expressed as:

- $P_{=?}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$
- $C_{min=?}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$
- $T_{min=?}(22 \leq x \leq 30U_{[0,50]}(release = true \wedge x < 30))$

We implemented the performance queries for requirements (1)–(3) according to Fig. 5. In addition, to evaluate our approach, we also based on a non-symbolic (explicit) model checking method to compute the performance metrics of the system. The results of the computation of the quantitative query formulas of the HPTCTS-TL, as well as the runtime and the memory consumption of the method implementation, are shown in Table 5.



In Table 5, we evaluated some performance indicators of the cockpit temperature control system. The system has a low probability of failure, but it still requires further improvement to ensure its reliability for CPS in the aerospace field. Moreover, this system only controls the temperature of the aircraft cockpit, which is a high-cost component for a running aircraft. Therefore, reducing the system cost is also a key task for future work. In this paper, our work is based on Boolean variables to symbolically represent the state space and state transition relationships of the HPTCTS model for CPS with Boolean functions, and store the Boolean functions in ORBDD. The system is quantitatively evaluated by Boolean operations as well as the algorithms that compute quantitative query formulas. Compared to the method that performs model checking by constructing all the reachable states of the system and using an exhaustive search of the state space, that is, the explicit model checking method, our work is lower in terms of runtime and memory usage for the final solution of the quantitative query formulas.

This section shows a typical CPS being modeled in detail by HPTCTS, and ORBDD can symbolically represent the HPTCTS model. Some quantitative indicators of this CPS are inscribed as quantitative solution formulas representing HPTCTS-TL, and quantitative evaluation methods can compute the results of these formulas.

## 8 Related works

The field of CPS research is complex and interdisciplinary, and various formal methods have been developed for modeling and analyzing CPS. There has been extensive research on performance modeling and quantitative evaluation of CPS, resulting in several achievements. Given the complex structure of CPS, researchers have focused their attention on specific characteristics of CPS.

Labeled transition systems (LTS) are a popular formalism used for modeling CPS due to their simplicity and expressiveness. Researchers have proposed several extensions to LTS to model different aspects of CPS such as hybrid, time, probability, and other non-functional properties. In [32], the authors present a hybrid probabilistic

**Table 6** Comparison of related works

References	FM	DB	CB	T	C	P	QE	ASSE
[25]	Markov decision processes	Y	N	N	N	Y	Y	Y
[28]	Probabilistic timed automata	Y	N	Y	N	Y	N	Y
[31]	Probabilistic labeled transition system	Y	Y	Y	N	Y	N	N
[32]	Probabilistic clock labeled transition system	Y	Y	Y	N	Y	Y	N
[33]	Stochastic hybrid automata	Y	Y	Y	N	Y	Y	N
[34]	Discrete-time Markov chains	Y	N	Y	N	Y	Y	Y
our work	HPTCTS	Y	Y	Y	Y	Y	Y	Y

Y: support; N: no support; FM: formal model; DB: discrete behavior; CB: continuous behavior; T: time; C: cost; P: probabilistic; QE: quantitative evaluation; ASSE: alleviate state space explosion

process calculus for modeling and reasoning about CPS, called pCCPS (probabilistic calculus of cyber-physical systems). The dynamics of the calculus is expressed in terms of a probabilistic labeled transition system (pLTS) in the style of Plotkin's SOS. In order to evaluate the CPS more carefully, verification methods for probabilistic metrics on the systems are provided.

The probabilistic extensions of clock constraint specification language (pCSSL) is proposed to describe the clock constraints and probabilistic behavior of CPS in [33]. In addition, the authors consider non-functional properties of the CPS (power and energy consumption) and performance analysis modeling. The semantic model of pCSSL is represented as a probabilistic clock labeled transition system (PCLTS). The statistical model checking was used to the quantitative and qualitative verification of the pCSSL model for CPS.

Based on a stochastic modeling approach, the non-functional properties of the CPS were captured and analyzed in [34], such as reliability, performance, and energy consumption. The authors addressed the cross-validation problem by modeling and evaluating case studies in the railway domain using different formats and tools. A stochastic activity network model for tracking turnout heaters and a stochastic hybrid automaton model developed to evaluate energy consumption and reliability metrics will be evaluated using Mobius and Uppaal SMC. The results obtained will be compared by the researchers to improve their credibility and provide insights into the design and analysis of energy efficient cyber-physical systems.

Model checking exhaustively searches the state space of a system to find all states with satisfiable properties. CPS is a large and complex system, and the process of model checking CPS models is often affected by state space explosion. Many effective solutions have been proposed by researchers to mitigate the problem of model checking. McMillan proposed symbolic model checking to represent system models efficiently, which is one of the successful methods to alleviate state space explosion [20]. Symbolic model checking uses Boolean functions to represent the state set and state transition relations in the system model, and it uses binary decision diagram data structures for storage, which ensures compressed representation and efficient operations of large-scale and structured models. The traditional representation of symbolic model checking has been extended by references such as [21, 35] with multi-terminal binary decision diagrams and matrix diagrams, among others, to further implement symbolic model checking for probabilistic systems. In [28], the authors proposed a symbolic model checking algorithm for probabilistic timed automata, which can verify both qualitative temporal logic properties (corresponding to probability 0 or 1) and quantitative properties (corresponding to any probability), while mitigating the problem of state space explosion in model checking.

The comparison between the present related work and our work is shown in Table 6.

In the aforementioned research related to CPS modeling and verification, researchers have considered its characteristics and behaviors from various aspects. However, it is impossible to model CPS in detail. We comprehensively capture the behavior and characteristics of CPS and consider the non-functional attributes of CPS, such as reliability and performance. We use probabilistic model checking techniques to quantitatively verify them, in order to quantitatively evaluate CPS. Our

research method is an effective supplement to existing methods and has practical application value. Moreover, we can understand that symbolic model checking has alleviated the problem of state space explosion to some extent, symbolic model checking has been verified in several types of systems. We integrate the main ideas of symbolic model checking and probabilistic model checking to comprehensively describe the various behaviors and characteristics of CPS and efficiently perform quantitative evaluation of CPS.

## 9 Conclusion

For CPS with complex structures, high reliability and performance requirements are necessary. We extend LTS to propose HPTCTS, which can model the continuous behavior, probabilistic behavior, concurrent behavior, temporal characteristics, and performance of CPS comprehensively. In order to be able to describe the functional and security properties of HPTCTS and also to calculate the performance metrics that the system needs to satisfy, we propose the HPTCTS-TL. We symbolically represent HPTCTS using ORBDD, and HPTCTS-TL formulas are also symbolically represented as Boolean functions. We use ORBDD and Boolean formulas of temporal logic as inputs for symbolic model checking to mitigate state space explosion when quantitatively evaluating the HPTCTS model. We implement some algorithms to quantitatively evaluate HPTCTS and demonstrate through case studies that our work contributes to the development of a more efficient and comprehensive approach to evaluating CPS performance.

The method proposed in this paper is crucial for the successful deployment of CPS in real-world applications. Furthermore, the proposed method enhances understanding of CPS and provides a better understanding of its functionality, performance, and reliability. The modeling method and temporal logic are proposed in this paper can also be extended to various CPS applications and will help advance research in the field. Although the introduction of symbolic thinking solves the problem of state space explosion, there are still limitations when facing larger-scale systems, and further improvements are needed in the quantification evaluation of CPS methods.

**Acknowledgements** The authors express their gratitude to the editors and referees for their encouragement and constructive comments on this article.

**Author Contributions** Zhen Li contributed to conceptualization, methodology, validation, formal analysis, investigation, resources, data curation, writing -original draft, writing-review & editing, visualization, project administration. Zining Cao contributed to conceptualization, methodology, validation, formal analysis, writing-review & editing, supervision, project administration, funding acquisition. Chao Xing contributed to validation, investigation, writing-review & editing.

**Funding** This work was supported in part by the Aviation Science Foundation of China under Grant 20185152035, in part by the National Natural Science Foundation of China under Grant 61572253, and in part by the Fundamental Research Funds for the Central Universities under Grants NJ2020022 and NJ2019010.

**Data availability** This declaration is not applicable.

## Declarations

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Conflict of interest** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Pivoto DGS, de Almeida LFF, da Rosa Righi R, Rodrigues JJPC, Lugli AB, Alberti AM (2021) Cyber-physical systems architectures for industrial internet of things applications in industry 4.0: a literature review. *J Manuf Syst* 58:176–192. <https://doi.org/10.1016/j.jmsy.2020.11.017>
2. Tyagi AK, Sreenath N (2021) Cyber physical systems: analyses, challenges and possible solutions. *Internet Things Cyber-Phys Syst* 1:22–33. <https://doi.org/10.1016/j.iotcps.2021.12.002>
3. Yaacoub J-PA, Salman O, Noura HN, Kaaniche N, Chehab A, Malli M (2020) Cyber-physical systems security: limitations, issues and future trends. *Microprocessors Microsyst* 77:103201. <https://doi.org/10.1016/j.micpro.2020.103201>
4. Sanislav T, Zeadally S, Mois GD, Fouchal H (2019) Reliability, failure detection and prevention in cyber-physical systems (cpss) with agents. *Concurr Comput: Pract Exp* 31(24):4481. <https://doi.org/10.1002/cpe.4481>
5. Yang C, Sun H, Liu J, Kang J, Yin W, Wang H, Li T (2021) Uncertainty modeling and quantitative evaluation of cyber-physical systems. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), pp 874–883. <https://doi.org/10.1109/COMPSAC51774.2021.00120>
6. Baier C, Katoen J-P (2008) Principles of model checking, vol 26202649
7. Clarke EM, Henzinger TA, Veith H, Bloem R (eds) (2018) Handbook of model checking. Springer, Cham. <https://doi.org/10.1007/978-3-319-10575-8>
8. Rashid A, Hasan O (2021) Formal analysis of the continuous dynamics of cyber-physical systems using theorem proving. *J Syst Architect* 112:101850. <https://doi.org/10.1016/j.sysarc.2020.101850>
9. Fränzle M, Shirmohammadi M, Swaminathan M, Worrell J (2022) Costs and rewards in priced timed automata. *Inf Comput* 282:104656. <https://doi.org/10.1016/j.ic.2020.104656>
10. Zhang X, Li J (2021) Power control for cognitive users of perception layer in complex industrial cps based on dqn. *IEEE Access* 9:25371–25382. <https://doi.org/10.1109/ACCESS.2021.3057911>
11. Tei K, Tahara Y, Ohsuga A (2022) Towards scalable model checking of reflective systems via labeled transition systems. *IEEE Trans Software Eng*. <https://doi.org/10.1109/TSE.2022.3174408>
12. Yang Y, Zu Q, Ke W, Zhang M, Li X (2019) Real-time system modeling and verification through labeled transition system analyzer. *IEEE Access* 7:26314–26323. <https://doi.org/10.1109/ACCESS.2019.2899761>
13. Rao L, Liu S, Peng H (2022) An integrated formal method combining labeled transition system and event-b for system model refinement. *IEEE Access* 10:13089–13102. <https://doi.org/10.1109/ACCESS.2022.3146390>
14. Cleaveland R, Roscoe AW, Smolka SA (2018) Process algebra and model checking. In: Clarke EM, Henzinger TA, Veith H, Bloem R (eds) Handbook of model checking. Springer, Cham, pp 1149–1195. [https://doi.org/10.1007/978-3-319-10575-8\\_32](https://doi.org/10.1007/978-3-319-10575-8_32)
15. Baier C, Haverkort BR, Hermanns H, Katoen J-P (2010) Performance evaluation and model checking join forces. *Commun ACM* 53(9):76–85. <https://doi.org/10.1145/1810891.1810912>
16. Baier C, Cloth L, Haverkort BR, Hermanns H, Katoen J-P (2010) Performability assessment by model checking of Markov reward models. *Formal Methods Syst Design* 36(1):1–36. <https://doi.org/10.1007/s10703-009-0088-7>
17. Qian L, Liu J (2020) Safe reinforcement learning via probabilistic timed computation tree logic. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp 1–8. <https://doi.org/10.1109/IJCNN48605.2020.9207384>

18. Henzinger TA, Nicollin X, Sifakis J, Yovine S (1994) Symbolic model checking for real-time systems. *Inf Comput* 111(2):193–244. <https://doi.org/10.1006/inco.1994.1045>
19. Chaki S, Gurfinkel A (2018) Bdd-based symbolic model checking. In: Clarke EM, Henzinger TA, Veith H, Bloem R (eds) *Handbook of model checking*. Springer, Cham, pp 219–245. [https://doi.org/10.1007/978-3-319-10575-8\\_8](https://doi.org/10.1007/978-3-319-10575-8_8)
20. Clarke EM, Grumberg O, Mcmillan KL, Zhao X (2003) Efficient generation of counterexamples and witnesses in symbolic model checking. *International Journal on Software Tools for Technology Transfer (STTT)*. [https://doi.org/10.1007/11513988\\_9](https://doi.org/10.1007/11513988_9)
21. Ciesinski F, Baier C, Größer M, Parker D (2008) Generating compact mtbdd-representations from probmela specifications. In: Havelund K, Majumdar R, Palsberg J (eds) *Model checking software*. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pp 60–76. [https://doi.org/10.1007/978-3-540-85114-1\\_7](https://doi.org/10.1007/978-3-540-85114-1_7)
22. Hermanns H, Kwiatkowska M, Norman G, Parker D, Siegle M (2003) On the use of mtbdds for performability analysis and verification of stochastic systems. *J Logic Algebraic Program* 56(1):23–67. [https://doi.org/10.1016/S1567-8326\(02\)00066-8](https://doi.org/10.1016/S1567-8326(02)00066-8)
23. Mikusek P (2009) Multi-terminal bdd synthesis and applications. In: 2009 International Conference on Field Programmable Logic and Applications, pp 721–722. <https://doi.org/10.1109/FPL.2009.5272326>
24. Fujita M, McGeer PC, Yang JC-Y (1997) Multi-terminal binary decision diagrams: an efficient data structure for matrix representation. *Formal Methods Syst Design* 10(2):149–169. <https://doi.org/10.1023/A:1008647823331>
25. Stefanakos I, Calinescu R, Douthwaite J, Aitken J, Law J (2022) Safety controller synthesis for a mobile manufacturing cobot. In: Schlingloff B-H, Chai M (eds) *Software engineering and formal methods*, vol 13550, pp 271–287. Springer, Cham. [https://doi.org/10.1007/978-3-031-17108-6\\_17](https://doi.org/10.1007/978-3-031-17108-6_17)
26. Debbi H (2021) Modeling and performance analysis of resource provisioning in cloud computing using probabilistic model checking. *Informatica*. <https://doi.org/10.31449/inf.v45i4.3308>
27. Guo X (2019) Performance analysis of Israeli–Jalfon’s algorithm using probabilistic model checking. *Concurr Comput: Pract Exp* 31(9):4973. <https://doi.org/10.1002/cpe.4973>
28. Kwiatkowska M, Norman G, Sproston J, Wang F (2007) Symbolic model checking for probabilistic timed automata. *Inf Comput* 205(7):1027–1077. <https://doi.org/10.1016/j.ic.2007.01.004>
29. Fränzle M, Hahn EM, Hermanns H, Wolovick N, Zhang L (2011) Measurability and safety verification for stochastic hybrid systems. In: *Proceedings of the 14th International Conference on Hybrid Systems: Computation and Control - HSCC '11*. ACM Press, Chicago, IL, USA, , p 43. <https://doi.org/10.1145/1967701.1967710>
30. Clarke EM (1997) Model checking. In: *Foundations of Software Technology and Theoretical Computer Science: 17th Conference Kharagpur, India, December 18–20, 1997 Proceedings* 17. Springer, pp 54–56
31. Shetty J, Lawson CP, Shahneh AZ (2015) Simulation for temperature control of a military aircraft cockpit to avoid pilot’s thermal stress. *CEAS Aeronaut J* 6:319–333
32. Lanotte R, Merro M, Tini S (2021) A probabilistic calculus of cyber-physical systems. *Inf Comput* 279:104618. <https://doi.org/10.1016/j.ic.2020.104618>
33. Du D, Huang P, Jiang K, Mallet F (2018) pcssl: A stochastic extension to marte/ccsl for modeling uncertainty in cyber physical systems. *Sci Comput Program* 166:71–88. <https://doi.org/10.1016/j.scico.2018.05.005>
34. Basile D, Di Giandomenico F, Gnesi S (2019) On quantitative assessment of reliability and energy consumption indicators in railway systems. In: Kharchenko V, Kondratenko Y, Kacprzyk J (eds) *Green IT Engineering: Social, Business and Industrial Applications*. Springer, Cham, pp 423–447. [https://doi.org/10.1007/978-3-030-00253-4\\_18](https://doi.org/10.1007/978-3-030-00253-4_18)
35. Parker DA (2003) Implementation of symbolic model checking for probabilistic systems. PhD thesis, University of Birmingham

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

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.

## Authors and Affiliations

Zhen Li<sup>1,2</sup> · Zining Cao<sup>1,2,3,4</sup> · Chao Xing<sup>1</sup>

✉ Zining Cao  
caozn@nuaa.edu.cn

Zhen Li  
lz\_6927@nuaa.edu.cn

Chao Xing  
nuaa\_xc@nuaa.edu.cn

- <sup>1</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, No. 29 Jiangjun Rd, Nanjing 211106, Jiangsu, China
- <sup>2</sup> Science and Technology on Electro-optic Control Laboratory, Luoyang 471000, Henan, China
- <sup>3</sup> Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, Jiangsu, China
- <sup>4</sup> Key Laboratory of Safety-Critical Software, Ministry of Industry and Information Technology, Nanjing 211106, Jiangsu, China