Sensor Tolerance Contracts for Safety Assurance in Cyber-Physical Systems

Jian Xiang

UNC Charlotte, Woodward Hall, Charlotte, NC 28262, USA

Abstract. Modern cyber-physical systems rely on complicated sensing pipelines to produce sensor estimates, which inherently contain uncertainty and may exhibit transient and recurring abnormalities. It is important to ensure system safety under such unavoidable uncertainty. We present a framework for CPS safety assurance under sensor uncertainty based on two key notions. First, we introduce *tolerance contracts* on sensor estimates, which specify how much, how long, and how frequently sensing abnormalities are permitted. Second, we leverage *quantitative safety*, which measures how close a CPS is to violating its safety requirements. Together, these notions enable rigorous reasoning about CPS safety in the presence of sensing abnormalities. We formalize the syntax and semantics of tolerance contracts and develop sound reasoning techniques for contracted CPSs. In particular, we formalize tolerance contracts within differential dynamic logic ($\mathsf{d}\mathcal{L}$) and develop a special invariant-style technique dedicated for reasoning with recurring abnormalities. A water tank case study demonstrates how different contract designs can be used to ensure system safety despite sensor uncertainty.

Keywords: Sensor uncertainty · quantitative safety · formal verification

1 Introduction

Cyber–physical systems (CPSs) control safety-critical infrastructures that directly affect human lives: from automated vehicles and power grids to aircraft and medical robots. These CPSs rely on complex sensing pipelines or techniques, e.g., cameras, LiDAR, IMUs, GPS, and learning-based perception modules, to produce *sensor estimates*, through complex filtering or multi-sensor fusion. Sensor estimates are inherently imperfect, i.e., they are not the true values of the physical entities being measured. It is critical to assure the safety of CPSs in the presence of sensor uncertainties, since control decisions are often made directly from these estimates. Sensor uncertainties manifest as *deviations* between sensor estimates and the physical values they measure, and their safety impact shows several key characteristics. First, deviations may have *large amplitude*, which can directly induce unsafe control actions. For example, a LiDAR returning a phantom obstacle several feet away may trigger unnecessary emergency braking and thus safety concerns. Second, deviations often persist for a non-negligible *duration*, and it is this persistence that enables them to affect a CPS's safety. For

instance, an IMU saturating over multiple control cycles during a sharp turn can mislead the controller into unsafe actions. Third, deviations may be *recurring* and intermittent, causing repeated exposure to unsafe estimates and leading to cumulative safety degradation over time. For example, repeated timing jitter in a camera pipeline caused by illumination may gradually drag the vehicle into unsafety. Together, these characteristics highlight that CPS safety is determined not by whether deviations occur, but *by how large they are, how long they persist, and how frequently they recur.*

To establish formal safety guarantees for CPSs in a systematic way, we introduce the notion of *tolerance contracts*, which express, intuitively, the constraints a CPS has on the sensor estimates. We can thus establish safety guarantees for *contracted CPSs*, i.e., CPSs whose controllers use sensor estimates governed by tolerance contracts. A tolerance contract focuses on defining *sensor abnormalities*, i.e., sensor estimates that are classified as abnormal by the contract. A contract specifies the following:

- normality conditions that classify estimates as normal or abnormal;
- the maximum allowed amplitude of abnormalities once they occur;
- the maximum allowed abnormality duration, i.e., how long continuous abnormalities may persist; and
- the minimum cooldown duration after an abnormality duration, i.e., how long the sensor estimates need to stay normal before the next abnormality duration.

For example, a tolerance contract for a LiDAR sensor on a mobile robot may specify a normality condition under which distance estimates are considered abnormal if they deviate from the normal range by more than 1.2 feet (ft), an amplitude bound limiting such deviations to at most 2 ft, a duration bound of 0.3 seconds (s) on continuous abnormalities, and a cooldown requirement of at least 0.5 s of continuous normal estimates following each abnormality duration.

Deploying tolerance contracts provides a foundation for safety assurance of CPSs subject to sensor uncertainty. For example, consider a mobile robot equipped with an obstacle-avoidance controller. Under perfect sensing, the robot can maintain a minimum safety distance (e.g., 1.6 ft) from any obstacles. When sensor estimates suffer from unrestricted or lightly restricted abnormalities, the robot can violate safety requirements and collide with the obstacles. However, when sensor estimates are governed by a tolerance contract that permits bounded and intermittent abnormalities (e.g., the example above), the contracted robot can temporarily approach obstacles more while maintaining a reduced but positive safety margin (e.g., 1.0 ft). As such, safety assurance of CPSs subject to sensor uncertainty can be reduced to reasoning about safety of contracted CPSs.

For the safety analysis of CPSs subject to transient and recurring sensor abnormalities, the common notion of Boolean safety, i.e., whether the CPS satisfies the safety requirement, becomes overly rigid or uninformative. In practice, a CPS's *degree of safety* may fluctuate significantly before, during, and after an abnormality duration. For example, a brief LiDAR error may cause a mobile robot's distance to a nearby obstacle to drop from 1.4 ft to 0.3 ft, and then gradually recover to 1.4 ft during the cooldown. While the Boolean safety property

"no collision with obstacles" is never violated in this scenario, this fluctuation represents a substantial and meaningful safety degradation. Such effects should not be ignored when reasoning about safety or when supporting other system objectives, such as trajectory tuning or control smoothness.

To address this challenge, we leverage *quantitative safety* (Q-safety), a safety margin that measures how close a CPS is to violating its safety requirements. This margin enables reasoning about how much safety is temporarily consumed during abnormality durations and how it may be subsequently recovered during cooldown. Compared to Boolean safety, Q-safety provides more informative support for safety analysis and control, particularly when deploying tolerance contracts. First, Q-safety enables safety analysis and control decisions in situations where Boolean safety is insufficient. For example, system engineers may wish to trigger safety actions, such as emergency stops, when the safety margin drops by more than 50% within 2.0 seconds; such requirements depend on the amplitude and rate of safety degradation, not only whether safety is violated. Second, the explicit safety margin provided by Q-safety enables systematic comparison and selection among different tolerance contract designs (or controller designs) based on operational context. For instance, engineers may choose an alternative contract that allocates an additional 0.4 ft of safety margin for improved path-planning efficiency, or configure runtime policies to apply stricter contracts when the safety margin degrades during operation.

Building on these insights, we introduce a framework for safety assurance of CPSs under sensor abnormalities by enforcing tolerance contracts on the estimates used by the controller. Instead of focusing on diagnosing or correcting sensing errors, which we view as a complementary problem, we focus on analyzing how much safety impact can be caused by sensor abnormalities. At a high level, tolerance contracts are enforced by a lightweight program *tc-hp*[1] embedded in the controller. At each control cycle, *tc-hp* checks the sensor estimates received by the controller and accepts them only if they satisfy the constraints in the contract. We enforce contracts at control cycles rather than sensor sampling cycles, since system dynamics are primarily affected by the estimates actually used in control decisions in common CPSs.

We illustrate the intuition behind a tolerance contract using a simplified example (the exact syntax is introduced later). Consider a robot controller retrofitted with a contract program *tc-hp* that filters LiDAR estimates before control decisions are made:

$$\textit{tc-hp} \equiv ?\Big(\psi_n \vee \big(\neg\psi_n \wedge \psi_t \wedge \psi_d \wedge \psi_{cd}\big)\Big)$$

Intuitively, the contract accepts an estimate if it is either normal (ψ_n), or abnormal but tolerable, meaning that its deviation is bounded (ψ_t), its continuous duration does not exceed a specified limit (ψ_d), and sufficient cooldown time has elapsed since the previous abnormality (ψ_{cd}). For example, ψ_n may require the estimate to deviate from a reference value (e.g., from analyzing dynamics) by at most 0.2 ft, while ψ_t allows larger deviations (e.g., up to 0.5 ft) for a bounded

[1] we name it *tc-hp* where *tc* for tolerance contracts and *hp* for hybrid programs used in $\mathsf{d}\mathcal{L}$ for modeling CPSs

duration (e.g., 0.3 s), followed by a minimum cooldown (e.g., 3 s) before another abnormality may occur. As such, proving safety for a contracted CPS reduces to reasoning about executions of the CPS with the embedded contract program *tc-hp* and establishing that it maintains a positive Q-safety margin.

However, reasoning about safety for a contracted CPS, i.e., a CPS with an embedded contract *tc-hp*, is particularly challenging when sensor abnormalities *recur intermittently* over time. Executions of the contracted CPS alternate between cooldown durations and abnormality durations that may start at arbitrary times and persist for varying lengths as permitted by the contract constraints. This makes it difficult to identify a conventional control-cycle invariant that holds across all execution phases. We address this challenge by focusing the reasoning on the *abnormality–cooldown cycles*, allowing safety proofs established for one such cycle to be extended to executions with recurring abnormalities.

We work within the formalism of differential dynamic logic ($\mathsf{d}\mathcal{L}$)[61, 62, 9], a language for verifying CPSs. We choose $\mathsf{d}\mathcal{L}$ because it provides a unified representation of controllers and dynamics that enables language-based techniques, and recent advances [79, 80, 16] extend $\mathsf{d}\mathcal{L}$ with notions of Q-safety and relevant reasoning techniques, which can help our analysis of contracted CPSs.

Contribution. This paper presents the design and reasoning techniques for tolerance contracts that promote safety assurance of CPSs subject to sensor uncertainty in the setting of $\mathsf{d}\mathcal{L}$. We make the following contributions:

- *The design of tolerance contracts.* We introduce the syntax and semantics of tolerance contracts, including (1) syntactic constructs for expressing meaningful constraints on tolerable sensor abnormalities and (2) definitions that can precisely capture the semantics of tolerance contracts (Section 3).
- *Reasoning techniques for contracted CPSs.* We develop sound modeling techniques for contract governance in $\mathsf{d}\mathcal{L}$ and propose an invariant-style reasoning approach for establishing safety of contracted CPSs, with particular emphasis on recurring sensor abnormalities (Section 4).
- *A case study.* We present a water tank case study that demonstrates how different contract designs can be applied to ensure system safety. It illustrates both the expressiveness of tolerance contracts and the effectiveness of the proposed reasoning techniques (Section 5).

In addition to the sections mentioned above, Section 2 introduces some key preliminaries. We discuss the related work in Section 6, and conclude in Section 7.

Note that this paper focuses on reasoning about CPS safety given tolerance contracts. The complementary problem of deriving or validating contracts from sensing pipelines is an important direction, involving system identification and runtime monitoring, and is left for future work.

2 Preliminaries

Differential dynamic logic [59, 62, 61] is a dynamic logic [27] for verifying safety properties of CPSs. Program constructs in $\mathsf{d}\mathcal{L}$, called *hybrid programs* [62], can

$$\begin{aligned}
\textbf{\textit{Term}}: &\quad \theta, \eta \ ::= x \mid c \mid \theta \oplus \eta \\
\textbf{\textit{Program}}: &\quad \alpha, \beta \ ::= x := \theta \mid x := * \mid x' = \theta \,\&\, \phi \mid ?\phi \mid \alpha \,;\, \beta \mid \alpha \cup \beta \mid \alpha^* \\
\textbf{\textit{Formula}}: &\quad \phi, \psi \ ::= \top \mid \theta \sim \eta \mid \neg\phi \mid \phi \wedge \psi \mid \forall\, x.\ \phi \mid [\alpha]\phi
\end{aligned}$$

Fig. 1: Syntax and semantics of $\mathsf{d}\mathcal{L}$ programs and formulas

express continuous evolution using differential equations (ODEs), as well as discrete transitions. Figure 1 gives the syntax of hybrid programs. Variables are real-valued and can be deterministically assigned ($x := \theta$, where θ is a real-valued term) or nondeterministically assigned ($x := *$). Program $x' = \theta \,\&\, \phi$ expresses the continuous evolution of variables: given the current value of variable x, the system follows ODE $x' = \theta$ for some (nondeterministically chosen) amount of duration so long as the formula ϕ holds for all of that duration. Hybrid programs also include the operations of Kleene algebra with tests: sequential composition, nondeterministic choice, repetition, and testing a formula.

Figure 1 also gives the syntax of $\mathsf{d}\mathcal{L}$ formulas. In addition to the standard logical connectives of first-order logic, $\mathsf{d}\mathcal{L}$ includes primitive propositions that allow comparisons of real-valued terms (which may include derivatives) and *modality of program necessity* $[\alpha]\phi$, which holds in a state if and only if after any possible execution of hybrid program α, formula ϕ holds. Modality of program necessity can be used to encode the *modality of program existence* $\langle\alpha\rangle\phi$, which holds in a state if and only if after *some* execution of hybrid program α, formula ϕ holds. In particular, we have $\langle\alpha\rangle\phi = \neg[\alpha]\neg\phi$. Common abbreviations for other logical connectives also apply, e.g., $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$.

Fig. 2 shows an example cooling engine working in an environment where temperature increases at a constant rate. Time is measured in seconds (s) and temperature is measured in degrees. Let $temp_p$ denote the physical temperature and $temp_s$ the sensed temperature. The sensing component is modeled by the assignment $temp_s := temp_p$, assuming error free sensing for now. Let t_c be a global clock representing real time, and t_l be the local clock of the controller that is reset at the beginning of each control cycle. Both clocks progress at a constant rate.

$$\begin{aligned}
\phi_{pre} &\equiv temp_p = 100 \wedge t_c = 0 \wedge \epsilon = 1 \\
\phi_{post} &\equiv temp_p \leq 105 \\
ctrl &\equiv temp_s := temp_p \ ;\ t_l := 0; \\
&\qquad (?temp_s > 100 \,;\, delta := -0.5\,) \\
&\qquad \cup\, (?temp_s \leq 100 \,;\, delta := 1) \\
plant &\equiv temp_p{}' = delta, t_l{}' = 1, t_c{}' = 1 \,\&\, (temp_p \geq 0 \wedge t_l \leq \epsilon) \\
\phi_{safety} &\equiv \phi_{pre} \rightarrow [(ctrl \,;\, plant)^*]\phi_{post}
\end{aligned}$$

Fig. 2: $\mathsf{d}\mathcal{L}$ model of a simple cooling engine

The controller $ctrl$ operates as follows. If the sensed temperature exceeds 100, cooling is activated and the temperature decreases at rate $delta = -0.5$. Otherwise, the temperature increases at rate $delta = 1$. The plant is modeled by a system of differential equations of the form $x' = \theta \,\&\, \phi$. The temperature evolves according to $temp_p{}' = delta$, while time progresses via $t_c{}' = 1$ and $t_l{}' = 1$. The evolution domain constraint ϕ restricts evolution to states satisfying $temp_p \geq 0$ and $t_l \leq \epsilon$, where $\epsilon = 1$ bounds the closed loop latency of a scan cycle.

The Boolean safety ϕ_{safety} often concerns if ϕ_{pre} holds then ϕ_{post} holds after any execution of $(ctrl\,;\,plant)^*$, which models the system as repetitions of a controller action followed by an update to the environment. The initial condition ϕ_{pre} has a clause of $t_c = 0$, which sets the initial time of the system to 0.

We later need to refer to variables that occur in a program [62, 61]. The *free variables* of a program α, denoted $\mathrm{FV}(\alpha)$, is the variables that may potentially be read by α. The *bound variables* of α, denoted $\mathrm{BV}(\alpha)$, is the set of variables that may potentially be written to by α. We write $\textsc{Var}(\alpha)$ for the set of all variables of α, i.e., $\mathrm{BV}(\alpha) \cup \mathrm{FV}(\alpha)$. These definitions apply to formulas similarly.

Q-safety Recent works brought the following notion of Q-safety into $\mathsf{d}\mathcal{L}$ [16], which estimates, given a precondition, how safe the CPS's reachable states are. It is defined as the *shortest* distance between the set of reachable states and the set of unsafe states. The larger this distance is, the safer the CPS is.

Definition 1 (Q-safety). *Given a real u, formulas ϕ_{pre}, ϕ_{post}, $\mathcal{H} \equiv \textsc{Var}(\phi_{post})$, a program α is u-safe for ϕ_{pre} and ϕ_{post}, if $u = \mathsf{inf}\{\mathsf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\alpha\rangle]\!]\}$. ($\mathsf{inf}$ is infimum and $[\![\phi]\!]$ denotes the set of states in which ϕ holds)*

Set $[\![\phi_{pre}\langle\alpha\rangle]\!]$ denotes the reachable states of α from an initial state in $[\![\phi_{pre}]\!]$, which corresponds to the *strongest postcondition* in Hoare logic. Its formal definition is $[\![\phi\langle\alpha\rangle]\!] = \{\nu \mid \exists\,\omega \text{ such that } \omega \in [\![\phi]\!] \text{ and } (\omega, \nu) \in [\![\alpha]\!]\}$.

$\mathsf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!])$ is a notion of distance between a state and a set of states, focusing on a set $\mathcal{H}$ of variables. Intuitively, variables in $\mathcal{H}$ are the ones that are relevant to the safety. And thus computing distance over these variables gives us the quantitative distance of interest. We use the *Euclidean metric* defined by $\rho_{\mathcal{H}}(\omega, \nu) = \sqrt{\sum_{x \in \mathcal{H}} (\omega(x) - \nu(x))^2}$. $\mathsf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!])$ builds on the definitions shown in Figure 3, adopted from existing works [21, 11].

$$\mathsf{dist}_{\mathcal{H}}(\omega, \mathcal{S}) = \mathsf{inf}\{\rho_{\mathcal{H}}(\omega, \nu) \mid \nu \in \mathcal{S}\}$$
$$\mathsf{depth}_{\mathcal{H}}(\omega, \mathcal{S}) = \mathsf{inf}\{\rho_{\mathcal{H}}(\omega, \nu) \mid \nu \in (\textsc{Sta} \setminus \mathcal{S})\}$$
$$\mathsf{Dist}_{\mathcal{H}}(\omega, \mathcal{S}) = \begin{cases} \mathsf{depth}_{\mathcal{H}}(\omega, \mathcal{S}), & \text{if } \omega \in \mathcal{S} \\ -\mathsf{dist}_{\mathcal{H}}(\omega, \mathcal{S}), & \text{if } \omega \notin \mathcal{S} \end{cases}$$

Fig. 3: Definitions on distance metrics

- $\mathsf{dist}_{\mathcal{H}}(\omega, \mathcal{S})$ is the *distance* between a state ω and a set of states $\mathcal{S} \subseteq \textsc{Sta}$ ($\textsc{Sta}$ denotes all states). It is the *shortest* distance between ω and all states in $\mathcal{S}$.
- $\mathsf{depth}_{\mathcal{H}}(\omega, \mathcal{S})$ is the *depth* of ω in $\mathcal{S}$ is the *shortest* distance between ω and the *boundary* of $\mathcal{S}$: the set of states at which any small perturbation may leave $\mathcal{S}$.
- $\mathsf{Dist}_{\mathcal{H}}(\omega, \mathcal{S})$ is the *signed distance* between ω and a set of states $\mathcal{S}$. It is positive in the first case and negative in the second case.

We assume two special values: $0^+$and 0^-, where $\mathsf{Dist}_{\mathcal{H}}(\omega, \mathcal{S}) = 0^+$ if $\omega \in \mathcal{S}$ and $\mathsf{depth}_{\mathcal{H}}(\omega, \mathcal{S}) = 0$; and $\mathsf{Dist}_{\mathcal{H}}(\omega, \mathcal{S}) = 0^-$ if $\omega \notin \mathcal{S}$ and $\mathsf{depth}_{\mathcal{H}}(\omega, \mathcal{S}) = 0$.

Consider the cooling system shown in Figure 2. We know $\phi_{pre} \equiv temp = 100$ and $\phi_{post} \equiv temp \leq 105$. During the execution of the system the temperature lies in the real interval $(99.5, 101]$, Therefore, we have Q-safety of 4: the system will always satisfy the postcondition with a safety margin of at least 4 (degrees).

Timed Q-safety Recent works extended the notion of Q-safety in $\mathsf{d}\mathcal{L}$ to a timed setting [80], denoted $\phi\langle\alpha\rangle_{[t_l, t_u]}$, which intuitively represents the *timed strongest postcondition* over a time interval $[t_l, t_u]$.

Definition 2 (Timed quantitative safety). *Given two time points* t_l, t_u*, a real* u*, formulas* ϕ_{pre}*,* ϕ_{post}*, and* $\mathcal{H} \equiv \textsc{Var}(\phi_{post})$*, a program* α *is* u*-safe for* ϕ_{pre} *and* ϕ_{post} *in the time interval* $[t_l, t_u]$*, denoted* $T\text{-}\textsc{safe}_u^{[t_l,t_u]}(\alpha, \phi_{pre}, \phi_{post})$*, for* $u = \inf\{\, \mathsf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\alpha\rangle_{[t_l,t_u]}]\!] \,\}$.

Where $[\![\phi_{pre}\langle\alpha\rangle_{[t_l,t_u]}]\!]$ extends the notion $[\![\phi_{pre}\langle\alpha\rangle]\!]$ with timing as: $\{\nu \mid \exists\, \omega \in [\![\phi_{pre}]\!], (\omega, \nu) \in [\![\alpha]\!]$ and $t_l \leq \nu(t_c) \leq t_u\}$. Reasoning with timed Q-safety often requires $\phi_{pre} \to t_c = 0$, where $t_c = 0$ sets the initial time of the system to 0.

3 Defining Tolerance Contracts

We introduce the formal definitions of the syntax and semantics of tolerance contracts. Intuitively, a tolerance contract is a specification of acceptable abnormality patterns for a set of sensor estimates. It specifies what constitutes an abnormality, and more importantly, how large, how long, and how frequently abnormalities may occur, via the following four components:

- *the normality condition*: it defines what qualifies as abnormality, by specifying the expected normal values of sensor estimates;
- *the constraint on abnormality amplitude*: the maximum allowed (tolerable) amplitude of abnormalities;
- *the constraint on abnormality duration*: the maximum allowed duration of continuous abnormalities; and
- *the constraint on abnormality cooldown*: the minimum cooldown duration after an abnormality duration, which implicitly limits the abnormality frequency.

For example, consider a set of temperature sensor estimates in the form of a time-series, used by the controller after the cooling engine starts operation:

$$\mathcal{S} \equiv \{(95.4, 1), (96.1, 2), (96.6, 3), (97.6, 4), (98.4, 5), (98.5, 6), (99.0, 7)\}$$

Every estimate is a measurement in degrees and its timestamp, and timestamps correspond to discrete control cycles executed once per second. We may be interested in enforcing a contract on this set of estimates: (1) *normality condition*: normal estimates don't deviate from their reference values (e.g., produced from model prediction) for more than 0.5 degrees; (2) *amplitude constraint*: any abnormality doesn't deviate from its reference value for more than 1 degree; (3) *duration constraint*: continuous abnormalities don't last longer than 2 s; and (4) *cooldown constraint*: after every duration of continuous abnormalities, a minimum cooldown of 2 s (continuous normal estimates) follows.

Assume a model-based estimator produces the following reference values:

$$\mathcal{S}_{ref} \equiv (95.5, 1), (96.0, 2), (96.5, 3), (97.0, 4), (97.5, 5), (98.0, 6), (98.5, 7)$$

The set $\mathcal{S}$ satisfies the tolerance contract defined above. In particular, abnormalities are identified pointwise at each timestamp when the deviation between an estimate and its reference exceeds 0.5 degrees. The maximum deviation never exceeds 1 degree, continuous abnormalities last no longer than 2 s, and each abnormality duration is followed by a minimum cooldown of 2 s normal estimates.

This example illustrates how a tolerance contract constrains sensor abnormalities within a single *tolerance cycle*, consisting of one abnormality duration

followed by a cooldown duration. However, real-life sensing abnormalities are rarely one time events and often occur *recurringly*. Figure 4 illustrates recurring tolerance behavior as three consecutive tolerance cycles, each satisfying the same contract constraints: (1) a normality condition (below the dashed line), (2) a bound on abnormality amplitude (below the dotted line), (3) a maximum abnormality duration of three control cycles, and (4) a minimum cooldown of two control cycles after each abnormality duration.

The two examples above highlight the essential aspects that a tolerance contract must capture. We now introduce the formal syntax and semantics of tolerance contracts, which form the foundation for the safety reasoning developed in the remainder of the paper.

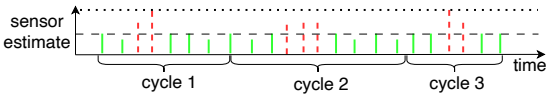

Fig. 4: Visualizing an example recurring tolerance with three cycles. The vertical solid and dashed lines visualize normal and abnormal estimates at cycle cycles. The horizontal dashed and dotted line visualize the conditions that guard the normality and the tolerable abnormality.

Syntax of tolerance contracts. Syntactically, tolerance contract is a specification of acceptable sensor abnormality patterns for a set of sensor estimates, denoted $\text{TC}(\psi_n, \psi_t, \tau, \delta)$, where the components specify constraints on normality (i.e., ψ_n), abnormality amplitude (i.e., ψ_t), abnormality duration (i.e., $\tau > 0$), and cooldown (i.e., $\delta > 0$).

The constraints in a tolerance contract must be clearly specified. We use a simple constraint language shown in Figure 5 for this purpose. It is the quantifier-free and modality-free fragment of $\mathsf{d}\mathcal{L}$ formulas, so the constraints can fit into $\mathsf{d}\mathcal{L}$ models. Variables in constraints are drawn from the $\mathsf{d}\mathcal{L}$ model for which the contract is defined. We later introduce a notion of well-formedness for contracted CPSs to precisely define these restrictions.

$$\textbf{\textit{Term:}}\ \theta, \eta\ ::= x \mid c \mid \theta \oplus \eta$$
$$\textbf{\textit{Formula:}}\ \phi, \psi\ ::= \top \mid \theta \sim \eta \mid \neg\phi \mid \phi \wedge \psi$$

Fig. 5: A language for specifying constraints

To precisely define the semantics of tolerance contracts, we build upon prior results on the robustness of CPS safety [80], which introduce the notion of *timed tolerance* for characterizing robustness against timed adversarial conditions. A CPS satisfies this tolerance notion if its safety evolves through three stages: (1) the CPS is safe, (2) the CPS may become unsafe for a bounded duration and amplitude, and (3) the CPS returns to safety. This notion of tolerance naturally captures timing aspects and is supported by proof rules for establishing timed tolerance of CPSs under attacks.

We instantiate the tolerance notion in the setting of sensor estimates and further extend it to *recurring* behaviors. Intuitively, a set of sensor estimates satisfies a tolerance contract if the estimates satisfy the *tolerance cycle for sensor estimates* (or simply *tolerance cycle*) defined below repeatedly over time:

Definition 3 (Tolerance cycle for sensor estimates). *Given a set of sensor estimates $\mathcal{S}$ with timestamps, and two time points $\mathtt{T}_s$ and $\mathtt{T}_e$ with $\mathtt{T}_s < \mathtt{T}_e$, the set $\mathcal{S}$ with normality condition ψ_n is said to tolerate abnormalities bounded by*

(ψ_t, τ, δ) in the interval $[\mathtt{T}_s, \mathtt{T}_e]$, denoted as $\mathsf{c\text{-}tol}_{[\mathtt{T}_s,\mathtt{T}_e]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$, if there exist time points $\mathtt{T}_{as}$, $\mathtt{T}_{ae}$, with $\mathtt{T}_s < \mathtt{T}_{as} < \mathtt{T}_{ae} < \mathtt{T}_e$, such that the following holds:
- *(normality) all estimates in interval $[\mathtt{T}_s, \mathtt{T}_{as}]$ are normal, i.e., they satisfy ψ_n;*
- *(abnormal duration) all estimates in interval $[\mathtt{T}_{as}, \mathtt{T}_{ae}]$ may be abnormal but tolerable under ψ_t, i.e., they may violate ψ_n but satisfy ψ_t, and $\mathtt{T}_{ae} - \mathtt{T}_{as} \leq \tau$;*
- *(cooldown duration) all estimates in interval $[\mathtt{T}_{ae}, \mathtt{T}_e]$ are normal, i.e., they satisfy ψ_n, and this interval is long enough: $(\mathtt{T}_e - \mathtt{T}_{ae}) \geq \delta$.*

Here, within the time interval $[\mathtt{T}_s, \mathtt{T}_e]$, the tolerance cycle permits a sub-interval $[\mathtt{T}_{as}, \mathtt{T}_{ae}]$ in which abnormalities are allowed but bounded. The duration of this sub-interval is bounded by τ, and the amplitude of abnormalities is constrained by ψ_t, which is often logically weaker than the normality condition ψ_n. The parameter δ specifies the required length of the cooldown duration following an abnormality duration, which also implicitly limits the frequency of abnormalities by enforcing a minimum separation between consecutive abnormality durations.

We can extend the definition of tolerance cycles recursively to capture the meaning of *recurring tolerance cycles*, as follows:

Definition 4 (Recurring tolerance cycles for sensor estimates). *Given a set of sensor estimates $\mathcal{S}$ and an interval $[\mathtt{T}_s, \mathtt{T}_e]$, the set $\mathcal{S}$ with normality condition ψ_n is said to recurringly tolerate abnormalities bounded by (ψ_t, τ, δ) over $[\mathtt{T}_s, \mathtt{T}_e]$, denoted as $\mathsf{rec\text{-}tol}_{[\mathtt{T}_s,\mathtt{T}_e]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$, if one of the following holds:*
- *(base case) $\mathsf{c\text{-}tol}_{[\mathtt{T}_s,\mathtt{T}_e]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$;*
- *(recursive case) there exists a time point $\mathtt{T}_m$ with $\mathtt{T}_s < \mathtt{T}_m < \mathtt{T}_e$ such that $\mathsf{c\text{-}tol}_{[\mathtt{T}_s,\mathtt{T}_m]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$ and $\mathsf{rec\text{-}tol}_{[\mathtt{T}_m,\mathtt{T}_e]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$.*

Now we can precisely define the semantics of tolerance contracts:

Semantics (Satisfaction) of tolerance contracts. A set of sensor estimates $\mathcal{S}$ satisfies a tolerance contract $\mathrm{TC}(\psi_n, \psi_t, \tau, \delta)$ over the interval $[\mathtt{T}_s, \mathtt{T}_e]$, denoted as $\mathcal{S} \models_{[\mathtt{T}_s,\mathtt{T}_e]} \mathrm{TC}(\psi_n, \psi_t, \tau, \delta)$, iff $\mathsf{rec\text{-}tol}_{[\mathtt{T}_s,\mathtt{T}_e]}(\mathcal{S}, \psi_n, \psi_t, \tau, \delta)$ holds.

4 Reasoning with the Safety of Contracted CPSs

This section presents how to reason about the safety of contracted CPSs. We first introduce how contract governance is modeled in $\mathsf{d}\mathcal{L}$, and then present an invariant-style reasoning approach that establishes safety under recurring tolerance cycles by reasoning about a single tolerance cycle.

We focus on contracted CPSs with a single tolerance contract to simplify the presentation and clarify the core proof ideas. Note that a single contract may apply to sensor estimates of multiple physical entities. For example, the normality condition ψ_n can express correlations between velocity and distance. Moreover, tolerance contracts that apply to independent physical entities can be naturally combined. We defer the analysis of potential interference between multiple contracts and their safety implications to future work.

4.1 Modeling Contract Governance in dℒ

Modeling contracted CPSs requires techniques to (1) capture the effect of constraints specified in tolerance contracts, and (2) model the timing structure of abnormalities, including their duration and recurrence. These two requirements are addressed together by inserting a special program, denoted $tc\text{-}hp(\psi_n, \psi_t, \tau, \delta)$, into the dℒ model of a CPS immediately after the sensing component. This program encodes the contract constraints as dℒ tests that restrict the sensor estimates provided by the sensing component, and introduces auxiliary variables to track the duration of continuous abnormalities and cooldown durations. Concretely, contract governance is modeled by transforming a controller $ctrl \equiv sensing\,;\, ctrl\_logic$ into $sensing\,;\, tc\text{-}hp(\psi_n, \psi_t, \tau, \delta)\,;\, ctrl\_logic$.

$$
\begin{array}{lll}
tc\text{-}hp(\psi_n, \psi_t, \tau, \delta) \equiv & & \\
\quad ?\psi_n\,; & & \textit{normal estimate} \\
\qquad ?cd & & \textit{cooldown continues} \\
\quad \cup\, ?\neg cd\,;\, cd := true & & \textit{cooldown starts} \\
\qquad t_{cd} := t_c & & \textit{record start time} \\
\cup\, ?(\neg\psi_n \wedge \psi_t)\,; & & \textit{tolerable abnormality} \\
\qquad ?cd\,;\, cd := false & & \textit{abnormalities start} \\
\qquad\quad ?(t_c - t_{cd} \geq \delta)\,; & & \textit{cooldown is enough} \\
\qquad\qquad t_{ab} := t_c & & \textit{record start time} \\
\quad \cup\, ?\neg cd\,; & & \textit{abnormalities continue} \\
\qquad\quad ?(t_c - t_{ab} \leq \tau) & & \textit{duration constraint} \\
\_tc := true\,;\, \_tc := false & & \textit{help track estimates}
\end{array}
$$

Fig. 6: Modeling contract governance in dℒ

Figure 6 illustrates the structure of the program $tc\text{-}hp(\psi_n, \psi_t, \tau, \delta)$ that encodes contract governance. The contract permits two major cases at each control cycle: (1) a *normal case*, where the current sensor estimate is considered normal and satisfies ψ_n; and (2) an *abnormal case*, where the estimate is a tolerable abnormality, i.e., $\neg\psi_n \wedge \psi_t$ holds. Each case is further divided based on whether the system is currently within a cooldown duration, indicated by a Boolean variable cd[2]. In the normal case, if the CPS was not previously in a cooldown duration ($?\neg cd$ holds), then a new cooldown duration begins at the current control cycle. The current global time is recorded in the auxiliary variable t_{cd} via $t_{cd} := t_c$. If the CPS was already in a cooldown duration, the cooldown continues without resetting t_{cd}. In the abnormal case, if an abnormal estimate occurs while the CPS is in a cooldown duration, the cooldown ends at the current cycle. In this situation, the cooldown duration must satisfy the contract constraint, namely $t_c - t_{cd} \geq \delta$. The start of the abnormality duration is then recorded by assigning $t_{ab} := t_c$. If an abnormal estimate occurs during an ongoing abnormality duration, the contract requires the duration constraint holds, i.e., $t_c - t_{ab} \leq \tau$. Finally, the program updates a fresh auxiliary variable $\_tc$, which is used to precisely track the estimates accepted by the contract program and thus the controller. We discuss it more later.

The model in Figure 6 captures recurring tolerance cycles by tracking and updating the start times of abnormality and cooldown durations, together with

[2] Boolean variables, e.g., cd, are not built-in to dℒ but can be encoded.

a Boolean flag *cd* that records the current phase. In particular, abnormality and cooldown durations alternate over time. If *cd* holds, indicating that the CPS is in a cooldown duration, and a tolerable abnormality occurs, then a new abnormality duration begins. A symmetric transition occurs when *cd* is false.

Contracted CPSs (CPSs with contract-governed estimates) are defined as:

Definition 5 (Contracted CPSs). *The contracted CPS for a CPS* $\alpha^* \equiv (\mathit{sensing}\,;\,\mathit{ctrl\_logic}\,;\,\mathit{plant})^*$ *and a contract* $\textsc{tc}(\psi_n, \psi_t, \tau, \delta)$ *is defined as:*
$\textsc{c-cps}(\alpha^*, \textsc{tc}(\psi_n, \psi_t, \tau, \delta)) \equiv (\mathit{sensing}\,;\,\mathit{tc\text{-}hp}(\psi_n, \psi_t, \tau, \delta)\,;\,\mathit{ctrl\_logic}\,;\,\mathit{plant})^*$
where $BV(\mathit{tc\text{-}hp}(\psi_n, \psi_t, \tau, \delta))$ *are fresh, including* cd, t_{ab}, t_{cd}, *and* $\mathit{\_tc}$.

Here, the bound variables of *tc-hp* are required to be fresh so that they do not interfere with the variables of the original CPS model.

We only consider contracted CPSs that are *well-formed*, which means, intuitively, that the contracts refer only to variables that are (1) accessible by the original CPS, and (2) not solely modifiable by the plant. The first requirement is natural, as a contract must be designed for a given CPS. The second requirement prevents contracts from relying on physical variables that are exclusively controlled by the plant and cannot be influenced or reliably accessed by the controller. This is a reasonable requirement, as tolerance contracts are designed to constrain the sensor estimates consumed by the controller, rather than the plant dynamics. Moreover, from a practical point of view, it is often impossible for a sensor contract to access the exact values of physical plant variables, such as temp_p. The well-formedness condition is formally defined as follows:

Well-formed contracted CPSs. For a contracted CPS (Definition 5), i.e., $\alpha_c^* \equiv \textsc{c-cps}(\alpha^*, \textsc{tc}(\psi_n, \psi_t, \tau, \delta))$, it is *well-formed* if the following holds:
$\big(\textsc{Var}(\psi_n) \cup \textsc{Var}(\psi_t)\big) \subseteq \big(\textsc{Var}(\alpha) \setminus \mathrm{BV}(\mathit{plant}) \cup \mathrm{BV}(\mathit{sensing}\,;\,\mathit{ctrl\_logic})\big)$

For example, in the cooling engine, well-formedness requires that the formulas ψ_n and ψ_t refer only to sensing and controller variables, and do not refer to variables solely modifiable by the plant, such as $\{\mathit{temp}_p, t_c\}$.

This modeling scheme of contract governance shown in Figure 6 is *sound*. Intuitively, soundness means that, starting from valid initial states of the original CPS, *all reachable states of a contracted CPS* α_c^* *that correspond to accepted sensor estimates should satisfy the encoded contract*. Note that not all reachable states of α_c^* contain accepted estimates, since the program *ctrl_logic* may modify them. We use an auxiliary variable $\mathit{\_tc}$ for locating these states. Recall the special assignments at the end of $\mathit{tc\text{-}hp}(\psi_n, \psi_t, \tau, \delta)$ in Figure 6. The assignments ensure that if $\mathit{\_tc}$ holds in a reachable state ω of α_c^*, then ω is guaranteed to (1) contain estimates produced by the sensing component *sensing*, and (2) satisfy the constraints of the contract encoded in $\mathit{tc\text{-}hp}(\psi_n, \psi_t, \tau, \delta)$.

With these insights, we define an operation $V\Downarrow_s \equiv \{\omega \in V \mid \omega(\mathit{\_tc}) = \mathit{true}\}$ that can extract the set of states with accepted sensor estimates from a set V. We can now state the soundness theorem: the set of reachable states of α_c^* that contain accepted estimates should satisfy the contracted being encoded:

Theorem 1 (Soundness of the modeling of contract governance). *Let* $\alpha_c^* \equiv$ *C-CPS*$(\alpha^*, TC(\psi_n, \psi_t, \tau, \delta))$ *be a well-formed contracted CPS. Then for any formula* ϕ_{pre} *that* $\text{VAR}(\phi_{pre}) \subseteq \text{VAR}(\alpha)$*, the following holds:*

$$((t_c = 0 \wedge t_{ab} = 0 \wedge t_{cd} = -\delta \wedge \phi_{pre})\langle \alpha_c^* \rangle)\Downarrow_s \models_{[0,\infty)} TC(\psi_n, \psi_t, \tau, \delta)$$

Theorem 1 has several subtleties. The valid initial states are captured by ϕ_{pre}, which is required not to mention variables outside $\text{VAR}(\alpha)$. The condition $t_c = 0$ sets the starting time to 0. The initial values of t_{ab} and t_{cd} are chosen carefully to ensure that all branches of $tc\text{-}hp(\psi_n, \psi_t, \tau, \delta)$ are reachable even at $t_c = 0$. Specifically, when a normal estimate is observed, both normal sub-cases are trivially reachable. When a tolerable abnormality is observed, the sub-case corresponding to the start of an abnormality duration is reachable even at $t_c = 0$ due to the initialization $t_{cd} = -\delta$. Similarly, the sub-case corresponding to a continuing abnormality is reachable even from the beginning due to $t_{ab} = 0$.

Theorem 1 can be proven by induction on the number of iterations of α_c^*.

4.2 Proving the Safety of Contracted CPSs

The main objective then is to determine whether a contracted CPS is safe. In particular, we aim to establish whether

$$\text{T-SAFE}_u^{[t_l, t_u]}(\alpha_c^*, (t_c = 0 \wedge t_{ab} = 0 \wedge t_{cd} = -\delta \wedge \phi_{pre}), \phi_{post})$$

holds for $\alpha_c^* \equiv \text{C-CPS}(\alpha^*, \text{TC}(\psi_n, \psi_t, \tau, \delta))$ and some $u \geq 0^+$ over a time interval $[t_l, t_u]$. We are particularly interested in the interval $[0, \infty)$, which corresponds to reasoning about all states reachable by the contracted CPS. The key challenge lies in reasoning about recurring behaviors, which are implicitly modeled in α_c^*. In particular, the values of timing-related variables, such as t_{cd} and t_{ab}, may vary across different abnormality durations. This variability makes it difficult to apply standard control-cycle based invariant reasoning, i.e., searching for a loop invariant ϕ_{inv} for α_c^* that holds at the end of every control cycle. Identifying such an invariant that is preserved across all control cycles is challenging, if not impossible, due to the unbounded nature of these timing variables. For example, an abnormality duration may span multiple control cycles, while the subsequent cooldown duration may last for an arbitrarily long time.

To address this challenge, we propose an invariant-based reasoning approach that operates at the granularity of *a single tolerance cycle*, rather than across all control cycles. Intuitively, for a time interval $[t_l, t_u]$ in which α_c^* exhibits recurring tolerance behavior (i.e., multiple tolerance cycles), our approach reduces the reasoning of safety over the entire interval $[t_l, t_u]$ to reasoning about safety over a single tolerance cycle, typically the first one in the interval. Invariants defined over tolerance cycles are easier to identify, as they align naturally with the semantics of tolerance cycles and abstract away unbounded history across control cycles. The following theorem formalizes this reasoning approach based on tolerance cycles. A subtlety here is that we must account for the maximum closed-loop latency ϵ; in particular, we rely on the fact that the controller is triggered at least once every ϵ time units.

Theorem 2 (Safety of contracted CPS via tolerance-cycle invariant). *Let α_c^* be a well-formed contracted CPS $\textit{C-CPS}(\alpha^*, \textit{TC}(\psi_n, \psi_t, \tau, \delta))$ with maximum closed-loop latency ϵ. If for formulas ϕ_{init}, ϕ_{pre}, ϕ_{inv}, and ϕ_{post}, a time point t_m, programs α_{ct} and α_n such that the following items hold:*

1. $\phi_{init} \equiv t_c = 0 \wedge t_{ab} = 0 \wedge t_{cd} = -\delta$;
2. $\alpha_n \equiv (\,\alpha_c\,;\, ?(cd = true)\,)$ *and* $\alpha_{ct} \equiv (\,\alpha_c\,;\, ?(t_{ab} = 0)\,)$;
3. $\phi_{pre} \to \phi_{inv}$, $\phi_{inv} \to [\alpha_n]\phi_{inv}$, *and* $\phi_{inv} \to \phi_{post}$;
4. $T\text{-}\textit{SAFE}_{u}^{[0,t_m]}(\alpha_{ct}^*, (\phi_{init} \wedge cd = false \wedge \phi_{inv}), \phi_{post})$ *for some* $u \geq 0^+$;
5. $T\text{-}\textit{SAFE}_{u_1}^{[t_m,\delta]}(\alpha_{ct}^*, (\phi_{init} \wedge cd = false \wedge \phi_{inv}), \phi_{inv})$ *for some* $u_1 \geq 0^+$;
6. $\tau \leq t_m \leq \delta - \epsilon$.

then we have $T\text{-}\textit{SAFE}_{u_2}^{[0,\infty)}(\alpha_c^*, \phi_{init} \wedge \phi_{pre}, \phi_{post})$ *for some* $u_2 \geq 0^+$.

Intuitively, Theorem 2 reduces reasoning about unbounded executions of a contracted CPS to reasoning about a *single, canonical tolerance cycle* consisting of one abnormality duration followed by a cooldown duration. The conditions are designed so that safety established for this cycle can be soundly extended to executions with recurring tolerance cycles. We elaborate on these conditions.

- **Initialization**. Item 1 initializes timing variables so they align with the final proof obligation and the obligation of a single canonical tolerance cycle.
- **Special programs**. Item 2 introduces two restricted programs used for reasoning. The program α_n restricts executions to remain in cooldown by enforcing the guard $?(cd = true)$ at the end of each iteration. As a result, every iteration of α_n admits only normal estimates, making it suitable for invariant-based reasoning about normal behavior (Item 3). The program α_{ct} restricts executions by enforcing the guard $?(t_{ab} = 0)$, which ensures that no new abnormality duration may begin once an abnormality has started (see Figure 6). When combined with the initial condition $cd = false$ assumed in Items 4 and 5, executions of α_{ct} are forced to follow a canonical tolerance cycle: an abnormality duration starting at time 0, followed by a cooldown duration.
- **Safety of normal estimates.** Item 3 establishes that normal estimates are safe to be consumed by the contracted CPS, by identifying an invariant ϕ_{inv} that is preserved by α_n and that implies the safety postcondition ϕ_{post}. This invariant also helps proving safety for executions that start with a cooldown duration before the abnormality duration starts. Note here ϕ_{inv} is a control-cycle invariant, which is common for executions without abnormalities.
- **Safety during abnormalities.** Item 4 ensures that the contracted CPS does not violate the safety postcondition ϕ_{post} during intervals in which abnormalities may occur. During this phase, the Q-safety margin with respect to ϕ_{inv} may become negative, even though all reachable states still satisfy ϕ_{post}, since ϕ_{inv} is typically stronger than the safety requirement itself. This highlights a key challenge: a single control-cycle invariant such as ϕ_{inv} generally cannot hold uniformly across both abnormality durations and cooldown durations.
- **Recovery to invariant region.** Item 5 guarantees that by time t_m, the contracted CPS has returned to the invariant region ϕ_{inv} after the abnormality duration and a cooldown duration.

– **Timing constraints.** Item 6 constrains the choice of t_m so that it occurs after any possible maximum abnormality duration and sufficiently before the end of a minimum cooldown duration, accounting for the maximum closed-loop latency ϵ. The constraints ensure the existence of a connecting state in which ϕ_{inv} holds and from which invariant preservation under α_n applies.

These conditions collectively allow the safety proofs for a single tolerance cycle to be re-applied whenever a new abnormality duration begins, establishing safety over unbounded executions. The proofs for a single tolerance cycle build on proving timed Q-safety (i.e., Item 4 and 5), which is not the focus of this work, but can benefit from recent results [80] and future advances.

A proof of Theorem 2 relies on the following key lemma, which connects executions of α_c^* considered in Theorem 2 with executions corresponding to abnormality durations that begin at arbitrary time points. Intuitively, the lemma states that for any execution fragment $(\omega, \nu) \in [\![\alpha_c^*]\!]$ where ω is the start state of an abnormality duration, there exists another execution fragment $(\omega', \nu') \in [\![\alpha_c^*]\!]$ that is aligned with the canonical start state assumed in Theorem 2 (Items 4 and 5), while agreeing with the original execution on all non-timing variables. In particular, the lemma shows that any abnormality-start state occurring later in time can be "shifted" to time 0 by adjusting only timing-related variables, without affecting the evolution of the remaining state variables.

Lemma 1 (Time-shift invariance of abnormality-start executions). *Consider a well-formed contracted CPS α_c^*. For any execution fragment $(\omega, \nu) \in [\![\alpha_c^*]\!]$ such that ω is the start state of an abnormality duration, i.e., $\omega(cd) = false$, $\omega(t_{ab}) = \omega(t_c)$, and $\omega(\_tc) = true$, there exist states ω' and ν' satisfying:*
– *$(\omega', \nu') \in [\![\alpha_c^*]\!]$;*
– *for all variables $x \notin \{t_c, t_{ab}, t_{cd}\}$, $\omega'(x) = \omega(x)$ and $\nu'(x) = \nu(x)$;*
– *ω' is a canonical abnormality-start state of the form required by Theorem 2 (Item 4 and 5), i.e., $\omega'(t_c) = 0$, $\omega'(t_{ab}) = 0$, $\omega'(t_{cd}) = -\delta$, and $\omega'(cd) = false$.*

This lemma holds, intuitively, because the two transition pairs differ only in timing variables, which do not affect controller or plant behavior after contract acceptance, making the executions behaviorally equivalent for safety reasoning.

Proof sketch of Theorem 2. We focus on proving a stronger claim: for any execution $(\omega, \nu) \in [\![\alpha_c^*]\!]$ with $\omega \models \phi_{init} \wedge \phi_{pre}$, all reachable states have at least $min(u, u_1)$-Q-safety and every abnormality duration starts from a state satisfying ϕ_{inv}. The proof proceeds by induction on the number k of abnormality durations.

Base case (k=1). Let ω_m be the start state of this abnormality duration. By Item 3, $\omega_m \models \phi_{inv}$. Fix the time point t_m satisfying Items 4–6. For any intermediate state ω_b, if it occurs before ω_m, then $\omega_b \models \phi_{inv}$ by invariant preservation. If it occurs during the interval $[\omega_m(t_c), \omega_m(t_c) + t_m]$, Lemma 1 yields a time-shifted execution to which Item 4 applies, implying u-Q-safety. If it occurs after t_m but before $\omega_m(t_c) + \delta$, Item 5 applies, yielding u_1-Q-safety. For later states, the bound on closed-loop latency ϵ guarantees the existence of a connecting state within the cooldown window where ϕ_{inv} holds, from which invariant preservation (Item 3) implies safety. Thus all states satisfy non-negative Q-safety.

Inductive step. Assume the claim holds for executions with k abnormality durations. Consider an execution with k+1 abnormality durations and let ω_m be the start of the (k+1)th one. By the induction hypothesis, the start of the kth abnormality satisfies ϕ_{inv}. Since successive abnormality starts are separated by at least δ and $t_m \leq \delta$ (Item 6), Item 5 implies $\omega_m \models \phi_{inv}$. The reasoning for this abnormality duration then follows exactly as in the base case. By induction, safety holds for all executions of α_c^*.

Note that Theorem 2 does not explicitly characterize the quantitative relationship between u, u_1, and u_2. In general, since $\phi_{inv} \rightarrow \phi_{post}$, it follows that $u_2 \geq min(u, u_1)$. However, this bound is often uninformative in practice, as the invariant ϕ_{inv} may be significantly stronger than the safety postcondition ϕ_{post}, causing u_1 to be overly conservative and not reflective of the actual Q-safety of reachable states over the interval $[t_m, \delta]$. Moreover, it frequently holds that $u_2 \geq u$, since the Q-safety of α_c^* typically improves after the abnormality phase, and the worst-case safety impact occurs during $[0, t_m]$ when abnormalities are present. A precise quantitative characterization of these relationships, as well as other trade-offs induced by tolerance contracts, such as how abnormality and cooldown durations affect overall Q-safety is left for future work.

5 A Case Study: A Water Tank

We present a water tank case study that demonstrates how tolerance contracts (Section 3) and the reasoning techniques (Section 4) can be applied.

$$(\textit{System Constants} : \epsilon = 1 \land x_L = 100 \land r = 2)$$
$$\phi_{pre} \equiv x_p = 36 \land t_c = 0$$
$$\phi_{post} \equiv x_p \leq 40$$
$$\textit{inc} \equiv ?x_s < 35\,;\, sc := 1$$
$$\textit{dec} \equiv ?x_s \geq 35\,;\, sc := 0$$
$$\textit{ctrl} \equiv t_l := 0\,;\, \textit{sensing}\,;\, (\textit{inc} \cup \textit{dec})$$
$$\textit{plant} \equiv (x_p{}' = r(sc - \frac{x_p}{x_L}), t_c{}' = 1, t_l{}' = 1)\,\&(x_p \geq 0 \land t_l \leq \epsilon)$$

Fig. 7: dℒ model of a water tank

Consider a water tank, inspired by literature [10], whose model is shown in Fig. 7. It mixes the salt and water inside the tank. Initially, it contains 36 lb salt ($x_p = 36$) dissolved in 100 gal of water ($x_L = 100$). An inflow of water with a salt concentration rate sc (lb of salt/gal) is entering the tank at a rate of $r = 2$ gal/min. The well-stirred mixture is draining from the tank at the same rate r. The tank has two modes of control: a salt decreasing mode with sc set to 0 if the measured salt level is high ($?x_s \geq 35$) and a salt increasing mode with sc set to 1 if the measured salt level is low ($?x_s < 35$). The rate of change of salt in the tank ${x_p}'$ is equal to the rate at which salt is flowing in minus the rate at which is flowing out: ${x_p}' = r(sc - x_p/x_L)$, where x_p/x_L computes the concentration of the salt. The safety condition ϕ_{post} of interest is enforcing an upper bound (i.e., $x_p \leq 40$).

To demonstrate the use of tolerance contracts, we assume a worst-case sensing scenario in which $\textit{sensing} \equiv x_s := *$, that is, the sensor estimates may take

arbitrary values at each control cycle. We present two tolerance contracts with different designs. While the first contract is deliberately conservative, the second admits a richer design space that can be tuned to balance safety, precision, and potentially other system objectives.

Simple safety contract. We consider a tolerance contract $\text{TC}(\psi_n, \psi_t, \tau, \delta)$ defined by $\psi_n \equiv x_s \geq 35$, $\psi_t \equiv \mathit{true}$, $\tau \equiv 1$, and $\delta \equiv 5.7$. Under normal estimates, the controller always chooses the decreasing mode. The contract permits up to one minute of arbitrary abnormal estimates, followed by a mandatory cooldown of at least 5.7 minutes.

We can apply Theorem 2 with $\phi_{inv} \equiv x_p \leq 36$ and $t_m = 4.7$, We analyze the details intuitively. In the worst case, the controller remains in the increasing mode throughout the abnormality duration and one additional control cycle due to closed-loop latency (1 minute). Starting from $x_p = 36$, this yields a sound upper bound $x_p \leq 38.6$, corresponding to a Q-safety of $40 - 38.6 = 1.4$. During cooldown, the system operates exclusively in the decreasing mode. The dynamics reduce the salt level below 36, re-establishing the invariant $x_p \leq 36$ at $t_m = 4.7$, before any subsequent abnormality may occur. Consequently, the safety condition $x_p \leq 40$ always holds for the contracted CPS.

Practical safety contract. The safety contract above is simple and ensures safety. However, it is overly restrictive, as it rejects all estimates below 35; we thus introduce a more flexible safety contract that requires sensor estimates to conservatively over-approximate the true state. In particular, we assume the sensing component prioritizes safety by producing estimates that are conservative upper bounds on the current physical values (i.e., $x_s \geq x_p$ holds when control decisions are made). This assumption allows us to impose a tolerance contract that suffices to establish safety guarantees, where $\tau \equiv 1$, $\delta \equiv 17$ and

$$\psi_n \equiv \big(x_o < 35 \wedge (x_o + 1.4 * t_l \leq x_s \leq 37)\big) \vee \big(x_o \geq 35 \wedge (x_o - 0.6 * t_l \leq x_s \leq 37)\big)$$

$$\psi_t \equiv \big(x_o < 35 \wedge (x_o + 0.7 * t_l \leq x_s)\big) \vee \big(x_o \geq 35 \wedge x_s \leq x_o\big)$$

The auxiliary variable x_o records the estimate used in the previous control cycle, which is modeled by adding the assignment $x_o := x_s$ at the beginning of the sensing program *sensing*, and by including $x_o = 36$ in ϕ_{pre}. This contract constrains successive estimates so that each new estimate x_s conservatively over-approximates the set of states reachable from the previous estimate x_o under the system dynamics. For example, when $x_o < 35$, the system is in the increasing mode and the rate of increase of salt is bounded by 1.4, which justifies the constraint $x_o + 1.4 * t_l \leq x_s$. We additionally impose a hard upper bound $x_s \leq 37$, consistent with the safety objective. The abnormality constraint ψ_t relaxes the over-approximation requirement while preserving monotonicity. As a result, abnormal estimates may lead to incorrect control decisions, for example, choosing the increasing mode when the physical salt level should decrease.

Again, we can prove the safety of the contracted water tank by applying Theorem 2 with $t_m = 16$ and the following tolerance-cycle invariant ϕ_{inv}:

$$\big((x_s \geq 35 \wedge x_p \leq x_s - 0.6 * t_l) \vee (x_s < 35 \wedge x_p \leq x_s + 1.4 * t_l)\big) \wedge 34 \leq x_s \leq 37$$

We explain the safety argument by tracking the key quantity $x_p - x_s$, which captures how far the physical state may deviate from the sensor estimate. In the

worst case, $x_p - x_s$ increases throughout the abnormality duration and one additional control cycle due to closed-loop latency (1 minute). Over this period, the increase is bounded by 0.7 per control cycle, as derived from the plant dynamics, without violating the amplitude constraint. Starting from $x_p - x_s = 1.4$ (as required by ϕ_{inv}), the deviation is therefore upper bounded by 2.8, which yields a Q-safety lower bound of $40 - 37 - 2.8 = 0.2$ (Item 4). Once the cooldown begins, normal estimates are enforced. Regardless of whether x_s is above or below the threshold 35, the contract's normality condition ensures that $x_p - x_s$ decreases monotonically at a minimum rate of 0.1 per minute. Consequently, the deviation returns to the invariant region by $t_m = 16$, accounting for 1 minute of abnormality duration, 1 minute of closed-loop latency, and 14 minutes of recovery. At this point, the invariant ϕ_{inv} holds again and is preserved thereafter.

The design of tolerance contracts is inherently flexible and supports meaningful trade-offs between different system objectives. For example, the constant 0.6 in the normality condition ψ_n can be increased (e.g., to 0.65) to admit more sensor estimates. While such a design is less conservative and arguably more precise, it also requires a longer cooldown duration to ensure recovery to the invariant region. We can also adjust the contract parameters, e.g., ψ_t and τ, to change estimates tolerance and affect the resulting Q-safety. For example, modifying ψ_t to $\big(x_o < 35 \wedge x_o + 1.0 * t_l \leq x_s\big) \vee \big(x_o \geq 35 \wedge x_s \leq x_o - 0.5 * t_l\big)$ would reject more estimates and increase the resulting Q-safety. Understanding the quantitative relations between parameters is an interesting future work.

Our experience with this case study also supports the motivation of Theorem 2. In particular, for the second contract, it is difficult to identify a control-cycle invariant for the contracted water tank that holds both during the first 2 minutes and afterwards. Instead, it is easier to directly analyze the accumulated adverse safety impact, given an invariant ϕ_{inv} that holds initially. Theorem 2 provides a viable solution to this challenge.

6 Related Work

Timed tolerance of safety Our work is inspired by a recent work by Xiang et al. on the robustness of CPSs. It introduces a notion called *timed tolerance of safety*, which intuitively defines how much a CPS's Q-safety can tolerate a bounded duration of unsafety in adversarial settings [80]. We adjust this notion to sensor estimates and extend it to recurring settings, so it can naturally capture sensor uncertainty. This prior work also has developed reasoning techniques for timed safety, which can be used to prove the Q-safety for one tolerance cycle, i.e., Item 4 and 5 in Theorem 2. Compared with this work, we focus on recurring tolerance cycles, a main piece missing from prior works.

State estimation with sensor uncertainties State estimation is the problem of reconstructing the state of a system, given a sequence of measurements and a prior model of the system [7], and has been extensively studied by the control community. State estimation methods such as the Kalman Filter are widely used to fuse noisy measurements and system models to produce refined

estimates of both sensor quantities. In these works, "tolerance" typically refers to an estimator's ability to absorb or smooth noisy measurements while keeping the estimation error bounded. In contrast, our notion of tolerance concerns how much sensor abnormality the CPS safety can tolerate. Our work is therefore orthogonal to state estimation: tolerance contracts specify explicit, enforceable constraints on abnormalities in the estimated sensor values, often the outputs of these estimators, rather than attempting to improve their accuracy.

Sensor uncertainty and CPS safety Meanwhile, a growing body of work has investigated how uncertainties arising in sensor measurements or environment affect the performance and safety of learning-based CPS applications [76, 25, 38, 26, 64, 34, 71, 81, 43]. In autonomous driving systems, the technique of uncertainty quantification has been applied to perception and planning tasks, such as object detection [13, 35, 42], lane detection [19], motion prediction [72, 44], trajectory forecasting [30], and uncertainty-aware steering control [37]. In this line of works, uncertainties are often treated as part of the input distribution and the learning models output confidence measures or uncertainty intervals. These approaches are often descriptive and do not constrain how sensor values may deviate, nor do they provide system-level safety guarantees.

Perception and sensor contracts A closely related line of work studies contract-based reasoning for sensing and/or perception components in CPSs [58, 55, 5, 68, 45, 18]. Early contract-based design work develop contracts as assumption–guarantee abstractions for component composition [58, 55, 18]. The notion of perception contracts have been developed for safety of ML-enabled systems, where contracts characterize perception error relative to ground truth and support safety reasoning for controllers interacting with neural perception modules [5, 68, 45]. Our work is complementary to these efforts, by focusing on admissible (recurring) abnormality patterns of sensor estimates and establishing Q-safety based formal guarantees.

Temporal distance metrics (STL) [51] is a specification formalism for expressing real-time temporal safety and performance properties, such as robustness, of CPSs. Recent results such as [36, 46, 49] investigate quantitative notions of temporal distance and robustness, providing metrics to measure how far a system execution is from satisfying a temporal specification and enabling robustness-based monitoring and analysis. Our work uses Euclidean distances for measuring distances between states and thus Q-safety. This is a design choice that can be adapted to other distance metrics, depending on the application and specification requirements.

Robustness of CPSs Broadly, this work investigates CPS resilience or robustness against sensor uncertainties. Recent work have investigated the robustness of CPSs under attack [6, 17]. Some researchers investigate the ability of the CPSs to mitigate the impact of covert attacks [6]. And some develop metrics to quantify the resilience of CPSs [17]. These works often tackle the resilience problem from the perspective of control theory and focus on the impact of attacks on the state variables. This work tackles the robustness problem by analyzing how abnormalities may threaten the safety properties of CPSs.

Quantitative verification Depending on the context, quantitative information can be a variety of things, e.g., probabilities, time, and energy. Different types of distance metrics have been employed in quantitative verification. Popular modeling formalisms include weighted automata and probabilistic systems, see [20] for more related work. For CPSs, quantitative verification has been commonly investigated for robustness against attacks [79, 80, 16] and other settings, e.g., [6, 22, 52, 65, 54]. This work explores how to integrate Q-safety and its verification seamlessly with CPS semantics, which are not the focus of prior work.

CPS verification The work promotes CPS verification of safety. The importance of ensuring the safety of CPSs motivates a growing body of work on formal verification for hybrid systems [2, 40, 41, 70, 73, 57, 56, 8, 12, 32, 39, 28, 15, 1, 78, 77]. Most of them are for Boolean safety. These works often fall into two main categories: reachability analysis (model checking) and deductive verification (proofs). Hybrid automata [3, 48] is a well-known formalism for hybrid systems and is verified primarily with reachability analysis. A lot of tools have been developed, e.g., flow* [14] and SpaceEx [24]. In recent years, a lot of work has emerged on the safety verification of CPSs with ML-based controllers [31, 47, 29, 69, 75, 74, 33, 82], most of which use reachability analysis. This work on $\mathsf{d}\mathcal{L}$ can be considered as the other category: deductive reasoning. $\mathsf{d}\mathcal{L}$ has been used in case studies [63, 60]. In addition to $\mathsf{d}\mathcal{L}$, deductive reasoning of hybrid systems has been explored in proof systems, such as PVS [67, 66], Coq [4, 50], and Isabelle/HOL [23, 53].

7 Conclusion

We presented a design of tolerance contracts for sensor estimates that enable safety guarantees for CPSs subject to sensor uncertainty. This work focuses on the theoretical foundations of tolerance contracts and their safety guarantees. The case study demonstrates how different contract designs lead to different Q-safety guarantees and levels of sensing tolerance. We argue that this combination of Q-safety and tolerance contracts has strong potential to support well-balanced CPS design and implementation, by tuning contract parameters.

Future Work We plan to develop extensions to explore this potential. Immediate directions include (1) understanding quantitative relations between contract parameters and Q-safety, (2) refining cooldown constraints to capture more expressive recovery policies, and (3) generating runtime enforcement, e.g., monitors, for the tolerance contracts in CPS implementations. More heavy-weight future directions include (i) developing systematic frameworks for trading off CPS objectives such as Q-safety, robustness, and performance, (ii) studying compositional tolerance contracts and their implications for Q-safety analysis, and (iii) synthesizing tolerance contracts from real-world datasets for learn-based CPSs.

References

1. Alan, A., Taylor, A.J., He, C.R., Ames, A.D., Orosz, G.: Control barrier functions and input-to-state safety with application to automated vehicles. IEEE Transac-

tions on Control Systems Technology **31**(6), 2744–2759 (2023)
2. Alur, R.: Principles of cyber-physical systems. MIT Press (2015)
3. Alur, R., Courcoubetis, C., Henzinger, T.A., Ho, P.H.: Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In: International hybrid systems workshop. pp. 209–229. Springer (1991)
4. Anand, A., Knepper, R.: ROSCoq: Robots powered by constructive reals. In: ITP. pp. 34–50. Springer (2015)
5. Astorga, A., Hsieh, C., Madhusudan, P., Mitra, S.: Perception contracts for safety of ML-enabled systems. Proceedings of the ACM on Programming Languages **7**(OOPSLA2), 2196–2223 (2023)
6. Barbeau, M., Cuppens, F., Cuppens, N., Dagnas, R., Garcia-Alfaro, J.: Resilience estimation of cyber-physical systems via quantitative metrics. IEEE Access **9**, 46462–46475 (2021)
7. Barfoot, T.D.: State estimation for robotics. Cambridge University Press (2024)
8. Barrère, M., Hankin, C., Nicolaou, N., Eliades, D.G., Parisini, T.: Measuring cyber-physical security in industrial control systems via minimum-effort attack strategies. J. Inf. Secur. Appl. **52**, 102471 (2020)
9. Bohrer, R., Tan, Y.K., Mitsch, S., Myreen, M.O., Platzer, A.: Veriphy: Verified controller executables from verified cyber-physical system models. In: PLDI. pp. 617–630 (2018)
10. Boyce, W.E., DiPrima, R.C.: Elementary differential equations and boundary value problems. Wiley (2012)
11. Boyd, S., Boyd, S.P., Vandenberghe, L.: Convex optimization. Cambridge university press (2004)
12. Bresolin, D., Collins, P., Geretti, L., Segala, R., Villa, T., Gonzalez, S.Z.: A computable and compositional semantics for hybrid automata. In: HSCC. pp. 18:1–18:11. ACM (2020)
13. Catak, F.O., Yue, T., Ali, S.: Prediction surface uncertainty quantification in object detection models for autonomous driving. In: AITest. pp. 93–100. IEEE (2021)
14. Chen, X., Ábrahám, E., Sankaranarayanan, S.: Flow*: An analyzer for non-linear hybrid systems. In: CAV. pp. 258–263. Springer (2013)
15. Choi, H., Lee, W.C., Aafer, Y., Fei, F., Tu, Z., Zhang, X., Xu, D., Deng, X.: Detecting attacks against robotic vehicles: A control invariant approach. In: CCS. pp. 801–816 (2018)
16. Chong, S., Lanotte, R., Merro, M., Tini, S., Xiang, J.: Quantitative robustness analysis of sensor attacks on cyber-physical systems. In: HSCC. pp. 1–12 (2023)
17. Dagnas, R., Barbeau, M., Boutin, M., Garcia-Alfaro, J., Yaich, R.: Exploring the quantitative resilience analysis of cyber-physical systems. In: IFIP. pp. 1–6. IEEE (2023)
18. Dreossi, T., Donzé, A., Seshia, S.A.: Compositional falsification of cyber-physical systems with machine learning components. Journal of Automated Reasoning **63**(4), 1031–1053 (2019)
19. Efrat, N., Bluvstein, M., Garnett, N., Levi, D., Oron, S., Shlomo, B.E.: Semi-local 3d lane detection and uncertainty estimation. arXiv preprint arXiv:2003.05257 (2020)
20. Fahrenberg, U.: A generic approach to quantitative verification. arXiv preprint arXiv:2204.11302 (2022)
21. Fainekos, G.E., Pappas, G.J.: Robustness of temporal logic specifications for continuous-time signals. Theoretical Computer Science **410**(42), 4262–4291 (2009)

22. Ferrère, T., Nickovic, D., Donzé, A., Ito, H., Kapinski, J.: Interface-aware signal temporal logic. In: HSCC. pp. 57–66. ACM (2019)
23. Foster, S., Huerta y Munive, J.J., Gleirscher, M., Struth, G.: Hybrid systems verification with Isabelle/HOL: Simpler syntax, better models, faster proofs. In: FM. LNCS, vol. 13047, pp. 367–386. Springer (2021)
24. Frehse, G., Le Guernic, C., Donzé, A., Cotton, S., Ray, R., Lebeltel, O., Ripado, R., Girard, A., Dang, T., Maler, O.: SpaceEx: Scalable verification of hybrid systems. In: CAV. pp. 379–395. Springer (2011)
25. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: ICML. pp. 1050–1059 (2016)
26. Hammam, A., Bonarens, F., Ghobadi, S.E., Stiller, C.: Predictive uncertainty quantification of deep neural networks using dirichlet distributions. In: CSCS. pp. 1–10 (2022)
27. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press (2000)
28. Hobbs, K.L., Mote, M.L., Abate, M.C., Coogan, S.D., Feron, E.M.: Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems. IEEE Control Systems Magazine **43**(2), 28–65 (2023)
29. Huang, C., Fan, J., Li, W., Chen, X., Zhu, Q.: Reachnn: Reachability analysis of neural-network controlled systems. ACM Transactions on Embedded Computing Systems **18**(5s), 1–22 (2019)
30. Itkina, M., Kochenderfer, M.: Interpretable self-aware neural networks for robust trajectory prediction. In: CoRL. pp. 606–617 (2023)
31. Ivanov, R., Weimer, J., Alur, R., Pappas, G.J., Lee, I.: Verisig: verifying safety properties of hybrid systems with neural network controllers. In: HSCC. pp. 169–178 (2019)
32. Jahandideh, I., Ghassemi, F., Sirjani, M.: An actor-based framework for asynchronous event-based cyber-physical systems. Software and Systems Modeling **20**, 641–665 (2021)
33. Kochdumper, N., Schilling, C., Althoff, M., Bak, S.: Open-and closed-loop neural network verification using polynomial zonotopes. In: NFM. pp. 16–36. Springer (2023)
34. Kong, L., Sun, J., Zhang, C.: SDE-net: Equipping deep neural networks with uncertainty estimates. arXiv preprint arXiv:2008.10546 (2020)
35. Kraus, F., Dietmayer, K.: Uncertainty estimation in one-stage object detection. In: ITSC. pp. 53–60 (2019)
36. Kuhlmann, I., Corea, C.: Inconsistency measurement in LTL_f based on minimal inconsistent sets and minimal correction sets. In: International Conference on Scalable Uncertainty Management. pp. 217–232 (2024)
37. Kumari, N., Priya, S.K., Kumar, A., Fogla, A., et al.: Automatic ai controller that can drive with confidence: steering vehicle with uncertainty knowledge. arXiv preprint arXiv:2404.16893 (2024)
38. Lakshminarayanan, B., Pritzel, A., Blundell, C.: Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems **30** (2017)
39. Lanotte, R., Merro, M., Munteanu, A., Viganò, L.: A Formal Approach to Physics-based Attacks in Cyber-physical Systems. ACM Transactions on Privacy and Security **23**(1), 3:1–3:41 (2020)
40. Larsen, K.G.: Verification and performance analysis for embedded systems. In: TASE. pp. 3–4 (2009)

41. Lee, E.A., Seshia, S.A.: Introduction to embedded systems: A cyber-physical systems approach. MIT press (2016)
42. Lee, M., Mudassar, B., Mukhopadhyay, S.: Lightweight model uncertainty estimation for deep neural object detection. In: IJCNN. pp. 1–8 (2022)
43. Li, D., Liu, B., Huang, Z., Hao, Q., Zhao, D., Tian, B.: Safe motion planning for autonomous vehicles by quantifying uncertainties of deep learning-enabled environment perception. IEEE Transactions on Intelligent Vehicles **9**(1), 2318–2332 (2023)
44. Li, G., Li, Z., Knoop, V.L., van Lint, H.: Unravelling uncertainty in trajectory prediction using a non-parametric approach. Transportation Research Part C: Emerging Technologies **163**, 104659 (2024)
45. Li, Y., Yang, B.C., Jia, Y., Zhuang, D., Mitra, S.: Refining perception contracts: Case studies in vision-based safe auto-landing. arXiv preprint arXiv:2311.08652 (2023)
46. Liu, R., Hou, A., Li, S., Yin, X.: Sagas: Semantic-aware graph-assisted stitching for offline temporal logic planning. arXiv preprint arXiv:2512.00775 (2025)
47. Lopez, D.M., Choi, S.W., Tran, H.D., Johnson, T.T.: NNV 2.0: the neural network verification tool. In: CAV. pp. 397–412 (2023)
48. Lynch, N., Segala, R., Vaandrager, F., Weinberg, H.B.: Hybrid I/O automata. In: International Hybrid Systems Workshop. pp. 496–510. Springer (1995)
49. Madsen, C., Vaidyanathan, P., Sadraddini, S., Vasile, C.I., DeLateur, N.A., Weiss, R., Densmore, D., Belta, C.: Metrics for signal temporal logic formulae. In: CDC. pp. 1542–1547 (2018)
50. Malecha, G., Ricketts, D., Alvarez, M.M., Lerner, S.: Towards foundational verification of cyber-physical systems. In: SOSCYPS. pp. 1–5 (2016)
51. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: FORMATS/FTRTFT. LNCS, vol. 3253, pp. 152–166 (2004)
52. Mohammadinejad, S., Deshmukh, J.V., Puranic, A.G.: Mining environment assumptions for cyber-physical system models. In: ICCPS. pp. 87–97 (2020)
53. Huerta y Munive, J.J., Struth, G.: Predicate transformer semantics for hybrid systems. Journal of Automated Reasoning **66**(1), 93–139 (2022)
54. Murino, G., Armando, A., Tacchella, A.: Resilience of cyber-physical systems: an experimental appraisal of quantitative measures. In: CyCon. vol. 900, pp. 1–19 (2019)
55. Naik, N., Nuzzo, P.: Robustness contracts for scalable verification of neural network-enabled cyber-physical systems. In: MEMOCODE. pp. 1–12 (2020)
56. Nigam, V., Talcott, C.L.: Formal Security Verification of Industry 4.0 Applications. In: ETFA. pp. 1043–1050 (2019)
57. Nigam, V., Talcott, C., Urquiza, A.: Towards the Automated Verification of Cyber-Physical Security Protocols: Bounding the Number of Timed Intruders. In: ESORICS. pp. 450–470 (2016)
58. Nuzzo, P., Sangiovanni-Vincentelli, A., Sun, X., Puggelli, A.: Methodology for the design of analog integrated interfaces using contracts. IEEE Sensors Journal **12**(12), 3329–3345 (2012)
59. Platzer, A.: Differential dynamic logic for hybrid systems. Journal of Automated Reasoning **41**(2), 143–189 (2008)
60. Platzer, A.: Logic & proofs for cyber-physical systems. In: IJCAR. pp. 15–21 (2016)
61. Platzer, A.: A complete uniform substitution calculus for differential dynamic logic. Journal of Automated Reasoning **59**(2), 219–265 (2017)

62. Platzer, A.: Logical Foundations of Cyber-Physical Systems. Springer (2018)
63. Platzer, A., Quesel, J.D.: European train control system: A case study in formal verification. In: ICFEM. pp. 246–265 (2009)
64. Postels, J., Ferroni, F., Coskun, H., Navab, N., Tombari, F.: Sampling-free epistemic uncertainty estimation using approximated variance propagation. In: ICCV. pp. 2931–2940 (2019)
65. Semertzis, I., Rajkumar, V.S., Ştefanov, A., Fransen, F., Palensky, P.: Quantitative risk assessment of cyber attacks on cyber-physical systems using attack graphs. In: MSCPES. pp. 1–6 (2022)
66. Slagel, J.T., Moscato, M., White, L., Muñoz, C.A., Balachandran, S., Dutle, A.: Embedding differential dynamic logic in PVS. arXiv preprint arXiv:2404.15214 (2024)
67. Slagel, J.T., White, L., Dutle, A.: Formal verification of semi-algebraic sets and real analytic functions. In: CPP. pp. 278–290 (2021)
68. Sun, D., Yang, B.C., Mitra, S.: Learning-based perception contracts and applications. arXiv preprint arXiv:2309.13515 (2023)
69. Sun, X., Khedr, H., Shoukry, Y.: Formal verification of neural network controlled autonomous systems. In: HSCC. pp. 147–156 (2019)
70. Tabuada, P.: Verification and control of hybrid systems: a symbolic approach. Springer (2009)
71. Tagasovska, N., Lopez-Paz, D.: Single-model uncertainties for deep learning. Advances in neural information processing systems **32** (2019)
72. Tang, X., Yang, K., Wang, H., Wu, J., Qin, Y., Yu, W., Cao, D.: Prediction-uncertainty-aware decision-making for autonomous vehicles. IEEE Transactions on Intelligent Vehicles **7**(4), 849–862 (2022)
73. Tiwari, A.: Logic in software, dynamical and biological systems. In: LICS. pp. 9–10 (2011)
74. Tran, H.D., Cai, F., Diego, M.L., Musau, P., Johnson, T.T., Koutsoukos, X.: Safety verification of cyber-physical systems with reinforcement learning control. ACM Transactions on Embedded Computing Systems **18**(5s), 1–22 (2019)
75. Tran, H.D., Yang, X., Manzanas Lopez, D., Musau, P., Nguyen, L.V., Xiang, W., Bak, S., Johnson, T.T.: NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In: CAV. pp. 3–17 (2020)
76. Wang, Y., Wang, T., Yue, T.: Uncertainty propagation from sensor data to deep learning models in autonomous driving. Information and Software Technology **183**, 107735 (2025)
77. Xiang, J., Chong, S.: Extending dynamic logics with first-class relational reasoning. In: NASA formal method symposium (2025)
78. Xiang, J., Fulton, N., Chong, S.: Relational analysis of sensor attacks on cyber-physical systems. In: CSF. pp. 1–16 (2021)
79. Xiang, J., Lanotte, R., Tini, S., Chong, S., Merro, M.: Measuring robustness in cyber-physical systems under sensor attacks. Nonlinear Analysis: Hybrid Systems **56**, 101559 (2025)
80. Xiang, J., Tini, S., Lanotte, R., Merro, M.: Formal robustness for cyber-physical systems under timed attacks. In: CSF (2025)
81. Zhang, W., Ma, Z.M., Das, S., Weng, T.W.L., Megretski, A., Daniel, L., Nguyen, L.M.: One step closer to unbiased aleatoric uncertainty estimation. In: AAAI (2024)
82. Zhang, Y., Xu, X.: Reachability analysis and safety verification of neural feedback systems via hybrid zonotopes. In: ACC. pp. 1915–1921 (2023)