Quantitative Robustness Analysis of Sensor Attacks on Cyber-Physical Systems

Stephen Chong*
chong@seas.harvard.edu
Harvard University
Boston, Massachusetts, USA

Ruggero Lanotte*
ruggero.lanotte@uninsubria.it
University of Insubria
Como, Italy

Massimo Merro*
massimo.merro@univr.it
University of Verona
Verona, Italy

Simone Tini*
simone.tini@uninsubria.it
University of Insubria
Como, Italy

Jian Xiang*
jxiang@seas.harvard.edu
Harvard University
Boston, Massachusetts, USA

Abstract

This paper contributes a formal framework for *quantitative analysis* of bounded *sensor attacks* on cyber-physical systems, using the formalism of *differential dynamic logic*. Given a precondition and postcondition of a system, we formalize two quantitative *safety* notions, quantitative forward and backward safety, which respectively express (1) how strong the strongest postcondition of the system is with respect to the specified postcondition, and (2) how strong the specified precondition is with respect to the weakest precondition of the system needed to ensure the specified postcondition holds. We introduce two notions, forward and backward *robustness*, to characterize the robustness of a system against sensor attacks as the loss of safety. Two simulation distances, which respectively characterize upper bounds of the degree of forward and backward safety loss caused by the sensor attacks, are developed to reason with robustness. We verify the two simulation distances by expressing them as formulas of differential dynamic logic, and proving the formulas with existing tool support. We showcase an example of an autonomous vehicle that needs to avoid a collision.

CCS Concepts: • **Theory of computation → Logic and verification**; • **Security and privacy → Formal security models**; • **Computer systems organization → Embedded and cyber-physical systems**.

Keywords: formal method, robustness, differential dynamic logic, quantitative analysis

ACM Reference Format:
Stephen Chong, Ruggero Lanotte, Massimo Merro, Simone Tini, and Jian Xiang. 2023. Quantitative Robustness Analysis of Sensor Attacks on Cyber-Physical Systems. In *Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control (HSCC '23), May 9–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3575870.3587118

*All authors contributed equally to this research.

HSCC '23, May 9–12, 2023, San Antonio, TX, USA

ACM ISBN 979-8-4007-0033-0/23/05…$15.00
https://doi.org/10.1145/3575870.3587118

1 Introduction

Cyber-physical systems (CPSs), which consist of both physical and cyber components, suffer from a broad attack surface, including both software controllers and physical components. A peculiar class of attacks in such systems is the so-called *physics-based attacks*: attacks targeting the physical devices (sensors and actuators) of CPSs [17, 26]. For instance, *sensor attacks*, such as DoS or integrity attacks on sensors, may lead to crashing the system under attack [41], or allow an adversary to control the system [8, 9].

The importance of ensuring the *safety* of CPSs motivates a growing body of work on formal verification for embedded and hybrid systems [2, 4, 7, 22, 29, 30, 33, 34, 42, 44], some of which focus on the analysis of sensor-related attacks [26, 27, 45]. Existing work often treats satisfaction of safety as a boolean predicate: either a system satisfies a desired safety property or it does not. However, a simple yes/no answer doesn't fit the setting of CPSs, which interact with continuous and quantitative entities, such as measurements of the controlled physical process. For example, under the same road conditions, a vehicle with a shorter braking distance towards an obstacle is considered safer than a vehicle with a longer braking distance, even if both of them can brake in time. Thus, when working with CPSs, a *quantitative* notion of safety can be much more informative than standard safety.

However, knowing the degree of safety of a correct CPS is not enough to analyze the effect of attacks targeting its sensors. For example, a vehicle with a very short braking distance may not be able to tolerate certain attacks on the *obstacle detection system*, resulting in unsafe runtime behaviors. Here, it is important to understand the *robustness* of a system's safety under sensor attacks, that is, how the safety may change because of sensor attacks. For example, consider a vehicle equipped with a self-braking system whose *safety requirement* is to brake from the speed of 100 km/h when an obstacle is detected 40 meters away. And suppose the vehicle, at that speed, starts braking when the obstacle is detected 60 meters away. Assume that an adversary is able to perturb the readings of the distance to an obstacle by 10 meters without being detected. Then, the vehicle is still safe as it starts braking, at the speed of 100 km/h, when the obstacle is 50 meters away; 10 meters more than the safety requirements. The degree of safety loss is a clear indicator of the vehicle's robustness against such an attack.

In this work, we define two notions of *quantitative safety* for CPSs and use them to analyze a system's robustness under sensor attacks. Our threat model assumes *bounded sensor attacks*, that is, attacks that may compromise a subset of sensors and offset their readings to some degree. Using bounded sensor attacks, the attacker may slowly drag the physical process of the target CPSs into unsafe states, in a (possibly) *stealthy* manner, i.e., without being (promptly) detected by Intrusion Detection Systems (IDSs). We do not model or discover the mechanisms by which attackers manipulate sensor values; we simply assume they are able to do so. We also assume every system has a known *precondition* and *postcondition*. The precondition specifies the initial conditions and environment when the system starts operating, and the postcondition specifies the desired condition that the system should always satisfy for it to be safe.

The first notion is *forward quantitative safety*, which estimates the room for maneuver to ensure that the system remains safe after any execution starting from a state satisfying the precondition. It basically estimates how strong the strongest postcondition is with respect to the desired safety postcondition. Said in other words, given a precondition, forward safety provides a quantification of the margins on possible strengthening of the safety postcondition with respect to the strongest postcondition. Technically, it is defined as the *shortest* distance between the set of states satisfying the strongest postcondition and the set of unsafe states. The larger this distance is, the further away the system's reachable states are from unsafe states, and thus the safer the system is. Built upon forward safety, we introduce *forward robustness*, which characterizes the impact of a sensor attack as a ratio: the degree of forward safety of the compromised system over the degree of forward safety of the original system. Intuitively, the closer the ratio gets to 1, the more robust the original system is against the attack. A ratio of 1 means the attack doesn't weaken the safety guarantee at all.

The second notion is *backward quantitative safety*, which provides a degree of safety by estimating the room for maneuver to ensure that the system remains safe with respect to a given postcondition by weakening the precondition. It basically estimates how strong the specified precondition is with respect to the weakest precondition needed to ensure the safety of the system after its execution. Said in other words, given a safety postcondition, backward safety provides a quantification of the precondition with respect to the weakest precondition. Technically, it is defined as the *shortest* distance between the set of states satisfying the weakest precondition and the set of "bad" initial states that may lead the system to unsafe states. The larger this distance is, the further away the system's states that satisfy precondition are from "bad" initial states, and thus the safer the system is. Built upon backward safety, we introduce *backward robustness* that characterizes the impact of a sensor attack as a ratio: the degree of backward safety of the compromised system over the original system. Similar to forward robustness, the closer the ratio gets to 1, the more robust the original system is against the attack.

The two robustness notions together give system designers a good way to understand and compare different design candidates by focusing either on preconditions or on postconditions. For example, if a system is likely to suffer from sensor attacks, a designer may simply choose a candidate design with better degrees of robustness. If one degree of robustness (e.g., forward robustness) is identical or similar among different designs, the designers may use the other (e.g., backward robustness) to compare the designs.

To reason about forward (and backward) robustness, we introduce a forward (and backward) simulation distance to, respectively, provide an upper bound of the degree of loss of forward (and backward) safety caused by sensor attacks. The simulation distances are defined based on the behavioral distances [16] between the original system and the system with compromised sensors. In particular, the forward simulation distance characterizes the *forward distance* between the two systems by quantifying the distance between their reachable states, given the same set of initial states. Thus, the forward distance between the original and the compromised system returns an upper bound on the admissible perturbations introduced by a sensor attack on the safety of the behaviors originating from a desired precondition. Analogously, the backward simulation distance characterizes the *backward distance* between the two systems by quantifying the distance between their sets of safe initial states, i.e., those states that never lead the system to an unsafe state, given the same set of safe final states. Thus, the backward distance between the original and the compromised system returns an upper bound on the admissible perturbations introduced by a sensor attack on the initial states leading to possible violations of safety, given a desired postcondition. We prove that the forward (and backward) simulation distance represents a sound proof-technique for calculating upper bounds of forward (and backward) robustness as it returns upper bounds of the loss of forward (and backward) safety caused by sensor attacks.

In the paper, we work within the formalism of hybrid programs and *differential dynamic logic* (d$\mathcal{L}$) [36–38]. Hybrid programs are a formalism for modeling systems that have both continuous and discrete dynamic behaviors. Hybrid programs can express continuous evolution (as differential equations) as well as discrete transitions. Differential dynamic logic is the dynamic logic of hybrid programs, which is used to specify and verify safety properties.

To verify forward (and backward) simulations, we express them as d$\mathcal{L}$ formulas and use the theorem prover developed for d$\mathcal{L}$, KeYmaera X [15], to verify the formulas. We present the two encodings and showcase with examples.

The main contributions of this paper are the following:

- The notions of forward and backward quantitative safety in the context of differential dynamic logic that models safety properties of CPSs (Section 3).
- The notions of forward and backward quantitative robustness for systems under bounded sensor attacks, defined using the two notions of quantitative safety (Section 4).
- Two simulation distances, forward and backward simulation distances over hybrid programs, to reason with robustness (Section 5).
- d$\mathcal{L}$ encodings to express the two simulation relations so we can verify them in KeYmaera X (Section 6).

We introduce preliminaries in Section 2. In Section 7, we demonstrate all notions and techniques with a *case study* on collision avoidance of autonomous vehicles. Section 8 discusses related work, and Section 9 concludes.

2 Preliminaries

2.1 Differential Dynamic Logic

Hybrid programs [38] are a formalism for modeling systems that have both continuous and discrete dynamic behaviors. Hybrid programs can express continuous evolution (as differential equations) as well as discrete transitions.

Figure 1 gives the syntax for hybrid programs. Variables are real-valued and can be deterministically assigned ($x := \theta$, where θ is a real-valued term) or nondeterministically assigned ($x := *$).

$$
\begin{aligned}
\theta, \delta \quad &::= \quad x \mid c \mid \theta \oplus \delta \\
\alpha, \beta \quad &::= \quad x := \theta \mid x := * \mid x' = \theta \,\&\phi \mid ?\phi \mid \alpha\,;\,\beta \mid \alpha \cup \beta \mid \alpha^* \\
\phi, \psi \quad &::= \quad \bot \mid \theta \sim \delta \mid \neg\phi \mid \phi \wedge \psi \mid \forall x.\, \phi \mid [\alpha]\phi
\end{aligned}
$$

Figure 1. Syntax of hybrid programs and d$\mathcal{L}$

Term semantics

$$
\begin{aligned}
\omega[\![x]\!] &= \omega(x) \\
\omega[\![c]\!] &= c \\
\omega[\![\theta \oplus \delta]\!] &= \omega[\![\theta]\!] \oplus \omega[\![\delta]\!] \text{ where } \oplus \text{ denotes corresponding arithmetic operations for } \oplus \in \{+, \times\}
\end{aligned}
$$

Program semantics

$$
\begin{aligned}
[\![x := \theta]\!] &= \{(\omega, \nu) \mid \nu(x) = \omega[\![\theta]\!] \text{ and for all other variables } z \neq x,\ \nu(z) = \omega(z)\} \\
[\![x := *]\!] &= \{(\omega, \nu) \mid \nu(z) = \omega(z) \text{ for all variables } z \neq x\} \\
[\![x' = \theta \,\&\phi]\!] &= \{(\omega, \nu) \mid \text{exists solution } \varphi : [0, r] \mapsto \textsc{Sta} \text{ of } x' = \theta \text{ with } \varphi(0) = \omega \text{ and } \varphi(r) = \nu, \text{ and } \varphi(t) \models \phi \text{ for all } t \in [0, r]\} \\
[\![?\phi]\!] &= \{(\omega, \omega) \mid \omega \models \phi\} \\
[\![\alpha\,;\,\beta]\!] &= \{(\omega, \nu) \mid \exists \mu, (\omega, \mu) \in [\![\alpha]\!] \text{ and } (\mu, \nu) \in [\![\beta]\!]\} \\
[\![\alpha \cup \beta]\!] &= [\![\alpha]\!] \cup [\![\beta]\!] \\
[\![\alpha^*]\!] &= [\![\alpha]\!]^*, \text{ the transitive, reflexive closure of } [\![\alpha]\!]
\end{aligned}
$$

Formula semantics

$$
\begin{aligned}
[\![\bot]\!] &= \emptyset \\
[\![\theta \sim \delta]\!] &= \{\ \omega \mid \omega[\![\theta]\!] \sim \omega[\![\delta]\!]\ \}, \text{ where } \sim \text{ denotes comparison for } \sim \in \{=, \leq, <, \geq, >\} \\
[\![\neg\phi]\!] &= \textsc{Sta} \setminus [\![\phi]\!] \\
[\![\phi \wedge \psi]\!] &= [\![\phi]\!] \cup [\![\psi]\!] \\
[\![\forall x.\, \phi]\!] &= [\![[x := *]\phi]\!] \\
[\![[\alpha]\phi]\!] &= \{\ \omega \mid \forall \nu \text{ if } (\omega, \nu) \in [\![\alpha]\!] \text{ then } \nu \in [\![\phi]\!]\}
\end{aligned}
$$

Figure 2. Semantics of hybrid programs and d$\mathcal{L}$ formulas

Hybrid program $x' = \theta \,\&\phi$ expresses the continuous evolution of variables: given the current value of variable x, the system follows the differential equation $x' = \theta$ for some (nondeterministically chosen) amount of time so long as the formula ϕ , the *evolution domain constraint*, holds for all of that time. Note that x can be a vector of variables and then θ is a vector of terms of the same dimension.

Hybrid programs also include the operations of Kleene algebra with tests [23]: sequential composition, nondeterministic choice, nondeterministic repetition, and testing whether a formula holds.

Differential dynamic logic (d$\mathcal{L}$) [36–38] is the dynamic logic [18] of hybrid programs. Figure 1 also gives the syntax for d$\mathcal{L}$ formulas. In addition to the standard logical connectives of first-order logic, d$\mathcal{L}$ includes primitive propositions that allow comparisons of real-valued terms (which may include derivatives) and *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 necessity can be used to encode modality of *existence*, i.e., $\langle\alpha\rangle\phi = \neg[\alpha]\neg\phi$. Common abbreviations for other logical connectives apply, e.g., $\phi \vee \psi = \neg(\neg\phi \wedge \neg\psi)$ and $\phi \rightarrow \psi = \neg\phi \vee \psi$.

The semantics of d$\mathcal{L}$ [36, 37] is a Kripke semantics in which the Kripke model's worlds are the states of the system. Let $\mathbb{R}$ denote the set of real numbers and $\mathbb{V}$ denote the set of variables. A state is a map $\omega : \mathbb{V} \mapsto \mathbb{R}$ assigning a real value $\omega(x)$ to each variable $x \in \mathbb{V}$. The set of all states is denoted by $\textsc{Sta}$. The semantics of hybrid programs and d$\mathcal{L}$ are shown in Figure 2. We write $[\![\phi]\!]$ to denote the set of states that satisfy formula ϕ. The value of term θ at state ω is denoted $\omega[\![\theta]\!]$. The semantics of a program α is expressed as a transition relation $[\![\alpha]\!]$ between states. If $(\omega, \nu) \in [\![\alpha]\!]$ then there is an execution of α that starts in state ω and ends in state ν.

Safety properties of a system are often defined as follows:

$$
\begin{aligned}
\phi_{pre} &\equiv temp = 100 \\
\phi_{post} &\equiv temp \leq 105 \\
ctrl &\equiv t := 0\,; \\
&\qquad (?temp > 100\,;\, delta := -0.5\,); \\
&\qquad \cup (?temp \leq 100\,;\, delta := 1)\,; \\
plant &\equiv temp' = delta, t' = 1 \,\&(temp \geq 0 \wedge t \leq 1) \\
\phi_{safety} &\equiv \phi_{pre} \rightarrow [(ctrl\,;\, plant)^*]\phi_{post}
\end{aligned}
$$

Figure 3. d$\mathcal{L}$ model of a cooling system

$$
\begin{aligned}
&\ldots \\
ctrl &\equiv temp_s := temp_p\,;\, t := 0\,; \\
&\qquad (?temp_s > 100\,;\, delta := -0.5\,); \\
&\qquad \cup (?temp_s \leq 100\,;\, delta := 1)\,; \\
plant &\equiv temp_p{}' = delta, t' = 1 \,\&(temp_p \geq 0 \wedge t \leq 1)
\end{aligned}
$$

Figure 4. d$\mathcal{L}$ model of a cooling system with sensing

Definition 1 (Safety). *A hybrid program α is* safe *for ϕ_{post} assuming ϕ_{pre}, denoted* $\textsc{safe}(\alpha, \phi_{pre}, \phi_{post})$, *if $\phi_{pre} \rightarrow [\alpha]\phi_{post}$ holds.*

$\textsc{safe}(\alpha, \phi_{pre}, \phi_{post})$ means if ϕ_{pre} is true then ϕ_{post} holds after any possible execution of α. The hybrid program α often has the form $(ctrl\,;\, plant)^*$, where $ctrl$ models atomic actions of the control system and does not contain continuous parts (i.e., differential equations); and $plant$ models evolution of the physical environment and has the form of $x' = \theta \,\&\phi$. That is, the system is modeled as unbounded repetitions of a controller action followed by an update to the physical environment.

For example, consider a simple cooling system that operates in an environment where temperature grows at the rate of 1 degree per minute, shown in Figure 3. Let $temp$ be the current temperature of the environment in degrees. The *safety condition* that we would like to enforce (ϕ_{post}) is that $temp$ is no greater than 105 degrees. Let $delta$ be the rate of change of the temperature (degrees per minute). Let t be the time elapsed since the controller was last invoked. The program $plant$ describes how the physical environment evolves over time interval (1 second): temperature changes according to $delta$ (i.e., $temp' = delta$) and time passes constantly (i.e., $t' = 1$). The differential equations evolve only within the time interval $t \leq 1$ and if $temp$ is non-negative (i.e., $temp \geq 0$).

The hybrid program $ctrl$ models the system's controller. If the temperature is above 100 degrees, the system activates cooling and the temperature drops at a rate of 0.5 degrees per time unit (i.e., $delta := -0.5$). The controller doesn't activate cooling under other temperatures. Then the temperature would grow at the rate of 1 degree per minute (i.e., $delta := 1$).

The formula to be verified, ϕ_{safety}, is shown at the last line of Figure 3. Given an appropriate precondition ϕ_{pre}, the axioms and proof rules of d$\mathcal{L}$ can be used to prove that the safety condition ϕ_{post} holds. For this model, assuming the precondition of initial temperature of 100 degrees, i.e., ϕ_{pre}, we want to ensure the temperature stays no greater than 105 degrees, i.e., ϕ_{post}. The tactic-based theorem prover KeYmaera X [15] provides tool support.

To present some of our definitions, we need to refer to the variables that occur in a hybrid program [37, 38]. We write $\textsc{Var}(\alpha)$ and $\textsc{Var}(\phi)$ to denote, respectively, the set of all variables of program α and formula ϕ. Their definitions can be found in Appendix A.

2.2 Modeling Sensor Attacks

Recent work introduces a framework for modeling and analyzing sensor attacks in the setting of hybrid programs and d$\mathcal{L}$ [45]. It models sensing by separately representing physical values and their sensor reads, and then requires that variables holding sensor reads are equal to the underlying sensor's value. See, for instance, Figure 3 and Figure 4. Here, $temp_p$ represents the actual physical temperature and it changes according to $delta$, while $temp_s$ represents the variable into which the sensor's value is read. The controller program $ctrl$ sets the sensed values equal to the physical values, i.e., $temp_s := temp_p$, to indicate the sensor is working correctly.

Models of a system under sensor attack can be then derived by manipulating the variables representing the sensor reads. For example, with the model shown in Figure 4, an attack on the temperature sensor can be modeled by replacing the constraint $temp_s = temp_p$ with $temp_s := *$, allowing $temp_s$ to take arbitrary values.

We later extend this approach to model *bounded sensor attacks.*

2.3 Distance Metrics

To conduct quantitative analysis, we define a notion of *distance between states*, using the *Euclidean distance* $\rho : \textsc{Sta} \times \textsc{Sta} \to \mathbb{R}$:

$$\rho(\omega, \nu) = \sqrt{\sum_{x \in \mathbb{V}} (\omega(x) - \nu(x))^2}$$

Notice that ρ is a metric, namely, it satisfies the following properties: (1) $\rho(\omega, \nu) = 0$ if and only if $\omega = \nu$, (2) $\rho(\omega, \nu) = \rho(\nu, \omega)$, and (3) $\rho(\omega, \nu) \leq \rho(\omega, \mu) + \rho(\mu, \nu)$ for $\omega, \nu, \mu \in \textsc{Sta}$.

For a state ω and a real $\epsilon > 0$, the ball of ray ϵ centered in ω is the set of states $\mathbf{B}(\omega, \epsilon) = \{\nu | \rho(\omega, \nu) \leq \epsilon\}$.

We adopt existing notions [6, 11] to specify the distance between a state and a set of states:

- The distance between a state ω and a set of states $\mathcal{S} \subseteq \textsc{Sta}$ is the *shortest* distance between ω and all states in $\mathcal{S}$, that is, $\mathbf{dist}(\omega, \mathcal{S}) = \mathbf{inf}\{\rho(\omega, \nu) | \nu \in \mathcal{S}\}$
- The *depth* of ω in $\mathcal{S} \subseteq \textsc{Sta}$ is the *shortest* distance between ω and the *boundary* of $\mathcal{S}$, that is, $\mathbf{depth}(\omega, \mathcal{S}) = \mathbf{inf}\{\rho(\omega, \nu) | \nu \in (\textsc{Sta} \setminus \mathcal{S})\}$
- The *signed distance* between ω and a set of states $\mathcal{S} \subseteq \textsc{Sta}$ is defined as follows:

$$\mathbf{Dist}(\omega, \mathcal{S}) = \begin{cases} \mathbf{depth}(\omega, \mathcal{S}), & \text{if } \omega \in \mathcal{S} \\ -\mathbf{dist}(\omega, \mathcal{S}), & \text{if } \omega \notin \mathcal{S} \end{cases}$$

Note that in the first case the distance is a positive real number, while in the second case the distance is negative. Thus, $\mathbf{Dist}(\omega, \mathcal{S}) > 0$ implies that $\mathbf{B}(\omega, \epsilon) \subseteq \mathcal{S}$ for all $\epsilon < \mathbf{Dist}(\omega, \mathcal{S})$, whereas $\mathbf{Dist}(\omega, \mathcal{S}) < 0$ implies that $\mathbf{B}(\omega, \epsilon) \subseteq (\textsc{Sta} \setminus \mathcal{S})$ for all $\epsilon < -\mathbf{Dist}(\omega, \mathcal{S})$. $\mathbf{Dist}(\omega, \mathcal{S}) = 0$ is not very informative.

In all these definitions, we assume that $\mathbf{inf}\, \emptyset = \infty$ and $\mathbf{inf}\, \mathbb{R} = -\infty$. And we consider the operator $\mathbf{inf}$ in the set of $\mathbb{R} \cup \{\infty, -\infty\}$, therefore every set has an infimum.

3 Quantitative Safety

The Boolean notion of safety in d$\mathcal{L}$, e.g., $\textsc{safe}(\alpha, \phi_{pre}, \phi_{post})$, does not provide any quantitative information on how "good" (i.e., safe) the system is. In this section, we introduce two quantitative notions of safety. The two notions are the foundation of defining forward and backward robustness. They are, respectively, built on the *strongest postcondition* and *weakest precondition* in the setting of d$\mathcal{L}$. In defining quantitative safety, we use hybrid program α to model a system of interest, formula ϕ_{pre} as the precondition of the system, and ϕ_{post} as the postcondition.

3.1 Extended d$\mathcal{L}$

To help define quantitative safety, we extend d$\mathcal{L}$ with another syntactic structure: $\phi\langle\alpha\rangle$, which intuitively represents the *strongest postcondition* after the execution of the program α in a state satisfying the precondition ϕ. Its formal definition is the following:

$$[\![\phi\langle\alpha\rangle]\!] = \{\ \nu \mid \exists\omega \text{ such that } \omega \in [\![\phi]\!] \text{ and } (\omega, \nu) \in [\![\alpha]\!]\}$$

Its dual is the modality of necessity $[\alpha]\phi$, which represents the *weakest precondition* to ensure that ϕ is satisfied after any execution of program α. Its formal definition is shown above in Figure 2.

3.2 Forward Quantitative Safety

A quantitative variation to the Boolean notion of safety, e.g., $\textsc{safe}(\alpha, \phi_{pre}, \phi_{post})$, is *forward quantitative safety*, which provides a degree of safety by estimating the room of maneuver to ensure that the system remains in the safety region after any admissible execution. It basically estimates how strong the strongest postcondition $\phi_{pre}\langle\alpha\rangle$ (obtained by the execution of program α in the precondition ϕ_{pre}) is with respect to the postcondition ϕ_{post}. In other words, this degree of safety gives an indication of the margins on possible strengthening of the postcondition ϕ_{post}.

Definition 2 (Forward quantitative safety)**.** *Given a real $u \in \mathbb{R}$ and formula ϕ_{pre} and ϕ_{post}, a hybrid program α is forward u-safe for ϕ_{pre} and ϕ_{post}, denoted F-$\textsc{safe}_u(\alpha, \phi_{pre}, \phi_{post})$, if $u = \mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\alpha\rangle]\!]\}$.*

Given a system α and a precondition ϕ_{pre}, the real number u measures the *shortest* distance between the set of states satisfying the strongest postcondition $\phi_{pre}\langle\alpha\rangle$ and the set of unsafe states. If u is positive, then all reachable states by the system α from initial states satisfying the precondition ϕ_{pre} stay safe. The bigger u is, the safer the system is. On the contrary, if u is negative, then some reachable states violate the safety condition ϕ_{post}. If u is 0, then the system cannot be considered safe as its safety may depend on very small perturbations of the system's variables [11].

For example, consider the cooling system shown in Figure 4, assuming the precondition ϕ_{pre}, during the execution of the system the temperature lays in the real interval $(99.5, 101]$. Then, we have F-$\textsc{safe}_u(\alpha, \phi_{pre}, \phi_{post})$, where $\alpha = (ctrl\,;\,plant)^*$ for $u = 4$. So, u is our "degree of safety" w.r.t. ϕ_{post}: the system will always satisfy the postcondition $temp_p \leq 105$ with a margin of at least 4 degrees. Suppose we have a different postcondition $\phi'_{post} \equiv temp_p <= 101$. In this case, we have F-$\textsc{safe}_u(\alpha, \phi_{pre}, \phi'_{post})$, for $u = 0$, and the system is actually safe, as $\textsc{safe}(\alpha, \phi_{pre}, \phi'_{post})$ holds. However, for a slightly different postcondition $\phi''_{post} \equiv temp_p < 101$, we still have F-$\textsc{safe}_u(\alpha, \phi_{pre}, \phi''_{post})$, for $u = 0$, but the system is actually unsafe, as $\textsc{safe}(\alpha, \phi_{pre}, \phi''_{post})$ is false. This shows that when the degree of safety is 0 we cannot assess the safety of the system.

3.3 Backward Quantitative Safety

Another quantitative safety notion is *backward quantitative safety*, which estimates how strong the precondition is with respect to the required initial condition for the system to be safe. It provides quantitative information on how "good" (i.e., strong) the precondition ϕ_{pre} is with respect to the weakest precondition $[\alpha]\phi_{post}$, while ensuring safety (i.e., ϕ_{post}) after executions of the system α. In other words, this degree of safety gives an indication of the margins on a possible weakening of the precondition ϕ_{pre}. It is defined as the *shortest* of all distances from states that satisfy the precondition to any "bad" initial states that can lead the system to unsafe states.

Definition 3 (Backward quantitative safety)**.** *Given a real $r \in \mathbb{R}$ and formula ϕ_{pre} and ϕ_{post}, a hybrid program α is backward r-safe for ϕ_{pre} and ϕ_{post}, denoted B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$, if $r = \mathbf{inf}\{\mathbf{Dist}(\omega, [\![[\alpha]\phi_{post}]\!]) \mid \omega \in [\![\phi_{pre}]\!]\}$.*

Here, if r is positive then any execution of the system that starts from initial states in ϕ_{pre} shall always stay safe. The bigger r is, the safer the system is. On the contrary, if r is negative, then some initial states in ϕ_{pre} can lead the system's execution to an unsafe state. Similar to the forward quantitative safety, if r is 0 the system cannot be considered safe.

For example, assuming the precondition ($temp_p = 100$) and the postcondition ($temp_p <= 105$), we have B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$, for $r = 5$, since the weakest precondition is $temp_p <= 105$. Then $r = 5$ is our "degree of safety" w.r.t. ϕ_{pre}: we have a room of maneuver of 5 on the precondition to ensure the postcondition after the execution of α.

The Boolean version of safety, SAFE(α , ϕ_{pre}, ϕ_{post}) of Definition 1, can be expressed in terms of backward quantitative safety.

Proposition 1. *Given a program α and formula ϕ_{pre} and ϕ_{post}.*

- *If there is $r > 0$ such that B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$, then SAFE$(\alpha, \phi_{pre}, \phi_{post})$.*
- *If SAFE$(\alpha, \phi_{pre}, \phi_{post})$ then there is $r \geq 0$ such that B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$.*

Note that the two quantitative notions of safety never contradict each other, i.e., if one degree of safety is positive, the other is non-negative. And if one degree is negative, the other is non-positive. So the following proposition holds:

Proposition 2.

- *If F-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$ for some $u > 0$, then B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$ for some $r \geq 0$;*
- *If B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$ for some $r > 0$, then F-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$ for some $u \geq 0$.*

However, the degree of safety of the two notions are not quantitatively related, i.e., given formula ϕ_{pre}, ϕ_{post}, and a hybrid program α, if B-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$ and F-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$ for some $u > 0$ and $r > 0$, the relationship between u and r can be arbitrary.

Note that given a system α, formula ϕ_{pre} and ϕ_{post}, forward quantitative safety, i.e., F-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$ *always* holds for some u, since the infimum always exists (even for unsafe systems whose u is non-positive). The same for backward safety.

The definitions of forward and backward safety build on the notion of weakest precondition and strongest postcondition. Existing work has introduced techniques to analyze them for hybrid programs [13, 21]. In this work, we focus on systems where we can compute the two conditions, without showing details of how we compute them.

4 Quantitative Robustness

In this section, we introduce the threat model, bounded sensor attacks, and two notions of robustness, developed using the two notions of quantitative safety.

$$
\begin{aligned}
ctrl' \equiv\ & temp_s := *;\\
& ?(temp_s \geq temp_p - 0.3 \wedge temp_s \leq temp_p + 0.3);\\
& \dots
\end{aligned}
$$

Figure 5. d$\mathcal{L}$ model of a cooling system under sensor attack (the omitted part of the model is the same as the model in Figure 4)

4.1 Bounded Sensor Attacks

Existing work [45] considers a threat model of sensor attacks that the attackers can arbitrarily manipulate the sensor readings, e.g., compromised temperature sensor is modeled by $temp_s := *$. The threat model is too coarse and strong, in particular, when the system under attacks is equipped with some sort of IDS (for instance, anomaly detection IDSs [17]) that the attacker would like to evade.

In this work, we consider more refined sensor attacks in which the measurement deviation is bounded. Such finer attacks can be modeled by assignments of the form $q_s = q_p + o$, where q_s and q_p respectively represent sensor and physical value of a real-world quantity, and o represents a suitable offset. The idea being that for low values of $|o|$ the attack may remain *stealthy*, i.e., undetected by IDSs. The attack can be formalized as follows:

Definition 4 (Bounded S_A-sensor attack)**.** *Given a hybrid program α, a set of sensors $S_A \subseteq$ VAR(α) and an offset function $o : S_A \rightarrow \mathbb{R}^{\geq 0}$, we write ATTACKED$(\alpha, S_A, o)$ to denote the program obtained by replacing in α all assignments to variables q_s in S_A, with programs of the form $q_s := *; ?(q_s \geq q_p - o(q_s) \wedge q_s \leq q_p + o(q_s))$.*

For example, for the cooling system shown in Figure 4, consider a sensor attack introducing an offset 0.3 to the temperature sensor. Figure 5 shows a model of the system with compromised sensors.

The following theorem states that forward safety is affected by bounded sensor attacks in proportional manner: the stronger the attack is, the lower the degree of safety the attacked system has.

Theorem 1. *Assume a hybrid program α, a set of sensors $S_A \subseteq$ VAR(α) and two offset functions $o_1 : S_A \rightarrow \mathbb{R}^{\geq 0}$ and $o_2 : S_A \rightarrow \mathbb{R}^{\geq 0}$, with $o_1(s) \leq o_2(s)$ for any $s \in S_A$, real numbers $u, u_1, u_2 \in \mathbb{R}$, and properties ϕ_{pre} and ϕ_{post}. Then, if*

- *F-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$*
- *F-SAFE$_{u_1}($ATTACKED$(\alpha, S_A, o_1), \phi_{pre}, \phi_{post})$*
- *F-SAFE$_{u_2}($ATTACKED$(\alpha, S_A, o_2), \phi_{pre}, \phi_{post})$*

then $u_2 \leq u_1 \leq u$.

Intuitively, the theorem holds because the behaviors of a system under a stronger attack subsumes a system under a weaker attack or no attack. The detailed proof can be found in Appendix B.

We can prove a similar theorem for backward safety:

Theorem 2. *Assume a hybrid program α, a set of sensors $S_A \subseteq$ VAR(α) and two offset functions $o_1 : S_A \rightarrow \mathbb{R}^{\geq 0}$ and $o_2 : S_A \rightarrow \mathbb{R}^{\geq 0}$, with $o_1(s) \leq o_2(s)$ for any $s \in S_A$, real numbers $r, r_1, r_2 \in \mathbb{R}$, and properties ϕ_{pre} and ϕ_{post}. Then, if*

- *B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$*
- *B-SAFE$_{r_1}($ATTACKED$(\alpha, S_A, o_1), \phi_{pre}, \phi_{post})$*
- *B-SAFE$_{r_2}($ATTACKED$(\alpha, S_A, o_2), \phi_{pre}, \phi_{post})$*

then $r_2 \leq r_1 \leq r$.

4.2 Quantitative Robustness

With the definitions of quantitative safety, we can characterize the robustness of a system against sensor attacks as the loss of safety. In particular, the robustness notions are defined by comparing the degree of safety of the original system and the system whose sensors have been compromised. We introduce two notions of quantitative robustness, forward and backward robustness, which are built on the notions of forward and backward safety, respectively.

Forward Robustness. The first robustness notion, *forward robustness*, measures, intuitively, how much an attack affects the system's reachable states if the system starts with the expected precondition. Forward robustness characterizes the impact of a sensor attack as a ratio: the degree of safety of the compromised system over the degree of safety of the original system.

Definition 5 (Quantitative forward robustness)**.** *Given a hybrid program α, a set of sensors $S_A \subseteq \textsc{Var}(\alpha)$, an offset function $o : S_A \to \mathbb{R}^{\geq 0}$, real numbers $u, u_1, \delta \in \mathbb{R}$, and properties ϕ_{pre} and ϕ_{post}, we say that α* is forward δ-robust under o-bounded S_A-attacks*, written $\textsc{F-robust}(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$, if*

- *$\textsc{F-safe}_u(\alpha, \phi_{pre}, \phi_{post})$, with $u > 0$*
- *$\textsc{F-safe}_{u_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$*
- *$\delta = u_1/u$.*

As expected, forward robustness applies only to systems that are safe when not exposed to sensor attacks, i.e., $u > 0$. The value of ratio δ indicates the system's robustness under the sensor attack. Note that by Theorem 1, we know that $u_1 \leq u$. We can analyze δ by the following cases:

- $\delta = 1$: the attack doesn't affect the system's forward safety.
- $0 < \delta < 1$: then $0 < u_1 < u$. Given initial states where the precondition holds, reachable states of both the original system and the compromised system stay safe. The value of $(1 - \delta)$ quantifies the *percentage of forward safety that is lost* due to the attack. The closer δ is to 1, the more robust the system is.
- $\delta \leq 0$: then $u > 0$ and $u_1 \leq 0$. Executions of the original system stay safe, but the attack may be able to "break" the system: some of its executions under attack may run into unsafe states. The lower the value of δ, the more effective the attack can be. Even in the case of $u_1 = 0$ ($\delta = 0$), the compromised system can no longer be considered safe.

Consider again the example of the cooling system. We know that $\textsc{F-safe}_4(\alpha, \phi_{pre}, \phi_{post})$ for $\phi_{pre} \equiv temp_p = 100$ and $\phi_{post} \equiv temp_p \leq 105$, where α models the original system. For the compromised system shown in Figure 5, starting again from ϕ_{pre}, during executions of $\textsc{attacked}(\alpha, S_A, o)$, the temperature lies in $(99.2, 101.3]$. Thus we know that $\textsc{F-safe}_{3.7}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$. The degree of forward robustness of the original system with respect to the attack is: $\delta = 3.7/4 = 0.925$.

The value of δ can help engineers evaluate or compare different defense mechanisms against potential attacks. For a specific set of attacks, a mechanism with less safety loss, i.e., bigger δ, may be considered better than another one with more safety loss.

Note that using a ratio for δ is a better indicator of robustness than using an absolute value, e.g., $u - u_1$: it is consistent regardless of the units of measurement used for safety. For example, the ratio of robustness for a braking system w.r.t. a sensor attack would be the same whether the safety is measured in feet or in meters.

Backward Robustness. The second robustness notion, *backward robustness*, measures, intuitively, how resilient the initial states that satisfy the precondition are to sensor attacks whose goal is to drag the system into unsafe states. It characterizes the impact of a sensor attack as a ratio: the degree of safety of the compromised system over the degree of safety of the original system.

Definition 6 (Quantitative backward robustness)**.** *Given a hybrid program α, a set of sensors $S_A \subseteq \textsc{Var}(\alpha)$, an offset function $o : S_A \to \mathbb{R}^{\geq 0}$, real numbers $r, r_1, \delta \in \mathbb{R}$, and properties ϕ_{pre} and ϕ_{post}, we say that α* is backward δ-robust under o-bounded S_A-attacks*, written $\textsc{B-robust}(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$, if*

- *$\textsc{B-safe}_r(\alpha, \phi_{pre}, \phi_{post})$, with $r > 0$*
- *$\textsc{B-safe}_{r_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$*
- *$\delta = r_1/r$.*

Again, backward robustness applies only to systems that are safe when not exposed to sensor attacks (i.e., $r > 0$). We can also analyze δ by the following cases. By Theorem 2, we know that $r_1 \leq r$.

- $\delta = 1$: the attack doesn't affect the system's backward safety.
- $0 < \delta < 1$: then $0 < r_1 < r$. Executions of either α or $\textsc{attacked}(\alpha, S_A, o)$ from initial states in $[\![\phi_{pre}]\!]$ won't violate the postcondition. The value of $(1 - \delta)$ quantifies *the percentage of backward safety that is lost* due to the attack. The close δ is to 1, the more robust the system is.
- $\delta \leq 0$: then $r > 0$ and $r_1 \leq 0$. The system is unsafe due to the attack. Some initial states where the precondition holds can lead the system to unsafe states, if the system is under attack. The lower the value of δ, the more effective the attack can be. In the case of $r_1 = 0$ (i.e., $\delta = 0$), the compromised system can no longer be considered safe.

For example, consider again the cooling system example. Given $\phi_{pre} \equiv temp_p = 100$ and $\phi_{post} \equiv temp_p \leq 105$, we already know that $\textsc{B-safe}_r(\alpha, \phi_{pre}, \phi_{post})$, where α models the original system, for $r = 5.0$. Consider a sensor attack that offset 0.3 degree of sensor readings, formula $[\textsc{attacked}(\alpha, S_A, o)]\phi_{post}$ is $temp_p \leq 105.0$. So we know $\textsc{B-safe}_{r_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ for $r_1 = 5.0$. Therefore, the degree of backward robustness of the original system w.r.t. the attack is: $\delta = 5.0/5.0 = 1$. Meaning the attack doesn't affect the backward safety of the system.

5 Reasoning about Quantitative Robustness

Using Definition 5 (and 6), we can compute forward (and backward) robustness of a system in terms of the forward (and backward) safety of the system, before and after a bounded sensor attack. However, the computation of forward and backward safety may be difficult, as they consider all admissible values to compute the infimum. This is particularly difficult for a system with compromised sensors, due to the complications caused by the offset function.

In this section, we introduce two *simulation distances* between hybrid programs, called *forward simulation distance* (or *forward distance*) and *backward simulation distance* (or *backward distance*). They quantify the behavioral distance between the original system and the compromised one, according to a forward and backward flavor, respectively. These distances allow us to compute an upper bound of the loss of forward (and backward) safety. The computed upper bounds are not necessarily tight bounds, but they are easier to reason with and can be verified with existing tools.

To define forward (and backward) simulation distance between two programs, we extend the notion of distance between states, i.e., $\rho(\omega, \nu)$, to support computing distance on a set $\mathcal{H}$ of variables [45]. Intuitively, variables in $\mathcal{H}$ are the ones that are relevant to the specified precondition and postcondition. And thus computing distance over these variables give us the quantitative distance of interest. Consider the cooling system example, we are interested in the behavioral distance between the original program and the compromised one w.r.t. the variable $temp_p$, rather than $temp_s$.

We introduce a new notion of distance between states w.r.t. a set $\mathcal{H}$ of variables, as follows:

Definition 7. *For a set of variables $\mathcal{H} \subseteq \mathbb{V}$, two states ω and ν are at $\mathcal{H}$-distance d, written $\rho_{\mathcal{H}}(\omega, \nu) = d$, if $\sqrt{\sum_{x \in \mathcal{H}} (\omega(x) - \nu(x))^2} = d$. We write $\mathbf{Dist}_{\mathcal{H}}(\omega, \mathcal{S})$ to denote $\mathbf{Dist}(\omega, \mathcal{S})$ where $\rho_{\mathcal{H}}(\omega, \nu)$ is used instead of $\rho(\omega, \nu)$. $\mathbf{depth}_{\mathcal{H}}(\omega, \mathcal{S})$ is defined in the same manner.*

The following proposition shows that computing the forward and backward safety (using $\rho(\omega, \nu)$) can be reduced to a computation using $\rho_{\mathcal{H}}(\omega, \nu)$ with the appropriate variable sets $\mathcal{H}$, i.e., $\textsc{Var}(\phi_{pre})$ or $\textsc{Var}(\phi_{post})$.

Proposition 3. *For $u, u_1, u_2 \in \mathbb{R}$, and formula ϕ, ψ, if*

- $u = \mathbf{inf}\{\mathbf{Dist}(\omega, [\![\phi]\!]) \mid \omega \in [\![\psi]\!]\}$
- $u_1 = \mathbf{inf}\{\mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!]) \mid \omega \in [\![\psi]\!]\}$
- $u_2 = \mathbf{inf}\{\mathbf{Dist}_{\textsc{Var}(\psi)}(\omega, [\![\phi]\!]) \mid \omega \in [\![\psi]\!]\}$

then $u = u_1 = u_2$.

Intuitively, the proposition holds because the infimum value of u is essentially decided by the distance calculated w.r.t. the relevant variables in ϕ or ψ. The detailed proof can be found in Appendix B.

Forward Simulation Distance. We introduce the notion of *forward simulation distance*. Intuitively, programs β and α are in forward simulation at distance d if given the same initial condition, α can mimic the behaviors of β, i.e., α is able to reach states whose distance with those reached by β is at most d.

Definition 8 (Forward simulation distance). *For hybrid programs β, α, formula ϕ_{pre} and a set of variables $\mathcal{H}$, β and α are at forward simulation distance d w.r.t. ϕ_{pre} and $\mathcal{H}$, written $\beta \sqsubseteq^{\mathrm{F}}_{\phi_{pre},\mathcal{H},d} \alpha$, if for each state $\nu_1 \in [\![\phi_{pre}\langle\beta\rangle]\!]$ there exists a state $\nu_2 \in [\![\phi_{pre}\langle\alpha\rangle]\!]$ such that $\rho_{\mathcal{H}}(\nu_1, \nu_2) \leq d$.*

Here, for programs α and $\textsc{attacked}(\alpha, S_A, o)$, the forward simulation $\textsc{attacked}(\alpha, S_A, o) \sqsubseteq^{\mathrm{F}}_{\phi_{pre},\mathcal{H},d} \alpha$ expresses that for each state ν_1 reachable by $\textsc{attacked}(\alpha, S_A, o)$, from some initial states in $[\![\phi_{pre}]\!]$, there is a state ν_2 reachable by α, from some initial state in $[\![\phi_{pre}]\!]$, such that ν_1 and ν_2 are at distance at most d, for a fixed variable set $\mathcal{H}$. The distance d gives an upper bound on the perturbation introduced by the attack on the safety of the behaviors originating from $[\![\phi_{pre}]\!]$. The set $\mathcal{H}$ here often refers to variables that are relevant to the system's postcondition, i.e., $\textsc{Var}(\phi_{post})$.

For example, let α be the program modeling the cooling system shown in Figure 4 and $\textsc{attacked}(\alpha, S_A, o)$ the attacked version shown in Figure 5. Let $\mathcal{H}$ be $\textsc{Var}(\phi_{post}) = \{temp_p\}$. The forward distance between $\textsc{attacked}(\alpha, S_A, o)$ and α w.r.t. ϕ_{pre} and $\mathcal{H}$ is 0.3, which we will show by the proof method in the next section. From the existing example shown after Definition 5, we know 0.3 is indeed an upper bound of the loss of forward safety.

The following theorem states that the forward simulation distance d between $\textsc{attacked}(\alpha, S_A, o)$ and α w.r.t. $\textsc{Var}(\phi_{post})$, is indeed an *upper bound* to the loss of forward safety due to the attack.

Theorem 3. *For a hybrid program α, a set of variables $S_A \subseteq \textsc{Var}(\alpha)$, formulas ϕ_{pre} and ϕ_{post}, an offset function o, and $d, u \in \mathbb{R}$, if*

- $\textsc{F-safe}_u(\alpha, \phi_{pre}, \phi_{post})$*, with* $u > 0$
- $\textsc{attacked}(\alpha, S_A, o) \sqsubseteq^{\mathrm{F}}_{\phi_{pre},\textsc{Var}(\phi_{post}),d} \alpha$

then $\textsc{F-robust}(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$, for some δ such that $\delta \geq (u - d)/u$.

The theorem says that d is an upper bound of the loss of forward safety, meaning that $\textsc{F-safe}_{u_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$, for some u_1 such that $d \geq u - u_1$.

Proof. Let $\mathcal{H}$ be the set of variables $\textsc{Var}(\phi_{post})$. We need to prove $\textsc{F-safe}_{u_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ for $u_1 \geq u - d$. By definition, we have $\textsc{F-safe}_{u_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ if $u_1 = \mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o)\rangle]\!]\}$. Consider an arbitrary state $\nu \in [\![\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o)\rangle]\!]$. By the hypothesis $\textsc{attacked}(\alpha, S_A, o) \sqsubseteq^{\mathrm{F}}_{\phi_{pre},\mathcal{H},d} \alpha$, we infer that there is some state $\nu' \in [\![\phi_{pre}\langle\alpha\rangle]\!]$ with $\rho_{\mathcal{H}}(\nu, \nu') \leq d$. The hypothesis $\textsc{F-safe}_u(\alpha, \phi_{pre}, \phi_{post})$ coincides, by definition, with property $u = \mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\alpha\rangle]\!]\}$. By Proposition 3, we infer $u = \mathbf{inf}\{\mathbf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!]) \mid \nu \in [\![\phi_{pre}\langle\alpha\rangle]\!]\}$. Since $\nu' \in [\![\phi_{pre}\langle\alpha\rangle]\!]$, we infer $\mathbf{Dist}_{\mathcal{H}}(\nu', [\![\phi_{post}]\!]) \geq u$. Since $\rho_{\mathcal{H}}(\_, \_)$ is a metric, it is symmetric, thus implying $\rho_{\mathcal{H}}(\nu, \nu') = \rho_{\mathcal{H}}(\nu', \nu)$, and satisfies the triangular property. By the triangular property we infer that for any $\nu'' \notin [\![\phi_{post}]\!]$,

$$\rho_{\mathcal{H}}(\nu', \nu'') \leq \rho_{\mathcal{H}}(\nu', \nu) + \rho_{\mathcal{H}}(\nu, \nu'').$$

By definition of $\mathbf{Dist}_{\mathcal{H}}(\nu', [\![\phi_{post}]\!])$ we know $\rho_{\mathcal{H}}(\nu', \nu'') \geq u$, and since $\rho_{\mathcal{H}}(\nu, \nu') \leq d$, then $\rho_{\mathcal{H}}(\nu, \nu'') \geq u - d$. By definition of $\mathbf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!])$ and the arbitrarity of ν, we infer

$$\mathbf{inf}\{\mathbf{Dist}_{\mathcal{H}}(\nu, [\![\phi_{post}]\!]) | \nu \in [\![\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o)\rangle]\!]\} \geq u - d.$$

By Proposition 3 we infer:

$$\mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) | \nu \in [\![\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o)\rangle]\!]\} \geq u - d.$$

This completes the proof. □

Note that Theorem 3 may also hold for supersets of $\textsc{Var}(\phi_{post})$, e.g., it holds for $\textsc{Var}(\phi_{pre}) \cup \textsc{Var}(\phi_{post})$. However, the forward simulation distance d w.r.t. a superset is no smaller than the value of d for $\textsc{Var}(\phi_{post})$, since $\rho_{\mathcal{H}}(\omega, \nu)$ increases when more variables are involved. A larger d would give us a loose bound of safety loss.

Backward Simulation Distance. Symmetrically, we introduce *backward simulation distance* to reason with upper bounds of loss of backward safety caused by sensor attacks. Intuitively, programs β and α are in backward simulation distance d if for the same postcondition, α can mimic the behaviors of β that may violate the postcondition. This means that initial states that can lead to violation of safety condition of the two systems are distant at most d.

Definition 9 (Backward simulation distance). *For hybrid programs β and α, formula ϕ_{post} and a set of variables $\mathcal{H}$, β and α are at backward simulation distance d w.r.t. ϕ_{post} and $\mathcal{H}$, formally written as $\beta \sqsubseteq^{\mathrm{B}}_{\phi_{post},\mathcal{H},d} \alpha$, if for each state $\omega_1 \in [\![\langle\beta\rangle\neg\phi_{post}]\!]$ there exists a state $\omega_2 \in [\![\langle\alpha\rangle\neg\phi_{post}]\!]$ such that $\rho_{\mathcal{H}}(\omega_1, \omega_2) \leq d$.*

Here, $\textsc{attacked}(\alpha, S_A, o) \sqsubseteq^{\mathrm{B}}_{\phi_{post},\mathcal{H},d} \alpha$ means that for each initial state ω_1, from which $\textsc{attacked}(\alpha, S_A, o)$ can reach a unsafe state in $[\![\neg\phi_{post}]\!]$, there is an initial state ω_2, from which α can reach a state in $[\![\neg\phi_{post}]\!]$, such that ω_1 and ω_2 are at distance at most d, w.r.t. a set of variables $\mathcal{H}$. Thus, the backward distance between the original and the compromised system returns an upper bound on the admissible perturbations introduced by a sensor attack on the initial states leading to possible violations of safety, fixed a desired postcondition ϕ_{post}. The set $\mathcal{H}$ often is the set of variables that are relevant to the system's precondition, i.e., $\textsc{Var}(\phi_{pre})$.

For example, let α be the program modeling the cooling system shown in Figure 4. We consider a sensor attack that introduces an offset 0.3 to the temperature sensor. Let $\mathcal{H}$ be $\text{VAR}(\phi_{pre}) = \{temp_p\}$. The backward distance between $\text{ATTACKED}(\alpha, S_A, o)$ and α w.r.t. ϕ_{post} and $\mathcal{H}$ is 0, which we will show by the proof method in the next section. From the existing examples, we know 0 is indeed an upper bound of the loss of backward safety.

The following theorem states that the backward simulation distance d between $\text{ATTACKED}(\alpha, S_A, o)$ and α w.r.t. variable set $\text{VAR}(\phi_{pre})$, is indeed an *upper bound* to the loss of backward safety due to the attack.

Theorem 4. *For a hybrid program α, a set of variables $S_A \subseteq \text{VAR}(\alpha)$, formulas ϕ_{pre} and ϕ_{post}, an offset function o and $d, r \in \mathbb{R}$, if*

- *B-SAFE$_r(\alpha, \phi_{pre}, \phi_{post})$, with $r > 0$*
- *$\text{ATTACKED}(\alpha, S_A, o) \sqsubseteq^{B}_{\phi_{post}, \text{VAR}(\phi_{pre}), d} \alpha$*

then B-ROBUST$(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$ for some δ such that $\delta \geq (r - d)/r$.

The theorem says that d is an upper bound of the loss of backward safety, meaning B-SAFE$_{r_1}(\text{ATTACKED}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ for some r_1 such that $d \geq r - r_1$. The theorem can be similarly proven as Theorem 3. The detailed proof can be found in Appendix B.

6 Proving Simulation Distances

This section shows that simulation distance $\sqsubseteq^{F}_{\phi_{pre}, \mathcal{H}, d}$ and $\sqsubseteq^{B}_{\phi_{pre}, \mathcal{H}, d}$ can be expressed as a d$\mathcal{L}$ formula, thus existing tools, such as KeYmaera X [15], can be used to check whether the relation holds between a given program and its attacked version.

6.1 Encoding Simulation Distances with Formulas

The forward and backward simulation distance are defined upon distance between states that respectively satisfy two formulas. For example, the forward distance is computed on states satisfying, respectively, $\phi_{pre}\langle\beta\rangle$ and $\phi_{pre}\langle\alpha\rangle$. Moreover, both distances are formalized in a "forall exists" manner. Therefore, a direct way to verify them, is to compute the relevant two formulas, and then verify the distance between states that satisfy the two formulas.

Based on this insight, the following formula can be instantiated with different formulas to verify both simulation distances:

$$(\phi \wedge (\overline{y} = \overline{x})) \rightarrow \exists \overline{x}.\ (\psi \wedge (\rho_{\mathcal{H}}(\overline{y}, \overline{x}) \leq d))$$

where ϕ and ψ are formulas specifying, respectively, conditions of the compromised system and the original system. They share the same set of variables. Here $\overline{x}$ are variables used by ϕ and ψ, and $\overline{y}$ are a list of *fresh* variables whose dimension is the same as $\overline{x}$. Variables in $\overline{y}$ are (implicitly) universally quantified. The fresh variables are used to store values of $\overline{x}$ that satisfy the first formula. The notation $\rho_{\mathcal{H}}(\overline{y}, \overline{x})$ computes the distance between two vectors of variables w.r.t. the set $\mathcal{H}$: $\sqrt{\sum_{\overline{x}(i) \in \mathcal{H}} (\overline{x}(i) - \overline{y}(i))^2}$, where $\overline{x}(i)$ and $\overline{y}(i)$ represent, respectively, the ith element in vector $\overline{x}$ and $\overline{y}$.

The encoding can be used to verify forward distance by letting ϕ and ψ, respectively, be $\phi_{pre}\langle\text{ATTACKED}(\alpha, S_A, o)\rangle$ and $\phi_{pre}\langle\alpha\rangle$.

Consider the cooling system example, we know $\phi_{pre}\langle\alpha\rangle$ and $\phi_{pre}\langle\text{ATTACKED}(\alpha, S_A, o)\rangle$ are, respectively, $99.2 < temp_p \leq 101.3$ and $99.5 < temp_p \leq 101$. We can express that $\text{ATTACKED}(\alpha, S_A, o)$ and α are at forward distance 0.3 w.r.t. ϕ_{pre} and $\mathcal{H} = \text{VAR}(\phi_{pre}) = \{temp_p\}$ with the following formula:

$$(99.2 < temp_p \leq 101.3 \ \wedge\ fv_p = temp_p) \rightarrow$$
$$(\exists temp_p.\ 99.5 < temp_p \leq 101 \ \wedge\ (\sqrt{(temp_p - fv_p)^2} \leq 0.3))$$

The encoding can also be instantiated for verifying backward simulation distance by letting ϕ and ψ be $\langle\text{ATTACKED}(\alpha, S_A, o)\rangle\neg\phi_{post}$ and $\langle\alpha\rangle\neg\phi_{post}$, respectively.

For the cooling system example, we know that $\langle\alpha\rangle\neg\phi_{post}$ and $\langle\text{ATTACKED}(\alpha, S_A, o)\rangle\neg\phi_{post}$ are both $temp_p > 105$. We can express $\text{ATTACKED}(\alpha, S_A, o)$ and α are at backward simulation distance 0 w.r.t. ϕ_{post} and $\mathcal{H} = \{temp_p\}$, with the following formula:

$$(temp_p > 105.0 \wedge fv_p = temp_p) \ \rightarrow$$
$$(\exists temp_p.\ temp_p > 105.0 \wedge \sqrt{(temp_p - fv_p)^2} \leq 0)$$

Both formulas can be easily verified with KeYmaera X.

Using this encoding requires computing the strongest postcondition and weakest precondition, which may be difficult, especially for systems with complex dynamics. To alleviate the problem, we can over-approximate the conditions for the compromised systems and under-approximate the conditions for the original systems. This would allows us to compute an upper bound of the loss of safety. And we would still be able to compare different system designs with such a bound. The quality of the upper bound depends on how good the approximations are.

6.2 Encoding Simulation Distance with Modalities

An alternative way to encode the two simulation distances is through modalities, which directly express program executions.

For forward distance, i.e., $\text{ATTACKED}(\alpha, S_A, o) \sqsubseteq^{F}_{\phi_{pre}, \mathcal{H}, d} \alpha$, we can express it as the following d$\mathcal{L}$ formula:

$$(\phi_{pre} \wedge \langle\text{ATTACKED}(\alpha, S_A, o)\rangle(\overline{y} = \overline{x})) \rightarrow$$
$$(\exists \overline{x}.\ \phi_{pre} \wedge \langle\alpha\rangle(\rho_{\mathcal{H}}(\overline{y}, \overline{x}) \leq d))$$

The first line encodes "for each state that can be reached from precondition ϕ_{pre} after an execution of the compromised program". The fresh variables of $\overline{y}$ are used to record the reachable states. The second line encodes "there is an execution of the original program under precondition ϕ_{pre} such that the distance between the corresponding final states is bound by d."

Similarly, we can use the following d$\mathcal{L}$ formula to express the backward simulation distance $\text{ATTACKED}(\alpha, S_A, o) \sqsubseteq^{B}_{\phi_{post}, \mathcal{H}, d} \alpha$:

$$((\overline{y} = \overline{x}) \ \wedge\ \langle\text{ATTACKED}(\alpha, S_A, o)\rangle\neg\phi_{post}) \rightarrow$$
$$(\exists \overline{x}.\ (\rho_{\mathcal{H}}(\overline{x}, \overline{y}) \leq d) \ \wedge\ \langle\alpha\rangle\neg\phi_{post})$$

The first line encodes "for each initial state that can lead the compromised system to unsafe states". The fresh variables of $\overline{y}$ are used to record the initial states. The second line encodes "there is an initial state that can lead the original program to unsafe states such that the distance between the two initial states is bound by d."

Verifying the modality-based encodings could be a nontrivial task. Such a "forall, exists" relational property is difficult to verify in general. Existing work have introduced some approaches that tackle similar problems using self-composition [45]. Exploring efficient ways to verify these encodings is an interesting future work.

7 Case Study

In this section, we showcase the concepts and techniques introduced in this work with a case study. Consider an autonomous

$$
\begin{aligned}
&(\textit{System Constants} : A = 1, B = 1, \epsilon = 1)\\
\phi_{pre} &\equiv (2Bd_p > v_p^2) \wedge v_p \geq 0\\
\phi_{post} &\equiv d_p > 0\\
\psi &\equiv 2Bd_s > v_s^2 + (A+B)(A\epsilon^2 + 2v_s\epsilon)\\
accel &\equiv ?\psi\, ;\, a := A\\
brake &\equiv a := -B\\
ctrl &\equiv d_s := d_p\, ;\, v_s := v_p\, ;\, (accel \cup brake)\\
plant &\equiv d_p{}' = -v_p, v_p{}' = a, t' = 1\,\&(v_p \geq 0 \wedge t \leq \epsilon)\\
\phi_{safety} &\equiv \phi_{pre} \rightarrow [(ctrl\, ;\, plant)^*]\phi_{post}
\end{aligned}
$$

Figure 6. d$\mathcal{L}$ model of an autonomous vehicle with sensing

vehicle that needs to stop before hitting an obstacle. [1] For simplicity, we model the vehicle in just one dimension. Figure 6 shows a d$\mathcal{L}$ model of such an autonomous vehicle with sensing. Let d_p and d_s, respectively, be the vehicle's physical and sensed distance from the obstacle. The safety condition that we would like to enforce (ϕ_{post}) is that d_p is positive. Let v_p be the vehicle's velocity towards the obstacle in meters per second (m/s) and v_s be its sensed value. Let a be the vehicle's acceleration (m/s^2). Let t be the time elapsed since the controller was last invoked. The hybrid program $plant$ describes how the physical environment evolves over time interval ϵ: distance changes according to $-v_p$ (i.e., $d_p{}' = -v_p$), velocity changes according to the acceleration (i.e., $v_p{}' = a$), and time passes at a constant rate (i.e., $t' = 1$). The differential equations evolve only within the time interval $t \leq \epsilon$ and if v_p is non-negative (i.e., $v_p \geq 0$).

Program $ctrl$ models the vehicle's controller. The vehicle can either accelerate at A m/s^2 or brake at $-B$ m/s^2. For the purposes of the model, the controller chooses nondeterministically between these options. Hybrid programs $accel$ and $brake$ express the controller accelerating or braking (i.e., setting a to A or $-B$ respectively). The controller can accelerate only if condition ψ is true, which captures that the vehicle can accelerate for the next ϵ seconds only if doing so would still allow it to brake in time to avoid the obstacle.

For the quantitative analysis of this model, we treat symbolic variables A, B, ϵ as the parameters of the system and set them as constants: $A = 1$, $B = 1$, and $\epsilon = 1$. In addition, in this case study, we verify the forward and backward distance using the d$\mathcal{L}$ encodings with formulas, after computing the relevant weakest preconditions and strongest postconditions using these constants.

Bounded Sensor Attack. The formula ϕ_{safety} specifies the desired (Boolean) safety property: given an appropriate precondition ϕ_{pre}, the safety condition ϕ_{post} holds after any execution of the system. The safety property indeed holds. Also, by definition, the system satisfies F-SAFE$_0(\alpha, \phi_{pre}, \phi_{post})$ and B-SAFE$_0(\alpha, \phi_{pre}, \phi_{post})$, where $\alpha = (ctrl\, ;\, plant)^*$. However, the system's safety has no room for sensing errors. Any sensor attacks that offset the readings can compromise the safety.

Consider a bounded sensor attack on the velocity sensor that deviate the readings of v_s from v_p up to 1 m/s. We can model it by replacing $v_s := v_p$ with $v_s := *; ?v_s \leq v_p + 1 \wedge v_s \geq v_p - 1$ in Figure 6. The system is not robust against this attack, i.e., the safety property no longer holds when the sensor is compromised.

[1] Platzer introduces this autonomous vehicle example [38].

A Safer Controller. Now, consider a different controller $ctrl'$ whose condition for acceleration is designed to tolerate the inaccuracy of sensed velocity at a maneuver of 2 m/s, then the system can then be modeled as follows:

$$
\begin{aligned}
\psi' &\equiv 2Bd_s > (v_s+2)^2 + (A+B)(A\epsilon^2 + 2(v_s+2)\epsilon)\\
ctrl' &\equiv d_s := d_p\, ;\, v_s := v_p\, ;\, ((?\psi'\, ;\, a := A) \cup a := -B)\\
&\dots
\end{aligned}
$$

Let β denote the new system, i.e., $\beta = (ctrl'\, ;\, plant)^*$. It still holds that F-SAFE$_0(\beta, \phi_{pre}, \phi_{post})$ and B-SAFE$_0(\beta, \phi_{pre}, \phi_{post})$.

Consider a different precondition:

$$\phi'_{pre} \equiv (2Bd_p > (v_p+2)^2) \wedge v_p \geq 0$$

Executing β given precondition ϕ'_{pre}, we get a strongest postcondition $(2d_p > (v_p+2)^2) \wedge v_p \geq 0$. So β is forward safe for a degree of 2 w.r.t. ϕ_{post}, i.e., F-SAFE$_2(\beta, \phi'_{pre}, \phi_{post})$.

Forward Simulation Distance. We can prove that program ATTACKED(β, S_A, o) and β are at forward distance 1.5 w.r.t. ϕ'_{pre} and $\mathcal{H}$ = VAR$(\phi_{post}) = \{d_p\}$. Here, $\phi'_{pre}\langle$ATTACKED$(\beta, S_A, o)\rangle$ is $(2Bd_p > (v_p+1)^2) \wedge v_p \geq 0$. Then the forward simulation distance can be expressed as:

$$
\begin{aligned}
&2d_p > (v_p+1)^2 \wedge v_p \geq 0 \wedge d_p = fd_p \rightarrow\\
&\quad \exists d_p\, v_p.((2d_p > (v_p+2)^2) \wedge v_p \geq 0 \wedge \sqrt{(d_p - fd_p)^2} \leq 1.5))
\end{aligned}
$$

Here, fd_p is a fresh variable. KeYmaera X can easily verify this formula. So, ATTACKED$(\beta, S_A, o) \sqsubseteq^{F}_{\phi'_{pre}, \{d_p\}, 1.5} \beta$, being 1.5 the upper bound of the loss of forward safety. Since F-SAFE$_2(\beta, \phi'_{pre}, \phi_{post})$, by Theorem 3 it follows:

$$\text{F-ROBUST}(\beta, \phi'_{pre}, \phi_{post}, S_A, o, \delta)$$

for some $\delta \geq \frac{0.5}{2} = 0.25$. So the system is still safe under the attack, and the percentage of forward safety loss is at most 75%.

Backward Simulation Distance. We already know it holds that B-SAFE$_0(\beta, \phi_{pre}, \phi_{post})$, so there is not much we can learn from backward simulation distance here.

Now consider the backward safety of β w.r.t. ϕ'_{pre} and a different postcondition $\phi'_{post} \equiv d_p > 0.5$. We can compute that $\langle\beta\rangle\neg\phi'_{post}$ is $d_p <= 0.5 \vee (2(d_p - 0.5) <= v_p^2 \wedge v_p >= 0)$, and further compute B-SAFE$_{\sqrt{2}}(\beta, \phi'_{pre}, \phi'_{post})$. Moreover, $\langle$ATTACKED$(\beta, S_A, o)\rangle\neg\phi'_{post}$ is $d_p <= 0.5 \vee (2d_p <= (v_p+1)^2 \wedge v_p >= 0)$. We can express that program ATTACKED(β, S_A, o) and β are at backward distance 1 w.r.t. ϕ'_{post} and VAR(ϕ'_{pre}):

$$
\begin{aligned}
&((d_p <= 0.5 \vee (2d_p <= (v_p+1)^2 \wedge v_p >= 0))\\
&\qquad \wedge fd_p = d_p \wedge fv_p = v_p) \rightarrow\\
&\quad \exists d_p\, v_p.(d_p <= 0.5 \vee (2(d_p - 0.5) <= v_p^2 \wedge v_p >= 0)\\
&\qquad \wedge \sqrt{(d_p - fd_p)^2 + (v_p - fv_p)^2} \leq 1)
\end{aligned}
$$

Again, the formula can be verified by KeYmaera X. Then by Theorem 4 and B-SAFE$_{\sqrt{2}}(\beta, \phi'_{pre}, \phi'_{post})$ it follows:

$$\text{B-ROBUST}(\beta, \phi'_{pre}, \phi'_{post}, S_A, o, \delta)$$

for some $\delta \geq \frac{\sqrt{2}-1}{\sqrt{2}}$. So the system is still backward safe under the attack, and the percentage of loss of backward safety due to the attack is at most $1/\sqrt{2} \approx 71\%$.

8 Related Work

Robustness of CPSs. Our work is a quantitative generalization of Xiang et at. [45], in the setting of hybrid programs and d$\mathcal{L}$. In that paper, the authors propose two notions of robustness for CPSs: *robustness of safety*, when (unbound) sensor attacks are unable to affect the system under attack, and *robustness of high-integrity state*, when high-integrity parts of the system cannot be compromised. In the current paper, we generalize the first of the two relations.

Fränzle et al. [14] classify the notions of robustness for CPSs as follows: (i) input/output robustness; (ii) robustness with respect to system parameters; (iii) robustness in real-time system implementation; (iv) robustness due to unpredictable environment; (v) robustness to faults. The notion of robustness considered in this paper falls in category (iv), where the attacks are the source of environment's unpredictability. Other works study robustness properties for CPSs [19, 20, 40, 43]. Some of them focus on robustness against attacks [19, 20], even adopting quantitative reasonings [40, 43].

Our notion of forward robustness shares similarities with some existing notions of robustness, such as invariance [3] and input-to-state stability [1]. These notions concern if a system stays in a safe region when small changes happen to initial conditions, while forward robustness concerns if a system stays in a safe region when under attack. Although it might be possible to reformulate existing notions of robustness to characterize our forward robustness, our formulation focuses on modeling attacks which makes it easier to analyze their impact.

Signal Temporal Logic (STL) [31] is a specification formalism for expressing real-time temporal *safety* and performance properties, such as *robustness*, of CPSs. Ferrère et al. [12] study a quantitative extension of STL that classifies signals as inputs and outputs to specify the system-under-test as an input/output relation instead of a set of correct execution traces. The idea behind their approach is quite similar to that followed in our forward robustness, as they express families of admissible patterns of both the model inputs and the model preconditions that guarantee the desired behavior of the model output. Mohammadinejad et al. [32] adopt a dual approach, similar to that followed in our backward robustness. Given an output requirement they propose an algorithm to mine an environment assumption, consisting of a large subset of input signals for which the corresponding output signals satisfy the output requirement.

Formal Analysis of Sensor Attacks. Lanotte et al. [25, 26, 28] propose process-calculus approaches to model and analyze the impact of physics-based attacks, as sensor attacks in CPSs. Their threat models consider attacks that may manipulate both sensor readings and control commands. Their model of physics is discrete and they focus on crucial timing aspects of attacks, such as beginning and duration. Bernardeschi et al. [5] introduce a framework to analyze the effects of attacks on sensors and actuators. Controllers of systems are specified using the formalism PVS [35]. The physics is described by other modeling tools. Their threat model is similar to ours: the effect of an attack is a set of assignments to the variables defined in the controller. Simulation is used to analyze the effects of attacks. Huang et al. [20] proposed a risk assessment method that uses a Bayesian network to model the attack propagation process and infers the probabilities of sensors and actuators to be compromised. These probabilities are fed into a stochastic hybrid system model to predict the evolution of the physical process. Then, the security risk is quantified by evaluating the system availability with the model.

9 Conclusion

A formal framework for *quantitative* analysis of bounded sensor attacks on CPSs is introduced. Given a precondition ϕ_{pre} and postcondition ϕ_{post} of a system α, we formalize two safety notions, *quantitative forward safety*, F-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$, and *quantitative backward safety*, B-SAFE$_u(\alpha, \phi_{pre}, \phi_{post})$, where $u \in \mathbb{R}$ respectively express: (1) how strong the strongest postcondition $\phi_{pre}\langle\alpha\rangle$ is with respect to the postcondition ϕ_{post}, and (2) how strong the precondition ϕ_{pre} is with respect to the weakest precondition $[\alpha]\phi_{post}$. The bigger u is, the safer the system is. On the contrary, if u is negative, then some reachable states violate the safety condition ϕ_{post}. If u is 0, then the system cannot be considered safe. We introduce *forward and backward robustness*, F-ROBUST$(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$ and B-ROBUST$(\alpha, \phi_{pre}, \phi_{post}, S_A, o, \delta)$ respectively, to quantify the robustness δ, with $\delta \leq 1$, for a system α against *bounded sensor attacks*, as the ratio between the safety of the attacked system and the degree of safety of the original system; here, the value of $(1 - \delta)$ quantifies the *percentage of safety that is lost* due to the attack. The closer δ is to 1, the more robust the system is. To reason about the notions of robustness, we introduce two simulation distances, *forward and backward simulation distance*, defined based on the behavior distances between the original system and the compromised system, to characterize upper bounds of the degree of forward and backward safety loss caused by the sensor attacks. We verify the two simulations by expressing them as d$\mathcal{L}$ formulas. A case study on autonomous vehicle is presented.

Applicability. The proposed approach can be applied to systems where we can compute (or over-approximate) strongest postcondition and weakest precondition. As mentioned, the computation can be done with existing work and benefits from future advances in the verification of CPSs, e.g., complex dynamics. Therefore, though the examples used in the paper are not very complex, we expect that the proposed approach can be used on complex systems.

Future work. As observed in [24, 26, 28], *timing* is a critical issue when attacking CPSs. We aim at generalizing our threat model to deal with more sophisticated time-sensitive sensor attacks, where the attacker may specify (possibly periodic) attack windows in which offsets might be potentially different in each window, depending on the system state. This might be necessary to implement stealthy attacks working around adaptive IDSs.

Modality-based encoding might be a more generic approach for reasoning with simulation distances, but such an encoding is often difficult to verify. A potential future work is to develop a proof system for verifying such an encoding, e.g., a relational logic to reason about the upper bound of behavior distance between two programs. We expect that such a logic would greatly help proof automation and let us reason about systems where the computation of the strongest postcondition and weakest precondition is difficult.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback on this paper. This material is based upon work in part supported by Intel. Massimo Merro has been partially supported by the project "Dipartimenti di Eccellenza 2018-2022", funded by the Italian Ministry of Universities and Research (MUR).

References

[1] Andrei A Agrachev, A Stephen Morse, Eduardo D Sontag, Héctor J Sussmann, Vadim I Utkin, and Eduardo D Sontag. 2008. Input to state stability: Basic concepts and results. *Nonlinear and optimal control theory: lectures given at the CIME summer school held in Cetraro, Italy June 19–29, 2004* (2008), 163–220.

[2] Rajeev Alur. 2015. *Principles of cyber-physical systems.* MIT Press.

[3] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. 2016. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. Automat. Control* 62, 8 (2016), 3861–3876.

[4] Martín Barrère, Chris Hankin, Nicolas Nicolaou, Demetrios G. Eliades, and Thomas Parisini. 2020. Measuring cyber-physical security in industrial control systems via minimum-effort attack strategies. *J. Inf. Secur. Appl.* 52 (2020), 102471.

[5] Cinzia Bernardeschi, Andrea Domenici, and Maurizio Palmieri. 2020. Formalization and co-simulation of attacks on cyber-physical systems. *Journal of Computer Virology and Hacking Techniques* 16, 1 (2020), 63–77.

[6] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. 2004. *Convex optimization.* Cambridge university press.

[7] Davide Bresolin, Pieter Collins, Luca Geretti, Roberto Segala, Tiziano Villa, and Sanja Zivanovic Gonzalez. 2020. A computable and compositional semantics for hybrid automata. In *HSCC*. ACM, 18:1–18:11.

[8] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. 2019. Adversarial sensor attack on LiDAR-based perception in autonomous driving. In *CCS*. 2267–2281.

[9] Drew Davidson, Hao Wu, Rob Jellinek, Vikas Singh, and Thomas Ristenpart. 2016. Controlling UAVs with sensor input spoofing attacks. In *WOOT*.

[10] Alexandre Donzé and Oded Maler. 2010. Robust Satisfaction of Temporal Logic over Real-Valued Signals. In *FORMATS (LNCS, Vol. 6246)*. 92–106.

[11] Georgios E Fainekos and George J Pappas. 2009. Robustness of temporal logic specifications for continuous-time signals. *Theoretical Computer Science* 410, 42 (2009), 4262–4291.

[12] Thomas Ferrère, Dejan Nickovic, Alexandre Donzé, Hisahiro Ito, and James Kapinski. 2019. Interface-aware signal temporal logic. In *HSCC*. ACM, 57–66.

[13] Simon Foster, Jonathan Julián Huerta y Munive, Mario Gleirscher, and Georg Struth. 2021. Hybrid Systems Verification with Isabelle/HOL: Simpler Syntax, Better Models, Faster Proofs. In *FM (LNCS, Vol. 13047)*. Springer, 367–386.

[14] Martin Fränzle, James Kapinski, and Pavithra Prabhakar. 2016. Robustness in Cyber-Physical Systems. *Dagstuhl Reports* 6, 9 (2016), 29–45.

[15] Nathan Fulton, Stefan Mitsch, Jan-David Quesel, Marcus Völp, and André Platzer. 2015. KeYmaera X: An axiomatic tactical theorem prover for hybrid systems. In *CADE (LNCS, Vol. 9195)*. Springer, 527–538.

[16] Alessandro Giacalone, Chi-Chang Jou, and Scott A. Smolka. 1990. Algebraic Reasoning for Probabilistic Concurrent Systems. In *Programming concepts and methods: Proceedings of the IFIP Working Group 2.2, 2.3 Working Conference on Programming Concepts and Methods, Sea of Galilee, Israel, 2-5 April, 1990*, Manfred Broy and Cliff B. Jones (Eds.). North-Holland, 443–458.

[17] J. Giraldo, D. I. Urbina, A. Cardenas, J. Valente, M. Faisal, J. Ruths, N. O. Tippenhauer, H. Sandberg, and R. Candell. 2018. A Survey of Physics-Based Attack Detection in Cyber-Physical Systems. *ACM Comput. Surv.* 51, 4 (2018), 76:1–76:36.

[18] David Harel, Dexter Kozen, and Jerzy Tiuryn. 2000. *Dynamic Logic.* MIT Press.

[19] Fei Hu, Yu Lu, Athanasios V. Vasilakos, Qi Hao, Rui Ma, Yogendra Patil, Ting Zhang, Jiang Lu, Xin Li, and Neal N. Xiong. 2016. Robust Cyber-Physical Systems: concept, models, and Implementation. *Future Gener. Comput. Syst.* 56 (2016), 449–475.

[20] K. Huang, C. Zhou, Y. Tian, S. Yang, and Y. Qin. 2018. Assessing the Physical Impact of Cyberattacks on Industrial Cyber-Physical Systems. *IEEE Trans. Industrial Electronics* 65, 10 (2018), 8153–8162.

[21] Jonathan Julián Huerta y Munive and Georg Struth. 2022. Predicate Transformer Semantics for Hybrid Systems. *Journal of Automated Reasoning* 66, 1 (2022), 93–139.

[22] I. Jahandideh, F. Ghassemi, and M. Sirjani. 2021. An actor-based framework for asynchronous event-based cyber-physical systems. *Software and Systems Modeling* 20 (2021), 641–665. Issue 3.

[23] Dexter Kozen. 1997. Kleene algebra with tests. *TOPLAS* 19, 3 (1997), 427–443.

[24] M. Krotofil and A. A. Cárdenas. 2013. Resilience of Process Control Systems to Cyber-Physical Attacks. In *NordSec (LNCS, Vol. 8208)*. Springer, 166–182.

[25] R. Lanotte, M. Merro, A. Munteanu, and S. Tini. 2021. Formal Impact Metrics for Cyber-physical Attacks. In *CSF*. IEEE, 1–16.

[26] Ruggero Lanotte, Massimo Merro, Andrei Munteanu, and Luca Viganò. 2020. A Formal Approach to Physics-based Attacks in Cyber-physical Systems. *ACM Transactions on Privacy and Security* 23, 1 (2020), 3:1–3:41.

[27] Ruggero Lanotte, Massimo Merro, Riccardo Muradore, and Luca Viganò. 2017. A formal approach to cyber-physical attacks. In *CSF*. IEEE, 436–450.

[28] R. Lanotte, M. Merro, and S. Tini. 2018. Towards a Formal Notion of Impact Metric for Cyber-Physical Attacks. In *IFM (LNCS, Vol. 11023)*. Springer, 296–315.

[29] Kim Guldstrand Larsen. 2009. Verification and performance analysis for embedded systems. In *TASE*. IEEE, 3–4.

[30] Edward Ashford Lee and Sanjit A Seshia. 2016. *Introduction to embedded systems: A cyber-physical systems approach.* MIT press.

[31] Oded Maler and Dejan Nickovic. 2004. Monitoring Temporal Properties of Continuous Signals. In *FORMATS/FTRTFT (LNCS, Vol. 3253)*. 152–166.

[32] Sara Mohammadinejad, Jyotirmoy V Deshmukh, and Aniruddh G Puranic. 2020. Mining environment assumptions for cyber-physical system models. In *ICCPS*. IEEE, 87–97.

[33] Vivek Nigam, Carolyn Talcott, and A.A. Urquiza. 2016. Towards the Automated Verification of Cyber-Physical Security Protocols: Bounding the Number of Timed Intruders. In *ESORICS (LNCS, Vol. 9879)*. Springer, 450–470.

[34] V. Nigam and C. L. Talcott. 2019. Formal Security Verification of Industry 4.0 Applications. In *ETFA*. IEEE, 1043–1050.

[35] Sam Owre, John M Rushby, and Natarajan Shankar. 1992. PVS: A prototype verification system. In *CADE (LNCS, Vol. 607)*. Springer, 748–752.

[36] André Platzer. 2008. Differential dynamic logic for hybrid systems. *Journal of Automated Reasoning* 41, 2 (2008), 143–189.

[37] André Platzer. 2017. A complete uniform substitution calculus for differential dynamic logic. *Journal of Automated Reasoning* 59, 2 (2017), 219–265.

[38] André Platzer. 2018. *Logical foundations of cyber-physical systems.* Vol. 662. Springer.

[39] M. Rocchetto and N. O. Tippenhauer. 2016. On attacker models and profiles for cyber-physical systems In *ESORICS (LNCS, Vol. 9879:427-449)*. Springer, 427–449.

[40] Matthias Rungger and Paulo Tabuada. 2016. A Notion of Robustness for Cyber-Physical Systems. *IEEE Trans. Autom. Control.* 61, 8 (2016), 2108–2123.

[41] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *USENIX Security*. 881–896.

[42] Paulo Tabuada. 2009. *Verification and control of hybrid systems: a symbolic approach.* Springer.

[43] Paulo Tabuada, Sina Yamac Caliskan, Matthias Rungger, and Rupak Majumdar. 2014. Towards Robustness for Cyber-Physical Systems. *IEEE Trans. Autom. Control.* 59, 12 (2014), 3151–3163.

[44] Ashish Tiwari. 2011. Logic in software, dynamical and biological systems. In *LICS*. IEEE, 9–10.

[45] Jian Xiang, Nathan Fulton, and Stephen Chong. 2021. Relational Analysis of Sensor Attacks on Cyber-Physical Systems. In *CSF*. IEEE, 1–16.

A Definitions

We present the definitions of bound variables, free variables, and variable sets for hybrid programs and d$\mathcal{L}$ formulas.

Definition 10 (Bound variables). *The set $BV(\phi)$ of bound variables of d$\mathcal{L}$ formula ϕ is defined inductively as:*

$$\begin{aligned} BV(\theta \sim \delta) &= \emptyset \sim \in \{<, \leq, =, >, \geq\} \\ BV(\neg\phi) &= BV(\phi) \\ BV(\phi \land \psi) &= BV(\phi) \cup BV(\psi) \\ BV(\forall x.\ \phi) &= \{x\} \cup BV(\phi) \\ BV([\alpha]\phi) &= BV(\alpha) \cup BV(\phi) \end{aligned}$$

The set $BV(\alpha)$ of bound variables of hybrid program α, i.e., those may potentially be written to, is defined inductively as:

$$\begin{aligned} BV(x := \theta) = BV(x := *) &= \{x\} \\ BV(?\phi) &= \emptyset \\ BV(x' = \theta \,\&\phi) &= \{x, x'\} \\ BV(\alpha\,;\,\beta) = BV(\alpha \cup \beta) &= BV(\alpha) \cup BV(\beta) \\ BV(\alpha^*) &= BV(\alpha) \end{aligned}$$

Definition 11 (Must-bound variables). *The set $MBV(\alpha) \subseteq BV(\alpha)$ of most bound variables of hybrid program α, i.e., all those that must be written to on all paths of α, is defined inductively as:*

$$\begin{aligned} MBV(x := \theta) = MBV(x := *) &= \{x\} \\ MBV(?\phi) &= \emptyset \\ MBV(x' = \theta \,\&\phi) &= \{x, x'\} \\ MBV(\alpha \cup \beta) &= MBV(\alpha) \cap MBV(\beta) \\ MBV(\alpha\,;\,\beta) &= MBV(\alpha) \cup MBV(\beta) \\ MBV(\alpha^*) &= \emptyset \end{aligned}$$

Definition 12 (Free variables). *The set $FV(\theta)$ of variables of term θ is defined as:*

$$FV(\theta) = \{\theta\}$$
$$FV(c) = \emptyset$$
$$FV(\theta \oplus \delta) = FV(\theta) \cup FV(\delta)$$

The set $FV(\phi)$ of free variables of $\mathrm{d}\mathcal{L}$ formula ϕ is defined as:

$$FV(\theta \sim \delta) = FV(\theta) \cup FV(\delta)$$
$$FV(\neg\phi) = FV(\phi)$$
$$FV(\phi \wedge \psi) = FV(\phi) \cup FV(\psi)$$
$$FV(\forall x.\ \phi) = FV(\phi) \setminus \{x\}$$
$$FV([\alpha]\phi) = FV(\alpha) \cup (FV(\phi) \setminus MBV(\alpha))$$

The set $FV(\alpha)$ of bound variables of hybrid program α is defined inductively as:

$$FV(x := \theta) = FV(\theta)$$
$$FV(x := *) = \emptyset$$
$$FV(?\phi) = FV(\phi)$$
$$FV(x' = \theta \,\&\phi) = \{x\} \cup FV(\theta) \cup FV(\phi)$$
$$FV(\alpha \cup \beta) = FV(\alpha) \cup FV(\beta)$$
$$FV(\alpha\, ;\, \beta) = FV(\alpha) \cup (FV(\beta) \setminus MBV(\alpha))$$
$$FV(\alpha^*) = FV(\alpha)$$

Definition 13 (Variable sets). *The set $\textsc{Var}(\alpha)$, variables of hybrid program α is $BV(\alpha) \cup FV(\alpha)$. The set $\textsc{Var}(\phi)$, variables of $\mathrm{d}\mathcal{L}$ formula ϕ is $BV(\phi) \cup FV(\phi)$.*

B Proofs

Proof of Theorem 1. We prove the inequality $u_1 \leq u$, the inequality $u_2 \leq u_1$ can be proved similarly. The behaviors of the system with compromised sensors subsume the behaviors of the original program, since the sensed values q_s can take the correct physical value q_p. Then we know $\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o_1)\rangle$ contains all states of $\phi_{pre}\langle\alpha\rangle$. Then, according to the definition of forward safety, $\mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) | \nu \in [\![\phi_{pre}\langle\alpha\rangle]\!]\}$ can only be no smaller than $\mathbf{inf}\{\mathbf{Dist}(\nu, [\![\phi_{post}]\!]) | \nu \in [\![\phi_{pre}\langle\textsc{attacked}(\alpha, S_A, o_1)\rangle]\!]\}$. □

Proof of Proposition 3. We show $u_1 = u$. Property $u_2 = u$ can be proved analogously.

Property $u_1 = u$ follows if we show that for all states $\omega \in [\![\psi]\!]$ it holds

$$\mathbf{Dist}(\omega, [\![\phi]\!]) = \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!]). \tag{1}$$

By definition, we have:

$$\mathbf{Dist}(\omega, [\![\phi]\!]) = \begin{cases} \mathbf{inf}\{\rho(\omega,\nu) \mid \nu \in [\![\neg\phi]\!]\} & \text{if } \omega \in [\![\phi]\!] \\ -\mathbf{inf}\{\rho(\omega,\nu) \mid \nu \in [\![\phi]\!]\} & \text{if } \omega \notin [\![\phi]\!] \end{cases}$$

and

$$\mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!]) = \begin{cases} \mathbf{inf}\{\rho_{\textsc{Var}(\phi)}(\omega,\nu) \mid \nu \in [\![\neg\phi]\!]\} & \text{if } \omega \in [\![\phi]\!] \\ -\mathbf{inf}\{\rho_{\textsc{Var}(\phi)}(\omega,\nu) \mid \nu \in [\![\phi]\!]\} & \text{if } \omega \notin [\![\phi]\!]. \end{cases}$$

We prove Eq. 1, by distinguishing two cases, $\omega \in [\![\phi]\!]$ and $\omega \notin [\![\phi]\!]$.

<u>Case $\omega \in [\![\phi]\!]$</u>. Since $\rho(\omega,\nu) \geq \rho_{\textsc{Var}(\phi)}(\omega,\nu)$ for all states ν, we infer $\mathbf{Dist}(\omega, [\![\phi]\!]) \geq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$. We can prove that also $\mathbf{Dist}(\omega, [\![\phi]\!]) \leq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$ holds, thus confirming Eq. 1. For an arbitrary $\nu \in [\![\neg\phi]\!]$, $\rho_{\textsc{Var}(\phi)}(\omega,\nu)$ is equal to $\rho(\omega,\nu')$, where

$$\nu' = \begin{cases} x \mapsto \nu(x) & \text{if } x \in \textsc{Var}(\phi) \\ x \mapsto \omega(x) & \text{otherwise} \end{cases}$$

and ν' belongs to $[\![\neg\phi]\!]$. By the arbitrarity of ν in $[\![\neg\phi]\!]$, we get $\mathbf{inf}\{\rho_{\textsc{Var}(\phi)}(\omega,\nu) \mid \nu \in [\![\neg\phi]\!]\} \geq \mathbf{inf}\{\rho(\omega,\nu) \mid \nu \in [\![\neg\phi]\!]\}$, which gives $\mathbf{Dist}(\omega, [\![\phi]\!]) \leq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$.

<u>Case $\omega \notin [\![\phi]\!]$</u>. Since $\rho(\omega,\nu) \geq \rho_{\textsc{Var}(\phi)}(\omega,\nu)$ for all states ν, we infer $\mathbf{Dist}(\omega, [\![\phi]\!]) \leq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$. We can prove that also $\mathbf{Dist}(\omega, [\![\phi]\!]) \geq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$ holds, thus confirming Eq. 1. For an arbitrary $\nu \in [\![\phi]\!]$, $\rho_{\textsc{Var}(\phi)}(\omega,\nu)$ is equal to $\rho(\omega,\nu')$, where

$$\nu' = \begin{cases} x \mapsto \nu(x) & \text{if } x \in \textsc{Var}(\phi) \\ x \mapsto \omega(x) & \text{otherwise} \end{cases}$$

and ν' belongs to $[\![\phi]\!]$. By the arbitrarity of ν in $[\![\phi]\!]$, we get $\mathbf{inf}\{\rho_{\textsc{Var}(\phi)}(\omega,\nu) \mid \nu \in [\![\phi]\!]\} \geq \mathbf{inf}\{\rho(\omega,\nu) \mid \nu \in [\![\phi]\!]\}$, which gives $\mathbf{Dist}(\omega, [\![\phi]\!]) \geq \mathbf{Dist}_{\textsc{Var}(\phi)}(\omega, [\![\phi]\!])$. □

Proof of Theorem 4. Let $\mathcal{H}$ be the set $\textsc{Var}(\phi_{pre})$. We have to prove $\textsc{B-safe}_{r_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ with $r_1 \geq r - d$. By definition, we have $\textsc{B-safe}_{r_1}(\textsc{attacked}(\alpha, S_A, o), \phi_{pre}, \phi_{post})$ if $r_1 = \mathbf{inf}\{\mathbf{Dist}(\omega, [\![[\textsc{attacked}(\alpha, S_A, o)]\phi_{post}]\!]) | \omega \in [\![\phi_{pre}]\!]\}$. Consider an arbitrary state $\omega \in [\![\phi_{pre}]\!]$. We have to prove that for each $\omega' \in [\![\langle\textsc{attacked}(\alpha, S_A, o)\rangle\neg\phi_{post}]\!]$ we have $\rho(\omega,\omega') \geq r - d$. The thesis is immediate if $\omega' \in [\![\langle\alpha\rangle\neg\phi_{post}]\!]$, since in that case the hypothesis $\textsc{B-safe}_r(\alpha, \phi_{pre}, \phi_{post})$, which coincides with $r = \mathbf{inf}\{\mathbf{Dist}(\omega, [\![[\alpha]\phi_{post}]\!]) \mid \omega \in [\![\phi_{pre}]\!]\}$, ensures that $\rho(\omega,\omega') > r$. The interesting case is $\omega' \in [\![[\alpha]\phi_{post}]\!]$. By the hypothesis $\textsc{attacked}(\alpha, S_A, o) \sqsubseteq^{\mathrm{B}}_{\phi_{post},\mathcal{H},d} \alpha$ there is an $\omega'' \in [\![\langle\alpha\rangle\neg\phi_{post}]\!]$ with $\rho_{\mathcal{H}}(\omega',\omega'') \leq d$. The hypothesis $\textsc{B-safe}_r(\alpha, \phi_{pre}, \phi_{post})$ and Proposition 3 ensure that $r = \mathbf{inf}\{\mathbf{Dist}_{\mathcal{H}}(\omega, [\![[\alpha]\phi_{post}]\!]) \mid \omega \in [\![\phi_{pre}]\!]\}$. This inequality together with $\omega'' \in [\![\langle\alpha\rangle\neg\phi_{post}]\!]$ give $\rho_{\mathcal{H}}(\omega'',\omega) \geq r$. Since $\rho_{\mathcal{H}}(\_,\_)$ is a metric, it is symmetric, thus implying that $\rho_{\mathcal{H}}(\omega',\omega'') = \rho_{\mathcal{H}}(\omega'',\omega')$. Moreover $\rho_{\mathcal{H}}(\_,\_)$ satisfies the triangular property, which ensures that

$$\rho_{\mathcal{H}}(\omega,\omega'') \leq \rho_{\mathcal{H}}(\omega,\omega') + \rho_{\mathcal{H}}(\omega',\omega'')\,.$$

From this inequality, $\rho_{\mathcal{H}}(\omega'',\omega) \geq r$ and $\rho_{\mathcal{H}}(\omega',\omega'') \leq d$ we infer $\rho_{\mathcal{H}}(\omega,\omega') \geq r - d$. By the arbitrarity of ω we infer

$$\mathbf{inf}\{\mathbf{Dist}_{\mathcal{H}}(\omega, [\![[\textsc{attacked}(\alpha, S_A, o)]\phi_{post}]\!]) | \omega \in [\![\phi_{pre}]\!]\} \geq r - d$$

By Proposition 3, this implies

$$\mathbf{inf}\{\mathbf{Dist}(\omega, [\![[\textsc{attacked}(\alpha, S_A, o)]\phi_{post}]\!]) | \omega \in [\![\phi_{pre}]\!]\} \geq r - d$$

This completes the proof. □