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

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 pre-condition 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 post-condition 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 weakening of the postcondition 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 (dL) [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 dL formulas and use the theorem prover developed for dL, 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).
- dL 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 (\(x := θ\), where \(θ\) is a real-valued term) or nondeterministically assigned (\(x := *\)).
Hybrid program $x' = \theta \land \phi$ 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 $\phi$, the \textit{evolution domain constraint}, holds for all of that time. Note that $x$ can be a vector of variables and then $\theta$ 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 arithmetic operations for the $\oplus$ ($+ \times$), and $\land \lor \implies \equiv \exists \forall$.

### Definition 1 (Safety)

A hybrid program $\alpha$ is safe for $\phi_{post}$ assuming $\phi_{pre}$, denoted $safety(\alpha; \phi_{pre}; \phi_{post})$, if $\phi_{pre} \implies [\alpha] \phi_{post}$ holds.

$safety(\alpha; \phi_{pre}; \phi_{post})$ means if $\phi_{pre}$ is true then $\phi_{post}$ holds after any possible execution of $\alpha$. The hybrid program $\alpha$ often has the form $(ctrl; plant)^t$, 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 $x' = \theta \land \phi$.

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 ($\phi_{post}$) 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 $temp$ 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 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 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 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., $delta = 1$).
2.2 Modeling Sensor Attacks
Recent work introduces a framework for modeling and analyzing sensor
attacks in the setting of hybrid programs and dL [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 tempera-
ture 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 exam-
ple, 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 := v, 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
\[ \rho(\omega, \nu) = \sqrt{\sum_{x \in \mathcal{Y}} (\omega(x) - \nu(x))^2} \]

Notice that \( \rho \) 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 \( \omega, \nu, \mu \in \text{STA} \).
For a state \( \omega \) and a real \( e > 0 \), the ball of ray \( e \) centered in \( \omega \) is
the set of states \( B(\omega, e) = \{ \nu | \rho(\omega, \nu) \leq e \} \).

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

- The distance between a state \( \omega \) and a set of states \( S \subseteq \text{STA} \)
  is the shortest distance between \( \omega \) and all states in \( S \), that is,
  \( \text{dist}(\omega, S) = \inf \{ \rho(\omega, \nu) | \nu \in S \} \)
- The depth of \( \omega \) in \( S \) is the shortest distance between \( \omega \)
  and the boundary of \( S \), that is, \( \text{depth}(\omega, S) = \inf \{ \rho(\omega, \nu) | \nu \in \text{STA} \setminus S \} \)
- The signed distance between \( \omega \) and a set of states \( S \subseteq \text{STA} \)
  is defined as follows:
  \[
  \text{Dist}(\omega, S) = \begin{cases} 
  \text{depth}(\omega, S), & \text{if } \omega \in S \\
  -\text{dist}(\omega, S), & \text{if } \omega \notin S 
  \end{cases}
  \]

Note that in the first case the distance is a positive real number,
while in the second case the distance is negative. Thus, \( \text{Dist}(\omega, S) > 0 \)
implies that \( B(\omega, e) \sqsubseteq S \) for all \( e < \text{Dist}(\omega, S) \), whereas \( \text{Dist}(\omega, S) \geq 0 \)
implies that \( B(\omega, e) \sqsubseteq \text{STA} \setminus S \) for all \( e < -\text{Dist}(\omega, S) \).
\( \text{Dist}(\omega, S) = 0 \) is not very informative. In all these definitions, we assume
that \( \inf \emptyset = \infty \) and \( \inf \mathbb{R} = -\infty \). And we consider the operator \( \inf \) in the set of \( \mathbb{R} \cup \{ \infty, -\infty \} \),
therefore every set has an infimum.

3 Quantitative Safety
The Boolean notion of safety in dL, e.g., \text{safe}(a, \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 dL. In defining quantitative safety, we use hybrid program \( a \) to
model a system of interest, formula \( \phi_{pre} \) as the precondition of
the system, and \( \phi_{post} \) as the postcondition.

3.1 Extended dL
To help define quantitative safety, we extend dL with another syn-
tactic structure: \( \phi(a) \), which intuitively represents the strongest
postcondition after the execution of the program \( a \) in a state satis-
fying the precondition \( \phi \). Its formal definition is the following:
\[
\phi(a) = \{ \nu | \exists \omega \text{ such that } \omega \in [\phi] \text{ and } (\omega, \nu) \in [a] \}
\]

Its dual is the modality of necessity \( \Box \phi \), which represents the
weakest precondition to ensure that \( \phi \) is satisfied after any execution
of program \( a \). 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.,
\text{safe}(a, \phi_{pre}, \phi_{post}), is \text{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 post-
condition \( \phi_{pre}(a) \) (obtained by the execution of program \( a \) in the
precondition \( \phi_{pre} \)) is with respect to the postcondition \( \phi_{post} \).
In other words, this degree of safety gives an indication of the margins
on possible strengthening of the postcondition \( \phi_{post} \).

Definition 2 (Forward quantitative safety). Given a real \( u \in \mathbb{R} \)
and formula \( \phi_{pre} \) and \( \phi_{post} \), a hybrid program \( a \) is forward u-
safe for \( \phi_{pre} \) and \( \phi_{post} \), denoted \( \text{F-safe}_{u}(a, \phi_{pre}, \phi_{post}) \), if \( u = \inf \{ \text{Dist}(\nu, [\phi_{post}]) | \nu \in \text{DIST}(\phi_{pre}(a)) \} \).

Given a system \( a \) and a precondition \( \phi_{pre} \), the real number \( u \)
measures the shortest distance between the set of states satisfying
the strongest postcondition \( \phi_{pre}(a) \) and the set of unsafe states.
If \( u \) is positive, then all reachable states by the system \( a \) from initial
states satisfying the precondition \( \phi_{pre} \) 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 \( \phi_{post} \). 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 \( \phi_{pre} \), during the execution of the system
the temperature lays in the real interval \([99.5, 101]\). Then, we have
\( \text{F-safe}_{4}(a, \phi_{pre}, \phi_{post}) \), where \( a = (\text{ctrl} \circ \text{plant})^* \) for \( u = 4 \). So, \( u \)
is our “degree of safety” w.r.t. \( \phi_{post} \): the system will always satisfy the
postcondition \( \text{temp}_{p} \leq 105 \) with a margin of at least 4 degrees.
Suppose we have a different postcondition \( \phi''_{post} \equiv \text{temp}_{p} <= 101 \).
In this case, we have \( \text{F-safe}_{0}(a, \phi_{pre}, \phi''_{post}) \), for \( u = 0 \), and the
system is actually safe, as \( \text{safe}(a, \phi_{pre}, \phi''_{post}) \) holds. However,
for a slightly different postcondition \( \phi'''_{post} \equiv \text{temp}_{p} < 101 \), we still have
\( \text{F-safe}_{0}(a, \phi_{pre}, \phi'''_{post}) \), for \( u = 0 \), but the system is actually unsafe,
as \( \text{safe}(a, \phi_{pre}, \phi'''_{post}) \) 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 \text{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 precondi-
tion \( \phi_{pre} \) is with respect to the weakest precondition \( [\alpha] \phi_{post} \).
while ensuring safety (i.e., $\phi_{post}$) after executions of the system $\alpha$. In other words, this degree of safety gives an indication of the margins on a possible weakening of the precondition $\phi_{pre}$. 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 $\phi_{pre}$ and $\phi_{post}$, a hybrid program $\alpha$ is backward $r$-safe for $\phi_{pre}$ and $\phi_{post}$, denoted $B\text{-SAFE}_r(\alpha, \phi_{pre}, \phi_{post})$, if $r = \inf \{\text{Dist}(\omega, \llbracket [\alpha] \phi_{post} \rrbracket) \mid \omega \in \llbracket \phi_{pre} \rrbracket\}$.

Here, if $r$ is positive then any execution of the system that starts from initial states in $\phi_{pre}$ 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 $\phi_{pre}$ can lead the system’s execution to an unsafe state. Similar to the forward quantitative safety, if $r = 0$ the system cannot be considered safe.

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

The Boolean version of safety, $\text{SAFE}(\alpha, \phi_{pre}, \phi_{post})$ of Definition 1, can be expressed in terms of backward quantitative safety.

**Proposition 1.** Given a program $\alpha$ and formula $\phi_{pre}$ and $\phi_{post}$.

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

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. If one degree is negative, the other is non-positive.

So the following proposition holds:

**Proposition 2.**

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

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

Note that given a system $\alpha$, formula $\phi_{pre}$ and $\phi_{post}$, forward quantitative safety, i.e., $F\text{-SAFE}_u(\alpha, \phi_{pre}, \phi_{post})$ 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.

\[ \text{ctrl'} = \text{temps} := \star; \]
\[ \text{if } (\text{temps} \geq \text{temp}_p - 0.3 \land \text{temps} \leq \text{temp}_p + 0.3); \]

**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 $\text{temps} := \star$. 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 $q_k = q_p + o$, where $q_k$ 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 $\alpha$, a set of sensors $S_A \subseteq \text{Var}(\alpha)$ and an offset function $o : S_A \rightarrow \mathbb{R}^2$, we write $\text{ATTACKED}(\alpha, S_A, o)$ to denote the program obtained by replacing in $\alpha$ all assignments to variables $q_k \in S_A$, with programs of the form $q_k := o; (q_k \geq q_p - o(q_k) \land q_k \leq q_p + o(q_k))$.

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 $\alpha$, a set of sensors $S_A \subseteq \text{Var}(\alpha)$ and two offset functions $o_1 : S_A \rightarrow \mathbb{R}^2$ and $o_2 : S_A \rightarrow \mathbb{R}^2$, 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 $\phi_{pre}$ and $\phi_{post}$. Then, if

- $F\text{-SAFE}_u(\alpha, \phi_{pre}, \phi_{post})$ for some $u > 0$;
- $F\text{-SAFE}_u(\alpha, \phi_{pre}, \phi_{post})$ for some $u > 0$;
- $F\text{-SAFE}_u(\alpha, \phi_{pre}, \phi_{post})$ for some $u > 0$.

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

Note that given a system $\alpha$, formula $\phi_{pre}$ and $\phi_{post}$, forward quantitative safety, i.e., $F\text{-SAFE}_u(\alpha, \phi_{pre}, \phi_{post})$ 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.

### 4.2 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 ro-
bustness, 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 \( \alpha \), a set of sensors \( S_A \subseteq \text{Var}(\alpha) \), an offset function \( \phi : S_A \to \mathbb{R}^0 \), real numbers \( u, u_1, \delta \in \mathbb{R} \), and properties \( \phi_{\text{pre}} \) and \( \phi_{\text{post}} \), we say that \( \alpha \) is forward \( \delta \)-robust under \( \phi \)-bounded \( S_A \)-attacks, written \( F\text{-ROBUST}(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}, S_A, \phi, \delta) \), if:

- \( F\text{-SAFE}_0(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}) \), with \( u > 0 \)
- \( F\text{-SAFE}_{u_1}(\text{ATTACKED}(\alpha, S_A, \phi), \phi_{\text{pre}}, \phi_{\text{post}}) \)
- \( \delta = u_1/u \).

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 \( \delta \) indicates the system’s robustness under the sensor attack. Note that by Theorem 1, we know that \( u_1 \leq u \). We can analyze \( \delta \) by the following cases:

- \( \delta = 1 \): the attack doesn’t affect the system’s forward safety.
- \( 0 < \delta < 1 \): then \( 0 \leq u_1 < u \). 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 proportion of forward safety that is lost due to the attack. The closer \( \delta \) is to 1, the more robust the system is.
- \( \delta < 0 \): then \( u > 0 \) and \( u_1 = 0 \). 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 \( \delta \), 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 \( F\text{-SAFE}_0(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}) \) for \( \phi_{\text{pre}} = \text{temp}_0 = 100 \) and \( \phi_{\text{post}} = \text{temp}_0 \leq 105 \), where \( \alpha \) models the original system. For the compromised system shown in Figure 5, starting again from \( \phi_{\text{pre}} \), during executions of \( \text{ATTACKED}(\alpha, S_A, \phi) \), the temperature lies in \([99.2, 101.3]\). Thus we know that \( F\text{-SAFE}_{0.7}(\text{ATTACKED}(\alpha, S_A, \phi), \phi_{\text{pre}}, \phi_{\text{post}}) \). The degree of forward robustness of the original system with respect to the attack is: \( \delta = 3.7/4 = 0.925 \).

The value of \( \delta \) 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 \( \delta \), may be considered better than another one with more safety loss.

Note that using a ratio for \( \delta \) 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, back-
ward 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 \( \alpha \), a set of sensors \( S_A \subseteq \text{Var}(\alpha) \), an offset function \( \phi : S_A \to \mathbb{R}^0 \), real numbers \( r_1, \delta \in \mathbb{R} \), and properties \( \phi_{\text{pre}} \) and \( \phi_{\text{post}} \), we say that \( \alpha \) is backward \( \delta \)-robust under \( \phi \)-bounded \( S_A \)-attacks, written \( B\text{-ROBUST}(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}, S_A, \phi, \delta) \), if:

- \( B\text{-SAFE}_1(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}) \), with \( r > 0 \)
- \( B\text{-SAFE}_{\delta}(\text{ATTACKED}(\alpha, S_A, \phi), \phi_{\text{pre}}, \phi_{\text{post}}) \)
- \( \delta = r_1/r \).

Again, backward robustness applies only to systems that are safe
when not exposed to sensor attacks (i.e., \( r > 0 \)). We can also analyze \( \delta \) 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 \( \alpha \) or \( \text{ATTACKED}(\alpha, S_A, \phi) \) from initial states in \( \phi_{\text{pre}} \) 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 \( \delta \) is to 1, the more robust the system is.
- \( \delta < 0 \): 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 \( \delta \), 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 \( \phi_{\text{pre}} = \text{temp}_0 = 100 \) and \( \phi_{\text{post}} = \text{temp}_0 \leq 105 \), we already know that \( B\text{-SAFE}_1(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}) \), where \( \alpha \) models the original system, for \( r = 5.0 \). Consider a sensor attack that offset 0.3 degree of sensor readings, formula \( \phi_{\text{pre}} \). We know that \( B\text{-SAFE}_{0.7}(\text{ATTACKED}(\alpha, S_A, \phi), \phi_{\text{pre}}, \phi_{\text{post}}) \) 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 infi-
mum. 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.,
\( \rho(\phi, \psi) \), 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 preclusion and postcondition. And thus computing distance
over these variables give us the quantitative distance of
Theorem 3. For a hybrid program $\alpha$, a set of variables $\mathcal{S} \subseteq \text{Var}(\alpha)$, formulas $\varphi_{\text{pre}}$ and $\varphi_{\text{post}}$, an offset function $\omega$, and $d, u \in \mathbb{R}$, if

- $F\text{-SAFE}_{\omega}(\alpha, \varphi_{\text{pre}}, \varphi_{\text{post}}, \mathcal{S}, \omega, d, u)$, with $u > 0$
- $\text{ATTACKED}(\alpha, \mathcal{S}, \omega) \subseteq F_{\varphi_{\text{pre}} \land \text{Var}(\varphi_{\text{post}})}d \alpha$

then $F\text{-ROBUST}(\alpha, \varphi_{\text{pre}}, \varphi_{\text{post}}, \mathcal{S}, \omega, \delta)$, for some $\delta$ such that $\delta \geq (u - d)/u$.

The theorem says that $d$ is an upper bound of the loss of forward safety, meaning that $F\text{-SAFE}_{\omega}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega), \varphi_{\text{pre}}, \varphi_{\text{post}})$, for some $u$ such that $d \geq u - u$. If $\mathcal{H}$ be the set of variables $\text{Var}(\varphi_{\text{post}})$. We need to prove $F\text{-SAFE}_{\omega}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega), \varphi_{\text{pre}}, \varphi_{\text{post}})$ for $u \geq u - d$. By definition, we have $F\text{-SAFE}_{\omega}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega), \varphi_{\text{pre}}, \varphi_{\text{post}})$ if $u_i = \text{inf}\{\text{Dist}(\nu, \varphi_{\text{post}})\} \in \varphi_{\text{pre}}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega))$. Consider an arbitrary state $v \in \varphi_{\text{pre}}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega))$. By the hypothesis $\text{ATTACKED}(\alpha, \mathcal{S}, \omega) \subseteq F_{\varphi_{\text{pre}} \land \mathcal{H}, d} \alpha$, we infer that there is some state $v' \in \varphi_{\text{pre}}(\alpha)$ with $\rho_n(v, v') \leq d$. The hypothesis $F\text{-SAFE}_{\omega}(\alpha, \varphi_{\text{pre}}, \varphi_{\text{post}})$ coincides, by definition, with property $u = \text{inf}\{\text{Dist}(\nu, \varphi_{\text{post}})\} \in \varphi_{\text{pre}}(\alpha)$. By Proposition 3, we infer $u = \text{inf}\{\text{Dist}(\nu, \varphi_{\text{post}})\} \in \varphi_{\text{pre}}(\alpha)$. Since $v' \in \varphi_{\text{pre}}(\alpha)$, we infer $\text{Dist}(\nu, \varphi_{\text{post}}) \geq u$. Since $\rho_n(\omega, \nu)$ is a metric, it is symmetric, thus implying $\rho_n(v, v') = \rho_n(v', v)$, and satisfies the triangular property. By the triangular property we infer that for any $v'' \not\in \varphi_{\text{post}}$

then $F\text{-ROBUST}(\alpha, \varphi_{\text{pre}}, \varphi_{\text{post}}, \mathcal{S}, \omega, \delta)$, for some $\delta \geq (u - d)/u$.

By definition of $\text{Dist}(\nu, \varphi_{\text{post}})$ we know $\rho_n(v, v') \leq u$, and since $\rho_n(v, v') \leq d$, then $\rho_n(v, v'') \geq u - d$. By definition of $\text{Dist}(\nu, \varphi_{\text{post}})$ and the arbitrariness of $v$, we infer $\text{Dist}(\nu, \varphi_{\text{post}}) \geq u - d$. By Proposition 3 we infer:

$\text{inf}\{\text{Dist}(\nu, \varphi_{\text{post}})\} \in \varphi_{\text{pre}}(\text{ATTACKED}(\alpha, \mathcal{S}, \omega)) \geq u - d$. This completes the proof.

Note that Theorem 3 may also hold for supersets of $\text{Var}(\varphi_{\text{post}})$, e.g., it holds for $\text{Var}(\varphi_{\text{pre}}) \cup \text{Var}(\varphi_{\text{post}})$. However, the forward simulation distance $d$ w.r.t. a superset is no smaller than the value of $d$ for $\text{Var}(\varphi_{\text{post}})$, since $\rho_n(\omega, \nu)$ 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 $\beta$ and $\alpha$ are in backward simulation distance $d$ if the postcondition, $\alpha$ can mimic the behaviors of $\beta$, i.e., $\alpha$ is able to reach states whose distance with those reached by $\beta$ is at most $d$.

Definition 9 (Backward simulation distance). For hybrid programs $\beta$ and $\alpha$, formula $\varphi_{\text{post}}$ and a set of variables $\mathcal{H}$, $\beta$ and $\alpha$ are at backward simulation distance $d$ w.r.t. $\varphi_{\text{post}}$ and $\mathcal{H}$, written $\beta \equiv B_{\varphi_{\text{post}} \land \mathcal{H}, d} \alpha$, if for each state $v_1 \in [\varphi_{\text{post}}(\beta)]$ there exists a state $v_2 \in [\varphi_{\text{pre}}(\alpha)]$ such that $\rho_n(v_1, v_2) \leq d$.
For example, let \( a \) 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 \( H \) be \( \text{VAR}(\phi_{\text{pre}}) = \{\text{temp}_{p}\} \). The backward distance between \( \text{ATTACKED}(a, S_A, o) \) and \( a \) w.r.t. \( \phi_{\text{post}} \) and \( H = 0 \), 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 \( \text{ATTACKED}(a, S_A, o) \) and \( a \) w.r.t. variable set \( \text{VAR}(\phi_{\text{pre}}) \), is indeed an upper bound to the loss of backward safety due to the attack.

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

\[
\begin{align*}
& \text{B-SAFE}_r(a, \phi_{\text{pre}}, \phi_{\text{post}}), \text{ with } r > 0 \\
& \text{ATTACKED}(a, S_A, o) \not\subset B_{\phi_{\text{post}}, \text{VAR}(\phi_{\text{pre}}), d}^B(a)
\end{align*}
\]

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

The theorem says that \( d \) is an upper bound of the loss of backward safety, meaning \( \text{B-SAFE}_r(a, \phi_{\text{pre}}, \phi_{\text{post}}) \). 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 \( e^F_{\phi_{\text{pre}}, H_d} \) and \( e^B_{\phi_{\text{pre}}, H_d} \) can be expressed as a dL 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 satisfy the two formulas. For example, the forward distance is computable on states satisfying, respectively, \( \phi_{\text{pre}}(\beta) \) and \( \phi_{\text{pre}}(\alpha) \). 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:

\[
(\phi \land (\overline{y} = \overline{x})) \rightarrow \exists \overline{x}, (\psi \land (\rho_o(\overline{x}, \overline{x}) \leq d))
\]

where \( \phi \) and \( \psi \) 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 \( \phi \) and \( \psi \), and \( \overline{y} \) are a list of fresh variables whose dimension is the same as \( \overline{x} \). Variables in \( \overline{y} \) are used to store values of \( \overline{x} \) that satisfy the first formula. The notation \( \rho_o(\overline{x}, \overline{y}) \) computes the distance between two vectors of variables w.r.t. the set \( H = \sqrt{\sum_{l \in H} (\overline{x}(i) - \overline{y}(i))^2} \), where \( \overline{x}(i) \) and \( \overline{y}(i) \) represent, respectively, the \( i \)th element in vector \( \overline{x} \) and \( \overline{y} \).

The encoding can be used to verify forward distance by letting \( \phi \) and \( \psi \), respectively, be \( \phi_{\text{pre}}(\text{ATTACKED}(a, S_A, o)) \) and \( \phi_{\text{pre}}(a) \).

Consider the cooling system example, we know \( \phi_{\text{pre}}(a) \) and \( \phi_{\text{pre}}(\text{ATTACKED}(a, S_A, o)) \), are, respectively, \( 99.2 < \text{temp}_{p} \leq 101.3 \) and \( 99.5 < \text{temp}_{p} \leq 101.0 \). We can express that \( \text{ATTACKED}(a, S_A, o) \) and \( a \) are at forward distance 0.3 w.r.t. \( \phi_{\text{pre}} \) and \( H = \text{VAR}(\phi_{\text{pre}}) = \{\text{temp}_{p}\} \) with the following formula:

\[
(99.2 < \text{temp}_{p} \leq 101.3 \land \neg \phi_{\text{post}}) \rightarrow (\exists \text{temp}_{p}, \neg \phi_{\text{post}})
\]

The encoding can also be instantiated for verifying backward simulation distance by letting \( \phi \) and \( \psi \) be \( \neg \phi_{\text{post}} \). The first line encodes "for each state that can be reached from precondition \( \phi_{\text{pre}} \) after an execution of the compromised program".

The second line encodes "there is an execution of the original program under precondition \( \phi_{\text{pre}} \) such that the distance between the corresponding final states is bound by \( d \)."

Similarly, we can use the following dL formula to express the backward simulation distance \( \text{ATTACKED}(a, S_A, o) \not\subset B_{\phi_{\text{post}}, H_d}^B(a) \):

\[
((\overline{y} = \overline{x}) \land (\overline{x} = \overline{x})) \rightarrow (\exists \overline{x}, \rho_o(\overline{x}, \overline{x}) \leq d) \land (\neg \phi_{\text{post}})
\]

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 reachable 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
vehicle that needs to stop before hitting an obstacle. For simplicity, we model the vehicle in just one dimension. Figure 6 shows a dL 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 (\( \psi_{post} \)) is that \( d_s \) 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/s²). 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 \( \epsilon \): distance changes according to \(-v_p\) (i.e., \( d_p' = -v_p \)), vehicle 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/s² or brake at \(-B \) m/s². 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 \( \psi \) is true, which captures that the vehicle can accelerate for the next \( \epsilon \) 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 parameters \( A, B, \epsilon \) 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 dL encodings with formulas, after computing the relevant weakest preconditions and strongest postconditions using these constants.

**Bounded Sensor Attack.** The formula \( \phi_{safety} \) specifies the desired (Boolean) safety property: given an appropriate precondition \( \phi_{pre} \), the safety condition \( \phi_{post} \) holds after any execution of the system. The safety property indeed holds. Also, by definition, the system satisfies F-SAFE(\( \phi_{pre}, \phi_{post} \)) and B-SAFE(\( \phi_{pre}, \phi_{post} \)), where \( a = (ctrl; plant) \). 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 \( v_s := v_p + \epsilon, \) \( 0 < \epsilon < 1 \) in Figure 6. The system is not robust against this attack, i.e., the safety property no longer holds when the sensor is compromised.

**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:

\[
\psi' = 2Bd_s > (v_p + 2)^2 + (A + B)(v_p + 2)\epsilon
\]

\[
\phi_{pre} = d_p > 0
\]

\[
\phi_{post} = d_p > 0
\]

Let \( \beta \) denote the new system, i.e., \( \beta = (ctrl'; plant) \). It still holds that F-SAFE(\( \beta, \phi_{pre}, \phi_{post} \)) and B-SAFE(\( \beta, \phi_{pre}, \phi_{post} \)). Consider a different precondition:

\[
\phi_{pre} = (2Bd_p > (v_p + 2)^2) \land v_p \geq 0
\]

Executing \( \beta \) given precondition \( \phi_{pre} \), we get a strongest postcondition \( (2d_p > (v_p + 2)^2) \land v_p \geq 0 \). So \( \beta \) is forward safe for a degree of 2 m/s. \( \phi_{post} \), i.e., F-SAFE(\( \beta, \phi_{pre}, \phi_{post} \)).

**Forward Simulation Distance.** We can prove that program ATTACKED(\( \beta, S_A, o \)) and \( \beta \) are at forward distance 1.5 w.r.t. \( \phi_{pre} \) and \( H = \{ \beta \} \). Here, \( \phi_{pre} \) (ATTACKED(\( \beta, S_A, o \))) is \((2Bd_p > (v_p + 1)^2) \land v_p \geq 0 \). Then the forward simulation distance can be expressed as:

\[
\exists d_p: (2d_p > (v_p + 2)^2) \land v_p \geq 0 \land \|d_p - fd_p\|^2 \leq 1.5)
\]

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

\[
\text{F-ROBUST}(\beta, \phi_{pre}, \phi_{post}, S_A, o, \delta)
\]

for some \( \delta \geq \frac{\sqrt{2}}{2} \approx 0.25 \). 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(\( \beta, \phi_{pre}, \phi_{post} \)), so there is not much we can learn from backward simulation distance here.

Now consider the backward safety of \( \beta \) w.r.t. \( \phi_{pre} \) and a different postcondition \( \phi_{post}' \) \( d_p > 0.5 \). We can compute that \( (\beta') \sim \phi_{post}' \) is \( d_p <= 0.5 \land (2d_p - 0.5) <= (v_p') \land v_p >= 0 \), and further compute B-SAFE(\( \beta', \phi_{pre}, \phi_{post}' \)). Moreover, ATTACKED(\( \beta, S_A, o \)) \( (v_p') \) is \( d_p <= 0.5 \land (2d_p <= (v_p + 1)^2) \land v_p >= 0 \). We can express that program ATTACKED(\( \beta, S_A, o \)) and \( \beta \) are at backward distance 1 w.r.t. \( \phi_{post}' \) and \( \text{VAR}(\phi_{pre}) \).

\[
((d_p <= 0.5 \land (2d_p <= (v_p + 1)^2) \land v_p >= 0)) \land (v_p' <= 0.5 \land (2d_p <= (v_p + 1)^2) \land v_p >= 0)) \land \|d_p - fd_p\|^2 + (v_p - fd_p) \leq 1
\]

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

\[
\text{B-ROBUST}(\beta, \phi_{pre}, \phi_{post}', S_A, o, \delta)
\]

for some \( \delta \geq \frac{\sqrt{2}}{2} \). 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 al. [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 $\phi_{\text{pre}}$ and postcondition $\phi_{\text{post}}$ of a system $\alpha$, we formalize two safety notions, quantitative forward safety, $F$-SAFE$_{\alpha}(\phi_{\text{pre}}, \phi_{\text{post}})$, and quantitative backward safety, $B$-SAFE$_{\alpha}(\phi_{\text{pre}}, \phi_{\text{post}})$, where $\alpha \in \mathbb{R}$ respectively express: (1) how strong the strongest postcondition $\phi_{\text{pre}}(\alpha)$ is with respect to the postcondition $\phi_{\text{post}}$, and (2) how strong the pre-condition $\phi_{\text{pre}}$ is with respect to the weakest precondition $[\alpha] \phi_{\text{post}}$. The bigger $u$ is, the safer the system is. On the contrary, if $u$ is negative, then some reachable states violate the safety condition $\phi_{\text{post}}$. If $u$ is 0, then the system cannot be considered safe. We introduce forward and backward robustness, $F$-ROBUST$(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}, S_\alpha, a, \delta)$ and $B$-ROBUST$(\alpha, \phi_{\text{pre}}, \phi_{\text{post}}, S_\alpha, a, \delta)$ respectively, to quantify the robustness $\delta$, with $\delta \leq 1$, for a system $\alpha$ 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 $\delta$ 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


A Definitions

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

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

$$ BV(\theta \circ \delta) = \emptyset \circ \in (\prec, \leq, =, \succ) $$

$$ BV(\neg \phi) = BV(\phi) $$

$$ BV(\phi \land \psi) = BV(\phi) \cup BV(\psi) $$

$$ BV(Vx. \phi) = \{x\} \cup BV(\phi) $$

$$ BV(\exists x. \phi) = BV(\phi) $$

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

$$ BV(x := \theta) = BV(x := \phi) $$

$$ BV(\theta) = \emptyset $$

$$ BV(x' = \theta \& \phi) = \{x', x\} $$

$$ BV(\alpha \colon \beta) = BV(\alpha \cup \beta) $$

$$ BV(\alpha) = BV(\alpha) $$

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

$$ MBV(x := \theta) = MBV(x := \phi) $$

$$ MBV(\theta) = \emptyset $$

$$ MBV(x' = \theta \& \phi) = \{x', x\} $$

$$ MBV(\alpha \cup \beta) = MBV(\alpha) \cap MBV(\beta) $$

$$ MBV(\alpha) = MBV(\alpha) $$

$$ MBV(\alpha) = \emptyset $$
We prove Eq. 1, by distinguishing two cases.

Case 1: if \( \omega, \nu \notin \phi \)

For an arbitrary \( v \in \phi \), \( \rho_{\text{MBV}}(\omega, v) \) is equal to \( \rho(\omega, v') \), where

\[
v' = \begin{cases} x \mapsto v(x) & \text{if } x \in \text{Var}(\phi) \\ x \mapsto \omega(x) & \text{otherwise}
\end{cases}
\]

\( \omega, \nu \) otherwise.

and \( v' \) belongs to \( \phi \). By the arbitrariness of \( v \) in \( \phi \), we get

\[
\inf \{ \rho_{\text{MBV}}(\omega, v) \mid v \in \phi \} \geq \inf \{ \rho(\omega, v) \mid v \in \phi \}.
\]

which gives \( \text{Dist}(\omega, \phi) \leq \text{Dist}_{\text{Var}}(\omega, \phi) \).

Case 2: if \( \omega, \nu \in \phi \).

By the arbitrarity of \( \nu \) for all states \( v, \) we infer \( \text{Dist}(\omega, \phi) \leq \text{Dist}_{\text{Var}}(\omega, \phi) \). We can prove that also \( \text{Dist}(\omega, \phi) \geq \text{Dist}_{\text{Var}}(\omega, \phi) \) holds, thus confirming Eq. 1.

We have to prove that for each \( \omega' \in (\text{ATTACKED}(\alpha, S_A, \emptyset)) \) we have \( \rho(\omega', \omega') \geq r \). The interesting case is \( \omega' \in (\text{ATTACKED}(\alpha, S_A, \emptyset)) \). By the hypothesis \( \text{ATTACKED}(\alpha, S_A, \emptyset) \subseteq \text{B-safe}_{\text{pre}, \text{Hsc}}(\alpha) \) there is an \( \omega'' \in (\omega') \) such that

\[
\text{Dist}(\omega, \phi) < \text{Dist}(\omega', \phi) \leq d
\]

with \( \rho_{\text{B}}(\omega', \omega'') \leq d \). The hypothesis \( \text{B-safe}_{\alpha, (\text{pre}, \text{post})} \) and Proposition 3 ensure that \( r = \inf \{ \text{Dist}(\omega, \phi) \mid \omega \in \phi \} \geq r - d \). By the arbitrariness of \( \omega \) we infer

\[
\text{Dist}(\omega, \phi) \geq r - d
\]

This completes the proof. \( \square \)