2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

# Stealthy Deactivation of Safety Filters

Daniel Arnström, André M.H. Teixeira

***Abstract*— Safety filters ensure that only safe control actions are executed. We propose a simple and stealthy false-data injection attack for deactivating such safety filters; in particular, we focus on deactivating safety filters that are based on control-barrier functions. The attack injects false sensor measurements to bias state estimates to the interior of a safety region, which makes the safety filter accept unsafe control actions. To detect such attacks, we also propose a detector that detects biases manufactured by the proposed attack policy, which complements conventional detectors when safety filters are used. The proposed attack policy and detector are illustrated on a double integrator example.**

## I. Introduction

Cyber-physical systems (CPSs) integrate communication, computation, and control technologies, which enable advanced control systems that are efficient, sustainable, and resilient [1]. The cyber components of CPSs do, however, also open up for new vulnerabilities since they enable cyber attacks [2][3, §4.C]. To address these novel vulnerabilities, several control-theoretic approaches that analyze and improve the security of CPSs have been proposed [4]. Such approaches are especially important for CPSs that are *safety critical*, since successful attacks to such systems can have severe, even fatal, consequences [3, §4.B].

In parallel with the development of the above-mentioned work on security for CSPs, a promising framework has been developed for safety of control systems through so-called *safety filters* [5]–[8]. A safety filter takes in a desired control action and the current state of the system, and outputs a filtered control action that guarantees a safe behaviour of the system. Such filters separate safety from performance, since any controller can be used to produce the desired control action; this allows for safety guarantees even when using unpredictable controllers such as experimental data-driven controllers [9].

In this paper we consider a cyber attack that injects false data on the communication channel from sensor measurements to a state observer, as illustrated in Figure 1. The goal of the attack is to produce synthetic measurements $y^a$ that “deactivate” the safety filter, which in turn allows for dangerous control actions to be applied to the plant. Our specific focus in this paper is on safety filters that are based on control-barrier functions (CBFs), but the main idea applies to other types of safety filters (see, e.g., [5] for a recent survey on safety filters).

D. Arnström and A. Teixeira are with the Division of Systems and Control, Depratment of Information Technology, Uppsala University, Sweden {daniel.arnstrom,andre.teixeira}@it.uu.se This work is supported by the Swedish Foundation for Strategic Research.

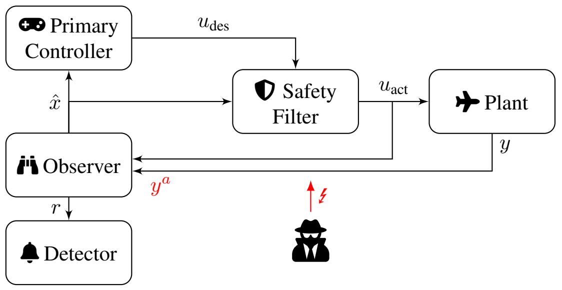


Fig. 1: Overview of the system architecture considered in this paper. The safety filter produces a safe control action $u_{\text{act}}$ given a desired control $u_{\text{des}}$ and the current estimated state $\hat{x}$. An adversary tries to deactivate this filter through false-data injections on the communication channel between the sensors and the observer by replacing the true measurement $y$ with a synthetic measurement $y^a$.

A common protection against false-data injections is anomaly detectors [10], which raise an alarm if the obtained measurement is too far from the expected measurement. Here we specifically consider *stealthy* attacks, which are designed to circumvent such anomaly detectors [2].

The proposed stealthy false-data injection attack biases the state estimates toward the center of a safe set, making the safety filter accept unsafe control actions. An important difference with the proposed attack and classical false-data injection attacks (see, e.g., [11]) is that the adversary does not need direct access to a dynamical model of the system under attack, which is due to the attack being tailored to deactivate safety filters. The contributions of this work can be interpreted as extending some of the work on adversarial examples from a static setting [12] to a dynamic setting.

In addition to an attack that biases state estimates towards the center of a safe set, we also propose a way of detecting such biases. To this end, we proposed a detector that correlates the difference of expected and actual measurements with directions towards the interior of the safe set.

To summarize, the main contributions of the paper are:

1) A stealthy false-data injection attack that biases state estimates to deactivate CBF-based safety filters.
2) An anomaly detector that detects measurement residuals that are biased toward the interior of the safe set.

### A. Definitions and notation

The gradient of a function $f(x)$ is denoted $\nabla f(x) \triangleq \frac{\partial f}{\partial x}$ and is represented as a column vector. The $i$th element of a vector $v$ is denoted $[v]_i$. The boundary of a set $S$ is denoted $\partial S$. A continuous function $\alpha : (-a, b) \to \mathbb{R}$, with $a, b > 0$, is said to be extended class $\mathcal{K}$ ($\alpha \in \mathcal{K}^e$) if it is strictly increasing and $\alpha(0) = 0$.

## II. Preliminaries and problem formulation

We consider dynamical systems of the form

$$\dot{x} = f(x, u), \tag{1}$$

with state $x \in \mathbb{R}^{n_x}$ and control $u \subseteq \mathbb{R}^{n_u}$. The dynamics of the system is given by $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. Moreover, the system generates measurements $y \in \mathbb{R}^{n_y}$ according to

$$y = h(x) + e, \tag{2}$$

with the measurement function $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ and measurement noise $e$ (for our purpose, we make no particular assumptions on $e$.)

An estimate $\hat{x} \in \mathcal{X}$ of the system state $x$ is obtained by an observer of the form

$$\dot{\hat{x}} = f(\hat{x}, u) + K(y - h(\hat{x})), \tag{3}$$

where the observer gain $K \in \mathbb{R}^{n_x \times n_y}$ might be time- and state-dependent. In addition to correcting the state estimate, the innovation $y - h(\hat{x})$ from the observer is used as the residual $r$ in an anomaly detector; in particular, an anomaly is flagged if $\|y - h(\hat{x})\| > \delta$, where $\delta > 0$ is a user-specified threshold that trades off sensitivity and false-detections.

### A. Safety and Invariance

In this paper we are interested in the *safety* of system (1), defined through admissible states and control actions.

*Definition 1 (Safety):* Given a set of admissible states $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and a set of admissible controls $\mathcal{U} \subseteq \mathbb{R}^{n_u}$, the system in (1) is said to be *safe* if

$$x(t) \in \mathcal{X} \text{ and } u(t) \in \mathcal{U} \text{ for all } t \geq 0. \tag{4}$$

A common way to ensure safety is to find a *forward invariant* subset of the admissible states. To define forward invariance, let $\kappa : \mathcal{X} \to \mathcal{U}$ be a control policy, which results in the closed-loop (autonomous) system

$$\dot{x} = f(x, \kappa(x)). \tag{5}$$

For an autonomous system of the form (5) we define forward invariance in the following way.

*Definition 2 (Forward invariance):* A set $\mathcal{S} \subseteq \mathbb{R}^{n_x}$ is *forward invariant* for the closed-loop system (5) if $x(0) \in \mathcal{S}$ implies that $x(t) \in \mathcal{S}$ for all $t \geq 0$.

Next, we make the connection between forward invariant sets and safety of (1) explicit.

*Lemma 1 (Forward invariance $\to$ safety):* The system in (1) is safe if there exists a control policy $\kappa : \mathcal{X} \to \mathcal{U}$ and a set $S \subseteq \mathcal{X}$ such that $S$ is forward invariant for the closed loop system (5).

*Proof:* From the range of $\kappa$ we directly have that $u(t) \in \mathcal{U}$ for all $t \geq 0$ since $u(t) = \kappa(x)$. Moreover, we have from the forward invariance of $S$ that $x(t) \in S$ for all $t \geq 0$, but since $S \subseteq \mathcal{X}$ we also have that $x(t) \in \mathcal{X}$ for all $t \geq 0$. ■

We will call invariant sets such that $\mathcal{S} \subseteq \mathcal{X}$ *safe sets*; additionally, we assume that safe sets can be characterized with a continuously differentiable function $h_S : \mathcal{X} \to \mathbb{R}$ as

$$\mathcal{S} = \{x \in \mathbb{R}^{n_x} : h_S(x) \geq 0\}. \tag{6}$$

### B. Safety filters

In the previous subsection we established the connection between the safety of (1) and the existence of a forward invariant set. To be able to use this in practice, however, a more pragmatic characterization of forward invariance is necessary. Most safety filters build on the following classical characterization of forward invariance:

*Theorem 1 (Nagumo's theorem [13]):* Let $S \subseteq \mathcal{X}$ be defined as $S \triangleq \{x \in \mathbb{R}^{n_x} | h_S(x) \geq 0\}$, where $h_S$ is continuously differentiable. Moreover, assume that the interior of $S$ is non-empty. Then $S$ is forward invariant for (5) if and only if

$$\dot{h}_S(x) = \nabla h_S(x)^T f(x, \kappa(x)) \geq 0 \tag{7}$$

for any $x \in \partial S$.

Intuitively, Theorem 1 states that forward invariance of $S$ is equivalent to the system state changing towards the interior of $S$ when it is at the boundary $\partial S$. The condition (7) is, however, not practical since it is only enforced on the boundary $\partial S$, which has measure zero. One motivation behind introducing *control-barrier functions* (CBFs) is to obtain a Nagumo-like condition that holds on the entire $S$. In particular, this is done through adding a strengthening term in the form of an extended class $\mathcal{K}$ function to the right-hand side of (7).

*Definition 3 (Control Barrier Function):* The function $h_S$ is a control-barrier function for (1) if there exists $\alpha \in \mathcal{K}^e$ such that

$$\sup_{u \in \mathcal{U}} \nabla h_S(x)^T f(x, u) \geq -\alpha(h_S(x)). \tag{8}$$

for any $x \in S$.

A key relationship between CBFs and forward invariance is stated in the following theorem.

*Theorem 2 (CBF $\to$ forward invariance [7]):* Let $S \subseteq \mathcal{X}$ be defined as $S \triangleq \{x \in \mathbb{R}^{n_x} | h_S(x) \geq 0\}$, where $h_S$ is a continuously differentiable CBF. Moreover, assume that $\nabla h_S(x) \neq 0$ for any $x \in \partial S$. Then any Lipschitz continuous control policy $\kappa : \mathcal{X} \to \mathcal{U}$ such that $\kappa(x) \in \tilde{\mathcal{U}}_S(x)$ makes $S$ forward invariant for the closed-loop system (5).

We introduce notation for the corresponding set of control actions $\tilde{\mathcal{U}}_S(x)$ that ensure forward invariance of $S$ given the state $x$; that is,

$$\tilde{\mathcal{U}}_S(x) \triangleq \{u \in \mathcal{U} : \nabla h_S(x)^T f(x, u) \geq -\alpha(h_S(x))\}. \tag{9}$$

A safety filter that is based on CBFs is the *Active Set Invariance Filter* (ASIF) [7] that filters control actions according to the policy

$$\begin{aligned} \kappa_{\text{ASIF}}(x, u_{\text{des}}) = &\underset{u \in \mathcal{U}}{\arg\min}\, \|u - u_{\text{des}}\|^2 \\ &\nabla h_S(x)^T f(x, u) \geq -\alpha(h_S(x)), \end{aligned} \tag{10}$$

where $u_{\text{des}} \in \mathbb{R}^{n_u}$ is a desired control action. If the set of admissible control actions $\mathcal{U}$ is a polyhedron and the dynamics $f$ is control affine, the resulting optimization problem in (10) is a quadratic program (QP). The ASIF is said to be *minimally invasive*, since it finds a safe control action that is as close as possible to the desired control action

$u_{\text{des}}$. In the rest of the paper, we will consider how safety filters of the form (10) can be "deactivated".

### C. Formulation of adversarial problem

The overarching objective of the adversary is to make the system unsafe; that is, to get $x \notin \mathcal{X}$. Since we assume that control actions applied to the system are filtered through a safety filter of the form (10), which ensures that $x \in \mathcal{X}$, an attacker *cannot* make the system unsafe as long as the state $x$ is estimated correctly. To this end, the adversary need to alter the measurement $y$ to bias the state estimate $\hat{x}$ such that the safety filter becomes "deactivated". We define *deactivation* of a CBF (and the corresponding ASIF given by that CBF) in the following way.

*Definition 4 (Deactivation):* The CBF is said to be *deactivated* by the state estimate $\hat{x}$ at time $t$ if $\exists u \in \tilde{\mathcal{U}}(\hat{x}(t))$ such that $u \notin \tilde{\mathcal{U}}(x(t))$.

In other words, a state estimate $\hat{x}$ deactivates a safety filter if there exist a control action that is deemed safe for $\hat{x}$ but unsafe for the true state $x$. When a safety filter is deactivated, there might exist a control signal that breaks the forward invariance of $S$ and, consequently, makes the system (1) unsafe. A delimitation we make in this paper is, however, that we are only interested in deactivating safety filters, not producing unsafe control actions, as stated in the following remark.

*Remark 1:* In this paper our focus is *not* on producing a control signal $u(\cdot)$ that makes the system unsafe, but rather to deactivate the safety filter such that unsafe control actions *could* be passed to the plant. Deactivation of the safety filter is a necessary first step in a two-pronged attack, where the second step would be to modify $u_{\text{des}}$ to make $x \notin \mathcal{X}$.

To deactivate the safety filter, we assume that the adversary has access to the following information:

*Assumption 1:* The adversary can freely modify $y \leftarrow y^a$.

*Assumption 2:* The adversary knows the observer gain $K$, the measurement function $h$, and the detector threshold $\delta$.

*Assumption 3:* The adversary can evaluate the function $h_S$ at the current state estimate $\hat{x}$.

*Assumption 4:* The adversary knows the current state estimate $\hat{x}$.

For satisfying Assumption 4, the adversary can either have direct access to the output of the observer, or run its own identical filter in parallel. In the latter case the adversary would need to know the dynamics of the system (i.e., $f$) and the initial state, in addition to $K$ and $h$ from Assumption 2.

In summary, the problem that the adversary is interested in solving is formalized as follows:

*Problem 1 (Problem formulation for adversary):* Under Assumptions 1–4, inject false measurements $y^a$ such that $\hat{x}$ deactivates the ASIF defined in (10), without being detected by the anomaly detector.

In the rest of the paper we consider a specific heuristic attack policy to tackle Problem 1.

## III. Stealthy deactivation of safety filter

The main idea behind the proposed attack policy is to inject false measurements $y^a$ to bias the state estimate $\hat{x}$ such that the safety filter gets a false sense of safety. To this end, the adversary injects measurements $y^a$ to increase the value of $h_S(\hat{x})$ (recall the $h_S$ defines the safe set $S$ in (6)), which can be interpreted as increasing the perceived safety margin to the boundary of $S$. As a result, since the safety margin is perceived to be larger than it actually is, the safety filter might erroneously accept dangerous control actions being applied to the plant.

To be more concrete, by increasing $h_S(\hat{x})$ the set $\tilde{\mathcal{U}}_S(\hat{x})$ of control actions that are perceived to be safe becomes larger since the right-hand-side of the inequality constraint in (9) becomes smaller. This incites deactivation of the safety filter.

For the adversary to remain stealthy, the injected measurement $y^a$ needs to be kept close to the expected measurement $h(\hat{x})$ to remain undetected by the anomaly detector; specifically, $\|y^a - h(\hat{x})\| \leq \delta$.

### A. Formalization of the attack

We formalize increasing the safety margin $h_S$ subject to a stealth constraint as finding $y$ that solves

$$\begin{aligned} y^a(\hat{x}) = \arg\max_{y} \ & \dot{h}_S(\hat{x}) \\ \text{subject to } & \|y - h(\hat{x})\| \leq \delta \\ & \text{and the dynamics in (3).} \end{aligned} \tag{11}$$

The following theorem makes the adversary's attack policy based on (11) more explicit.

*Theorem 3:* Solving (11) is equivalent to solving

$$\begin{aligned} y^a(\hat{x}) = \arg\max_{y} \ & \nabla h_S(\hat{x})^T K y \\ \text{subject to } & \|y - h(\hat{x})\| \leq \delta. \end{aligned} \tag{12}$$

*Proof:* Expanding $\dot{h}_S(\hat{x})$ and using (3) yields

$$\begin{aligned} \dot{h}_S(\hat{x}) &= \nabla h_S(\hat{x})^T \dot{\hat{x}} \\ &= \nabla h_S(\hat{x})^T \left( f(\hat{x}, u) + K(y - h(\hat{x})) \right). \end{aligned} \tag{13}$$

Since the only term that contains the decision variable $y$ is $\nabla h_S(\hat{x})^T K y$, we can use that term instead of $\dot{h}_S(\hat{x})$ in the objective function and still get the same optimizer. ∎

An interpretation of the objective in (12) is that $Ky$ is the change to the state due to the measurement $y$. Hence, $\nabla h_S(x)^T K y$ is a measure of how much the change in $x$ aligns with the gradient of the safety margin $h_S$.

The attack policy obtained through the optimization problem in (12) does, in fact, take a closed form when the stealth constraint is expressed in terms of the 2- or $\infty$-norm.

*Corollary 1 (Closed-form attack policy for 2-norm):* If the stealth constraint in (12) is posed in terms of the 2-norm and $\|K^T \nabla h_S(\hat{x})\| \neq 0$, the solution $y^a(\hat{x})$ takes the closed form

$$y^a(\hat{x}) = h(\hat{x}) + \delta \frac{K^T \nabla h_S(\hat{x})}{\|K^T \nabla h_S(\hat{x})\|_2}. \tag{14}$$

*Proof:* A proof is provided in Appendix A. ∎

*Corollary 2 (Closed-form attack policy for $\infty$-norm):* If the stealth constraint in (12) is posed in terms of the $\infty$-norm, the solution $y^a(\hat{x})$ takes the closed form

$$y^a(\hat{x}) = h(\hat{x}) + \delta \operatorname{sgn}(K^T \nabla h_S(\hat{x})), \tag{15}$$

where sgn is evaluated element-wise.

*Proof:* A proof is provided in Appendix B. ■

Both Corollary 1 and 2 make it straightforward for the adversary to select a false measurement $y^a$. No matter the norm, however, one can show that the resulting false-data injection results in a positive bias for $\dot{h}$, this is the main motivation behind the proposed attack.

*Theorem 4:* Let $\|\cdot\|_*$ denote the dual norm of $\|\cdot\|$, defined as $\|z\|_* = \max_{\|x\|\leq 1} z^T x$ (see, e.g., [14, §A.1.6] for details.) If the attack policy $y = y^a(\hat{x})$ is used, the change in the safety margin $h_S$ is given by

$$\dot{h}_S(\hat{x})\big|_{y=y^a(\hat{x})} = \nabla h_S(\hat{x})^T f(\hat{x}, u) + \delta \|K^T \nabla h_S(\hat{x})\|_*. \tag{16}$$

*Proof:* A proof is provided in Appendix C. ■

From Theorem 4, we see that the attack policy result in a nonnegative bias $\delta\|K^T\nabla h_S(\hat{x})\|$ to $\dot{h}_S(\hat{x})$, which is accordance with the objective of the adversary to make the safety margin $h_S(\hat{x})$ larger (which in turn makes the set $\tilde{\mathcal{U}}_S(\hat{x})$ larger.)

An approach for the attacker to remain as covert as possible, and performing attacks with high impact, is to only perform an attack if the system is operating close to the edge of the safe set. For example, an attack can be initiated if the attack condition $h_S(x) < \gamma$ is satisfied for some user-specified threshold $\gamma$.

We summarize the proposed false-data injection attack in Algorithm 1.

**Algorithm 1** False-data injection attack to deactivate the safety filter in (10).

```
Input: $K$, $h(\cdot)$, $h_S(\cdot)$
1: while attack condition is satisfied do
2:     get $\hat{x}$ (either directly access $\hat{x}$ or run parallel filter)
3:     $y^a \leftarrow$ formulate and solve (12)
4:     inject $y^a$ to the system
```

## B. Detections of attack

The anomaly detector that the attack policy in (12) circumvents considers only the *magnitude* of the of the residual $y - h(\hat{x})$. However, the attack policy builds on producing measurements that are biased towards the interior of the safe set $S$. As a result, the direction of the residuals will "unnaturally" align with $\nabla h_S$. A possible counter-measure to the attack is, hence, to detect if the residual aligns with $\nabla h_S$. A metric that measures such an alignment is

$$\rho(y, \hat{x}) \triangleq \frac{\nabla h_S(\hat{x})^T K(y - h(\hat{x}))}{\delta \|K^T \nabla h_S(\hat{x})\|}. \tag{17}$$

The denominator of (17) is a normalization factor to ensure that $|\rho(y, \hat{x})| \leq 1$. A large value of $\rho$ signifies that the change in the state $x$ due to the residual $(K(y - h(\hat{x})))$ aligns with $\nabla h_S$, i.e., is biased towards the interior of $S$.

*Corollary 3:* For the attack policy $y = y^a(\hat{x})$, the correlation measure $\rho$ in (17) is

$$\rho(y^a(\hat{x}), \hat{x}) = \frac{\|K^T \nabla h_S(\hat{x})\|_*}{\|K^T \nabla h_S(\hat{x})\|}. \tag{18}$$

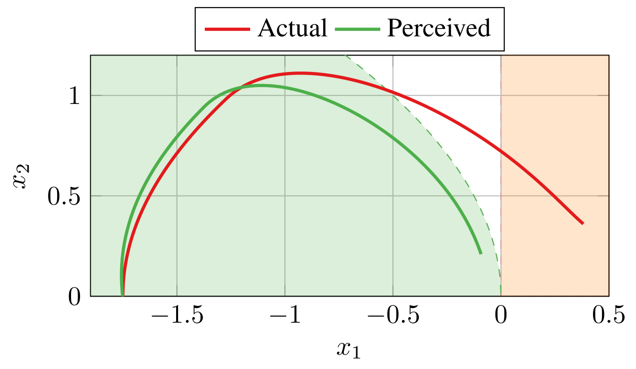


Fig. 2: The actual trajectory $x(t)$ and the perceived trajectory $\hat{x}(t)$ when the false-data injection attack defined by (12) is performed.

*Proof:* This is a direct biproduct of the proof of Theorem 4. Specifically, see (30). ■

A detector based on (17) can, for example, be implemented by thresholding a moving average of $\rho$ as

$$\frac{1}{T}\int_{t-T}^{t} \rho(y(\tau), \hat{x}(\tau)) d\tau > \nu, \tag{19}$$

with the horizon $T$ and threshold $\nu$.

## IV. Numerical Experiments

To illustrate the proposed false-data injection attack given in Algorithm 1, we consider the double integrator example in [15]. Code for all reported experiments is available at `https://github.com/darnstrom/ecc24-sf`.

The dynamics of the double integrator system is

$$f(x, u) = \begin{pmatrix} x_2 \\ u \end{pmatrix}. \tag{20}$$

For safety, the admissible states are $\mathcal{X} = \{x \in \mathbb{R}^2 : x_1 \leq 0\}$, and admissible control actions are $\mathcal{U} = \{u \in \mathbb{R} : |u| \leq 1\}$.

We consider the control invariant set $S$ defined as the zero-superlevel set of $h_S(x) = -2x_1 - x_2^2$. With the extended class $\mathcal{K}$ function $\alpha(x) = 2x$, the function $h_S$ is a control-barrier function with the corresponding set of safe control actions

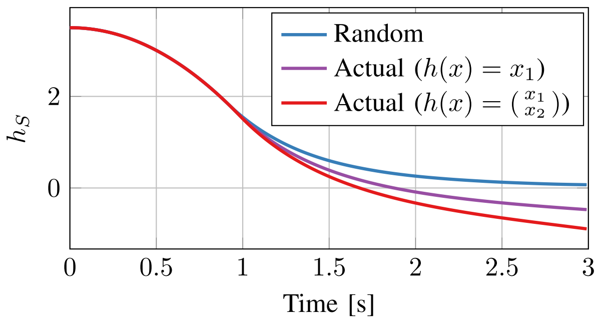
$$\tilde{\mathcal{U}}_S(x) = \{u \in \mathcal{U} : -2x_2(1 + u) \geq 4x_1 + 2x_2^2\}. \tag{21}$$

For the observer, the gain $K$ is selected as the stationary Kalman gain (obtained by using the process noise $Q = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and measurement noise $R = \left(\begin{smallmatrix} 0.001 & 0 \\ 0 & 0.001 \end{smallmatrix}\right)$.) Two scenarios of measurement functions $h$ are consider: either both $x_1$ and $x_2$ are measured (i.e., $h(x) = \left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right)$), or only $x_1$ is measured (i.e., $h(x) = x_1$). In the detector the 2-norm is used, and its threshold is set to $\delta = 0.001$.

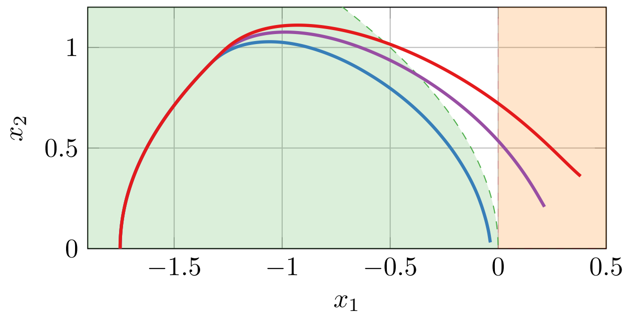
To activate the safety filter, the desired control action is constantly saturated at $u_{\text{des}} = 1$, which drives $x_1(t) \to \infty$ as $t \to \infty$ if there would be no safety filter.

For the implementation of the safety filter, the quadratic programming solver DAQP [16] is used to solve optimization problems of the form (12).

The attack is active for the entire duration of the simulation. Since the 2-norm is used in the detector we have that



(a) Safety margin



(b) Trajectories

Fig. 3: The safety margin and resulting state trajectories when an attack with random directions according to (22) is occurring, and when an adversary performs false-data injection attacks according to (12) with $h(x) = x_1$ and $h(x) = \binom{x_1}{x_2}$, respectively.

$y(t) = h(\hat{x}(t)) + \delta \frac{K^T \nabla h_S(\hat{x}(t))}{\|K^T \nabla h_S(\hat{x}(t))\|_2}$ for all $t \geq 0$ from the closed-form expression in Corollary 1.

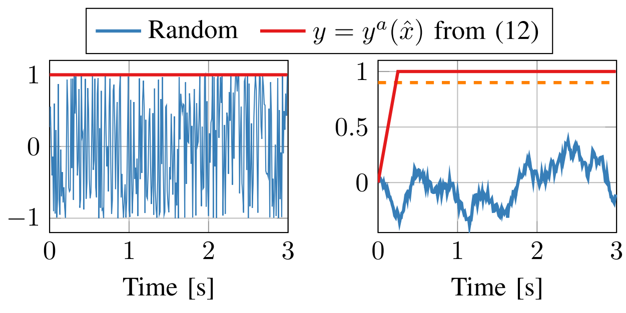
First we consider an attack when both states are measured. The system is started in $x(0) = \binom{-1.75}{0}$ and is simulated for 3 seconds. Figure 2 illustrates the resulting *perceived* trajectory (i.e., $\hat{x}(t)$) and the *actual* trajectory of the system (i.e., $x(t)$). The green region is the safe set $S$ and the red region marks the inadmissible states. As can be seen, the false-data injection makes the observer believe that the states remain within the safe set, while in actuality they leave it and pass over to the inadmissible states; the adversary's objective of making the system unsafe is, hence, achieved.

Next, we consider the same setup except that only the first state is measured (i.e., $h(x) = x_1$). The resulting trajectory $x(t)$ is shown in Figure 3b together with the trajectory from the first scenario and for when a stealthy attack with random directions of the form

$$y = h(\hat{x}) + \delta \frac{e}{\|e\|_2}, \qquad e \sim \mathcal{N}(0, 1), \tag{22}$$

is performed. The corresponding safety margin over the simulation is shown in Figure 3a. The results illustrate that the safety filter remains active when a naive random attack of the form (22) is performed. In contrast, the attack in (12) deactivates the safety filter, and the safe set is exited quicker when both states are measured, which is expected since then the adversary is given more degrees of freedom in the attack.

Finally, we illustrate how the correlation measure $\rho(y, \hat{x})$ in (17) can expose the proposed attack. Figure 4a shows $\rho$ from the first scenario when both states are measured and the attack is active throughout the entire simulation. In addition, Figure 4a shows $\rho$ for when an attack with random directions according to (22) is performed. Note that, in both instances, the attacks are stealthy w.r.t. the residual $y - h(\hat{x})$. In contrast to a random attack, $\rho$ is constantly 1 for the attack from (12) (as is expected from Corollary 3 since the dual norm of the 2-norm is also the 2-norm.) Figure 4b shows a corresponding moving average of $\rho$ according to (19) (with a horizon of $T = 0.25$ seconds) for the two attacks. For a threshold of, e.g., $\nu = 0.9$, an attack that biases state estimates toward the interior of $S$ is correctly detected when the attack in (12) is active, while the random attack correctly remains undetected.



(a) Correlation $\rho(y, \hat{x})$ (b) Moving average ($T = 0.25$)

Fig. 4: The correlation measure $\rho$ defined in (17) and a moving average according to (19) under two different attacks: a stealthy false-data injection attack with random directions according to (22), and a stealthy false-data injection attack according to (12). An example threshold ($\nu = 0.9$) for a detector of the form (19) is shown as a dashed line, which would detect the attack in (12) after about 0.2 seconds.

## V. Conclusion

We have proposed a simple and stealthy false-data injection attack for deactivating CBF-based safety filters. The attack injects false sensor measurements to bias state estimates to the interior of a safety region, which makes the safety filter accept unsafe control actions. We have also proposed a detector that detects biases manufactured by the proposed attack policy, which complements conventional detectors when safety filters are used. We have illustrated that an adversary can successfully make a system unsafe by performing the proposed false-data injection attack on a double integrator system. Moreover, we have shown with the same example that the proposed detector can be used to detect when such a false-data injection attack is happening.

Future work includes considering similar attacks but with less information required by the adversary; for example by considering that only the observations $y$ are available to the adversary. We will also explore if a receding horizon attack can be more effective that the attack considered herein.

## References

[1] S. M. Dibaji, M. Pirani, D. B. Flamholz, A. M. Annaswamy, K. H. Johansson, and A. Chakrabortty, "A systems and control perspective of CPS security," *Annual reviews in control*, vol. 47, pp. 394–411, 2019.

[2] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[3] A. Annaswamy, K. Johansson, and G. Pappas, *Control for Societal-scale Challenges: Road Map 2030*. IEEE Control Systems Society, May 2023.

[4] M. S. Chong, H. Sandberg, and A. M. Teixeira, "A tutorial introduction to security and privacy for cyber-physical systems," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 968–978.

[5] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[6] C. J. Tomlin, I. Mitchell, A. M. Bayen, and M. Oishi, "Computational techniques for the verification of hybrid systems," *Proceedings of the IEEE*, vol. 91, no. 7, pp. 986–1001, 2003.

[7] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[8] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 7130–7135.

[9] J. Coulson, J. Lygeros, and F. Dörfler, "Data-enabled predictive control: In the shallows of the DeePC," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 307–312.

[10] C. Murguia and J. Ruths, "On model-based detectors for linear time-invariant stochastic systems under sensor attacks," *IET Control Theory & Applications*, vol. 13, no. 8, pp. 1051–1061, 2019.

[11] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Preprints of the 1st workshop on Secure Control Systems*, vol. 1, 2010.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6572

[13] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[14] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[15] K. L. Hobbs, M. L. Mote, M. C. Abate, S. D. Coogan, and E. M. Feron, "Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems," *IEEE Control Systems Magazine*, vol. 43, no. 2, pp. 28–65, 2023.

[16] D. Arnström, A. Bemporad, and D. Axehill, "A dual active-set solver for embedded quadratic programming using recursive LDL$^T$ updates," *IEEE Transactions on Automatic Control*, vol. 67, no. 8, pp. 4362–4369, 2022.

## Appendix

### A. Proof of Corollary 1

*Proof:* First, note that the constraint $\|y - h(\hat{x})\|_2 \leq \delta$ can be equivalently stated in the differentiable form $\|y - h(\hat{x})\|_2^2 \leq \delta^2$. The KKT-conditions for (12) with the 2-norm constraint are then given by

$$K^T \nabla h_S(\hat{x}) + \lambda(y^* - h(\hat{x})) = 0 \tag{23a}$$

$$\|y^* - h(\hat{x})\|_2 \leq \delta \tag{23b}$$

$$\lambda \geq 0 \tag{23c}$$

$$\lambda(\delta - \|y^* - h(\hat{x})\|_2) = 0 \tag{23d}$$

If $K^T \nabla h_S(\hat{x}) \neq 0$ we have that $\lambda \neq 0$ from the stationarity condition (23a). Hence, we can rewrite (23a) as

$$y^* - h(\hat{x}) = \frac{K^T \nabla h_S(\hat{x})}{\lambda}. \tag{24}$$

Since $\lambda \neq 0$, the complementarity condition (23d) implies that $\|y^* - h(\hat{x})\|_2 = \delta$. Taking the 2-norm of both sides of (24) therefore gives

$$\delta = \frac{\|K^T \nabla h_S(\hat{x})\|_2}{\lambda} \Leftrightarrow \lambda = \frac{\|K^T \nabla h_S(\hat{x})\|_2}{\delta}. \tag{25}$$

Inserting (25) into (24) gives

$$y^* = h(\hat{x}) + \delta \frac{K^T \nabla h_S(\hat{x})}{\|K^T \nabla h_S(\hat{x})\|_2}. \tag{26}$$

■

### B. Proof of Corollary 2

*Proof:* First, note that the constraint $\|y - h(\hat{x})\|_\infty \leq \delta$ can be split into the linear inequalities $y - h(\hat{x}) \leq \delta$ and $-y + h(\hat{x}) \leq \delta$. The KKT-conditions for (12) with the $\infty$-norm constraint are then given by

$$K^T \nabla h_S(\hat{x}) = -\lambda^+ + \lambda^-, \tag{27a}$$

$$y - h(\hat{x}) \leq \delta, \quad -y + h(\hat{x}) \leq \delta \tag{27b}$$

$$\lambda^+ \geq 0, \quad \lambda^- \geq 0 \tag{27c}$$

$$[\lambda^+]_i [\delta - h(\hat{x}) + y]_i = 0, \quad [\lambda^-]_i [-\delta + h(\hat{x}) - y]_i. \tag{27d}$$

If $[K^T \nabla h_S(\hat{x})]_i > 0$, we have that $[\lambda^+]_i > 0$ for the stationarity condition in (27a) to hold. Similarly, $[K^T \nabla h_S(\hat{x})]_i < 0$ requires $[\lambda^-] > 0$. This together with the complementary conditions (27d) gives

$$[y]_i = \begin{cases} h(\hat{x}) + \delta, & \text{if } [K^T \nabla h_S(\hat{x})]_i > 0, \\ h(\hat{x}) - \delta, & \text{if } [K^T \nabla h_S(\hat{x})]_i < 0, \end{cases} \tag{28}$$

which can be compactly written as

$$y = h(\hat{x}) + \delta \operatorname{sgn}(K^T \nabla h_S(\hat{x})). \tag{29}$$

■

### C. Proof of Theorem 4

*Proof:* First we expand the expression for $\dot{h}(\hat{x})$ and insert $y = y^a(\hat{x})$ from (15), which gives

$$\begin{aligned} \dot{h}_S(\hat{x}) &= \nabla h_S(\hat{x}) \dot{\hat{x}} \\ &= \nabla h_S(\hat{x})^T \left( f(\hat{x}, u) + K\left(y^a(\hat{x}) - h(\hat{x})\right)\right) \\ &= \nabla h_S(\hat{x})^T f(\hat{x}, u) + \nabla h_S(\hat{x})^T K \left(y^a(\hat{x}) - h(\hat{x})\right). \end{aligned}$$

Next, we intend to rewrite the second term $\nabla h_S(\hat{x})^T K \left(y^a(\hat{x}) - h(\hat{x})\right)$. From the definition of $y^a(\hat{x})$ in (12) we have

$$\begin{aligned} \nabla h_S(\hat{x})^T K y^a(\hat{x}) &= \max_{y : \|y - h(\hat{x})\| \leq \delta} \nabla h_S(\hat{x})^T K y \\ &= \max_{\tilde{y} : \|\tilde{y}\| \leq 1} \nabla h_S(\hat{x})^T K (\delta \tilde{y} - h(\hat{x})) \\ &= \max_{\tilde{y} : \|\tilde{y}\| \leq 1} \delta \nabla h_S^T K \tilde{y} - \nabla h_S^T K h(\hat{x}) \\ &= \|\delta K^T \nabla h_S(\hat{x})\|_* - \nabla h_S(\hat{x})^T K h(\hat{x}), \end{aligned}$$

where the variable change $\tilde{y} = \frac{1}{\delta}(y - h(\hat{x}))$ has been made in the second equality and the definition of $\|\cdot\|_*$ has been used in the last equality. Equivalently, we then have

$$\nabla h_S(\hat{x})^T K (y^a(\hat{x}) - h(\hat{x})) = \|\delta K^T \nabla h_S(\hat{x})\|_*. \tag{30}$$

Inserting this in the expression of $\dot{h}_S(\hat{x})$ gives the desired result. ■