# Data-Driven and Stealthy Deactivation of Safety Filters

**Daniel Arnström**                                                   DANIEL.ARNSTROM@IT.UU.SE
*Department of Information Technology, Uppsala University*
**André M.H. Teixeira**                                               ANDRE.TEIXEIRA@IT.UU.SE
*Department of Information Technology, Uppsala University*

**Editors:** N. Ozay, L. Balzano, D. Panagou, A. Abate

## Abstract

Safety filters ensure that control actions that are executed are always safe, no matter the controller in question. Previous work has proposed a simple and stealthy false-data injection attack for deactivating such safety filters. This attack injects false sensor measurements to bias state estimates toward the interior of a safety region, making the safety filter accept unsafe control actions. The attack does, however, require the adversary to know the dynamics of the system, the safety region used in the safety filter, and the observer gain. In this work we relax these requirements and show how a similar data-injection attack can be performed when the adversary only observes the input and output of the observer that is used by the safety filter, without any a priori knowledge about the system dynamics, safety region, or observer gain. In particular, the adversary uses the observed data to identify a state-space model that describes the observer dynamics, and then approximates a safety region in the identified embedding. We exemplify the data-driven attack on an inverted pendulum, where we show how the attack can make the system leave a safe set, even when a safety filter is supposed to stop this from happening.

## 1. Introduction

While learning-based controllers can improve performance over classical controllers (Coulson et al., 2019; Dörfler, 2023), they are seldom used in safety-critical applications due to their lack of *safety guarantees*. So-called *safety filters* (Wabersich et al., 2023; Tomlin et al., 2003; Ames et al., 2017; Wabersich and Zeilinger, 2018; Hobbs et al., 2023) can, in a modular fashion, augment any such unsafe learning-based controller with safety guarantees. 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. Since such filters separate safety from performance, any controller from the plethora of data-driven and learning-based controllers can be combined with a safety filter to give good performance together with safety guarantees.
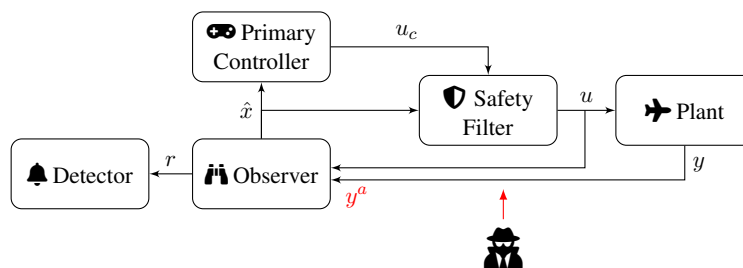


Figure 1: Overview of the system architecture considered in Arnström and Teixeira (2024) and in this paper. The safety filter produces a safe control action $u$ given a desired control $u_c$ and estimated state $\hat{x}$. An adversary tries to deactivate this filter through false-data injections on the communication channel between the sensors and the observer by replacing the true measurement $y$ with a synthetic measurement $y^a$.

At the same time, cyber-physical systems (CPSs) are becoming more prevalent and enable advanced control systems that are efficient and resilient (Dibaji et al., 2019). CPSs potential comes from their integration of communication, computation, and control technologies. The cyber component of CPSs do, however, open up for new vulnerabilities in the form of cyber attacks (Teixeira et al., 2015)(Annaswamy et al., 2023, §4.C). Since safety filters are the last layer before a control command is applied, they are prime targets for cyber attacks; moreover, if such an attack successfully compromises a safety-critical system, the consequence can be severe (Annaswamy et al., 2023, §4.B).

In this paper we consider cyber attacks that target safety filters. In particular, we consider an 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. The same attack scenario has been considered in Arnström and Teixeira (2024), but here we consider the far more realistic scenario when the adversary has less information. Specifically, in contrast to Arnström and Teixeira (2024), we assume that the adversary: $(i)$ does not know the dynamics of the system; $(ii)$ does not know the safety set that the safety filter uses; $(iii)$ does not know the observer gain. Instead, the adversary learns all of these components by just observing data from the observer. In addition, the attack framework considered herein also applies to safety filters that are not based on control barrier functions (CBFs), while the work in Arnström and Teixeira (2024) only considered CBF-based safety filters.

**Notation** Given a set $\mathcal{S} \triangleq \{s : g(s) \geq 0\}$, defined as the superlevel set of a function $g : \mathbb{R}^n \to \mathbb{R}^m$, we let its boundary be defined as $\partial S \triangleq \{s : g(s) = 0\}$ and its interior be defined as $\text{int } S \triangleq \{s : g(s) > 0\}$.

## 2. Problem formulation

In this paper we consider dynamical systems of the form

$$x_{k+1} = f(x_k, u_k), \tag{1}$$

with state $x \in \mathbb{R}^{n_x}$, control $u \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, and the dynamics $f : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. Specifically, we are interested in the *safety* of such systems. A system is safe if the states remain within a set $\mathcal{X} \subseteq \mathbb{R}^{n_x}$, where $\mathcal{X}$ encodes safety requirements. We formalize the notion of safety as follows

**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 $x_k \in \mathcal{X}$ and $u_k \in \mathcal{U}$ for all $k \geq 0$.*

Moreover, the system of interest generates measurements $y \in \mathbb{R}^{n_y}$ according to

$$y_k = h(x_k) + e_k, \tag{2}$$

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

To control the system an estimate $\hat{x}$ of the state $x$ is needed. To this end, the controller uses an observer of the form

$$\hat{x}_{k+1} = \hat{f}(\hat{x}_k, u_k, y_k), \qquad \hat{y}_k = h(\hat{x}_k), \tag{3}$$

with the goal of making the state estimate $\hat{x} \approx x$. A common form of the observer is

$$\hat{f}(x, u, y) = f(x, u) + K(x)(y - h(x)), \tag{4}$$

where $K : \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_x \times n_y}$ is an observer gain. This form covers, for example, classical (Kalman, 1960), extended (Jazwinski, 2007), and unscented (Julier and Uhlmann, 2004) Kalman filters.

Furthermore, the controller uses an anomaly detector (Murguia and Ruths, 2019) that triggers an alarm if the measurement $y$ is too different from the predicted measurement $\hat{y}$. That is, we assume that the detector triggers an alarm if the residual $r \triangleq y - \hat{y} = y - h(\hat{x})$ is large, e.g., in terms of a cumulative sum (Kurt et al., 2018) or a $\chi^2$-test (Brumback and Srinath, 1987). In particular, we consider anomaly detectors that trigger an alarm if

$$\|r\| > \delta, \tag{5}$$

where $\| \cdot \|$ is some norm and $\delta > 0$ is a threshold that determines the sensitivity of the detector.

Finally, the controller uses a *safety filter* that ensures that states remain within a safe set $\mathcal{S} \subseteq \mathcal{X}$. This safe set is defined as the super-level set of a function $h_S$:

$$\mathcal{S} \triangleq \{x : h_S(x) \geq 0\}. \tag{6}$$

The set of control actions that retain the state in $S$ is defined as $\mathcal{U}_S(x) \triangleq \{u \in \mathcal{U} : f(x, u) \in \mathcal{S}\}$, and the corresponding safety filter is given by

$$u = \underset{u \in \mathcal{U}_S(x)}{\arg \min} \|u - u_c\|, \tag{7}$$

where the desired control $u_c$ is projected onto the safe controls $\mathcal{U}_S$ to produce the actual control $u$ that is applied to the system. Safety filters of the form (7) can be based on, for example, control barrier functions (Ames et al., 2017), predictive safety filters (Wabersich and Zeilinger, 2021), or Hamilton-Jacobi reachability (Bansal et al., 2017). Excellent surveys on safety filters are given in Wabersich et al. (2023) and Hobbs et al. (2023).

### 2.1. Adversary model

As mentioned in the introduction, and visualized in Figure 1, we consider a scenario when an adversary performs a false-data injection attack on the measurement channel. Formally, we assume that the adversary has access to the following vulnerability:

**Assumption 1 (Observe and modify measurements)** *The adversary can observe and modify $y$.*

The ultimate objective of the adversary is to make the system state leave the safe set, i.e., to make $x \notin \mathcal{S}$. If the observer in (3) perfectly estimates the state, that is, if $\hat{x} = x$, this would be impossible due to the safety filter. The adversary's aim is, hence, to bias the state estimate $\hat{x}$ to give a false sense of safety, which could allow for unsafe control actions to be let through the safety filter. This is formalized with the concept of *deactivation*

**Definition 2 (Deactivation)** *For a safety filter of the form (7), the state estimate $\hat{x}$ is said to deactivate the safety filter at state $x$ if $\exists u \in \mathcal{U}_S(\hat{x})$ such that $u \notin \mathcal{U}_S(x)$.*

Briefly put, then, the adversary wants to modify the measurement $y$ such that the state estimate $\hat{x}$ deactivates the safety filter at the actual state.

If, in addition to Assumption 1, the adversary knows the state estimate $\hat{x}$, the observer dynamics $\hat{f}$, and the measurement function $h$, they can, as is shown in Arnström and Teixeira (2024), perform a false-data injection attack by replacing the true measurement $y$ with

$$y^a = \arg\max_{y^a} \ \nabla h_S(\hat{x})^T \left( \frac{\partial \hat{f}}{\partial y}\Big|_{x=\hat{x}} y^a \right) \text{ subject to } \|y^a - h(\hat{x})\| \leq \delta. \tag{8}$$

Intuitively, the objective of (8) is a local change in distance between $\hat{x}$ and the edge $\partial S \triangleq \{x : h_S(x) = 0\}$ of the safe set $S$; the constraint, in turn, ensures that the detector is not triggered, i.e., that the adversary remains stealthy. Thus, the attack will stealthily bias the state estimates towards the interior of the safe set, which makes the safety filter overestimate the safety of the system, and might therefore let through unsafe control actions to the plant. For some norms, the optimization problem in (8) has a closed-form solutions (cf. Corollary 1 and 2 in Arnström and Teixeira (2024).)

**Remark 3 (Independence of $\frac{\partial \hat{f}}{\partial y}$ with respect to $y$ and $u$)** *Note that the term $\frac{\partial \hat{f}}{\partial y}\big|_{x=\hat{x}}$ in (8) implicitly assumes that the partial derivative of $\hat{f}$ with respect to $y$ is independent of $y$ and $u$. This is, for example, the case for observers of the common form (4).*

In contrast to Arnström and Teixeira (2024), this work considers the case when the observer dynamics in (3) and the safe set in (6) are *unknown* to the adversary, which make the *ideal* attack in (8) impossible to implement. Instead, we assume that the adversary can observe the input $(y, u)$ and output $\hat{y}$ of the observer, which motivates appending the following assumptions to Assumption 1.

**Assumption 2 (Observe control actions)** *The adversary can observe $u$.*

**Assumption 3 (Observe predicted measurement)** *The adversary can observe $\hat{y}$.*

Assumption 3 can be satisfied in practice through several means. It is, for example, equivalent to observing the residual $r$ that is sent to the anomaly detector, since $\hat{y} = y - r$, with $y$ already being known from Assumption 1. Another, stronger, assumption that implies Assumption 3 is that the adversary can directly observe the state estimate $\hat{x}$ (and knows the measurement function $h$.) Furthermore, if the adversary has some knowledge about the observer and controller structures they might perfectly estimate their internal states, which can give information about $r$, $\hat{y}$, and $\hat{x}$ (Umsonst and Sandberg, 2021; Umsonst et al., 2019).

In summary, the objective of the adversary, and the main problem formulation considered in this paper, is as follows:

> **Problem 1**: By observing the inputs $u$ and $y$ and the output $\hat{y}$ from an observer of the form (3), modify the measurement $y$ such that the state estimate $\hat{x}$ deactivates the safety filters of the form (7) at the true state $x$.

## 3. Data-driven attack

The main idea behind the adversary's attack is to replace the model-based components in (8) by surrogates that are learned using data from the system. For example, the adversary learns a mapping $\tilde{f}$ that predicts the observer output $\hat{y}$ given its input $u$ and $y$ to substitute the unknown observer dynamics $\hat{f}$. Furthermore, the collected data is used to identify a safe set similar to (6). Since the states used by the observer in (3) is assumed to be unknown, the learned state-space model will have states $z$ that do not necessarily correspond to $\hat{x}$. As a result, the safe set is identified in the *latent* space where $z$ resides, and not the nominal state space where $\hat{x}$ resides.

The learned model and safe set are then used to perform a false-data injection similar to (8), where the unknown mappings $\hat{f}, h$, and $h_S$ are replaced with learned mappings $\tilde{f}, \tilde{h}$, and $\tilde{h}_S$. To perform such an attack, the adversary also makes use of the learned dynamics and online data to update the latent state $z$ (instead of, as in the ideal attack in (8), updating $\hat{x}$.)

The attack can, hence, be split into an *online* phase and an *offline* phase, which will now be described in detail in Section 3.1 and 3.2, respectively.

### 3.1. Offline phase

In the initial *offline* phase the adversary collects data by observing $u, y$, and $\hat{y}$, which is used to: $(i)$ identify a state-space model similar to (3); $(ii)$ estimate a safe set similar to (6) in latent space; $(iii)$ lower-bound the detector threshold $\delta$; Details on $(i)$, $(ii)$, and $(iii)$ are given in Section 3.1.1, 3.1.2, and 3.1.3 respectively.
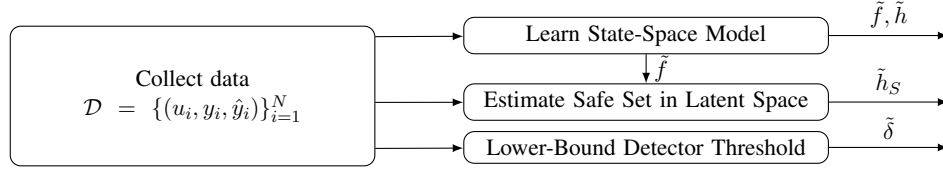


Figure 2: The pipeline for the offline phase of the data-driven attack.

#### 3.1.1. SYSTEM IDENTIFICATION

Assume that the adversary has observed the data sequence $\mathcal{D} = \{(u_i, y_i, \hat{y}_i)\}_i$ from (3). To ensure that the detector is not triggered when the measurement $y$ is modified, the adversary needs to have an idea where the controller's predicted measurement will be, i.e., it needs a predictive model that outputs $\tilde{y}$ such that $\tilde{y} \approx \hat{y}$. Given such a model, the adversary can stealthily modify the measurement $y$ by selecting it such that $\|y - \tilde{y}\| \leq \delta$ (if the threshold $\delta$ is known, see Section 3.1.3.)

The adversary is, hence, interested in learning a state-space model with a *latent state* $z \in \mathbb{R}^{n_z}$ of the form

$$z_{k+1} = \tilde{f}(z_k, u_k, y_k), \qquad \tilde{y}_k = \tilde{h}(z_k). \tag{9}$$

such that $\tilde{y}_k \approx \hat{y}_k$ for all the data in $\mathcal{D}$. Note that $\tilde{y} \approx \hat{y}$ does not necessarily imply that the latent state $z$ is the same as the estimated state $\hat{x}$, which complicates the safe set estimation (see Section 3.1.2.)

There are several approaches to identifying models of the form (9), including subspace methods (Van Overschee and De Moor, 1994), Maximum Likelihood (Schön et al., 2011), deep learning

5

(Gedon et al., 2021), variational inference (Doerr et al., 2018), and direct nonlinear programming (Bemporad, 2024), to name a few.

**Remark 4 (Motivation for state-space models)** *A predictive model that estimates $\hat{y}$ can be of other forms than the state-space model in (9) (For example, in the form of an ARX model, or from any other classical model classes in system identification (Ljung, 1999)) . Having a state-space model will, however, be essential for the estimation of a safe set that is described in Section 3.1.2.*

Typically, we have that $\varepsilon_k \triangleq \hat{y}_k - \tilde{y}_k \neq 0$, since the $\tilde{f}$ never completely match $\hat{f}$ due to, for example, measurement and process noise. To ensure that $\tilde{y}$ remains close to $\hat{y}$ (i.e., to remain stealthy), one can amend the state-space model in (9) with an observer, resulting in the *forward innovation model* (Van Overschee and De Moor, 2012, Ch. 3)

$$z_{k+1} = \tilde{f}(z_k, u_k, y_k) + \tilde{K}(z_k)\varepsilon_k, \qquad \hat{y}_k = \tilde{h}(z_k) + \varepsilon_k, \tag{10}$$

where the gain $\tilde{K}(\cdot)$ also needs to be identified. This enables feedback from $\hat{y}$ during the online updates to ensure that $\hat{y} \approx \tilde{h}(z)$. For convenience, we introduce the compact notation $\tilde{f}(z, u, y, \hat{y}) \triangleq \tilde{f}(z, u, y) + \tilde{K}(z)(\tilde{h}(z) - \hat{y})$.

A well-studied (see, .e.g., Van Overschee and De Moor (2012)) special case of identifying state-space models of the form (10) is when $\tilde{f}$ and $\tilde{h}$ are linear, which gives the linear state-space model

$$z_{k+1} = \tilde{A}z_k + \begin{bmatrix} \tilde{B}_u & \tilde{B}_y \end{bmatrix} \begin{bmatrix} u_k \\ y_k \end{bmatrix} + \tilde{K}\varepsilon_k, \qquad \hat{y}_k = \tilde{C}z_k + \varepsilon_k. \tag{11}$$

Learning $\tilde{f}$ and $\tilde{h}$, hence, boils down to identifying the matrices $\tilde{A}, \tilde{B}_u, \tilde{B}_y, \tilde{K}$, and $\tilde{C}$. Note that we have that $\frac{\partial \tilde{f}}{\partial y} = \tilde{B}_y$, which is independent of both $u$ and $y$ in accordance with Remark 3.

**Remark 5 (Inherent structure of $\hat{f}$)** *Since we are are trying to identify the dynamics of an observer, which is often based on simplified models that make the observer implementation manageable, the underlying dynamics is more structured than that of a "natural" system for which system identification is often applied.*

### 3.1.2. ESTIMATING SAFE SET

To deactivate the safe filter, the adversary wants to bias state estimates toward the interior of the safe set $\mathcal{S}$. Therefore, the adversary needs some information about the safe set $\mathcal{S}$. A complication is, however, that the adversary does not even know about the state space where $x$ resides, so "estimating $\mathcal{S}$" is not a well-defined problem given the available information. To address this complication, the adversary instead estimates a set in the space where the latent state $z$, from an identified model of the form (9), resides. To connect these two state spaces, and the corresponding dynamics of $\hat{x}$ and $z$, we introduce the notion of topological equivalence, similar to what is introduced in Jongeneel et al. (2021).

**Definition 6 (Topological equivalence)** *The dynamics $\hat{f}$ in (3) and $\tilde{f}$ (9) are topologically equivalent if for all $u \in \mathbb{R}^{n_u}$ and $y \in \mathbb{R}^{n_x}$ there exists a homeomorphism $\mathcal{T} : \mathcal{X} \to \mathcal{Z}$ such that $\tilde{f}(\cdot, u, y) \circ \mathcal{T} = \mathcal{T} \circ \hat{f}(\cdot, u, y)$, where $\circ$ denotes composition of functions.*

Basic topology gives the following results:

**Proposition 7** *If $\tilde{f}$ and $\hat{f}$ are topologically equivalent through the homeomorphism $\mathcal{T}$, then $\partial \mathcal{T}(\mathcal{S}) = \mathcal{T}(\partial \mathcal{S})$ and $\mathrm{int}\, \mathcal{T}(\mathcal{S}) = \mathcal{T}(\mathrm{int}\, \mathcal{S})$.*

This shows that points in the interior of the safe set $\mathcal{S}$ directly map onto corresponding points in the interior of $\mathcal{T}(\mathcal{S})$. Consequently, if we bias the latent state $z$ into the interior of $\mathcal{T}(\mathcal{S})$, this will bias the state estimate $\hat{x}$ into the interior of $\mathcal{S}$. This connection is made more explicit in the following lemma.

**Lemma 8 (Safety of latent states)** *Consider any sequence of inputs $\{u_i\}_{i=1}^N$ and outputs $\{y_i\}_{i=0}^N$ such that $u_i \in \mathcal{U}$ for all $i$. Assume that, subject to these sequences, the system (3) produces the trajectory $\{\hat{x}_i\}_{i=0}^N$ that is safe, i.e., $\hat{x}_i \in \mathcal{S}$ for all $i$. Similarly, assume and that system (9) produces the trajectory $\{z_i\}_{i=0}^N$ subject to the same input and output sequence, and with the initial state $z_0 = \mathcal{T}(\hat{x}_0)$. Then $z_i \in \mathcal{T}(\mathcal{S})$ if $f$ and $\tilde{f}$ are topologically equivalent through $\mathcal{T}$.*

**Proof** From Definition 6 we have that $\tilde{f}(\mathcal{T}(\hat{x}_k), u_k, y_k) = \mathcal{T}\left(\hat{f}(\hat{x}_k, u_k, y_k)\right) = \mathcal{T}(\hat{x}_{k+1})$. If it holds that $z_k = \mathcal{T}(\hat{x}_k)$ then we get that $z_{k+1} = \tilde{f}(z_k, u_k, y_k) = \mathcal{T}(\hat{f}(\hat{x}_k, u_k, y_k)) = \mathcal{T}(\hat{x}_{k+1})$. Since the base case $z_0 = \mathcal{T}(\hat{x}_0)$ holds, it follows by induction that $z_i = \mathcal{T}(\hat{x}_i), \quad \forall i = 0, \ldots, N$. This together with the assumption that $\hat{x}_i \in \mathcal{S}$ gives the desired result that $z_i \in \mathcal{T}(\mathcal{S})$. ∎

**Linear special case** When $\hat{f}$ is linear, the model class in (11) contains several models that are topologically equivalent to $\hat{f}$ through a homeomorphism $\mathcal{T}$ that is linear ($\mathcal{T}$ is better known as a *similarity transform* in this context.) Moreover, it is well known that subspace methods such as N4SID (Van Overschee and De Moor, 1994) identify one of these topologically equivalent models (cf., e.g., Chapter 2 in Van Overschee and De Moor (2012).)

**Remark 9 (Linear observers)** *We want to emphasize that, related to Remark 5, the special case of $\hat{f}$ being linear is quite common in practice since $\hat{f}$ is the observer dynamics that is designed by a user. For straightforward synthesis, linearized models are commonly used for observer design. For example, this is the case when a classic Kalman filter is used.*

Lemma 8 implies that latent trajectories give information about the underlying safe set $\mathcal{S}$ if the system is assumed to be safe (i.e., if the safety filter is deployed). Since, again, both $\mathcal{S}$ and $\mathcal{X}$ are unknown, we will leverage Lemma 8 to approximate $\mathcal{T}(\mathcal{S})$ instead. Let $\mathcal{Z}$ be a set that covers the latent trajectory $\{z_i\}_{i=1}^N$ generated by forward simulation with the data in $\mathcal{D}$. Moreover, let $\mathcal{Z}$ have an explicit representations as a super-level sets of the form $\mathcal{Z} = \{z : \tilde{h}_S(z) \geq 0\}$. While any cover of this form suffices for the attacker, two common covers are the convex hull of the trajectory, or a minimum covering ellipse.

### 3.1.3. ESTIMATING DETECTOR THRESHOLD

Even if the adversary can predict exactly where the observer's predicted measurement $\hat{y}$ is, it also needs to know the detector threshold $\delta > 0$ to be able to know how much it can modify the measurement $y$. To ensure that modifications of $y$ are stealthy (i.e., that $\|y - \hat{y}\| < \delta$) the adversary, hence, needs to at least lower-bound the threshold $\delta$.

If we assume that the detector was not triggered while the data $\mathcal{D} = \{(u_i, y_i, \hat{y}_i)\}_i$ was collected,

$$\|y - \hat{y}\| \leq \delta, \quad \forall (u, y, \hat{y}) \in \mathcal{D}. \tag{12}$$

Hence, one way to find a lower-bound $\tilde{\delta}$ of $\delta$ is as $\tilde{\delta} = \gamma \max_{(u,y,\hat{y}) \in \mathcal{D}} \|y - \hat{y}\|$, for some constant $\gamma \in (0, 1]$; a smaller $\gamma$ reduce the risk of the adversary being detected, but also restricts how much the measurement $y$ can be modified. We formalize this in the following lemma the directly follows from (12).

**Lemma 10** *Assume that the system uses a detector of the form* (5) *and that the detector was not triggered when collecting the data set $\mathcal{D}$. Then $\tilde{\delta}$ lower-bounds the detector threshold $\delta$(i.e., $\tilde{\delta} \leq \delta$.)*

### 3.2. Online phase

In the online phase of the attack, the adversary performs a false-data attack by injecting false measurements $y^a$ according to the policy

$$
y^a(z_k) = \arg\max_{y^a} \ \nabla \tilde{h}_S(z_k)^T \left( \frac{\partial \tilde{f}}{\partial y}\bigg|_{z=z_k} y^a \right) \text{ subject to } \|y^a - \tilde{h}(z_k)\| \leq \tilde{\delta}, \tag{13}
$$

which is equivalent to the policy in (8), but with an estimated system dynamics ($\tilde{f}$ and $\tilde{h}$), an estimated safe set ($\tilde{h}_S$), and a lower-bound $\tilde{\delta}$ of $\delta$. Moreover, the adversary updates the latent state $z$ by using the identified model of the form (10) with $u$, $y^a$ and $\hat{y}$, resulting in the update

$$
z_{k+1} = \tilde{f}(z_k, u_k, y^a(z_k)) + \tilde{K}(z_k)(\tilde{h}(z_k) - \hat{y}_k) \triangleq \tilde{f}(z_k, u_k, y^a(z_k), \hat{y}_k). \tag{14}
$$

The attack is summarized in Algorithm 1, and we will now make its relationship to the deactivation of safety filters more explicit.

#### 3.2.1. RELATIONSHIP BETWEEN ALGORITHM 1 AND DEACTIVATION OF SAFETY FILTERS

As mentioned before, the high-level idea behind the attack is to bias state estimates $\hat{x}$ toward the interior of $\mathcal{S}$. This is based on the set of safe control actions $\mathcal{U}_S(x)$ at $x$ often having the following property: $\mathcal{U}_S(x) \subset \mathcal{U}_S(\tilde{x})$ if $x$ is closer to $\partial \mathcal{S}$ than $\tilde{x}$, i.e., more control actions are safe further away from the border of the safe set. So if $x$ is closer to $\partial S$ than $\hat{x}$, we typically have that $\mathcal{U}_S(x) \subset \mathcal{U}_S(\hat{x})$, which, from Definition 2, would deactivate the safety filter.

To relate this to the latent state $z$, Proposition 7 implies that biasing $\hat{x}$ into the interior of $\mathcal{S}$ is equivalent to biasing the latent state $z$ into the interior of $\mathcal{T}(\mathcal{S})$ if $\tilde{f}$ is topologically equivalent to $\hat{f}$ through the homeomorphism $\mathcal{T}$. Since the attacker does not have access to $\mathcal{T}(S)$ directly, they approximate it with $\mathcal{Z} = \{z : \tilde{h}_S(z) \geq 0\}$, which is a cover of points in $\mathcal{T}(S)$ (from Lemma 8).

To complete the argument, the measurement $y^a$ should bias the latent state $z$ towards the interior of $\mathcal{Z}$. Biasing $z$ towards the interior of $\mathcal{Z}$ means increasing the corresponding safety margin $\tilde{h}_S$. The measurement $y$ indirectly affects $\tilde{h}_S$ through the time update $z_{k+1} = \tilde{f}(z_k, u_k, y)$. How a change in $y$ locally affects $\tilde{h}_S$ can be seen through the gradient $\nabla_y \left( \tilde{h}_S \circ \tilde{f}(z_k, u_k, y) \right)$. The chain rule then gives

$$
\nabla_y \left( \tilde{h}_S \circ \tilde{f}(z_k, u_k, y) \right) = \nabla \tilde{h}_S(z)^T \frac{\partial \tilde{f}}{\partial y}\bigg|_{z=z_k}, \tag{15}
$$

which is exactly the linear term in the objective of (13). In other words, the objective of (13) aligns the false measurement $y^a$ in the direction that locally increases $\tilde{h}_S$ the most.

---
**Algorithm 1** Data-driven false-data injection attack to deactivate safety filters.

---
**Input:** $\tilde{f}, \tilde{h}, \tilde{h}_S, \tilde{\delta}$ (from offline phase) and $z_0$
$k \leftarrow 0; \quad z \leftarrow z_0$
**repeat**
    observe $u_k, y_k$ and $\hat{y}_k$
    replace $y_k$ with $y^a$ by solving (13) for $z_k$
    $z_{k+1} \leftarrow \tilde{f}(z_k, u_k, y_k, \hat{y}_k); \quad k \leftarrow k+1$
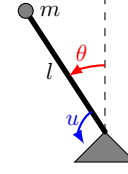
---



Figure 3: Inverted pendulum with mass $m$ and length $l$. The angle $\theta$ is controlled by applying the torque $u$.

## 4. Numerical Experiments

We exemplify the proposed attack on an inverted pendulum system that was also considered in Wabersich et al. (2023) and Alan et al. (2023), visualized in Figure 3. The dynamics of the system is $\frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \dot{\theta} \\ \frac{g}{l} \sin \theta \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix} u$, with parameter values $m = 2$ kg, $l = 1$ m , and $g = 10$ m/s$^2$ (the same as in Wabersich et al. (2023); Alan et al. (2023)). Moreover, the applied torque $u$ is limited between $\pm 3$ Nm, and the angle $\theta$ is constrained to be between $\pm 0.25$ radians out of safety reasons. To ensure that these constraints are satisfied, a safety filter that keeps the state in a safe set $\mathcal{S} = \{(\theta, \dot{\theta}) : h_S(\theta, \dot{\theta}) \geq 0\}$ is used, where $h_S(\theta, \dot{\theta})$ is a control barrier function (Ames et al., 2017). Specifically, we use the control barrier function $h_S(\theta, \dot{\theta}) = 1 - 16\theta^2 - 8\theta\dot{\theta} - 4\dot{\theta}^2$ that is used in Alan et al. (2023). The corresponding optimization problems in (7) are quadratic programs, which are solved using the quadratic programming solver `DAQP` (Arnström et al., 2022). Code for all experiments is available at `https://github.com/darnstrom/ddsd-sf`.

Only the angle $\theta$ is measured, with measurement noise that is normally distributed with zero-mean and standard deviation $10^{-3}$ radians. The controller estimates $\dot{\theta}$ with an observer of the form (3), where the gain $K$ is the static Kalman gain for a linearized model around the origin.

For the attack, the adversary first collects data from the system by observing $(u, y, \hat{y})$ for 100 seconds (at a sample rate of 100 Hz). The collected data is shown in Figure 4. Based on this data, the adversary identifies a linear state-space model of the form (11), with dimension $n_z = 2$, using `N4SID` (Van Overschee and De Moor, 1994).
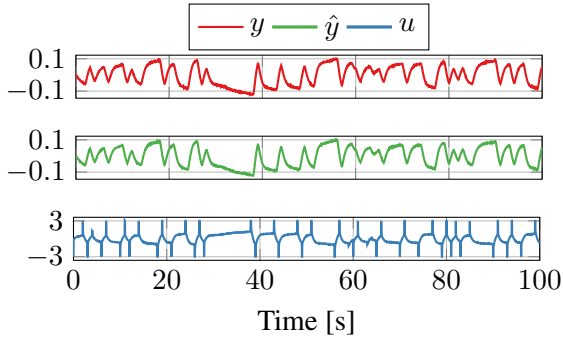


Figure 4: Collected data $\mathcal{D}$ from the system, used for learning $\tilde{f}, \tilde{h}, \tilde{h}_S$ and $\tilde{\delta}$.
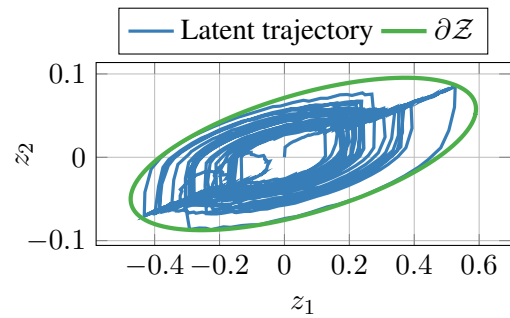


Figure 5: Latent trajectory of simulating the latent dynamics $\tilde{f}$ in (9) with the collected data $\mathcal{D}$.
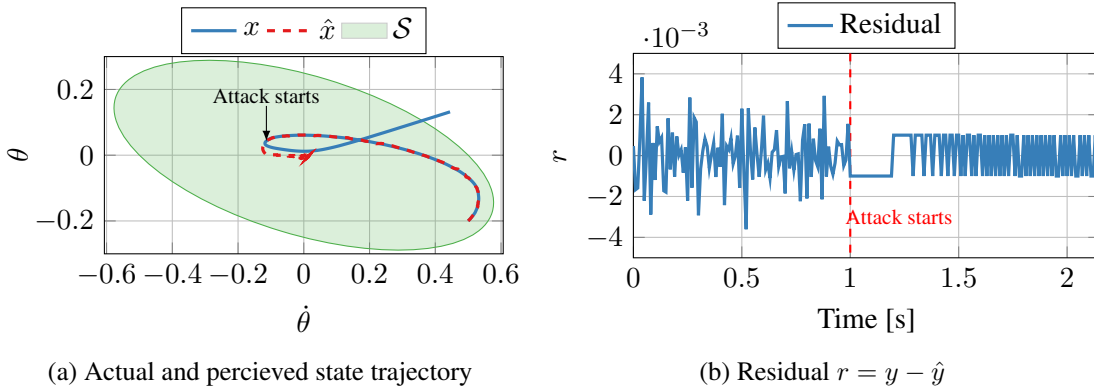
(a) Actual and percieved state trajectory

(b) Residual $r = y - \hat{y}$

Figure 6: The actual and perceived state trajectories, and the corresponding residual $r$, when the false-data attack in Algorithm 1 is initiated at $t = 1$. Consequently, the state leaves the safe set.

Using the identified state-space model, the adversary reuses the observed data $(u, y, \hat{y})$ and simulate the system, producing the latent trajectory $\{z_i\}$ shown in Figure 5. An approximate safe set $\mathcal{Z} = \{z : \tilde{h}_S(z) \geq 0\}$ in latent space is then determined by finding a minimally covering ellipse for the entire latent trajectory, also shown in Figure 5.

Next, the attacker performs the online phase of the attack, which uses the identified $\tilde{f}, \tilde{h}$ and $\tilde{h}_S$ to attack the system with Algorithm 1. During the attack, the controller that produces desired control actions is an exact-linearized LQR controller given by $u_c(\theta, \hat{\theta}) = -mlg \sin(\theta) - \begin{bmatrix} 1.5 & 1.5 \end{bmatrix} \begin{bmatrix} \theta \\ \hat{\theta} \end{bmatrix}$ The attack scenario starts with $\dot{\theta} = 0.5$ and $\theta = -0.2$ at $t = 0$. To remain stealthy, the adversary does not initiate the attack directly at $t = 0$; this is due to the latent state at $t = 0$ being unknown, which, hence, requires some time for $z$ to converge such that $\tilde{h}(z) \approx \hat{y}$. The adversary initiates the attack at $t = 1$ second, which, as in shown in Figure 6a, drives the state $x$ outside the safe set $\mathcal{S}$ while the state estimates $\hat{x}$ remain inside $\mathcal{S}$, creating a false belief of safety. Moreover, the corresponding residuals used by the anomaly detector are shown Figure 6b, where one can see that their magnitudes do not exceed the magnitude of the measurement noise. Note, however, that there is a clear bias in the residual after the attack. This can be used to create an anomaly detector that specifically detects attacks that try to deactivate safety filters, complementing classical detectors such as (5). For details of how to construct such a detector, see Arnström and Teixeira (2024).

## 5. Conclusion

We have shown how a simple, data-driven, and stealthy false-data injection attack can deactivate safety filters. In contrast to the work in Arnström and Teixeira (2024), the adversary does not use any prior knowledge about the dynamics, safe set, or observer gain. The attack injects false sensor measurements to bias latent states to the interior of an estimated safety region in a latent embedding, which, in turn, biases state estimates to the interior of the actual safety region. This can make the safety filter let through unsafe control actions. The attack was shown to be able to successfully make a inverted pendulum system leave its safe set that a safety filter is supposed to keep invariant. Future work includes formally analysing the impact of model mismatch for the identified model, as well as relaxing even more of the assumptions (primarily the assumption on measuring $\hat{y}$.)

## Acknowledgments

## References

Anil Alan, Andrew J Taylor, Chaozhe R He, Aaron D Ames, and Gábor Orosz. Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Transactions on Control Systems Technology*, 31(6):2744–2759, 2023.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2017. doi: 10.1109/TAC.2016.2638961.

Anuradha M. Annaswamy, Karl H. Johansson, and George J. Pappas. *Control for Societal-scale Challenges: Road Map 2030*. IEEE Control Systems Society, May 2023.

Daniel Arnström and André M.H. Teixeira. Stealthy deactivation of safety filters. In *2024 European Control Conference (ECC)*, pages 3077–3082, 2024. doi: 10.23919/ECC64448.2024.10590978.

Daniel Arnström, Alberto Bemporad, and Daniel Axehill. A dual active-set solver for embedded quadratic programming using recursive $\text{LDL}^T$ updates. *IEEE Transactions on Automatic Control*, 67(8):4362–4369, 2022. doi: 10.1109/TAC.2022.3176430.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

Alberto Bemporad. Linear and nonlinear system identification under $\ell_1$-and group-Lasso regularization via L-BFGS-B. *arXiv preprint arXiv:2403.03827*, 2024.

B. Brumback and M. Srinath. A chi-square test for fault-detection in kalman filters. *IEEE Transactions on Automatic Control*, 32(6):552–554, 1987. doi: 10.1109/TAC.1987.1104658.

Jeremy Coulson, John Lygeros, and Florian Dörfler. Data-enabled predictive control: In the shallows of the DeePC. In *2019 18th European Control Conference (ECC)*, pages 307–312. IEEE, 2019.

Seyed Mehran Dibaji, Mohammad Pirani, David Bezalel Flamholz, Anuradha M Annaswamy, Karl Henrik Johansson, and Aranya Chakrabortty. A systems and control perspective of CPS security. *Annual reviews in control*, 47:394–411, 2019.

Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Trimpe Sebastian. Probabilistic recurrent state-space models. In *International conference on machine learning*, pages 1280–1289. PMLR, 2018.

Florian Dörfler. Data-driven control: Part two of two: Hot take: Why not go with models? *IEEE Control Systems Magazine*, 43(6):27–31, 2023. doi: 10.1109/MCS.2023.3310302.

Daniel Gedon, Niklas Wahlström, Thomas B Schön, and Lennart Ljung. Deep state space models for nonlinear system identification. *IFAC-PapersOnLine*, 54(7):481–486, 2021.

Kerianne L Hobbs, Mark L Mote, Matthew CL Abate, Samuel D Coogan, and Eric M Feron. Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems. *IEEE Control Systems Magazine*, 43(2):28–65, 2023.

Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

Wouter Jongeneel, Tobias Sutter, and Daniel Kuhn. Topological linear system identification via moderate deviations theory. *IEEE Control Systems Letters*, 6:307–312, 2021.

Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Mehmet Necip Kurt, Yasin Yılmaz, and Xiaodong Wang. Distributed quickest detection of cyber-attacks in smart grid. *IEEE Transactions on Information Forensics and Security*, 13(8):2015–2030, 2018. doi: 10.1109/TIFS.2018.2800908.

Lennart Ljung. *System identification (2nd ed.): theory for the user*. Prentice Hall PTR, USA, 1999. ISBN 0136566952.

Carlos Murguia and Justin Ruths. On model-based detectors for linear time-invariant stochastic systems under sensor attacks. *IET Control Theory & Applications*, 13(8):1051–1061, 2019.

Thomas B Schön, Adrian Wills, and Brett Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.

André Teixeira, Iman Shames, Henrik Sandberg, and Karl Henrik Johansson. A secure control framework for resource-limited adversaries. *Automatica*, 51:135–148, 2015.

Claire J Tomlin, Ian Mitchell, Alexandre M Bayen, and Meeko Oishi. Computational techniques for the verification of hybrid systems. *Proceedings of the IEEE*, 91(7):986–1001, 2003.

David Umsonst and Henrik Sandberg. On the confidentiality of controller states under sensor attacks. *Automatica*, 123:109329, 2021.

David Umsonst, Ehsan Nekouei, André Teixeira, and Henrik Sandberg. On the confidentiality of linear anomaly detector states. In *2019 American Control Conference (ACC)*, pages 397–403, 2019. doi: 10.23919/ACC.2019.8814731.

Peter Van Overschee and Bart De Moor. N4SID: Subspace algorithms for the identification of combined deterministic-stochastic systems. *Automatica*, 30(1):75–93, 1994. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(94)90230-5. Special issue on statistical signal processing and control.

Peter Van Overschee and BL0888 De Moor. *Subspace identification for linear systems: Theory—Implementation—Applications*. Springer Science & Business Media, 2012.

Kim P Wabersich and Melanie N Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

Kim P. Wabersich, Andrew J. Taylor, Jason J. Choi, Koushil Sreenath, Claire J. Tomlin, Aaron D. Ames, and Melanie N. Zeilinger. Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023. doi: 10.1109/MCS.2023.3291885.

Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.