Optimization-based attack against control systems with CUSUM-based anomaly detection

Gabriele Gualandi  
Mälardalen University, Sweden  
gabriele.gualandi@mdh.se

Martina Maggio  
Saarland University, Germany  
maggio@cs.uni-saarland.de

Alessandro Vittorio Papadopoulos  
Mälardalen University, Sweden  
alessandro.papadopoulos@mdh.se

Abstract—Security attacks on sensor data can deceive a control system and force the physical plant to reach an unwanted and potentially dangerous state. Therefore, attack detection mechanisms are employed in cyber-physical control systems to detect ongoing attacks, the most prominent one being a threshold-based anomaly detection method called CUSUM. Literature defines the maximum impact of stealth attacks on a plant as the maximum deviation in the plant’s state an undetectable attack can introduce and formulates it as the solution to an optimization problem. This paper proposes an optimization-based attack with different saturation models, and it investigates how the attack duration significantly affects the impact of the attack on the state of the control system. We show that more dangerous attacks can be discovered when allowing saturation of the control system actuators. The proposed approach is compared with the geometric attack, showing how longer attack durations can lead to a greater impact of the attack while keeping the attack stealthy.

I. INTRODUCTION

Cyber-physical systems realize critical infrastructure control (e.g., electric power, water resources), distributed robotics (telepresence, telemedicine, automated manufacturing), healthcare systems, assisted living, environmental control, traffic control, advanced automotive systems, unmanned vehicles, and more. Infamous cyber-attacks on nuclear centrifuges (Stuxnet in 2010) and power networks (Havex/Dragonfly in 2014) raised awareness on the vulnerability of control systems, and the risks for society. According to Kaspersky Lab, 41.2% of factories were attacked by malicious software at least once in the first half of 2018. Hence, many different companies tried to secure their industrial control systems [1], [2].

This led to the prominence of intrusion detection methods for cyber-physical systems [3], [4]. The detection process is usually based on the concept of anomaly, and it involves some form of verification through a model. A way to detect anomalies is to verify the controller calculations using redundancy, e.g., [5]. However, if sensors or actuators are attacked, redundant execution is not enough to guarantee detection, because the compromised component is not the execution of the control software. To detect anomalies at the interfaces (sensors and actuators), there is a need for physics-based detection [6]. The anomaly detection strategy, in this case, usually consists of comparing the signals produced and consumed by sensors and actuators with a physical system model, i.e., a digital twin that mimics the behavior of the physical process. In control terms, this component is often realized with a state estimator, that determines the expected state of the system and compares it with the state that is apparent via measurements. The difference between the signals of physical plant and those of the digital twin are called residuals. Due to the existence of unknown disturbances, model uncertainties, and numeric approximation, the detection process is necessarily probabilistic, and at run-time, a threshold is employed to discriminate the null hypothesis (absence of attacks) from the presence of an attack in progress.

Many researches on cyber-physical security use a stateless anomaly detector based on the current residual e.g., [7], [8]. Other approaches investigate stateful approaches, which sum up the residual, therefore, considering the history of the residual signal. Usually, these systems result in a Cumulative Sum (CUSUM)-based intrusion detection method [9], [10], [11].

Attacks that take into account the presence of anomaly detection strategies are called stealth, and they are designed to remain undetected [12]. This work focuses precisely on stealth sensor attacks or stealth attacks that consist of injecting false measurements in sensors. Many stealth attack functions have been experimented with [12]. However, the literature usually does not take into account implementation elements like the saturation of the control signal (i.e., physical limitations that prevent the control signal from exceeding maximum values or being below prescribed minimum values).

We devise a new optimisation-based method to deviate as much as possible the state of the control plant from its nominal value. An optimisation problem is solved once, to produce a vector of sensor attack values that can be injected into the system at every time step. We show that the way actuators’ saturations are modeled can produce significantly different results.

Related work: The literature explored simulated stealth attacks, based on specific attack functions and empirically evaluated the minimum time required to cause damage [13] or the maximum impact that can be caused [12]. The takeaway message is that an exponentially-shaped attack on the sensors, also called geometric attack, is the most
effective attack function because it is capable of introducing the most significant deviation from the nominal plant conditions. For example, stealth-by-design geometric attacks have been applied to a robot [14]. In the works mentioned above, there are two main limitations. First, the definition of stealthiness for an attack is limited to physical auto-regressive models, i.e., control systems that do not employ a state estimator. Second, considering only specific attack functions like the geometric attack may hinder more dangerous unknown attack functions.

Recent works like Urbina et al. [15], Umanston et al. [11], and Teixeira et al. [7] avoid considering specific attack functions and solve an optimization problem to obtain the “worst case” stealth attack, i.e., the one delivering functions and solve an optimization problem to obtain and Teixeira et al. [7] avoid considering specific attack functions like the geometric attack may hinder more effective attack function because it is capable of introducing the most significant deviation from the nominal plant conditions. For example, stealth-by-design geometric attacks have been applied to a robot [14]. In the works mentioned above, there are two main limitations. First, the definition of stealthiness for an attack is limited to physical auto-regressive models, i.e., control systems that do not employ a state estimator. Second, considering only specific attack functions like the geometric attack may hinder more dangerous unknown attack functions.

According to our threat model, an attacker forges \( a_k \) to divert the output \( y_k \) of the system from its desired value \( w_k \) to reach a dangerous state.

We assume that the plant is linear, time-invariant, controllable, and observable. The discrete-time dynamic equations of the plant under attack are

\[
\begin{align*}
x_{k+1} &= Ax_k + Bu_k \\
y_k &= Cx_k + Du_k
\end{align*}
\]

where \( A \in \mathbb{R}^n, B \in \mathbb{R}^{n \times m}, C \in \mathbb{R}^{p \times n}, D \in \mathbb{R}^{p \times m}, n \) is the number of states, \( m \) is the number of inputs, and \( p \) is the number of outputs.

The controller is a Linear Quadratic Regulator (LQR) that is designed to stabilize the system, it employs a Luenberger observer to produce an estimate \( \hat{x}_k \) of the state of the plant at time \( k \), and a state-feedback control law \( K \) to drive the system toward \( w_k \). Specifically, the controller equations are

\[
\begin{align*}
\hat{x}_{k+1} &= A \hat{x}_k + Bu_k + L(\hat{y}_k - \hat{y}) \\
\hat{y}_k &= C \hat{x}_k + Du_k \\
u_k &= -K(w_k - \hat{y}_k) \\
\hat{u}_k &= \text{sat}(u_k)
\end{align*}
\]

According to our threat model, an attacker forges \( a_k \) to divert the output \( y_k \) of the system from its desired value \( w_k \) to reach a dangerous state.

We assume that the plant is linear, time-invariant, controllable, and observable. The discrete-time dynamic equations of the plant under attack are

\[
\begin{align*}
\hat{x}_{k+1} &= A \hat{x}_k + Bu_k + L(\hat{y}_k - \hat{y}) \\
\hat{y}_k &= C \hat{x}_k + Du_k \\
u_k &= -K(w_k - \hat{y}_k) \\
\hat{u}_k &= \text{sat}(u_k)
\end{align*}
\]

where \( \hat{x}_k \) and \( \hat{y}_k \) are respectively the estimated plant state and output, and \( u_k \) and \( \hat{u}_k \) are the control signals respectively produced by the controller and received by the plant, due to saturation levels. The function \( \text{sat}(\cdot) \) saturates the control signal within the values \([u_{\min}, u_{\max}]\).

### B. Attack detection system

State-of-the-art defenses for control systems typically include an Attack Detection System (ADS) that determines whether an attack is occurring [19]. ADS usually employ models of a system that, under the null hypothesis of absence of attacks, receives the same input as the real system, and produces as output a prediction of the system output. The difference between the predicted output and the real one (under the null hypothesis) is due to disturbances like noise and model inaccuracies. If the null hypothesis is false, however, the predicted system output diverges with respect to the measured one, and the difference can be used to detect an ongoing attack. Our controller already includes an observer, that determines \( \hat{y}_k \), see Eq. (3), which we can reuse for attack detection.

Stateful ADS accumulate the difference between the predicted output \( \hat{y}_k \) and the measured output \( y_k \) over multiple periods \( k \), and fire an alert once the cumulative
difference exceeds a specific threshold $\tau$. Figure 1 shows the closed-loop and detection system. Around each block, in gray, we highlight specific block parameters and, in parenthesis, internal block states.

We consider a CUSUM-based ADS in non-parametric form [20], because this formulation can identify anomalies without any a priori hypothesis about a particular attack and attack function [13]. The equations for the detection mechanisms are

\[
\text{Detection} = \begin{cases} 
  r_k = \hat{y}_k - \tilde{y}_k \\
  z_k = \|r_k\| \\
  S_k = \max(0, S_{k-1} + z_k - b) \\
  S_1 = 0 \\
  \text{alarm} = \text{logic}(S_k > \tau)
\end{cases}
\]

where $r_k$ is the residual between the measured output $\hat{y}_k$ and the predicted output $\tilde{y}_k$. Here, $z_k$ is a norm of the residual and $S_k$ is the nonparametric CUSUM statistic at time $k$ (initialized to zero).

Finally, $b \in \mathbb{R}$ is a small nonnegative constant, chosen to implement a forgetting factor. A common choice is $b = d \sigma$, where $\sigma$ is the standard deviation of the output noise, and $d$ is a positive constant determining an arbitrary low percentage of outliers when the attack signal $a = 0$ (e.g., $\approx 0.3\%$ when $d = 2$). For our purposes, we assume there is no output noise and hence we select $b = 0$.

At run-time, an alarm is fired if $S_k > \tau$. When this happens, $S_{k+1}$ is set to 0 and the detection mechanism is reinitialized.

C. State-of-the-art attacks

Cardenas et al. [13] define various (sensor) attack functions. A surge attack is a constant signal, a bias attack is a constantly increasing signal, and a geometric attack is an exponentially increasing signal. Literature studies [13], [21], [14], [22] report that the geometric attack is one of the most effective, i.e., the one that provokes the maximum state deviation. As a consequence, we compare our proposed attack against geometric attacks. A geometric attack consists of defining

\[
a_k = -\beta \alpha^k, \quad k \in \{1, 2, \ldots, h\}
\]

where $\alpha \in \mathbb{R}, 0 < \alpha < 1$, determines the shape of an exponential function, $\beta \in \mathbb{R}^+$ is a positive scaling factor, and $h$ is the duration of the attack. In the literature, the geometric attack is performed on systems that do not use a state estimator [13], and hence can be defined as stealthy by construction. In this case, the attack consumes exactly $\tau$, independently of the values selected for $\alpha$, $\beta$, and $h$. However, this is not the case when a state estimator is included in the controller. To properly compare our proposed attack (Section III) with the best geometric attack, we enforce stealthiness as a constraint in an optimization problem and determine the optimal values for $\alpha$ and $\beta$ (and various values of $h$ among which the optimal one) with respect to the deviation provoked to the plant’s state (see Section IV).

Works like [7] search for the maximum deviation from a stealth attack through an optimization problem without restricting to a particular function (like the geometric attack). However, no works include an explicit model of the actuators’ saturation in the optimization problem.

To the best of our knowledge, the literature selects the attack duration of an attack $h$ using thumb-rule criteria.

III. OPTIMIZATION-BASED ATTACK

In order to study the security of a control system, we want to assess if at least one dangerous state is reachable by a stealth attack, starting from the steady-state condition. The state deviation is a map $D: \mathbb{R}^{n \times 1} \rightarrow \mathbb{R}$ that expresses how much the state $x$ deviates from 0, defined as:

\[
D(x) = \|x\|^2.
\]

We aim at performing a stealth attack, and ensuring that the system state $x$ reaches a deviation as large as possible from the origin, i.e., the attack that maximizes Eq. (6), while remaining undetected.

A. Optimization problems

Figure 2 shows in a graph the influences of an attack $a_k$ for two steps of the dynamic evolution of the plant and controller from Eqs. (2) and (3). For generality, we set up the optimization problem for $h$ steps (i.e., for the entire attack duration).
We aim to find the maximal impact from a stealth attack to sensors. Therefore we search for the attack with duration $h$ maximizing the state deviation with respect to the setpoint. We assume that the attacker has prior knowledge of the system matrices ($A$, $B$, $C$, $D$), the controller parameters ($L$, $K$) and the ADS threshold ($\tau$). Moreover, we assume that when the attack begins the state is on the setpoint. The problem is formulated as:

$$\begin{align*}
\text{maximize} & \quad D(x_{k+h}) \\
\text{subjected to:} & \quad x_{i+1} = A x_i + B \tilde{u}_i, \quad i = k, \ldots, k + h - 1 \\
& \quad y_i = C x_i + D \tilde{u}_i, \quad i = k, \ldots, k + h \\
& \quad \tilde{y}_i = y_i + a_i, \quad i = k, \ldots, k + h \\
& \quad \tilde{x}_{i+1} = A \tilde{x}_i + C \tilde{u}_i + L (r_i), \quad i = k, \ldots, k + h - 1 \\
& \quad \tilde{y}_i = C \tilde{x}_i + D \tilde{u}_i, \quad i = k, \ldots, k + h \\
& \quad u_i = -K \tilde{x}_i, \quad i = k, \ldots, k + h \\
& \quad \tilde{u}_i = \text{sat}(u_i), \quad i = k, \ldots, k + h \\
& \quad r_i = \tilde{y}_i - \tilde{y}_i, \quad i = k, \ldots, k + h \\
& \quad S_i = S_{i-1} + \|r_i\|, \quad i = k + 1, \ldots, k + h \\
& \quad S_k = 0 \\
& \quad S_{k+h} = \tau
\end{align*}$$

(7a)-(7j) are the plant equations, (7d) models the sensor attack, (7e)-(7g) are the controller equations, (7h) models the saturated control signal, and finally (7i)-(7l) model the CUSUM dynamics, setting that the attack should consume the whole available CUSUM, while remaining stealthy. The solution of the optimization problem is the optimal attack sequence $[a_k, \ldots, a_{k+h}]$ of duration $h$, which provides maximum final state deviation, see Eq. (10).

Eq. (7h) models the saturation of the system from Eq. (3), but depending on how it is modeled, the resulting attack can be more or less effective. In the following we discuss two different ways to model the saturation in the optimization problem:

1) **Overflow-prevent** (Opt-P) constraint:

$$\begin{align*}
& \text{subjected to:} \\
& \quad u_{\min} \leq \tilde{u}_i \leq u_{\max}, \quad i = k, \ldots, k + h \\
& \quad \tilde{u}_i = u_i, \quad i = k, \ldots, k + h
\end{align*}$$

The constraint imposes that the attack vector ($a_k, \ldots, a_{k+h}$) by construction ensures that $u$ does not exceed the saturation values (it prevents overflowing the saturations). Such an approach is similar to the way saturations are typically included in optimal control problems, e.g., in Model Predictive Control (MPC).

2) **Overflow-allow** (Opt-A) constraint:

$$\begin{align*}
& \tilde{u}_k = \max(\min(u_k, u_{\max}), u_{\min}), \quad i = k, \ldots, k + h
\end{align*}$$

The constraint does not impose any constraint on $u$, but it clamps $\tilde{u}$, i.e., the attack vector ($a_k, \ldots, a_{k+h}$) may produce $u$ which exceeds the actuators’ saturation values $u_{\max}$ and $u_{\min}$, but only their saturated value is applied to the plant (it allows overflowing the saturations).

For the geometric attack we solve the following optimization problem (G-A):

$$\begin{align*}
\text{maximize} \quad D(x_{k+h}) \\
\text{subjected to constraints} (7b)-(7l), \quad (7h)\text{ is implemented as in Equation (9), and the constraint:} \\
& \quad a_k = -\beta a^{1+h-k}, \quad i = k, \ldots, k + h - 1
\end{align*}$$

(10)

IV. **EXPERIMENTAL RESULTS**

This Section compares the numerical solutions for the maximal impact when overflow-prevent and overflow-allow constraints are employed. More specifically, we compare the impact from Opt-A, Opt-P and the geometric attack (G-A). Results show that the most dangerous attack is discovered with Opt-A, and that both Opt-A and Opt-P are more dangerous than G-A for the considered system.

The simulated plant (in continuous time) is the mass-spring-damper system,

$$\begin{align*}
\begin{bmatrix}
\dot{x}_1(t) = x_2(t) \\
\dot{x}_2(t) = -\frac{K}{M} x_1(t) - \frac{D}{M} x_2(t) + \frac{1}{M} u(t) \\
y(t) = x_1(t)
\end{bmatrix}
\end{align*}$$

with mass $M = 1$ kg, elastic constant $K = 1$ N/m, and damping $D = 0.01$ Ns/m. The model is discretized trough zero-order hold with sampling period $T_s = 0.05$ s. The actuator limits are $u_{\max} = 1$, $u_{\min} = -0.5$. The controller’s parameters obtained with a LQR optimization are $K = [2.015, 3.52]$, and $L = [0.49, 1.06]^T$.

For the ADS we set $\tau = 10$. We employ YALMIP and Gurobi to solve the optimization problems defined in Section III-A. For G-A we optimize Eq. (III-A) using the Matlab function fmincon. As we are dealing with time-invariant systems, we assume without the loss of generality that an attack starts at $k = 0$, i.e., the last sample of an attack is when $k = h$.

**Effects of the attack duration.**

Figure 3 shows on the top diagram the maximal impact as the attack duration $h$ varies, for the two optimization problems Opt-A and Opt-P, and G-A. Each point in the figures is the impact of the optimal attack sequence, given $h$. For both Opt-A and Opt-P we iteratively increase $h$ of a fixed quantity $h_\Delta = 50$, and we use as a stop criteria for $h$ the condition $\|a_0 \ldots a_h\| \approx 0$, because this means that the attack becomes significant only after $k = h_\Delta$ (i.e., the attack is delayed). We noticed that further increasing $h$ results in extremely long computation time without significantly improving the solutions. The bottom diagram of Figure 3 shows the computation time for Opt-A and Opt-P (time to optimize G-A is negligible). We enforce
a maximum computation time of $10^4$ seconds for each optimization problem, which is sufficient to render true $[a_0 \cdots a_h \Delta] \approx 0$.

The results show that both Opt-A and Opt-P have a monotonically increasing impact with the attack duration, while G-A has a maximum when $h = 60$. The G-A has been considered as one of the most effective attacks in the literature [13], [21], [14], [22], and for a short duration of attack ($h < 60$) its maximum impact is the same from Opt-A and Opt-P. However, for longer attack duration, the impact of G-A is orders of magnitude smaller than the one from Opt-A and Opt-P. In particular, Opt-A provides the overall greater maximal when $h = 550$. On the other hand, Opt-P always performs worse than (or at most equal to) the Opt-A. Therefore Opt-A is the preferable attack. In our simulations, a difference in the maximal impact between Opt-A and Opt-P manifests only if the attack duration is long enough (i.e., $h = 300$).

Notice that Opt-A can outperform the Opt-P approach, as we are considering a sensor attack (1), that can influence the observer dynamics directly (3). Decoupling the constraint of $\tilde{u}$ from $u$ means that even $\hat{x}$ and $\hat{y}$ are not constrained, yielding a higher degree of freedom to change $\hat{y}$ in a way that does not increase the CUSUM state, while keep attacking the system.

**Solutions from Opt-A and Opt-P.**

Figure 4 shows the comparison of the results of two optimization problems Opt-A and Opt-P with the highest maximal impact (i.e., $h = 550$, corresponding to 27.5 seconds). Both Opt-A and Opt-P produces a quasi-periodic signal on $\tilde{u}$ having the same frequency of the resonance frequency of the plant, i.e., $1.91 \text{ rad/s}$.

Figures 5 and 6 details the signals respectively from Opt-A and Opt-P. The graphs of the control signal $u$ and $\tilde{u}$ highlight how the choice of the saturation model of Opt-A allows creating a value of $u$ that manages to keep $\hat{y}$ and $\tilde{y}$ close to each other, without significantly increasing the CUSUM $S$. However, Opt-A produces instantaneous increases of $S$ (i.e., $z$) that are an order of magnitude greater than Opt-P. Therefore, a stateless detector could detect more easily an attack from Opt-A (e.g., in the considered system, a stateless threshold of 1 detects the attack of Opt-A at time $2.3s$, while it cannot detect the attack of Opt-P).
Fig. 6. Optimal solution for the Opt-P problem (h = 550), with the mass-spring-damper system.

V. CONCLUSION

In this paper, we present a novel formulation for a maximum impact attack on a cyber-physical control system. We show that parameters like attack duration and the choice of how to implement actuator saturation are crucial variables for determining the maximum impact of a stealth sensor attack. In particular, we compare two optimization-based approaches implementing the saturation function. The optima could be found using trivial stop criteria, i.e., increasing the attack duration and stopping when no further improvement is experienced.

We found that allowing actuators’ saturation (overflow-allow constraints and the corresponding Opt-A problem) results in more dangerous attacks, as shown for a mass-spring-damper system. The second way to implement actuator saturation prevents attack signals to induce corresponding actuator commands that exceed the saturation limits (overflow-prevent constraints and the corresponding Opt-P problem). While Opt-P results in less dangerous attacks, it produces attack sequences that could be harder to detect using a stateless anomaly detection system. The takeaway messages is that security assessment of a control system should be performed using the Opt-A model, while the overall security can benefit by combining a stateful ADS (like the CUSUM) with a stateless ADS.

REFERENCES


