Testbed Evaluation of DoS Attacks on PID-controllers

(Short Paper)

Viktor Tuul and Henrik Sandberg*

KTH Royal Institute of Technology, Stockholm, Sweden. {tuul,hsan}@kth.se

Abstract. We present ongoing work in evaluating the performance of PID-controllers under DoS attacks. The experiments are conducted in a recently developed virtual testbed, which is openly available. An important observation is that also benign physical processes may exhibit potentially dangerous oscillations under DoS attacks unless care is taken in the control implementation. An event-based PID-controller with adaptive gain shows promising performance under DoS attack.

Keywords: Cyber-physical attack · PID-control · DoS attacks

1 Introduction

Cyberattacks on Industrial Control Systems (ICS) and other cyber-physical critical infrastructures is a growing concern in society [1]. Often these infrastructures have lagging cyber-defenses and successful cyberattacks may have devastating physical consequences. As complement to traditional IT-security solutions in these infrastructures, researchers have worked on security solutions tailored towards the cyber-physical nature of these systems. See [8], and references therein. In this vein, we will in this short paper present ongoing work in a virtual testbed for evaluation of cyberattacks and defenses in continuously controlled physical infrastructures.

The testbed has been developed in C# with the purpose of evaluating cyber-physical impact of attacks and defences in controlled critical infrastructures. The testbed uses five main software modules: 1) `Control Center`, 2) `Controller`, 3) `Plant`, 4) `Anomaly Detector`, and 5) `Channel`. Two instantiations of the `Channel` module are the *Supervisory control network* and the *Field communications network*) in Fig. 1. Due to the modular framework and use of UDP/IP-networking, the testbed is simple to deploy on standard computer architectures. The virtual testbed software can be downloaded from GitHub [7].

A contribution of this paper is to highlight the importance of the implementation of low-level controllers with respect to cyberattacks. We focus the tests on DoS attacks [4] in the field communications network, see Fig. 1 (right), and propose a novel event-based PID-controller. Event-based control is not new,

* Work funded by the Swedish Civil Contingencies Agency (project CERCES).

see [2,6], for example, but our design uses an adaptive gain (suppress factor) that explains its improved performance under DoS attacks.

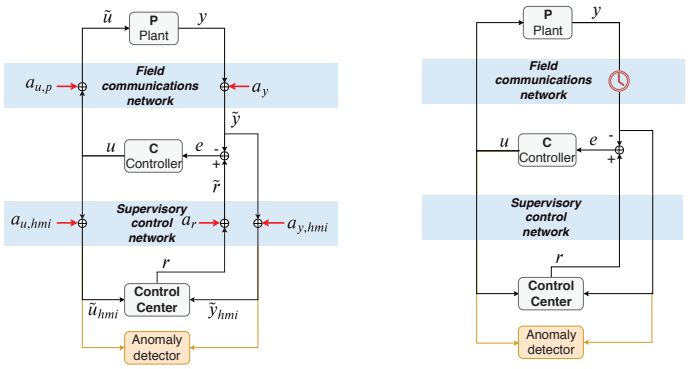

Fig. 1. Block diagrams illustrating the modules of the testbed, data-injection attacks in all channels (left), and DoS attacks on the sensor channel (right, see Section 3).

2 Three PID-controllers

The testbed `Control` module currently supports three PID-controller implementations: PID-, PIDplus-, and PIDsuppress-control. The testbed has been used for evaluating these controllers in various attack scenarios, see Section 3. In particular, the PIDsuppress-controller shows promising results under DoS attacks.

The ideal PID-controller [3] in continuous time is given by

$$u(t) = K_P e(t) + K_I \int_0^t e(\tau)\, d\tau + K_D \frac{de(t)}{dt}, \tag{1}$$

where $e(t) = r(t) - y(t)$ is the control error, $r(t)$ the setpoint, and $y(t)$ is the measured process variable at time t. The parameters K_P, K_I, and K_D are the proportional, integral, and derivative gains, respectively. In the testbed, the signals and controllers are sampled with period $\Delta t_k := t_k - t_{k-1}$, where k is sample number, and e_k denotes the sample $e(t_k)$, for instance.

Time-triggered PID-control. In the time-triggered PID-controller, the integral term is discretized with *uniform* sampling intervals $\Delta t_k = \Delta t$, yielding $\sum_{i=1}^{k} e_i \Delta t$. Backward finite-difference approximation is used to approximate the derivative, $de/dt|_{t_k} \approx (e_k - e_{k-1})/\Delta t$. The sampled controller takes the form

$$u_k = K_P e_k + K_I \sum_{i=1}^{k} e_i \Delta t + K_D \frac{e_k - e_{k-1}}{\Delta t}, \tag{2}$$

and the controller (2) is executed at time instants $t_k = k\Delta t$. If a measurement does not arrive at time t_k, because of DoS attacks, for instance, the last received measurement is used instead.

Event-triggered PIDplus-control. The PIDplus-controller [6] was introduced to better handle packet drops and time-varying sampling periods Δt_k in networked control systems. The PIDplus-controller acts (almost) as a normal PID-controller when there is no packet loss. The proportional, integral, and derivative terms are event based and only updated upon the arrival of a new packet $e_k = r_k - y_k$. The integral term is computed recursively using a filter state F_k,

$$F_k = F_{k-1} + (U - F_{k-1})(1 - \exp(-\Delta t_k / T_{\text{res}})). \tag{3}$$

Here U is the last confirmed actuator setting, and T_{res} is a tuning parameter for the integral action, which can be set as $T_{\text{res}} = K_P/K_I$. The derivative term is given by $D_k = K_D(e_k - e_{k-1})/\Delta t_k$, and the proportional term is $P_k = K_p e_k$. This yields the controller $u_k = P_k + F_k + D_k$.

An advantage with (3) is that after a long communication outage (Δt_k large), $F_k \approx U$, which avoids potentially dangerous wind-up phenomenon and harmful control commands. For comparison, in (2) the integral will increase at rate $e_i \Delta t$ per sample under packet drops.

Event-triggered PIDsuppress-control. The PIDsuppress-controller is a new event-based PID-implementation [7], which adapts *both* the proportional and integral gains depending on sampling period Δt_k. This is achieved using the *suppress*-factor $\gamma_k \in (0, 1]$. The control at time k is

$$u_k = K_P \gamma_k e_k + K_I \sum_{i=1}^{k} \gamma_i e_i \Delta t_i + K_D \frac{e_k - e_{k-1}}{\Delta t_k}. \tag{4}$$

The suppress factor γ_k changes the control characteristics depending on the communication quality (the sampling period). The derivative term does not include γ_k since it is already small for large Δt_k. The factor γ_k is updated using Algorithm 1. The tuning parameter T_{supp} is an order of magnitude larger than the nominal Δt_k, and $\beta \in (0, 1]$ is a low-pass filter constant. In case sampling time increases, Algorithm 1 makes γ_k small immediately. Conversely, γ_k is only slowly increased to ensure a sustained small suppress factor if packets arrive in bursts.

Algorithm 1 Suppress factor γ_k

```
function γ_k(γ_{k−1}, β, Δt_k, T_supp)
    γ̂_k ← exp(−Δt_k/T_supp)
    if γ̂_k ≤ γ_{k−1} then return γ̂_k
    else return (1 − β)γ_{k−1} + βγ̂_k
```

To better understand the role of γ_k, please note that a sampling period Δt_k approximately introduces a time delay $\Delta t_k/2$ in the loop gain transfer function L (assuming zero-order hold sampling). To avoid an unstable or oscillatory feedback system for large Δt_k, one should decrease the system bandwidth (to increase the phase margin). The bandwidth is approximately given by the cross-over frequency ω_c, satisfying $|L(i\omega_c)| = 1$. Around the nominal cross-over frequency $\omega_{c,\text{nom}}$, we have $|L(i\omega)| \approx (\omega_{c,\text{nom}}/\omega)^\alpha$, $\alpha \approx 1.5$, for well-tuned systems. If we include the suppress factor $\gamma_k \leq 1$, we obtain an updated $\omega_c = \omega_{c,\text{nom}} \gamma_k^{1/\alpha}$,

which implies a decreased bandwidth in response to large Δt_k and reduced risk for instability.

3 Testbed Evaluation of DoS Attack

In the following testbed experiments, the plant is a simulated nonlinear cascade of fluid tanks. The measurement y is a fluid level, and the input u controls the fluid inflow by means of a pump. The exact model is available in [7]. The performance of the three control algorithms, PID-, PIDplus-, and PIDsuppress-control, are compared by conducting experiments where the communication between the plant and controller is subject to packet-drop/DoS attacks, see Fig. 1 (right). The DoS attack model and three experiments are described next.

DoS Attack Model. A realistic DoS attack model should include long periods of communication outage. For this, we use a simple two-state Markov chain, namely the Gilbert-Elliott model [5]. The channel state X_k at sample k is dependent on the previous state X_{k-1} according the the state transition probabilities $P_d^p = P(X_k = \text{pass}|X_{k-1} = \text{drop})$, and $P_p^d = P(X_n = \text{drop}|X_{k-1} = \text{pass})$. A topic for future work is to implement more realistic DoS attack models in the testbed, such as the ones listed in [4].

Change in Setpoint. The first experiment is conducted along the following procedure: (1) The setpoint r is changed from 0 to 6 (at Time $\approx$ 2 sec). (2) A DoS attack starts, with the channel state transition probabilities set to $P_d^p = 0.9$ and $P_p^d = 0.1$ (at Time $\approx$ 5 sec). (3) r is changed from 6 to 12 (at Time $\approx$ 6 sec). Fig. 2 shows the step responses. The results show that the three controllers exhibit similar nominal performance when there is no DoS attack (Time $\in [2, 5]$ sec). This is not the case when packet-drop starts (Time $\in [5, 15]$ sec). The time-triggered PID-controller exhibits very undesirable non-decaying oscillations in this phase, potentially causing physical damage. In contrast, in the PIDplus-controller the oscillations are more damped. In the PIDsuppress-controller, the oscillations are almost removed, indicating that this is a promising implementation under both nominal and attack conditions. The explanation is that the PIDsuppress-controller automatically decreases the bandwidth, making the step response slightly slower during attack, and thus significantly decreases the oscillations.

Varying DoS Attack Intensity. Step-response trials as in Section 3 are also conducted with varying attack intensity. For each drop-out setting, five trials are conducted. The results are summarized in Fig. 3 where average step-response settling time under attack is plotted against the drop-out setting (the state transition probabilities P_d^p and P_p^d in %). The average time spent in each channel state before switching, is reported in Table 1.

Table 1. Average time spent in the *pass* and *drop* channel states before switching.

Drop-out setting [%]	100-0	20-80	15-85	10-90	7-93	5-95
Average pass time [sec]	-	0.25	0.24	0.22	0.22	0.21
Average drop time [sec]	-	1.00	1.34	2.00	2.86	4.00

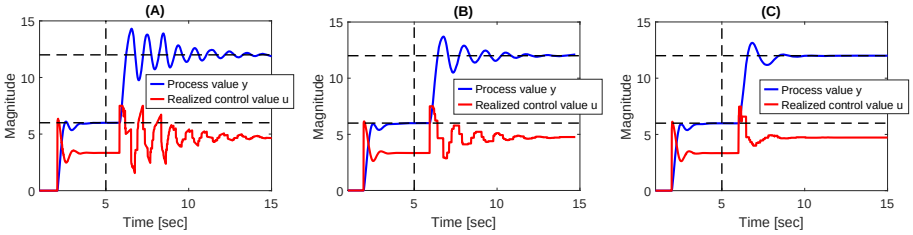

Fig. 2. (A) Time-triggered PID-control. (B) PIDplus-control. (C) PIDsuppress-control. The DoS attack starts at 5 sec and causes various degree of oscillations following a setpoint change.

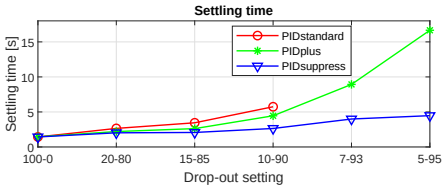

Fig. 3. Average settling time using time-triggered PID-, PIDplus-, and PIDsuppress-control with varying packet drop-out.

The time-triggered PID-controller did not manage to settle the system in any trial with the *7-93* and *5-95* drop-out. It is also seen that the PIDsuppress ensures faster settling time than PIDplus and time-triggered PID-control for all drop-out settings.

Influence of Measurement Noise. An experiment in steady state (no set-point change) but in the presence of measurement noise is also conducted. Fig. 4 shows a case where noise of standard deviation 0.2 is applied to the measured process value. The DoS attack starts at 6.5 sec and the *5-95* drop-out setting is used.

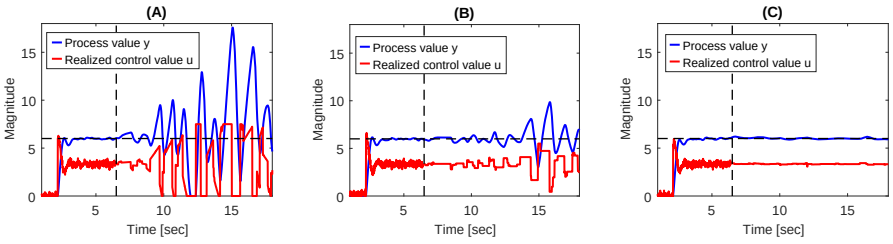

Fig. 4. (A) Time-triggered PID-control. (B) PIDplus-control. (C) PIDsuppress-control. The DoS attack starts at 6.5 sec and causes various degree of oscillations, as triggered by natural measurement noise.

As before, the three controllers exhibit similar performance when there is no DoS attack (Time $\in [1, 6.5]$ sec). When the attack is active (Time $\in [6.5, 18]$ sec), the time-triggered PID and PIDplus-controller both exhibit non-diminishing oscillations, which is not the case for PIDsuppress. The occurrence of oscillations is because the received measurement varies quickly around the correct value due to noise and gets randomly "stuck" above, or below, the setpoint under attack, and the controller reacts accordingly. In the PIDsuppress case, the bandwidth is decreased to the extent that this noise is filtered out. The cost is a slower control system, as already illustrated in the first experiment.

4 Conclusions

We compared the performance of three different PID-controllers under DoS attack using experiments in a recently developed virtual testbed [7]. An event-based implementation (PIDsuppress) with adaptive gain outperformed the other implementations under attack. An important observation is that even a benign physical process (fluid tanks), exhibits potentially dangerous oscillations under DoS attack unless care is taken in the implementation of these low-level controllers.

For future work, the proposed implementations could be evaluated under more realistic DoS attack models [4]. Also more advanced attacks, such as data-injection attacks, Fig. 1 (left), could be similarly evaluated in the virtual test bed.

References

1. Guide to increased security in industrial information and control systems. Swedish Civil Contingencies Agency (MSB) (2014)
2. Årzén, K.E.: A simple event-based PID controller. IFAC Proceedings Volumes **32**(2), 8687 – 8692 (1999), 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July
3. Åström, K.J., Hägglund, T.: Advanced PID control. The Instrumentation, Systems, and Automation Society (2006)
4. Cetinkaya, A., Ishii, H., Hayakawa, T.: An overview on denial-of-service attacks in control systems: Attack models and security analyses. Entropy **21**(2) (2019)
5. Haßlinger, G., Hohlfeld, O.: The Gilbert-Elliott model for packet loss in real time services on the internet. In: 14th GI/ITG Conference-Measurement, Modelling and Evalutation of Computer and Communication Systems. pp. 1–15. VDE (2008)
6. Song, J., Mok, A.K., Chen, D., Nixon, M., Blevins, T., Wojsznis, W.: Improving PID control with unreliable communications. In: ISA EXPO Technical Conference (2006)
7. Tuul, V.: https://github.com/viktortuul/Modular_Control_System (2019)
8. Urbina, D.I., Giraldo, J.A., *et al.*: Limiting the impact of stealthy attacks on industrial control systems. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 1092–1105. CCS '16, ACM, New York, NY, USA (2016)